

CLEARPEAKS BI LAB

INFORMATICA PERFORMANCE OPTIMIZATION TECHNIQUES

July, 2016

Author: Syed

CLEARPEAKS



TABLE OF CONTENTS

INFORMATICA PERFORMANCE OPTIMIZATION TECHNIQUES	3
STEP 1: IDENTIFYING BOTTLENECKS	3
STEP 2: RESOLVING DIFFERENT PERFORMANCE BOTTLENECKS	7
1. Optimizing Sources and Targets	7
2. Optimizing Buffer Block Size	8
3. Push Down Optimization	10
4. Informatica Partitioning	12
5. Dynamic Session Partitioning	16
6. Concurrent Workflow Execution	18
7. Load Balancing for Workload Distribution on the Grid	20
8. Conclusion	21
ABOUT CLEARPEAKS	21

INFORMATICA PERFORMANCE OPTIMIZATION TECHNIQUES

The purpose of the article is to walk you through the steps to identify performance bottlenecks in Informatica and different optimization techniques to fix them. After reading through the article you will have a working understanding of the performance breakpoints for consideration while designing the mapping or fine tuning the existing ones.

Below are the steps to follow in the tuning process:

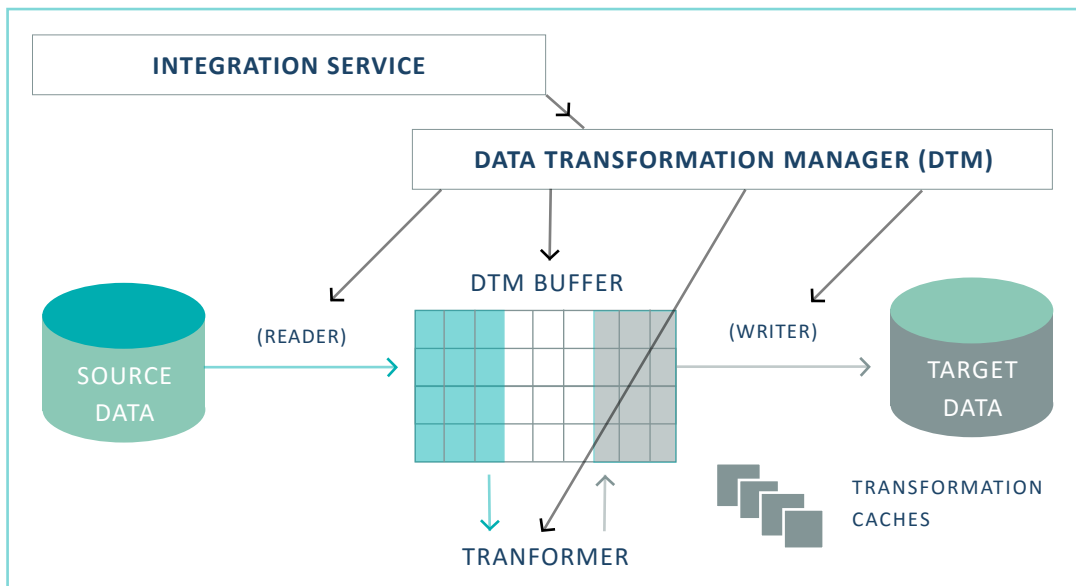
STEP 1: IDENTIFYING BOTTLENECKS

The first step in performance tuning is to identify performance bottlenecks. Performance bottlenecks can occur in the source and target, the mapping, the session, and the system. Determining the best way to improve performance can be complex. An iterative method of identifying one bottleneck at a time and eliminating it, then identifying and eliminating the next bottleneck until an acceptable throughput is achieved is more effective.

When a PowerCenter session is triggered, Integration Service starts Data Transformation Manager (DTM), which is responsible for starting the reader thread, transformation thread and writer thread.

Shown below is the pictorial representation of a session. The reader thread reads data from the source and data transformation is done by the transformation thread, and then finally loaded into the target by the writer thread.

FIG 1 ANATOMY OF A SESSION



Thread Statistics in the session log give run time information from all three threads: reader, transformation and writer. The session log provides enough run time thread statistics to help us to understand and pinpoint the performance bottleneck.

- **Run Time:** Amount of time the thread runs.
- **Idle Time:** Amount of time the thread is idle. Includes the time the thread waits for other thread processing.
- **Busy Time:** Percentage of the run time. It is $(\text{run time} - \text{idle time}) / \text{run time} \times 100$.
- **Thread Work Time:** The percentage of time taken to process each transformation in a thread.

If you look at the snippet of a session log below closely, you will see the reader, transformation and writer threads, how much time is spent on each thread, and how busy each thread is.

```
***** RUN INFO FOR TGT LOAD ORDER GROUP [1], CONCURRENT SET [1] *****
Thread [READER_1_1_1] created for [the read stage] of partition point [SQ_W_WORKORDER_F] has completed.
  Total Run Time = [75.317835] secs
  Total Idle Time = [0.312221] secs
  Busy Percentage = [99.585462]
Thread [TRANSF_1_1_1] created for [the transformation stage] of partition point [SQ_W_WORKORDER_F] has completed.
  Total Run Time = [1.337162] secs
  Total Idle Time = [0.000000] secs
  Busy Percentage = [100.000000]
Transformation-specific statistics for this thread were not accurate enough to report.
Thread [READER_1_2_1] created for [the read stage] of partition point [SQ_W_EAM_PCRT_D] has completed.
  Total Run Time = [7.152990] secs
  Total Idle Time = [0.000000] secs
  Busy Percentage = [100.000000]
Thread [TRANSF_1_2_1] created for [the transformation stage] of partition point [SQ_W_EAM_PCRT_D] has completed. The total run time was insufficient for any meaningful statistics.
Thread [READER_1_3_1] created for [the read stage] of partition point [SQ_W_EAM_HSECES_SITE_G] has completed. The total run time was insufficient for any meaningful statistics.
Thread [TRANSF_1_3_1] created for [the transformation stage] of partition point [SQ_W_EAM_HSECES_SITE_G] has completed. The total run time was insufficient for any meaningful statistics.
Thread [WRITER_1_*_1] created for [the write stage] of partition point [W_EAM_HSECES_F] has completed.
  Total Run Time = [68.503361] secs
  Total Idle Time = [66.720620] secs
  Busy Percentage = [2.602414]
```

