Architecting for the next generation of Big Data
Hortonworks® HDP 2.0 on Red Hat® Enterprise Linux® 6 with OpenJDK® 7

Yan Fisher
Senior Principal Product Marketing Manager, Red Hat

Rohit Bakhshi
Product Manager, Hortonworks
Agenda

• Introduction to Red Hat and Hortonworks Alliance
• Trusted open source technologies
• Architecting an enterprise Big Data solution
  – Physical system deployments
    • Setting up the infrastructure and configuring the data platform
  – Exploring virtualization and consolidation use cases
  – Installation validation and performance assessment
• Summary
A Deepened Strategic Alliance

Companies strengthen relationship to bring Enterprise Apache Hadoop to the open hybrid cloud

- Engineer solutions for seamless customer experience
- Joint go to market activities
- Collaborative customer support
Infrastructure overview
Red Hat Enterprise Linux 6 and OpenJDK 7
Trusted Enterprise Platform: RHEL and OpenJDK

Red Hat Enterprise Linux

– Trusted by 90% of FORTUNE 500 companies
– According to a recent user survey* RHEL is used in:
  • Infrastructure (70% of respondents)
  • Big Data analytics, BI and data visualization (28% of respondents)
  • Big Data processing: Hadoop, MapReduce (21% of respondents)

Note: this is a multiple-choice question – response percentages may not add up to 100.
Trusted Enterprise Platform: RHEL and OpenJDK

OpenJDK

– Open source implementation of the Java® SE specification
– Red Hat has a leadership role in the OpenJDK project
– World Record* SPECjbb2013®-Composite result for Critical-JOPS running on Red Hat Enterprise Linux
– Same or better performance compared to Oracle JDK**
  • Hadoop infrastructure (Sort and Terasort)
  • Machine Learning (Bayesian classification and k-Means)

* As of April 3, 2014. SPEC® and SPECjbb are registered trademarks of the Standard Performance Evaluation Corporation. For more information about SPEC and it’s benchmarks see www.spec.org
** BIG DATA TECHNOLOGY ON RED HAT ENTERPRISE LINUX: OPENJDK VS. ORACLE JDK report
Hadoop® overview
Hortonworks Data Platform™ (HDP) 2.0
Hortonworks Data Platform 2.0 (HDP 2.0)
HDP 2.0 – What’s changed

Single Use System
Batch Apps

1\textsuperscript{st} Gen of Hadoop

MapReduce
(cluster resource management & data processing)

HDFS
(redundant, reliable storage)

Multi-Use Data Platform
Batch, Interactive, Online, Streaming, …

2\textsuperscript{nd} Gen of Hadoop

Efficient Cluster Resource Management & Shared Services
(YARN)

Redundant, Reliable Storage
(HDFS)

Classic Hadoop Apps

Flexible Data Processing
Hive, Pig, others...

Online Data Processing
HBase, Accumulo

Stream Processing
Storm

Batch MapReduce

Batch & Interactive Tez

Others…
Architecting an enterprise Big Data solution
Physical system deployments
Physical and Logical Configurations

Two master nodes
- 2 x Intel Xeon® X5670; 96GB RAM; 6 HDDs

Four data nodes
- 2 x Intel Xeon® X5670; 48GB RAM; 8 HDDs
Setup and Configuration

Workflow summary

OS Installation
- NTP
- File system
- Disable Services
- OpenJDK
- Tuned

HDP 2.0 Installation
- Configure repos
- Install packages

HDFS Configuration
- Folders and permissions
- HDP configuration files

YARN Configuration
- Folders and permissions
Setup and Configuration

OS Installation

• Use PXE, Kickstart or physical media
  – Generally, defaults are acceptable to start

Post-install

• Synchronize clocks to the same NTP server
• Configure the file systems on each server
• Disable unnecessary services (cups, autofs, postfix etc., etc., etc.)
• Install the latest version of OpenJDK 7 and make it the default
• Install `tuned` and set it to `enterprise-storage` profile
Setup and Configuration

Hortonworks Data Platform 2.0 installation

Configure Repo
- Set up Repositories
  - Ambari repo
  - HDP Stack repo
  - HDP Utils repo

Deploy HDP
- Options to Deploy
  - Ambari
    - Automated and GUI driven
  - Scripted
    - RPM: Script driven

Configure HDP
- Configure
  - Modify default XML configs
  - Optimize for Hardware, cluster layout and workloads
HDFS – Configuration Best Practices

• For Master nodes, utilize redundant hardware
• For Slave nodes, use JBOD and commodity hardware
• Separate Operation System partition for logs
• Use Rack Awareness for fault tolerance and performance across racks
YARN – Configuration Best Practices

Configure for optimized allocation of distributed resources across multi-workload processing.
## YARN – Configuration Best Practices

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| yarn.scheduler.minimum-allocation-mb                   | Smallest Container Allowed in MB  
All Containers must be an multiple of the minimum container size  
ie- 1024 allows for 1024, 2048, 3072, 4096, etc.  |
| yarn.scheduler.maximum-allocation-mb                   | Largest Container Allowed. A Multiple of the minimum-allocation-mb above  
Depending on your setup you may want to allow the entire node for MR, or restrict it to smaller then a node to prevent potential malicious actions.  |
| mapreduce.map.memory.mb                                | The size of the container for the Mapper task  
The java opts for the Mapper JVM, make sure that the max heap is less than the size of the container.  |
| mapreduce.map.java.opts                                 |                                                                                                                                         |
| mapreduce.reduce.memory.mb                             | The size of the container for the Reducer task  
The java opts for the Reducer JVM, make sure that the max heap is less than the size of the container.  |
| mapreduce.reduce.java.opts                             |                                                                                                                                         |
Architecting an enterprise Big Data solution
Exploring virtualization and consolidation use cases
Flexible Deployments

