HITACHI
Inspire the Next

Clearing the Mondrian Cache with Pentaho

Change log (if you want to use it):

Date	Version	Author	Changes

Contents

Overview

This document provides methods for clearing and managing the Mondrian cache in clustered environments. The method you choose will depend on how the Pentaho server is used and how the extract/transform/load (ETL) execution processes results.

Some of the topics that will be discussed in this document are using Pentaho Data Integration (PDI) or external tools to clear the Mondrian cache with application program interfaces (APIs), using the Scheduler to monitor processes and manage a `CLEAR ALL CACHE` event, and using the Dynamic Schema Processor (DSP) to segment a new cache.

The intention of this document is to speak about topics generally; however, these are the specific versions covered here:

Software	Version(s)
Pentaho	6.x, 7.x, 8.0

The Components Reference in Pentaho Documentation has a complete list of supported software and hardware.

Use Case: Clearing the Cache to Increase Querying Speed

Fabiola has a sales database for a busy company, and it contains frequently changing data. She needs to clear her Mondrian cache and find an effective way to manage it because its information becomes out of date, rapidly, and it is no longer useful to speed up queries. Of the few methods available, she uses the Clear Cache method. This method requires the ETL process to implement a call to the Pentaho server, just after a transformation.

Cache Management

There are different cache strategies that can be used based on how the Reporting (ETL) server is deployed and how the ETL execution processes results in a clustered environment.

You will only need to call one of the nodes to clear the cache; other nodes will become cleared automatically.

The following methods describe different techniques that can be used to manage the cache in Mondrian:

- Clearing the Cache with APIs
- Using the Scheduler
- Using DSP to Segment a New Cache

Clearing the Cache with APIs

To clear the cache, after a transformation, the ETL process that loads new and successfully modified data must implement a call to the Pentaho server. For security reasons, you should apply an IP-trusted filter to accept requests from the calling ETL without authentication.

You need to use Analyzer to call `Clear Cache` because the API is an Analyzer API. You will only need to call one of the nodes to clear the cache; other nodes will become cleared automatically.

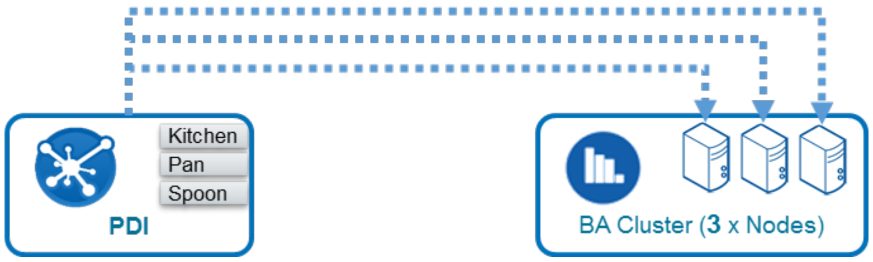

Figure 1: Using PDI to Clear Cache in a Cluster of BA-Only Servers

You will need to have administrator permissions to run these APIs. You will receive a TRUE response when this is done successfully, and an ERROR response, otherwise.

- Run this API to clear one catalog only:

```
../api/repos/xanalyzer/service/ajax/clearCache?catalog=SteelWheels&time=146
4256845958
```

- Run this API to clear all catalogs:

```
/api/system/refresh/mondrianSchemaCache
```

The `time=<timestamp>` is not required in the case of clearing one or all catalogs. Instead, it is used to force browsers or clients to submit the request and not use the cache.

During distributed cache implementation in clustered environments, calling any of the nodes will clear the cache for the entire cluster of nodes. If no distributed cache is applied, the external process will need to make the API call to the known nodes.

Using the Scheduler

This method is based on a recurrent monitoring process that can be established on a `CLEAR-ALL-CACHE` event from the API call. Scheduled Events can be logged in an ETL Status Table, Time Trigger (Fixed Time), or other.

You will only need to call one of the nodes to clear the cache; other nodes will become cleared automatically.

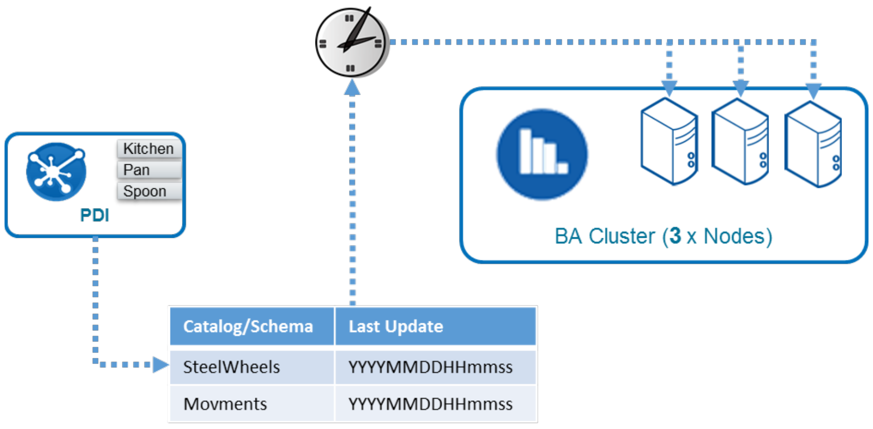

Figure 2: Using the Scheduler

During distributed cache implementation in clustered environments, calling any of the nodes will clear the cache for the entire cluster of nodes. If no distributed cache is applied, the external process will need to make the API call to the known nodes.

