Optimizing Transformation Caches in Informatica 10.0
Abstract

The Data Integration Service allocates cache memory to process Aggregator, Joiner, Lookup, Rank, and Sorter transformations in a mapping. For optimal mapping performance, configure the cache sizes so that the Data Integration Service can process the complete transformation in memory. This article describes how the default auto cache mode works, how to analyze transformation statistics in the mapping log to determine the optimal cache sizes, and how to modify the cache sizes in version 10.0.

Supported Versions

- Data Services 10.0
- Data Quality 10.0

Table of Contents

Overview ................................................. 2
Cache Size ............................................. 3
  Auto Cache Size ............................... 3
  Specific Cache Size ......................... 4
Cache Size Increase by the Data Integration Service .............................................. 4
Cache Size for Partitioned Caches ................................................................. 5
Cache Size Optimization ................................................. 5
  Step 1. Set the Tracing Level to Verbose Initialization ........................................... 5
  Step 2. Run the Mapping in Auto Cache Mode .................................................... 6
  Step 3. Analyze Caching Performance ............................................................. 6
  Step 4. Configure Specific Cache Sizes .......................................................... 6

Overview

When you run a mapping that uses an Aggregator, Joiner, Lookup, Rank, or Sorter transformation, the Data Integration Service creates caches in memory to process the transformation. You can configure the cache sizes for these transformations. The cache size determines how much memory the Data Integration Service allocates for each transformation cache at the start of a mapping run.

If the cache size is larger than the available memory on the machine, the Data Integration Service cannot allocate enough memory and the mapping run fails.

If the cache size is smaller than the amount of memory required to run the transformation, the Data Integration Service processes some of the transformation in memory and stores overflow data in cache files. When the Data Integration Service pages cache files to the disk, processing time increases. For optimal performance, configure the cache size so that the Data Integration Service can process the complete transformation in memory.

By default, the Data Integration Service automatically calculates the memory requirements at run time, based on the maximum amount of memory that the service can allocate. After you run a mapping in auto cache mode, you can tune the cache sizes for the transformations. You analyze the transformation statistics in the mapping log to determine the cache sizes required for optimal performance, and then configure specific cache sizes for the transformations.
Cache Size

Cache size determines how much memory the Data Integration Service allocates for each transformation cache at the start of a mapping run. You can configure a transformation cache size to use auto cache mode or to use a specific value.

Auto Cache Size

By default, a transformation cache size is set to Auto. The Data Integration Service automatically calculates the cache memory requirements at run time. You define the maximum amount of memory that the service can allocate.

The Data Integration Service uses the following guidelines to automatically allocate the memory:

Allocates more memory to transformations with higher processing times.

The Data Integration Service allocates more memory to transformations that typically have higher processing times. For example, the Data Integration Service allocates more memory to the Sorter transformation because the Sorter transformation typically takes longer to run.

Allocates more memory to the data cache than to the index cache.

Aggregator, Joiner, Lookup, and Rank transformations require an index cache and a data cache. When the Data Integration Service divides the memory allocated for the transformation across the index and data cache, it allocates more memory to the data cache.

Sorter transformations require a single cache. The service allocates all of the memory allocated for the transformation to the Sorter cache.

Maximum Memory for Auto Cache Size

You define the maximum amount of memory that the Data Integration Service can allocate to transformation caches in the Maximum Memory Per Request property for Data Integration Service modules in the Administrator tool.

Each module runs different types of requests that have different memory requirements. For example, mapping and profile requests typically require more cache memory than SQL service or web service requests. You can configure the Maximum Memory Per Request property for the following Data Integration Service modules:

- Mapping Service Module
- Profiling Service Module
- SQL Service Module
- Web Service Module

Note: Mapping Service Module requests include mappings and mappings run from Mapping tasks within a workflow.

For the Profiling Service Module, Maximum Memory Per Request defines the maximum amount of memory that the Data Integration Service can allocate for each mapping run for a single profile request.

