NOKIA

Performance Management Data Retrieval From External Hadoop Using Hive Interface

DECEMBER 2022

ISSUE 1

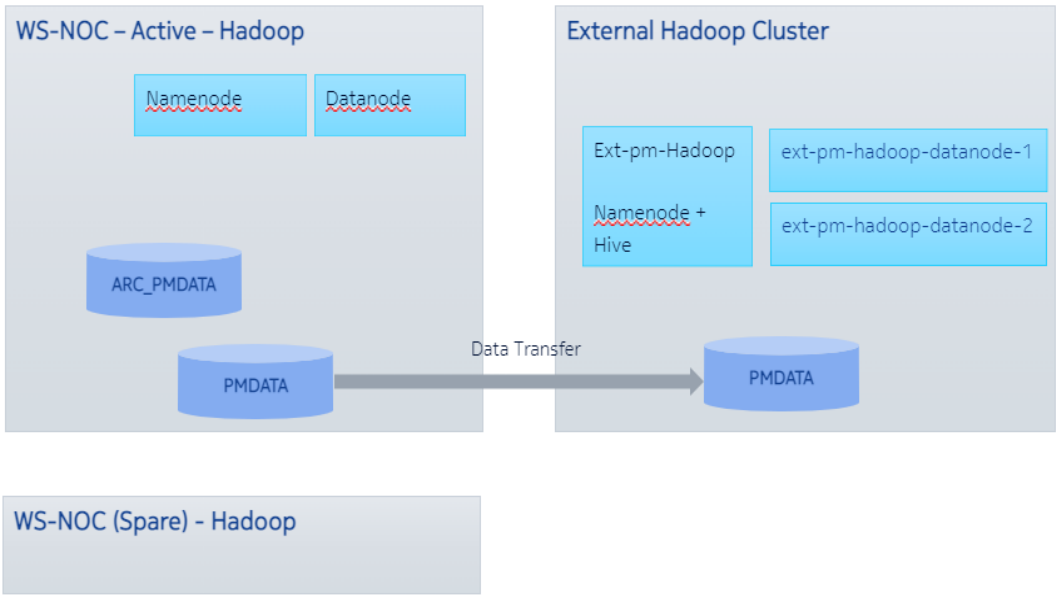
Table of Contents

1.1 Overview

External Hadoop is an OPTIONAL configuration that is required only in case of extended data storage for Performance Management. With the R22.12, Performance management supports the storage of PM data up to 1 year in an External Hadoop cluster.

Note: The External Hadoop can be deployed ONLY on a Distributed setup. In a Classic/ Standalone setup, the External Hadoop is not applicable.

The following diagram shows the minimal cluster setup that depicts the WS-NOC-Active (local) and Standby Hadoop (Spare) along with the External Hadoop cluster that stores an additional 1 year of PM data. PM data is copied from the WS-NOC Active Hadoop system to the External Hadoop cluster (once every 6 hours).



In a local Hadoop, PM data has a maximum storage of up to 30 days of raw data and 30 days of archived data, so a maximum of 2 months of data can be stored. This can be

configured as per the user's requirement. With the R22.12, depending on the user requirement, data from local Hadoop can be stored in an External Hadoop for a period of 1 year, for the user analytics application to analyze the PM data.

PM NextGen stores the PM data for each NE and Port in separate records in AVRO format in the Hadoop file system. The files are written under partitioned sub-directories in the Hadoop file system.

Folders that are created with partitionkey=value name format are called partitioned folders in the Hadoop file system. Partitioning the data provides performance benefits and helps in organizing the data better. A few examples of partitioning are ds (indicating date) and ts (indicating timestamp).