Using the DSP to Segment a New Cache

This method is based on changing the Mondrian schema signature (hash file signature) to create a new cache segment. The DSP will force the OLAP engine to segment the new cache by injecting the last successful data load `TIMESTAMP` as part of the schema definition.

You can use session or global platform variables to specify the last successful `TIMESTAMP`, depending on your needs. This new data may not be available until the new cache finishes processing.

Pentaho's documentation has more detailed information and methods about the Dynamic Schema Processor and the Schema Processor.

We recommend segmenting the cache in a controlled manner if new cache segments do not change often (many times a day).

Each project and schema has a DSP. The DSP will evaluate or search for specific session information (parameter-based) to modify or update the schema `TIMESTAMP` placeholder. This will force the OLAP engine to segment a new cache area.

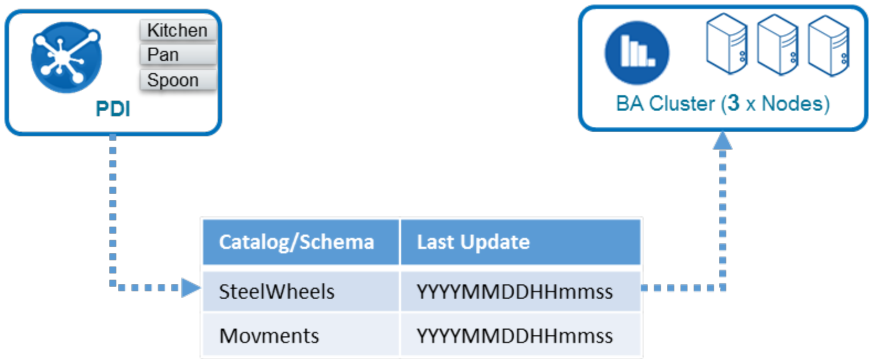

Figure 3: Using DSP to Segment a New Cache

The DSP will use a `TIMESTAMP` injection in a comment on the file or in an SQL filter, based on the situations and options below:

Table 1: How the DSP Uses the `TIMESTAMP` Injection

Situation	`TIMESTAMP` Injection Details
Adding accessible information on last update to data	When data is up-to-date to a certain point in time, the information is saved somewhere that the Pentaho server(s) can see and read it. For example, you can store it in a table in a database called a control table (so called because it contains a lot of metadata).
Reading the `TIMESTAMP`	A session variable is loaded with the `TIMESTAMP` of the ETL status or fixed time. The name of the session variable can be `schema` or `project`. We recommend reading the timestamp when a user logs into any Pentaho server.
Keeping Data Integrity (optional)	The `TIMESTAMP` should force an SQL injection to keep user data integrity. The pattern is similar to the following: `<SQL dialect="generic"> (DATE_FIELD <= TIMESTAMP) </SQL>`
Using the DSP to Inject a Condition into All Queries	This makes sure that only data that has been vetted and processed entirely is included in the reports that the user produces. After the ETL processes are completed, the DSP can use a new `TIMESTAMP`, which will instruct Mondrian to include the new data as well from that point on.
Updating `TIMESTAMP` in session or global variables	`TIMESTAMP` can be updated in session or global variables. The schema cache will change for current logged users, in the case of global variables.

This method works in passive mode. There is no need to do maintenance to the node's list, because each node will run the DSP and independently detect the timestamp change by itself. The DSP signature will force make multiple servers to use the same cache segment.

Related Information

Here are some links to information that you may find helpful while using this best practices document:

- Dynamic Schema Processor
- Schema Processor
- Pentaho Components Reference

Finalization Checklist

This checklist is designed to be added to any implemented project that uses this collection of best practices, to verify that all items have been considered and reviews have been performed.

Name of the Project:____

Date of the Review:____

Name of the Reviewer:____

Item	Response	Comments
Did you run APIs to clear your cache?	YES_______ NO_______	
Did you use the Scheduler?	YES_______ NO_______	
Did you use DSP to segment a new cache?	YES_______ NO_______	
Did you use a control table?	YES_______ NO_______	
Which method did you use to clear the cache, and why?	YES_______ NO_______	