Running Hadoop in the Cloud

Tom White
tomwhite@apache.org
ApacheCon Europe 2009
Wednesday, 25 Mar 2009
About me

- Apache Hadoop Committer, PMC Member, Apache Member
- Employed by Cloudera
- Writing a book on Hadoop for O’Reilly
  - http://hadoopbook.com
Agenda

- Cloud Computing and Hadoop
- Hadoop on Amazon EC2
  - Storage options
  - Deployment options
- Demo
- Case Study
Cloud Computing


- “Cloud” = data center hardware and software
  - Public cloud – service sold to the public
  - Private cloud – internal to an organization

- Three new aspects (of public clouds)
  1. The illusion of infinite computing resources available on demand
  2. The elimination of an up-front commitment by Cloud users
  3. The ability to pay for use of computing resources on a short-term basis as needed
Why run Hadoop in the “public” cloud?

- “Infinite” resources
  - Hadoop scales linearly
- No upfront commitment
  - Try before you buy
  - Will Hadoop solve my problem?
- Pay as you go
  - Elasticity
  - Run a large cluster for a short time
  - Grow or shrink a cluster on demand
- **Lower administration costs and total cost of ownership**
Requirements for Hadoop

- Hardware
  - Hadoop needs lots of memory and disks
- Storage
  - Hadoop works best when storage is integrated with compute nodes
- Networking topology
  - Prefer control over placement of machines
  - Hadoop needs visibility into topology
- Bandwidth control
  - Dedicated switches are best
Hadoop on Amazon EC2
Hadoop on Amazon EC2

- **Hardware**
  - E.g. High-CPU XLarge:
    - 8 cores, 7GB memory, 1690 GB storage

- **Storage**
  - Choice of local disks, S3, EBS

- **Networking topology**
  - Finest granularity is “availability zone”

- **Bandwidth control**
  - Large instances have “high” I/O performance (no guarantees)
Storage options
Hadoop on EC2 with S3 storage

- Pros
  - Elastic
  - Use existing S3 data
  - Cheap
- Cons
  - No locality
  - Poor transfer speed
Hadoop on EC2 with local storage

Pros
- Data locality

Cons
- No rack locality
- Cluster is always on
Hadoop on EC2 with EBS storage

**Pros**
- Data locality
- Elastic

**Cons**
- No rack locality
- No pre-existing AMIs
## Economics

<table>
<thead>
<tr>
<th>Storage</th>
<th>Price (per GB month)</th>
<th>Transfer cost (within AWS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>EBS</td>
<td>$0.10</td>
<td>$0.10 per million I/O requests</td>
</tr>
<tr>
<td>S3</td>
<td>$0.15</td>
<td>$0.01 per thousand HTTP requests</td>
</tr>
<tr>
<td>EC2 local storage</td>
<td>$0.32</td>
<td></td>
</tr>
<tr>
<td>(reserved instance for one year)</td>
<td></td>
<td>$0.00</td>
</tr>
</tbody>
</table>

- EBS is a good fit for clusters with low–medium utilization
Deployment options
S3 Filesystems

- Hadoop Filesystem abstraction
  - HDFS (hdfs://), KFS (kfs://), local (file://)
- S3 Native Filesystem (s3n://)
  - 5GB file size
  - Use with existing S3 data and tools
- S3 Block Filesystem (s3://)
  - Unlimited file size
  - Can’t use with existing S3 data or tools
Hadoop S3 Examples

- Copy from HDFS to S3
  - `hadoop fs -cp hdfs://namenode/path s3n://bucket/path`

- Parallel copy from HDFS to S3
  - `hadoop distcp hdfs://namenode/path s3://bucket/path`

- Run MapReduce on S3 data
  - `hadoop jar hadoop-*-examples.jar grep \
    s3n://bucket/input s3n://bucket/output pattern`
Hadoop EC2 AMIs

