IBM® Security Identity Manager

Versions 6.0/7.0

Performance Tuning Guide

Note: Before using this information and the product it supports, read the information in “Notices”.

5th Edition

Edition notice

Note: This edition applies to version 6.x/7.x of IBM Security Identity Manager (product number 5724-C34) and to all subsequent releases and modifications until otherwise indicated in new editions.

Table of Contents

About This Publication

The *IBM® Security Identity Manager Performance Tuning Guide* provides information on tuning middleware for IBM Security Identity Manager versions 6.x and 7.x It includes tuning settings for:

- WebSphere® Application Server
- Database servers (IBM DB2®, Oracle)
- Directory servers (IBM Security® Directory Server)
- IBM Security Directory Integrator
- IBM Security Identity Manager Application
- IBM Security Identity Manager Adapters

This edition includes a troubleshooting, best practices, and regular maintenance sections as well. This publication is a working document and is updated as more information becomes available.

Access to Publications and Terminology

This section provides:

- A list of publications in the IBM Security Identity Manager library.
- Links to "Online publications."
- A link to the "IBM Terminology website."

IBM Security Identity Manager Library

The following documents are available in the IBM Security Identity Manager library

- *IBM Security Identity Manager Quick Start Guide*, CF3L2ML
- *IBM Security Identity Manager Product Overview Guide*, GC14-7692
- *IBM Security Identity Manager Scenarios Guide*, SC14-7693
- *IBM Security Identity Manager Planning Guide*, GC14-7694
- *IBM Security Identity Manager Installation Guide*, GC14-7695
- *IBM Security Identity Manager Configuration Guide*, SC14-7696
- *IBM Security Identity Manager Security Guide*, SC14-7699
- *IBM Security Identity Manager Administration Guide*, SC14-7701
- *IBM Security Identity Manager Troubleshooting Guide*, GC14-7702
- *IBM Security Identity Manager Error Message Reference*, GC14-7393
- *IBM Security Identity Manager Reference Guide*, SC14-7394
- *IBM Security Identity Manager Database and Directory Server Schema Reference*, SC14-7395
- *IBM Security Identity Manager Glossary*, SC14-7397

On-line Publications

IBM posts product publications when the product is released and when the publications are updated at the following locations:

IBM Security Identity Manager Knowledge Center
The https://www.ibm.com/support/knowledgecenter/SSRMWJ_6.0.2/com.ibm.isim.doc/kc-homepage.html site displays the information center welcome page for this product.

IBM Security Identity Manger Virtual Appliance Knowledge Center
https://www.ibm.com/support/knowledgecenter/SSRMWJ_7.0.2/com.ibm.isim.doc/kc-homepage.html

IBM Publications Center
The http://www-05.ibm.com/e-business/linkweb/publications/servlet/pbi.wss site offers customized search functions to help you find all the IBM publications you need.

IBM Terminology Website

The IBM Terminology website consolidates terminology for product libraries in one location. You can access the Terminology website at http://www.ibm.com/software/globalization/terminology.

Accessibility

Accessibility features help users with a physical disability, such as restricted mobility or limited vision, to use software products successfully. With this product, you can use assistive technologies to hear and navigate the interface. You can also use the keyboard instead of the mouse to operate all features of the graphical user interface. For additional information, see the topic "Accessibility features for IBM Security Identity Manager" in the *IBM Security Identity Manager Reference Guide*.

Technical Training

For technical training information, see the following IBM Education website at http://www.ibm.com/software/Security/education.

Support Information

IBM Support provides assistance with code-related problems and routine, short duration installation or usage questions. You can directly access the IBM Software Support site at http://www.ibm.com/software/support/probsub.html.

The IBM Security Identity Manager Troubleshooting Guide provides details on topics such as

- What information to collect before contacting IBM Support.
- The various methods for contacting IBM Support.
- How to use IBM Support Assistant.
- Instructions and problem-determination resources to isolate and fix the problem yourself.

Note: The Community and Support tab on the product information center can provide additional support resources.

Chapter 1. Tuning for High-Yield Performance Improvements

Small changes in indexes and memory allocation to the database can yield large performance improvements.

There are several thousand different parameters that you can modify to tune WebSphere Application Server, the IBM Security Identity Manager product, directory servers, and database servers. When setting up an acceptance or production environment, read each topic and tune your systems. The database statistics tuning are a vital part of the IBM Security Identity Manager product performance.

If you are setting up a test environment and want to get started as quickly as possible, focus on the following areas:

- "Adjusting the Java virtual machine size"
  IBM Security Identity Manager, version 6.0, runs on 64-bit JVMs on supported platforms. Using a 64-bit JVM, you can allocate 3 GB or more of memory. You might need to allocate more memory for large (more than 6 million accounts) reconciliations.
- "Configuring Buffer Pools for the IBM Security Identity Manager Database"
  DB2 buffer pools must be large enough so that most table searches can read directly from memory instead of the disk. You can measure this value by looking at the hit ratio for the buffer pools.
- "Updating IBM Security Identity Manager database statistics for DB2 databases"
  DB2 requires statistics on the number of rows in the tables and available indexes to efficiently execute queries. DB2 version 9 can update the statistics automatically, or you can manually update the statistics.
- "Updating IBM Security Identity Manager database statistics for Oracle databases" You must gather and update database statistics at regular intervals. Intervals can be one week to one month on a production IBM Security Identity Manager system or after processing a large amount of data.

Chapter 2. The Initial Tuning

You can implement most tuning in either a new or an existing environment. When tuning the database in a new environment, you must prime your database statistics for better performance.

To prime the statistics, start by loading a small set of users and accounts and updating the database statistics. For DB2, use the RUNSTATS command and the corresponding manual cardinality tuning. Failing to prime the database can result in poor performance or transaction rollbacks.

Consider enabling automatic statistics collection for DB2 versions 9 and 10.

Related tasks

- "Updating IBM Security Identity Manager database statistics for DB2 databases"
  DB2 requires statistics on the number of rows in the tables and available indexes to efficiently execute queries. DB2 version 9 can update the statistics automatically, or you can manually update the statistics.
- "Updating IBM Security Identity Manager database statistics for Oracle databases" You must gather and update database statistics at regular intervals. Intervals can be one week to one month on a production IBM Security Identity Manager system or after processing a large amount of data.
- "Updating Security Directory Server Database Statistics"
  DB2 requires information about the number of rows in the tables and what indexes are available so that it can efficiently fulfill queries.
- "Configuring automatic statistics collection for the IBM Security Identity Manager database"
  Administrators can configure automatic statistics collection so that DB2 automatically updates database statistics. Automatic collection eliminates the necessity of manually running a periodic statistics collection against the database.

Chapter 3. Resource Allocation

Use the correct tuning values for the memory, processor, and disk resources to avoid over allocating them.

Managing tuning values becomes more complex when more than one middleware component run on the same system. An example is running the IBM Security Identity Manager(ISIM) server, DB2, and Security Directory Server (SDS) all on the same server. **It is recommended to isolate each middleware component on a separate system.** An example is running ISIM server, DB2, and SDS all on separate machines.

Regardless of configuration, you must calibrate the following resources:

- "Allocating memory"
- "Allocating processor usage"
- "Allocating disk space for storage"

Allocating Memory

You can adjust how much memory middleware components use. When calculating how to allocate memory to middleware components, keep in mind the following considerations.

- If middleware memory settings are too high, the operating system might swap out memory to disk if the physical memory is exceeded. Memory swapping results in poor performance. After setting up or changing the memory values, monitor the memory and swap space. If anything is swapped out to disk, readjust the settings to correct the problem.
- IBM Security Directory Server has internal caches that contribute to the size of their processes. When the LDAP server reaches the designated limit, it refuses new connections and fails. For IBM Security Directory Server, the **entry cache** size limit determines the **number of entries in the cache**, not the size of the cache. The size of each cache entry varies based on the IBM Security Identity Manager configuration and any extensions to the base IBM Security Directory Server schema. In rare cases, the default cache values might exceed the designated limit.
- **Buffer pools** account for a large amount of the memory used by IBM DB2. The application control heaps, the sort heaps, and the statement heaps also use memory. In addition to database-wide memory heaps, each database connection results in memory allocations. Do not overlook these per-connection memory requirements when computing how much memory to allocate to IBM DB2.
- A large part of the WebSphere Application Server memory usage is the JVM size. The size of the JVM does not set an upper bound on the amount of memory that the WebSphere Application Server uses.
- Operating system limits can prevent processes from accessing all available memory. Confirm the appropriate ulimit values for your system to ensure that they do not artificially limit the amount of memory available. Determine the limits by using ulimit -a. Increase memory and file limits to high or unlimited values before starting IBM Security Identity Manager or related middleware.

Related tasks

- "Tuning IBM WebSphere Application Server"
  Regardless of the installation type (single server or cluster), you can think of the IBM Security Identity Manager server as two components: WebSphere Application Server (the J2EE application server that is running the application) and the IBM Security Identity Manager application itself. You must tune both components.

Allocating Processor Usage

All IBM Security Identity Manager components are processor-intensive so you must consider how to manage CPU for optimum performance. Both IBM Security Directory Server and IBM DB2 are multi-threaded (and multi-process in the case of DB2 applications) that show optimum performance on a multiprocessor server. Even in a well-tuned environment the system bottlenecks can occur. It is critical to take into account *network latency, the processor, available memory, and disk I/O* on the IBM Security Identity Manager server, the

directory server, and the database server.
Based on results in the lab, Security Performance recommends deploying the IBM Security Identity Manager server, the directory server, and the database server on *separate servers* as it will improve performance and will also facilitate the process of troubleshooting performance bottlenecks.
If separate servers are not possible, put the database and directory server on a server with a high performance disk configuration. Multiple disks and a high performance RAID configuration provide fast read and write capacity. ISIM Performance labs have tested with DB2 and ISDS on RAID 5 configuration with good performance results.

Allocating Disk Space for Storage

Each middleware component uses different amounts of disk space for various purposes.

- WebSphere Application Server and the IBM Security Identity Manager application use disk space beyond their installation size because of log files. These log files include the msg.log and trace.log files. Adjust the number of archives and size of the msg.log and trace.log files in the enRoleLogging.properties file.
- IBM Security Directory Server uses disk space from both the IBM Security Directory Server process (log files like ibmslapd.log and audit.log) and the IBM DB2 database. IBM Security Directory Server uses system-managed space (SMS) table spaces so that the system can manage the amount of disk space used. You cannot specify the upper boundaries of SMS table spaces, so you monitor the amount of disk space used to prevent the drive from becoming full.
- The IBM Security Identity Manager DB2 database uses directory-managed space (**DMS**) table spaces. These table spaces require manual allocation of disk space for the database. IBM Security Identity Manager enables auto-resize on these DMS table spaces so they can grow as needed.
- In addition to the table spaces for the database data, IBM DB2 uses disk space for the **transaction logs**. Configure the transaction logs to ensure enough disk space for the log files.
- IBM Security Identity Manager creates Oracle data files so that they can grow as needed.

Related tasks
- "Configuring Transaction Logs for the Security Directory Server Database"
  DB2 keeps logs during transaction processing. During large transactions, the default log number and sizes might be too small and cause transaction rollbacks. Increase the size and number of log files available to DB2.
- "Configuring Transaction Logs for DB2 databases"
  DB2 keeps logs during transaction processing. During large transactions, the default log number and sizes might be too small and cause transaction rollbacks. Increase the size and number of log files to resolve this issue.
- "Configuring Table Spaces for IBM DB2 Databases"
  IBM Security Identity Manager uses a database managed space (DMS) table space to store data. This type of table space performs better than system managed space (SMS) table spaces, but you must pre-allocate disk space for the database to use. The tables spaces created by the installer have auto-resize enabled and grow as needed.
- "Configuring Open Cursors"
  IBM Security Identity Manager uses prepared statements through the WebSphere Application Server JDBC interface. Each prepared statement requires an open cursor in Oracle. If you receive an error message about too many open cursors, you can increase the maximum number of open cursors.

Chapter 4. Upgrading From a Previous Version

If you upgrade from a previous version of IBM Security Identity Manager, consider the following tasks:

- Disable the IBM Security Identity Manager recycle bin if it is not required for your environment.
- Consider changing the IBM Security Identity Manager database table spaces to use autoresize.
- Take advantage of IBM DB2, version 9, features, such as the self-tuning memory manager and automatic runstats.

Related concepts

"Using the recycle bin"

When you enable the recycle bin and then delete objects from IBM Security Identity Manager, the software moves them to the recycle bin.

Related tasks

- "Enabling Automatic Resizing of Table Spaces"
  Enable automatic resizing so that containers for a table space can grow automatically if they become full.
- "Enabling the Self-Tuning Memory Manager"
  The self-tuning memory manager removes the guesswork in determining the memory values for areas such as buffer pools, the sort heap, and the package heap. With self-tuning memory enabled, DB2 can move memory between areas based on system need. DB2 versions 9 and 10 databases have the self-tuning memory manager enabled by default.
- "Configuring automatic statistics collection for the IBM Security Directory Server Database"
  Administrators can configure automatic statistics collection so that DB2 automatically updates database statistics. Automatic collection eliminates the necessity of manually running a periodic statistics collection against the database.
- "Configuring automatic statistics collection for the BM Security Identity Manager Database"
  Administrators can use automatic statistics collection so that DB2 automatically updates the necessary database statistics. Automatic collection eliminates the necessity of manually running a periodic statistics collection against the database.

Chapter 5. Tuning IBM WebSphere Application Server

Regardless of the installation type (single server or cluster), you can think of the IBM Security Identity Manager server as two components: WebSphere Application Server (the J2EE application server that is running the application) and the IBM Security Identity Manager application itself. You must tune both components.

WebSphere Application Server provides various settings for tuning the environment.

Related information
WebSphere Application Server product documentation

Adjusting the Java Virtual Machine Size

IBM Security Identity Manager, version 6.0, runs on 64-bit JVMs on supported platforms. Using a 64-bit JVM, you can allocate 2 GB or more of memory. You might need to allocate more memory for large (more than 6 million accounts) reconciliations.

About this task

For cluster installations, IBM Security Identity Manager uses two application servers per node: one for the application and one for the messaging (JMS) engine. All JVMs in this topic are used by the application, not the messaging engine. The JVM used by the messaging engine application server uses default values.

The IBM Security Identity Manager regular installer sets the maximum JVM size to 1024 MB and the initial size to 512 MB. These values are adequate for most small and medium systems. If your server has available RAM, for 64-bit JVMs, increase the initial JVM size to 1024 MB and maximum JVM size to 4096 MB.

The IBM Security Identity Manager Launchpad Single-server installer sets the maximum JVM size to 512 MB. This size is adequate for Proof Of Concept and demonstration environments. Increase the maximum JVM size if you have adequate memory.

Important:
The maximum heap size on 64-bit JVMs can be much higher than 2 GB. *A larger value larger can result in long delays during full garbage collections.* Do not set the maximum JVM size higher than necessary. **Typically, 4 GB is an adequate maximum.** It is more challenging to troubleshoot WebSphere Application Server issues if the JVM heap size is set to values higher than 4096. Thread analysis activities such as heapdump and javacore generation become increasingly difficult to manage as the heap is increased (larger file sizes, restriction on mechanism to create files).

Do not set the JVM heap size larger than the physical RAM. The WebSphere Application Server suffers significant performance degradation if the operating system swaps out the JVM to swap space. Setting the heap size larger than the physical RAM can cause slow user interface (UI) performance, transaction rollbacks, timeouts, and high disk utilization.

Use the following parameters to set the JVM heap size:

initial_jvm_heap_size	Specifies the initial size of the JVM heap in megabytes. Use 1024 MB for 64-bit JVMs
max_jvm_heap_size	Specifies the maximum size of the JVM heap in megabytes. Use 4096 MB for 64-bit JVMs

Procedure

1. Open the WebSphere Integrated Solutions Console.

2. Expand the **Servers** list.
3. Select **Application Servers**.
4. Select the application (not JMS) server to manage.
5. Expand the **Java and Process Management** list under the **Server Infrastructure** pane.
6. Select **Process Definition**.
7. Select **Java Virtual Machine** from the **Additional Properties** pane on the right.
8. Set the **Initial Heap Size** with *initial_jvm_heap_size*.
9. Set the **Maximum Heap Size** with *max_jvm_heap_size*.
10. Click **OK**.
11. Save the settings to the master configuration.
12. Repeat this procedure for each IBM Security Identity Manager server.
13. Restart all application servers for the changes to take effect.

Related information

- Understanding the IBM Java Garbage Collector - Find out how objects are allocated in the Java heap for garbage collection.
- Tuning Garbage Collection with the Sun Java Virtual Machine - See information about the general features of the Sun JVM garbage collection and tuning options to take the best advantage of those features.
- Sun Java HotSpot VM Options - See information about typical command-line options and environment variables that can affect the performance characteristics of the Java HotSpot Virtual Machine.

Configuring WebSphere Performance Monitoring Infrastructure

Disable or adjust Performance Monitoring Infrastructure to prevent performance degradation for the administrative console.

About this task

By default, WebSphere Application Server has the Performance Monitoring Infrastructure (PMI) enabled and set at the Basic level. At this level, URIRequestCount and URIServiceTime monitoring is enabled. Enabling both parameters causes performance problems when using the administrative console because unique URLs are generated. To prevent performance degradation, disable Performance Monitoring Infrastructure entirely or disable these specific flags.

Note: For cluster installations, IBM Security Identity Manager uses two servers per node: one for the application and one for the messaging (JMS) engine. All changes in this topic are used by the application, not the messaging engine. The messaging engine application server uses default values.

Tip: Consider disabling Performance Monitoring Infrastructure entirely unless you are actively pursuing a performance-related problem.

Procedure

1. Open the WebSphere Integrated Solutions Console.
2. Expand the **Monitoring and Tuning** list.
3. Select **Performance Monitoring Infrastructure (PMI)**.
4. Select the server you want to manage.
5. Take one of the following actions:
   - To disable PMI entirely, clear **Enable Performance Monitoring Infrastructure (PMI)**.
   - To disable just the URIRequestCount and URIServiceTime counters:
     a. Select **Custom**.
     b. Select **Web Applications** from the tree listing.
     c. Select the check box next to **URIConcurrentRequests**.
     d. Select the check box next to **URIRequestCount**.
     e. Select the check box next to **URIServiceTime**.

f. Click **Disable** at the top of the pane.

6. Save the settings to the master configuration.
7. Repeat this procedure for each IBM Security Identity Manager application server.
8. Restart all application servers for the changes to take effect.

Configuring WebSphere JDBC Connections

IBM Security Identity Manager server uses JDBC connections from WebSphere Application Server to communicate with the database.

About this task

The JMS architecture in WebSphere, versions 7.*x and 8.x*, provides IBM Security Identity Manager, version 6.0, an additional JDBC data source for the JMS cluster members database connectivity. This Bus data source requires database connections in addition to those connections required for the application cluster members.

The number of connections from the application server to the database depends on the needs of the application. The maximum connection values are set independently on each application server. Typically, you do not need to increase the maximum connection values from the following default values. The value for the ISIM data source will need to be increased if 50 or more concurrent users are managing accesses on the *Identity Service Center UI*

- 30 (IBM Security Identity Manager Bus data source)
- 30 (IBM Security Identity Manager Bus Shared data source)
- 50 (IBM Security Identity Manager data source)

Decrease the number of connections if the database cannot service all the concurrent requests due to resource limitations.

Use the following parameters to configure JDBC connections:

bus_data_source_size	Specifies the maximum JDBC pool size of the IBM Security Identity Manager Bus data source. Initial value: 30.
bus_shared_data_source_size	Specifies the maximum JDBC pool size of the IBM Security Identity Manager Bus Shared data source. This number is allocated by each cluster member. Initial value: 30.
data_source_size	Specifies the maximum JDBC pool size of the IBM Security Identity Manager data source. This number is allocated by each cluster member. Initial value: 50. (This value will need to be increased if 50 or more concurrent users are managing accesses on the *Identity Service Center UI*) New recommended value: 75

Procedure

1. Open the WebSphere Integrated Solutions Console.
2. Expand **Resources**.
3. Expand the **JDBC** list.
4. Select **Data sources**.
5. Select the data source to update.
6. Select **Connection pool properties** from the **Additional Properties** pane.
7. Set the **Maximum connections** to the corresponding value for the data source selected.
8. Click **OK**.
9. Save the settings to the master configuration.

10. Repeat this procedure for each data source you want to change.
11. Restart all application servers for the changes to take effect.

Performance Implications for Java 2 Security

Java 2 Security can degrade system performance on specific WebSphere Application Server versions. WebSphere Application Server versions before version 6.1.0.9 had a significant performance penalty when Java 2 Security was enabled. For version 6.1.0.9 and later, you can enable Java 2 Security with minimal performance impact after setting some system properties. If you do not need Java 2 Security, disable it.

Tuning of WebSphere Application Server Thread Pools

A thread pool enables components of WAS application to reuse threads. Reusing threads instead of creating new threads saves time and resources. A thread pool could be considered as a queuing mechanism used to throttle active requests concurrently running at any given time in the ISIM application. It is best to apply a 'funnel based approach' to sizing various thread pools.
Ex: IHS (400) -> WAS (50) -> WAS DB connection pool (30)
Per WebSphere documentation, in most typical configurations, applications need 10 or fewer threads per processor In a clustered environment, tune each application server in the cluster.

Thread Pools

Using the WebSphere Administrative Console, workflow processes and EJB requests can be tuned and balanced. The maximum number of threads must be set so that the CPU is less than **80%** under load. It is important to take other resources into account, such as network, database server, CPU/disk IO, and LDAP CPU/ disk IO. These resources must also follow the same rule of less than **80%** under load. Resources must be monitored during the testing period.

Object Request Broker Thread Pool

EJB calls (client API) are processed by the Object Request Broker (ORB) thread pool of the application server. You must specify minimum and maximum number of threads in the pool.
WebSphere application server clusters > Application_Cluster > Cluster members > application_clusterMember_name > Thread pools

ORB Thread pool Minimum Size: 10 Threads
ORB Thread pool Maximum Size: 50 Threads

Activation Specifications

A JMS activation specification is associated with one or more message-driven beans and provides the configuration necessary for them to receive messages. JMS activation specifications are used to bind application MDBs to the message queues which drive the MDB onMessage methods. Two activation specification parameters of particular interest are the maximum batch size and the maximum concurrent MDB invocations per endpoint.
Resources > JMS > Activation Specifications

There are total of 10 Activation Specifications in ISIM6.
Excluding *ITIMLocalWorkflowActivationSpec* (1, 50),
ITIMMailServicesActivationSpec (1, 3) and
ITIMImportExportActivationSpec (1, 2), the other 7 activation specifications will be tuned as follows:
(*max_batch_size, max_concurrent)*

Maximum batch size: 1
Maximum concurrent MDB invocations per endpoint : 5
Retry interval : 30 seconds

Web Container Thread Pools

Application servers > application_server_Name > Thread pools
Threads in the Web Container thread pool are used for handling incoming HTTP and Web Services requests. This thread pool is shared by all applications deployed on the server and must be tuned to a higher value than the default. The web container thread pool is the most common bottleneck in an application environment. If you adjust the number of threads too low, the web server threads can wait for the web container. If you adjust the number of threads too high, the server can be inundated with too many requests. For both situations, the consequence is an increase of the response time.

WebContainer Minimum Size: 50
WebContainer Maximum Size: 50
WebContainer Thread inactivity timeout: 60000 ms

Message Listener Service Thread Pool

Application servers > application_server > Message listener service > Thread Pool
The message listener service maintains a pool of threads to process messages through MDB. You can specify minimum and maximum number of threads in the pool.
Note: All listener ports in that server use threads from the same thread pool.

Message Listener Service thread pool minimum size: 10
Message Listener Service thread pool maximum size: 50
Message Listener Service thread inactivity timeout: 3500 ms

Chapter 6. Tuning IBM HTTP Server

The WebSphere Application Server uses the IBM HTTP Server as a front-end server in a single-server installation. It uses the IBM HTTP Server as a load balancer between nodes in a cluster installation. Small and medium configurations can typically use default configuration parameters for the IBM HTTP Server. Larger ISIM configurations will need to increase certain IHS parameters to compensate for concurrency.