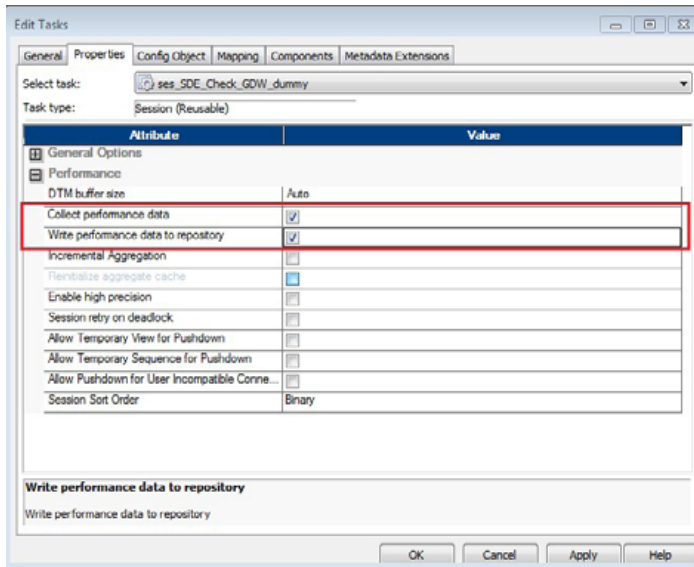
From the simple example given above, the reader threads READER_1_1_1 and READER_1_2_1 have a busy percentage of 99.5 % and 100% respectively. Even though the busy percentage of the transformation thread also shows 100%, consider the run time of 1.3 seconds, which is much less compared to the reader threads.

In this case we can clearly see that reader threads READER_1_1_1 and READER_1_2_1 are bottlenecks, and if we tune them then the idle time of the writer thread will automatically become zero.

- **Hint:** The thread with the highest busy percentage is the bottleneck

All transformations have counters to help measure and improve their performance. Analyzing these performance details can help you to identify session bottlenecks. The Integration Service tracks the number of input rows, output rows, and error rows for each transformation.

You can set up the session to gather performance counters in the Workflow Manager. The image below shows the required configuration for a session to collect transformation performance counters.



The image below shows the performance counters for a session, which you can see from the Workflow Monitor session run properties. You can see the transformations in the mapping and the corresponding performance counters.

Performance Counter	Counter Value
AGGTRANS	p150034
Aggregator_inputrows	150000
Aggregator_newgroupkey	150000
Aggregator_outputrows	150000
Aggregator_readfromcache	867192
Aggregator_readfromdisk	61945
Aggregator_writeloc	229689
Aggregator_writelodisk	70424
EXPTRANS	p150634
Expression_inputrows	150000
Expression_outputrows	150000
EXPTRANS1	p150634
Expression_inputrows	150000
Expression_outputrows	150000
FILTRANS	p150634
Filter_inputrows	150000
Filter_outputrows	150
SEQTRANS	p150634
Sequence_outputrows	150000
UPDTRANS	p150634
UpdateStrategy_inputrows	150000
UpdateStrategy_outputrows	150000

While running the PowerCenter Session, Integration Service tracks the number of input rows, output rows, and error rows for each transformation; some transformations also have performance counters.

To increase session performance, you can use the following performance counters:

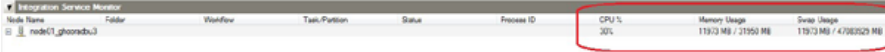
Readfromcache and Writetocache Counters & Readfromdisk and Writetodisk Counters:

If a session contains Aggregator, Rank, or Joiner transformations, examine the Transformation_readfromcache and Transformation_writetocache counters along with the Transformation_readfromdisk and Transformation_writetodisk counters to analyze how the Integration Service reads from or writes to disk. If these counters display any number other than zero, you can increase the cache sizes to improve session performance.

To view the performance details file:

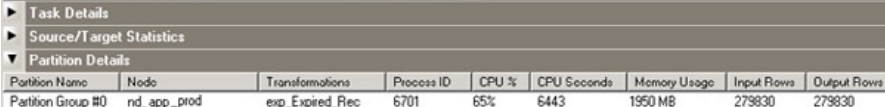
1. Locate the performance details file. The Informatica Server names the file `session_name.perf`, and stores it in the same directory as the session log.
2. Open the file in any text editor.

You can view the Integration Service properties in the Workflow Monitor to see CPU, memory, and swap system usage when you are running task processes on the Integration Service. Use the following Integration Service properties to identify performance issues:



The screenshot shows the Integration Service Monitor interface. A table lists task processes with columns for Node Name, Folder, Workflow, Task-Partition, State, Process ID, CPU %, Memory Usage, and Swap Usage. The CPU % column for the task 'ndc(t1_ghozndu3)' is highlighted with a red box, showing a value of 30%. The Memory Usage is 11912 MB / 21950 MB, and the Swap Usage is 11912 MB / 4706329 MB.

Node Name	Folder	Workflow	Task-Partition	State	Process ID	CPU %	Memory Usage	Swap Usage
ndc(t1_ghozndu3)						30%	11912 MB / 21950 MB	11912 MB / 4706329 MB



The screenshot shows the Task Details, Source/Target Statistics, and Partition Details sections of the Integration Service Monitor. The Partition Details table is expanded, showing the following data:

Partition Name	Node	Transformations	Process ID	CPU %	CPU Seconds	Memory Usage	Input Rows	Output Rows
Partition Group #0	nd_app_prod	exp_Expired_Rec	6701	65%	6443	1950 MB	279830	279830

CPU%:

The percentage of CPU usage includes other external tasks running on the system. A high CPU usage indicates the need for additional processing power required by the server.

Memory Usage:

The percentage of memory usage includes other external tasks running on the system. If the memory usage is close to 95%, check if the tasks running on the system are using the amount indicated in the Workflow Monitor, or if there is a memory leak. To troubleshoot, use system tools to check the memory usage before and after running the session and then compare the results to the memory usage while running the session.

