Tuning Enterprise Data Catalog Performance
Abstract

This article provides information about tuning Enterprise Data Catalog performance. Tuning Enterprise Data Catalog performance involves tuning parameters for metadata ingestion, ingestion database, search, and tuning data profiling.

The performance of Enterprise Data Catalog depends on the size of data that must be processed. The article lists the parameters that you can tune in Enterprise Data Catalog and the steps that you must perform to configure the parameters based on the data size.

The profiling tuning section includes Data on tuning data profiling in Enterprise Data Catalog. Tuning data profiling involves tuning parameters for the Data Integration Service and the profiling warehouse database properties.

Supported Versions

- Enterprise Data Catalog 10.2.1

Table of Contents

Overview ............................................................................. 3
Minimum System Requirements for a Hadoop Node in the Cluster ................................................. 4
Enterprise Data Catalog Sizing Recommendations ........................................................................ 5
  Low .............................................................................. 5
  Medium ......................................................................... 6
  High ............................................................................. 8
Tuning Performance Based on the Size of Data ............................................................................. 9
Tuning Performance for Similarity ................................................................................................. 14
Improving Search Performance Based on Data Size ................................................................. 14
Tuning Profile Warehouse ............................................................................................................ 16
Appendix A Enterprise Data Catalog Performance Tuning Parameters .................................... 16
  Enterprise Data Catalog Performance Tuning Parameters ......................................................... 16
Appendix B Tuning for Profiling Performance ............................................................................. 21
Appendix C Data Integration Service Parameters .......................................................................... 22
Appendix D Profile Configuration in Data Integration Service .................................................. 22
  Profiling Warehouse Database Properties ................................................................................. 23
  Advanced Profiling Properties .................................................................................................. 24
Appendix E Data Integration Service Profiling Properties .......................................................... 24
Appendix F Calculate Maximum Profile Execution Pool Size and Maximum Execution Pool Size (DIS) ......................... 26
Overview

Tuning Enterprise Data Catalog Performance involves tuning the performance parameters at different stages of extracting metadata, profiling, identifying similar columns, tuning the catalog, and searching for assets in the catalog.

After extracting metadata from external sources, such as databases, data warehouses, business glossaries, data integration resources, or business intelligence reports, the metadata extraction, storage, search and related operations in an organization include the following phases:

1. Storing the extracted metadata into a catalog for ease of search and retrieval. A catalog represents a centralized repository for storing metadata extracted from different sources. This phase is referred to as the metadata ingestion phase. Enterprise Data Catalog uses Apache HBase as the database for ingesting data.
2. Validate the quality of data with data profiling.
3. Search for the related data assets in the catalog. Enterprise Data Catalog uses Apache Solr to search for data assets.

The performance of Enterprise Data Catalog depends on the size of data being processed. Enterprise Data Catalog classifies data sizes into three categories based on the size of data. Based on the data size, you can configure custom properties in Informatica Administrator to assign predefined parameter values for metadata ingestion, Apache HBase database tuning, and search. Alternatively, you can also individually configure the values for the performance tuning parameters based on your requirement.

The profiling tuning involves tuning parameters for the Data Integration Service and the profiling warehouse database properties.

The following image shows the various stages at which you can tune the performance of Enterprise Data Catalog:
See the following table to identify the stages and the corresponding tuning parameters:

<table>
<thead>
<tr>
<th>Stage</th>
<th>Description</th>
<th>Parameter List</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Sources and Metadata Sources</td>
<td>Extraction of metadata from different sources.</td>
<td>Maximum # of Connections that denotes the maximum number of connections or resources.</td>
</tr>
</tbody>
</table>
| Scanner Framework                                | A framework that runs scanners and manages a registry of available scanners in Enterprise Data Catalog. A scanner is a plug-in component of Enterprise Data Catalog that extracts specific metadata from external data sources. | • Maximum # of Connections  
• Metadata extraction scanner memory parameter                                                      |
| Enterprise Data Catalog Performance Tuning:     | The process of ingesting the extracted metadata to the catalog.                                   | • Apache Spark parameters  
• Enterprise Data Catalog custom ingestion options                                                     |
| Parameters for Tuning Metadata Ingestion        |                                                                                                  |                                                                                                          |
| Enterprise Data Catalog Performance Tuning:     | The indexed archive used by Enterprise Data Catalog to store all the extracted metadata for search and retrieval. | • Apache HBase parameters  
• Apache Titan parameters  
• Apache Solr parameters                                                                 |
| Parameters for Tuning Apache HBase, Apache Titan, and Solr |                                                                                                  |                                                                                                          |
| Data Integration Service for Profiling          | An application service that performs data integration tasks for Enterprise Data Catalog and external applications. | • Profiling Warehouse Database  
• Maximum Profile Execution Pool Size  
• Maximum Execution Pool Size  
• Maximum Concurrent Columns  
• Maximum Column Heap Size  
• Maximum # of Connections |
| Profile Configuration in Data Integration Service: | The data store where the profiling information is stored before the profiling information is moved to the catalog. | • Maximum Patterns  
• Maximum Profile Execution Pool Size                                                                 |
| Profiling Warehouse Database Properties         |                                                                                                  |                                                                                                          |
| Similarity                                       | Similarity discovers similar columns in the source data within an enterprise.                     | • sparkJobCount  
• sparkExecutorCount  
• sparkExecutorCoreCount  
• sparkExecutorMemory |

**Minimum System Requirements for a Hadoop Node in the Cluster**

The following table lists the system recommendations for deploying Enterprise Data Catalog in the embedded Hadoop cluster.

The system requirements listed in the table are for a Hadoop node in the cluster. Make sure that you allocate the required additional system requirements for the operating system, processes, and other applications running on the node.
The values listed are recommendations by Informatica for improving performance. You can increase the values based on your requirements.

<table>
<thead>
<tr>
<th>System Requirement</th>
<th>Minimum Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>Eight physical cores with medium clock speed.</td>
</tr>
<tr>
<td>Memory</td>
<td>24 GB of unused memory available for use.</td>
</tr>
<tr>
<td>Hard disk drive</td>
<td>4 to 6 or more high-capacity SATA hard disk drives with 200 MB to 1 TB capacity and 7200 RPM for each disk. The hard disks must be configured as a Just a Bunch Of Disks (JBOD). Do not use any logical volume manager tools on the hard disk drives.</td>
</tr>
<tr>
<td>File system</td>
<td>ext4 file system.</td>
</tr>
<tr>
<td>Network</td>
<td>Bonded Gigabit Ethernet or 10 Gigabit Ethernet between the nodes in the cluster.</td>
</tr>
</tbody>
</table>

**Note:** If you upgraded from Enterprise Data Catalog version 10.2, follow the system requirements listed in the 10.2 version of this article available in the Informatica Knowledge Base.

