Chameleon
Expanding Open-Source Ambari for HPC

Korea Institute of Science and Technology Information
Jieun Choi
Motivation (Trend Perspective)

>>> HPC and Bigdata is converging

- BDEC (BigData and Exascale Computing)

Image Source: 2015 BDEC Pathways to Convergence Report
HPC and Bigdata is converging

- BDEC (BigData and Exascale Computing)
- Hadoop 3 roadmap

Hadoop 3 roadmap:

- Hadoop 3.0
  - 2017.12
  - Erasure Coding in HDFS

- Hadoop 3.1
  - 2018.04
  - GPU/Container support in YARN

- Hadoop 3.2
  - 2019.01
  - FPGA support in YARN

RDMA-based Apache Hadoop 3.x from High-Performance Big Data (HIBD) of the Ohio State University
Motivation (Trend Perspective)

- HPC and Bigdata is converging
  - BDEC (BigData and Exascale Computing)
  - Hadoop 3 roadmap
  - Intel’s unified cluster architecture for AI+HPC+HPDA (High Performance Data Analysis) workload

Genome analysis by Our scientist

- Some stages of data pipeline begin to support bigdata platform
  - Alignment: *BigBWA* (Hadoop), *SparkBWA* (Spark)
  - Variant Discovery: *GATKv4* (Spark)

- Computation performed on HPC cluster, data stored in LustreFS
HPC based bigdata platform operation management system

Expanding open-source Apache Ambari for HPC

- Hadoop on Lustre
- Web user-interface
- Advanced YARN app Monitoring
- HPC Resource Monitoring

- Lustre Installation
- Lustre Configuration
- Backup and Recovery
- LNET Configuration
- Multiple filesystem
1. Ambari Overview

2. LustreFS Overview

3. Chameleon
   - Lustre operation management
   - Hadoop on Lustre
   - Advanced YARN app monitoring
   - HPC resource monitoring
Ambari Overview
Apache Ambari is a 100% open-source platform for provisioning, managing, and monitoring Hadoop clusters.
Extension Points for custom service development

- Ambari view is a plugin that provides a way to connect custom functions to the web UI
- Ambari stack defines a set of everything needed to define services such as HDFS and YARN
## Terminology for custom service development

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>STACK</td>
<td>Defines a set of Services, where to obtain the software packages and how to manage the lifecycle.</td>
<td>HDP-2.3, HDP-2.6 * HDP: Hortonworks Data Platform)</td>
</tr>
<tr>
<td>SERVICE</td>
<td>Defines the Components that make-up the service.</td>
<td>HDFS, YARN,</td>
</tr>
<tr>
<td>COMPONENT</td>
<td>The building-blocks of a Service, that adhere to a certain lifecycle.</td>
<td>NAMENODE, DATANODE,</td>
</tr>
<tr>
<td>CATEGORY</td>
<td>The category of Component, that depends on lifecycle commands (Install, Configure, Status, Start, Stop)</td>
<td>MASTER, SLAVE, CLIENT</td>
</tr>
<tr>
<td>REPO</td>
<td>Repository metadata where the artifacts reside</td>
<td><a href="http://public-repo-1.hortonworks.com/">http://public-repo-1.hortonworks.com/</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td>HDP/centos6/2.x/GA/2.3.0.0</td>
</tr>
</tbody>
</table>

- Determine master/slave/client in terms of development
  - Client does not include ‘Start’ and ‘Stop’ in default lifecycle commands
  - Masters must be managed separately, and Slaves can run at once (ex. service start/stop)

Source: 2015 Hortonworks Technical Workshop
Chameleon, Ambari extension for HPC

Chameleon is an advanced extension of Ambari for provisioning, managing, and monitoring Hadoop clusters and Lustre Storage.
LustreFS Overview
LustreFS Overview

An open-source, object-based, distributed, parallel, clustered file system

- Highest I/O performance available for the world’s largest supercomputers
- Designed for maximum performance at massive scale
- Capable of Exascale capacities
- POSIX complaint
- Efficient and cost effective
- LNET (Lustre Network)
Difference between HDFS and LustreFS