Swap Usage:

Swap usage is a result of paging due to possible memory leaks or a high number of concurrent tasks.

STEP 2: RESOLVE DIFFERENT PERFORMANCE BOTTLENECKS

① **Optimizing Sources and Targets**

To achieve good performance, we first need to ensure that there are no bottlenecks at source or target; to ensure that we are utilizing the database to an optimum level, consider the following points:

Load Balancing:

Perform reading/writing/sorting/grouping/filtering data in the database. Use Informatica for the more complex logic, outside joins, data integration, multiple source feeds, etc... The balancing act is difficult without DBA knowledge. In order to reach a balance, you must be able to recognize which operations are best in the database, and which are best in Informatica. This does not degrade the use of the ETL tool, but enhances it- it's a must if you are performance tuning for high-volume throughput.

Estimating size of data:

Don't be afraid to estimate: small, medium, large, and extra-large source data set sizes (in terms of: numbers of rows, average number of bytes per row), expected throughput for each, and turnaround time for load. Give this information to your DBAs and ask them to tune the database for "worst case".

Help them to assess which tables are expected to be high read/high write, which operations will sort, order, etc... Moving disks or assigning the right table to the right disk space could make all the difference. Utilize a PERL script to generate "fake" data for small, medium, large, and extra-large data sets. Run each of these through your mappings- in this way, the DBA can watch or monitor throughput as a real load size occurs.

Optimizing query:

If a session joins multiple source tables in one Source Qualifier, you might be able to improve performance by optimizing the query with optimizing hints. The database optimizer usually determines the most efficient way to process the source data. However, you might know properties of the source tables that the database optimizer does not. The database administrator can create optimizer hints to tell the database how to execute the query for a particular set of source tables.

Using Bulk Loads:

Use Bulk Load session property that inserts a large amount of data into a DB2, Sybase ASE, Oracle, or Microsoft SQL Server database, bypassing the database log which speeds up performance.

Using External Loaders :

To increase session performance, configure PowerCenter to use an external loader utility. External loaders can be used for Oracle, DB2, Sybase and Teradata.

Dropping Indexes and Key Constraints:

When you define key constraints or indexes in target tables, you slow down the loading of data to those tables. To improve performance, drop indexes and key constraints before you run the session. You can rebuild those indexes and key constraints after the session completes in the Post SQL of the session.

Increasing Database Network Packet Size:

If you read from Oracle, Sybase ASE, or Microsoft SQL Server sources, you can improve performance by increasing the network packet size. Increase the network packet size to allow larger packets of data to cross the network at the same time.

Localization:

Localize stored procedures, functions, views and sequences in the source database and avoid synonyms. This could potentially affect the performance as much as 3 times or more. If you are reading from a flat file, make sure the file is copied to Informatica Server before reading data from the file.

Avoiding DB Level Sequences:

Avoid any database based sequence generators; if you absolutely must, have a shared sequence generator, then build a staging table from the flat file, add a SEQUENCE ID column, and call a POST TARGET LOAD stored procedure to populate that column. Place the post target load procedure into the flat file to staging table load mapping. A single call to inside the database, followed by a batch operation to assign sequences, is the fastest method for utilizing shared sequence generators.

Minimizing Deadlocks:

Encountering deadlocks can slow session performance. You can increase the number of target connection groups in a session to avoid deadlocks. To use a different target connection group for each target in a session, use a different database connection name for each target instance.

Increasing Database Checkpoint Intervals:

The Integration Service performance slows down each time it waits for the database to perform a checkpoint. To decrease the number of checkpoints and increase performance, increase the checkpoint interval in the database.

② Optimizing Buffer Block Size

When the Integration Service initializes a session, it allocates blocks of memory to hold source and target data. Sessions that use a large number of sources and targets might require additional memory blocks.

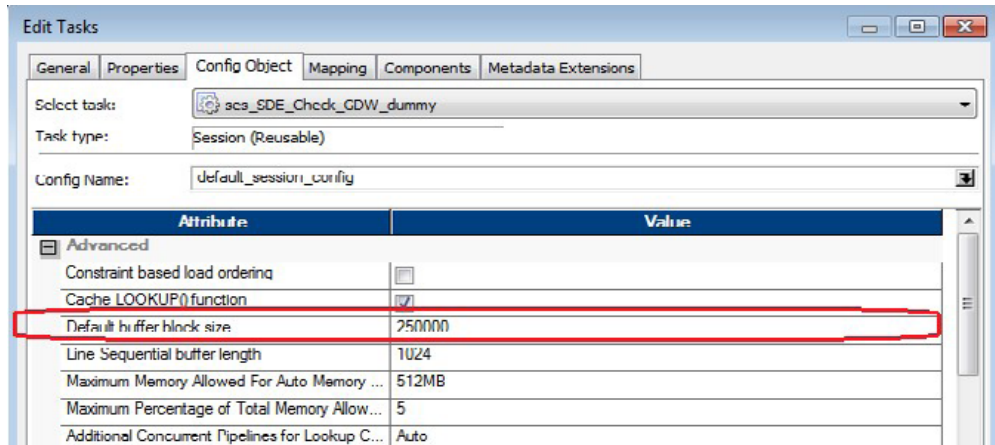
Not having enough buffer memory for DTM processes can slow down reading, transforming or writing, and cause large fluctuations in performance; adding extra memory blocks can keep the threads busy and improve session performance. You can do this by adjusting the buffer block size and DTM Buffer size.

DTM Buffer is the temporary storage area for data, divided into blocks. Buffer size and block size are tunable session properties; the default value for both of these session parameters is Auto.