Related information
IBM HTTP Server product documentation

Optimizing IBM HTTP Server Connections

You can set the number of connections that the IBM HTTP Server accepts at one time. The default value might be too small if the servers experience many concurrent users.

About this task

The IBM HTTP Server supports the HTTP/1.1 KeepAlive request that allows a client to make multiple HTTP requests through a single persistent connection. A single connection can accept only a limited number of KeepAlive requests. After reaching this limit, the connection closes, and another connection must be established. The default value might be too small for some external provisioning processes, such as a Java Naming and Directory Interface (JNDI) feed.

Use the following parameters to optimize server connections:

ibmhttp_home	Specifies the home directory of the IBM HTTP Server, such as /usr/IBMHttpServer.
max_clients	Specifies the maximum number of connections that can be made to the HTTP server at one time. Set this parameter to the maximum number of concurrent users you expect on your system.
max_keepalive	Specifies the maximum number of requests for a single connection.
threads_per_child	Sets the number of threads created by each child process. The child creates these threads at startup and never creates more. The total number of threads should be high enough to handle the common load on the server.
thread_limit	Sets the maximum configured value for ThreadsPerChild for the lifetime of the Apache process. Do not set this parameter any higher than your greatest predicted setting of ThreadsPerChild for the current run of Apache.
timeout	The number of seconds before receives and sends time out

Tip: The MaxClients parameter on Windows is called ThreadsPerChild. You might need to adjust the ServerLimit, ThreadLimit, and ThreadsPerChild parameters on UNIX systems when adjusting the MaxClients parameter. For more information, see the IBMF HTTP Server documentation.

Procedure

1. Edit the *ibmhttp_home*/conf/httpd.conf file and update the following entries:
   MaxClients *max_clients*
   MaxKeepAliveRequests *max_keepalive*
   ThreadsPerChild threads_per_child
   ThreadLimit thread_limit
   Timeout timeout
2. Stop and restart the IBM HTTP Server for these changes to take effect.

Enabling Content Compression for the IBM HTTP Server

The IBM HTTP Server shipped with WebSphere Application Server includes the mod_deflate plug-in. Use this plug-in to compress pages before sending them to the client.

About this task

Typically, the mod_deflate plug-in yields better results for administrative console users, particularly if they are not on the same LAN as the IBM Security Identity Manager server. Enabling the mod_deflate plugin for the Self-Service interface is not necessary due to the smaller size of returned pages. Enabling it can increase page response times.
Use these parameters to compress pages:

ibmhttp_home	Specifies the home directory of the IBM HTTP Server, such as /usr/IBMHttpServer.
itim_console_location	Specifies the base URL of the IBM Security Identity Manager Console application, such as /itim/console.

Procedure

1. Edit the *ibmhttp_home*/conf/httpd.conf file and add the following lines:
   Note: The <Location *itim_console_location*> stanza must include the *itim_console_location* value.

```
LoadModule deflate_module modules/mod_deflate.so

# Compress content for ITIM Administrative Console interface.
#
#Requires modules:
#    LoadModule deflate_module modules/mod_deflate.so
<Location itim_console_location>
     # All ITIM supported web browsers correctly declare their support for
     # compressed content via Accept-Encoding so we don't Vary the compression
     # based on User-Agent.

     # Insert filter for compression
     SetOutputFilter DEFLATE

     # Don't compress images
     SetEnvIfNoCase Request_URI \.(?:gif|jpe?g|png|ico)$ no-gzip
</Location>
```

2. Stop and restart the IBM HTTP Server for these changes to take effect.

Improving the Caching of Static Content Served From the IBM HTTP Server

You can use the IBM HTTP Server to improve the caching of static content both in the browser and in any intermediate caching proxies.

About this task

You can improve the use of caches for static content between the user and the WebSphere Application Server instance. Adjust the Expire and Vary headings from the IBM HTTP Server.

If you set the Expire header to the distant future, caches can store and serve the unchanging static content without refreshing it from the WebSphere Application Server. Removing the Vary header from images

instructs caching proxies to serve the images from the cache, irrespective of the browser User-Agent. Removing the
Vary header increases cache hits and improves overall interface performance.

Use these parameters in the following procedure:

ibmhttp_home	Specifies the home directory of the IBM HTTP Server, such as /usr/IBMHttpServer.
itim_location	Specifies the base URL of the IBM Security Identity Manager application, such as /itim.

Procedure

1. Edit the *ibmhttp_home*/conf/httpd.conf file and add the following lines:
   Note: The <Location *itim_location*> stanza must include the *itim_location* value.

```
LoadModule headers_module modules/mod_headers.so
LoadModule expires_module modules/mod_expires.so

# Ensure static content is cached by the browser and intermediate proxies
# as efficiently as possible. This applies both to the Administrative Console
# as well as the Self-Service interface.
# Static content includes images (gif/jpeg/png/ico), stylesheets (css) and
# Javascript files (js).
#
#
Requires modules:
#   LoadModule headers_module modules/mod_headers.so
#   LoadModule expires_module modules/mod_expires.so
<Location itim_location>
    # Set the Expires header for static content to +1 month
    ExpiresActive On
    ExpiresByType image/gif "access plus 1 month"
    ExpiresByType image/jpeg "access plus 1 month"
    ExpiresByType image/png "access plus 1 month"
    ExpiresByType image/x-icon "access plus 1 month"
    ExpiresByType text/css "access plus 1 month"
    ExpiresByType application/x-javascript "access plus 1 month"

    # Don't Vary image content at all. This allows caching proxies to cache
    # the images once and serve it to all browsers. Note we can't include
    # css/js content in case some browsers request compressed content and
    # some don't. The mod_deflate plugin will automatically set a Vary
    # header against Accept-Encoding, we just need to not override it.
    # Caching proxies will still cache css/js content but will cache
    # two or more copies and serve them accordingly.
    SetEnvIfNoCase Request_URI \.(?:gif|jpe?g|png|ico)$ dont-vary
    Header unset Vary env=dont-vary
</Location>
```

2. Stop and restart the IBM HTTP Server for these changes to take effect.

Edge Side Include Caching

The IBM HTTP plug-in interfaces with single-server WebSphere environments and balances GUI requests in clustered WebSphere environments. The plug-in has built-in support for Edge Side Include (ESI), which does page- and fragment-level caching. Edge Side Include does more than caching static content, but IBM Security Identity Manager primarily uses it to cache images, JavaScript, and CSS files. The HTTP plug-in enables Edge Side Include by default with a cache size of 1024 KB and a cache timeout value of 300 seconds (5 minutes).

Configuring the Edge Side Include Cache Size

If both the administrative console and the Self-Service GUI are heavily used, increase the size of the Edge Side Include cache.

About this task

The approximate size of the static content from the administrative console is 825 KB and for the Self-Service GUI is 550 KB. Static content includes images, JavaScript, and CSS. If the bulk of traffic is from one or the other GUI, the default 1024 KB cache is adequate to completely cache the static content. If you have both kinds of traffic, you can increase the cache size to 1536 KB or 2048 KB. An example of both kinds of traffic is when users service their own requests for password changes and help desk personnel do account maintenance.

Important: Use caution when increasing this value. There is one ESI cache per HTTP process. The total memory used by the ESI cache is cache_size * num_HTTP_processes.

ibmhttp_home	Specifies the home directory of the IBM HTTP Server, such as /usr/IBMHttpServer.
pluginxml_file	Specifies the name of the configuration file for the HTTP plug-in. You can find the name in the WebSpherePluginConfig parameter in the IBM HTTP Server configuration file. The configuration file is at *ibmhttp_home*/conf/httpd.conf.
cache_size	Specifies the size (in kilobytes KB) of the ESI cache. If you use both the administrative console and the Self-Service GUI, set the value to 2048. Default value: 1024.

Procedure

1. Edit *pluginxml_file*.
2. Locate the line for the ESIMaxCacheSize value. For example:
   <Property Name="ESIMaxCacheSize" Value="1024"/>
3. If the ESIMaxCacheSize line does not exist, add it.
4. Set the value to cache_size:
   <Property Name="ESIMaxCacheSize" Value="*cache_size*"/>
5. Save and exit the file.
6. Stop and restart the IBM HTTP Server for these changes to take effect.

Configuring the Edge Side Include Cache Timeout

The Edge Side Include cache timeout controls how long an entry can exist in the cache before it expires. Adjusting length of this timeout might improve performance under load.

About this task

By default entries are valid in the Edge Side Include cache for 5 minutes. After 5 minutes, the data expires, and a subsequent request is passed back to WebSphere.

A busy environment in which new users access IBM Security Identity Manager every few minutes results in requests to WebSphere every 5 minutes.

IBM Security Identity Manager static content does not change every 5 minutes. You can change the timeout value to higher value, like one hour. Add the following parameter to the JVM command line for the application servers that run IBM Security Identity Manager.

cache_timeout	Specifies the Edge Side Include cache timeout value in seconds. To set it for one hour, use 3600. Default value: 300.

Procedure

1. Open the WebSphere Integrated Solutions Console.
2. Expand the **Servers** list.
3. Select **Application Servers**.
4. Select the application server you want to manage. Do not select the JMS server.
5. Expand the **Java and Process Management** list under the **Server Infrastructure** pane.
6. Select **Process Definition**.
7. Select **Java Virtual Machine** from the **Additional Properties** pane.
8. Add the following line to the **Generic JVM arguments**:
   -Dcom.ibm.servlet.file.esi.timeOut=*cache_timeout*
9. Click **OK**.
10. Save the settings to the master configuration.
11. Repeat this procedure for each IBM Security Identity Manager server.
12. Restart all application servers for the changes to take effect.

For further information, consult the IHS Tuning document:
http://publib.boulder.ibm.com/httpserv/ihsdiag/ihs_performance.html

Chapter 7. Tuning IBM Security Identity Manager

Tune IBM Security Identity Manager by adjusting values in configuration files and through the administrative console.

IBM Security Identity Manager includes several configuration files for tuning various parts of the application performance. These files are in the data/ directory in the IBM Security Identity Manager home directory.

Related information
- IBM Security Identity Manager Product Documentation
- IBM Security Identity Manager wiki

Configuring LDAP Connection Pooling

IBM Security Identity Manager can reuse LDAP connections to the directory server to decrease the performance impact of establishing new connections.

About this task

IBM Security Identity Manager can use LDAP connection pooling to communicate with the LDAP data store. A single connection consists of a bind; an operation, such as a search, add, modify, or delete; and an unbind. Connection pooling improves performance by allowing multiple LDAP operations to reuse a single connection with the same bind credentials. The single connection eliminates the performance impact of bind and unbind.

You can enable connection pooling for non-SSL connections (plain), SSL connections, or both. By default IBM Security Identity Manager is configured to pool only plain connections. Establishing SSL connections can cause a significant performance impact. For environments that use SSL to connect to the directory server, configure the server to pool the SSL connections.

Procedure

1. Edit the enRole.properties file and change the following property:
   enrole.connectionpool.protocol=plain ssl
2. In a clustered environment, edit the enRole.properties file on each node.
3. Restart the IBM Security Identity Manager application for this value to take effect.

Configuring List Controls

The ui.properties file has several parameters. These parameters control how many entries, such as viewing the people in an organizational unit, and how many pages are in a list.

About this task

Setting the following values too high can result in Java OutOfMemory errors due to heap fragmentation.

isim_home	Specifies the home directory for IBM Security Identity Manager, such as /opt/IBM/isim.
page_size	Specifies the number of entries to show on a page. Default value: 50.
page_link_max	Specifies the number of pages a user can access for a single search. Default value: 10.
max_search_results	Specifies the maximum number of results to return from a search. Increasing this value can result in heap fragmentation issues. Always

	make this value equal to or greater than (*page_size* * *page_link_max*). If (*page_size* * *page_link_max*) is larger than 1000, decrease one of the two parameters until the product is less than 1000. Default value: 1000.

Change the values on all nodes in a clustered environment.

Procedure

1. Edit the *isim_home*/data/ui.properties file and change the following properties:
   enrole.ui.pageSize=*page_size*
   enrole.ui.pageLinkMax=*page_link_max*
   enrole.ui.maxSearchResults=*max_search_results*
2. In a clustered environment, edit the *isim_home*/data/ui.properties file on each node.
3. Restart the IBM Security Identity Manager application for these values to take effect.

Configuring Report Data Synchronization

You must synchronize data before you can generate reports against IBM Security Identity Manager. Synchronizing report data pulls configuration and user information from the configured LDAP and uses it to populate the database.

About this task

This task synchronizes the data and the related Access Control Information (ACIs). If only IBM Security Identity Manager administrators create reports, populating the ACI data when synchronizing report data is not required. ACI data is not applied to reports generated by administrators. Similarly, if you use IBM Security Common Reporting to generate reports, populating the ACI data when synchronizing report data is not required. IBM Security Common Reporting does not enforce ACIs during report generation. Disabling ACI synchronization can improve report data synchronization performance by an order of magnitude. Improvement depends on the number and complexity of the configured ACIs and the structure of your organizational tree. The *availableForNonAdministrators* parameter in the *adhocreporting.properties* file controls enabling and disabling ACI synchronization. Setting the parameter to true synchronizes ACIs. After disabling report data synchronization with ACI data, non-administrators cannot generate reports from IBM Security Identity Manager. Disabling ACI synchronization for IBM Security Identity Manager, version 6.0, requires IF23 or later.

IBM Security Identity Manager, version 6.0, IF17 reduced the completion time for ACI data synchronization by half for most systems. Make sure that you are at this level or higher if the population of ACIs data is necessary.

synchronize_ACIs	Specify true to enable ACI synchronization and false to disable it. Default value: true. These values are case-sensitive.

Procedure

1. Edit the adhocreporting.properties file and change the following property:
   availableForNonAdministrators=synchronize_ACIs
2. In a clustered environment, edit the adhocreporting.properties file on each node.
3. Restart the IBM Security Identity Manager application for this change to take effect.

Improving Report Data Synchronization Performance

This topic provides information about optimizing performance for Security Identity Manager report data synchronization.

Mapping only the required number of entity attributes
Minimize the number of entity attributes that are synchronized during the data synchronization activity.

Mapping only the required attributes improves the performance of data synchronization by:
- Reducing the volume of data retrieved from the directory server.
- Reducing the data written to the report tables in the Security Identity Manager relational database.

Users can configure the number of entity attributes synchronized by using the schema mapping feature.

For more information about schema mapping, see
http://www-01.ibm.com/support/knowledgecenter/SSRMWJ/welcome

Configuring the commitFrequency property
The *commitFrequency* property setting in the *adhocreporting.properties* file determines how frequently the database updates are committed during data synchronization. The default value is zero (0). The default value indicates that the database updates are committed once at the end, after each mapped entity and ACI information is processed for each entity. The recommended value is **1000**. *Avoid setting a larger value, as it might increase data loss in the event of an error.*

To configure the property, complete these steps:

1. Open the adhocreporting.properties file in *ITIM_HOME*/data.
2. Edit the adhocreporting.properties file and set the value for the property commitFrequency=1000.
3. For the clustered environment only, edit the adhocreporting.properties file on each node.
4. Restart the Security Identity Manager application.

Configuring Java heap size while running the stand alone report data synchronization utility
Before running the report data synchronization utility, configure the Java heap size by setting the IBM_JAVA_OPTIONS operating system environment variable.

- Windows Operating System:
  set IBM_JAVA_OPTIONS=-Xms*minsize* -Xmx*maxsize*
- Linux or UNIX Operating System:
  export IBM_JAVA_OPTIONS='-Xms*minsize* -Xmx*maxsize*'

Where *minsize* and *maxsize* are dependent on Java virtual machine (JVM) type.

Minimum and maximum size for the 64-bit JVM types:

JVM type	Minimum size	Maximum size
64-bit	1024m	4096m

Configuring Report Batch Sizes

Adjust the CSV report batch size to improve report scalability.

About this task

Generating large CSV reports can require adjusting values in the adhocreporting.properties file to avoid Java OutOfMemory errors for large reports.

batch_size	Specifies the number of items requested at a time from the reporting tables. If you do not set a value or comment out the line, all items are fetched. Try setting this value to 10000.

Procedure

1. Edit the adhocreporting.properties file and change the following property: reportBatchSize=*batch_size*
2. Verify that the line is not commented out.
3. In a clustered environment, edit the adhocreporting.properties file on each node.
4. Restart the IBM Security Identity Manager application for these values to take effect.

Configuring Email Notifications

Configuring the system to send email notifications when no email addresses exist can slow down provisioning actions.

About this task

When you configure the system to send an email for an action, the software checks if the user has an email address on the person record. For example, creating an account is an action. If the software finds no email address, it checks the manager of the user. If the manager does not have an email address or the user does not have a manager, the software sends an email to the system administrators.

For large populations, the LDAP search for system administrators can take a while and might slow down provisioning actions. Ensure that user records have email addresses if you use email notifications. If you do not want email notifications, disable them to avoid the lookup.

Procedure

1. Access IBM Security Identity Manager as a system administrator.
2. Expand **Configure System**.
3. Select **Workflow Notification Properties**.
4. In **E-mail Notification Templates**, locate the notification you want to disable.
5. In the **Status** column, hover and select **Disable**.
6. Click **OK**.

Results

The change takes effect immediately.

Using the Recycle Bin

When you enable the recycle bin and then delete objects from IBM Security Identity Manager, the software moves them to the recycle bin.

When you delete objects from IBM Security Identity Manager, they are moved to the recycle bin in the LDAP directory. Deleting objects does not remove them from the underlying directory server. You can delete objects either from the graphical user interface or the application programming interface. Examples of objects
include people, accounts, roles, and provisioning policies.

Using the recycle bin can have negative performance impacts. You might use this feature for a business policy that prohibits reusing a deleted user ID. You can, however, use custom code to enforce the policy and then disable the recycle bin.

The recycle bin is implemented as the following LDAP container:
ou=recycleBin, ou=itim, ou=<tenant>, <suffix>

When you delete objects, the following process occurs:
1. The software moves the LDAP entries under this DN after you delete them.
2. The software sets the erIsDeleted attribute to Y.
3. The Y value tells IBM Security Identity Manager not to display these objects to users or act on them.

Important: The default behavior of the recycle bin changed with IBM Security Identity Manager, version 5.0. Previously, the recycle bin was enabled by default. With version 5.0 and later, it is disabled by default. If you are upgrading from version 4.6, disable the recycle bin unless your environment requires it.

Disabling the Recycle Bin

Disable the recycle bin to avoid performance degradation.

About this task

Disable the recycle bin to avoid performance degradation under the following circumstances:

- You upgraded from a previous version of IBM Security Identity Manager. The default behavior of the recycle bin changed with IBM Security Identity Manager, version 6.0. Previously, the recycle bin was enabled by default. With version 6.0 and later, it is disabled by default.
- Your business policy does not prohibit reusing a deleted user ID. You can, however, use custom code to enforce the policy and then disable the recycle bin.

Procedure

1. Edit enRole.properties.
2. Set the property enrole.recyclebin.enable to false. If the enrole.recyclebin.enable property does not exist, add it to the end of the file with the value of false.
3. Stop all IBM Security Identity Manager nodes.
4. Empty the recycle bin.
5. Restart all IBM Security Identity Manager nodes.

Emptying the recycle bin

Keep the size of the recycle bin as small as possible for optimum performance.

About this task

Use the ldapClean script that is included with IBM Security Identity Manager to remove items from the recycle

bin. This script does not delete workflow records that are in the recycle bin but that are still used by outstanding activities. Use the following parameters for this task:

isim_home	Specifies the home directory for IBM Security Identity Manager, such as /opt/IBM/isim.
script_dir	Specifies the location of the ldapClean script. It is in *isim_home*/bin/*os* where *os* is unix for UNIX or Linux systems or win for Windows systems.

Procedure

1. Edit enRole.properties.
2. Set the property enrole.ldapserver.agelimit to -1.
3. Run the ldapClean script:
   script_dir/ldapClean

What to do next

For an IBM Security Directory Server, run runstats to instruct IBM DB2 pick up the changes.

Related tasks
"Updating Security Directory Server Database Statistics"
DB2 requires information about the number of rows in the tables and what indexes are available so that it can efficiently fulfill queries. If Security Directory Server database is running DB2, version 9, you can set RUNSTATS to run automatically. Version 9 is the default for Security Directory Server, version 6.1. RUNSTATS eliminates the need for running it manually.

Working with Reconciliations

Scheduled Reconciliations

Scheduled reconciliations operate via the **scheduler** – just like any other scheduled item in ISIM. This results in a row being added or updated in the ISIM database for each reconciliation activity when scheduled. Each ISIM node has its own scheduler which periodically (default 30 seconds in SIM 6.0) queries the database to see if something needs to be kicked off. Because each ISIM node has its own scheduler, there is no way to control which node starts the reconciliation – it is essentially random. This means that it is possible for scheduled reconciliations to not be load balanced thus resulting in one node running more reconciliations than other nodes if its scheduler happens to pick up the scheduled reconciliation record before its peers. Security Performance lab testing shows that this does not happen and reconciliations are fairly evenly distributed across all nodes in a cluster.

After the scheduler picks up a scheduled reconciliation, it next creates a *start-the-reconciliation* JMS message on the *ISIM_rs* queue. The *ISIM_rs* queue is a local queue and is not shared with any of the nodes in a cluster. At this point, the reconciliation has not been recorded in the audit trail. When this message is received the workflow engine is notified to start the reconciliation workflow, which creates a pending process in the workflow audit trail. When the workflow engine executes the main reconciliation activity, another JMS message to run the reconciliation is placed on the ISIM_rs queue. The two messages on the ISIM_rs queue are important when examining the timing of reconciliation processes appearing in the audit trail.

If a node is running the maximum number of concurrent reconciliations (*Maximum Concurrent Endpoint value*) and the scheduler picks up another scheduled reconciliation record and puts a start-the-reconciliation message on the ISIM_rs queue, no record of it will show up in the audit log until an ISIM_rs message listener thread is made available (after one of the concurrent reconciliation complete) to process the start-the-reconciliation message and create the run-the-reconciliation message. Thus it is possible for reconciliation to be in 'pending' state: i.e. the scheduler has picked up the scheduled reconciliation record and has processed it, but not be visible in *View Requests.* This does not mean that the reconciliation will not be processed, just that the ISIM_rs message listener hasn't gotten to that JMS message yet. The *ISIM_rs* queue is distinct from the

other queues and *by default has a maximum of 5 threads* allocated to processing messages.

This parameter can be accessed on the WAS Admin console:
Resources -> JMS -> Activation specifications -> ISIMRemoteServicesActivationSpec -> Maximum concurrent endpoints
This means that by default any given ISIM node can only have at most 5 reconciliations running concurrently. Once a *run-the-reconciliation* message is pulled from *ISIM_rs*, ISIM enters a three-phase process. Work in these phases is done by default 8 threads (**enRole.properties: enrole.reconciliation.threadcount**) in addition to the messaging thread which receives the message to run the reconciliation.

Phase 1
ISIM initiates a search for the accounts on the endpoint while concurrently starting an ISIM LDAP search to pulling out the corresponding accounts, if any. If the endpoint search finishes before the ISIM LDAP search does, the endpoint search is blocked from returning results until the ISIM LDAP search finishes.
While reading in the results from ISIM LDAP, if more than 2000 (**enRole.properties: enrole.reconciliation.accountcachesize**) are found, for the remainder of the accounts only the erGlobalID and eruid are stored instead of the entire account object to minimize memory footprint.

Phase 2
This phase begins after ISIM finishes pulling back all the accounts from the ISIM LDAP. First the the accounts are read from the adapter on the message thread and next the accounts are placed onto an in-memory, fixed size queue. As they are pulled off the queue by one of the default 8 worker threads, they are compared against the account found in the in-memory list. If only the erGlobalID and eruid were stored due to the **accountcachesize** threshold, the full account object is looked up prior to comparing it to the record pulled from the adapter. In addition, adoption scripts are executed to find the account's owner if deemed necessary.