The following PM data gets transferred from local Hadoop to External Hadoop.
• /PMDATA/FIFTEEN_MINS/ - The PM data of 15 minutes granularity for all NEs and Ports are stored in this table.
• /PMDATA/ONE_DAY/ - The PM data of 24 hours granularity for all NEs and Ports are stored in this table.
• /PMDATA/KPIAGGR/FIFTEEN_MINS/ - The Connection level KPI data such as Span loss and
Channel Margin of 15 minutes granularity for all NEs and Ports are stored in this table.
• /PMDATA/KPIAGGR/ONE_DAY/ - The Connection level KPI data such as Span loss and Channel Margin of 24 hours granularity for all NEs and Ports is stored in this table.

The data transfer happens once every 6 hours and the PM data is copied from the current WS-NOC Active(local) Hadoop to External Hadoop.

For R22.12, the high availability (HA) is provided only at the data node level using the replication mechanism provided by Hadoop.

The Replication Factor is set to 2 for an External Hadoop cluster. Any data transfer that happens from local Hadoop to External Hadoop can be stored in two different data nodes. When one data node goes down, the user can still retrieve the PM data from the other data node.

PM Data can be retrieved from External Hadoop using the Hadoop Native libraries by directly connecting to HDFS file system and reading the Avro files.

PM NextGen also provides Hive SQL interface to read PM data from these Hadoop files. Hive enables data summarization, querying, and analysis of data. Hive is a data warehouse framework for querying and analysis of data that is stored in HDFS.

The Hive interface is exposed from the system where ext-pm-hadoop container is running. The port used to access is 8765. The SQL connection string format is jdbc:hive2://" + ip + ":8765/;saslQop=auth-conf, where IP address is of the system where the ext-pm-hadoop container is running. This interface is protected with User ID and password authentication mechanism. Use the same User ID and password that was used to login to WS-NOC GUI to connect to this JDBC interface.

Hive provides SQL-like interface called HiveQL. It allows users familiar with SQL to retrieve data from Hadoop with large amount (petabytes) of data. Hive queries are written in HiveQL, which is a query language like SQL.

External Hadoop GUI provides the data content and other Hadoop related details. It can be accessed using either of the following URL:
• https://<EXTERNAL_HADOOP_IP_ADDRESS>:9870/
• https://<WS-NOC_ACTIVE_SYSTEM_IP_ADDRESS>:1975/

When WS-NOC HA system switchover occurs, the External Hadoop identifies the newly ACTIVE system and the data transfer happens from the newly ACTIVE system only. And the frequency of the data transfer is as per the properties defined in the ext-pm-hadoop:/nfmt/config/ sparkjobconfig.properties file. The change to the frequency properties will be effective from the next run of the scheduler job.

1.2 Hive SQL Interface

PM NextGen defines Hive tables to retrieve the PM data in an organized manner from the Hadoop files. The tables are defined inside the pmdb database.
The following tables are defined for retrieval of PM data:
• nemetric15mins: The PM data of 15 minutes granularity for all NEs and Ports are stored in this table.
• nemetriconeday: The PM data of 24 hours granularity for all NEs and Ports are stored in this table.
• nemetrickpi15mins: The Connection level KPI data such as Span loss and Channel Margin of 15 minutes granularity for all NEs and Ports are stored in this table.
• nemetrickpioneday: The Connection level KPI data such as Span loss and Channel Margin of 24 hours granularity for all NEs and Ports are stored in this table.

1.3 Reading Data through Hadoop File System

This section explains a sample Java client program on how to connect to the Hadoop File system and read the Avro files.

Three libraries are required to build and run the JAVA Hadoop Client program,
'org.apache.hadoop:hadoop-client:3.3.3'
'org.apache.hadoop:hadoop-common:3.3.3'
'org.apache.hive:hive-exec:4.0.0-alpha-2'

The client java program can be created in any maven or Gradle build setup. Below is a sample Gradle project setup that you can create in your development machine.