- Public AMIs provided by Apache and Cloudera
  - Use the same launch scripts
    - http://wiki.apache.org/hadoop/AmazonEC2
    - http://www.cloudera.com/hadoop-ec2
  - Launch scripts do cluster coordination and configuration
- AMIs are easy to customize
  - Patches
  - Extra software
Apache Hadoop and Cloudera’s Distribution

- Apache hosts Hadoop development
  - Nightly builds
  - Releases
- Cloudera’s Distribution for Hadoop
  - Based on most recent stable version
  - Uses RPMs for deployment (other packages coming soon)
    - Linux Filesystem Hierarchy Standard
    - Standard Linux service management
    - Dependency management
Demo
# Your Amazon Account Number.
AWS_ACCOUNT_ID=

# Your Amazon AWS access key.
AWS_ACCESS_KEY_ID=

# Your Amazon AWS secret access key.
AWS_SECRET_ACCESS_KEY=

# Location of EC2 keys.
# The default setting is probably OK if you set up EC2 following the Amazon Getting Started guide.
EC2_KEYDIR="dirname "$EC2_PRIVATE_KEY"

# The EC2 key name used to launch instances.
# The default is the value used in the Amazon Getting Started guide.
KEY_NAME=tom

# Where your EC2 private key is stored (created when following the Amazon Getting Started guide).
# You need to change this if you don't store this with your other EC2 keys.
PRIVATE_KEY_PATH=`echo "$EC2_KEYDIR"/id_rsa_cloudera`

# SSH options used when connecting to EC2 instances.
SSH_OPTS=`echo -i "$PRIVATE_KEY_PATH" -o StrictHostKeyChecking=no -o ServerAliveInterval=30`

# The version of Hadoop to use. Note that this is the version of the AMI of Cloudera's Distribution for Hadoop here.
HADOOP_VERSION=0.3.0

# The Amazon S3 bucket where the Hadoop AMI is stored.
# The default value is for public images, so can be left if you are using running a public image.
# Change this value only if you are creating your own (private) AMI
# so you can store it in a bucket you own.
# [Changed to Cloudera bucket]
S3_BUCKET=cloudera-ec2-hadoop-images
launch-cluster tom-hadoop 10
Testing for existing master in group: tom-hadoop
Starting master with AMI ami-9136d1f8
Waiting for instance i-b3f16ada to start
........Started as ip-10-250-74-242.ec2.internal
Warning: Permanently added 'ec2-75-101-176-200.compute-1.amazonaws.com,75.101.176.200' (RSA) to the list of known hosts.
Copying private key to master
d_id_rsa.cloudera
Master is ec2-75-101-176-200.compute-1.amazonaws.com, ip is 75.101.176.200, zone is us-east-1c.
Adding tom-hadoop node(s) to cluster group tom-hadoop with AMI ami-9136d1f8
i-9bf16af2
i-9af16af3
i-9df16af4
i-9cf16af5
i-9ff16af6
i-9ef16af7
i-91f16af8
i-90f16af9
i-93f16afa
i-92f16afb
loy:cloudera-for-hadoop-on-ec2-0.3.0 tom$
[root@ip-10-250-74-242 ~]# hadoop distcp s3n://cloudera-datasets/wiki-articles/ wikipedia
09/03/20 10:44:05 INFO tools.DistCp: srcPaths=[s3n://cloudera-datasets/wiki-articles]
09/03/20 10:44:05 INFO tools.DistCp: destPath=wikipedia
09/03/20 10:44:09 INFO tools.DistCp: srcCount=302
09/03/20 10:44:12 INFO mapred.JobClient: Running job: job_200903201031_0001
09/03/20 10:44:13 INFO mapred.JobClient: map 0% reduce 0%
09/03/20 10:44:34 INFO mapred.JobClient: map 1% reduce 0%
09/03/20 10:44:49 INFO mapred.JobClient: map 3% reduce 0%
09/03/20 10:45:05 INFO mapred.JobClient: map 10% reduce 0%
ip-10-250-74-242 Hadoop Map/Reduce Administration