**Enterprise Data Catalog Sizing Recommendations**

Based on the size of data, you must add additional memory and CPU cores to tune the performance of Enterprise Data Catalog. You must also note the minimum number of cores that are required to deploy supported data sizes.

The size of data depends on the number of assets (objects) or the number of datastores. Assets include the sum total of databases, schemas, columns, data domains, reports, mappings and so on in the deployment environment. A datastore represents a repository that stores and manages all the data generated in an enterprise. Datastores include data from applications, files, databases, and so on. Determine if you need to change the default data size for the Enterprise Data Catalog installation. Enterprise Data Catalog has low, medium, and high data sizes that you can configure in Informatica Administrator using custom properties. Data sizes are classified based on the amount of metadata to process and the number of nodes used to process metadata.

After installation, you can switch the data size from a lower data size to a higher data size. For example, if you had selected a low data size during installation, you can change the data size to medium or high after installation. However, if you had selected a higher data size value during installation, for example, high, you cannot change the data size to a lower data size, such as medium or low after installation.

**Note:** Make sure that you restart and index the Catalog Service if you switch the data size after you install Enterprise Data Catalog.

**Low**

Low represents one million assets or 30-40 datastores.

<table>
<thead>
<tr>
<th>System Requirement</th>
<th>Infrastructure</th>
<th>Profiling</th>
<th>Hadoop Cluster System Requirements for a Single Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Cores</td>
<td>8</td>
<td>16</td>
<td>8</td>
</tr>
<tr>
<td>Memory</td>
<td>10 GB</td>
<td>32 GB</td>
<td>24 GB</td>
</tr>
<tr>
<td>Storage</td>
<td>100 GB</td>
<td>20 GB</td>
<td>120 GB</td>
</tr>
</tbody>
</table>
Infrastructure

Infrastructure includes the following Informatica services:

- Informatica domain services
- Model Repository Service
- Catalog Service
- Content Management Service
- Informatica Cluster Service

Profiling

Profiling includes the Data Integration Service.

The following are the requirements for the profiling warehouse database:

- Number of cores: 4
- Memory: 16 GB
- Storage: 50 GB

Service Level Agreement

The Service Level Agreement (SLA) to process metadata extraction and profiling for a low data size is approximately one business day. The SLA calculation is for a basic RDBMS resource such as Oracle.

Concurrency

The following table lists the recommended concurrency values for low data size:

<table>
<thead>
<tr>
<th>Concurrency</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recommended resource concurrency</td>
<td>1 medium data size resource.</td>
</tr>
<tr>
<td>Maximum user concurrency to access Enterprise Data Catalog</td>
<td>20</td>
</tr>
</tbody>
</table>

Note: The complexity of transformations present in the extracted metadata affect the memory required for lineage. Ensure that you increase the memory for infrastructure as required.

Catalog Service Tuning

The following are the system requirements to tune the Catalog Service for 20 concurrent users:

<table>
<thead>
<tr>
<th>System Requirement</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Cores</td>
<td>8</td>
</tr>
<tr>
<td>Memory</td>
<td>2 GB</td>
</tr>
</tbody>
</table>

When you calculate the system requirements, make sure that you add the system requirements for infrastructure and the Hadoop cluster node, and as required for profiling and Catalog Service tuning.

Medium

Medium represents 20 million assets or 200-400 datastores.
The following table lists the number of CPU cores, memory, and storage recommended to run the services in Enterprise Data Catalog:

<table>
<thead>
<tr>
<th>System Requirement</th>
<th>Infrastructure</th>
<th>Profiling</th>
<th>Hadoop Cluster System Requirements for a Single Node</th>
<th>Hadoop Cluster System Requirements for Three Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Cores</td>
<td>8</td>
<td>32</td>
<td>8</td>
<td>24</td>
</tr>
<tr>
<td>Memory</td>
<td>10 GB</td>
<td>64 GB</td>
<td>24 GB</td>
<td>72 GB</td>
</tr>
<tr>
<td>Storage</td>
<td>100 GB</td>
<td>100 GB</td>
<td>600 GB</td>
<td>2 TB</td>
</tr>
</tbody>
</table>

**Note:** A medium data size requires a minimum of three nodes to run the Hadoop cluster.

When you calculate the system requirements, make sure that you add the system requirements for infrastructure and profiling along with the system requirements for six Hadoop cluster nodes.

**Infrastructure**

Infrastructure includes the following Informatica services:

- Informatica domain services
- Model Repository Service
- Catalog Service
- Content Management Service
- Informatica Cluster Service

**Profiling**

Profiling includes the Data Integration Service.

The following are the requirements for the profiling warehouse database:

- Number of cores: 8
- Memory: 32 GB
- Storage: 200 GB

**Service Level Agreement**

The Service Level Agreement (SLA) to process metadata extraction and profiling for a medium data size is approximately two business weeks. The SLA calculation is for a basic RDBMS resource such as Oracle.

**Concurrency**

The following table lists the recommended concurrency values for medium data size:

<table>
<thead>
<tr>
<th>Concurrency</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recommended resource concurrency</td>
<td>3 medium data size resources.</td>
</tr>
<tr>
<td>Maximum user concurrency to access Enterprise Data Catalog</td>
<td>50</td>
</tr>
</tbody>
</table>
Catalog Service Tuning

The following are the system requirements to tune the Catalog Service for 50 concurrent users:

<table>
<thead>
<tr>
<th>System Requirement</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Cores</td>
<td>18</td>
</tr>
<tr>
<td>Memory</td>
<td>4 GB</td>
</tr>
</tbody>
</table>

When you calculate the system requirements, make sure that you add the system requirements for infrastructure and three Hadoop cluster nodes, and as required for profiling and Catalog Service tuning.

**High**

High represents 50 million assets or 500-1000 datastores.

The following table lists the number of CPU cores, memory, and storage recommended to run the services in Enterprise Data Catalog:

<table>
<thead>
<tr>
<th>System Requirement</th>
<th>Infrastructure</th>
<th>Profiling</th>
<th>Hadoop Cluster System Requirements for a Single Node</th>
<th>Hadoop Cluster System Requirements for Six Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Cores</td>
<td>16</td>
<td>32</td>
<td>8</td>
<td>48</td>
</tr>
<tr>
<td>Memory</td>
<td>16 GB</td>
<td>64 GB</td>
<td>24 GB</td>
<td>144 GB</td>
</tr>
<tr>
<td>Storage</td>
<td>100 GB</td>
<td>500 GB</td>
<td>2 TB</td>
<td>12 TB</td>
</tr>
</tbody>
</table>

**Note:** A high data size requires a minimum of six nodes to run the Hadoop cluster.

**Infrastructure**

Infrastructure includes the following Informatica services:

- Informatica domain services
- Model Repository Service
- Catalog Service
- Content Management Service
- Informatica Cluster Service

**Profiling**

Profiling includes the Data Integration Service.

