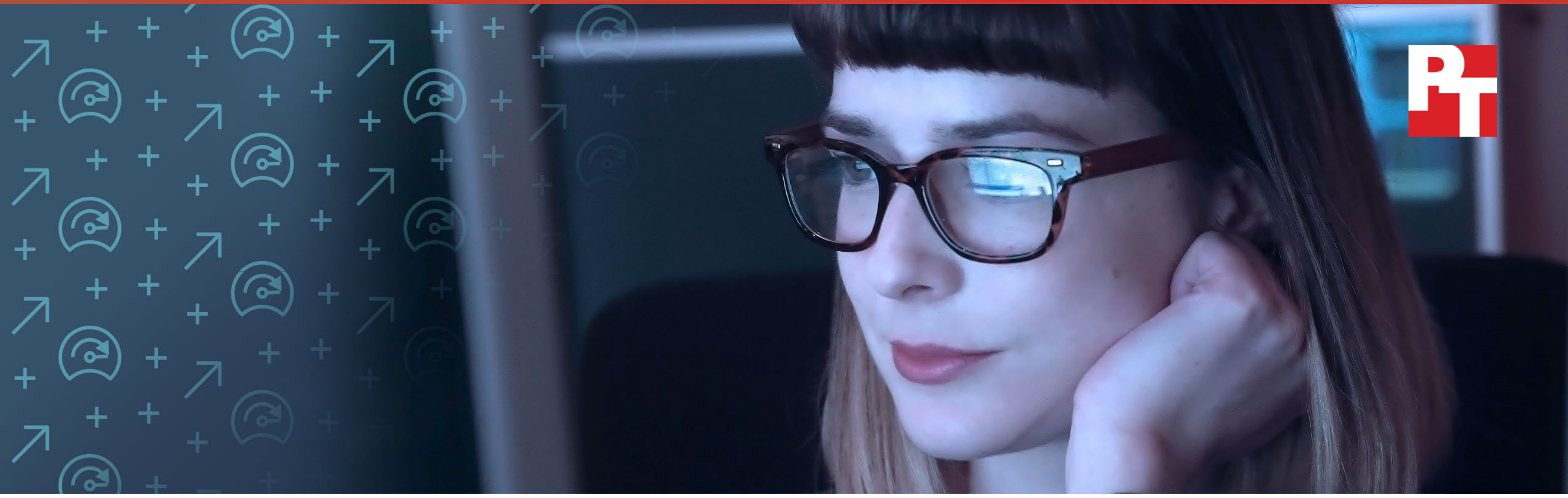


Run big data analytics on a powerful on-premises Dell EMC PowerEdge FX2 solution and save money over three years

Compared to AWS, the PowerEdge FX2 could reduce spending on big data while saving time on data analysis workloads

When it comes to processing Hadoop® and Apache Spark™-based data analytics workloads against large sets of data, an on-premises Dell EMC™ PowerEdge™ FX2 solution can save money over time and also deliver speed.

We found a Dell EMC PowerEdge FX2 solution featuring PowerEdge FC640 server nodes powered by Intel® Xeon® Scalable processors had a lower three-year total cost of ownership (TCO) than an Amazon® Web Services (AWS®) Elastic Compute Cloud® (EC2) configuration. The Dell EMC solution also categorized, characterized, and clustered data faster.

The sooner your big data workloads finish, the sooner data analysts and decision-makers can gain the insight they need to create strategies for growth. The Dell EMC solution could give your organization speedy data analytics to meet modern challenges while helping your bottom line.