State: RUNNING
Started: Fri Mar 20 10:31:35 EDT 2009
Version: Unknown, rUnknown
Compiled: Unknown by Unknown
Identifier: 200903201031

Cluster Summary

<table>
<thead>
<tr>
<th>Maps</th>
<th>Reduces</th>
<th>Total Submissions</th>
<th>Nodes</th>
<th>Map Task Capacity</th>
<th>Reduce Task Capacity</th>
<th>Avg. Tasks/Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>10</td>
<td>20</td>
<td>10</td>
<td>3.00</td>
</tr>
</tbody>
</table>

Running Jobs

Running Jobs

none

Completed Jobs

<table>
<thead>
<tr>
<th>Jobid</th>
<th>User</th>
<th>Name</th>
<th>Map % Complete</th>
<th>Map Total</th>
<th>Maps Completed</th>
<th>Reduce % Complete</th>
<th>Reduce Total</th>
<th>Reduces Completed</th>
</tr>
</thead>
<tbody>
<tr>
<td>job_200903201031_0001</td>
<td>root</td>
<td>distcp</td>
<td>100.00%</td>
<td>75</td>
<td>75</td>
<td>100.00%</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Failed Jobs

Failed Jobs

none
[root@ip-10-250-74-242 hadoop]# hadoop jar hadoop-0.18.3-examples.jar grep wikipedia/wiki-articles grep-out '(hadoop|pig)'
09/03/20 11:18:52 INFO mapred.FileInputFormat: Total input paths to process : 300
09/03/20 11:18:52 INFO mapred.FileInputFormat: Total input paths to process : 300
09/03/20 11:18:52 INFO mapred.JobClient: Running job: job_200903201031_0006
09/03/20 11:18:53 INFO mapred.JobClient: map 0% reduce 0%
09/03/20 11:19:01 INFO mapred.JobClient: map 1% reduce 0%
09/03/20 11:19:02 INFO mapred.JobClient: map 3% reduce 0%
09/03/20 11:19:03 INFO mapred.JobClient: map 5% reduce 0%
09/03/20 11:19:04 INFO mapred.JobClient: map 6% reduce 0%
09/03/20 11:19:11 INFO mapred.JobClient: map 8% reduce 0%
09/03/20 11:19:12 INFO mapred.JobClient: map 9% reduce 0%
09/03/20 11:19:13 INFO mapred.JobClient: map 10% reduce 0%
09/03/20 11:19:14 INFO mapred.JobClient: map 11% reduce 0%
09/03/20 11:19:15 INFO mapred.JobClient: map 12% reduce 0%
09/03/20 11:19:18 INFO mapred.JobClient: map 13% reduce 0%
09/03/20 11:19:19 INFO mapred.JobClient: map 13% reduce 1%
09/03/20 11:19:20 INFO mapred.JobClient: map 13% reduce 2%
09/03/20 11:19:21 INFO mapred.JobClient: map 14% reduce 2%
09/03/20 11:19:22 INFO mapred.JobClient: map 16% reduce 3%
09/03/20 11:19:23 INFO mapred.JobClient: map 17% reduce 3%
09/03/20 11:19:24 INFO mapred.JobClient: map 18% reduce 3%
09/03/20 11:19:25 INFO mapred.JobClient: map 18% reduce 4%
09/03/20 11:19:26 INFO mapred.JobClient: map 19% reduce 4%
09/03/20 11:19:28 INFO mapred.JobClient: map 20% reduce 4%
09/03/20 11:19:30 INFO mapred.JobClient: map 21% reduce 4%
09/03/20 11:19:32 INFO mapred.JobClient: map 22% reduce 5%
09/03/20 11:19:33 INFO mapred.JobClient: map 23% reduce 5%
09/03/20 11:19:34 INFO mapred.JobClient: map 25% reduce 5%
09/03/20 11:19:35 INFO mapred.JobClient: map 26% reduce 5%
**Hadoop job** _200903201031_0006 on **ip-10-250-74-242**