- In the event when the adoption script is executed and the account is still considered an orphaned account, it's added/updated in the ISIM LDAP.
- If it's an owned account and policy checking for the reconciliation is disabled, or if policy checking is enabled and it's compliant, then the account is added/updated to the ISIM LDAP.
- If it's an owned non-compliant account, and policy checking is enabled, then the account is added to one of two in-memory lists (*non-compliant and disallowed accounts in-memory lists*). These will be handled in Phase 3.

When all results are pulled from the adapter, any accounts left in the in-memory list from the ISIM LDAP are removed from LDAP. Also, for any newly compliant accounts of deleted accounts where ISIM has a record of compliance issues in the ISIM LDAP, those accounts are added to a third in-memory list for action in Phase 3.
At the end of Phase 2, the 8 worker threads are terminated, and the messaging thread is returned to the pool. The reconciliation workflow continues to the next set of steps in Phase 3.

Phase 3
Policy violations are acted upon. For non-compliant and disallowed accounts, the actions depend on the policy enforcement setting for the service. Also, any stale compliance issues located during Phase 2 are removed. Each of the three lists is implemented as an ISIM workflow loop that takes the necessary actions on each list entry.
In a default ISIM environment it is possible for a node to have over 45 threads (5 * 9) working on reconciliation at one time if 5 or more reconciliations are running concurrently.
The CPU required to run a reconciliation depends on if the account value has changed and if it was changed if it is compliant or not. Unchanged accounts have the least overhead, followed by changed but compliant accounts. Changed and non-compliant accounts have the most overhead.

Tuning Reconciliations

As mentioned earlier, the following parameters can be adjusted to tune reconciliations:
Number of concurrent recons for a single ISIM node can be set on the WAS Admin Console:
Resources -> JMS -> Activation specifications -> ISIMRemoteServicesActivationSpec -> Maximum concurrent endpoints

Recommended range: 5 to 20

The number of concurrent recons a single ISIM node can run concurrently depends on several factors:

- CPU available to run reconciliations
- Complexity of the workflows
- Amount of memory available to hold in-memory structures

If CPU is available on the ISIM node, consider increasing this value up *from the default of 5 to as high as 20*. The CPU required to run the additional concurrent reconciliations should *scale linearly* with the number of reconciliations running concurrently assuming the workflow complexities for the provisioning policies are the same.

Reconciliations are resource-intensive operations.(CPU & I/O) Reconciliations for services with a large account population can affect performance. You can improve reconciliation performance by limiting the number of attributes returned by the adapter and processed by IBM Security Identity Manager. Large reconciliations can exceed the default Max Duration, but you can increase the value. Larger reconciliations can also benefit from using paged searches.

Note: The default value for the *enrole.reconciliation.accountcachesize* parameter in *enRole.properties* file is optimized. Do not change the value of this parameter unless instructed by IBM Support. Increasing this value can decrease reconciliation performance.

Related tasks
"Configuring Paging Parameters"
IBM Security Identity Manager, version 6.0 and later, incorporates LDAP paged searches to alleviate JavaOutOfMemory errors in large environments.

Account Cache Size

Recommended value: 500 to 2000
enRole.properties: enrole.reconciliation.accountcachesize
The account cache is a performance and scalability compromise. For performance, it would be best to store the entire account object in memory. Unfortunately this does not scale for services with millions of accounts. The compromise is to use a value that will enable most services to have all of their accounts in the cache. In general 2000 is a good starting point. Consider the following scenarios:

- 99% of your services have less than 2000 accounts and and the remaining 1% are above it. The default account cache size of 2000 would yield good performance for the 99% while still scaling for the remaining 1%.

- 50% of your services have 100 accounts but the remaining 50% have 50k accounts. If you are using 5 concurrent recons, the default values should be fine as the maximum number of in-memory accounts in a worst-case scenario would be 2000 * 5 = 10,000. If you are using more than 5 threads (for example, 20) you might want to consider decreasing the number of cached accounts to 200 to avoid the worst-case scenario of having all the accounts being reconciled at one time are of the 50k size, thus having 2000 * 20 = 40,000 full accounts in memory at one time.

- 100% of your services have 2500 accounts. If you are using the default of 5 concurrent recons, increasing the cache size to 2500 would improve reconciliation performance but would still limit the total number of in-memory accounts to 2500 * 5 = 12,500.

Limiting Attributes Returned From the Adapter

Limiting attributes returned from the adapter can reduce the amount of work required by the adapter. It can also reduce the amount of data sent to IBM Security Identity Manager. Some adapters (such as the adapter for Microsoft Active Directory) can limit the attributes that are returned to the IBM Security Identity Manager server during reconciliations. Consult the adapter documentation for information specific to that adapter.

Related tasks
"Configuring Attributes Returned During an Active Directory Reconciliation"

Removing calculated attributes returned from an Active Directory reconciliation can improve performance.

Reducing Policy Enforcements

You can reduce the number of policy enforcements by limiting the attributes that are evaluated during the reconciliation. You can also ensure that provisioning policies do not specify mandatory enforcement for attributes that are not reconciled. The reconciliation process updates any changed attributes in the IBM Security Identity Manager directory server. Before this update takes place, the process evaluates the new value against the provisioning policy that governs the account. The validation ensures that the policy permits the change. If not, a policy enforcement is triggered. Any change to the account triggers the policy evaluation for that account regardless if the change invalidates the policy.

Limiting Attributes Evaluated During Reconciliation

To reduce the number of policy evaluations, limit the attributes that are evaluated during reconciliation.

About this task

Some endpoints (such as Microsoft Active Directory) contain attributes that change frequently but are seldom used to enforce policy. An example of this type of attribute is last login time. If these attributes are required, consider:

- Setting up a second reconciliation to reconcile them on a more infrequent schedule.
- Remove them from the more frequently running reconciliations.
- If possible, reconcile only those attributes that are required for policy evaluation.

Use the following parameter:

excluded_attributes	Specifies the list of attributes that are returned from the adapter to exclude from processing in IBM Security Identity Manager. Ideally, you exclude all attributes except those attributes that are required for policy evaluation.

Procedure

1. Access IBM Security Identity Manager as a user with sufficient privileges to edit the service you want to reconcile.
2. Select **Manage Services**.
3. Search for the service you want to reconcile.
4. Select **Set up reconciliation**.
5. Select the reconciliation schedule to modify.
6. Select the **Query** tab.
7. Select all *excluded_attributes*.
8. Click **Remove**.
9. Click **OK**.

Optimizing Entitlement Enforcement

To reduce unnecessary policy enforcement, set entitlement parameter enforcement to mandatory only for attributes that are returned during reconciliation for that service type. Some attributes can be provisioned to a service, but they are not included during a reconcile for that service type. If mandatory enforcement is configured for not-returned attributes, Security Identify Manager updates their value on the endpoint during a reconciliation. This process occurs whether or not the value changed. This process causes unnecessary provisioning actions on the endpoint and increases the load on IBM Security Identity Manager. See the individual adapter documentation for information about which attributes are returned during reconciliation.

Configuring Reconciliation Threads

Each reconciliation process creates additional threads to process the accounts returned from the adapter. Decreasing the number of threads can decrease resource usage while maintaining reconciliation performance.

About this task

The *enrole.reconciliation.threadcount* parameter controls the number of reconciliation threads that are started for a single reconciliation process. The default number is 8. A single thread can process accounts faster than most adapters can return them. Reducing the number of threads decreases the number of idle threads and the JVM resources required to create, track, and delete them. For adapters that return accounts faster than a single thread can process them, decreasing the number of threads can decrease processor utilization. This problem is caused by thread contention while maintaining the reconciliation throughput.

The number of threads running per reconciliation is dependent upon two main factors:

- Speed of the ISIM node
- Throughput from the adapter

If the adapter is returning results back faster than ISIM can process them, then the more threads that are available the faster that single reconciliation will finish. The more common case is for the ISIM node to be faster than the adapter can return the accounts. In this case, adding more threads will not cause the reconciliation to finish faster but instead will result in idle threads consuming nominal resources. The speed that a single reconciliation thread can process accounts depends on if the account has changed. If the account has been modified, the thread must evaluate policy and therefore the complexity of the policy analysis and associated workflows will determine the overall speed.

As a general rule, only the Windows Active Directory and RACF adapters can return accounts faster than a single ISIM node can process them. For these adapters, start with 4 concurrent threads per recon. For all other adapters start with 2 concurrent threads per reconciliation.
Even if the adapter returns results faster than ISIM can process them, *more threads does not equate to faster due to thread synchronization and locking issues*. Testing in Security Performance Labs indicates that once the number of threads goes beyond the number of physical CPUs, the throughput remains mostly flat while CPU usage increases due to context switching. Paradoxically, decreasing the number of threads can improve CPU utilization without impacting reconciliation throughput.

Use the following variable when specifying the number of threads:

reconciliation_threads	Specifies the number of threads a single reconciliation process starts. Default value: 8. Typical value: 2 – 7. • In DSML feed reconciliation/provisioning scenarios where no approval workflow is used, increasing number of threads to (n-1) where n= number of processors available to ISIM, performance is improved. • In DSML feed reconciliation/provisioning scenarios where approval workflow is used, increasing number of threads to (n-1) where n= number of processors available to ISIM, performance degrades and failure rate increases • In endpoint reconciliation scenarios, reconciliation threads set to 5-7 is optimal and produces the best results

Procedure

1. Edit the enRole.properties file and change the following property:
   enrole.reconciliation.threadcount=reconciliation_threads
2. In a clustered environment, edit the enRole.properties file on each node.
3. Restart the IBM Security Identity Manager application for these values to take effect.

Configuring the Maximum Duration of a Reconciliation

Large reconciliations sometimes exceed the default maximum duration specified in the reconciliation schedule. When this limit is reached, the reconciliation halts.

About this task

Increase the limit to allow longer-running reconciliations to complete by using the following variable:

max_duration	Specifies the number for minutes that the reconciliation runs. To calculate this value, do an initial run with a large duration and measure the time. Consider setting the maximum duration to 10% above this time. Default value: 600.

Procedure

1. Access IBM Security Identity Manager as a user with sufficient privileges to edit the information you want to reconcile service.
2. Select **Manage Services**.
3. Search for the service you want to reconcile.
4. Select **Set up reconciliation**.
5. Select the reconciliation schedule you want to modify.
6. Set the **Maximum duration** to *max_duration.*
7. Click **OK**.

Configuring Paged Searches

IBM Security Identity Manager, version 6.0 and later, incorporates LDAP paged searches to alleviate Java Out-Of-Memory errors in large environments.

About this task

Note: Paged searches are useful only for directory servers that support them, such as the IBM Security Directory Server. The Sun Enterprise Directory Server does not support paged searches; enabling paging has no effect.
Paged searches are used in areas that potentially result in large data sets, including:

- Reconciliations
- Provisioning policy creation, modification, deletion, and preview
- Service enforcement changes
- Dynamic role creation, modification, and deletion
- Report data synchronization

Paged searches are disabled by default. They place an additional load onto the LDAP server; some LDAP servers have a limit on the number of concurrent paged searches. When enabling this parameter, configure the underlying LDAP server to accept as least as many paged search requests as the concurrent activities.

Use paged searches for the following large (500,000 or more) data sets:

- Accounts
- People in an organizational tree node
- People in a single role

Tip: A related parameter governs the enabling of server-side sorting. Do not enable server-side sorting. See "Enabling server-side sorting."

Use the following properties to set up paged searches:

paging_enabled	Enables LDAP paging for searches that support it. Valid values: true or false. Default value: false.
paging_size	Specifies the size of the paging request to the LDAP server. If you set this value too high, the LDAP server might ignore the paging request. Do not set this value larger than 128. Default value: 128

Procedure

1. Access the enRole.properties file
2. Change or add the following properties:
   enrole.search.paging.enable=*paging_enabled*
   enrole.search.paging.pagesize=*paging_size*
3. In a clustered environment, edit the enRole.properties file on each node.
4. Restart the IBM Security Identity Manager application for these values to take effect.

Related concepts
"Enabling server-side sorting"
Enabling the server-side sorting property can have a negative impact when viewing large organizational units. Typically, this option remains disabled.

Enabling Server-Side Sorting

Enabling the server-side sorting property can have a negative impact when viewing large organizational units. Typically, this option remains disabled. When retrieving lists of objects from LDAP to display in the interface, IBM Security Identity Manager sorts the results before presenting them to the user. When you enable paged searches, IBM Security Identity Manager also supports the LDAP server that sorts the results. Enabling server-side sorting (through the enrole.search.sss.enabled property in enRole.properties) can have a negative impact when viewing large organizational units. Do not enable this option for most environments.

Related tasks
"Configuring Paging Parameters"
IBM Security Identity Manager, version 6.0 and later, incorporates LDAP paged searches to alleviate JavaOutOfMemory errors in large environments.

Configuring the ACI Cache

Adjusting the time between ACI refreshes and the size of the ACI cache can improve performance in some cases.

About this task

The following properties control the ACI cache. They can improve performance or reduce memory requirements.

refresh_interval	Specifies the number of minutes between ACI cache refreshes. Increasing this value can result in better ACI performance, but ACI changes might take longer to be enforced. Default value: 5.
user_cache_size	Specifies the maximum number of ACI evaluation results to cache per user. Increasing this value can result in better performance for systems with many ACIs. Increasing the value requires more memory from the JVM. Default value: 50.
cache_size	Specifies the maximum size of the ACI cache. Increasing this value can result in better performance. Increasing the value requires more memory from the JVM.

	Default value: 1000.

Procedure

1. Edit the enRole.properties file and change or add the following properties:
   enrole.accesscontrollist.refreshInterval=*refresh_interval*
   enrole.userACICache.maxSize=*user_cache_size*
   enrole.accesscontrollist.maxSize=*cache_size*
2. In a clustered environment, edit the enRole.properties file on each node.
3. Restart the IBM Security Identity Manager application for these values to take effect.

Controlling the Size of the Database

To maintain optimum performance, use the DBPurge utility included with IBM Security Identity Manager to automate removing entries over a certain age from the database.

About this task

The IBM Security Identity Manager database stores data for:
- In-progress system transactions.
- Completed system transactions.
- Auditing information.

The database has no growth boundaries. For best performance, keep as little data as necessary in the live database. Use database backups for older data sets.

The DBPurge utility works with all supported databases. It processes all time-based data, including transaction, audit, and reconciliation records. Use the following variables with this utility:

isim_home	Specifies the home directory for IBM Security Identity Manager, such as /opt/IBM/isim.
os_type	Specifies the operating system time of the IBM Security Identity Manager server. Use either win (for Windows) or unix (for UNIX).
days_to_retain	Specifies the number of days of data to retain records. The utility removes any records in the database older than this value.
purge_trans	Specifies whether to remove transactional data older than *days_to_retain*. Default value: true.
purge_audit	Specifies whether to remove the audit data older than *days_to_retain* during the purge. Default value: true.
purge_reconciliation	Specifies whether to remove reconciliation data older than *days_to_retain* during the purge. Default value: true.

Procedure

Run the following command on one line:

```
isim_home/bin/os_type/DBpurge
-age days_to_retain
-workflow purge_trans
-audit purge_audit
-reconciliation
purge_reconciliation
```

Chapter 8. IBM Security Identity Manager Adapters

Sometimes you must tune IBM Security Identity Manager adapters when doing large provisioning changes or reconciliations. This information supplements, rather than supersedes, the documentation provided for each adapter.

Tuning the Microsoft Active Directory adapter

Changing parameters on the Microsoft Active Directory adapter can improve reconciliation and provisioning performance.

Configuring Attributes Returned During an Active Directory Reconciliation

Removing calculated attributes returned from an Active Directory reconciliation can improve performance.

About this task

During a reconciliation, the Microsoft Active Directory adapter returns attributes to IBM Security Identity Manager that are not directly retrieved from Active Directory. These attributes are calculated from other Windows sources. Querying these external sources can slow down Active Directory reconciliations. You can disable the query if these attributes are not needed.

Working with Windows Terminal Services attributes can also slow down provisioning and reconciliation.

To disable calculated attributes, review and adjust the keys in the adapter registry.

Procedure

- Set ReconHomeDirSecurity and ReconMailboxPermissions to FALSE if the Home Directory Security and Mailbox Permissions attributes are not required. Retrieving this information requires looking up the appropriate access control entry, which can slow down reconciliation. Disabling these attributes improves throughput.
- Set ReconPrimaryGroup to FALSE.
  Disabling this attribute can significantly improve Active Directory reconciliation performance.
- Set WtsEnabled to FALSE.
  This key controls adapter access to Windows Terminal Services attributes. If you set this value to TRUE, the adapter can provision and reconcile the attributes. If you set this key to FALSE, the adapter cannot provision the attributes if requested or return them during reconciliation. Default value: FALSE.
- Set WtsDisableSearch to TRUE.
  This key applies only if the WtsEnabled key is set to TRUE. It controls whether the adapter returns Windows Terminal Services attributes during a reconciliation, which is a search from an adapter perspective. If set to TRUE, a reconciliation does not return the attributes, but it updates the attributes in account provisions. If this key is set to FALSE, the reconciliation returns the attributes. Default value: TRUE.
- Set LyncDisableSearch to TRUE.
  Disabling this attribute can improve performance of slow Lync attributes on reconciliation

Configuring the Number of Threads for the Active Directory Adapter

Increasing the number of threads allocated to provisioning actions can increase the provisioning throughput of the Microsoft Active Directory adapter.

About this task

By default, each provisioning action is configured to use three threads. Doubling the number of threads from

3 to 6 can improve the account provisioning throughput by approximately 100%. Increasing these values too much can result in error message in the adapter log that indicate that directory service is busy. These messages indicate that Active Directory cannot accept the configured number of concurrent threads from the adapter.

You can specify the number of threads that are dedicated for each provisioning action.

Procedure

In the adapter configuration **Advanced Settings** menu, change the appropriate parameter or parameters: ADD, MODIFY, DELETE, or SEARCH.

What to do next

If you receive directory service is busy messages, decrease the number of threads until the error goes away.

Tuning the LDAP Adapter

Reconciling large LDAP directories by using the IBM Security Directory Integrator-based LDAP adapter might require enabling the LDAP paging control on the adapter. It might also require increasing the amount of memory available to the IBM Security Directory Integrator JVM.

About this task

Servers that support the LDAP paging control include IBM Security Directory Server and Microsoft Active Directory. The Sun Enterprise Directory Server does not support the paging control.

Procedure

- You use the LDAP adapter with a server that supports the paging control and enable it. If you do so, the adapter can fetch data from the LDAP server in distinct chunks. See *Directory Integrator-Based LDAP Adapter Installation and Configuration Guide* for more information about enabling the paging control.

Tuning the RACF Adapter

You can adjust RACF adapter reconciliation performance with the PDU_ENTRY_LIMIT environment variable.

About this task

By default this value is not set and reverts to 3000. This value provides good reconciliation performance for most IBM Security Identity Manager environments.

Procedure

- Sometimes the IBM Security Identity Manager server that runs the reconciliation cannot process entries as quickly as they are streamed back from the RACF adapter. The RACF adapter can continue to use up memory as it buffers the requests that are being sent. This process can negatively affect other workloads on the computer due to paging. If this problem occurs, set the size of the PDU_ENTRY_LIMIT environment variable to a lower number, such as 1000 or 500.
- If the IBM Security Identity Manager server can process data faster than the RACF adapter can stream back the accounts, increase the PDU_ENTRY_LIMIT environment variable to decrease total reconciliation time. For example, you might use 4000 or 5000.

Chapter 9. Tuning Security Directory Integrator

Security Directory Integrator is often used in a IBM Security Identity Manager environment both for adapters shipped with the product and for creating custom adapters.

Related information
- IBM Security Directory Integrator 7.2.0.3
- IBM Security Directory Integrator 7.0: Users Guide

Configuring Logging Levels for Security Directory Integrator

The default logging level for Security Directory Integrator is INFO. You can change the logging level to WARN or ERROR to prevent security and administrative issues in production environments.

About this task

The default logging level for IBM Security Directory Integrator is INFO. At the INFO level, the software writes informational messages to the log file. For production systems, this setting has potential security and administrative issues.

Security	At the INFO level, entity attributes and their corresponding values are printed to the log. Password values are not included in the log. IDs and other attributes are included, which might create privacy issues.
Administrative	In busy environments, the log file can grow quickly and might fill up the disk.

You can change the logging level to WARN or ERROR by using the following variables:

itim_solution_dir	Specifies the name of the IBM Security Identity Manager solution directory. It is located underneath the home directory for Security Directory Integrator.
log_level	Specifies the logging level to use, such as WARN or ERROR. Default value: INFO.

Procedure

1. Stop IBM Security Directory Integrator.
2. Edit *itim_solution_dir*/etc/log4j.properties.
3. Change log4j.rootCategory to the level you want.
   log4j.rootCategory=*log_level*
4. Restart IBM Security Directory Integrator for these changes to take effect.

Related information
- Security Identity Manager V6.0.0.10 Directory Integrator RMI Dispatcher Installation and Configuration Guide
- Security Identity Manager V5.1 Directory Integrator RMI Dispatcher Installation and Configuration Guide (PDF)

Using the DSML connector with Security Directory Integrator

You can use the DSML connector to create custom agents for returning information to IBM Security Identity Manager.

The DSML connector can return information as a single unit or in smaller units by using the chunked encoding mechanism. Each method has advantages and disadvantages.

Chunked encoding:
- Applies to all responses to IBM Security Identity Manager, although it is most relevant for reconciliations.

- Prevents the DSML file from being created in-memory in Security Directory Integrator.
- Begins processing IBM Security Identity Manager account reconciliations sooner. The adapter starts streaming accounts back to the server after collecting enough accounts to populate the first chunk.

Without chunked encoding, the DSML file is created in-memory in the IBM Security Directory Integrator. Large reconciliations can cause OutOfMemory errors.

Enable chunked encoding in Security Directory Integrator for all DSML-feed-based adapters.

Tuning the RMI Dispatcher

The IBM Security Identity Manager RMI Dispatcher services requests for RMI-based adapters in Security Directory Integrator.

Configuring Timeouts for Large Reconciliations

The Dispatcher uses timeout values to remove assembly lines that are no longer needed.

About this task

For large reconciliations, the default value of timeouts, such as SearchALUnusedTimeout, can be too small. Small values can result in removing the assembly line before all results are returned. Use the following variables to configure timeouts:

itdi_home	Specifies the home directory for IBM Security Directory Integrator.
searchal_timeout	Specifies the number of seconds before unused assembly lines are cleaned up. Default value: 600.

Procedure

1. Stop the RMI Dispatcher.
2. In *itdi_home*/itim_listener.properties, update the following option: SearchALUnusedTimeout=*searchal_timeout*
3. Restart the RMI Dispatcher for this change to take effect.

Configuring Assembly Line Caching

The RMI Dispatcher caches assembly lines, one per service instance, to improve performance for repeated requests to service instances. You can change the default value that controls the number of lines to prevent reusing stale connections.

About this task

The cached assembly lines improve performance by:
- Retaining ready-to-use copies of the assembly lines in memory.
- Holding open connections to remote endpoints.

The size of the assembly line cache might require downward adjustments to compensate for memory constraints on the system. If the assembly line cache is too large, connections to remote endpoints might time out before they are reused. A timeout causes a failure of provisioning actions that use these stale connections.

The ALCacheSize configuration parameter (Dispatcher version 5.010) controls the number of cached assembly lines. In environments that manage many service instances, change this value from 100 (the default) to 1 to prevent the reuse of stale connections. Use the following variables:

itdi_home	Specifies the home directory for IBM Security Directory Integrator.
num_cached_ALs	Specifies the number of assembly lines to cache. Default value: 100.

Procedure

1. Stop the RMI Dispatcher.
2. In *itdi_home*/itim_listener.properties, update or add the following option:
   ALCacheSize=*num_cached_ALs*
3. Restart the RMI Dispatcher for this change to take effect.

Configuring the Number of Concurrently Running Assembly Lines

Using RMI Dispatcher controls, you can decrease the number of concurrently running assembly lines to prevent an OutOfMemory condition.

About this task

The GlobalRunALCount parameter controls the maximum number of assembly lines that can run concurrently. The default is 100. If the RMI Dispatcher receives a request that exceeds this limit, it places the request in the wait queue. The request stays in the wait queue until the number of running assembly lines is less than the specified limit. The MaxWaitingALcount parameter controls the number of waiting assembly lines. The default is 0 or no limit.