Download the sampleclient.zip from network developer portal (https://network.developer.nokia.com/learn/optical-management-apis/NFMT-downloads/), and create a Gradle project sampleclient folder with the below 7 files:

- sampleclient/build.gradle
- sampleclient/src/main/java/NePmDataReader.java
- sampleclient/src/main/java/com/nokia/umcpm/model/pmdb/MetricGroup.java
- sampleclient/src/main/java/com/nokia/umcpm/model/pmdb/Metric.java
- sampleclient/src/main/java/com/nokia/umcpm/model/pmdb/PmDataHdfs.java
- sampleclient/src/main/java/com/nokia/umcpm/model/pmdb/Port.java
- sampleclient/src/main/java/com/nokia/umcpm/model/pmdb/MetricGroupKey.java

Note: Prerequisite: Customer should have java11 and gradle installed in their development server.

Example steps to install gradle on Linux are provided here :

• Download the gradle package using below command.
wget https://services.gradle.org/distributions/gradle-5.1-bin.zip

• Create the /opt/gradle directory using below command
mkdir /opt/gradle

• Execute “unzip -d /opt/gradle gradle-5.1-bin.zip” command

• Execute “export PATH=$PATH:/opt/gradle/gradle-5.1/bin” command

• Check the version of Gradle using “gradle –v”

Reference: https://www.vultr.com/docs/how-to-install-gradle-on-centos-7

1.4 To Build the Gradle Project from the development machine

$ cd sampleclient

$ gradle clean build

Transfer the build/libs/sampleclient-0.0.1.jar file to /opt/pmhadoop/ folder in ext-pm-container and run below command:

Login to ext-pm-hadoop VM and then do the following commands:

```
$ docker exec -it -u otn:gadmin ext-pm-hadoop bash
$ cd /opt/pmhadoop/
$ /usr/lib/jvm/java-11-openjdk-11.0.16.1.1-1.el8_6.x86_64/bin/java -cp
/opt/pmhadoop/sampleclient-0.0.1.jar:/opt/hadoop-3.3.1/share/hadoop/tools/lib/hadoop-client-3.3.1.jar:/opt/hadoop-3.3.1/share/hadoop/common/hadoop-common-3.3.1.jar:/opt/pmhive/lib/hive-exec-3.1.2.jar:/opt/hadoop-3.3.1/share/hadoop/client/hadoop-client-runtime-3.3.1.jar:/opt/slf4j/1.7.36/slf4j-api.jar:/opt/slf4j/1.7.36/jcl-over-slf4j-1.7.36.jar:/opt/hadoop-3.3.1/share/hadoop/hdfs/hadoop-hdfs-client-3.3.1.jar:/opt/pmhive/lib/super-csv-2.2.0.jar:/opt/hadoop-3.3.1/share/hadoop/common/lib/*:
NePmDataReader  <container name> <granularity> <DATE> <TIME> <NEID>
```

Note:

- <container name> - To read from External Hadoop, the container name is ext-pm-hadoop
- <granularity> - Should be 15mins or 24hours
- <DATE> - The UTC bin date in yyyyMMdd format
- <TIME> - The UTC bin time in HHmmss format. Give double quotes ("") if all timestamps in that dated folder should be included. For 24-hour granularity, this should be 000000
- <NEID> - Identifier of NE used inside WS-NOC. Give double quotes ("") if all NE IDs should be included.
- The NE Id can be obtained from the WS-NOC Nodes table UI page or by invoking the GET on "https://<WS-NOC IP>:8443/oms1350/data/npr/nes" Rest API.
- Due to the catchup mechanism, the HDFS Avro files may contain duplicate records. If duplicate records are present, use the latest record.
- This program can be run inside the ext-pm-container where External Hadoop's namenode is installed and running. The generated CSV files are kept in the /nfmt/maintenance/pm-hadoop/pmdata/ folder inside the container.
- The /nfmt/maintenance/pm-hadoop/pmdata/ folder is accessible from External Hadoop VM as well, users can shift these files to their target servers/systems using "scp" or "sftp"