- **User:** root
- **Job Name:** grep-search
- **Status:** Running
- **Started at:** Fri Mar 20 11:18:52 EDT 2009
- **Running for:** 46sec

### Task Status

<table>
<thead>
<tr>
<th>Kind</th>
<th>% Complete</th>
<th>Num Tasks</th>
<th>Pending</th>
<th>Running</th>
<th>Complete</th>
<th>Killed</th>
<th>Failed/Killed Task Attempts</th>
</tr>
</thead>
<tbody>
<tr>
<td>map</td>
<td>27.98%</td>
<td>300</td>
<td>201</td>
<td>16</td>
<td>83</td>
<td>0</td>
<td>0 / 0</td>
</tr>
<tr>
<td>reduce</td>
<td>6.94%</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>0 / 0</td>
</tr>
</tbody>
</table>

### System Counters

<table>
<thead>
<tr>
<th>Counter</th>
<th>Map</th>
<th>Reduce</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Systems</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HDFS bytes read</td>
<td>5,407,561,922</td>
<td>0</td>
<td>5,407,561,922</td>
</tr>
<tr>
<td>Local bytes written</td>
<td>24,819</td>
<td>0</td>
<td>24,819</td>
</tr>
<tr>
<td>Job Counters</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Launched reduce tasks</td>
<td>0</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>Launched map tasks</td>
<td>0</td>
<td>0</td>
<td>99</td>
</tr>
<tr>
<td>Data-local map tasks</td>
<td>0</td>
<td>0</td>
<td>99</td>
</tr>
<tr>
<td>Reduce input groups</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Combine output records</td>
<td>88</td>
<td>0</td>
<td>88</td>
</tr>
</tbody>
</table>
[root@ip-10-250-74-242 hadoop]# hadoop fs -cat grep-out/part-00000
75524  pig
47      hadoop
[root@ip-10-250-74-242 hadoop]#
Running Hadoop instances:

<table>
<thead>
<tr>
<th>INSTANCE</th>
<th>ami-9136d1f8</th>
<th>ec2-75-101-176-200.compute-1.amazonaws.com</th>
<th>ip-10-250-74-242.ec2.internal</th>
<th>ru</th>
</tr>
</thead>
<tbody>
<tr>
<td>i-b3f16ada</td>
<td>0</td>
<td>c1.medium</td>
<td>2009-03-20T14:30:02+0000</td>
<td>us-east-1c</td>
</tr>
<tr>
<td>INSTANC</td>
<td>i-b9f16af2</td>
<td>am</td>
<td>0</td>
<td>c1.medium</td>
</tr>
<tr>
<td>INSTANC</td>
<td>i-9af16af3</td>
<td>am</td>
<td>0</td>
<td>c1.medium</td>
</tr>
<tr>
<td>INSTANC</td>
<td>i-9df16af4</td>
<td>am</td>
<td>0</td>
<td>c1.medium</td>
</tr>
<tr>
<td>INSTANC</td>
<td>i-9cf16af5</td>
<td>am</td>
<td>0</td>
<td>c1.medium</td>
</tr>
<tr>
<td>INSTANC</td>
<td>i-9f16af6</td>
<td>am</td>
<td>0</td>
<td>c1.medium</td>
</tr>
<tr>
<td>INSTANC</td>
<td>i-9ef16af7</td>
<td>am</td>
<td>0</td>
<td>c1.medium</td>
</tr>
<tr>
<td>INSTANC</td>
<td>i-91f16af8</td>
<td>am</td>
<td>0</td>
<td>c1.medium</td>
</tr>
<tr>
<td>INSTANC</td>
<td>i-90f16af9</td>
<td>am</td>
<td>0</td>
<td>c1.medium</td>
</tr>
<tr>
<td>INSTANC</td>
<td>i-93f16afa</td>
<td>am</td>
<td>0</td>
<td>c1.medium</td>
</tr>
<tr>
<td>INSTANC</td>
<td>i-92f16afb</td>
<td>am</td>
<td>0</td>
<td>c1.medium</td>
</tr>
<tr>
<td>INSTANC</td>
<td>i-91f16af2</td>
<td>am</td>
<td>0</td>
<td>c1.medium</td>
</tr>
<tr>
<td>INSTANC</td>
<td>i-90f16af9</td>
<td>am</td>
<td>0</td>
<td>c1.medium</td>
</tr>
<tr>
<td>INSTANC</td>
<td>i-93f16afa</td>
<td>am</td>
<td>0</td>
<td>c1.medium</td>
</tr>
<tr>
<td>INSTANC</td>
<td>i-92f16afb</td>
<td>am</td>
<td>0</td>
<td>c1.medium</td>
</tr>
</tbody>
</table>

