Increasing Performance for PowerCenter Sessions that Use Partitions

© 1993-2015 Informatica LLC. No part of this document may be reproduced or transmitted in any form, by any means (electronic, photocopying, recording or otherwise) without prior consent of Informatica LLC. All other company and product names may be trade names or trademarks of their respective owners and/or copyrighted materials of such owners.
Abstract

A partition is a pipeline stage that executes in a single reader, transformation, or writer thread. You can configure multiple partitions for a single pipeline stage. This article explains how to increase performance in sessions that use partitions.

Supported Versions

- PowerCenter 9.x - 10.0

Table of Contents

Pipeline Partition Tuning Overview .................................................. 2
Pipeline Partitioning ....................................................................... 3
Data Volume Tuning ........................................................................ 3
Adding Partitions to a Pipeline .......................................................... 3
Number of Partitions Tuning .............................................................. 4
Deleting Partitions from a Pipeline .................................................... 4
Memory Allocation Tuning ................................................................. 4
Configuring the Cache Size for a Transformation ................................ 5
Dynamic Partitioning ........................................................................ 5
Configuring Dynamic Partitioning ..................................................... 6

Pipeline Partition Tuning Overview

When running a session, the PowerCenter Integration Service can achieve higher performance by partitioning the pipeline. Configure pipeline partitioning to perform the extract, transformation, and load for each partition in parallel.

A pipeline consists of a Source Qualifier transformation and all the transformations and targets that receive data from that Source Qualifier transformation. A partition is a pipeline stage that executes in a single reader, transformation, or writer thread.

If the source of a session contains a large volume of data, you can add partitions to each pipeline stage to increase data throughput and session performance.

If you increase the number of partitions and session performance decreases, you may have to perform one of the following tasks:

- Tune the number of partitions. Each additional partition can use a separate CPU core. Configure the number of partitions based on the number of available CPUs.
- Tune the memory allocation for transformations in the mapping. Increase system memory and optimize memory allocation for transformations to ensure that the PowerCenter Integration Service processes all transformations in memory.
- Enable dynamic partitioning. When you enable dynamic partitioning, the PowerCenter Integration Service dynamically determines the number of partitions to create at run time.
**Pipeline Partitioning**

Partitions distribute data to threads within a pipeline. The number of partitions in any pipeline stage equals the number of threads in that stage.

Partition points mark the thread boundaries in a pipeline and divide the pipeline into stages. When you create a session, the Workflow Manager creates one partition point at each transformation in the pipeline. At each partition point in a pipeline, the default number of partitions is one. However, you can define multiple partitions at a partition point in order to control the data transformation operations within the pipeline.

When you define multiple partitions, the session processes groups of data in parallel threads. For example, setting the number of partitions at a partition point to three divides the data into three segments, and performs the extract, transformation, and load for each partition in parallel. Partitioning the pipeline can increase session performance.

You can define up to 64 partitions at any partition point in a pipeline. When you increase or decrease the number of partitions at any partition point, the Workflow Manager increases or decreases the number of partitions at all partition points in the pipeline. The number of partitions remains consistent throughout the pipeline. If you define three partitions at any partition point, the Workflow Manager creates three partitions at all other partition points in the pipeline. In certain circumstances, the number of partitions in the pipeline must be set to one.

The PowerCenter Integration Service runs the partition threads concurrently. When you run a session with multiple partitions, the threads run as follows:

1. The reader threads run concurrently to extract data from the source.
2. The transformation threads run concurrently in each transformation stage to process data. The PowerCenter Integration Service redistributes data among the partitions at each partition point.
3. The writer threads run concurrently to write data to the target.

**Data Volume Tuning**

Multiple partitions can improve session performance by dividing data into multiple threads and then processing the threads in parallel. Processing data in multiple threads decreases data volume and increases data throughput.

For example, a simple mapping with a single partition and a source, transformation, and target processes 100 million rows of data. Due to an input/output data bottleneck, the session performance is poor. By increasing the number of partitions in the pipeline, you can increase performance by processing smaller amounts of data concurrently.

When you add \( n \) partitions to a pipeline, you decrease the session run time by a factor of \( n \). For example, a session with one partition processes one thread containing 100 million rows of data in 30 minutes. If you change the number of partitions to four, the session processes four threads, each with 25 million rows of data, in seven to eight minutes.

When you add partitions, performance increases as long as the hardware has sufficient resources to process the data.

When you increase the number of partitions in a session, you may still experience decreased performance if each partition reads data from the same source location. To gain optimal performance from a partitioned session, distribute the data across multiple database partitions or disks. If you are using key-range partitioning, the source database must also be partitioned. Otherwise, adding partitions to the pipeline will not improve session performance.

**Adding Partitions to a Pipeline**

Add partitions to a pipeline from the **Mappings** tab of the session properties.

1. To add partitions to a transformation that is not already a partition point, select the transformation and click the **Add a Partition Point** button on the **Partitions** view of the **Mapping** tab. Select the partition type for the partition point or accept the default value.

   **Tip:** You can select a transformation from the Non-Partition Points node.
2. To add partitions to an existing partition point, select the partition point in the Partitions Point node. Click **Edit Partition > Add**.

3. Click **OK**.

   The transformation appears in the Partition Points node in the **Partitions** view on the **Mapping** tab of the session properties.

---

### Number of Partitions Tuning

The number of CPUs in a session with multiple partitions must be high enough to process the required number of threads and amount of data. If you increase the number of partitions and session performance decreases, you may need to delete some of the partitions.