To identify the optimal buffer block size, sum up the precision of individual source and targets columns. The largest precision among them should be the buffer block size for one row. Ideally, a buffer block should accommodate at least 100 rows at a time.

$$\text{Buffer Block Size} = \text{Largest Row Precision} * 100$$

You can change the buffer block size in the session configuration as shown below:

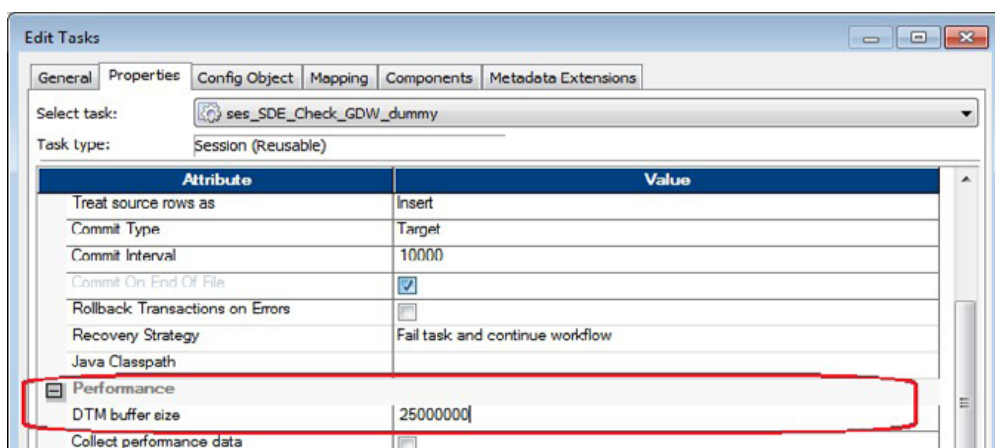


When you increase the DTM Buffer memory, the Integration Service creates more buffer blocks, which improves performance. You can identify the required DTM Buffer size based on this calculation:

$$\text{Session Buffer Blocks} = (\text{total number of sources} + \text{total number of targets}) * 2$$

$$\text{DTM Buffer size} = \text{Session Buffer Blocks} * \text{Buffer Block size} / 0.9$$

You can change the DTM Buffer size in the session configuration as shown below:



Push Down Optimization

When Informatica session is configured for Push Down Optimization (PDO), it enables data transformation processing, where appropriate, to be pushed down into any relational database to make the best use of database assets. With this option, PowerCenter is the first data integration software on the market to offer a choice of where data processing occurs.

In general, PDO can improve session performance in the following circumstances:

- 1: When it reduces the number of rows passed between Oracle and PowerCenter. For example, pushing a Filter transformation to the Oracle source can reduce the number of rows PowerCenter extracts from the source.
- 2: When the database server is more powerful than the PowerCenter server. For example, pushing a complex Expression transformation to the source or target improves performance when the database server can perform the expression faster than the server on which the PowerCenter Integration Service runs.
- 3: When the generated query can take advantage of prebuilt indexes. For example, pushing a Joiner transformation to the Oracle source improves performance when the database can join tables using indexes and statistics that PowerCenter cannot access.
- 4: PDO can help to improve session performance due to its ability to leverage the Oracle Super cluster data processing capabilities or Teradata DBMS parallel engine while maintaining the supportability, flexibility, and metadata lineage of the process as if it were entirely within the ETL GUI.

③ Push Down Optimization

You can configure the following types of pushdown optimization:

Source-side pushdown optimization: The Integration Service pushes as much transformation logic as possible to the source database.

Target-side pushdown optimization: The Integration Service pushes as much transformation logic as possible to the source database.

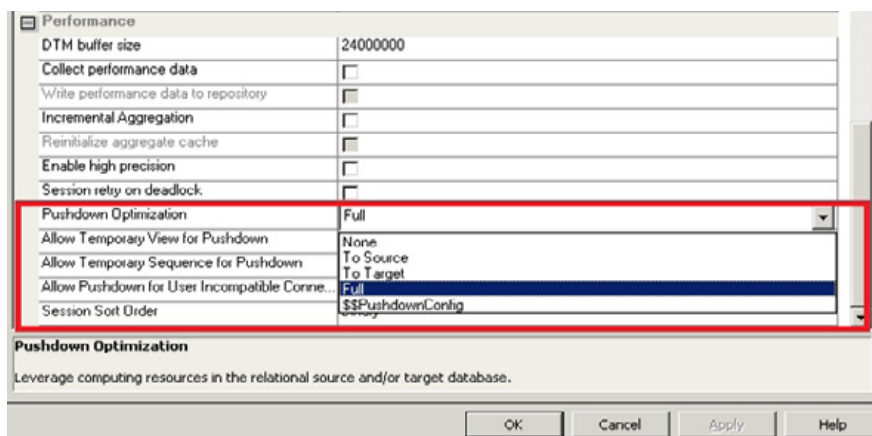
Full pushdown optimization: The Integration Service attempts to push all transformation logic to the target database. If the Integration Service cannot push all transformation logic to the database, it performs both source-side and target-side pushdown optimization.

We can configure sessions for pushdown optimization with any of the databases like Oracle, IBM DB2, Teradata, Microsoft SQL Server, Sybase ASE, or databases that use ODBC drivers.

Decision Matrix for Choosing PDO vs Informatica classic ETL						
Note: Consider Oracle as Target database for the scenarios given below.						
Scenario	Source	Target	PDO supported Transformations?	Lookup Table	Recommendation	Notes
S1	Oracle	Oracle	Yes	Oracle	Full PDO	
S2	Non-Oracle	Oracle	Yes	Non-Oracle	Partial PDO on Source	If it is direct extract, i.e. without any transformations, use Informatica classic ETL (without PDO)
S3	Heterogeneous Sources and at least one of the sources is in Oracle	Oracle	Yes	Oracle	Consider two-step approach: 1. Load Non-Oracle sources to Oracle staging table using loader utilities or PowerCenter engine (Informatica classic ETL). 2. In second map (Oracle Stage -> Transformations -> Oracle Target) use full PDO option.	
S4	Oracle	Oracle	No	Oracle	Go with classic ETL, with appropriate Informatica Partitioning approach to improve performance.	
S5	Flat File	Oracle	NA	Non-Oracle	Informatica classic ETL (without PDO)	

Configuring Session for Pushdown Optimization

A session can be configured to use pushdown optimization from Informatica PowerCenter Workflow Manager. You can open the session and choose the Source, Target or Full pushdown optimization as shown below:

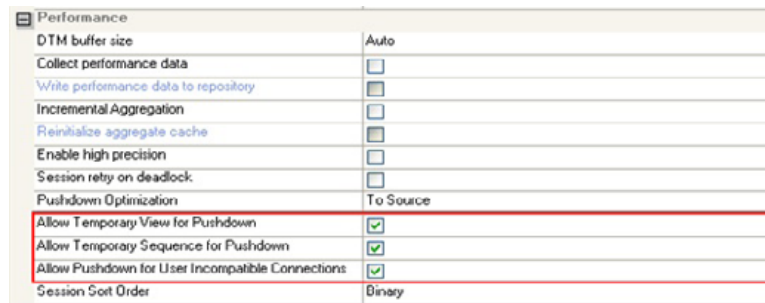


You can configure the following types of pushdown optimization:

Allow Temporary View for Pushdown: Allows the Integration Service to create temporary view objects in the database when it pushes the session to the database.

Allow Temporary Sequence for Pushdown: Allows the Integration Service to create temporary sequence objects in the database.

Allow Pushdown for User Incompatible Connections: Indicates that the database user of the active database has read permission on the idle databases.



Performance	
DTM buffer size	Auto
Collect performance data	<input type="checkbox"/>
Write performance data to repository	<input type="checkbox"/>
Incremental Aggregation	<input type="checkbox"/>
Reinitialize aggregate cache	<input type="checkbox"/>
Enable high precision	<input type="checkbox"/>
Session retry on deadlock	<input type="checkbox"/>
Pushdown Optimization	To Source
Allow Temporary View for Pushdown	<input checked="" type="checkbox"/>
Allow Temporary Sequence for Pushdown	<input checked="" type="checkbox"/>
Allow Pushdown for User Incompatible Connections	<input checked="" type="checkbox"/>
Session Sort Order	Binary

④ Informatica Partitioning

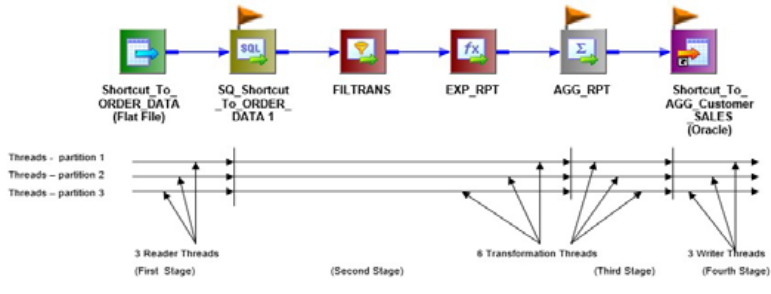
The Informatica PowerCenter Partitioning Option increases the performance of PowerCenter through parallel data processing. The partitioning option will let you split the large data set into smaller subsets which can be processed in parallel to get a better session performance.

Partitioning Terminology

Let's run through some partitioning terminology before we go into more details:

- 1: Partition:** The Integration Service pushes as much transformation logic as possible to the source database.
- 2: Number of partitions:** We can divide the data set into smaller subsets by increasing the number of partitions. When we add partitions, we increase the number of processing threads, which can improve session performance.
- 3: Stage:** A stage is the portion of a pipeline, which is implemented at run time as a thread.
- 4: Partition Point:** This is the boundary between two stages and it divides the pipeline into stages. A partition point is always associated with a transformation.
- 5: Partition Type:** An algorithm for distributing data among partitions, which is always associated with a partition point. The partition type controls how the Integration Service distributes data among partitions at partition points.

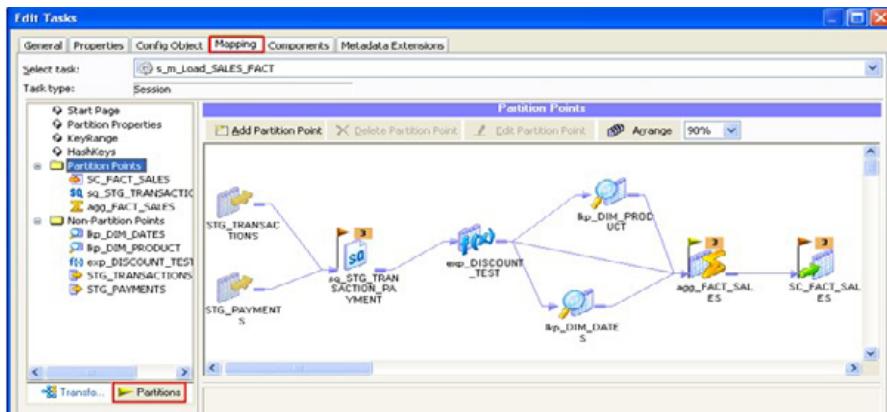
The image below shows the points mentioned above. We can see three partitions and three partition points in the demo session:



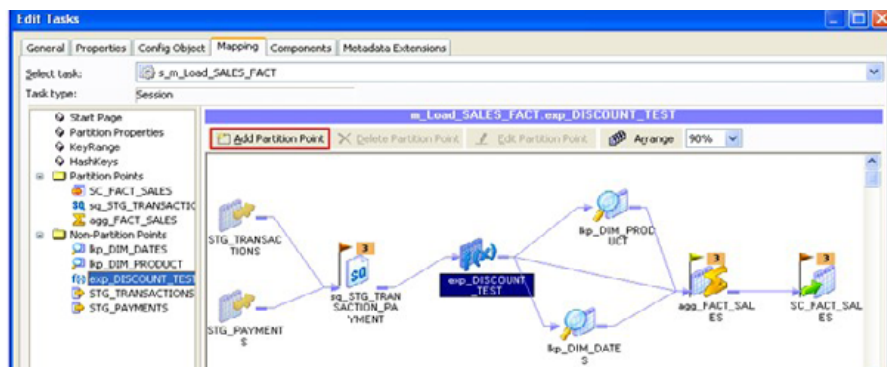
Setting up Session Partitions

Let's see what is required to set up a session with partition enabled.

