Using Lustre and SLURM to process Hadoop and Spark workloads and extending to the WLCG

Daniel Traynor, Terry Froy, School of Physics and Astronomy, Queen Mary University of London

Using our existing Lustre file system and SLURM batch system we used the magpie scripts to run Hadoop and Spark workloads and demonstrate that jobs can be run using standard grid commands.

The Problem: Hadoop/Spark

Hadoop is a data distribution and processing framework made up of several tools (Yarn, HDFS, Mapreduce) and a ecosystem of related projects (Pig/Hive/Storm/Spark). The Hadoop ecosystem is hugely popular in the commercial world and is under active development.

With an increasing use of Hadoop/Spark in HEP and need to serve new communities beyond HEP. How can we provide a service without wasting resources?

Can't afford to run dedicated cluster for Hadoop ecosystem

Don't have the money to use a commercial cloud solution

Will have to try and integrate Hadoop and Spark into existing infrastructure

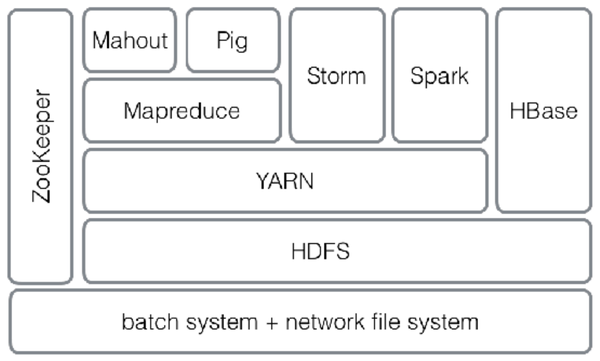
The Solution: Magpie

Magpie project contains scripts for running Big Data software in HPC environments. It supports running over any generic network filesystem (including Lustre) and several scheduler/resource manager (such as Slurm)

Magpie supported Hadoop ecosystem

Magpie is primarily developed by Al chu at LLNL

(https://github.com/LLNL/magpie)



Setup and Use

0) Hadoop/Spark and Magpie software must be available on every node in a cluster. This could be done via NFS or even CVMFS

1) A user sets a number of variables at job submission in magpie_hadoop_script.sh which define the parameters of the Hadoop cluster. This script runs at job start up.

2) The job is started by the local batch system. It then launches daemons on all reserved nodes creating a Hadoop cluster as requested. Then the user script is run

3) With the hadoop cluster up and running data can be copied into HDFS is needed.

4) Final results are then copied back to the user directly or via grid SE

example user job steps

```
voms-proxy-init --voms dteam
glite-ce-job-submit -a -o jobIDfile2 -r ce08.esc.qmul.ac.uk:8443/cream-slurm-magpie magpi_hadoop_script.jdl
glite-ce-job-status -i jobIDfile2
glite-ce-job-output https://ce08.esc.qmul.ac.uk:8443/CREAMXXXXX
```

example CE submission jdl

```
Executable = "magpie_hadoop_script.sh";
inputSandbox = {  "magpie_hadoop_script.sh","my-job-script.sh" };
StdOutput = "hello.out";
StdError = "hello.err";
OutputSandbox = {"hello.out","hello.err"};
outputsandboxbasedesturi = "gsiftp://localhost";
CpuNumber= 24;
HostNumber=3;
WholeNodes=true;
```

example magpie startup script

```
export JAVA_HOME="/usr/lib/jvm/jre/"
export MAGPIE_SUBMISSION_TYPE="sbatchsrun"
export MAGPIE_SCRIPTS_HOME="/opt/shared/magpie"
export MAGPIE_LOCAL_DIR="${LOCAL_HOME}/magpie"
export MAGPIE_JOB_TYPE="hadoop"
export HADOOP_SETUP=yes
export HADOOP_SETUP_TYPE="MR2"
export HADOOP_VERSION="2.8.4"
export HADOOP_HOME="/opt/shared/hadoop-${HADOOP_VERSION}"
export HADOOP_LOCAL_DIR="${LOCAL_HOME}/hadoop"
export HADOOP_MODE="script"
export HADOOP_FILESYSTEM_MODE="hdfsoverlustre"
export HADOOP_HDFS_REPLICATION=1
export HADOOP_HDFS_PATH_CLEAR="yes"
export HADOOP_HDFSOVERLUSTRE_PATH="/mnt/lustre/hadoop/"
export HADOOP_HDFSOVERLUSTRE_REMOVE_LOCKS=yes
export HADOOP_PER_JOB_HDFS_PATH="yes"
export HADOOP_SCRIPT_PATH="${LOCAL_HOME}/my-job-script.sh"
srun --no-kill -W 0 $MAGPIE_SCRIPTS_HOME/magpie-check-inputs
srun --no-kill -W 0 $MAGPIE_SCRIPTS_HOME/magpie-setup-core
srun --no-kill -W 0 $MAGPIE_SCRIPTS_HOME/magpie-setup-projects
srun --no-kill -W 0 $MAGPIE_SCRIPTS_HOME/magpie-setup-post
srun --no-kill -W 0 $MAGPIE_SCRIPTS_HOME/magpie-pre-run
srun --no-kill -W 0 $MAGPIE_SCRIPTS_HOME/magpie-run
srun --no-kill -W 0 $MAGPIE_SCRIPTS_HOME/magpie-cleanup
srun --no-kill -W 0 $MAGPIE_SCRIPTS_HOME/magpie-post-run
```