Decreasing the number of concurrently running assembly lines might prevent an OutOfMemory condition in Security Directory Integrator. This condition shows up as a Failed to fork OS thread message in either a javacore or the ibmdi.log file.

Use the following variables to set limits:

itdi_home	Specifies the home directory for IBM Security Directory Integrator.
max_running_ALs	Specifies the maximum number of assembly lines that can run concurrently. Zero indicates no limit. Default value: 100.
max_waiting_ALs	Specifies the maximum number of assembly lines that can wait at one time if the maximum number of running assembly lines is reached. Zero indicates no limit. Default value: 0.

Procedure

1. Stop the RMI Dispatcher.
2. In *itdi_home*/itim_listener.properties, update the following configuration options:
   GlobalRunALCount=*max_running_ALs*
   MaxWaitingALcount=*max_waiting_ALs*
3. Restart the RMI Dispatcher for this change to take effect.

Chapter 10. Database Servers Used with ISIM

IBM Security Identity Manager supports the following databases: DB2 and Oracle Database. Each database requires slightly different tuning. Tuning the database is one of the most important tuning procedures for IBM Security Identity Manager.

Each database server requires at least one processor and 1 GB of RAM. The database can be on a single-processor server by itself or share a multiprocessor server with other applications. The database server requires a minimum of 1 GB of RAM per processor.

Related information
- Database server requirements
- See the information about supported database products and versions.

Tuning IBM DB2

IBM Security Identity Manager, version 6.0 and later, works with DB2 for Linux, UNIX, and Windows starting with Version 9. Version 9 has auto-tuning mechanisms that can reduce administrative and maintenance tasks.

About this task

Tuning DB2 to run with Security Identity Manager includes:
- Adjusting the buffer pools.
- Modifying the number of connections.
- Modifying internal database values.
- Adding table space.
- Adjusting logs.
- Indexing.
- Updating statistics.

Related information
- IBM Security Identity Manager
  See the information about supported versions of IBM DB2.
- Database server requirements
  See the information about supported database products and versions.

Enabling the Self-Tuning Memory Manager

The self-tuning memory manager removes the guesswork in determining the memory values for areas such as buffer pools, the sort heap, and the package heap.

With self-tuning memory enabled, DB2 can move memory between areas based on system need. DB2 versions 9 and 10 databases have the self-tuning memory manager enabled by default.

About this task

The DATABASE_MEMORY parameter determines the total amount of memory available for database-level memory areas. The memory setting depends on the operating system and the installer.

Installer	Operating System	Setting
IBM Security Identity Manager regular installer	AIX, Microsoft Windows, Linux and Sun Solaris	Self tuning with the AUTOMATIC value. The database memory grows or shrinks as needed, based on free operating system memory.

Typically, the actual value determined by DB2 with the AUTOMATIC setting is sufficient. You can to manually raise or lower the value when DB2 shares a system with other components or databases.

The amount of memory available to the global database pool depends on a number of factors, including the following ones:
- The amount of system memory.
- The memory used by other components on the system.
- The number of active database connections.

IBM Security Identity Manager enables its buffer pools for automatic sizing and retains the default value of AUTOMATIC for the sort heap and package cache. If you upgrade from a previous version, the previous values for these settings are retained. You must set the value to AUTOMATIC to enable self-tuning.

Use the following variables when enabling self-tuning:

itim_database_name	Specifies the name of the IBM Security Identity Manager database, such as itimdb.
db_mem_size	Specifies the amount of memory, measured in 4 KB pages, that DB2 uses for self-tuning. You can use the self-tuning value of AUTOMATIC

Procedure

1. Optional: Determine the current value of DATABASE_MEMORY on your system.
   a. Connect to the database.
   b. Run the following commands:
   db2 get db cfg for *itim_database_name* show detail
   The commands display the current value for DATABASE_MEMORY. On Linux or Solaris, it also displays the computed future value.
   c. Multiply the value from step 1b by 4096 for the number of bytes.
2. Update the database configuration to use the self-tuning memory manager:
   db2 update db cfg for *itim_database_name* using SELF_TUNING_MEM ON
3. Set the amount of memory available to the self-tuning memory manager:
   db2 update db cfg using DATABASE_MEMORY *db_mem_size*
4. Restart IBM DB2 for this change to take effect.

What to do next

If you upgraded from version 8 to version 9, enable self-tuning of the sort heap and package heaps. Enter the following commands:

```
db2 update db cfg using sortheap AUTOMATIC
db2 update db cfg using sheapthres_shr AUTOMATIC
db2 update db cfg using pckcachesz AUTOMATIC
```

If self-tuning is active, these settings take effect immediately and do not require an additional restart.

Related tasks
"Configuring database connections for DB2 databases"
DB2 requires enough memory for all possible JDBC connections to run statements without using swap space. If the system does not have sufficient memory, consider decreasing the maximum sizes for the JDBC data sources connection pools.

Related information
Self-tuning memory (DB2 V9.7 information center)
Starting in DB2 Version 9, a memory-tuning feature simplifies the task of memory configuration by

automatically setting values for several memory configuration parameters. When enabled, the memory tuner dynamically distributes available memory resources among the following memory consumers: buffer pools, locking memory, package cache, and sort memory.

Configuring Row-Level Compression

Row-level compression decreases the on-disk footprint of the database. It also improves performance by decreasing I/O wait. It improves buffer pool usage even with the additional processor usage required by compression.

About this task

Attention: Row-level compression is included with the DB2 Storage Optimization, which is a separately purchasable feature. The DB2 license that is included with IBM Security Identity Manager and Security Directory Server prohibits installing any separately purchasable features. To use row-level compression, you must purchase fully licensed versions of both DB2 Enterprise Server Edition and DB2 Storage Optimization.

DB2 version 9 can estimate how well a table can compress. Use the following variables and steps to determine whether a specific table is a good compression candidate and to enable row-level compression.

tablename	Specifies the name of the table for which you want to estimate compression savings.
instancename	Specifies the name of the instance to which the tables belong.

Procedure

1. Evaluate your tables:
   a. Run the DB2 INSPECT command to determine whether the table is a good candidate for row-level compression.
   db2 inspect rowcompestimate table name *tablename*
   schema *instancename* results keep *tablename*.inspect
   This command creates the sqllib/db2dump/*tablename*.inspect binary file.
   b. Format sqllib/db2dump/*tablename*.inspect into a readable format. Enter the following command:
   db2inspf *tablename*.inspect *tablename*.inspect_out
   c. Review the results in the *tablename*.inspect_out file.
   The report shows the percent of pages and space saved by compressing the table. If compression reduces the number of pages by at least 50%, the table is a good candidate for compression.
2. For each table that is a good candidate for compression, enable compression.
   a. Turn off IBM Security Identity Manager.
   b. Connect to the database as an administrator.
   c. Enter the following commands, each on a separate line:

```
db2 alter table instancename.tablename compress yes
db2 reorg table instancename.tablename
db2 reorg indexes all for table instancename.tablename
```

3. Run RUNSTATS on the table.

Configuring Buffer Pools for the IBM Security Identity Manager Database

DB2 buffer pools must be large enough so that most table searches can read directly from memory instead of the disk. You can measure this value by looking at the hit ratio for the buffer pools.

About this task

The IBM Security Identity Manager database has the following buffer pools:

IBMDEFAULTBP	Used as a buffer for table spaces with small extent sizes (4 KB).
ENROLEBP	Used as a buffer for table spaces with large extent sizes (32 KB). Most IBM Security Identity Manager database tables use the table space with a large extent size.

If the buffer pools are not set to AUTOMATIC, use a 1:3 memory ratio between the IBMDEFAULTBP and ENROLEBP buffer pools. Use the following variables in the configuration procedure:

mem_for_itimdb	Specifies the amount of memory in bytes to allocate to the IBM Security Identity Manager database buffer pools. Make this value small enough to be in physical memory so that it is not swapped out to disk. Suggested value: 500000000 (500 MB) or greater.

Procedure

1. Connect to the database as the database administrator.
2. **Optional**: View the current buffer pool sizes. Enter the following command at a command prompt:
   db2 select bpname, npages, pagesize from syscat.bufferpools
   An npages value of -1 indicates that the buffer pools are sized according to the BUFFPAGE parameter. A value of -2 indicates that the buffer pools use automatic sizing.
3. Calculate the optimum size, measured in pages, for the buffer pools:
   ibmdefaultbp_npages = (*mem_for_itimdb* / 4096) * 0.25
   enrolebp_npages = (*mem_for_itimdb* / 32768) * 0.75
4. Alter the buffer pool sizes for the database by running the following commands on separate lines:

```
db2 alter bufferpool ibmdefaultbp size ibmdefaultbp_npages
db2 alter bufferpool enrolebp size enrolebp_npages
```

Related tasks

- "Calculating the buffer pool hit ratio"
  The buffer pool hit ratio gives a good indication of how many data reads come from the buffer pool and how many from the disk. The larger the hit ratio, the less disk I/O used. Calculate the buffer pool hit ratio by enabling buffer pool monitoring and taking a database snapshot.
- "Enabling the self-tuning memory manager"
  The self-tuning memory manager removes the guesswork in determining the memory values for areas such as buffer pools, the sort heap, and the package heap. With self-tuning memory enabled, DB2 can move memory between areas based on system need. DB2, version 9, databases have the self-tuning memory manager enabled by default.

Configuring Database Connections for DB2 Databases

DB2 requires enough memory for all possible JDBC connections to run statements without using swap space. If the system does not have sufficient memory, consider decreasing the maximum sizes for the JDBC data sources connection pools.

About this task

The default value for MAXAPPLS is AUTOMATIC, which is sufficient for most environments. If you require an explicit value set MAXAPPLS to five more than the total maximum number of connections.

When determining memory allocation for DB2, you must consider the number of active connections. Each connection is assigned a DB2 agent that is allocated its own private agent memory (applheapsz). DB2 allocates additional memory when running a statement (stmtheap). To calculate the amount of memory in megabytes required for a single connection, use this formula:
per_connection_memory = (*applheapsz* + *stmtheap*) * 4.096 / 1000

Use the following variables in the configuration procedure:

itim_database_name	Specifies the name of your IBM Security Identity Manager database, such as itimdb.
num_connections	Specifies the maximum number of connections.

Procedure

1. Connect to the database as the database administrator.
2. Run the following command:
   db2 update db cfg for *itim_database_name* using maxappls *num_connections*

Related tasks
"Configuring WebSphere JDBC connections"
IBM Security Identity Manager server uses JDBC connections from WebSphere Application Server to communicate with the database.

Configuring Table Spaces for IBM DB2 Databases

IBM Security Identity Manager uses a database managed space (DMS) table space to store data. This type of table space performs better than system managed space (SMS) table spaces, but you must preallocate disk space for the database to use.

The tables spaces created by the installer have autoresize enabled and grow as needed.

About this task

You might need to define additional table space containers, depending on your specific environment, disk restrictions, and table space layouts.

Adding Additional Table Space Containers

DB2 performs better if a table space has multiple containers on multiple drives. You can add more containers to a table space to increases the amount of space available to tables.

About this task

Add more table spaces with the DB2 alter tablespace command. If possible, add files that are located on another physical drive. The creation and adoption of altered table spaces is not immediate. Examine the output of the alter tablespace command as it runs and rerun the command if the database is busy altering a table space.

Use the following variables in the procedure:

tablespace_name	Specifies the name of the table space for which you want to add containers. IBM Security Identity Manager table space names are *ENROLE_DATA, ENROLE_INDEXES, and TEMP_DATA*.
database_home	Specifies the home directory of your database administrator, such as /home/db2inst1.
instance	Specifies the name of the instance, such as db2inst1. This is a subdirectory in *database_home*.
container	Specifies the name of the file you want to create to hold the table space container, such as enrole_data2.
num_pages	Specifies the number of 32 KB pages you want to add to the table space. To calculate the number of pages from the amount of disk space, divide the size in megabytes by 0.032768. A 512 MB table space is 15625 pages.

	Note: This container might grow if you set the table space to auto-resize.

Procedure

1. As the database administrator, connect to the database.
2. Run the following command for each table space:
   db2 "ALTER TABLESPACE *tablespace_name*
   ADD (FILE '/*database_home*/*instance*/NODE0000/SQL00001/*container*' *num_pages*)"

Enabling Automatic Resizing of Table Spaces

Enable automatic resizing so that containers for a table space can grow automatically if they become full.

About this task

Automatic resizing can decrease the administrative workload for DMS table spaces. However, make sure that the disks on which the containers are located do not become full.

Tip: All table spaces created for IBM Security Identity Manager, version 6, have automatic resizing enabled by default. If you upgrade from versions 4.5.1 or 4.6, you can benefit from enabling automatic resizing on existing table spaces.

Use the DB2 alter tablespace command and the following variables to enable automatic resizing on both new and existing table spaces.

database_name	Specifies the name of the database, such as db2inst1.
tablespace_name	Specifies the name of the table space on which you want to enable autoresize. IBM Security Identity Manager table space names are *ENROLE_DATA, ENROLE_INDEXES, and TEMP_DATA.*

Procedure

1. Connect to the database as the database administrator.
2. **Optional**: View the status of automatic resizing for the table space.
   a. Enter the following command at a command prompt:
   db2 get snapshot for tablespaces on *database_name*
   b. In the stanza that describes the table space look for the line:
   Auto-resize enabled =
3. Turn on automatic resizing by running the following command:
   db2 ALTER TABLESPACE *tablespace_name* AUTORESIZE YES

Setting the Table Space Pre-fetch Size

The default prefetch sizes of the ENROLE_DATA, ENROLE_INDEXES, and TEMP_DATA table spaces are not optimal. Change the prefetch size to AUTOMATIC so DB2 can control this parameter.

About this task

Use the following variable to set the table prefetch size:

tablespace_name	Specifies the name of the table space for which to set the prefetch size. IBM Security Identity Manager table space names are ENROLE_DATA, ENROLE_INDEXES, and TEMP_DATA.

Procedure

1. Connect to the database as the database administrator.

2. Run the following command for each table space:
   db2 ALTER TABLESPACE *tablespace_name* PREFETCHSIZE AUTOMATIC

Updating Table Space Overhead and Transfer Rate

The DB2 overhead and transfer rate parameters used by IBM Security Identity Manager table spaces might not be optimal for upgraded databases.

About this task

The optimizer uses DB2 overhead and transfer rate parameters to calculate query plan costs. IBM Security Identity Manager table spaces use the version 8 default values for these parameters. In DB2, version 9, the default values changed to account for faster I/O subsystems. The following table shows the default values for both versions.

Overhead and transfer rate values:

DB2 version	Overhead rate parameter	Transfer rate parameter
Version 9/10 (new databases)	7.5	0.06

You can determine the actual overhead and transfer rate values for your subsystem by using the formulas in the DB2 version 9 *Performance Guide*. If you cannot determine the value for your hardware, use the version 9 migration values for older hardware or new values for new hardware.

Use the following variables when updating the overhead and transfer rates:

tablespace_name	Specifies the name of the table space for which to set the prefetch size. The IBM Security Identity Manager table space names are *ENROLE_DATA, ENROLE_INDEXES, and TEMP_DATA.*
overhead	Specifies the number of milliseconds required by the container before reading any data into memory. Suggested value: The calculated value for your hardware. Use the new databases values if you are running on new hardware. Use the migrated database values if you are running on old hardware. See the previous table.
transferrate	Specifies the number of milliseconds required to read one page of data into memory. Recommended value: the calculated value for your hardware, if possible. Use the new databases values if you are running on new hardware. Use the migrated database values if you are running on old hardware. See the previous table.

Procedure

1. Connect to the database as the database administrator.
2. **Optional**: View the current overhead and transfer rates. Enter the following command:
   db2 select tbspace, overhead, transferrate from syscat.tablespaces
3. Run the following command for each table space:
   db2 ALTER TABLESPACE *tablespace_name* OVERHEAD *overhead* TRANSFERRATE *transferrate*

Disabling File System Caching

IBM Security Identity Manager table spaces are created with file system caching enabled. If the buffer pools are adequately sized, the file system cache is not necessary and can reduce performance due to double-buffering.

About this task

When the file system cache is disabled on a table space, DB2 uses Direct I/O (DIO) and bypasses the file system cache. DB2 can use Concurrent I/O (CIO) on some platforms with some file systems increasing I/O performance when file system caching is disabled.

If you tuned the buffer pools so that the buffer pool hit ratio is above 95%, disable file system caching. Use the following variables for this procedure:

database_name	Specifies the name of the database, such as db2inst1.
tablespace_name	Specifies the name of the table space for which you want to disable file system caching. IBM Security Identity Manager table space names are ENROLE_DATA, ENROLE_INDEXES, and TEMP_DATA.

Procedure

1. Connect to the database as the database administrator.
2. **Optional**: View the current caching status.
   a. Enter the following command:
   db2 get snapshot for tablespaces on *database_name*
   b. In the stanza describing the desired table space look for the following line:
   File system caching = Yes
3. Run the following command for each table space:
   db2 ALTER TABLESPACE *tablespace_name* NO FILE SYSTEM CACHING
4. Stop IBM Security Identity Manager.
5. Stop and restart DB2.
   The new caching policy becomes effective when DB2 restarts.

Table Compression Candidates for the IBM Security Identity Manager Database

The IBM Security Identity Manager database can use row-level compression, which was introduced in DB2, versions 9 and 10.

Typically, the following tables are good compression candidates:

- activity
- process
- processlog
- audit_event
- audit_mgmt_provisioning
- audit_mgmt_target
- audit_mgmt_delegate

Because building a compression dictionary requires the tables to have data, compression is not enabled by default.

Related tasks
"Configuring row-level compression"
Row-level compression decreases the on-disk footprint of the database. It also improves performance by decreasing I/O wait. It improves buffer pool usage even with the additional processor usage required by compression.

Configuring Transaction Logs for DB2 Databases

DB2 keeps logs during transaction processing. During large transactions, the default log number and sizes might be too small and cause transaction rollbacks.

Increase the size and number of log files to resolve this issue.

About this task

Tip: *For best performance, move transaction logs to a different physical drive than the one where the database is located. Intelligent data storage devices might not require a different physical drive.*

The IBM Security Identity Manager Middleware Configuration utility increases the size of the transaction logs to 10000 and updates the number of secondary logs to 12.

DB2 has the following types of transaction log files:

Primary logs	Allocated when the database is started. They remain allocated until the database is stopped.
Secondary logs	Allocated as needed after the primary logs are full. They are released when they are no longer needed.

Increase the number of secondary logs in preparation for large transactions. The default size of log files is 1000 4 KB pages or 4 MB. Increase this value to 10000 4 KB pages, or 40 MB. The following procedure, which uses these variables, increases the size of primary and secondary log files.

itim_database	Specifies the name of the IBM Security Identity Manager database, such as itim.
logs_secondary	Specifies the number of secondary logs. Suggested value: 12.
logs_size	Specifies the size of the primary and secondary logs in 4 KB pages. Suggested value: 10000.
log_path	Specifies the path where you want to put the transaction logs.

Procedure

1. Connect to the database as the database administrator.
2. Update the database configuration by running the following commands on separate lines.
   db2 update db cfg for *itim_database* using logsecond *logs_secondary*
   db2 update db cfg for *itim_database* using logfilsiz *logs_size*
   db2 update db cfg for *itim_database* using newlogpath *log_path*
3. Stop and restart the database instance. The changes take effect when the database instance restarts.

Configuring Database Application Heaps

Some of the queries that the IBM Security Identity Manager application submits to the DB2 server result in complex SQL statements. If you see transaction rollback errors in the *trace.log* file, increase the values of the heaps in increments of 256 until the errors stop.

About this task

The IBM Security Identity Manager Middleware Configuration utility increases the application heap size to 2048 and the application control heap size to 1024.

IBM Security Identity Manager adjusts the values of the following parameters from their default state. Appropriate tuning requires additional adjustments.

itim_database	Specifies the name of the IBM Security Identity Manager database, such as itim.
applheap_size	Specifies the value of applheapsz in 4 KB pages. Initial value: 2048. (Recommendation: Set to Automatic)
appctl_size	Specifies the value of app_ctl_heap_sz in 4 KB pages. Initial value: 1024.

Procedure

1. Connect to the database as the database administrator.
2. Update the database configuration. Run the following commands separate lines:
   db2 update db cfg for *itim_database* using applheapsz *applheap_size*
   db2 update db cfg for *itim_database* using app_ctl_heap_sz *appctl_size*
3. Stop and restart the database instance. The changes take effect when the database instance restarts.

Configuring Automatic Statistics Collection for the IBM Security Identity Manager Database

Administrators can configure automatic statistics collection so that DB2 automatically updates database statistics. Automatic collection eliminates the necessity of manually running a periodic statistics collection against the database.

About this task

Automatic statistics collection is not enabled by default. For the WebSphere version 7.x and later, JMS implementation to operate properly, exclude the SIBOWNER table from the automatic statistics collection. To improve performance, exclude the 'SCHEDULED_MESSAGE','PROCESSDATA', 'PROCESS', 'and ACTIVITY tables.

Important: Enabling automatic statistics collection without excluding the SIBOWNER table results in database lockups.

For newly created databases, run manual statistics collection (RUNSTATS) after a small data load, even if automatic collection is enabled. RUNSTATS provides statistics for good performance until DB2 initiates the first automatic collection.

Use the following variable in the procedure:

itim_database_name	Specifies the name of your IBM Security Identity Manager database, such as itimdb.

Tip: If the database server does not have the DB2 Control Center, perform this task from a remote system by connecting to the IBM Security Identity Manager database.

Procedure

1. Use DB2 Control Center to update the DB2 maintenance policies:
   a. Start the DB2 Control Center.
   b. Connect to your database with database administrator authority.
   Note: If you do not see your database in Control Center, add it to the catalog before you can continue.
   c. Browse to *itim_database_name*.
   d. Right-click *itim_database_name*.
   e. Click **Configure Automatic Maintenance**.
   f. Click **Next** until you access **Activities**.
   g. Select **Optimize data access (RUNSTATS)**.
   h. Click **Configure Settings**.
   i. Click **Selected tables**.
   j. Select **Use the custom filter**.
   k. In the **Conditions** field, type:
   TABNAME NOT IN ('SIBOWNER','SCHEDULED_MESSAGE','PROCESSDATA', 'PROCESS', 'ACTIVITY')
   l. Click **Refresh Resulting Tables**.
   m. Confirm that **Resulting tables (SCHEMA.NAME)** is populated with all tables except for the ones that you specified in step1k.
   n. Click **OK**.
   o. Click **Finish**.
   p. Confirm the message that no errors were encountered.
   q. Quit Control Center.

2. Enable automatic statistics collection:
   a. As the database administrator, connect to the database at the command prompt.
   b. Run the following command:
   db2 update db cfg for *itim_database_name* using auto_runstats on

How to Update Maintenance Policies *Without* using the Control Centre:

First you will need an AUTO_RUNSTATS policy.

To get the current AUTO_RUNSTATS policy, connect to the database and use:
  db2 "call sysproc.automaint_get_policyfile('AUTO_RUNSTATS', 'AutoRunstats.xml')"
The AutoRunstats.xml file will be written to the $HOME/sqllib/tmp directory.

To set the AUTO_RUNSTATS policy, use:
  db2 "call sysproc.automaint_set_policyfile('AUTO_RUNSTATS', 'AutoRunstats.xml')"
The AutoRunstats.xml file location is relative to the $HOME/sqllib/tmp directory.

The following AutoRunstats.xml can be used for the ***ITIM DB:***

```
<?xml version="1.0" encoding="UTF-8"?>
<DB2AutoRunstatsPolicy
xmlns="http://www.ibm.com/xmlns/prod/db2/autonomic/config" >
 <RunstatsTableScope>
  <FilterCondition>tabname not in ('SIBOWNER','SCHEDULED_MESSAGE','PROCESSDATA', 'PROCESS',
'ACTIVITY')</FilterCondition>
 </RunstatsTableScope>
</DB2AutoRunstatsPolicy>
```

Related information

- Control Center overview
  See the information about using IBM DB2 Control Center.
- RUNSTATS command
  See the information about using the RUNSTATS command.