Virtualization (as a trivial case of consolidation)
– Easy transition to the updated hardware or the cloud
– Power savings and simplified datacenter management
– Well-defined VMs are easily provisioned and destroyed

Consolidation
– Higher utilization of server resources, low overhead with KVM
– More than adequate performance for several workload types
  – Hadoop infrastructure (Sort) and Machine Learning (Naive Bayes)
Virtualized Setup and Configuration

Workflow summary

- Host installation and KVM configuration
  - OS installation
  - KVM installation
  - Disk configuration
  - Tuned
  - Numad
  - Clone

- OS installation
  - SSH
  - NTP
  - File system
  - Disable Services
  - OpenJDK
  - Tuned

- HDP 2.0 installation
  - Configure repos
  - Install packages

- HDFS configuration
  - Folders and permissions
  - HDP configuration files

- YARN configuration
  - Folders and permissions
Physical and Logical Configurations

Virtualization

Two master nodes
- 2 x Intel Xeon® X5670; 96GB RAM; 6 HDDs

Four virtualized data nodes (guests)
- 24 vCPUs; 44GB vRAM; 1 x 20GB system disk; 6 x 300GB data disks

Four physical servers
- 2 x Intel Xeon® X5670; 48GB RAM; 8 HDDs
Setup and Configuration

Virtualization

OS Installation on Hypervisor

- Use PXE, Kickstart or physical media

Post-install

- Install KVM: Virtualization, Platform and Client
- Synchronize clocks to the same NTP server
- Disable unnecessary services (cups, autofs, postfix etc etc etc)
- Install tuned and set it to virtual-host profile
- Configure disk storage for virtual guests
- Configure virtual networking
Setup and Configuration
Virtualization
OS Installation on Guest
  • Use PXE, Kickstart or physical media
Post-install
  • Synchronize clocks to the same NTP server
  • Disable unnecessary services (cups, autofs, postfix etc etc etc)
  • Install tuned and set it to virtual-guest profile
  • Install the latest version of OpenJDK 7 and make it default
  • Clone this guest as many times as needed
  • Move images to the respective hypervisor and attached disks
  • Start each guest and configure their networking and file system
Physical and Logical Configurations
Consolidation

Two master nodes: 2 x Intel Xeon® X5670; 96GB RAM; 6 HDDs

Four virtualized data nodes (guests): 12 vCPUs; 48GB vRAM; 1 x 20GB system disk; 8 x 73GB data disks

- One physical server: 2 x Intel Xeon® E5-2697 v2; 384GB RAM; 24 HDDs
Setup and Configuration
Consolidation

OS installation on Hypervisor

• Follow the same installation instructions as before

Follow same instructions for post-install, plus

• Configure data disks for guests to use

OS installation on Guests

• Follow the same installation instructions as before

• Divide the total number of cores on the system equally among all guests and create first VM

• **Clone** this guest as many times as needed
Best Practices

Rack Awareness for Virtualized Infrastructures
Post-installation validation
Physical and virtual
Post-installation Validation

Use several common workloads from Intel Hibench suite
• Select tests with most balanced CPU, IO, and network profiles
• Observe OpenJDK performance and compare to Oracle JDK

Test Hadoop infrastructure
• Sort - micro benchmark performing sort operations, a critical feature of many MapReduce jobs
• Terasort - is a Big Data version of Sort. It sorts 10 billion 100-byte records produced by the TeraGen generator program
Post-installation Validation

Test real-world workloads (Machine learning)

• Naïve Bayesian Classification - machine learning and classification implementation; used for finding patterns and assigning data sets to classes

• K-Means - implement highly used and well-understood K-Means clustering algorithm operating on a large set of randomly generated numerical multidimensional vectors with specific statistical distributions
Post-installation validation

Findings

- Low-level Data Manipulation: Sort
- Data Classification: Naïve Bayes
- Data Manipulation: TeraSort
- Data Clustering: K-Means

Time in seconds (lower is better)
Post-installation Validation

Findings

<table>
<thead>
<tr>
<th>Task</th>
<th>OpenJDK</th>
<th>Oracle JDK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sort</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TeraSort</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bayes classifier</td>
<td></td>
<td></td>
</tr>
<tr>
<td>K-means</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Time in seconds (lower is better)
Summary

• The Hortonworks Data Platform is a completely open source production-ready distribution of Apache Hadoop.

• Result prove that Red Hat Enterprise Linux with OpenJDK provide a solid foundation for enterprise deployments of HDP.

• OpenJDK performs as well as Oracle JDK when running HDP.

• This infrastructure works well in both physical and virtual domains.

• This powerful infrastructure platform for Hadoop deployments can support your organization’s needs today and well into the future.
Resources

Get the Hortonworks Data Platform

Exploring the next generation of Big Data solutions with Hadoop 2

Big Data Technology on Red Hat Enterprise Linux: OpenJDK vs Oracle JDK
Questions?

Contact:

Yan Fisher
yfisher@redhat.com

Rohit Bakhshi
rohit@hortonworks.com