The following are the requirements for the profiling warehouse database:

- Number of cores: 16
- Memory: 64 GB
- Storage: 500 GB

**Service Level Agreement**

The Service Level Agreement (SLA) to process metadata extraction and profiling for a high data size is four business weeks. The SLA calculation is for a basic RDBMS resource such as Oracle.
Concurrency

The following table lists the recommended concurrency values for high data size:

<table>
<thead>
<tr>
<th>Concurrency</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recommended resource concurrency</td>
<td>5 medium data size resources.</td>
</tr>
<tr>
<td>Maximum user concurrency to access Enterprise Data Catalog</td>
<td>100</td>
</tr>
</tbody>
</table>

Catalog Service Tuning

The following are the system requirements to tune the Catalog Service for 100 concurrent users:

<table>
<thead>
<tr>
<th>System Requirement</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Cores</td>
<td>32</td>
</tr>
<tr>
<td>Memory</td>
<td>9 GB</td>
</tr>
</tbody>
</table>

When you calculate the system requirements, make sure that you add the system requirements for infrastructure and six Hadoop cluster nodes, and as required for profiling and Catalog Service tuning.

Tuning Performance Based on the Size of Data

Enterprise Data Catalog includes predefined values for the performance tuning parameters based on the size of supported data sizes. You can specify the required data size when you create the Catalog Service.

After you specify the data size, Enterprise Data Catalog uses the predefined values associated with the data size to configure the performance tuning parameters. You can also tune each parameter based on your requirements.

The option to specify the data size appears in the New Catalog Service - Step 4 of 4 dialog box as shown in the following image when you create a Catalog Service:
Click the **Load Type** drop-down list and select one of the following options to specify the required data size:

- low
- medium
- high

See the *Informatica Enterprise Data Catalog Installation and Configuration Guide* for more information about creating the Catalog Service.

### Predefined Parameter Values for Data Sizes

The following tables lists the parameters that you can use to tune the performance in Enterprise Data Catalog. The table also lists the predefined values configured in Enterprise Data Catalog for a small, medium, and large data sizes:

#### Ingestion Parameters

The set of parameters includes parameters for Apache Spark and Enterprise Data Catalog custom options. The following tables lists the ingestion parameters that you can use to tune the metadata ingestion performance of Enterprise Data Catalog:

<table>
<thead>
<tr>
<th>Apache Spark Parameters</th>
<th>Low</th>
<th>Medium</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>driver-memory</td>
<td>1G</td>
<td>2G</td>
<td>2G</td>
</tr>
<tr>
<td>executor-memory</td>
<td>10G</td>
<td>4G</td>
<td>4G</td>
</tr>
<tr>
<td>num-executors</td>
<td>2</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>executor-cores</td>
<td>5</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>
### Apache Spark Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Low</th>
<th>Medium</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>driver-java-options=-XX.MaxPermSize</td>
<td>1024M</td>
<td>1024M</td>
<td>1024M</td>
</tr>
<tr>
<td>conf.spark.executor.extraJavaOptions=-XX:PermSize</td>
<td>1024M</td>
<td>1024M</td>
<td>1024M</td>
</tr>
<tr>
<td>conf.spark.storage.memoryFraction</td>
<td>0.4</td>
<td>0.3</td>
<td>0.3</td>
</tr>
</tbody>
</table>

### Enterprise Data Catalog Ingestion Options Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Low</th>
<th>Medium</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>ingest.batch.delay.ms</td>
<td>120000</td>
<td>120000</td>
<td>120000</td>
</tr>
<tr>
<td>ingest.max.ingest.xdocs.int</td>
<td>4000</td>
<td>10000</td>
<td>20000</td>
</tr>
<tr>
<td>ingest.batch.time.ms</td>
<td>120000</td>
<td>600000</td>
<td>600000</td>
</tr>
<tr>
<td>ingest.partitions.int</td>
<td>50</td>
<td>100</td>
<td>160</td>
</tr>
<tr>
<td>ingest.propagation.partitions.int</td>
<td>96</td>
<td>96</td>
<td>96</td>
</tr>
<tr>
<td>ingest.sleep.time.ms</td>
<td>45000</td>
<td>90000</td>
<td>120000</td>
</tr>
<tr>
<td>ingest.xdoc.memory.buffer.bytes</td>
<td>104857600</td>
<td>314572800</td>
<td>524288000</td>
</tr>
<tr>
<td>titan.ids.num-partitions</td>
<td>1</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>titan.cluster.max-partitions</td>
<td>8</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td>titan.storage.hbase.region-count</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>ingest.hbase.table.regions</td>
<td>8</td>
<td>24</td>
<td>48</td>
</tr>
<tr>
<td>hclient.phoenix.query.timeoutMs</td>
<td>600000</td>
<td>600000</td>
<td>1800000</td>
</tr>
</tbody>
</table>

### Metadata Extraction Scanner Memory Parameters

Depending on the size of data for a resource, you can use one of the following parameters to configure the memory requirements for the scanner to extract metadata. The values listed in the Memory column indicate the default values configured for the scanner based on the data size:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Memory Option</th>
<th>Object Count</th>
<th>Default Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>LdmCustomOptions.scanner.memory.low</td>
<td>low</td>
<td>Up to 1,000,000 objects.</td>
<td>1024 MB</td>
</tr>
<tr>
<td>LdmCustomOptions.scanner.memory.medium</td>
<td>medium</td>
<td>Up to 1 million objects.</td>
<td>4096 MB</td>
</tr>
<tr>
<td>LdmCustomOptions.scanner.memory.high</td>
<td>high</td>
<td>Up to 10 million objects.</td>
<td>12288 MB</td>
</tr>
</tbody>
</table>

You can set the resource memory in the Advanced Properties section on the Metadata Load Settings tab page for the resource, as shown in the following sample image:
Note:

- You must increase the Informatica PowerCenter scanner memory based on the complexity of the mappings.
- You must select the number of concurrent scanners based on the memory type and the available resources in the cluster. See the Scanner column in the Low, Medium, or High tables for more information about memory type.