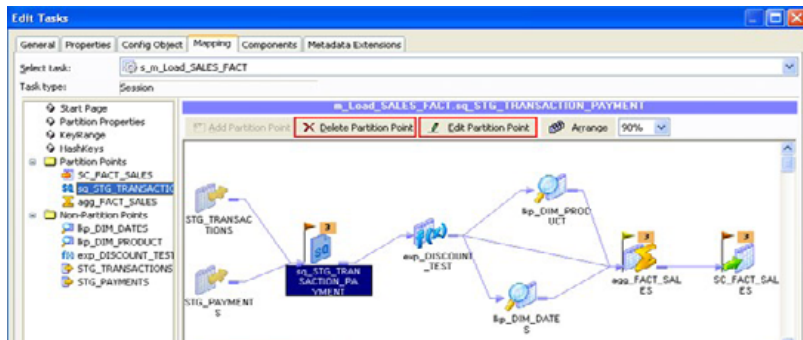
You can invoke the user interface for session partition, as shown below, from your session using the menu Mapping-> Partitions.



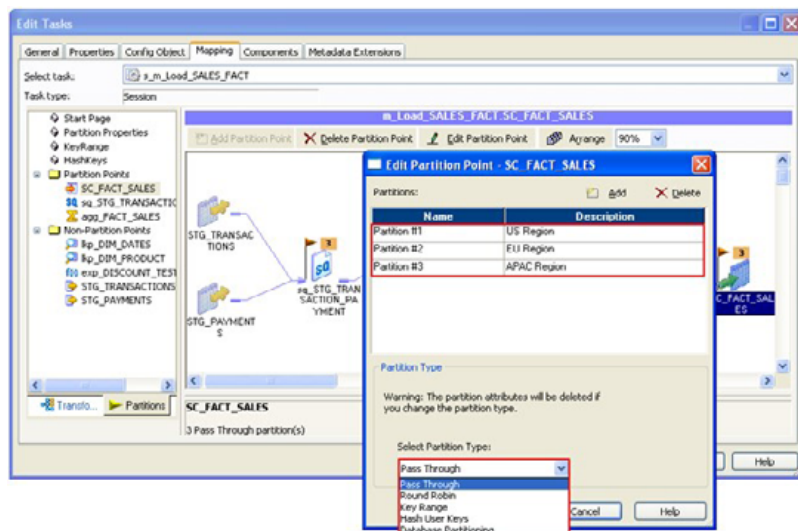
The interface will let you add/modify partitions, partition points and choose the type of partition algorithm. Choose any transformation from the mapping and the "Add Partition Point" button will let you add additional partition points.



Choose any transformation from the mapping and the “Delete Partition Point” or “Edit Partition Point” buttons will let you modify partition points.



The “Add/Delete/Edit Partition Point” opens up an additional window which lets you modify the partition and choose the type of the partition algorithm as shown below:



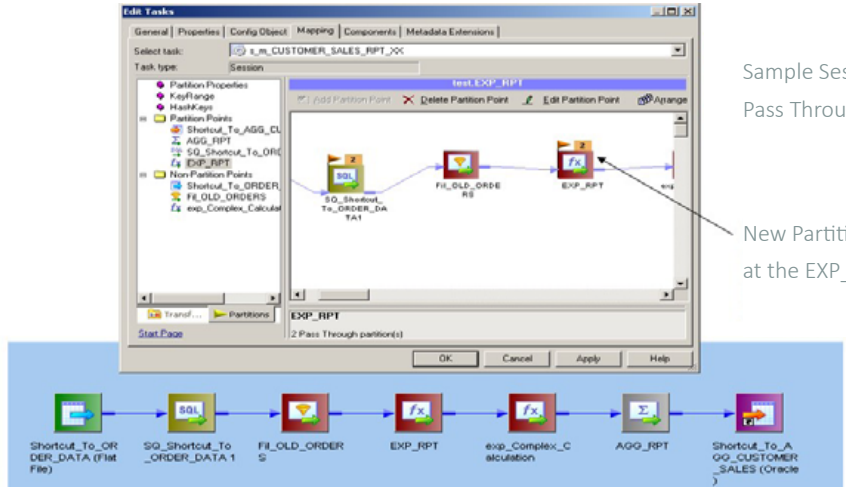
Partition Types

Different types of partition algorithms:

Pass-through Partitioning

In pass-through partitioning, the Integration Service processes data without redistributing rows among partitions. All rows in a single partition stay in the partition after crossing a pass-through partition point. Choose pass-through partitioning when you want to create an additional pipeline stage to improve performance, but do not want to change the distribution of data across partitions.

Example of Pass-through Partitioning:



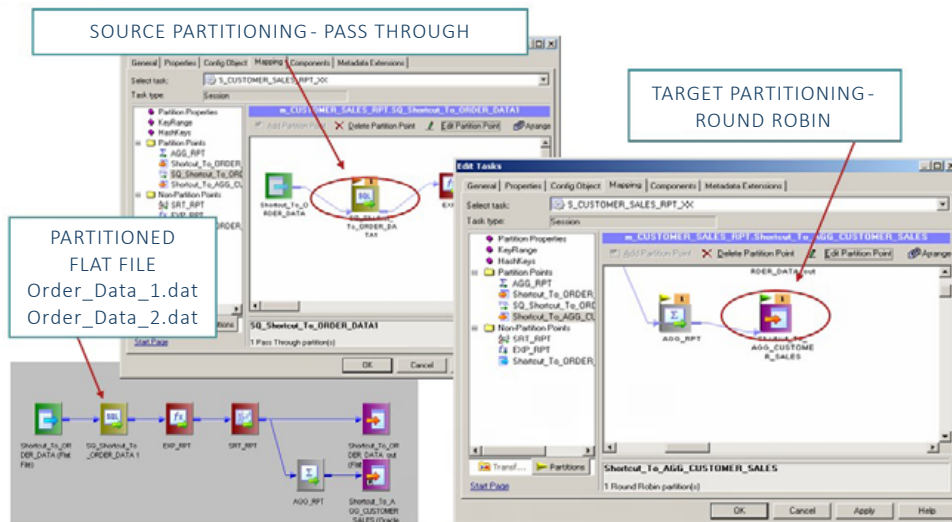
Sample Session with Pass Through Partitioning

New Partition Point at the EXP_RPT Transformation

Round Robin Partitioning

The Integration Service distributes data evenly among all partitions. Use round robin partitioning where you want each partition to process approximately the same number of rows.