Terminate all instances? [yes or no]: yes

<table>
<thead>
<tr>
<th>INSTANCE</th>
<th>i-b3f16ada</th>
<th>running shutting-down</th>
</tr>
</thead>
<tbody>
<tr>
<td>INSTANC</td>
<td>i-b9f16af2</td>
<td>running shutting-down</td>
</tr>
<tr>
<td>INSTANC</td>
<td>i-9af16af3</td>
<td>running shutting-down</td>
</tr>
<tr>
<td>INSTANC</td>
<td>i-9df16af4</td>
<td>running shutting-down</td>
</tr>
<tr>
<td>INSTANC</td>
<td>i-9cf16af5</td>
<td>running shutting-down</td>
</tr>
<tr>
<td>INSTANC</td>
<td>i-9f16af6</td>
<td>running shutting-down</td>
</tr>
<tr>
<td>INSTANC</td>
<td>i-9ef16af7</td>
<td>running shutting-down</td>
</tr>
<tr>
<td>INSTANC</td>
<td>i-91f16af8</td>
<td>running shutting-down</td>
</tr>
<tr>
<td>INSTANC</td>
<td>i-90f16af9</td>
<td>running shutting-down</td>
</tr>
<tr>
<td>INSTANC</td>
<td>i-93f16afa</td>
<td>running shutting-down</td>
</tr>
<tr>
<td>INSTANC</td>
<td>i-92f16afb</td>
<td>running shutting-down</td>
</tr>
</tbody>
</table>

loy:cloudera-for-hadoop-on-ec2-0.3.0 tom$
Case Study
Case Study: Adknowledge

- Ad network broker www.adknowledge.com
- Already using AWS for some time
- Use Hadoop and AWS to analyze clickstream events
Adknowledge Data Flow

- Continually loading clickstream data into S3. 1TB/month (compressed).
- Daily batch jobs on a Hadoop cluster of 100 EC2 extra large instances.
- Map to load data into HDFS from S3.
- 9 MapReduce jobs take 3.5 hours to run.
- Final result 1.5 GB (compressed) copied back to S3 before tearing down cluster.
Adknowledge Lessons Learned

- 2% of EC2 instances fail. Problem if one’s a namenode.
- System was not stable until Hadoop 0.18 (previously 2–3 complete failures per week)
- Error handling between dependent jobs is not robust
  - Hadoop Workflow System (HADOOP-5303) will improve this
Future

- Hadoop on EBS
- Use a hybrid local disk/EBS storage model
  - 1 replica on EBS, 2 local
- Share HDFS clusters
  - Use EBS snapshot facility
  - Like Amazon Public Datasets but for HDFS
- Hadoop on more cloud providers
Questions

- Apache Hadoop on EC2 and S3
  - http://wiki.apache.org/hadoop/AmazonEC2
  - http://wiki.apache.org/hadoop/AmazonS3
- Cloudera’s Distribution for Hadoop
  - http://www.cloudera.com/hadoop
  - http://www.cloudera.com/community-support
- Tom White
  - tomwhite@apache.org
  - tom@cloudera.com