POSIX-compliant

- Lustre uses a normal POSIX interface
- HDFS is not a POSIX-compliant file system

System-level VS. Application-level

- Lustre requires the use of a lustre-patched linux kernel

High-performance network interconnect

- Lustre uses Infiniband and provides own network protocol, which will be more efficient for bulk data transfer than the HDFS’s HTTP protocol
LustreFS and Chameleon

- Execute ambari-agent in both Lustre clients and Hadoop clusters
- Support for Lustre installation, configuration, and management

Image Source: Lustre & Intel Omni-Path Fabric SC16 Presentation
Chameleon
Chameleon

- Lustre Operation Management
Chameleon Components LustreFS Operation Management

- LustreKernelUpdater
- LustreManager
- AccountManager
- Hadoop on Lustre
- LustreManager View
- AccountManager View
LustreFS Management Service

Lustre Kernel Installation function (LustreKernelUpdater)
- Install lustre-patched kernel on MDS/OSS nodes
- Execute ‘reboot’ on MDS/OSS nodes

LustreFS Management function (LustreManager)
- LustreFS add/remove, mount/umount
- MDS, OSS node setup
- MDT, OST device setting
- OST active/deactive, backup/restore
- Lnet configuration, start/stop
This is a short video about LustreFS Configuration
https://www.youtube.com/watch?v=GzazkUWIs6M
Lustre Manager View (1/3)

**MDS Setting**

- Node: node09
- MDT device and size: /dev/lop
- IO network: erp003 (link/ether)

**OSS Setting**

- Node: node10
- OST device and size: /dev/lop
- IO network: erp003 (link/ether)
1 Lustre Manager View (2/3)

Client Setting

LNET Setting

Client Mount Point

 Lustre0

node04 CLIENT device and size

node05 CLIENT device and size

node06 CLIENT device and size

LNET Setting
OST Backup

OST Restore
Account Management Service

- Hadoop does not support strong authentication by default
  - Hadoop supports Kerberos for that, but, causes performance degradation

- Lustre depends on OS-level account management
  - Account information needs to be synchronized between Hadoop cluster and Lustre filesystem
Chameleon

- Lustre Operation Management
- Hadoop on Lustre
Comparison between HDFS and Lustre

HDFS
- NameNode
- DataNode
- Master
- Slave
- Resource Manager
- Node Manager
- MR app Master
- Map task
- Reduce task

Lustre
- Master
- Slave
- Resource Manager
- Node Manager
- MR app Master
- Map task
- Reduce task
- MDS/MDT
- OSS
- OST
- Shared storage

Aggregation of node-local storages
Related works for Hadoop-on-Lustre

- Xyratex
  - MapReduce Job shows theoretical performance gains on an appropriately designed Lustre based HPC cluster with Infiniband network

- Seagate’s lustrefs plugin
  - Is suitable for single tenancy
  - Needs kerberos to provide multi-tenancy

- Intel’s Enterprise Edition of Lustre
  - Eliminates ‘shuffle’ because Lustre is a shared file system

- We developed Hadoop-on-Lustre execution environment as custom Ambari service
Hadoop-on-Lustre execution environment

- Works for diskless cluster backed by Lustre
- Uses secure container configuration for multi-tenancy
- Uses Lustre as if it were a local directory for intermediate data

* mount --bind option
This is a short video about Hadoop-on-Lustre
https://www.youtube.com/watch?v=QvXcTf0ddA0
## LustreFS Mount

### Before mount

```
[root@node04 ~]# df
Filesystem  1K-blocks  Used  Available  Use% Mounted on
/dev/mapper/cl-root  38770180  7113908  31656272  19% /
devtmpfs  8123228     0     8123228   0% /dev
tmpfs  8133992     12     8133980   1% /dev/shm
tmpfs  8133992  112180  8021812   2% /run
tmpfs  8133992     0     8133992   0% /sys/fs/cgroup
/dev/sdal  1038336  141092  897244   14% /boot
tmpfs  1626800     0     1626800   0% /run/user/1000
tmpfs  1626800     0     1626800   0% /run/user/0
```