Apache HBase Parameters

The set of parameters includes parameters for HBase site configuration, HBase region server, HBase master properties, and the HBase slider app master properties. The following tables list the parameters that you can use to tune the performance of the HBase database:

HBase Site Configuration Parameters

<table>
<thead>
<tr>
<th>HBase Site Configuration Parameters</th>
<th>Low</th>
<th>Medium</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>hbase_master_heapsize</td>
<td>2048M</td>
<td>3096M</td>
<td>5120M</td>
</tr>
<tr>
<td>hbase_regionserver_heapsize</td>
<td>10240M</td>
<td>5120M</td>
<td>5120M</td>
</tr>
<tr>
<td>hbase.master.handler.count</td>
<td>100</td>
<td>300</td>
<td>600</td>
</tr>
<tr>
<td>hbase.regionserver.handler.count</td>
<td>250</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>hbase.hstore.blockingStoreFiles</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>hbase.hregion.majorcompaction</td>
<td>0</td>
<td>86400000</td>
<td>86400000</td>
</tr>
</tbody>
</table>
HBase Region Server Parameters

<table>
<thead>
<tr>
<th>HBase Region Server Parameters</th>
<th>Low</th>
<th>Medium</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>yarn.component.instances</td>
<td>1</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>yarn.memory</td>
<td>10240</td>
<td>6656</td>
<td>6656</td>
</tr>
<tr>
<td>yarn.vcores</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Parameters for Tuning the HBase Master Properties

<table>
<thead>
<tr>
<th>HBase Master Properties Parameters</th>
<th>Low</th>
<th>Medium</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>yarn.component.instances</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>yarn.memory</td>
<td>2048</td>
<td>3096</td>
<td>5120</td>
</tr>
<tr>
<td>yarn.vcores</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Parameters for Tuning the HBase Slider App Master Properties

<table>
<thead>
<tr>
<th>HBase Slider App Master Properties Parameters</th>
<th>Low</th>
<th>Medium</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>yarn.component.instances</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>yarn.memory</td>
<td>512</td>
<td>512</td>
<td>512</td>
</tr>
<tr>
<td>yarn.vcores</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>jvm.heapsize</td>
<td>256M</td>
<td>256M</td>
<td>256M</td>
</tr>
</tbody>
</table>

Apache Solr Parameters

The set of parameters includes parameters for Solr slider app master properties and the Enterprise Data Catalog custom options Solr node properties. The following tables list the parameters that you can use to tune the performance of Apache Solr for search operations:

Parameters for Tuning the Solr Slider App Master Properties

<table>
<thead>
<tr>
<th>Solr Slider App Master Properties Parameters</th>
<th>Low</th>
<th>Medium</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>yarn.component.instances</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>yarn.memory</td>
<td>512</td>
<td>512</td>
<td>512</td>
</tr>
<tr>
<td>yarn.vcores</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>jvm.heapsize</td>
<td>256M</td>
<td>256M</td>
<td>256M</td>
</tr>
</tbody>
</table>
Parameters for Tuning the Enterprise Data Catalog Custom Options Solr Node Properties

<table>
<thead>
<tr>
<th>Custom Options for Solr Node Properties Parameters</th>
<th>Low</th>
<th>Medium</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>xmx_val</td>
<td>5g</td>
<td>10g</td>
<td>20g</td>
</tr>
<tr>
<td>xms_val</td>
<td>5g</td>
<td>10g</td>
<td>20g</td>
</tr>
<tr>
<td>yarn.component.instances</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>yarn.memory</td>
<td>5120</td>
<td>10240</td>
<td>20480</td>
</tr>
<tr>
<td>yarn.vcores</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>solr.replicationFactor</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>solr.numShards</td>
<td>1</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>

### Tuning Performance for Similarity

The following table lists the default values for parameters associated with column similarity, based on the size of data:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Low</th>
<th>Medium</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>sparkJobCount</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>sparkExecutorCount</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>sparkExecutorCoreCount</td>
<td>3</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>sparkExecutorMemory (in GB)</td>
<td>2</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

Note the following points before you tune the parameters based on your requirements:

- Similarity performance scales linearly when you increase the `sparkExecutorCount` and the `sparkExecutorCoreCount` parameters.
- Scale-up performance during similarity profile run depends on the values you configure for the total parallel mappings including native profiling and similarity profiling.
- Each similarity job utilizes 22 cores for Medium and High data sizes. Multiply sparkJobCount by 22 to calculate the actual CPU cores required.

### Improving Search Performance Based on Data Size

The following section lists the recommended values for the configuration parameters based on the data size for Enterprise Data Catalog Apache Solr custom options.

The following are the Enterprise Data Catalog Apache Solr custom options and the parameters:

- `LdmCustomOptions.solrdeployment.solr.options`
  - `MaxDirectMemorySize`. Maximum total size of java.nio (New I/O package) direct buffer allocations.
  - `UseLargePages`. Use this option to optimize processor translation lookaside buffers.
- `Dsolr.hdfs.blockcache.slab.count`. Number of memory slabs to allocate.
- `Dsolrconfig.updatehandler.autocommit.maxdocs`. Maximum number of Solr documents stored in memory before performing an auto commit. For example, the default configuration `LdmCustomOptions.solrdeployment.solr.options=-Dsolrconfig.updatehandler.autocommit.maxdocs=100000` specifies that Enterprise Data Catalog performs an auto commit every time after storing 100000 Solr documents in memory.

**Note:** Default is 100000 for all data sizes.

- `LdmCustomOptions.SolrNodeProperties`
  - `yarn.component.instances`
  - `yarn.memory`
  - `yarn.vcores`
  - `xmx_val`
  - `xms_val`

**Solr Deployment Options Parameter Default Values**

**Low.** `-XX:MaxDirectMemorySize=3g -XX:+UseLargePages -Dsolr.hdfs.blockcache.slab.count=24 -Dsolrconfig.updatehandler.autocommit.maxdocs=100000`

**Medium.** `-XX:MaxDirectMemorySize=5g -XX:+UseLargePages -Dsolr.hdfs.blockcache.slab.count=38 -Dsolrconfig.updatehandler.autocommit.maxdocs=100000`

**High.** `-XX:MaxDirectMemorySize=10g -XX:+UseLargePages -Dsolr.hdfs.blockcache.slab.count=76 -Dsolrconfig.updatehandler.autocommit.maxdocs=100000`

**Solr Node Properties Parameter Default Values**

**Low.** `yarn.component.instances=1,yarn.memory=3072,yarn.vcores=1,xmx_val=3g,xms_val=3g`

**Medium.** `yarn.component.instances=1,yarn.memory=5120,yarn.vcores=1,xmx_val=5g,xms_val=5g`

**High.** `yarn.component.instances=1,yarn.memory=10240,yarn.vcores=1,xmx_val=10g,xms_val=10g`

**Approximate Size of Index Files**

**Low.** 800 MB

**Medium.** 16 GB

**High.** 40 GB

**Note:**

- The configuration shown for the low data size caches the whole index.
- The configuration shown for the medium data size assumes that you are running HDFS and Apache Solr on a host with 32 GB of unused memory.
Tuning Profile Warehouse

The profiling warehouse stores profiling results. More than one Profiling Service Module may point to the same profiling warehouse. The main resource for the profiling warehouse is disk space.