Save up to 42 percent* over three years
Running big data workloads on-premises can pay off over time

Process more data per second
Up to 36 percent better throughput

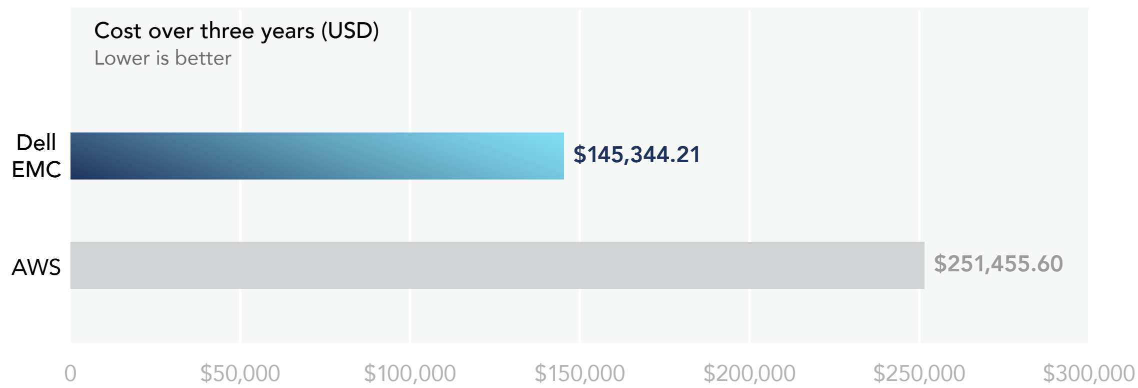
24 percent less waiting for data analysis
Completed a big data workload over six minutes faster

*See Appendix C. We tested a single configuration, but multiple configurations are available.

Save money over three years with an on-premises solution

Cost may inhibit some organizations from getting or expanding big data analysis. For instance, discrete and uncontrolled public cloud usage can cause your organization to pay for unoptimized IaaS usage, which can hurt your bottom line. And even if your organization can control these shadow IT occurrences, IT staff monitoring and managing cloud usage may not fully understand the pricing and tiering options.

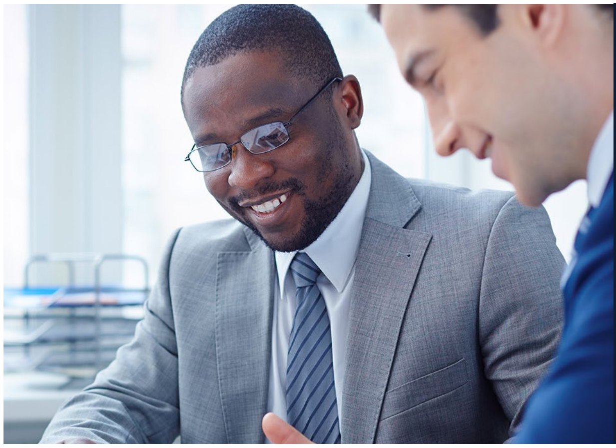
Save up to 42 percent over three years
Running big data workloads on-premises can pay off over time



The on-premises Dell EMC solution powered by Intel Xeon Scalable processors can give your users the managed virtual resources they need in an on-demand, flexible environment that can make budgeting easier. Based on our analysis, an organization can save up to $106,111 over three years with the Dell EMC solution processing machine learning tasks. This is a 42 percent reduction in spending compared to the AWS solution, and you avoid any unexpected IT costs related to public cloud usage. Your analysis may vary for on-premises datacenter and administration costs. For the full set of assumptions and a breakdown of costs, see Appendix C.

Complete data analytics workloads in less time

Big data can generate valuable insights that help an organization predict behaviors or outcomes. Decision-makers and their teams can use the output to orchestrate initiatives such as targeted email campaigns that drive sales or customer feedback analysis that improves product quality. Organizations with the on-premises Dell EMC solution may have an advantage over competitors, as they can work with large data sets more quickly than those who use AWS.