For the remaining modules, the behavior of Maximum Memory Per Request depends on the Data Integration Service configuration. The behavior depends on the Launch Job Options property and the Maximum Memory Size property on the Data Integration Service.
The following table describes the behavior of Maximum Memory Per Request for the mapping, SQL service, and web service modules based on the Data Integration Service configuration:

<table>
<thead>
<tr>
<th>Data Integration Service Configuration</th>
<th>Maximum Memory Per Request Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Runs jobs in separate local or remote system processes, or Maximum Memory Size is 0 (default)</td>
<td>Maximum amount of memory, in bytes, that the Data Integration Service can allocate for all transformations that use auto cache mode in a single request. The value that you define for Maximum Memory Per Request affects only transformations that use auto cache mode. The Data Integration Service allocates memory separately to transformations for which you configure a specific cache size. The total memory used by the request can exceed the value of Maximum Memory Per Request. For example, Maximum Memory Per Request is set to 800 MB. A mapping has three transformations that require caching. You configure two transformations to use auto cache mode and configure the third transformation to use a total of 500 MB for the cache sizes. The Data Integration Service allocates a total of 1,300 MB of memory for all of the transformation caches.</td>
</tr>
<tr>
<td>Runs jobs in the Data Integration Service process, and Maximum Memory Size is greater than 0</td>
<td>Maximum amount of memory, in bytes, that the Data Integration Service can allocate for a single request. The value that you define for the Maximum Memory Per Request property affects all transformations. The total memory used by the request cannot exceed the value of Maximum Memory Per Request.</td>
</tr>
</tbody>
</table>

When you increase the maximum amount of memory used for auto cache mode, you increase the maximum cache size that can be used for all requests to the module. You can increase the maximum amount of memory to ensure that no cache files are paged to the disk. However, because this value is used for all requests, the Data Integration Service might allocate more memory than is needed for some requests.

**Specific Cache Size**

You can configure a specific cache size for a transformation. The Data Integration Service allocates the specified amount of memory to the transformation cache at the start of the mapping run. Configure a specific value in bytes when you tune the cache size.

The first time that you configure a cache size, use auto cache mode. After you run the mapping, analyze transformation statistics in the mapping log to determine the cache sizes required to run the transformations in memory. When you configure the cache size to use the value specified in the mapping log, you can ensure that no allocated memory is wasted. However, the optimal cache size varies based on the size of the source data. Review the mapping logs after subsequent mapping runs to monitor changes to the cache size. If you configure a specific cache size for a reusable transformation, verify that the cache size is optimal for each use of the transformation in a mapping.

To define specific cache sizes, configure the cache size values in the transformation properties in the Developer tool.

**Cache Size Increase by the Data Integration Service**

The Data Integration Service creates each memory cache based on the configured cache size. In some situations, the Data Integration Service might increase the configured cache size because it requires more cache memory.

The Data Integration Service might increase the configured cache size for one of the following reasons:

**Configured cache size is less than the minimum cache size required to process the operation.**

The Data Integration Service requires a minimum amount of memory to initialize each mapping. If the configured cache size is less than the minimum required cache size, then the Data Integration Service
increases the configured cache size to meet the minimum requirement. If the Data Integration Service cannot allocate the minimum required memory, the mapping fails.

**Configured cache size is not a multiple of the cache page size.**

The Data Integration Service stores cached data in cache pages. The cached pages must fit evenly into the cache. For example, if you configure 10 MB (1,048,576 bytes) for the cache size and the cache page size is 10,000 bytes, then the Data Integration Service increases the configured cache size to 1,050,000 bytes to make it a multiple of the 10,000-byte page size.

When the Data Integration Service increases the configured cache size, it continues to run the mapping and writes the following messages in the mapping log:

```
INFO: MAPPING, TE_7212, Increasing [Index Cache] size for transformation <transformation name> from <configured cache size> to <new cache size>.  
INFO: MAPPING, TE_7212, Increasing [Data Cache] size for transformation <transformation name> from <configured cache size> to <new cache size>.  
```

**Cache Size for Partitioned Caches**

If your license includes partitioning, the Data Integration Service uses cache partitioning for partitioned Aggregator, Joiner, Rank, Lookup, and Sorter transformations. Cache partitioning creates a separate cache for each partition that runs the transformation. During cache partitioning, each partition stores different data in a separate cache. When the Data Integration Service uses cache partitioning for these transformations, the service divides the allocated cache size across the partitions.

For example, you configure the transformation cache size to be 100 MB. The Data Integration Service uses four partitions to run the transformation. The service divides the cache size value so that each partition uses a maximum of 25 MB for the cache size.

**Cache Size Optimization**

For optimal mapping performance, configure the cache sizes so that the Data Integration Service can run the complete transformation in memory.

To configure optimal cache sizes, perform the following tasks:

1. Set the tracing level to verbose initialization.
2. Run the mapping in auto cache mode.
3. Analyze caching performance in the mapping log.
4. Configure specific values for the cache sizes.