### After mount

```
[root@node04 ~]# df
Filesystem  1K-blocks  Used  Available  Use% Mounted on
/dev/mapper/cl-root  38770180  7113832  31656348  19% /
devtmpfs  8123228     0     8123228   0% /dev
tmpfs  8133992     12     8133980   1% /dev/shm
tmpfs  8133992  112404  8021588   2% /run
tmpfs  8133992     0     8133992   0% /sys/fs/cgroup
/dev/sdal  1038336  141092  897244   14% /boot
tmpfs  1626800     0     1626800   0% /run/user/1000
tmpfs  1626800     0     1626800   0% /run/user/0
192.168.0.118@tcp:/lustre0  5848464  2435104  3000488  45% /lustre0
```
Run Hadoop Wordcount (1/2)

**HDFS**

```
[root@node04 ~]# hadoop fs -ls /
Found 7 items
drwxrwxrwx  - yarn  hadoop  0 2019-06-20 01:53 /app-logs
drwxr-xr-x  - yarn  hadoop  0 2019-06-20 01:51 /ats
drwxr-xr-x  - hdfs  hdfs   0 2019-06-20 01:51 /hdp
drwxr-xr-x  - mapred hdfs   0 2019-06-20 01:51 /mapred
drwxrwxrwx  - mapred hadoop 0 2019-06-20 01:51 /mr-history
drwxr-xr-x  - hdfs  hdfs   0 2019-06-20 01:52 /tmp
drwxr-xr-x  - hdfs  hdfs   0 2019-06-20 01:51 /user
```

**LustreFS**

```
[root@node04 ~]# hadoop fs -ls /
19/08/06  05:55:06 INFO lustrefs.LustreFileSystem: Configuring LustreFS
19/08/06  05:55:06 INFO lustrefs.LustreFileSystem: Initializing LustreFS.
19/08/06  05:55:06 INFO lustrefs.LustreFileSystem: Configuring LustreFS
19/08/06  05:55:06 INFO lustrefs.LustreVolume: Root of Lustre file system is /lustre0/hadoop
19/08/06  05:55:06 INFO lustrefs.LustreVolume: Working directory is : lustrefs:/lustre0/hadoop/user/root
19/08/06  05:55:06 INFO lustrefs.LustreVolume: Write buffer size : 131072
```
Run Hadoop Wordcount (2/2)

HDFS

```bash
[user1@node04 ~]$ hadoop jar /usr/hdp/2.6.3.0-235/hadoop-mapreduce/hadoop-mapreduce-examples.jar wordcount input wordcount_test3
19/07/29 00:20:16 INFO  lustrefs.LustreFilesystem: Configuring LustreFS
19/07/29 00:20:16 INFO lustrefs.LustreFileSystem: Initializing LustreFS.
19/07/29 00:20:16 INFO lustrefs.LustreFileSystem: Configuring LustreFS
19/07/29 00:20:16 INFO lustrefs.LustreVolume: Root of Lustre file system is /lustre0/hadoop
19/07/29 00:20:16 INFO lustrefs.LustreVolume: Working directory is: lustrefs:/lustre0/hadoop/user/user1
19/07/29 00:20:16 INFO lustrefs.LustreVolume: Write buffer size : 131072
19/07/29 00:20:17 INFO client.RMProxy: Connecting to ResourceManager at node05/192.168.0.74:8050
```

LustreFS

```bash
[user1@node04 ~]$ hadoop jar /usr/hdp/2.6.3.0-235/hadoop-mapreduce/hadoop-mapreduce-examples.jar wordcount _libjars /usr/hdp/current/hadoop-yarn-nodemanager/lib/lustrefs-hadoop-0.9.1.jar input wordcount_test
19/07/29 00:20:16 INFO lustrefs.LustreFilesystem: Configuring LustreFS
19/07/29 00:20:16 INFO lustrefs.LustreFileSystem: Initializing LustreFS.
19/07/29 00:20:16 INFO lustrefs.LustreFileSystem: Configuring LustreFS
19/07/29 00:20:16 INFO lustrefs.LustreVolume: Root of Lustre file system is /lustre0/hadoop
19/07/29 00:20:16 INFO lustrefs.LustreVolume: Working directory is: lustrefs:/lustre0/hadoop/user/user1
19/07/29 00:20:16 INFO lustrefs.LustreVolume: Write buffer size : 131072
19/07/29 00:20:17 INFO client.RMProxy: Connecting to ResourceManager at node05/192.168.0.74:8050
```
Chameleon