Measure big data capabilities with HiBench

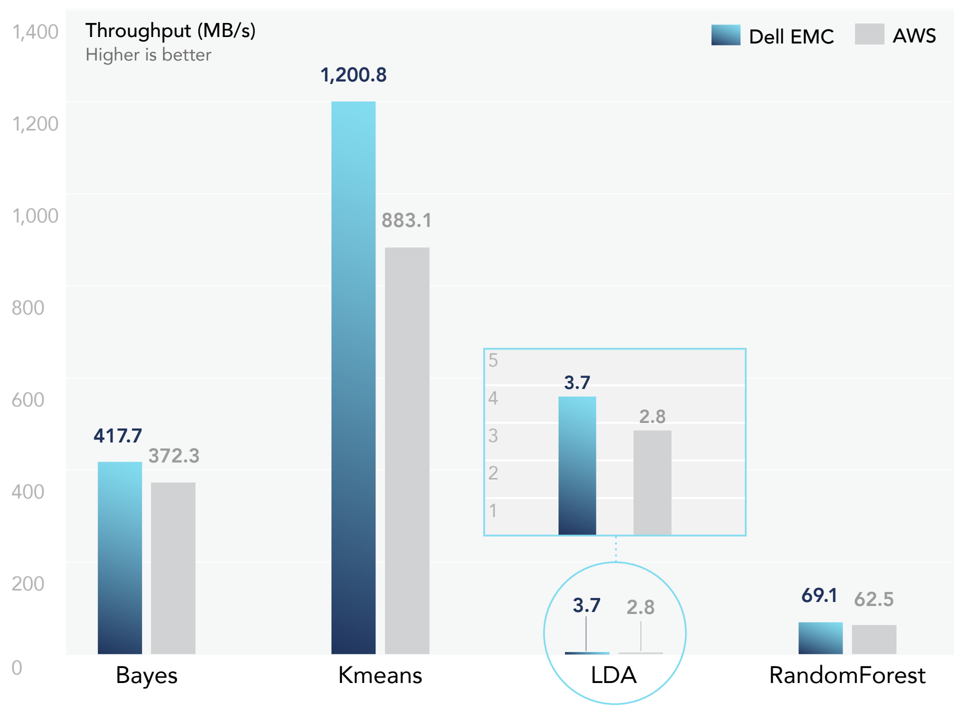
Intel HiBench 7.1 is a big data benchmark suite for Hadoop. Some tools in the suite are synthetic micro-benchmarks while others are real-world Hadoop applications. The output of the tools can demonstrate a solution's processing speed, throughput, bandwidth, CPU utilization, data access patterns, and other metrics as they relate to processing big data workloads.

To see how the two solutions could handle real-world big data work, we ran four HiBench tests on them:

- **Bayes**: Naïve Bayesian Classifiers use probability to categorize data; as an example, Bayesian classifiers can be used to identify and filter spam emails based on the presence of specific words in the body and header and embedded HTML and JavaScript
- **Kmeans**: The K-means clustering algorithm divides data or users into subgroups; businesses use it for targeted advertising or marketing or to help determine behaviors of groups of uncategorized people
- **Latent Dirichlet allocation (LDA)**: LDA is a technique to dynamically identify the topics discussed in a given document; it analyzes the words in a document, temporarily categorizes them, refines the topic of the document with the previously identified categories, and summarizes the document; a business could use LDA, for example, to organize customer reviews on a product or service
- **RandomForest (RF)**: Random forests can be used to make predictions by running multiple decision trees with slightly different weights and comparing the outcome between the trees to prevent overfitting—this can greatly increase the accuracy of a decision tree in classification and regression; for example, a bank or financial institution could use RF to make credit risk predictions

The chart below shows the throughput results from our testing. When all available compute nodes in our configuration were running the K-means analysis in particular, the Dell EMC solution powered by Intel Xeon Scalable processors delivered a throughput of more than 1,200 MB per second. That's a 36 percent increase over AWS. We chose four AWS m4.16xlarge instances for our comparison because they were the closest comparison at the time of testing to the FC640 blades we used in the FX2 solution.