Updating IBM Security Identity Manager Database Statistics for DB2 Databases

DB2 requires statistics on the number of rows in the tables and available indexes to efficiently execute queries. DB2 version 9 can update the statistics automatically, or you can manually update the statistics.

About this task

If enabling automatic statistics collection is not feasible, you must run the RUNSTATS command manually. Update table and index statistics after large Directory Server Markup Language (DSML) loads, HR feeds, and reconciliations.

Note: DB2 REORGCHK does not update index statistics and is not a replacement for RUNSTATS.

If you experience high processor usage or poor DB2 performance, run RUNSTATS on all of the tables in the database. To update index statistics, run the RUNSTATS command on each table individually. IBM Security Identity Manager performance tuning scripts (perftune_runstats.sh and perftune_runstats.bat) detect the version of DB2 and run the RUNSTATS command against all tables for a specific schema in a database.

If you run the RUNSTATS command in a working environment, make sure that the connected applications can continue to write to the database. Use the allow write access option so users can write to a database while RUNSTATS runs.

Use RUNSTATS on an idle or lightly used database because it requires update locking on the system statistics

table to update the database statistics. The system acquires locks on the tables that are used by the database optimizer to fulfill queries. The locks might cause transaction rollbacks on a database with a heavy load.

In addition to running RUNSTATS on all tables in the database, you must manually update the statistics table for the ACTIVITY, PROCESS, PROCESSDATA, and SCHEDULED_MESSAGE table. Updating the statistic tables ensures a minimum cardinality. Setting a minimum cardinality on these tables helps the DB2 query optimizer and can decrease locking issues in the database.

The following procedure runs RUNSTATS on every table in the ITIMUSER schema.

Procedure

1. Connect to the database as the database administrator.
2. Generate a listing of all tables in the schema by running the following command:
   db2 list tables for all | grep ITIMUSER
3. For each table in the ITIMUSER schema, run the following command on a single line:
   db2 runstats on table ITIMUSER.*table_name*
   on all columns with distribution
   and detailed indexes all allow write access
4. Manually update the database statistics table for the workflow tables by running the following commands on separate lines:

```
db2 update sysstat.tables
   set card = 50000
   where tabname = 'ACTIVITY' and card < 50000
db2 update sysstat.tables
   set card = 50000
   where tabname = 'PROCESS' and card < 50000
db2 update sysstat.tables
   set card = 50000
   where tabname = 'PROCESSDATA' and card < 50000
db2 update sysstat.tables
   set card = 50000
   where tabname = 'SCHEDULED_MESSAGE' and card < 50000
```

Related tasks
"Configuring automatic statistics collection for the IBM Security Identity Manager database"
Administrators can configure automatic statistics collection so that DB2 automatically updates database statistics. Automatic collection eliminates the necessity of manually running a periodic statistics collection against the database.

Related information
ISIM Performance Tuning Scripts
Download performance tuning scripts for IBM Security Identity Manager.

Changing the Maximum Number of Open Files

To work well with other applications that run on the system, DB2 sets a limit on the number of files it keeps open with the maxfilop setting. You can adjust this number to meet the needs of your environment.

About this task

After reaching the specified limit, DB2 closes a currently open file to open the new one. This process can cause a performance loss on systems that do not require a restriction on the number of open files. The IBM Security Identity Manager installation raises the default value. If database snapshots show that database files are closed, increase this value in increments of 64.

The IBM Security Identity Manager Middleware Configuration utility increases the maximum number of open files to 256.

IBM Security Identity Manager adjusts the values of the following parameters from their default state. Further adjustment might be required.

iSDS_database	Specifies the name of the IBM Security Identity Manager database, such as itimdb.
max_files_open	Specifies the maximum number of files DB2 has open at any one time. Initial value: 256.

Procedure

1. Connect to the database as the database administrator.
2. Run the following command on a single line:
   db2 update db cfg for *itim_database* using maxfilop *max_files_open*

Adjusting Lock List and Maximum Locks

The default settings for the DB2 lock list (locklist) and maximum locks (maxlocks) are adequate for most environments.

Increase these values if the local DB2 administrator tells you to do so.

Changing the Lock Timeout

The default lock timeout value (locktimeout) in the IBM Security Identity Manager database is infinity. You can adjust this value if locking problems occur.

If you see locking problems, you can change this value from infinity, represented by -1. The configured value must be greater than or equal to the WebSphere total transaction timeout value, which has a default value of 1200. Setting this value to less than the WebSphere total transaction timeout is unsupported. It can cause transaction rollback errors because not all components recover from a lock timeout.

Improving Disk I/O Performance

Disk I/O performance depends upon the drive types, layout, and configuration. You can change some registry

variables to improve performance on some systems.

About this task

The following DB2 registry variables might improve performance:

Systems	Parameter	Value
All systems	DB2_USE_ALTERNATE_PAGE_CLEANING	ON
Systems with SAN, RAID, or other advanced disk subsystem:	DB2_PARALLEL_IO	*

Running the Related Indexes for Privileged Identity Manager

IBM Security Identity Manager 6.0 provides new indexes for Privileged Identity Manager that help improve your system performance. These new indexes will be need to be created in the ISIM relational database.

These indexes include:

- SA_EVALUATION_BU (L_DN ASC, NAME DESC)
- SA_EVALUATION_BU_HIERARCHY (L_BU_DN ASC, CHILD_DN DESC)
- SA_EVALUATION_BU_HIERARCHY (L_BU_DN ASC, CHILD_DN DESC)
- SA_EVALUATION_CREDENTIAL (L_SERVICE_DN ASC, ACCOUNT_STATUS ASC, L_DN ASC, IS_EXCLUSIVE ASC, IS_SEARCHABLE ASC, USE_GLOBAL_SETTINGS ASC, ACCOUNT_UID ASC, DN ASC)
- SA_EVALUATION_CREDENTIAL_POOL (L_SERVICE_DN ASC, L_BU_DN ASC, NAME ASC, DN ASC)
- SA_EVALUATION_SERVICE_TAG (SERVICE_DN ASC, TAG ASC)
- SA_POLICY (ID ASC)

Procedure

1. Connect to the ISIM relational database as the database administrator.
2. Run the following commands on separate lines:
   - CREATE INDEX "<SCHEMA_NAME>"."SA_EVAL_BUX" ON "<SCHEMA_NAME>"."SA_EVALUATION_BU" ("L_DN" ASC, "NAME" DESC) ALLOW REVERSE SCANS ;
   - CREATE INDEX "<SCHEMA_NAME>"."SA_EVAL_BU_HIERX" ON "<SCHEMA_NAME>"."SA_EVALUATION_BU_HIERARCHY" ("L_BU_DN" ASC, "CHILD_DN" DESC) ALLOW REVERSE SCANS ;
   - CREATE INDEX "<SCHEMA_NAME>"."ISA_EVAL_BU_HIERX2" ON "<SCHEMA_NAME>"."SA_EVALUATION_BU_HIERARCHY" ("L_BU_DN" ASC, "CHILD_DN" DESC) ALLOW REVERSE SCANS ;
   - CREATE INDEX "<SCHEMA_NAME>"."SA_EVAL_CREDX" ON "<SCHEMA_NAME>"."SA_EVALUATION_CREDENTIAL" ("L_SERVICE_DN" ASC, "ACCOUNT_STATUS" ASC, "L_DN" ASC, "IS_EXCLUSIVE" ASC, "IS_SEARCHABLE" ASC, "USE_GLOBAL_SETTINGS" ASC, "ACCOUNT_UID" ASC, "DN" ASC) ALLOW REVERSE SCANS ;
   - CREATE INDEX "<SCHEMA_NAME>"."SA_EVAL_CRED_POOLX" ON "<SCHEMA_NAME>"."SA_EVALUATION_CREDENTIAL_POOL" ("L_SERVICE_DN" ASC, "L_BU_DN" ASC,"NAME" ASC, "DN" ASC) ALLOW REVERSE SCANS ;
   - CREATE INDEX "<SCHEMA_NAME>"."SA_EVAL_SERV_TAGX" ON "<SCHEMA_NAME>"."SA_EVALUATION_SERVICE_TAG" ("SERVICE_DN" ASC, "TAG" ASC) ALLOW REVERSE SCANS ;

- CREATE UNIQUE INDEX "<SCHEMA_NAME>"."SA_POLX" ON "<SCHEMA_NAME>"."SA_POLICY" ("ID" ASC) INCLUDE ("STATUS", "L_BU_DN", "BU_DN", "SCOPE") ALLOW REVERSE SCANS ;

Running the Related Indexes for DBPurge

The IBM Security Identity Manager **DBPurge** operation, by default, uses 4 threads for IBM DB2 database. DBPurge operation can be executed with 1 thread by specifying the "-threads 1" argument to DBPurge command. If DBPurge operation is executed without "-threads 1" option for IBM DB2 database, then the DBPurge operation may fail with errors similar to shown below. DB2 SQL Error: SQLCODE=-1476, SQLSTATE=40506,SQLERRMC=-911 This indicates that either a database time-out or deadlock has occurred. This issue is due to deadlock condition within the multiple threads of DBPurge operation. The tables that have foreign key constraints defined on it and do not have index defined on foreign key column may lead to deadlock or a lock timeout in the database system.

These tables are referenced by the DBPurge utility which does not have any index defined on foreign key column.
1. *ACTIVITY_LOCK* table does not have index for foreign key *ACTIVITY_ID* column.
2. The *PENDING* Table and *PENDING_REQUESTS* tables do not have index explicitly defined on foreign key column but this table has the foreign key and primary key defined on the same column, *PROCESS_ID*. So database creates the index internally for *PROCESS_ID* column.
3. *PROCESSDATA* and *RECONCILIATION_INFO* tables have indexes defined that includes foreign key column, but these tables do not have index that contains only the foreign key columns. The DB2 infocenter documentation specifies that an index containing only the foreign key columns have to be created to resolve deadlock issue.

The following additional indexes should be created in the ISIM database to resolve this issue.

- CREATE INDEX <DATABASE_OWNER>.ACTIVITY_LOCK_AIDX ON <DATABASE_OWNER>.ACTIVITY_LOCK (ACTIVITY_ID ASC) MINPCTUSED 10 ALLOW REVERSE SCANS;

- CREATE INDEX <DATABASE_OWNER>.PROCESSDATA_PIDX ON <DATABASE_OWNER>.PROCESSDATA (PROCESS_ID ASC) MINPCTUSED 10 ALLOW REVERSE SCANS;

- CREATE INDEX <DATABASE_OWNER>.RECONCILIATION_INFO_RIDX ON <DATABASE_OWNER>.RECONCILIATION_INFO (RECONID ASC) MINPCTUSED 10 ALLOW REVERSE SCANS;

- CREATE INDEX ENROLE.ADT_MGMT_OB_RES_OBLIG_X ON ENROLE.AUDIT_MGMT_OBLIGATION_RESOURCE (OBLIGATION_ID ASC) MINPCTUSED 10 ALLOW REVERSE SCANS

This will ensure that DBPurge completes without deadlock on a DB2 database when multiple threads of the DBPurge operation execute simultaneously.

Tuning Oracle

IBM Security Identity Manager supports Oracle databases, starting with version 10g on some operating systems.

About this task

Tuning Oracle to run with IBM Security Identity Manager requires configuring table spaces, indexing, and updating statistics.

Related information
Database server requirements
See the information about supported database products and versions.

Configuring the init.ora Configuration File

The default Oracle configuration uses the small settings in the init.ora file for the database. Using the middle or large values can provide faster performance.

For more information about tuning the Oracle server, see an Oracle DBA or administrator or the Oracle documentation.

Configuring Database Connections for Oracle Databases

The Oracle SESSIONS parameter controls the number of database connections. By default this value is derived from the PROCESSES parameter. You might need to increase the default value.

About this task

Increase the PROCESSES parameter if the default derived value is not high enough. If an explicit value is needed, set PROCESSES to 5 more than the total maximum number of connections.

If the SESSIONS parameter is not derived from the PROCESSES parameter, it might be necessary to update both parameters.

Use the following variable to configure the number of database connections:

num_connections	Specifies the maximum number of connections.

Procedure

1. Connect to the database as the database administrator.
2. Run the following command:
   alter system set processes=*num_connections* scope=spfile;
3. Stop and restart the database instance. Changes take effect when the instance is restarted.

Related tasks
"Configuring WebSphere JDBC connections"
IBM Security Identity Manager server uses JDBC connections from WebSphere Application Server to communicate with the database.

Enabling XA Recovery Operations

You must enable XA recovery operations after installing IBM Security Identity Manager on Oracle.

About this task

Failure to enable XA recovery can result in a WTRN0037 message that indicates that the transaction service encountered an error on an xa_recover operation. Use the following variable to enable XA recovery operations:

itim_db_user	Specifies the user that owns the IBM Security Identity Manager database, such as itimuser.

Procedure

1. Connect to the database as the database administrator.
2. Run the following commands on separate lines:

```
grant select on pending_trans$ to public;
grant select on dba_2pc_pending to public;
grant select on dba_pending_transactions to public;
grant execute on dbms_system to itim_db_user;
```

3. Stop and restart the database instance.

Related information

WTRN messages

See the information about WTRN messages that are issued by WebSphere Application Server.

Configuring Open Cursors

IBM Security Identity Manager uses prepared statements through the WebSphere Application Server JDBC interface. Each prepared statement requires an open cursor in Oracle. If you receive an error message about too many open cursors, you can increase the maximum number of open cursors.

About this task

The message ORA-01000 from the Oracle server indicates that too many open cursors exist. You must either increase the amount of OPEN_CURSORS in Oracle or decrease the maximum number of JDBC connections in WebSphere Application Server.

To increase the amount of OPEN_CURSORS, use the following variable:

num_open_cursors	Specifies the maximum number of open cursors. Default value: 50. Suggested value: 1000.

Procedure

1. Connect to the database as the database administrator.
2. Run the following command:
   alter system set open_cursors=*num_open_cursors* scope=both;
3. Stop and restart the database instance. Changes take effect when the instance is restarted.

Configuring Table Spaces for Oracle Databases

During database configuration, IBM Security Identity Manager creates several small table spaces that can automatically extend as necessary. You can add additional data files.

Additionally, consider spreading the data across multiple physical disks either during the initial database configuration or afterward by adding additional table space containers.

Spreading database data across multiple disks

Spreading the database files across multiple disks decreases the I/O contention in the database and improves Oracle performance. When you create the Oracle database, consider spreading the table space files across multiple disks.

About this task

Use the following variable in the procedure:

disk#	Specifies the disk onto which you want to spread the table spaces.

See *itim_home*/config/rdbms/oracle/enrole_admin_template.sql for information about the default table space definitions.

The *itim_home*/config/rdbms/oracle/create_rollbackSegment.sql script puts the rollback segment table space on ORACLE_HOME (because it specifies only the file name). Consider using a different disk if your environment supports it.

Important: The following SQL statements are for illustrative purposes only. They are environment-specific for both the file system and the size that are allocated to each table space. Consult your Oracle DBA, and tailor the statements to your environment before you apply them.

Use the following procedure to distribute the Oracle database across four hard disk drives: *disk1* through *disk4* with Oracle on *disk1*.

Procedure

1. Create the TEMP table space on *disk2*:

```
create temporary tablespace TEMP
  tempfile '/disk2/oradata/temp01.dbf'
  size 1000m
  reuse
  autoextend on next 32m
  maxsize unlimited;
```

2. Create the ENROLE_DATA table space on *disk3*:

```
create tablespace ENROLE_DATA
  datafile '/disk3/oradata/enrole_data_01.dbf'
  size 64m
  autoextend on next 64m
  maxsize unlimited;
```

3. Create the ENROLE_INDEXES table space on *disk4*:

```
create tablespace ENROLE_INDEXES
  datafile '/disk4/oradata/enrole_indexes_01.dbf'
  size 32m
  autoextend on next 32m
  maxsize unlimited;
```

Related information

- IBM Security Identity Manager Server Installation and Configuration Guide
  See the information about installing and configuring IBM Security Identity Manager servers.

Adding Table Space Data Files

It might be necessary to define additional data files on separate physical devices to provide enough disk space for large deployments.

About this task

The initial data files are created with autoextend on and maxsize unlimited. Use the following variables when adding additional data files to a table space:

tablespace_name	Specifies the name of the IBM Security Identity Manager table space to alter, such as ENROLE_DATA, ENROLE_INDEXES, or ITIML000_DATA.
datafile_name	Specifies the name of the file to use when adding additional data files to a table space or modifying an existing data file. Example value: /data/ou1/app/oracle/oradata/itimdb/enrole_data2.dbf.
initial_size	Specifies the initial size of the data file. Example value: 512m.
maxsize_string	Specifies the string used to set the maximum size of the data file. Use UNLIMITED if you want the data file to grow unbounded. Use maxsize <*number*> to limit it to a specific size. Example: maxsize 2048M.

Procedure

1. Connect to the database as the database administrator.
2. Add data files to a table space. Run the following command:

```
alter tablespace tablespace_name
  add datafile 'datafile_name'
  size initial_size
  autoextend on maxsize_string;
```

3. **Optional**: To alter the maximum size of an existing table space. Run the following command:
   alter database datafile '*datafile_name*' autoextend on *maxsize_string*;

Configuring IBM Security Identity Manager Indexes for Oracle Databases

Adding an index to a heavily used table can greatly increase performance. Without indexes, Oracle must scan every row of the table until it finds the specified data. With an index, it uses a more efficient search method.

About this task

Operational database queries require the following indexes:

- ACCT_CHANGE (POLICY_ANALYSIS_ID ASC, OPERATION_TYPE ASC, REASON ASC)
- ACTIVITY_LOCK (PROCESS_ID ASC)
- ACTIVITY (PROCESS_ID DESC)
- BULK_DATA_INDEX (DATAOBJECTID DESC)
- BULK_DATA_INDEX (STOREID DESC)
- BULK_DATA_STORE (SERVICEID DESC)
- POLICY_ANALYSIS (LAST_ACCESSED ASC, ANALYSIS_ID)
- PROCESS (PARENT_ACTIVITY_ID ASC, ID DESC)
- PROCESS (REQUESTER ASC, PARENT_ID ASC, TENANT ASC)
- PROCESSLOG (ACTIVITY_ID ASC)
- PROCESSLOG (PROCESS_ID ASC)
- RECONCILIATION_INFO (ACCOUNTID ASC, RECONID DESC)
- RESOURCE_PROVIDERS (RESOURCE_STATUS ASC, RESTART_TIME ASC, PROVIDER_ID ASC)

- TASKS_VIEWABLE (VIEW_ID ASC, VIEWABLE ASC, TASK_ID ASC)

DBPurge performance improves with the creation of the following indexes:
- AUDIT_EVENT (WORKFLOW_PROCESS_ID ASC, ID DESC)
- AUDIT_MGMT_DELEGATE (EVENT_ID ASC)
- AUDIT_MGMT_PROVISIONING (EVENT_ID ASC)
- AUDIT_MGMT_TARGET (EVENT_ID ASC)
- LCR_INPROGRESS_TABLE (CHILD_ID ASC)
- RECONCILIATION (COMPLETED ASC)
- RECONCILIATION_INFO (RECONID ASC, OPERATION ASC)
- WORKFLOW_CALLBACK (PROCESS_ID ASC)
- AUDIT_MGMT_OBLIGATION_RESOURCE (OBLIGATION_ID ASC)

The preceding indexes are defined in the itimIndexes/isim_indexes_for_oracle.sql file that is included with IBM Security Identity Manager performance tuning scripts. Use this file to apply the indexes to ensure consistent naming. Indexes that apply to specific conditions, such as viewing completed requests sorted by completion time, are included in the file but commented out. You can edit the file to uncomment these indexes.

Procedure

1. Enter sqlplus at a command prompt.
2. Connect to the database as the system user.
3. In the SQLPlus interface, run the following command:
   @ isim_indexes_for_oracle.sql

What to do next

Update database statistics.

Related tasks
"Updating IBM Security Identity Manager database statistics for Oracle databases"
You must gather and update database statistics at regular intervals. Intervals can be one week to one month on a production IBM Security Identity Manager system or after processing a large amount of data.

Related information
ISIM Performance Tuning Scripts
Download performance tuning scripts for IBM Security Identity Manager.

Updating IBM Security Identity Manager Database Statistics for Oracle Databases

You must gather and update database statistics at regular intervals. Intervals can be one week to one month on a production IBM Security Identity Manager system or after processing a large amount of data.

Before you begin

Install the DBMS_STAT package.

About this task

Oracle uses statistics to make query decisions on locating information that impact how fast Oracle can return requests. Use the following variable with the Oracle DBMS_STAT commands:

database_instance	Specifies the name of the database instance, such as enrole.

Tip: Generate statistics during off-peak times. Generating statistics can take from several minutes to several hours for a large database.

Procedure

1. Create a file named Oracle_dbms.stat_cmds.txt.
2. Edit the file and insert the following text:
   exec dbms_stats.gather_schema_stats(ownname => '*database_instance*',
   cascade => true);
3. Enter sqlplus at a command prompt.
4. Connect to the database as the system user.
5. In the SQLPlus interface, run the following command:
   @ Oracle_dbms.stat_cmds.txt

Chapter 11. Directory Servers

Tuning IBM Security Directory Server

When tuning IBM Security Directory Server, it is important to understand the interaction between the IBM Security Directory Server process and DB2.

In a well-tuned environment, the Security Directory Server process and the DB2 processes use approximately the same amount of processor cycles. DB2 can max out the processor usage while trying to fulfill queries in a poor manner.

Both Security Directory Server and DB2 have caches that speed up data retrieval. Optimizing available memory is the key to tuning IBM Security Directory Server. When a read request comes in to Security Directory Server, it checks the filter cache to see whether it saw that search filter previously. If it has, it pulls the results from the cache, otherwise the query goes to DB2. After evaluating the search filter, Security Directory Server pulls the entries that match the search filter from the entry cache.

If the values are not in the entry cache, it queries DB2. For each request, DB2 checks to see whether the data is in a buffer pool. If not, it reads the value from the disk. Ideally, all requests to the directory server register a Security Directory Server cache hit or a DB2 buffer pool hit for the quickest response. Queries that require disk access can be slow.

Related information

- IBM Security Directory Server V6.4 Performance tuning and capacity planning
  See product information about tuning the IBM Security Directory Server V6.4
- IBM Security Directory Server V6.3.1.5 Performance Tuning and Capacity Planning Guide
  See product information about tuning the IBM Security Directory Server V6.3.1.5

Tuning ISDS Database Connections

The default number of connections created between Security® Directory Server and DB2® is 15, which suffices for most environment in which directory server is used. However based on the requirement, user can increase the database connections as needed. Depending on the server load and the nature of connections, performance might improve will the increase in the number of back-end connections. The best way to make this decision is by determining the result of the monitor search and look for available_workers threads in the output. Database connection is basically the number of worker threads, so to increase the workers threads or to increase the back-end connections, user must set the attribute ibm-slapdDbConnections.

ibm-slapdDbConnections	Specify the number of DB2 connections the server will dedicate to the DB2 backend. The value must be between 5 and 50 (inclusive). Default value: 15 Suggested value: 30

Procedure

1. Stop IBM Security Directory Server.
2. In ibmslapd.conf, update the following configuration options:
   ibm-slapdDbConnections: recommended value
3. Restart IBM Security Directory Server for these changes to take effect.

Configuring Cache Sizes

You can configure the Security Directory Server caches to increase performance and meet the needs of your environment.