- Lustre Operation Management
- Hadoop on Lustre
- Advanced YARN app. Monitoring
Dynamic metrics management is required

- Hadoop ResourceManager REST APIs can only collect predefined limited information
- Metrics and time interval should be configurable as needed

There is lots of Linux performance monitoring tools from Linux ecosystem
YARN Application Monitoring Service

- **Time-series data monitoring**
  - Uses TimeScale DataBase
  - Provides application history data management

- **User-defined metrics management**
  - All of Linux monitoring utilities are available (process-id based)
  - User can add/delete/modify metrics dynamically through MetricRegistry view
  - When updated, collector script is sent to all the client nodes and registered in crontab

- **Visualization**
  - Shows the graph per node or per application
Open-Source time-series database optimized for fast ingest and complex queries

‘Hypertable’ concept enables scalable data management for time-series data
- TimescaleDB automatically splits each hypertable into chunks
- Each chunk is implemented using a standard database table

Full SQL interface for all SQL natively supported by PostgreSQL
Many people use the NoSQL database as their time-series database engine.

Relational databases include: MySQL, MariaDB, Server, PostgreSQL. NoSQL databases include: Elastic, InfluxDB, MongoDB, Cassandra, Couchbase, Graphite, Prometheus, ClickHouse, OpenTSDB, DallmatinerDB, KairosDB, RiekTS. Source: https://www.percona.com/blog/2017/02/10/percona-blog-poll-database-engine-using-store-time-series-data/
But, We use **TimeScaleDB** instead of NoSQL database engines

- Outperforms in terms of high-cardinality datasets

Data management structure

Alter Table

- New ‘write-io-usage’ metric can be added
- Causes sparse data

<table>
<thead>
<tr>
<th>timestamp</th>
<th>appid</th>
<th>hostname</th>
<th>containerid</th>
<th>pid</th>
<th>cpu-util</th>
<th>mem-usage</th>
<th>write-io-usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>NULL</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Our Approach

- After adding ‘write-io-usage’, a table is newly created
- Changes table name for tracking chronological order:
  0.apptable.20190101 ➔ 1.apptable.20190131
- Causes dense data
Metric Registry View

Registered metrics:

- pidstat.cpu
- pidstat.mem
- proc.write-io
- proc.read-io

Modify metric

- Name: pidstat.cpu
- Description: Collect cpu usage by pidstat
- PID Symbol: [[pid]]
- Unit Label: CPU usage (%)

Parser script:

```
pidstat -p [[pid]] 1 1 | awk '$1 ~ /^Average/ {print $7}'
```

pidstat -l -p [[pid]] 1 1 | awk '$1 ~ /^Average/ {print $7}'
Yarn Application Monitor (1)
Yarn Application Monitor (2)

per application
Chameleon

- Lustre Operation Management
- Hadoop on Lustre
- Advanced YARN app. Monitoring
- HPC resource monitoring
HPC Resources Monitoring

Provides HPC monitoring information through web UI
- Lustre Filesystem (MDS, OSS)
- GPU, Infiniband (on-going)

Shows compute partition and storage partition, respectively
- Per-node graph
- Multiple nodes summary graph

Provides dynamic node alignment
- User can freely move each node on each partition
- Node utilization including health check is represented in color

Remote access to the node via WebSSH
4 LustreFS Monitoring

![LustreFS Monitoring Diagram](image)
Dynamic Node Alignment
4 System Information – per node
System Information - Multiple Nodes Summary
Remote Console
Conclusion
HPC and big data convergence makes the distinction between data analytics and computational science's ecosystem disappear.

Chameleon is a Bigdata platform operation management system considering HPC environment.

Chameleon helps to merge Hadoop ecosystem and LustreFS.

- Full video has been uploaded to YouTube
- We will soon release the chameleon source code to git hub