**Hanythingon demand – Hadoop clusters on HPC clusters**

*FOSDEM, 31 January 2016*

**Ewan Higgs**

DICT - UGent, VSC

ewan.higgs@ugent.be

Ewan Higgs
Big Data Coordinator for HPC
Ghent University
Agenda

- What is hanythingondemand?
- Why have we made hanythingondemand?
- Wade into hod (no deep dives)
- Use cases
- Developer things
HOD – Hanythingondemand

- https://github.com/hpcugent/hanythingondemand

Run a Hadoop cluster in our HPC clusters

Extensive good documentation:

- https://hod.readthedocs.org


hod create – Create a new cluster.
hod connect – Connect to your cluster.
hod batch – create a new cluster to run a script.
hod list – list your clusters.
Create a new cluster

$ hod create --label mycluster -n 4 --dist Hadoop-2.6.0-cdh5.4.5-native

Connect to the cluster

$ hod connect mycluster

Run jobs on the cluster:

$ yarn jar wordcount.jar WordCount wordcount/input wordcount/output
Create a new cluster and run a script

```
$ hod batch --dist Hadoop-2.6.0-cdh5.4.5-native -n16 --script myscript.sh
```

Elide arguments with environment variables:

```
export HOD_BATCH_DIST=Hadoop-2.6.0-cdh5.4.5-native
hod batch -n16 --script myscript.sh
```
Create an IPython Notebook

$ hod create --dist IPython-notebook-3.2.3
-n2

To use, make an SSH tunnel to the head node and set a proxy.
Why not just buy a big data system?

Why not just go cloud?
HPC | Big Data

HPC
- Login Node
- Worker Node
- Worker Node
- Worker Node
- Scratch

Big Data
- Login Node
- Worker Node
- Worker Node
- Worker Node
- hdfs
European HPC

- Tier-0: Europe
- Tier-1: Regional and national
- Tier-2: University
- Tier-3: Desktop
<table>
<thead>
<tr>
<th></th>
<th>raichu</th>
<th>delcatty</th>
<th>phanpy</th>
<th>golett</th>
<th>swalot</th>
</tr>
</thead>
<tbody>
<tr>
<td># nodes</td>
<td>64</td>
<td>158</td>
<td>16</td>
<td>200</td>
<td>128</td>
</tr>
<tr>
<td># cores</td>
<td>1024</td>
<td>2528</td>
<td>384</td>
<td>4800</td>
<td>2560</td>
</tr>
<tr>
<td>Interconnect</td>
<td>Ethernet</td>
<td>Infiniband FDR</td>
<td>Infiniband FDR</td>
<td>Infiniband FDR-10</td>
<td>Infiniband FDR</td>
</tr>
<tr>
<td>CPU</td>
<td>Intel Xeon Sandy Bridge</td>
<td>Intel Xeon Sandy Bridge</td>
<td>Intel Xeon Haswell</td>
<td>Intel Xeon Haswell</td>
<td>Intel Xeon Haswell</td>
</tr>
<tr>
<td>Clock (GHz)</td>
<td>2.6</td>
<td>2.6</td>
<td>2.5</td>
<td>2.5</td>
<td>2.6</td>
</tr>
<tr>
<td>Memory per node (GiB)</td>
<td>32</td>
<td>64</td>
<td>512</td>
<td>64</td>
<td>128</td>
</tr>
</tbody>
</table>

**HPC-UGent: STEVIN infrastructure**
<table>
<thead>
<tr>
<th></th>
<th>muk (Ghent)</th>
<th>Tier1b (Leuven)</th>
<th>Swalot (Tier2)</th>
</tr>
</thead>
<tbody>
<tr>
<td># nodes</td>
<td>528</td>
<td>580</td>
<td>128</td>
</tr>
<tr>
<td># cores</td>
<td>8448</td>
<td>16240</td>
<td>2560</td>
</tr>
<tr>
<td>Interconnect</td>
<td>Infiniband FDR</td>
<td>Infiniband EDR</td>
<td>Infiniband FDR</td>
</tr>
<tr>
<td>CPU</td>
<td>Intel Xeon Sandy Bridge</td>
<td>Intel Xeon Broadwell</td>
<td>Intel Xeon Haswell</td>
</tr>
<tr>
<td>Clock (GHz)</td>
<td>2.6</td>
<td>3.2</td>
<td>2.6</td>
</tr>
<tr>
<td>Memory per node (GiB)</td>
<td>64</td>
<td>128/256</td>
<td>128</td>
</tr>
<tr>
<td>Installed</td>
<td>2013</td>
<td>2016</td>
<td>2016</td>
</tr>
</tbody>
</table>
Applications Run Natively IN Hadoop