About this task

IBM Security Directory Server has the following types of caches:

Access control list (ACL) cache	Because IBM Security Identity Manager server binds as an authoritative user, this cache is used only for internal processes. The allocated size can be small, and the memory can be used, which increases Security Directory Server performance.
Filter cache	This cache helps programs that issue more read requests than write or update requests, because the entire filter cache is invalidated/refreshed at every write. *IBM Security Identity Manager frequently updates the directory server, so it is not beneficial to allocate a large filter cache*. Enable the filter cache, but keep it small.
Entry cache	You can control how many entries the entry cache can store. You cannot restrict the size of the cache. The size of each entry is based on the number and the size of attributes that a specific LDAP entry has. Typically, many entries are users and their accounts, which have a fairly constant size. When setting the value for the entry cache, calculate the size of the average entry. Divide the size of the average memory into the amount of memory used by the Security Directory Server process. Users with few attributes can generate entry sizes that are approximately 4 KB. Users with more attributes can generate entry sizes around 9 KB. See the *IBM Security Directory Server Performance Tuning Guide* for the procedure to determine the average entry size. Do not set the entry cache size larger than available physical memory. If the Security Directory Server process size exceeds the amount of available memory, swapping the page causes significant performance degradation. When increasing the cache size, make sure the amount of memory required does not exceed the maximum amount a process can allocate. **Example**: For an average entry cache size of 9 KB, setting the entry cache size to 75,000 would require 675 MB (75,000 * 9 KB = 675,000 KB = 675 MB) of physical RAM. The requirement does not include the 128 MB for the server process.
Attribute cache	Performance metrics suggest that the attribute cache available in Security Directory Server, version 6.0 and later, does not provide a significant performance boost. You can allocate the memory elsewhere.

Use the following variables for configuring the cache sizes:

acl_cache	Specifies whether the ACL cache is used. Suggested value: TRUE (enabled).
acl_cache_size	Specifies the size of the ACL cache. Suggested value: 100.
filter_cache_size	Specifies the size of the filter cache. Suggested value: 100.
entry_cache_size	The size of the entry cache. Suggested value: *max_users* * (*average_accounts* + 1) For example, if you have 25,000 users with two accounts each: 25,000 * (2+1) = 75,000. This value is bounded by the amount of memory allocated to the Security Directory Server process minus the size of the process itself (about 128 MB).

Procedure

1. Stop IBM Security Directory Server.
2. Update the following configuration options in ibmslapd.conf:

```
ibm-slapdACLCache: acl_cache
ibm-slapdACLCacheSize: acl_cache_size
ibm-slapdFilterCacheSize: filter_cache_size
ibm-slapdEntryCacheSize: entry_cache_size
```

3. Restart IBM Security Directory Server for these changes to take effect.

What to do next

Caches are only one part of tuning the IBM Security Directory Server. Tuning the underlying IBM DB2 database has equal or greater performance impact than tuning the caches. Do not skip the DB2 tuning.

Related information

- IBM Security Directory Server V6.4 Performance tuning and capacity planning
  See product information about tuning the IBM Security Directory Server V6.4
- IBM Security Directory Server V6.3.1.5 Performance Tuning and Capacity Planning Guide
  See product information about tuning the IBM Security Directory Server V6.3.1.5

Configuring Paging Parameters

Security Directory Server supports returning search results in pages so that the client has more control over receiving the data. If you enable paged searches in IBM Security Identity Manager, you must set the paged search parameters correctly for optimum performance.

About this task

Because paged searches require more resources on the directory server, you can specify whether non-administrative users can perform paged searched. You can also specify the number of concurrent paged searches.

Use the following variables when configuring paging parameters:

allow_non_admin	Specifies whether non-administrative users can request paged searches. Suggested value: TRUE, if IBM Security Identity Manager is binding as a non-administrative user. If not, specify FALSE. Default value: TRUE
concurrent_paged_searches	Specifies the maximum number of concurrent paged searches. Set this value to 1 more than the maximum expected number of paged searches. Default value: 3 This value will need to be increased in production environments running concurrent reconciliations. Testing with 4 concurrent reconciliations, *ibm-slapdPagedResLmt* was increased to **5**. When increasing the number of concurrent paged searches, monitor resource utilization on the directory server to ensure that overall performance does not degrade. Also ensure that the number of back-end database connections is larger than the total number of paged searches.

Procedure

4. Stop IBM Security Directory Server.
5. In ibmslapd.conf, update the following configuration options:
   ibm-slapdPagedResAllowNonAdmin: *allow_non_admin*
   ibm-slapdPagedResLmt: *concurrent_paged_searches*
6. Restart IBM Security Directory Server for these changes to take effect.

DB2 Selectivity

IBM Directory Server Version 6.4 – Selectivity
Using a SELECTIVITY clause to influence the optimizer

subtree vs single level

SDS 6.3.1 code was modified so that it is no longer necessary to set either cardinality or selectivity. SDS utilizes *literal values* for the searches on EIDs in both *LDAP_DESC* and *LDAP_ENTRY* tables so that the optimizer can differentiate between large tree vs small tree searches and make the appropriate query plan. The difference between the subtree and single level search may be due to the different cardinalities set for LDAP_ENTRY vs LDAP_DESC. There are two parts of a subtree search that are of interest: *the subtree and the filter*. The subtree part gets transformed into a select against LDAP_DESC on AEID, while the filter part gets transformed into a combination of selects against the tables and columns corresponding to the attributes in the filter, with everything joined together by EID. The desired outcome is for the filter to be resolved using the indexes on the attributes, (to be called "attribute indexes"). In the most common scenario, a search is looking under a large subtree for a small set of entries matching the filter (for example searching for a user by name). In that scenario, the desired outcome is for DB2 to first use the attribute indexes to find the small set of candidate entries and then join against LDAP_DESC to find the ones that are part of the subtree. The other common scenario is a search that looks under a very small subtree for an entry that matches a very common filter (for example looking under a user's subtree for a particular objectclass with application information). In that scenario, the desired outcome is for DB2 to use LDAP_DESC to find the handful of entries that are children of the subtree and then use the attribute indexes to find the entry that matches. Relatively small directories do not find this to be an issue because none of the indexes are very large, however, on the case of larger directories (millions of entries) the order of execution of the query plan can make a significant difference in both response time and CPU utilization. If DB2 uses the wrong indexes first in its query plan, it can end up spending a long time retrieving a large set of candidate EIDs, most of which will be thrown away on subsequent joins. If DB2 uses the right index first, it can very quickly come up with a small set of candidate EIDs at the start.

It is for this reason that it is critical for DB2 to utilize the most efficient query plan for executing searches. Security Performance has attempted several different strategies to help the optimizer make the best decisions. Due to the utilization of parameter markers (in order to allow query plans to be cached and reused for similar searches), the DB2 optimizer did not know whether a subtree is large (that is, there are many rows with that value) or small, nor whether an attribute filter is using common or rare values. Since it couldn't compare the actual values to collected distribution statistics, it could only use the relative sizes of the tables (the cardinality) to make a guess on what the best query plan would be. Therefore, the first approach to influencing the optimizer in the right direction was setting the cardinality of LDAP_DESC to a very large number, thus making sure that the index on AEID and DEID was used after the attribute indexes. This essentially always optimized for the large subtree search scenario. In order to handle both large subtrees and small subtrees well at the same time, we needed to tell DB2 on each select whether it should be using the attribute indexes or the LDAP_DESC index first. We did this by using selectivity in the queries on small subtrees to tell DB2 that the comparison on the AEID column is likely to produce a small number of candidate rows (see: Using a SELECTIVITY clause to influence the optimizer) . It was decided whether a subtree was small or large at startup, by some complicated logic that looks for up to ten values that most frequently show up as AEIDs.

In the IBM Security Directory Server (SDS), subtree searches can be very expensive when the query is associated with search bases that are high in a directory tree. In order to improve these searches, SELECTIVITY in Structured Query Language (SQL) could be utilized. Including SELECTIVITY in SQL enables the DB2 optimizer in the formation of data access sequence to more efficiently resolve the search requests. The data access sequence identifies which tables to access first during searches. The optimizer uses data from DB2 statistics to accurately identify entries that are high in the tree (i.e. having numerous child entries). If a subtree search is done by using one of these entries as the search base, the SELECTIVITY clause is added to the SQL query. When the SELECTIVITY clause is added, DB2 uses the search filter to narrow down the search

results. DB2 narrows down the search results before it reads from the table that identifies the entries that are descendants of a base in a search. There are settings both in DB2 and in SDS that must be configured in order for selectivity to be utilized

DB2

To use SELECTIVITY, DB2_SELECTIVITY must be set to YES in the DB2 registry for the database instance. It is recommended that DB2_SELECTIVITY is set when creating a database instance.

As the instance owner:
db2set DB2_SELECTIVITY=YES

SDS

IBM Security Directory Server can influence the DB2 optimizer so that the most efficient access plan is chosen to fetch data from the tables associated with LDAP.

These are the two environment variables that should be set in order to influence the DB2 optimizer:

1. LDAP_MAXCARD
Valid values: YES, ONCE, NO
The *LDAP_MAXCARD* environment variable sets the cardinality of the LDAP_DESC table. The LDAP_DESC table is used to evaluate the subtree scope on a ldapsearch command. This table contains a list of parent and child LDAP entry relationships using two columns:

- A Descendant EID or DEID column.
- An Ancestor EID or AEID column.

For each LDAP entry, there is a full list of parents for that LDAP entry in the LDAP_DESC table. For this reason, in large ISIM deployments it is imperative to manage the cardinality logically. *If LDAP_MAXCARD is set to 'YES'. a very large cardinality of **9E18** is assigned to the LDAP_DESC table*. The purpose of inflating this cardinality value is to influence the data access sequence of the DB2 optimizer. In terms of priority, DB2 will first resolve all attribute filters before it considers the LDAP_DESC table for query evaluation. In addition, if LDAP_MAXCARD is set to to YES, this cardinality adjustment prevents expensive scans of large subtree data. The cardinality statistic is set at the server startup and periodically thereafter. If the variable is set to ONCE, the cardinality is set during the server startup and not later when the server is running. If the variable is set to NO or not set, the cardinality statistic is not set during the server startup. LDAP_MAXCARD instigates the ldap server to call a function called *rdbm_tune_stats*. This particular function enables the cardinality of the LDAP_DESC table to be increased.

2. IBMSLAPD_USE_SELECTIVITY
Valid values: YES, NO
If the *IBMSLAPD_USE_SELECTIVITY* variable is not set to any value or is set to NO, selectivity is not used to influence DB2 access sequence. If *IBMSLAPD_USE_SELECTIVITY* is set to YES and *LDAP_MAXCARD* is not set to YES, selectivity is used to influence the data access sequence of DB2 during the subtree search on a large subtree. In addition, when IBMSLAPD_USE_SELECTIVITY is set to YES, the LDAP server will not override the statistics on the LDAP_DESC table, but will apply the SELECTIVITY clause to subtree searches based on whether the subtree is one of the top subtrees in size. By letting runstats set the statistics for LDAP_DESC to its true size, this allows searches on small subtrees to use the index on LDAP_DESC first to get the members of that small subtree. Note: If *LDAP_MAXCARD* and *IBMSLAPD_USE_SELECTIVITY* are set to YES, the directory server generates a message and does **not** use selectivity.

idsldapmodify -h host -p port -D adminDN -w adminPW
dn: cn=Front End, cn=configuration
changetype: modify
add: ibm-slapdSetEnv
ibm-slapdSetEnv: IBMSLAPD_USE_SELECTIVITY=YES

idsldapmodify -h host -p port -D adminDN -w adminPW
dn: cn=Front End, cn=configuration

changetype: modify
add: ibm-slapdSetEnv
ibm-slapdSetEnv: LDAP_MAXCARD=YES

Security Performance Recommendations

The vast majority of ISIM customers will not need to set either SELECTIVITY or LDAP_MAXCARD . In some instances, LDAP_MAXCARD can be the most efficient method to ensure optimized query completion. However, in certain rare situations, enabling SELECTIVITY will result in better search times than LDAP_MAXCARD. Currently, some ISIM deployments enjoy good performance with SELECTVITY enabled. Regardless of the strategy, the ISIM tuning guide script sets cardinality for several tables in addition to LDAP_DESC, and this technique should still be done even when enabling SELECTIVITY.

For SDS versions < 6.3.1, Security performance recommends inflating cardinality on the LDAP_DESC table because of the utilization of *parameterized markers* thus the DB2 optimizer doesn't have enough information to make the most accurate of decisions for subtree searches.

Selectivity Notes

- Selectivity affects how the optimizer uses the LDAP_DESC table.

- The ERPARENT and LDAP_ENTRY statistics changes are independent of that

(ERPARENT is ISIM's equivalent of LDAP_DESC in the way it is used to scope a search to a particular ISIM org)

- The vast majority of ISIM customers will not need to set either SELECTIVITY or LDAP_MAXCARD

Configuring Bufferpools for the IBM Security Directory Server Database

DB2 buffer pools are the secondary buffer for Security Directory Server. These buffer pools must be large enough so that most table searches can be read directly from memory instead of using the disk.

About this task

Security Directory Server database has the following buffer pools:

IBMDEFAULTBP	Used as a buffer for table spaces with small extent sizes (4 KB). Most of the tables in the database have table spaces with a small extent size and use IBMDEFAULTBP.
LDAPBP	Used as a buffer for table spaces with large extent sizes (32 KB).

DB2, version 9, is the default for Security Directory Server, version 6.3. If you use version 9, set the buffer pools to AUTOMATIC so that the self-tuning memory manager can adjust the memory settings for your workload.

If the buffer pools are not set to AUTOMATIC, use a 3:1 memory ratio between IBMDEFAULTBP and LDAPBP. Allocate enough memory to the DB2 buffer pools so the buffer pool hit ratio is greater than 95%. Allocate the remaining memory to the Security Directory Server process and the caches.

Use the following variables to configure buffer pools:

ldap_database	Specifies the name of the IBM Security Directory Server database, such as ldapdb2.
mem_for_ldapdb2_bps	Specifies the amount of memory in bytes to allocate to the ldapdb2 buffer pools. Make this value small enough so that it is in physical memory and is not swapped out to disk. Suggested value: 500000000 (500 MB) or greater.

Procedure

1. Connect to the database as the database administrator.
2. **Optional**: View the current buffer pool sizes by entering the following command at a command prompt:
   db2 select bpname, npages, pagesize from syscat.bufferpools
   An npages value of -1 indicates that the buffer pools are sized according to the BUFFPAGE database configuration parameter. A value of -2 indicates that the buffer pools use automatic sizing.
3. Calculate the size for the buffer pools, measured in pages:
   ibmdefaultbp_npages = (*mem_for_ldapdb2_bps* / 4096) * 0.75
   ldapbp_npages = (*mem_for_ldapdb2_bps* / 32768) * 0.25
4. Alter the buffer pool sizes for the database by running the following commands on separate lines:
   db2 alter bufferpool ibmdefaultbp size *ibmdefaultbp_npages*
   db2 alter bufferpool ldapbp size *ldapbp_npages*

Related tasks

- "Configuring Database Connections"

Disabling File System Caching

Both Security Directory Server table spaces (LDAPSPACE and USERSPACE1) are created with file system caching enabled. If the buffer pools are adequately sized, the file system cache is unnecessary and can reduce performance due to double-buffering.

About this task

When the file system cache is disabled on a table space, DB2 uses Direct I/O (DIO) and bypasses the file system cache. DB2 can use Concurrent I/O (CIO) on some platforms. Some file systems increase I/O performance when file system caching is disabled.

If you tuned the buffer pools so that their hit ratio is over 95%, disable file system caching.

Security Directory Server, version 6.3, disables file system caching by default.

Use the following variable when disabling the file caching system:

tablespace_name	Specifies the name of the table space for which you want to disable file system caching. IBM Security Identity Manager table space names are LDAPSPACE and USERSPACE1.

Procedure

1. Connect to the database as the database administrator.
2. Run the following command for each table space:
   db2 ALTER TABLESPACE *tablespace_name* NO FILE SYSTEM CACHING
3. Stop the IBM Security Directory Server.
4. Stop and restart IBM Security Directory Server database. The new caching policy becomes effective after you disconnect all database connections.
5. Start the IBM Security Directory Server.

Related information

Creating table spaces without file system caching
See the list of I/O methods used when file system caching is disabled for IBM DB2 table spaces.

Table Compression Candidates for the IBM Security Directory Server Database

IBM Security Directory Server database can use row-level compression with DB2, versions 9 and 10.

Because building a compression dictionary requires the tables to have data, compression is not enabled by default.

The following tables are good compression candidates for IBM Security Identity Manager:
- ldap_entry
- objectclass
- erparent
- erservice
- erroles
- owner
- manager
- secretary

Because each LDAP attribute is stored in a separate table, there are many possible candidates. A good candidate depends on the composition of person and account objects in your environment. Other attributes to consider:
- street
- l (location)
- st (state)
- title
- description
- erlostpasswordanswer
- erchangepswdrequired
- mobile, telephonenumber, and facsimileTelephoneNumber (if your users have a common set of area codes or prefixes)

Security Directory Server, version 6.3.x, includes the **idsdbmaint** command. This command automatically evaluates tables for compression and compresses good candidates.

Related tasks
"Configuring row-level compression"
Row-level compression decreases the on-disk footprint of the database. It also improves performance by decreasing I/O wait. It improves buffer pool usage even with the additional processor usage required by compression.

Configuring Transaction Logs for the Security Directory Server Database

DB2 keeps logs during transaction processing. During large transactions, the default log number and sizes might be too small and cause transaction rollbacks.

Increase the size and number of log files available to DB2.

About this task

DB2 has the following types of log files:

Primary logs	Are allocated when the database is started and remain allocated until the database is stopped.
Secondary logs	Are allocated as needed when the primary logs are full and released when they are no longer needed.

For best performance, move the transaction logs to a different physical drive than the database. Intelligent data storage devices might not require a different physical drive.

Increase the number of secondary logs to prepare for large transactions. The default size of the log files is

1000 4 KB pages (4 MB). Increase the size to 10000 4 KB pages (40 MB). Increasing the default changes the size of both primary and secondary log files.

Use the following variables when configuring logs:

ldap_database	Specifies the name of the Security Directory Server database, such as ldapdb2.
logs_secondary	Specifies the number of secondary logs. Suggested value: 24.
logs_size	Specifies the size of the primary and secondary logs in 4 KB pages. Suggested value: 10000.
log_path	Specifies the path to where the transaction logs are located.

Procedure

1. Connect to the database as the database administrator.
2. Run the following commands on separate lines:
   db2 update db cfg for *ldap_database* using logsecond *logs_secondary*
   db2 update db cfg for *ldap_database* using logfilsiz *logs_size*
   db2 update db cfg for *ldap_database* using newlogpath *log_path*
3. Stop and restart the database instance. The changes take effect when the database instance restarts.

Configuring Database Statement Heaps

You can increase the size of the DB2 statement heap (stmtheap) to eliminate errors caused by long queries.

About this task

IBM Security Identity Manager can submit long LDAP queries to the Security Directory Server. Some queries might not fit in the DB2 statement heap (stmtheap).

Security Directory Server returns an error to IBM Security Identity Manager.

The statement heap is allocated per agent (connection). Increasing this value can dramatically increase the memory used by the DB2 server.

Use the following variables to configure database statement heaps:

ldap_database	Specifies the name of the Security Directory Server database, such as ldapdb2.
stmtheap_size	Specifies the value of stmtheap in 4 KB pages. Default value: 2048. Suggested value: 4096.

Procedure

1. Connect to the database as the database administrator.
2. Run the following command:
   db2 update db cfg for *ldap_database* using stmtheap *stmtheap_size*
3. Stop IBM Security Directory Server
4. Stop and restart the IBM Security Directory Server database.
5. Start IBM Security Directory Server.

Configuring System Limits

System limits (ulimits) might prevent the Security Directory Server process from accessing enough real or virtual memory. To avoid memory dumps or stopping without indication, increase the ulimits.

About this task

Use the following variables to configure system limits:

process_data_size	Specifies the maximum data segment size for the process. Minimum value: 256000 (256 MB). Suggested value: unlimited.
virtual_mem_size	Specifies the maximum virtual memory size for the process. Minimum value: 256000 (256 MB). Suggested value: unlimited.

Procedure

1. Connect to the database as the user who starts the Security Directory Server process.
2. (**AIX only**) Update the /etc/security/limits file:
   a. In /etc/security/limits, locate the stanza for the user who starts the Security Directory Server process.
   b. If the stanza does not exist, add it.
   c. Change the data limit to *process_data_size* or to -1 for unlimited. If the limit setting is not there, add it.
   d. Change the rss limit to *virtual_mem_size*, or to -1 for unlimited. If the limit setting is not there, add it.
   e. Log out of the current session and log back in for the changes to take effect.
3. (**Solaris only**): Run the following commands before starting ibmslapd or place them into the shell startup files for the user:
   ulimit -d *process_data_size*
   ulimit -v *virtual_mem_size*

Related information

Adjusting user process resource limits for ISDS

See more information about setting the limit correctly for IBM Security Directory Server.

Configuring Attribute Indexes for Security Directory Server

Indexing the attributes on which applications search increases Security Directory Server performance. Security Directory Server indexes automatically translate into DB2 indexes when you update the Security Directory Server schema for those attributes.

About this task

Index those attributes on which you intend to search, if you extend the LDAP schema in Security Directory Server to include additional attributes. Any filter in IBM Security Identity Manager (such as with dynamic roles) is translated into a search string for the Security Directory Server.

Security Directory Server reports messages in the ibmslapd.log file for attributes on which a search was run and the attributes were not indexed. Consider indexing attributes that have more than 100 searches.

IBM Security Identity Manager provides the perfanalyze_indexes.pl script in its performance scripts. You can use it to find attributes that were searched, but not indexed. The script can generate an LDIF that you can use to index the attributes.

See the documentation that comes with the performance scripts for detailed information about using perfanalyze_indexes.pl.

Use the following variables when configuring attribute indexes:

root_dn	Specifies the root DN of the IBM Security Directory Server server.
root_password	Specifies the password for the root DN.

Procedure

1. Use the perfanalyze_indexes.pl script to create an LDIF to index the attributes. For example:
   perfanalyze_indexes.pl -i audit.log -d
   /home/idsinst/idsslapd-idsinst/etc -l indexes.ldif
2. Edit the resulting indexes.ldif file to remove any stanzas for attributes you do not want to index.
   Tip: Indexes add additional overhead for update events. Not every attribute needs an index.
3. Run the following command to import the LDIF into IBM Security Directory Server:
   ldapmodify -D *root*_dn -w *root_password* -f indexes.ldif
4. After updating the LDAP schema, run RUNSTATS on the database to update the statistics for the newly created indexes.

Related information
ISIM Performance Tuning Scripts
Download performance tuning scripts for IBM Security Identity Manager.

Configuring DB2 Indexes

Adding indexes directly to specific tables in the underlying DB2 database can improve performance for some Security Directory Server queries.

About this task

The following indexes improve search performance for some queries and are included in later versions of IBM Security Directory Server:

- LDAP_DESC (AEID ASC, DEID ASC)
- OBJECTCLASS (EID ASC, OBJECTCLASS ASC)
- OBJECTCLASS (OBJECTCLASS ASC, EID ASC)

Use the following variable when configuring DB2 indexes:

schema_name	Specifies the schema for IBM Security Directory Server tables.

Procedure

1. Connect to the database as the database administrator.
2. Run the following commands on separate lines:

```
d2 'create index schema_name.LDAP_DESC_DEID
  on schema_name.LDAP_DESC ("AEID" ASC, "DEID" ASC)
  MINPCTUSED 10 ALLOW REVERSE SCANS'

db2 'create index schema_name.OBJECTCLASS_EOC
  on schema_name.OBJECTCLASS ("EID" ASC, "OBJECTCLASS" ASC)
  MINPCTUSED 10 ALLOW REVERSE SCANS'

db2 'create index schema_name.OBJECTCLASS2
  on schema_name.OBJECTCLASS ("OBJECTCLASS" ASC, "EID" ASC)
  MINPCTUSED 10 ALLOW REVERSE SCANS'
```

Some of these indexes might exist, possibly with different names. If there are "unable to create index" errors, you can ignore them as duplicates.

Configuring Automatic Statistics Collection for the Security Directory Server Database

Administrators can use automatic statistics collection so that DB2 automatically updates the necessary database statistics. Automatic collection eliminates the necessity of manually running a periodic statistics collection against the database.

Before you begin

Enabling automatic statistics collection for Security Directory Server database requires the creation of a DB2 administration server on the system. Connect to it using the DB2 Control Center.

About this task

Automatic statistics collection is not enabled by default. For Security Directory Server to operate properly, you must exclude the LDAP_DESC, LDAP_ENTRY, and ERPARENT tables from the automatic statistics collection. You must also exclude any other tables with artificial cardinalities.