See the following table to identify the parameters that you need to tune to improve the performance of the profile warehouse based on the size of data:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Low</th>
<th>Medium</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tablespace</td>
<td>The maximum amount of disk space required to temporarily store the profiling results before these results get stored in the Enterprise Data Catalog.</td>
<td>50 GB</td>
<td>200 GB</td>
<td>500 GB</td>
</tr>
<tr>
<td>CPU</td>
<td>The number of cores required by the profiling warehouse</td>
<td>4</td>
<td>8</td>
<td>16</td>
</tr>
</tbody>
</table>

Note: See the Informatica Data Explorer Sizing Guidelines How To Library or the Appendixes in this article for more information about profiling warehouse guidelines.

Appendix A Enterprise Data Catalog Performance Tuning Parameters

You can configure the following types of parameters to tune the performance of Enterprise Data Catalog:

- Metadata ingestion parameters.
- Apache HBase database tuning parameters.
- Apache Solr parameters.

Tuning the parameters based on the size of the data helps to improve the performance of Enterprise Data Catalog. Data sizes are classified based on the amount of metadata that Enterprise Data Catalog processes and the number of nodes in the Hadoop cluster. You can calculate the size of data based on the total number of objects in data, such as tables, views, columns, schemas, and business intelligence resources.

Enterprise Data Catalog Performance Tuning Parameters

You can configure the following types of parameters to tune the performance of Enterprise Data Catalog:

- Metadata ingestion parameters.
- Apache HBase database tuning parameters.
- Apache Solr parameters.

Tuning the parameters based on the size of the data set helps to improve the performance of Enterprise Data Catalog. Data sets are classified based on the amount of metadata that Enterprise Data Catalog processes and the number of nodes in the Hadoop cluster. You can calculate the size of a data set based on the total number of objects in the data set, such as tables, views, columns, schemas, and business intelligence resources.
Parameters for Tuning Metadata Ingestion

The following table lists the metadata ingestion parameters that you can configure in Enterprise Data Catalog to tune the ingestion performance:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>driver-memory*</td>
<td>Memory required by the spark driver.</td>
</tr>
<tr>
<td>executor-memory*</td>
<td>Memory required by spark executors.</td>
</tr>
<tr>
<td>num-executors</td>
<td>Number of executors spawned by spark to process the metadata.</td>
</tr>
<tr>
<td>executor-core 1</td>
<td>Number of CPU cores allocated for each executor.</td>
</tr>
<tr>
<td>driver-java-options=-XX:MaxPermSize</td>
<td>Perm gen size for the spark driver.</td>
</tr>
<tr>
<td>conf spark.executor.extraJavaOptions=-XX:PermSize</td>
<td>Perm gen size for the spark executor.</td>
</tr>
<tr>
<td>conf spark.storage.memoryFraction</td>
<td>Fraction of java heap space used for spark-specific operations such as aggregation.</td>
</tr>
<tr>
<td>ingest.batch.delay.ms</td>
<td>Represents clock skew across nodes of the cluster. Default is two minutes and the value must be set higher than the skew.</td>
</tr>
<tr>
<td>ingest.max.ingest.xdocs.int</td>
<td>This parameter is deprecated. Maximum number of documents to be processed in one batch.</td>
</tr>
<tr>
<td>ingest.max.ingest.facts.int</td>
<td>Maximum amount of facts about an object that can be processed in single batch of ingestion.</td>
</tr>
<tr>
<td>ingest.batch.time.ms</td>
<td>All the documents remaining from the previous batch processed and the present batch. These documents get processed with the next batch. This value is restricted to the batch size specified earlier.</td>
</tr>
<tr>
<td>ingest.partitions.int</td>
<td>Number of partitions to be used for ingestion.</td>
</tr>
<tr>
<td>ingest.propagation.partitions.int</td>
<td>Number of partitions to be used for propagation.</td>
</tr>
<tr>
<td>ingest.sleep.time.ms</td>
<td>Interval between different ingestions.</td>
</tr>
<tr>
<td>ingest.xdoc.memory.buffer.bytes</td>
<td>Total amount of memory that can be used by XDOCS.</td>
</tr>
<tr>
<td>Note: If you increase the value of this parameter, make sure that you increase the value for the metadata extraction scanner memory parameter. Failing to increase the scanner memory parameter might result in out of memory errors.</td>
<td></td>
</tr>
<tr>
<td>titan.ids.num-partitions</td>
<td>Used by Titan to generate random partitions of the ID space and helps avoiding region-server hotspot. This value must be equal to the number of region servers.</td>
</tr>
<tr>
<td>titan.cluster.max-partitions</td>
<td>Determines the maximum number of virtual partitions that Titan creates. This value must be provided in multiples of two.</td>
</tr>
<tr>
<td>titan.storage.hbase.region-count</td>
<td>Number of HBase regions to be used for the TITAN table.</td>
</tr>
</tbody>
</table>
To enable Spark dynamic allocation, set the

 Restart the node managers.

 Add or modify the following parameters in the

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HbaseTableSplits</td>
<td>Controls the number of pre-splits in the Hbase table.</td>
</tr>
<tr>
<td>ingest.phoenix.query.timeoutMs</td>
<td>Amount of time within which the HBase operation must complete before failing.</td>
</tr>
</tbody>
</table>

* When you increase the value for this parameter, it is recommended that you increase the maximum memory allocation in YARN for a container. Failing to increase the memory allocation might result in YARN shutting down the applications. It is recommended that when you increase the memory configuration of any component, for example, ingestion, you must keep a buffer of 30% of the actual memory required for the component. For example, if a component requires 100 MB of memory, you must increase the memory configuration to 130 MB for that component.

1 For external clusters, when you increase the value for this parameter, it is recommended that you increase the maximum number of cores in YARN, for a container.

**Enabling Spark Dynamic Resource Allocation**

Perform the following steps to configure an existing cluster to enable dynamic resource allocation for Spark applications:

1. To enable Spark dynamic allocation, set the LdmCustomOptions.ingest.spark.dynamic.allocation.enable custom property in Catalog Service to true.

2. Add or modify the following parameters in the yarn-site.xml file present on every host:
   - yarn.nodemanager.aux-services.spark_shuffle
   - yarn.nodemanager.aux-services.spark_shuffle.class
     org.apache.spark.network.yarn.YarnShuffleService

3. Copy spark-<version>-yarn-shuffle.jar to the /usr/hdp/current/hadoop-yarn-nodemanager/lib/ directory of every host where node managers are running.
   **Note:** You can get the spark-<version>-yarn-shuffle.jar file from the /lib directory located inside the Spark binary tar file or you can download the .jar file. Make sure that the version of the .jar file is the same as the Spark version.