- BATCH (MapReduce)
- INTERACTIVE (Tez)
- ONLINE (HBase)
- STREAMING (Storm, S4,...)
- GRAPH (Giraph)
- IN-MEMORY (Spark)
- HPC MPI (OpenMPI)
- OTHER (Search) (Weave...)

YARN (Cluster Resource Management)

HDFS2 (Redundant, Reliable Storage)
Applications Run Natively **IN** Hadoop

- **BATCH** (MapReduce)
- **INTERACTIVE** (Tez)
- **ONLINE** (HBase)
- **STREAMING** (Storm, S4, ...)
- **GRAPH** (Giraph)
- **IN-MEMORY** (Spark)
- **HPC MPI** (OpenMPI)
- **OTHER** (Search) (Weave...)

**YARN** (Cluster Resource Management)

**GPFS**

**PBS**
Disk-Locality in Datacenter Computing Considered Irrelevant

Ganesh Ananthanarayanan, Ali Ghodsi, Scott Shenker, Ion Stoica
University of California, Berkeley
\{ganesha, alig, shenker, istoica\}@cs.berkeley.edu

1 Introduction

Data center computing is becoming pervasive in many organizations. Computing frameworks such as MapReduce [17], Hadoop [6] and Dryad [25], split jobs into small tasks that are run on the cluster's compute nodes. Through these frameworks, computation can be performed on large datasets in a fault-tolerant way, while hiding the complexities of the distributed nature of the cluster. For these reasons, a considerable work has been done to improve the efficiency of these frameworks.

over-subscription. Such topologies have already been adopted in several datacenters [12]. This ensures that bandwidth across racks will be equal to the bandwidth within a rack.

Another trend that strengthens our thesis is the need to save more and more data in clusters. The need for storage space outweighs affordable storage, and the gap is projected to continue to expand [2]. The ever-increasing demand for storage not only makes solid state devices (SSDs) economically infeasible to deploy as a primary storage medium [5, 9, 16], but more importantly has led
But locality of data...

...as much as 90% most jobs fit on a single node based on a report from Cloudera’s (n.b.: data from 2012).
Wade into HOD
What's a dist?

```
$ ls etc/hod/IPython-notebook-3.2.3/
hod.conf  nodemanager.conf  screen.conf
ipython.conf  resourcemanager.conf  start-notebook.sh
```
{-#* cdf -*-
# vim: ft=cfg

[Meta]
version=1

[Config]
# The start-notebook.sh requires IPython, matplotlib, and Spark to be loaded.
# These are set here:
modules=Hadoop/2.6.0-cdh5.4.5-native,Spark/1.6.0,IPython/3.2.3-intel-2015b-
Python-2.7.10,matplotlib/1.4.3-intel-2015b-Python-2.7.10
master_env=HADOOP_HOME,EBR00THADOOP,JAVA_HOME,PYTHONPATH
services=resourceManager.conf,nodemanager.conf,ipython.conf,screen.conf
config_writer=hod.config.writer.hadoop_xml
# Point the workdir to a path on the parallel file system using the command
# line named argument: --config-workdir=...
#workdir=
autogen=ipython_notebook
directories=${localworkdir}/dfs/name,${localworkdir}/dfs/data
~
~
#*- ccf -*
# vim: ft=cfg
[Unit]
Name=nodemanager
RunsOn=all

[Service]
# note: The format is not a daemon since we wait for it to complete.
ExecStart=$$EBRO0THADOOP/sbin/yarn-daemon.sh start nodemanager
ExecStop=$$EBRO0THADOOP/sbin/yarn-daemon.sh stop nodemanager