For newly created databases, run manual statistics collection (RUNSTATS) after a small data load, even if automatic collection is enabled. RUNSTATS provides statistics for good performance until DB2 initiates the first automatic collection.

Use the following variable when configuring and enabling automatic statistics collection:

SDS_database_name	Specifies the name of the Security Directory Server database, such as isimldap.

Procedure

1. Use DB2 Control Center to update the DB2 maintenance policies:
   a. Start the DB2 Control Center on a remote machine.
   b. Connect to your database with database administrator authority. If you do not see your database in Control Center, then add it to the catalog before you continue.
   c. Browse to *SDS_database_name*.
   d. Right-click *SDS_database_name*.
   e. Click **Configure Automatic Maintenance**.
   f. Click **Next** until you reach **Activities**.
   g. Select **Optimize data access (RUNSTATS)**.
   h. Click **Configure Settings**.
   i. Click **Selected tables**.
   j. Select **Use the custom filter**.
   k. At Conditions, type:
   TABNAME NOT IN ('LDAP_DESC','LDAP_ENTRY','ERPARENT')
   l. Click **Refresh Resulting Tables**.
   m. Confirm that **Resulting tables (SCHEMA.NAME)** is populated with all tables other than the ones that you specified.
   n. Click **OK** to accept the configuration.
   o. Click **Finish** to complete the wizard.
   p. Confirm the message that no errors were encountered executing the command.
   q. Quit Control Center.
2. Enable automatic statistics collection:
   a. As the database administrator, connect to the database at the command prompt.
   b. Run the following command:
   db2 update db cfg for *SDS_database_name* using auto_runstats on

How to Update Maintenance Policies *Without* Using the DB2 Control Centre:

First you will need an AUTO_RUNSTATS policy.

To get the current AUTO_RUNSTATS policy, connect to the database and use:

db2 "call sysproc.automaint_get_policyfile('AUTO_RUNSTATS', 'AutoRunstats.xml')"
The AutoRunstats.xml file will be written to the $HOME/sqllib/tmp directory.

To set the AUTO_RUNSTATS policy, use:
db2 "call sysproc.automaint_set_policyfile('AUTO_RUNSTATS', 'AutoRunstats.xml')"
The AutoRunstats.xml file location is relative to the $HOME/sqllib/tmp directory.

Here's a version of the policy that can be used for the ***ITDS DB***:

```
<?xml version="1.0" encoding="UTF-8"?>
<DB2AutoRunstatsPolicy
xmlns="http://www.ibm.com/xmlns/prod/db2/autonomic/config" >
 <RunstatsTableScope>
  <FilterCondition>tabname not in ('LDAP_ENTRY','LDAP_DESC','ERPARENT')</FilterCondition>
 </RunstatsTableScope>
</DB2AutoRunstatsPolicy>
```

Related information

- RUNSTATS command
  See the information about using the RUNSTATS command.

Updating Security Directory Server Database Statistics

DB2 requires information about the number of rows in the tables and what indexes are available so that it can efficiently fulfill queries. If Security Directory Server database is running DB2, versions 9 or 10, RUNSTATS can be configured to run automatically.

About this task

Note: DB2 REORGCHK does not update index statistics and is not a replacement for RUNSTATS.

If enabling automatic statistics collection is not feasible, you must run RUNSTATS manually. It is important to update table and index statistics after large Directory Server Markup Language (DSML) loads, HR feeds, and reconciliations.

If you experience high processor usage or poor DB2 performance, run RUNSTATS on all of the tables in the database. To update index statistics, run the RUNSTATS command on each table individually. IBM Security Identity Manager performance tuning scripts (perftune_runstats.sh and perftune_runstats.bat) detect the version of DB2 and run the RUNSTATS command against all tables for a specific schema in a database.

If you run the RUNSTATS command in a working environment, make sure that the connected applications can continue to write to the database. Use the allow write access option so users can write to a database while RUNSTATS runs.

Use RUNSTATS on an idle or lightly used database because it requires update locking on the system statistics table to update the database statistics. The system acquires locks on the tables that are used by the database optimizer to fulfill queries. The locks might cause transaction rollbacks on a database with a heavy load.

In addition to running RUNSTATS on all tables in the database, you must manually update the statistics table for the LDAP_DESC, LDAP_ENTRY, and ERPARENT tables. The choices DB2 makes about when to use these tables for fulfilling queries for the Security Directory Server are not ideal for IBM Security Identity Manager. Manually
adjusting the statistics table helps DB2 make better choices and use these tables at the end of the access plan instead of the beginning. *In scale environments, it is recommended to sto ibmslapd before runsats/reorg.*
The reason is because reorg completion could be delayed as it waits for locks held slapd.
Specifically updates to LDAP_ENTRY table will be delayed until ibmslapd is stopped

The following procedure uses RUNSTATS on every table in the ISIMLDAP schema.

Procedure

1. Connect to the database as the database administrator.
2. Generate a listing of all tables in the schema by running the following command:
   db2 list tables for all | grep ISIMLDAP
3. For each table in the ISIMLDAP schema, run the following command on a single line:
   db2 runstats on table ISIMLDAP.*table_name* and indexes all allow write access
4. Manually update the database statistics table for the LDAP_DESC, LDAP_ENTRY, and ERPARENT tables. Run the following commands on separate lines:

```
db2 update sysstat.tables
  set card = 9E18
  where tabname = 'LDAP_DESC' and card <> 9E18
db2 update sysstat.tables
  set card = 9E18
  where tabname = 'LDAP_ENTRY' and card <> 9E18
db2 update sysstat.tables
  set card = 9E10
  where tabname = 'ERPARENT' and card <> 9E10
```

Related tasks
"Configuring Automatic Statistics Collection for the IBM Security Identity Manager Database"
Administrators can use automatic statistics collection so that DB2 automatically updates the necessary database statistics. Automatic collection eliminates the necessity of manually running a periodic statistics collection against the database.

Related information
ISIM Performance Tuning Scripts
Download performance tuning scripts for IBM Security Identity Manager.

Configuring the Maximum Open Files

To work well with other applications that are running on the system, DB2 sets a limit on the number of files it keeps open with the maxfilop setting. You can adjust this number to meet the needs of your environment.

About this task

After reaching the specified limit, DB2 closes a currently open file to open the new one. This process can cause a performance loss on systems that do not require a restriction on the number of open files. The default value is often too small, particularly for larger directories.

Increasing this value is important for the SMS table spaces that IBM Security Directory Server uses.

Use the following variables when configuring the maximum open files:

ISDS_database	Specifies the name of the Security Directory Server database, such as ldapdb2.
max_files_open	Specifies the maximum number of files DB2 can have open at any one time. Initial value: 64. Suggested value: 256.

Procedure

1. Connect to the database as the database administrator.
2. Run the following command:
   db2 update db cfg for *iSDS_database* using maxfilop *max_files_open*

Disabling Hash Joins

The DB2 optimizer can use several different join types when determining the most efficient means to fulfill a query.

About this task

For Security Directory Server, a hash join is seldom, if ever, the correct approach. This procedure describes how to disable hash joins.

Procedure

1. As the database administrator, run the following command:
   db2set DB2_HASH_JOIN=NO
2. Stop and restart the IBM Security Directory Server database. The changes take effect when the database restarts.

Improving Disk I/O Performance

Disk I/O performance is highly dependent upon the drive types, layout, and configuration.

About this task

The following DB2 registry variables might improve performance on some systems.

See the DB2 documentation to find out whether the setting applies to your environment.

System	**Setting**
Systems with SAN, RAID, or other advanced disk subsystem	DB2_PARALLEL_IO=*
All systems	DB2_USE_ALTERNATE_PAGE_CLEANING=ON DB2_SELECTIVITY=YES DB2_ANTIJOIN=EXTEND

Chapter 12. Improving Operating System Performance

You can improve performance on some systems by making some operating system-specific changes. This information serves only as a guideline. Consult the documentation for your middleware, and apply any required operating system tuning.

AIX

You might improve performance by tuning the virtual memory-management (VMM) settings such as minperm and maxperm. For more information about this subject, consult the AIX documentation.

Ensure that there is at least as much swap space as there is physical RAM on the system. Insufficient swap space can result in out of memory messages due to how the operating system handles memory allocations.

Enable Large File support for all file systems by using Journaled File Systems (JFS). Large File support is not required for file systems that use Enhanced Journaled File Systems (JFS2) as JFS2 supports large files natively.

Solaris

Ensure that there is at least as much swap space as there is physical RAM on the system. Insufficient swap space can result in out of memory messages due to how the operating system handles memory allocations.

Chapter 13. Multiple Account Per Person (MAPP) Tuning

The **datacenter scenario** is a relatively recent phenomenon. In this scenario there are relatively few accounts (from thousands to half a million) , several thousand services and a large majority of these accounts are owned by a small number of users (typically service owners). There is much variety in the types of customers using this approach (telecommunications, banks, large retailers, and some internal IBM deployments)

ISIM 6.0 handles these environments with varying degrees of success. New challenges are constantly being discovered; troubleshooting the challenges in the datacenter scenario is happening both pro-actively in the lab and reactively to PMRs and critical situations in the field.

Multiple Account per Person (MAPP) Specific Tuning Modifications

WAS Threads

- Decrease JMS worker threads per node to 3 (default is 5)

Justification:

Decreasing the number of JMS worker threads per node appears to be critical in avoiding out of memory (OOM) error conditions. When running with the default of 5, MaPP scenarios consistently resulted in OOM errors. It appears that decreasing the concurrency causes the server to utilize less of available heap memory for the MaPP scenarios to complete without incident.

Buses > itim_bus > Bus members > Messaging engines for Application_Cluster > Application_Cluster.000-itim_bus > Mediation thread pool

Buses > itim_bus > Bus members > Messaging engines for Application_Cluster > Application_Cluster.001-itim_bus > Mediation thread pool

Buses > itim_bus > Messaging engines > Application_Cluster.001-itim_bus > Mediation thread pool

WAS LDAP Cache changes

- Increase default LDAP cache sizes by 3x

(default is 20k and 2k increased to 60K long term, 6K short term cache)

Justification:

After cache has reached capacity, relatively simple searches will take extended amounts of time to complete

- Offload LDAP cache to disk

(default is for cache to be stored in memory)

Justification:

Instead of storing cache in memory, better performance is observed for those cluster machines with good I/O

ISIM Policy Simplification

- Modify default ISIM Identity Policy and simplify as needed
- Simplify JavaScript in all provisioning and identity policies. In terms of JavaScript rules, simplify as much as possible

ISIM Relational Database changes

- Transaction log size increased from 10K to 20K on ISIM Relational DB

Justification:

This is to account for potential increase in ISIM related transaction times

Chapter 14. Virtual Machines and Virtual Appliances

VA Tier

Install IBM Security Identity Manager as a virtual appliance or as a virtual machine. These deployments consist of either a single server or as a clustered server environment with member nodes. Clustered virtual appliances/virtual machines are recommended due to the distinct advantages of load balancing, fault tolerance and task scalability.

Data Tier

For performance tuning/troubleshooting purposes, it is recommended that both the database server and directory server are located on separate *physical machines* and are configured for optimal disk drive I/O performance. This recommendation is based on the overhead associated with the virtualization of I/O bound activities.

ISIM Virtual Appliance or Virtual Machine Resource Allocation

Resources such as memory, CPU (virtual sockets and cores), and storage (HDD/SSD) should be efficiently allocated at virtual machine creation time.

Virtual Appliance/Virtual Machine Recommendations

- ***CPU***

  4 Socket, 2 Cores per socket (dedicated)

- ***Storage***

  100GB HDD/SSD

- ***Memory***

  16-20 GB Memory

Data Base Recommendations

- ***CPU***

  4-8 Sockets, 2 Cores per socket (dedicated)

- ***Storage***

  150 GB HDD/SSD

- ***Memory***

  8-12 GB Memory

Directory Server Recommendations

- ***CPU***

  4-8 Sockets, 2 Cores per socket (dedicated)

- ***Storage***

  300 GB HDD/SSD

- ***Memory***

  12-16 GB Memory

Physical and Logical Processors

When running the ISM Virtual Appliance or ISIM on a virtual machine, it is very important to dedicate actual physical CPUs on the hypervisor to the logical CPUs on the virtual machine. (ex: If the hypervisor has 'x' total sockets, dedicate 4 sockets for each Virtual Appliance instance). This would also apply to the data tier if running on a virtual machine (not recommended for I/O bound applications and activities). Since DB2 is enabled for multi-threaded applications, applications will perform best on a multi-processor server which has allocated dedicated sockets to ISIM operations. Keep in mind, even in a well-tuned environment, system bottlenecks might vary between the processor, memory, and disk on the associated ISIM components.

Performance limiting factors to take into consideration:

- network throughput constraints
- firewall throttling
- network intrusion prevention systems
- network intrusion detection systems

VA Tier and Data Tier Storage

System administrators and database administrators can adjust the amount of disk space available for the VA/VM Tier and Data Tier. Each of the middleware components uses different amounts of disk space for various purposes.

Virtual Machines: Thick Clients

Virtual machine thick provisioning is based on the concept of reserving all necessary space on the hard drive at the time of virtual machine creation.

Thick Provisioning Advantages

- Restricts allocation of virtual data stores based on physical HDD constraint thus preventing error cases in which virtual capacity exceeds physical capacity
- Under certain conditions, benchmarks seem to indicate better I/O performance because all of the blocks on the disk will be pre-zeroed, thus removing the need to zero the blocks at write time. This performance benefit only applies to a latency sensitive, IO bound applications.

Thick Provisioning Disadvantages

- Storage space is allocated at a substantially faster rate.
- Inefficient usage of storage if initial size estimates are incorrect.

Thick Provisioning Options

- **Lazy Zeroed Thick:** The hypervisor allocates the space on the VMFS at virtual machine creation time. However, blocks of data are utilized on the back-end data store at **write-time** on the virtual machine.
- **Eager Zeroed Thick (recommended)** Hypervisor both reserves all the space on the VMFS and zeros out the disk blocks at creation time. Based on results in the laboratory, virtual machines created with eager zeroed thick provisioning will require more time to initialize. In terms of long term performance, this is the optimal deployment solution because the blocks have already been zeroed-out thus reducing the overhead associated with writing to disk.

Virtualization References

Configuring Disks to Use VMware Paravirtual SCSI (PVSCSI) Adapters

Performance Best Practices for VMware vSphere® 6.5

Performance Best Practices forVMware vSphere 6.7

Virtualization–The Catalyst for Change: Best Practices for Virtualizing IBM DB2 with Technologies from VMware, Intel, and NetApp

Chapter 15. Best practices

You can set up and configure IBM Security Identity Manager in many ways. Use this information to determine the best configuration for your environment.

Hardware best practices:

Consideration	Best practice
Database and directory activity can be CPU- and memory-intensive.	Allocate each application at least **4 processors(sockets) and 8 GB of RAM**. More processors are better. *For optimal performance, do not have all IBM Security Identity Manager components on the same system.*
In general, network latency is not a major performance bottleneck, but components can degrade performance. Components include the IBM Security Identity Manager server components, the directory server, database server, agents, and agent endpoints.	Try to have as few hops as possible between components. If possible, install all components on the same subnet or no more than one hop away. Put components on a 1 gigabit or faster network. Do not attempt to span a Websphere Cell across different data centers
Allocation of processor resources in an AIX LPAR can affect system performance.	Suggested actions listed in order of potential benefit: • Disable SMT for IBM Security Identity Manager nodes. • Use the most current version of the WebSphere Application Server JVM. • Give at least 4 physical CPUs to each AIX LPAR. • Use dedicated processors rather than virtual ones.
Disk bottlenecks can negatively affect performance.	Use multiple disks rather than a single large disk: • IBM DB2 and Oracle can use multiple disks, but you must configure them to do so. • High-end I/O backplanes or other advanced storage systems can balance the I/O load across multiple disks automatically.
Do not use physical disks in a SAN failover environment across multiple data stores. For example: If the database for each LDAP server is on the same physical devices, I/O performance problems are likely to develop.	Use separate physical devices in the SAN for the underlying data store of each failover.

Software best practices:

Consideration	Best practice
Each agent modifies the LDAP schema by adding new attributes to support a new service. These attributes are created without indexes.	For services that manage thousands of users, you realize significant benefit by adding indexes to attributes that have many members.
Complicated provisioning policies can result in complicated directory and database queries with poor performance.	Policies with small numbers of roles and services perform best.
Provisioning policies without account approval work flows perform better than policies with account approval workflows due to optimizations for the former case. Provisioning policies created by the	If it is not needed, remove the default account workflow from the provisioning policy to improve performance.

system when a service is created use a Default Account Request Workflow.	
Dynamic roles affect people in a scope, either one-level or subtree. When a person object in that scope is modified or added, the system must re-evaluate that role. This process is true for every dynamic role in the system. Take for example a situation in which there are three dynamic roles with subtree scope. If a person object in that scope is updated, the system must reexamine all three dynamic roles.	• Limit the number of dynamic roles, either by number or by scope, that affect person objects that are modified frequently. It does not matter if the dynamic role ultimately enrolls the person or not: the evaluation affects the performance. • When creating dynamic roles that apply to all people within an organizational unit, place the dynamic role inside the organizational unit and use the filter (objectclass=*). This filter yields better performance from the directory server than a filter like (cn=*).
When creating a role hierarchy, the order in which the hierarchy is created can affect performance due to any associated provisioning policies that are enforced.	When adding multiple new roles to an existing role hierarchy, create the parent-child relationship between all new roles first. Then, create the parent-child relationship between the new role and the existing ones. This process limits the number of policy evaluations. If possible, create the entire hierarchy before adding any of the involved roles to a provisioning policy.
Evaluation of ACIs affects performance.	• Limiting the scope (through placement within the organizational tree) and number of ACIs increases performance by requiring fewer evaluations. • When doing a person search through the APIs, limit the scope of your search to be as narrow as possible. Limiting the scope prevents unnecessary evaluations.
When updating a person object, the system must re-evaluate all provisioning policies in which the user to see whether the update changes a provisioning action. The guideline applies to both manual or automated methods such as an HR feed or JNDI update.	• Store only the person information that is needed for policy evaluation and account management in IBM Security Identity Manager. This practice reduces attribute updates that are not used for policy enforcement. • Minimize person object updates when possible.
IBM Security Identity Manager includes searches on the O, OU, and L attributes for the organizational chart. This search can slow down if large numbers of users have these attributes. This consideration is important for large user populations.	When loading users in bulk, do not include O, OU, or L attributes on the person records. Follow this guideline when using a DSML file or an IDI Feed.
Numeric erGlobalIDs allows the application to make more efficient use of memory when processing reconciliations.	When loading objects such as people or accounts directly into the IBM Security Identity Manager directory server, use all numeric values for the GlobalID, not alphanumeric values. You might perform this action during an initial LDIF data load.
Having the same value for the family name (sn) attribute for all users in a test environment results in poor performance. This attribute is used by the default identity policy to determine a unique UID for an account. Therefore, performance degrades when the identity policy creates an ID for a service due to the resulting LDAP lookups.	When loading a test environment, make sure that users have unique IDs for their family name (sn) attribute.

Often administrative accounts on target systems are mapped to a single person object. This mapping can result in 1 person with possibly thousands of accounts. It can degrade performance or result in Java OutOfMemory errors.	Limit the number of accounts that person objects has.
Complex workflows (operational, account request, and access request workflows) can degrade performance.	Keep frequently used workflows, such as the modify person operational workflow, as simple as possible.
In a workflow, the result of each transition is that a message is placed on the JMS queue Transitions. The JMS queue Transitions also serializes and deserializes data from the database.	Design workflows so that they have the fewest number of transitions from start to finish as possible. Consider reducing the number of nodes by: • Combining adjacent scripts nodes • Combining a non-script node followed by a script node by moving the script node contents into the PostScript for a non-script node • Creating an initial node at the beginning of a long workflow to jump to a specific node later to shortcut unnecessary transitions for common paths
Poor non-cached LDAP lookups within a workflow can negatively affect performance.	Store the results of redundant lookups as relevant data items for reuse in later nodes.
Relevant data must be serialized and deserialized from the database for each node transition.	Keep the quantity and size of workflow relevant data objects as small as possible.
Each call to process.auditEvent() adds more data that must be written to the database.	Minimize the amount of logging done in workflow nodes. You can use process.auditEvent() calls during development and testing but comment out these lines before promoting the code into production.
HR feeds for IBM Security Directory Integrator use the following types: **Push feed** Security Directory Integrator uses the IBM Security Identity Manager JNDI unsolicited notification feature to push records into IBM Security Identity Manager. The push method is single-threaded. It requires that IBM Security Identity Manager confirm the JNDI operation completed successfully before proceeding to the next object. **Pull feeds** IBM Security Identity Manager requests all available records from a Security Directory Integrator DSML version 2 Event Handler assembly line through a reconciliation. This method streams all objects directly into IBM Security Identity Manager. Any available cluster member can then act on the update operations.	To ensure optimal HR feed performance, use the pull method wherever possible. Use the push method for asynchronous updates or updates that are not performance-sensitive.
Environment stability can be compromised by having other applications deployed in the same WebSphere JVM where IBM Security Identity Manager is deployed.	When installing into a shared WebSphere environment, install IBM Security Identity Manager into an existing cell or node but on a separate application server. You can tune the application server without affecting other applications.
Using an existing instance when installing into a shared DB2 environment might limit tuning	Use a separate instance for the IBM Security Identity Manager database to yield the best performance.

possibilities and negatively affect other databases in the instance. This limitation includes the instance for Security Directory Server.	

ISIM Virtual Appliance/Virtual Machine Best Practices

Virtualization Consideration	Best practice
Selecting machines for ISIM Application Server and Data Tier	• It is recommended that customers planning for an ISIM deployment use RedHat/SuSe as the Guest OS for ISIM application servers and use physical machines for deploying SDS and DB2 (Data Tier) • Deploying ISIM 6 using physical machines provided the best performance.
Creating Virtual Machine in ESXI Server	• Select Virtual machine version 8 as the target VMware configuration • Install VMware tools on your guest OS. When installed on the guest O/S, VMware tools provide additional benefits to the ISIM 6 deployment such as paravirtualized drivers, better time synchronization between VM and the vSphere host, and better memory management capability.
Deploying the ISIM Virtual Appliance or ISIM on a virtual machine	Follow best practices in place for virtual infrastructure systems designed using VMware vSphere. Refer to Performance Best Practices for VMware vSphere available at: http://www.vmware.com/pdf/Perf_Best_Practices_vSphere5.1.pdf http://www.vmware.com/pdf/Perf_Best_Practices_vSphere5.5.pdf http://www.vmware.com/files/pdf/techpaper/VMware-PerfBest-Practices-vSphere6-0.pdf
Deploying ISIM DB2 or SDS on a virtual machine	Use Paravirtual SCSI (PVSCSI) adapters. The PVSCSI adapters are high performance storage adapters that can result in greater throughput and lower CPU utilization.
Allocating virtual sockets and cores on the virtual appliance/machine	• Set the Number of virtual sockets and the Number of cores per socket to match the number of cores and sockets on your machine. Do not exceed the physical constraints of the system • Generally speaking, the type of workload will determine the performance of the ISIM virtual deployment. Significant degradation in performance was observed during reconciliation (I/O intensive) related tasks while the least degradation in performance was observed for Self-service UI driven (CPU intensive) tasks.

Virtualization Consideration	Best practice
Designating Virtual Network Adapters for Virtual Machines or for the Virtual Appliance.	Use VMXNet 3 for Network adapter as this demonstrates significant performance improvements as compared to the default E1000.
Running SDS and the ISIM RDB on a virtual machine (not recommended)	Customers planning to virtualize their entire ISIM deployment should use RedHat/SuSe as the Guest OS running on ESXi 5.1/5.5/6.0 Hypervisor.

Related information
IBM Security Identity Manager best practices wiki
See the information about best practices for IBM Security Identity Manager.

Chapter 15. Planning a Maintenance Schedule

Perform regular maintenance to maintain optimal performance for IBM Security Identity Manager environment.

About this task

You can find the latest up-to-date best practices on the IBM Security Identity Manager wiki at ISIM Wiki. These suggestions are from that wiki. To maintain your environment, complete the following tasks at appropriate intervals.

