



Performance Tuning

Performance Tuning

- First step in performance tuning is to identify the performance bottleneck in the following order:
 - Target
 - Source
 - Mapping
 - Session
 - System
- The most common performance bottleneck occurs when the Integration Service writes to a target database

Target Bottlenecks

- Identify
 - A target bottleneck can be identified by configuring the session to write to a flat file target
- Optimize
 - Dropping indexes and key constraints before loading
 - Increasing commit intervals
 - Use of Bulk Loading / External Loading

Source Bottlenecks

- Identifying
 - Add a filter condition after Source qualifier to false so that no data is processed past the filter transformation. Even then if the new session takes same amount of time, then there is a source bottleneck
 - In a test mapping remove all the transformations
- Optimizing
 - Optimizing the Query by using hints
 - Use Informatica Conditional Filters if the source system lacks indexes

Mapping Bottlenecks

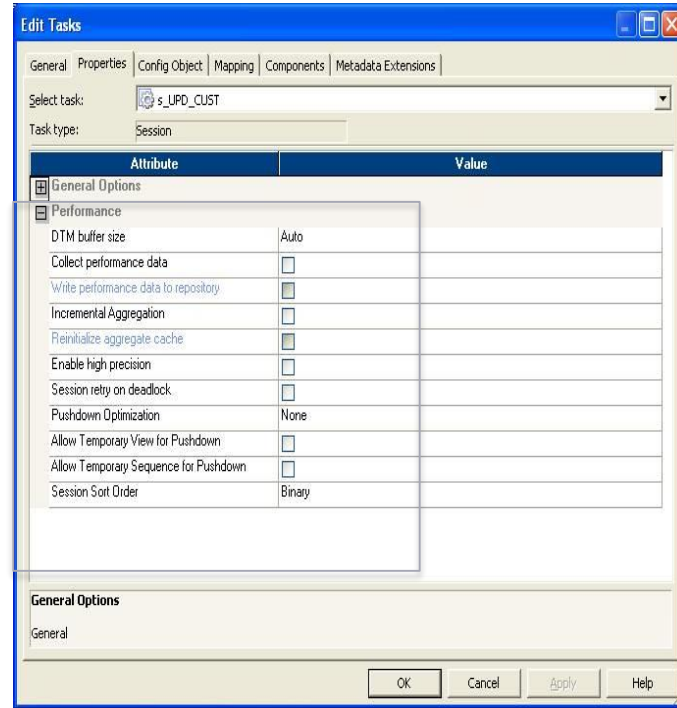
- Identifying
 - If there is no Source bottleneck, add a Filter transformation in the mapping before each target definition. Set the filter condition to false so that no data is loaded into the target tables. If the time taken to run remains same, then there is a mapping bottleneck
- Optimize
 - Configure for Single-Pass reading
 - Avoid unnecessary data type conversions
 - Avoid database reject errors
 - Use Shared Cache / Persistent Cache

Session Bottlenecks

- Identifying
 - If there is no Source, Target or Mapping bottleneck, then there may be a session bottleneck
 - Use Collect Performance Details : Any value other than zero in the read from disk and write to disk counters for Aggregator, Rank and Joiner transformations indicate a session bottleneck.
- Optimizing
 - Increase the number of partitions
 - Tune session parameters
 - DTM Buffer Size (6M – 128M)
 - Buffer Block Size (4K – 128K)
 - Data (2M – 24 M)/ Index (1M-12M) Cache Size
 - Use incremental Aggregation if possible

Session Bottlenecks - Memory

- Configure the index and data cache memory for the Aggregator, Rank, and Joiner transformations in the Configuration Parameters dialog box
- The amount of memory you configure depends on partitioning, the transformation that requires the largest cache, and how much memory cache and disk cache you want to use



System Bottlenecks

- Identifying
 - If there is no Source, Target, Mapping or Session bottleneck, then there may be a system bottleneck
 - Use system tools to monitor CPU usage, memory usage, and paging
 - On Windows : Task Manager
 - On Unix : Systems tools like sar iostat. For E.g. sar -u (%usage on user, idle time, i/o waiting time)
- Optimizing
 - Improve network speed
 - Improve CPU performance
 - Check hard disks on related machines
 - Reduce paging

Activity: Choose all that are correct

A. To optimize system bottleneck, we need to:

1. Decrease network speed
2. Improve CPU performance
3. Check hard disks on related machines
4. Increase paging

B. To resolve the target bottleneck, we need to:

1. Dropping indexes and key constraints before loading
2. Decrease commit intervals
3. Choose Normal Load
4. Create indexes on the target tables

C. We set the incremental aggregation property :

1. Session properties
2. Mapping properties
3. Workflow property
4. There is no such property

PMCMD

- Can use the command line program PMCMD to communicate with the Integration Service
- Can perform the following actions with pmcmd:
 - Determine if the Integration Service is running
 - Start Workflows
 - Stop Workflows
 - Recover Workflows
- Can configure repository usernames and passwords as environmental variables with pmcmd
- Can also customize the way pmcmd displays the date and time on the machine running the Integration Service process
- pmcmd returns zero on success and non-zero on failure
- You can use pmcmd with operating system scheduling tools like cron to schedule sessions, and you can embed pmcmd into shell scripts or Perl programs to run or schedule sessions

PMCMD (Contd.).

- Need the following information to use PMCMD:
 - Repository username
 - Repository password
 - Connection Information – Domain Name, Integration Service name
 - Workflow name - The names of any workflows you want to start or stop
 - Folder name - The folder names for those workflows
 - Parameter file

Summary

In this session, we learnt to:

- Distinguish Various performance bottlenecks
- Using PMCMD



Thank You