[Environment]
YARN_NICENESS=1 /usr/bin/ionice -c2 -n0
HADOOP_CONF_DIR=$localworkdir/conf
YARN_LOG_DIR=$localworkdir/log
YARN_PID_DIR=$localworkdir/pid
~
Auto generated configurations
```yaml
# -*- cfg -*-
# vim: ft=cfg
[Meta]
version=1

[Config]
modules=Hadoop/2.6.0-cdh5.4.5-native,Spark/1.5.0,IPython/3.2.1-intel-2015a-Python-2.7.10,matplotlib/1.4.3-intel-2015a-Python-2.7.10
master_env=HADOOP_HOME,EBROOTHADOOP,JAVA_HOME,PYTHONPATH
services=resourcemanager.conf,nodemanager.conf,ipython.conf,screen.conf
config_writer=hod.config.writer.hadoop_xml
# Point the workdir to a path on the parallel file system using the command
# line named argument: --config-workdir=...
# workdir=
# autogen=ipython_notebook
autogen=ipython_notebook
directories=${localworkdir}/dfs/name,${localworkdir}/dfs/data

[yarn-site.xml]
yarn.app.mapreduce.am.resource.cpu-vcores=1
yarn.scheduler.maximum-allocation-vcores=16
yarn.scheduler.minimum-allocation-vcores=1
yarn.nodemanager.resource.cpu-vcores=16
~
~
```
User Stories
Halvade: scalable sequence analysis with MapReduce

Dries Decap¹,⁵, Joke Reumers²,⁵, Charlotte Herzeel³,⁵, Pascal Costanza⁴,⁵ and Jan Fostier¹,⁵,*

¹Department of Information Technology, Ghent University - iMinds, Gaston Crommenlaan 8 bus 201, 9050 Ghent, Belgium, ²Janssen Research & Development, a division of Janssen Pharmaceutica N.V., 2340 Beerse, Belgium, ³Imec, Kapeldreef 75, 3001 Leuven, Belgium, ⁴Intel Corporation Belgium and ⁵ExaScience Life Lab, Kapeldreef 75, 3001 Leuven, Belgium.

Associate Editor: Prof. Gunnar Ratsch

ABSTRACT

Motivation: Post-sequencing DNA analysis typically consists of read mapping followed by variant calling. Especially for whole genome sequencing, this computational step is very time-consuming, even when using multithreading on a multi-core machine.

Results: We present Halvade, a framework that enables sequencing pipelines to be executed in parallel on a multi-node and/or multi-core computer for an overview. Especially for whole genome sequencing, applying such tools is a computational bottleneck. To illustrate this, we consider the recently proposed Best Practices pipeline for DNA sequencing analysis (Van der Auwera et al., 2013) that consists of the Burrow-Wheeler Aligner (BWA) (Li and Durbin, 2009) for the alignment step, Picard (http://picard.sourceforge.net) for data preparation and the Genome Analysis Toolkit (GATK) (McKenna
Table 2. Runtime as a function of the number of parallel tasks (mappers/reducers) on the Intel Big Data cluster and Amazon EMR. The time for uploading data to S3 over the internet is not included in the runtimes for Amazon EMR.

<table>
<thead>
<tr>
<th>Cluster</th>
<th># worker nodes</th>
<th># parallel tasks</th>
<th># CPU cores</th>
<th>runtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel Big Data cluster</td>
<td>1</td>
<td>3</td>
<td>18</td>
<td>47h 59min</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>15</td>
<td>90</td>
<td>9h 54min</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>31</td>
<td>186</td>
<td>4h 50min</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>59</td>
<td>354</td>
<td>2h 39min</td>
</tr>
<tr>
<td>Amazon EMR</td>
<td>1</td>
<td>4</td>
<td>32</td>
<td>38h 38min</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>8</td>
<td>64</td>
<td>20h 19min</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>16</td>
<td>128</td>
<td>10h 20min</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>32</td>
<td>256</td>
<td>5h 13min</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>64</td>
<td>512</td>
<td>2h 44min</td>
</tr>
</tbody>
</table>
Code

Limitations

Community
https://github.com/hpcugent/hanythingondemand

• Python 2.7
• GPL v2
• ~80% code coverage
• Jenkins builds
Limitations

• Only PBS/Torque

• Server coding in Python2 and without twisted
Would you like to use this at your site?

Are there any tools you need?

Do you need slurm or Grid Engine?
Wrapping up

• HOD lets HPC users use Hadoop ecosystem

• Auto generated configurations

• Being used for actual research

• HPC clusters can make good Big Data clusters

• Check it out!
Questions?

Further contact:
ewan.higgs@ugent.be