Example of Round Robin Partitioning:



Key Range Partitioning

With key range partitioning, the Integration Service distributes rows of data based on a port or set of ports that you define as the partition key. For each port, you define a range of values. The Integration Service uses the key and ranges to send rows to the appropriate partition. Use key range partitioning when the sources or targets in the pipeline are partitioned by key range.

Example of Key Range Partitioning:

1: **Based on source partitioning:** Determines the number of partitions using database partition information. The number of partitions is the maximum number of partitions at the source.

2: **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.

3: **Based on number of nodes in the 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.

4: **Based on number of partitions:** 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

• **Note:** Do not configure dynamic partitioning for a session that contains manual partitions. If you set dynamic partitioning and you manually partition the session, the session will be invalid.

Rules and Guidelines for Dynamic Partitioning

Use the following rules and guidelines for dynamic partitioning:

1: Dynamic partitioning uses the same connection for each partition.

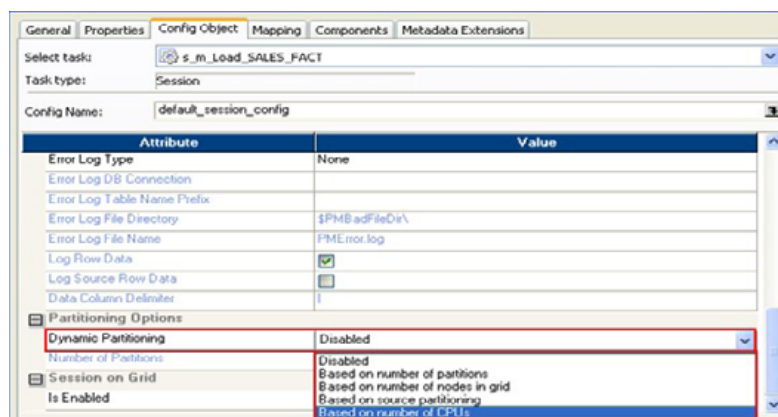
2: You cannot use dynamic partitioning with XML sources and targets.

3: Sessions that use SFTP fail if you enable dynamic partitioning.

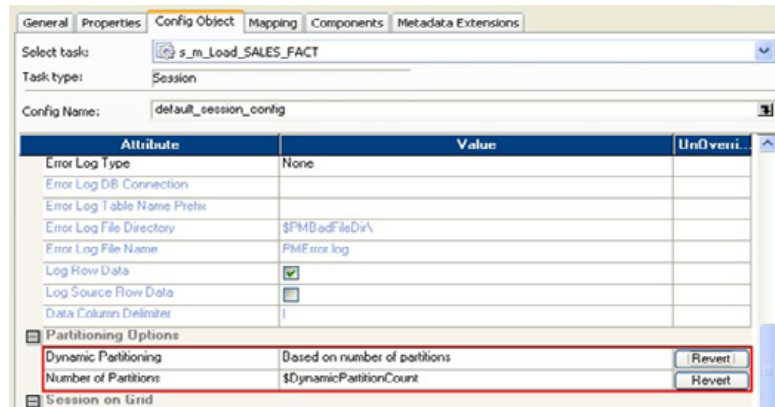
4: When you set dynamic partitioning to a value other than disabled, and you manually partition the session on the Mapping tab, you invalidate the session.

Session Configuration with Dynamic Partitioning

You can set up sessions for dynamic partition from Config Object Tab in the session properties. You can choose the available partitioning options from the drop-down list as shown in image below:



If the dynamic partition is setup based on “Based on number of partitions”, the “Number of Partitions” property can be set up using the session parameter `$DynamicPartitionCount`, or by entering a number greater than 1 as shown below:



6 Concurrent Workflow Execution

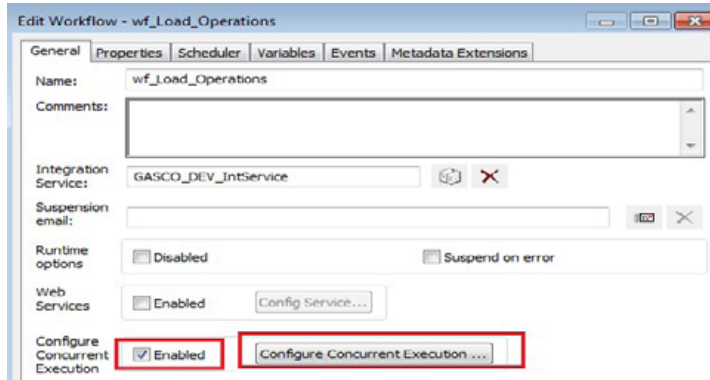
In large data integration projects, it is not uncommon to source data from multiple systems, sources, regions etc... As the number of data sources increases, the ETL load time also increases because of the increasing data volume. One way to reduce the load time is by running different ETL processes in parallel. Informatica PowerCenter’s capability to run workflow concurrently can be used in such scenarios to reduce the ETL load time.

A concurrent workflow is a workflow that can run as multiple instances concurrently. A workflow instance is a representation of a workflow. We can configure two types of concurrent workflows.