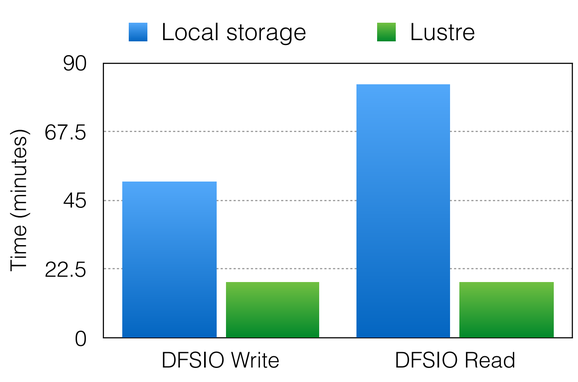
example user job script

```
yarn jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.8.3.jar teragen -Dmapreduce.jobs.maps=1000 100000000 random-data
yarn jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.8.3.jar terasort random-data sorted-data
yarn jar /usr/local/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.8.3.jar teravalidate sorted-data report
```

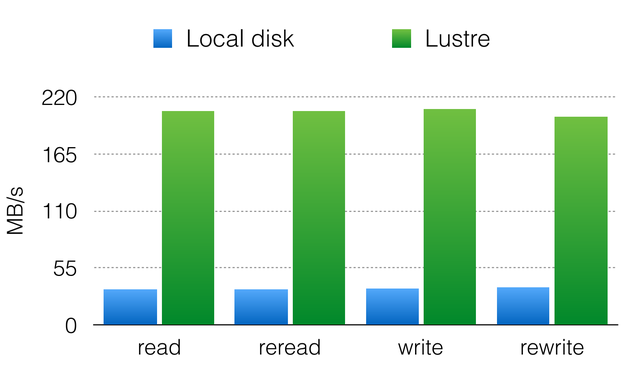
Optimisations

Running Hadoop (HDFS) over Lustre is quicker than using local disk for small number of servers.

Hadoop DFSIO benchmark using one name node and two data nodes. local disks are mode up of 3 1TB hard drives per node. Lustre uses the existing Lustre storage system.



IOZONE test showing higher throughput with Lustre than with three nodes using 3 local disk per node.



Limitations

- **User submitting jobs must know local details about the site they are using (mount points, software paths, etc…**
- **Requesting whole nodes only works at present, requesting less ends up creating one data node per core with significant performance impact**
- **The Lustre plugin for Hadoop (https://github.com/whamcloud/lustre-connector-for-hadoop) was not able to worth with the version of Lustre we use (2.8).**
- **The Hadoop ecosystem is hugely popular in the commercial world. Still limited use in HEP. However rapid development ongoing for ROOT Spark plugin. https://github.com/diana-hep/spark-root**
- **While in principle running Hadoop/Spark jobs on grid / HTC clusters works the service is some distance away from a production service. In order two devlop a production service we need a clear use case to support and test and tune the service.**

Links:

Hadoop: http://hadoop.apache.org/

Spark: https://spark.apache.org/

Magpie: https://github.com/LLNL/magpie

CHEP2012: Scalable Petascale Storage for HEP using Lustre: Journal of Physics: C.J. Walker D.P. Traynor and A.J. Martin. Conference Series 396 (2012) 042063

CHEP2016: Upgrading and Expanding Lustre Storage for use with the WLCG: Journal of Physics: D.P. Traynor, C.J. Walker, T. Froy. Conference Series 898 (2017) 062004

IOzone: http://www.iozone.org/

Contacts:

Daniel Traynor: d.traynor@qmul.ac.uk

Terry Froy: t.froy@qmul.ac.uk

School of Physics and Astronomy, Queen Mary University of London, Mile End Road, London, E1 4NS

GridPP UK Computing for Particle Physics

Queen Mary University of London

Science & Technology Facilities Council