**Step 1. Set the Tracing Level to Verbose Initialization**

In the Developer tool, set the tracing level to verbose initialization to enable the Data Integration Service to write transformation statistics to the mapping log. The transformation statistics list the cache sizes required for optimal performance. By default, the tracing level is set to normal.

Set the tracing level to verbose initialization in one of the following ways:

- Modify the advanced properties for each transformation that uses a cache.
- Modify the default mapping configuration properties if you plan to run the mapping for the first time from the Developer tool. For more information, see the Informatica Developer Tool Guide.
- Modify the advanced properties for an application that contains the mapping if you plan to run the deployed mapping for the first time from the command line. For more information, see the Informatica Developer Tool Guide.
### Step 2. Run the Mapping in Auto Cache Mode

The first time that you run the mapping, use auto cache mode for the transformation cache sizes.

You can run the mapping from the Developer tool. Or, you can add the mapping to an application and then deploy the application to the Data Integration Service so that you can run the mapping from the command line.

### Step 3. Analyze Caching Performance

After you run the mapping in auto cache mode, analyze the transformation statistics in the mapping log to determine the cache sizes required for optimal mapping performance.

When an Aggregator, Joiner, Lookup, or Rank transformation pages to the disk, the mapping log specifies the index and data cache sizes required to run the transformation in memory. For example, you run an Aggregator transformation called AGG_TRANS. The mapping log contains the following text:

```plaintext
CMN_1791, The index cache size that would hold [1098] aggregate groups of input rows for [AGG_TRANS], in memory, is [286720] bytes
CMN_1790, The data cache size that would hold [1098] aggregate groups of input rows for [AGG_TRANS], in memory, is [1774368] bytes
```

The log shows that the index cache requires 286,720 bytes and the data cache requires 1,774,368 bytes to run the transformation in memory without paging to the disk.

When a Sorter transformation pages to the disk, the mapping log states that the Data Integration Service made multiple passes on the source data. The Data Integration Service makes multiple passes on the data when it has to page to the disk to complete the sort. The message specifies the number of bytes required for a single pass, which is when the Data Integration Service reads the data once and performs the sort in memory without paging to the disk.

For example, you run a Sorter transformation called SRT_TRANS. The mapping log contains the following text:

```plaintext
SORT_40427, Sorter Transformation [SRT_TRANS] required 2-pass sort (1-pass temp I/O: 1312621824 bytes). You may try to set the cache size to 14128 MB or higher for 1-pass in-memory sort.
```

The log shows that the Sorter cache requires 14,128 MB so that the Data Integration Service makes one pass on the data.

### Step 4. Configure Specific Cache Sizes

For optimal performance, configure the transformation cache sizes to use the values specified in the mapping log. Update the index and data cache size transformation properties in the Developer tool.

1. In the Developer tool, open the reusable or non-reusable transformation.
2. Locate the cache size properties depending on the following transformation types:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reusable Aggregator, Joiner, Rank, or Sorter transformation</td>
<td>Click the Advanced view.</td>
</tr>
<tr>
<td>Non-reusable Aggregator, Joiner, Rank, or Sorter transformation</td>
<td>Click the Advanced tab in the Properties view.</td>
</tr>
<tr>
<td>Reusable Lookup transformation</td>
<td>Click the Run-time view.</td>
</tr>
<tr>
<td>Non-reusable Lookup transformation</td>
<td>Click the Run-time tab in the Properties view.</td>
</tr>
</tbody>
</table>

3. Enter the values in bytes that the mapping log recommended for the index and data cache sizes.
The following image shows a non-resuable Aggregator transformation that has specific values configured for the index and data cache sizes:

![Image of Aggregator transformation properties](image)

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache Directory</td>
<td>CacheDir (Parameter)</td>
</tr>
<tr>
<td>Aggregator Data Cache Size</td>
<td>1.6922 megabytes (1774388 bytes)</td>
</tr>
<tr>
<td>Aggregator Index Cache Size</td>
<td>280.0000 kilobytes (286720 bytes)</td>
</tr>
<tr>
<td>Sorted Input</td>
<td></td>
</tr>
<tr>
<td>Tracing Level</td>
<td>Normal</td>
</tr>
</tbody>
</table>

4. Click File > Save.

**Author**

Alison Taylor  
Principal Technical Writer

**Acknowledgements**

The author would like to acknowledge Guy Boo, Avinash Kodikal, Mohammed Morshed, Valentin Moskovich, and Yongqin Xiao for their contributions to this article.