ABSTRACT

• SAS solutions run in Hadoop, providing customers with the best tools to transform data in Hadoop all the way to providing insights to make the right decisions for your business.
• Incorporating SAS products and solutions into your shared Hadoop cluster in a cooperative fashion with YARN to manage the resources is possible.
• Best practices and customer examples will be provided and discussed around how to build and manage a shared cluster with SAS applications and products.

HADOOP BACKGROUND

• Apache Hadoop – Open-Source software based on HDFS, YARN/MR
• Hadoop Environment – HDFS, YARN/MR, Hive, Pig, Spark, Impala, ZooKeeper, Oozie, etc
• Hadoop Distribution – Cloudera, Hortonworks, MapR, etc
• Hadoop - Cheap environment for distributed storage and distributed compute with linear expansion

HADOOP CONCERNS

• More User and More Services = Resource Contention = User Complaining
• Hadoop Solution = YARN
  • Fix the shared cluster
  • Resource Negotiator
  • Capabilities
    • Scheduling of resources
    • Containers to attempt resource restriction

YARN RM
16 GB Total Memory
1 GB Min Container size
1 shared queue

Beeline (Hive)
• 4 containers
• 2 GB per container

Spark-Submit (Spark)
• 2 containers
• 4 GB per container

YARN Default queue

YARN RM
8 GB Memory
1 - 4 GB
4 - 2 GB
2 - 2 GB

YARN NM
8 GB Memory
2 - 4 GB
3 - 2 GB
1 - 2 GB
SAS® and Hadoop Share Cluster Architecture

James Kochuba
SAS Senior Architect

SAS TECHNOLOGIES ON HADOOP

- SAS/ACCESS® Interfaces
- SAS® In-Database Analytics
- SAS® In-Memory Analytics
- SAS® Grid Computing

RESULTS (CLICK TO EDIT)

SAS® Grid/Server
libname hadoop hdfs (hdmd)
libname hadoop (hive)
libname hawq / impala
SAS reporting (SQL)
Scoring Accel calls
Code Accel calls
Data Quality Accel calls
HPA procs/LSAR

SAS® TKGrid
Hadoop
Hive
Hawq / Impala
HDMD

SAS In-memory (TKGrid / LASR)
SAS In-database (Embedded Process - EP)
SAS Access
Hadoop Cluster

SAN

High Speed Network
SAS® and Hadoop Share Cluster Architecture

James Kochuba
SAS Senior Architect

CONCLUSIONS

- YARN is the Hadoop answer to resource orchestration.
- SAS provides many technologies to run a SAS program that provide the best environment for the specific action.
- All of these technologies can run on a shared Hadoop environment and integrate with YARN as a central resource orchestrator.
- YARN is not perfect, and negotiation for resources during resource constrained times can be slow to change.
- The best tuning in the shared Hadoop environment for the YARN settings and OS will need to balance the software usage requirement and the business users’ expectations.
- The best practice is to first set up a simple resource orchestration that meets license requirements. Then slowly tune the settings to remove less-important items from taking resources, or reserve more resources for high-priority items. Leave the middle-priority items as is, because these will be extremely hard to modify with a resource negotiator.
- Remember, tools act differently when resources become constrained with a resource negotiator.

REFERENCES