4. Restart the node managers.

**Note:** The steps listed are not applicable if you have deployed Enterprise Data Catalog on an embedded cluster.

**Parameters for Tuning Apache HBase**

The Apache HBase tuning parameters include parameters for the Apache HBase site configuration properties, the HBase region server properties, the HBase master properties, and the HBase Slider Master App properties. The following tables list the parameters for tuning different properties of Apache HBase:

**HBase Site Configuration Properties**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hbase_master_heapsize*</td>
<td>Heap size for HBase master.</td>
</tr>
<tr>
<td>hbase_regionserver_heapsize*</td>
<td>Heap space for region server.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>hbase.master.handler.count</td>
<td>Total number of RPC instances on HBase master to serve client requests.</td>
</tr>
<tr>
<td>hbase.regionserver.handler.count</td>
<td>Handler count for the region server.</td>
</tr>
<tr>
<td>hbase.hstore.blockingStoreFiles</td>
<td>Number of store files used by HBase before the flush is blocked.</td>
</tr>
<tr>
<td>hbase.hregion.majorcompaction</td>
<td>Time between major compactions of all the store files in a region. Setting this parameter to 0 disables time-based compaction.</td>
</tr>
</tbody>
</table>

* When you increase the value for this parameter, it is recommended that you increase the maximum memory allocation in YARN, for a container. Failing to increase the memory allocation might result in YARN shutting down the applications. It is recommended that when you increase the memory configuration of any component, for example, ingestion, you must keep a buffer of 30% of the actual memory required for the component. For example, if a component requires 100 MB of memory, you must increase the memory configuration to 130 MB for that component.

**HBase Region Server Properties**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>yarn.component.instances²</td>
<td>Number of instances of each component for slider. This parameter specifies the number of region servers that are run.</td>
</tr>
<tr>
<td>yarn.memory</td>
<td>Amount of memory allocated for the container hosting the region server.</td>
</tr>
<tr>
<td>yarn.vcores¹</td>
<td>Number of cores allocated for the region server.</td>
</tr>
</tbody>
</table>

1 For external clusters, when you increase the value for this parameter, it is recommended that you increase the maximum number of cores in YARN, for a container.

2 Before increasing this parameter, you must add the required number of nodes to the cluster.

**HBase Master Properties**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>yarn.component.instances²</td>
<td>Number of instances of each component for slider. This parameter specifies the number of master servers that are run.</td>
</tr>
<tr>
<td>yarn.memory</td>
<td>Amount of memory allocated for the container hosting the master server.</td>
</tr>
<tr>
<td>yarn.vcores¹</td>
<td>Number of cores allocated for the master server.</td>
</tr>
</tbody>
</table>

1 For external clusters, when you increase the value for this parameter, it is recommended that you increase the maximum number of cores in YARN, for a container.

2 Before increasing this parameter, you must add the required number of nodes to the cluster.
HBase Slider App Master Properties

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>jvm.heapsize</td>
<td>Amount of memory allocated for the container hosting the master server.</td>
</tr>
<tr>
<td>yarn.component.instances²</td>
<td>Number of instances of each component for slider. This parameter specifies the number of master servers that are run.</td>
</tr>
<tr>
<td>yarn.memory</td>
<td>Number of cores allocated for the master server.</td>
</tr>
<tr>
<td>yarn.vcores¹</td>
<td>Memory used by the slider app master.</td>
</tr>
</tbody>
</table>

¹ For external clusters, when you increase the value for this parameter, it is recommended that you increase the maximum number of cores in YARN, for a container.

² Before increasing this parameter, you must add the required number of nodes to the cluster.

Parameters for Tuning Apache Solr

These parameters include the Apache Solr Slider app master properties and the Enterprise Data Catalog custom options for the Apache Solr node. The following tables list the Apache Solr parameters that you can use to tune the performance of search in Enterprise Data Catalog:

Apache Solr Slider App Master Properties

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>jvm.heapsize</td>
<td>Memory used by the slider app master.</td>
</tr>
<tr>
<td>yarn.component.instances²</td>
<td>Number of instances of each component for slider. This parameter specifies the number of master servers that are run.</td>
</tr>
<tr>
<td>yarn.memory</td>
<td>Amount of memory allocated for the container hosting the master server.</td>
</tr>
<tr>
<td>yarn.vcores¹</td>
<td>Number of cores allocated for the master server.</td>
</tr>
</tbody>
</table>

¹ For external clusters, when you increase the value for this parameter, it is recommended that you increase the maximum number of cores in YARN, for a container.

² Before increasing this parameter, you must add the required number of nodes to the cluster.

Enterprise Data Catalog Custom Options Apache Solr Node Properties

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>xmx_val*</td>
<td>Solr maximum heap</td>
</tr>
<tr>
<td>xms_val</td>
<td>Solr minimum heap</td>
</tr>
<tr>
<td>yarn.component.instances²</td>
<td>Number of instances of each component for slider. This parameter specifies the number of master servers that are run.</td>
</tr>
<tr>
<td>yarn.memory</td>
<td>Amount of memory allocated for the container hosting the master server.</td>
</tr>
</tbody>
</table>
Parameter | Description
--- | ---
yarn.vcores$^1$ | Number of cores allocated for the master server.
solr.replicationFactor | Number of Solr index replications.
solr.numShards | Number of Solr shards.

* When you increase the value for this parameter, it is recommended that you increase the maximum memory allocation in YARN for a container. Failing to increase the memory allocation might result in YARN shutting down the applications. It is recommended that when you increase the memory configuration of any component, for example, ingestion, you must keep a buffer of 30% of the actual memory required for the component. For example, if a component requires 100 MB of memory, you must increase the memory configuration to 130 MB for that component.

$^1$ For external clusters, when you increase the value for this parameter, it is recommended that you increase the maximum number of cores in YARN for a container.

$^2$ Before increasing this parameter, you must add the required number of nodes to the cluster.

**Appendix B Tuning for Profiling Performance**

Tuning profiling performance involves configuring the data integration service parameters, the profile database warehouse properties, and the advanced profiling properties.

See the following table to identify the parameters that you can tune to improve profiling performance:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profiling Warehouse Database</td>
<td>Connection name to the profiling warehouse database. In addition to the profile results, the profiling warehouse holds the persisted profile job queue. Verify that no profile job runs when you change the connection name. Otherwise, the profile jobs might stop running because the profile jobs run on the Data Integration Service where the Profiling Service Module submitted the profile jobs. You set the default value when you create the instance.</td>
</tr>
<tr>
<td>Maximum Profile Execution Pool Size</td>
<td>The number of profile mappings that the Profiling Service Module can run concurrently when the Data Integration Service runs on a single node or on a grid. The pool size is dependent on the aggregate processing capacity of the Data Integration Service, which you specify for each node on the Processes tab of the Administrator tool. The pool size cannot be greater than the sum of the processing capacity of all nodes. When you plan for a deployment, consider the threads used for profile tasks. It is important to understand the mixture of mappings and profile jobs so that you can configure the Maximum Execution Pool Size parameter. Default is 10.</td>
</tr>
<tr>
<td>Maximum Execution Pool Size</td>
<td>The maximum number of requests that the Data Integration Service can run concurrently. Requests include data previews, mappings, and profiling jobs. This parameter has an impact on the Data Integration Service.</td>
</tr>
</tbody>
</table>
### Maximum Concurrent Columns

The number of columns that a mapping runs in parallel. The default value of 5 is optimal for most of the profiling use cases. You can increase the default value for columns with cardinality lower than the average value. Decrease the default value for columns with cardinality higher than the average value. You might also want to decrease this value is when you consistently run profiles on large source files where temporary disk space is low.