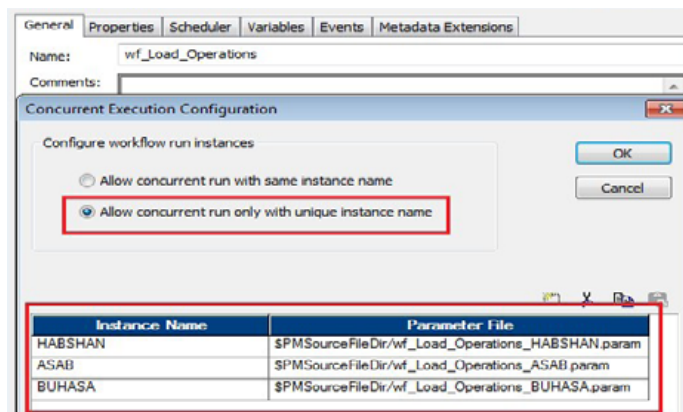
- 1: **Allow concurrent workflows with the same instance name:** Configure one workflow instance to run multiple times concurrently. Each instance has the same source, target, and variables parameters. The Integration Service identifies each instance by the run ID.
- 2: **Configure unique workflow instances to run concurrently:** Define each workflow instance name and configure a workflow parameter file for the instance. You can define different sources, targets, and variables in the parameter file.

For the demonstration, let’s consider a scenario where we need to load daily transaction data from different sites. These three files are expected to be available at around the same time, so we create one workflow to load the operations data, and the same workflow will be used to load all 3 files, which can execute concurrently.

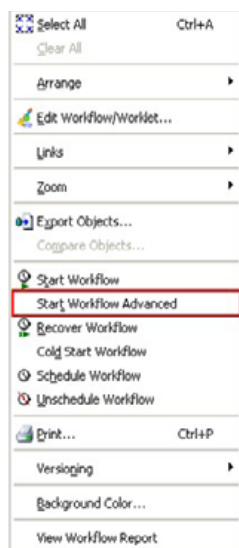
Once the workflow is created, enable concurrent execution as shown in the image below:



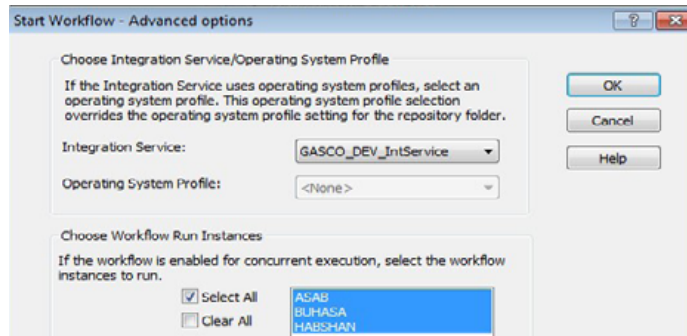
Now click on “*Configure Concurrent Execution*” with the properties shown in the image below. Provide three different parameter files, which contain the source file information for the corresponding region.



Now the concurrent workflow configuration is done; to trigger the workflow, you use the “*Start Workflow Advanced*” option as shown below:



Choose the workflow instance name from the pop-up window and click OK to run the selected workflow instance.



From the Workflow Monitor you can see the running instance of the workflow, with the instance name appended [] to the workflow name which is running concurrently.

7 Load Balancing for Workload Distribution on the Grid

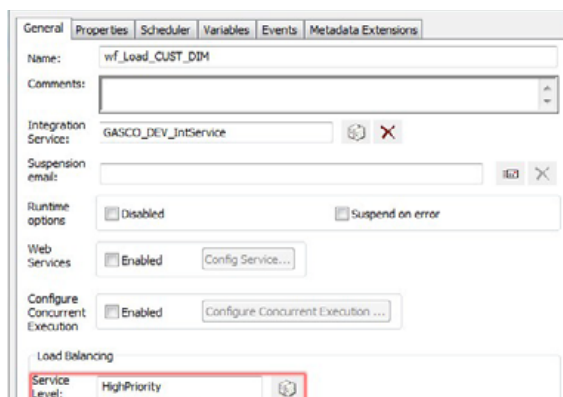
Informatica load balancing is a mechanism which distributes the workloads across the nodes in the grid. When you run a workflow, the Load Balancer dispatches different tasks in the workflow such as Session, Command, and predefined Event-Wait tasks to different nodes running the Integration Service. Load Balancer matches task requirements with resource availability to identify the best node to run a task; it may dispatch tasks to a single node or across nodes on the grid.

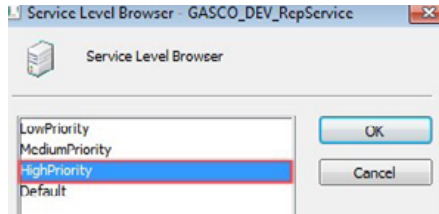
Assigning Service Levels to Workflows

Service levels determine the order in which the Load Balancer dispatches tasks from the dispatch queue. When multiple tasks are waiting to be dispatched, the Load Balancer dispatches high priority tasks before low priority tasks. You create service levels and configure the dispatch priorities in the Administrator tool.

You give Higher Service Level for the workflows which need to be dispatched first, when multiple workflows are running in parallel. Service Levels are set up in the Admin console.

You assign service levels to workflows on the General tab of the workflow properties as shown below:





⑧ Conclusion

Do not use all the techniques discussed above at once; instead, set up a bench mark for each data load scenario and start with simple techniques first to improve performance, and when acceptable performance is reached, stop tuning because each tuning technique is progressively harder, costs more money and is trickier to maintain.

ABOUT CLEARPEAKS

We are a specialist Business Intelligence consulting firm, headquartered in Barcelona, with offices in Dubai & Abu Dhabi and sales presence in London. We provide BI services to customers in 15 vertical industries and across several countries in Europe, Middle East and Africa. We are Oracle & Tableau partners.

Our Business Intelligence solutions place valuable information to help our customers make important business decisions around sales operations, marketing, finance, supply chain management or resource management.

We differentiate ourselves through in-depth Business Intelligence knowledge and a dedicated approach to customer satisfaction. Our pragmatic style, coupled with our customer loyalty and quality focus has been the key to our success over the years.

See more in:

CLEARPEAKS

www.clearpeaks.com

info@clearpeaks.com



Barcelona (Head Office)

Travessera de Gràcia 56,
6^º 1^ª - 08006

Abu Dhabi

Office No. 1954, 19th floor
Al Ghaith Tower, Hamdan Street AYA
Business Center

Dubai

Office No. 101, 1st floor,
Building A, Dubai Outsource Zone P.O.
Box 500069