Procedure

- Clean out the recycle bin.
  If enabled, regularly empty the IBM Security Identity Manager recycle bin. As the number of objects in the recycle bin increase, LDAP performance can degrade. The frequency with which you empty the recycle bin depends on how frequently deletes occur in the system. Disable the recycle bin for systems that do not need it.

- Update database statistics.
  Update database statistics after many updates or on a weekly basis for most environments. Updating database statistics in the underlying databases can significantly improve performance. This maintenance task applies to the DB2 or Oracle Database used by IBM Security Identity Manager. It also applies to the DB2 database used by the Security Directory Server. Consider enabling automatic RUNSTATS if your environment meets the software and configuration criteria.

- Clean out the IBM Security Identity Manager database.
  Keep the IBM Security Identity Manager database as small as possible for efficient database access. Cleaning involves regular purging of database records that are no longer required for auditing or transactional purposes.

- Evaluate and apply fixes.
  IBM releases software updates on a regular basis for IBM Security Identity Manager and its supported middleware. Check for updates on a quarterly basis to ensure that your environment is up to date. Test fixes thoroughly in a test environment before applying them to the production environment. Consider the information IBM provides about compatibility before applying updates.

- Access the latest version of the tuning guide.
  Just like software updates, the tuning guide is updated on a regular basis with new tuning information to improve performance. Check the IBM website every quarter to see whether a new guide is released. Always test a new tuning change in a test environment, including load testing, before applying it to the production environment. Look for the latest tuning scripts, which also undergo revisions, from IBM.

- Take regular backups.
  Backups do not contribute towards performance of the IBM Security Identity Manager environment, but perform them as part of regular maintenance.

Related concepts
"Using the recycle bin"
When you enable the recycle bin and then delete objects from IBM Security Identity Manager, the software moves them to the recycle bin.

Related tasks
- "Updating IBM Security Identity Manager database statistics for DB2 databases"
  DB2 requires statistics on the number of rows in the tables and available indexes to efficiently execute queries. DB2 version 9 can update the statistics automatically, or you can manually update the statistics.

- “Updating Security Directory Server Database Statistics”
  DB2 requires information about the number of rows in the tables and what indexes are available so that it can efficiently fulfill queries. If Security Directory Server database is running DB2, version 9, you can set RUNSTATS to run automatically. Version 9 is the default for Security Directory Server, version 6.1. RUNSTATS eliminates the need for running it manually.
- “Updating IBM Security Identity Manager database statistics for Oracle databases”
  You must gather and update database statistics at regular intervals. Intervals can be one week to one month on a production IBM Security Identity Manager system or after processing a large amount of data.
- “Configuring automatic statistics collection for the IBM Security Identity Manager database”
  Administrators can configure automatic statistics collection so that DB2 automatically updates database statistics. Automatic collection eliminates the necessity of manually running a periodic statistics collection against the database.
- “Configuring automatic statistics collection for the Security Directory Server database”
  Administrators can use automatic statistics collection so that DB2 automatically updates the necessary database statistics. Automatic collection eliminates the necessity of manually running a periodic statistics collection against the database.
- “Controlling the size of the database”
  To maintain optimum performance, use the DBPurge utility included with IBM Security Identity Manager to automate removing entries over a certain age from the database.

Chapter 16. Troubleshooting ISIM

Middleware dependencies can complicate the task of finding performance problems with IBM Security Identity Manager. For example, a slow DSML feed with account provisioning might be caused by a slow directory server, database locking, or insufficient worker threads.

This information is designed to assist you in identifying problem areas and provide some pointers on fixing them. Information is provided with the assumption that you read and applied the tuning.

Security Directory Server Outages

Incorrect system or product configuration can cause Security Directory Server to fail, hang, or disappear due to resource restrictions.

Symptoms

The directory server fails or hangs for no obvious reason.

Diagnosing the problem

Check the size of your entry cache. If the entry cache size causes the Security Directory Server process to grow beyond what is supported by your operating system memory model it can fail. A typical system memory model is 2 GB on 32-bit operating systems.

The ibmslapd process might be hitting an artificial system limit, such as a ulimit.

Resolving the problem

Decrease the size of the Security Directory Server entry cache, or increase the ulimits for the process.

Related tasks
"Configuring system limits"
System limits (ulimits) might prevent the Security Directory Server process from accessing enough real or virtual memory. To avoid memory dumps or stopping without indication, increase the ulimits.

Security Directory Server Slow Queries

Slow queries from IBM Security Directory Server can degrade overall system performance.

Symptoms

Poor search performance when using IBM Security Directory Server.

Causes

You might see poor performance due to:
- Long-running queries that need indexes.
- Low buffer pool hit ratio.

Diagnosing the problem

Determine the specific cause or causes for poor search performance.

Long-Running Queries

Check for long-running queries that need indexes. The Security Directory Server uses DB2 to process LDAP queries. By checking DB2 for long-running queries, you can discover what attributes need indexing.

1. To find how long each query takes, turn on statement cache monitoring in DB2.
   db2 update dbm cfg using DFT_MON_STMT ON
2. Stop the directory server, restart the database, and restart the directory server.
3. After monitoring is turned on, duplicate the suspected action in IBM Security Identity Manager.
4. Get a snapshot of the statement cache:
   db2 get snapshot for dynamic SQL on *database_name*
   Example: The snapshot contains stanzas like this one:

```
Number of executions = 1
Number of compilations = 1
Worst preparation time (ms) = 3
Best preparation time (ms) = 3
Internal rows deleted = 0
Internal rows inserted = 0
Rows read = 10024
Internal rows updated = 0
Rows written = 0
Statement sorts = 0
Total execution time (sec.ms) = 136.000663
Total user cpu time (sec.ms) = 62.010000
Total system cpu time (sec.ms) = 10.000000
Statement text =
     SELECT distinct E.EID
     FROM LDAPDB2.LDAP_ENTRY AS E,    LDAPDB2.LDAP_ENTRY as pchild
     WHERE E.EID=pchild.EID AND pchild.PEID=?
     AND E.EID IN (SELECT EID FROM LDAPDB2.OU WHERE OU = ?)
```

5. Calculate the average execution time per query. Divide the total execution time by the number of executions: total execution time / number of executions.

   In the preceding example: 136 / 12 = 11.33 seconds per execution.
   Queries typically take one second or less. Queries that take longer might be searching on columns that are not indexed by DB2. If they are not indexed in DB2, they are not indexed in the Security Directory Server.

Another symptom of this problem is a high average number of rows read, which is calculated by dividing the rows read by the number of executions.

In the preceding example, the column OU is probably not indexed. IBM Security Identity Manager tuning scripts provide the perfanalyze_dynamicsql.pl script that calculates the time per execution for all stanzas and sorts the results.

Low Buffer Pool Hit Ratio

Security Directory Server uses DB2 to process LDAP queries. Security Directory Server database requires a high (greater than 95%) hit ratio. If the buffer pools are not large enough, DB2 must read more information from the
disk. Reading the disk can result in high I/O wait.

See "Calculating the buffer pool hit ratio".

IBM Security Identity Manager tuning scripts provide the perfanalyze_bufferpools.pl script that calculates the hit ratio for all buffer pools.

Resolving the problem

Take the appropriate action or actions to improve query performance:

Long-running queries	Index any attribute in the Security Directory Server that is not indexed. See "Configuring attribute indexes for Security Directory Server."
Low buffer pool hit ratio	Increase the memory allocated to the buffer pools. See."Configuring database buffer pools for the Security Directory Server database."

Related tasks

- "Configuring attribute indexes for Security Directory Server."
  Indexing the attributes on which applications search increases Security Directory Server performance. Security Directory Server indexes automatically translate into DB2 indexes when you update the Security Directory Server schema for those attributes.
- "Configuring database buffer pools for the Security Directory Server database."
  DB2 buffer pools are the secondary buffer for Security Directory Server. These buffer pools must be large enough so that most table searches can be read directly from memory instead of using the disk.

Related information

ISIM Performance Tuning Scripts
Download performance tuning scripts for IBM Security Identity Manager.

Governing Policy Search Errors

Searches for governing policies can fail due to statement heap constraints.

Symptoms

The trace.log file contains the Error searching for governing policies message.

Causes

The statement heap for the Security Directory Server database is too small, which causes large LDAP queries to fail.

Resolving the problem

Increase the statement heap.

Related tasks

"Configuring database statement heaps"
You can increase the size of the DB2 statement heap (stmtheap) to eliminate errors caused by long queries.

Java Out Of Memory Errors

Out Of Memory (OOM) errors can occur if the Java Virtual Machine (JVM) heap is too small.

Symptoms

The trace.log file contains Java OutOfMemory errors.

Causes

The message is from WebSphere Application Server. The Java virtual machine (JVM) ran out of heap size.

Resolving the problem

Increase the maximum heap size if possible, and restart the application server. If the heap size is already at the limit, break up transactions. For example, you might use fewer services or roles in a provisioning policy.

Related concepts

"Using the DSML connector with Security Directory Integrator"
You can use the DSML connector to create custom agents for returning information to IBM Security Identity Manager.

Related tasks

- "Adjusting the Java virtual machine size"
  IBM Security Identity Manager, version 6.0, runs on 64-bit JVMs on supported platforms. Using a 64-bit JVM, you can allocate 2 GB or more of memory. You might need to allocate more memory for large (more than 6 million accounts) reconciliations.
- "Configuring paged searches"
  IBM Security Identity Manager, version 6.0 and later, incorporates LDAP paged searches to alleviate JavaOutOfMemory errors in large environments.

Transaction Rollback Errors

Transaction rollback errors can occur due to database resource constraints.

Symptoms

The trace.log file contains transaction rollback errors.

Causes

Transaction rollbacks can occur for several different reasons, most of them database-related. An error message in the trace.log file can provide more information about what went wrong. Some areas to check when you get a transaction rollback:

- Lack of database storage space.
- Database locking issues.
- Database memory issues.

Diagnosing the problem

Determine the specific cause or causes for rollback errors using the following table:

Storage space	If the database runs out of storage space for the table spaces, a transaction rollback error can occur.
Locking	If the database encounters extreme locking issues, it might cause a transaction rollback error.
Memory	If there is not enough memory available to database structures to fulfill the requested query, a transaction rollback error might occur. The JNDI error in the trace.log file can indicate which database heap to increase.

Resolving the problem

Take the appropriate action or actions.

Storage space	Increase the amount of disk space allocated to the table spaces.
Locking	Confirm that the locks are tuned appropriately. Update the table statistics.
Memory	Increase the appropriate heap for the specific middleware.

Related concepts

- "Adjusting lock list and maximum locks"
  The default settings for the DB2 lock list (locklist) and maximum locks (maxlocks) are adequate for most environments.
- "Changing the lock timeout"
  The default lock timeout value (locktimeout) in the IBM Security Identity Manager database is infinity. You can adjust this value if locking problems occur.

Related tasks

- "Configuring table spaces for IBM DB2 databases"
  IBM Security Identity Manager uses a database managed space (DMS) table space to store data. This type of table space performs better than system managed space (SMS) table spaces, but you must preallocate disk space for the database to use. The tables spaces created by the installer have autoresize enabled and grow as needed.
- "Configuring table spaces for Oracle databases"
  During database configuration, IBM Security Identity Manager creates several small table spaces that can automatically extend as necessary. You can add additional data files.
- "Updating IBM Security Identity Manager database statistics for DB2 databases"
  DB2 requires statistics on the number of rows in the tables and available indexes to efficiently execute queries. DB2 version 9 can update the statistics automatically, or you can manually update the statistics.
- "Updating IBM Security Identity Manager database statistics for Oracle databases"
  You must gather and update database statistics at regular intervals. Intervals can be one week to one month on a production IBM Security Identity Manager system or after processing a large amount of data.
- "Configuring database application heaps"
  Some of the queries that the IBM Security Identity Manager application submits to the DB2 server result in complex SQL statements. If you see transaction rollback errors in the trace.log file, increase the values of the heaps in increments of 256 until the errors stop.

Chapter 17. Identifying Performance Bottlenecks

Multiple middleware dependencies can complicate finding performance problems with IBM Security Identity Manager. Identifying the performance bottleneck requires a step-wise approach.

The following guidelines can help you identify performance problems:

- Monitor the processor and disk usage of every server to see which server is most heavily used. The servers include IBM Security Identity Manager nodes, directory, and database. Based on this information, review the monitoring and tuning steps specific to that component.

- Either the database or the directory server might be a bottleneck during heavy usage or large provisioning changes. IBM Security Identity Manager makes intense usage of its database and directory server. The database is an information staging area and audit trail for provisioning actions. The directory server is a permanent storage location that can be heavily queried when evaluating provisioning policies.

- An incorrectly tuned directory server can become the bottleneck as the IBM Security Identity Manager server waits for the result set before starting the required provisioning action.

  During an action that evaluates a large provisioning policy that affects many users, the affected users must be queried from the directory server. Examples of a large evaluation include adding a policy or modifying an existing policy. The directory server evaluates the query and returns the matching users. Make sure that the directory server fulfills the requested queries as quickly and efficiently as possible to minimize this behavior.

- After the result set is returned, the IBM Security Identity Manager server begins enforcing the provisioning policy for each user. This process is multithreaded and benefits from multiple processors. If it seems that the processors on the server are not fully used, check for a bottleneck on the LDAP or database server.

- Enforcement actions can cause access contention and locking in the database. The database stores any enforcement action required for a user (account addition, modification, or deletion) as a workflow item. When a thread becomes available, it queries the database for the next workflow item that requires processing and then acts on that item. Appropriate indexes and access plan statistics can minimize the number of required locks for filling these requests.

Related concepts

- "Tuning Security Directory Server"
  When tuning IBM Security Directory Server, it is important to understand the interaction between the IBM Security Directory Server process and DB2.

Related tasks

- "Tuning IBM DB2"
  IBM Security Identity Manager, version 6.0 and later, works with DB2 for Linux, UNIX, and Windows starting with Version 9. Version 9 has auto-tuning mechanisms that can reduce administrative and maintenance tasks.
- "Tuning Oracle"
  IBM Security Identity Manager supports Oracle databases, starting with version 10g on some operating systems.

Chapter 18. IBM Security Identity Manager Monitoring

Tuning an IBM Security Identity Manager system requires monitoring systems to determine environment bottlenecks.

Related information
Tuning information management systems
See the information in the DB2 V9 information center.

IBM Security Identity Manager Deployment Health Monitoring

IBM Security Identity Manager provides deployment monitoring features. These features include monitoring of performance and availability of various requests in the key components. The provisioning and workflow components are added with instrumentation that tracks events in the WebSphere Performance Monitoring Infrastructure (PMI) system. They provide statistics about the server runtime operations.

The provisioning statistics include failure and recovery information and performance information summarized by the service type. The following table describes the provisioning statistics:

Statistic Name	Description
AccountRequests	The total number of account requests that were attempted for the service.
AccountRequestTime	The average response time for all account requests in milliseconds.
AccountSearchRequests	The total number of account search requests (reconciliations) that were attempted for the service.
AccountSearchTime	The average response time for the initial search and first entry to be returned from the adapter in milliseconds.
ActiveRequests	The number of concurrently active account provisioning requests.
BlockedRequestsCompleted	The number of provisioning requests that were re-executed and completed after the service recovery.
BlockedRequestsCreated	The number of provisioning requests that were blocked due to a service failure.
HungRequests	The number of requests that are currently suspected to hang.
ServiceFailuresDetected	The number of times a service was determined to fail due to a communication or authentication problem.
ServiceRecoveriesAttempted	The number of times a service recovery was attempted. Service recoveries occur periodically in the background or interactively through the “Retry blocked requests” action.
ServiceRecoveriesCompleted	The number of times a service was successfully detected as available after a failure.

The workflow request statistics include performance information summarized by the workflow process type. Statistics are tracked for both requests and processes, where requests are a subset of the processes that are the top-level or root workflow processes. The following table describes the request statistics:

Statistic name	Description
RequestsStarted	The number of requests of this type that were started.
RequestsSubmitted	The number of requests of this type that were submitted.

RequestsCompleted	The number of requests of this type that were completed.
RequestCompletionTime	The average time to complete requests of this type. It includes any time that the request was waiting on user input, or any child processes to complete.
ProcessesStarted	The number of processes of this type that were started.
ProcessesSubmitted	The number of processes of this type that were submitted.
ProcessesCompleted	The number of processes of this type that were completed.
ProcessesCompletionTime	The average time to complete processes of this type. It includes any time that the request was waiting on user input or any child processes to complete.

All statistics are tracked on a per-server basis and are reset if the server is restarted. The IBM Security Identity Manager server PMI statistics are available through WebSphere Java Management Extensions (JMX) API. Additionally, application API access is provided to gather per-service statistics directly from the IBM Security Identity Manager server. Each can be used to provide monitoring through a third-party monitoring tool, such as IBM Security Monitoring.

Viewing the Monitored Values

Use WebSphere Application Server administrative console to view the monitored values.

Procedure

1. Log on to the WebSphere Application Server administrative console.
2. Expand **Monitoring and Tuning** > **Performance Viewer**.
3. Select **Current activity**.
4. Select an application server.
5. Expand **Performance Modules**.
6. Select IBM Security Identity Manager Module and then click **View Module**. Or, optionally select the more specific monitoring modules such as Provisioning Statistics, Request Statistics, or any of their child monitors.
7. View the monitored values from the table. Or, select statistics to view in the graph.

Enabling the Health Monitoring

Use the WebSphere Application Server administrative console to enable the health monitoring feature.

Procedure

1. Log on to the WebSphere Application Server administrative console.
2. Expand **Monitoring and Tuning**.
3. Select **Performance Monitoring Infrastructure (PMI)**.
4. Select an application server.
5. Select the **Enable Performance Monitoring Infrastructure (PMI)** check box.
6. For **Currently monitored statistic set**, the monitors are enabled automatically at the **Basic**, **Extended**, or **All** setting. Select **Custom** to avoid problems with other statistics and the URL structure of the IBM Security Identity Manager user interface.
7. Click the **Custom** link and then go to the **Runtime** tab.
8. Expand the IBM Security Identity Manager Module and select **All**.
9. Click **Enable**.

What to do next

For clustered installation, repeat Step 4 to Step 9 for each IBM Security Identity Manager application server. After completing the steps for all servers, save the configuration changes and restart all application servers for the changes to take effect.

Disabling the Health Monitoring

Use the WebSphere Application Server administrative console to disable the health monitoring feature.

Procedure

1. Log on to the WebSphere Application Server administrative console.
2. Expand **Monitoring and Tuning**.
3. Select **Performance Monitoring Infrastructure (PMI)**.
4. Select the application server that you want to manage.
5. Clear the **Enable Performance Monitoring Infrastructure** check box.
6. Save the configuration changes.
7. Repeat this procedure for each application server.
8. Restart all application servers for the change to take effect.

IBM Security Identity Manager Monitoring Utility

The IBM Integrated Service Management Library website has a monitoring utility for IBM Security Identity Manager. You can use this utility, in conjunction with IBM Security Monitoring, to monitor IBM Security Identity Manager performance and availability.

You can download the utility from Integrated Service Management Library to your system that has IBM Security Identity Manager installed. On the website, locate the utility by searching for *IBM Security Identity Manager v6.0 Monitoring Solution*.

Note: IBM Integrated Service Library has additional information for monitoring related IBM products, such as IBM Security Access Manager and IBM Security Directory Server.

IBM DB2 Performance Monitoring

IBM DB2 provides several tools for troubleshooting and analyzing performance problems.

Enabling DB2 Monitoring

To gather performance information, turn on the DB2 monitoring flags.

About this task

Do not enable the table monitor. IBM Security Identity Manager does not need it. It has a slight performance impact when enabled.

Procedure

1. As the database administrator, connect to the database and run the following commands for each database:

```
db2 update dbm cfg using DFT_MON_STMT ON
db2 update dbm cfg using DFT_MON_BUFPOOL ON
db2 update dbm cfg using DFT_MON_LOCK ON
db2 update dbm cfg using DFT_MON_SORT ON
db2 update dbm cfg using DFT_MON_TIMESTAMP ON
db2 update dbm cfg using DFT_MON_UOW ON
```

2. Stop and restart the database instance for the monitoring to take effect.

Collecting DB2 Snapshots

Use snapshots to view the internal state of various IBM DB2 components.

Procedure

- To access specific IBM DB2 snapshots:

```
db2 get snapshot for database on database_name
db2 get snapshot for dynamic sql on database_name
db2 get snapshot for bufferpools on database_name
db2 get snapshot for tables on database_name
db2 get snapshot for locks on database_name
```

- To gather all snapshots:
  db2 get snapshot for all on *database_name*

Configuring the DB2 Statement Monitor

Use the statement monitor to examine what is occurring for each request sent to the database.

About this task

The monitor collects a large amount of information. Activate it only for a short time to gather requests.

Procedure

1. Enter the following command to create the monitor dstatement writing to /tmp/dstatements:
   db2 "create event monitor dstatement for statements
   write to file '/tmp/dstatements'"
2. If it does not exist, create the directory /tmp/dstatements:
   mkdir /tmp/dstatements
3. The first time you generate an explain plan on this database, set up the explain tables with the following command:
   db2 -tf sqllib/misc/EXPLAIN.DDL

What to do next

Monitor the database statements.

Using the DB2 statement monitor

After it is enabled, the statement monitor collects detailed information about each request sent to the database.

Before you begin

Configure the statement monitor.

About this task

When you enable the statement monitor, it records each SQL request. You can examine the query results for missing indexes, execution time, preparation time, database scans, and index scans. Activate the monitor only for a short time to gather requests, because it collects a great deal of information.

Tip: The otherTools/do_statement_monitoring.sh script in the *Tuning Guide* scripts package automates this process. You can customize the script for your system. You can use the explainSQL.sh script to have DB2 explain how the optimizer processes a particular query, including any index usage.

Procedure

1. Connect to the database, clear out any previous data, and turn on the monitor by entering the following commands:
   db2 connect to ldapdb2
   rm -f /tmp/dtatements/*
   db2 "set event monitor dstatement state 1"
2. Run the query or the action that you want to monitor.
3. Turn off the monitor with the following command:
   db2 "set event monitor dstatement state 0"
4. Convert the data so that you can read it:
   db2evmon -path /tmp/dstatements > /tmp/dstate.out

Related tasks
"Configuring the DB2 statement monitor"
Use the statement monitor to examine what is occurring for each request sent to the database.

Related information
ISIM Performance Tuning Scripts
Download performance tuning scripts for IBM Security Identity Manager.

Calculating the Buffer Pool Hit Ratio

The buffer pool hit ratio gives a good indication of how many data reads come from the buffer pool and how many from the disk. The larger the hit ratio, the less disk I/O used. Calculate the buffer pool hit ratio by enabling buffer pool monitoring and taking a database snapshot.

Use the following formula to calculate the buffer pool hit ratio:

```
P = buffer pool data physical reads + buffer pool index physical reads
L = buffer pool data logical reads + buffer pool index logical reads
Hit ratio = (1-(P/L)) * 100%
```

Related tasks
"Collecting DB2 snapshots"
Use snapshots to view the internal state of various IBM DB2 components.

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features contained in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM

product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM might have patents or pending patent applications that cover subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing

IBM Corporation

North Castle Drive

Armonk, NY 10504-1785

U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing

Legal and Intellectual Property Law

IBM Japan Ltd.

1623-14, Shimotsuruma, Yamato-shi

Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it to enable: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation

J46A/G4

555 Bailey Avenue

San Jose, CA 95141-1003

U.S.A.

Such information might be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments might vary significantly. Some measurements might have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements might have been estimated through extrapolation. Actual results might vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements, or other publicly available sources.

IBM has not tested those products and cannot confirm the accuracy of performance, compatibility, or any other claims related to non-IBM products.

Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding the future direction or intent of IBM are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing, or distributing application programs that conform to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows: © (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp.

If you are viewing this information softcopy, the photographs and color illustrations might not appear.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both: http://www.ibm.com/legal/copytrade.shtml

Microsoft, Windows and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States,

other countries, or both.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

The Oracle Outside In Technology included herein is subject to a restricted use license and can only be used in conjunction with this application.

Other company, product, and service names might be trademarks or service marks of others.