Increasing the number of partitions or partition points increases the number of threads. Therefore, increasing the number of partitions or partition points also increases the load on the node. If the node contains enough CPU bandwidth, processing rows of data in a session concurrently can increase session performance. However, if you create a large number of partitions or partition points in a session that processes a large amount of data, you can overload the system.

Check the number of CPUs available in a system before running a session with multiple partitions. A simple session that runs in two partitions typically requires twice the amount of CPU than when the session runs in a single partition. If the number of available CPUs is too low to achieve the desired session performance, you may need to adjust the session parameters. Reduce the number of partitions in a session when the number of CPUs is insufficient.

---

### Deleting Partitions from a Pipeline

Remove partition points from a pipeline from the **Mappings** tab of the session properties. Each partition point must have at least one partition.

1. To delete partitions from an existing partition point, select the partition point in the Partitions Point node. Click **Edit Partition > Delete**.

2. Select the partition that you want to delete.

---

### Memory Allocation Tuning

When you run a session, the PowerCenter Integration Service allocates cache memory for XML targets and Aggregator, Joiner, Lookup, Rank, and Sorter transformations in a mapping. Increasing the size of the cache for transformations can improve session performance by increasing the amount of data that can be processed in the cache memory.

If the session contains multiple partitions, the PowerCenter Integration Service may use cache partitioning for the Aggregator, Joiner, Lookup, Rank, and Sorter transformations. When the PowerCenter Integration Service partitions a cache, it creates a separate cache for each partition and allocates the configured cache size to each partition. Each cache contains only the rows needed by one partition. As a result, the PowerCenter Integration Service requires a portion of total cache memory for each partition. To configure the memory requirements for a transformation with cache partitioning, calculate the total requirements for the transformation and divide by the number of partitions. When the PowerCenter Integration Service uses cache partitioning, it accesses the cache in parallel for each partition.

If the session does not require cache partitioning, the PowerCenter Integration Service creates one memory cache for all partitions. The PowerCenter Integration Service accesses the cache serially for each partition.

When you create a session, you can configure the cache size for each transformation instance in the session properties. If the PowerCenter Integration Service requires more memory than what you configure, it stores overflow values in cache files.
Writing data to cache files can slow session performance. For optimal performance, configure the session to process the transformation in memory without writing overflow data to cache files. Set the cache size to the total amount of memory required for the transformation. The cache size must be large enough to process all the data in the transformation thread.

The amount of memory that you can allocate to the cache depends on the amount of available memory in the system. If your system only has 8 GB of available memory, but you allocate 10 GB of cache memory, the session will not run.

You can perform additional memory allocation tuning after you run a session. View the transformation statistics in the session log to see how much cache memory was used in each transformation. You can then reconfigure the memory allocations for each transformation in the session to match the necessary requirements.

**Configuring the Cache Size for a Transformation**

Configure the cache size for a transformation in the session properties. When you configure the cache size, you specify the total memory requirements for the transformation.

1. In the Workflow Manager, open the session.
2. Click the **Mapping** tab.
3. Select the mapping object in the left pane.
   The right pane of the Mapping tab shows the object properties where you can configure the cache size.
4. Select one of the following options:
   - Enter a value for the cache size. Default units are bytes. However, you can specify the units by entering the units after the numeric value. For example: 1 GB, 20 KB.
   - Select the Auto mode to limit the amount of cache memory allocated to the transformation. You can limit memory by providing a maximum amount, or a percentage of the total. The PowerCenter Integration Service will calculate the required memory at the start of the session.
   - Select the Calculate mode to calculate the total memory requirement for the transformation. Provide input based on the transformation type.
5. Click **OK**.

**Dynamic Partitioning**

When you use dynamic partitioning, you can configure the partition information so the PowerCenter Integration Service determines the number of partitions to create at run time. Dynamic partitioning can optimize session performance by adjusting the number of partitions at the start of a session.

Dynamic partitioning will create the optimal number of partitions for a session based on the number of CPUs, the volume of data, the number of nodes in a grid, or the number of source database partitions. This can improve session performance if the system configuration changes prior to the start of a session.

However, the session may not achieve maximum performance if the memory allocation is not large enough. When using dynamic partitioning, be sure to tune the cache size for each transformation before running a session.

If any transformation in a stage does not support partitioning, or if the partition configuration does not support dynamic partitioning, the PowerCenter Integration Service does not scale partitions in the pipeline. The data passes through one partition.

Do not configure dynamic partitioning for a session that contains manual partitions. If you set dynamic partitioning to a value other than disabled and you manually partition the session, the session is not valid.
Configuring Dynamic Partitioning

Configure dynamic partitioning on the Config Object tab of session properties.

You can use one of the following methods to configure dynamic partitioning:

- **Based on number of partitions.** Sets the partitions to a number that you define in the Number of Partitions attribute. Use the $DynamicPartitionCount session parameter, or enter a number greater than 1.

- **Based on number of nodes in grid.** Sets the partitions to the number of nodes in the grid running the session. If you configure this option for sessions that do not run on a grid, the session runs in one partition and logs a message in the session log.

- **Based on source partitioning.** Determines the number of partitions using database partition information. The number of partitions is the maximum of the number of partitions at the source. For Oracle sources that use composite partitioning, the number of partitions is the maximum of the number of subpartitions at the source.

- **Based on number of CPUs.** Sets the number of partitions equal to the number of CPUs on the node that prepares the session. If the session is configured to run on a grid, dynamic partitioning sets the number of partitions equal to the number of CPUs on the node that prepares the session multiplied by the number of nodes in the grid.

Author

Informatica Documentation Team