Default is 5.

### Maximum Column Heap Size

The cache size for each column profile mapping for flat files. You can increase this value to prevent the Data Integration Service from writing some parts of the intermediate profile results to temporary disk. However, this effect does not apply to large data sources. The default setting is optimal for most of the profiling use cases. In Enterprise Data Catalog, profiling is done with a lower data volume, for example, in a batch of 10000 rows. To avoid creating multiple mappings and to prevent an impact on compilation performance, you can combine multiple columns in a single mapping. It is recommended to set the value for the Maximum Column Heap Size to 512 to avoid temporary disk usage with combined mapping.

Default is 64.

### Appendix C Data Integration Service Parameters

You need to configure the Temporary Directories and Maximum Execution Pool Size parameters for the Data Integration Service. You can configure parameters, such as Reserved Profile Threads that apply to the Profiling Service Module. Before you use the parameter recommendations, verify that you have identified a node or grid and the requirement is to configure the node or grid optimally to run profiles.

The following configuration parameters have an impact on different components of the profiling and discovery installation. You can increase or decrease the values for these parameters based on the performance requirements:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Execution Pool Size</td>
<td>The maximum number of requests that the Data Integration Service can run concurrently. Requests include data previews, mappings, and profiling jobs. This parameter has an impact on the Data Integration Service.</td>
</tr>
<tr>
<td>Maximum Profile Execution Pool Size</td>
<td>The total number threads to run profiles. This parameter has an impact on the Profiling Service Module.</td>
</tr>
<tr>
<td>Temporary Directories</td>
<td>Location of temporary directories for the Data Integration Service process on the node. This parameter has an impact on the Data Integration Service machine.</td>
</tr>
</tbody>
</table>

### Appendix D Profile Configuration in Data Integration Service

The Data integration Service has a set of parameters specific to the Profiling Service Module that controls the performance of a profiling job. You must configure the parameters for each deployment.

You can configure the following types of parameters:
Profiling warehouse database properties

Constitute the majority of the parameters and apply to the Profiling Service Module across the deployment. For example, the Maximum Profile Execution Pool Size parameter controls the total number of mappings that the Profiling Service Module generates. The profiling warehouse database properties apply to all the nodes that constitute the Profiling Service Module.

Advanced profiling properties

Apply to a single Data Integration Service node. You must configure the advanced profiling parameters for each node in the Data Integration Service if you use a grid.

You can use Informatica Administrator to configure both profiling warehouse database and advanced profiling parameters.

You can use the following sections in the Administrator tool to configure these parameters for the Data Integration Service:

- Profiling warehouse database properties on the Properties tab.
- Advanced profiling properties on the Properties tab.
- Execution options on the Processes tab.

Profiling Warehouse Database Properties

The profiling warehouse database properties apply to the Profiling Service Module across the deployment.

You can set the following parameters:

**Maximum Patterns**

The maximum number of patterns that each column stores. Sometimes, it is important to store as many patterns as possible. You can set the Maximum Patterns parameter to a large number, such as 10,000, and adjust the Pattern Threshold Percentage parameter to .01. Setting a high value for this parameter has negligible impact on performance.

Default is 10.

**Maximum Profile Execution Pool Size**

The number of profile mappings that the Profiling Service Module can run concurrently when the Data Integration Service runs on a single node or on a grid. The pool size is dependent on the aggregate processing capacity of the Data Integration Service, which you specify for each node on the Processes tab of the Administrator tool. The pool size cannot be greater than the sum of the processing capacity of all nodes.

When you plan for a deployment, consider the threads used for profile tasks.

It is important to understand the mixture of mappings and profile jobs so that you can configure the Maximum Execution Pool Size parameter. For optimal performance, verify that the total number of threads in the three categories adds up to the aggregate total of the Maximum Execution Pool Size parameter. The Maximum Execution Pool Size parameter is located under the Properties tab of the Administrator tool. The Maximum Profile Execution Pool Size parameter value is applicable for a single Data Integration Service node. You can calculate the value for the Maximum Execution Pool Size parameter by adding the values of the Maximum Profile Execution Pool Size parameter for all the nodes.

Default is 10.

**Note:**

- If the Data Integration Service runs on a single node, it is recommended to keep the Maximum Execution Pool Size parameter value as the same as the total number of threads in the Data Integration Service.
If the Data Integration Service runs on a grid, it is recommended to keep the Maximum Execution Pool Size parameter value as the same as the total number of processes in the Data Integration Service.

**Advanced Profiling Properties**

The advanced profiling properties apply to a single Data Integration Service node. You must configure the parameters for each node in the Data Integration Service.

You can configure the following advanced profiling properties:

**Pattern Threshold Percent**

The minimum percentage of rows matching up to two decimal places for a pattern to appear in the results.

Default is 5.00.

**Maximum String Length**

The maximum length of a string that the Profiling Service Module mappings process internally. The default is set to the maximum value of 255. If you decrease the value, the Data Integration Service truncates the value. Decreased string lengths can have a minor impact on the amount of tablespace required for the profiling warehouse and negligible impact on the overall performance.

Default is 255.

**Maximum Numeric Precision**

The maximum precision, which is the number of significant digits in the number, for numeric decimal datatypes. If you set a low value for this parameter, the Data Integration Service might process additional numeric datatypes as strings instead of numbers.

Default is 38.

### Appendix E Data Integration Service Profiling Properties

The following table shows you how to tune the Data Integration Service concurrency parameters for profiling:

<table>
<thead>
<tr>
<th>Concurrency Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Execution Pool Size (Up to 10.1.1 HF1)</td>
<td>Number of cores x 0.6</td>
<td>Maximum parallel mappings executed by the Data Integration Service that includes all types of mappings.</td>
</tr>
<tr>
<td>Maximum On-Demand Execution Pool Size</td>
<td>Number of cores x 0.6</td>
<td>Maximum parallel mappings executed by the Data Integration Service that includes all types of mappings.</td>
</tr>
<tr>
<td>Maximum Profile Execution Pool Size</td>
<td>Number of cores x 0.6</td>
<td>Maximum parallel profiling mappings executed by the Data Integration Service</td>
</tr>
<tr>
<td>Maximum Concurrent Profile Jobs</td>
<td>Number of cores x 0.6</td>
<td>Maximum parallel tables or files processed.</td>
</tr>
</tbody>
</table>
### Concurrency Parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Concurrent Profile Threads</td>
<td>Number of cores x 0.6</td>
<td>Maximum parallel mappings executed by the Data Integration Service for File sources.</td>
</tr>
<tr>
<td>AdvancedProfilingServiceOptions.FileConnectionLimit (For CSV, XML, JSON metadata sources)</td>
<td>Number of cores x 0.6</td>
<td>Maximum parallel profiling mappings executed by the Data Integration Service for File sources.</td>
</tr>
<tr>
<td>AdvancedProfilingServiceOptions.HiveConnectionLimit (For Hive metadata sources)</td>
<td>Number of cores x 0.6</td>
<td>Maximum parallel profiling mappings executed by the Data Integration Service for Hive sources.</td>
</tr>
<tr>
<td>Maximum # of Connections</td>
<td>Number of cores x 0.6</td>
<td>Throttle number of database connections for each resource.</td>
</tr>
</tbody>
</table>

**Note:** The Value column indicates the value of the parameter if the number of rows is less than or greater than 100,000 rows with sampling of rows enabled or disabled.

As shown in the table, to calculate the value for concurrency parameters, multiply the number of cores with 0.6 and round off the value to the nearest whole value.

For example, if the number of cores configured for the Data Integration Service is 16, then the concurrency parameter value is calculated as follows: $16 \times 0.6 = 9.6$. The concurrency parameter value is rounded off to the nearest whole value, namely, 10.

The following table lists the values for the Data Integration Service profiling tuning parameters:

<table>
<thead>
<tr>
<th>Tuning Parameter</th>
<th>Value for 100,000 or lesser Rows with Row Sampling Enabled</th>
<th>Value for 100,000 or more Rows with Row Sampling Disabled</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Concurrent Columns</td>
<td>100</td>
<td>5</td>
<td>Maximum number of columns merged in single mapping.</td>
</tr>
<tr>
<td>Maximum Column Heap Size</td>
<td>512</td>
<td>512</td>
<td>Maximum cache size allocated for a mapping.</td>
</tr>
<tr>
<td>AdvancedProfilingServiceOptions.ColumnsThreshold</td>
<td>100</td>
<td>Default</td>
<td>Maximum number of columns merged for a mapping. Applicable only for column profiling.</td>
</tr>
<tr>
<td>AdvancedProfilingServiceOptions.DomainMappingColumnThreshold</td>
<td>100</td>
<td>50</td>
<td>Maximum number of columns merged for a mapping. Applicable only for data domain discovery.</td>
</tr>
<tr>
<td>AdvancedProfilingServiceOptions.MaxDataDomains</td>
<td>150</td>
<td>Default</td>
<td>Maximum number of data domains combined for a data domain discovery mapping.</td>
</tr>
</tbody>
</table>
As shown in the table, to calculate the value for concurrency parameters, multiply the number of cores with 0.6 and round off the value to the nearest whole value.

For example, if the number of cores configured for the Data Integration Service is 16, then the concurrency parameter value is calculated as follows: 16 * 0.6 = 9.6. The concurrency parameter value is rounded off to the nearest whole value, namely, 10.

### Appendix F Calculate Maximum Profile Execution Pool Size and Maximum Execution Pool Size (DIS)

The Maximum Profile Execution Pool Size parameter determines the total number DTM threads that the Data Integration Service uses to run profiles. You can calculate the number of DTM threads allocated to the Profiling Service Module based on the expected mix of profile jobs and the number of CPU cores. The Maximum Execution Pool Size parameter determines the maximum number of requests that the Data Integration Service can run concurrently.

To calculate the expected mix of profile jobs, divide the mix of profile jobs into two groups before you calculate the number of DTM threads. The first group can include the profile jobs that the Data Integration Service cannot transfer to the relational source. The second group can include the profile jobs that the Data Integration Service can transfer to the relational source. You can use the first group proportion to compute the number of threads for the jobs that the system cannot transfer to the relational sources. You can then add this number to the Maximum DB Connections value to compute the final estimate for the number of DTM threads. Each profiling type uses a different number of CPU cores. Use different weights for different profile job types.

In the following worksheet, enter the values in the A and B columns as required. Multiply the values in the A and B columns for each row and then update the A x B column.
Use the following worksheet to record the values for profile operations that the Data Integration Service cannot transfer to the relational sources:

<table>
<thead>
<tr>
<th>Profile Operation</th>
<th>Estimated Number for Each 100 Runs (A)</th>
<th>Factor (B)</th>
<th>A x B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column profile on a relational source</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data domain discovery</td>
<td></td>
<td>0.5</td>
<td></td>
</tr>
</tbody>
</table>

**Calculation**

Add all the values in the A x B column and divide the total value by 100. You can then multiply the result with the number of CPU cores for the nodes in the Data Integration Service machine and the recommended load factor of 2.5. The final value is the number of DTM threads for profile jobs that the Data Integration Service cannot transfer to the relational source.

**Final Calculation**

To calculate the Maximum Profile Execution Pool Size value and the Maximum Execution Pool Size value, add the following values:

- The number of DTM threads for profile jobs that the Data Integration Service cannot transfer to the relational source
- The Reserved Profiling Threads parameter value
- The Maximum # of Connections that denotes the maximum number of connections or resources. This parameter is located in the Connections > Pooling tab on the Informatica Administrator interface.

**Appendix G Tuning Profiling Mapping for Unstructured Data Sources**

Large unstructured files in PDF format require more memory during profiling mapping. You can tune the ExecuteContextOptions.JVMMaxMemory Data Integration Service parameter in the Custom Properties section to tune memory based on the file size.

The following table lists the memory that you can configure for the ExecuteContextOptions.JVMMaxMemory parameter based on the size of the PDF file:

<table>
<thead>
<tr>
<th>PDF File Size</th>
<th>Memory (in MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Less than or equal to 25 MB</td>
<td>1024M</td>
</tr>
<tr>
<td>Less than or equal to 50 MB</td>
<td>2048M</td>
</tr>
<tr>
<td>Less than or equal to 100 MB</td>
<td>5120M</td>
</tr>
</tbody>
</table>

**Authors**

Suraj Jayan
Lead Technical Writer
Acknowledgements

The author would like to acknowledge Gaurav Pathak, Prem Krishna, Adarsh Praharaj, Chandrasekaran Radhakrishnan, Sumeet Sharma, Zameer Malim, Siddiq Hussain, and Eshan Rastogi for their contributions to this article.