Process more data per second
Up to 36 percent better throughput



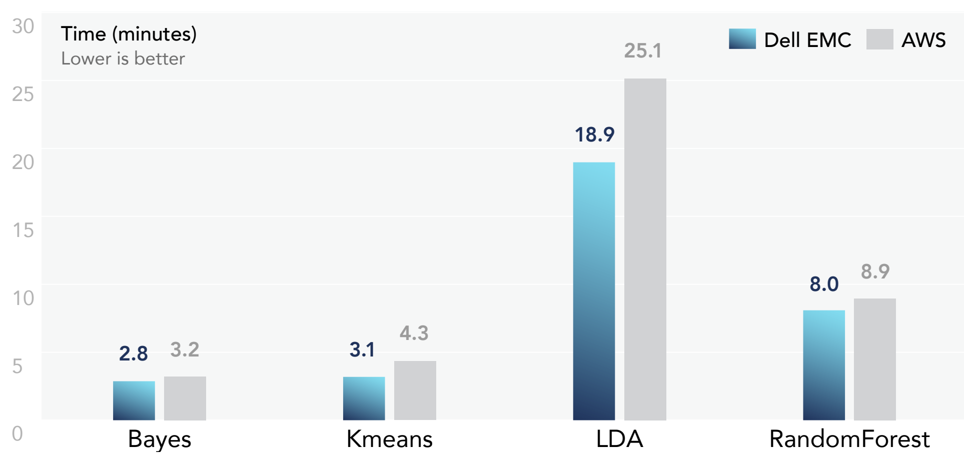
About the Dell EMC solution powered by Intel Xeon Scalable processors

The Dell EMC PowerEdge FX2 is a modular server platform that can combine servers, storage, and networking into a single 2U chassis. The FX2 in our solution contained four new two-socket, 1U Dell EMC PowerEdge FC640 server nodes featuring the new Intel Xeon Gold 5120 processor. Your organization can configure the FC640 nodes with up to 2 TB of memory and up to 12.8 TB of PCIe SSD storage. For various performance requirements, including big data analytics, the FX2 can house 1U Dell EMC PowerEdge FD332 storage modules as well.

To learn more about the Dell EMC PowerEdge FX architecture, visit www.dell.com/en-us/work/shop/cty/pdp/spd/poweredge-fx.

Thanks to mobile devices, widespread internet accessibility, the Internet of Things, and other technological advances, the rate of data creation is high. Organizations are gathering data constantly, learning things such as how long users are on a web page, how many views ads get on social media platforms, and what links shoppers click on a product ordering page. Staying on top of a changing world requires speed when processing large volumes of data. The faster a solution can process and analyze data, the sooner a marketing team, for example, can determine who to target for their next email campaign.

The Dell EMC solution powered by Intel Xeon Scalable processors needed 3 minutes and 11 seconds to process 224.4 GB of data for the Kmeans test. Compared to the AWS solution, which needed 4 minutes and 20 seconds, that's a 26 percent decrease in duration. We saw the biggest time difference between the two solutions in the LDA test—the Dell EMC solution saved 6 minutes and 10 seconds (a 24 percent decrease from the AWS time). The chart below shows the times to complete all four tests for both solutions.



Maintain security, control, and compliance: More benefits of an on-premises solution

Compared to the public cloud, an on-premises Dell EMC solution powered by Intel Xeon Scalable processors can offer many benefits that could be right for your organization and its big data goals:

- **Use conventional internal IT support and a familiar backup and security strategy.**
  You likely have many crucial management processes, policies, and tools in place that your organization can expand to include your new Dell EMC solution.
- **Customize as necessary with comprehensive options.**
  Your organization may require high levels of customization or extensive custom programming capabilities. An on-premises Dell EMC solution can offer the granular control to meet these requirements.
- **Ensure data stays confined to one geographic location.**
  If your organization requires regulatory compliance related to data location, then running data analytics workloads on an on-premises Dell EMC solution can help meet those regulations.
- **Apply existing security policies.**
  Your organization likely already has strict datacenter security guidelines. You can tailor the Dell EMC solution to meet these guidelines, rather than determine compatibility or learn new security policies with a public cloud.
- **Avoid complicated data migration.**
  It can be challenging to move large sets of data from your datacenter to the cloud. In addition, that migration can lengthen the time it takes to get helpful analysis. An on-premises Dell EMC solution makes it so data scientists or IT staff don't have to move ever-changing data sets.

Build a sturdy framework for big data workloads

Apache Spark™ is a widely used in-memory framework for advanced processing of big data analytics. Companies like Amazon, eBay, NASA Jet Propulsion Laboratories, TripAdvisor, and Yahoo use this tool to help them get real-time bid optimization, machine learning-based user data modeling, forecasting, and other predictive analytics. We used Apache Spark for our test configurations.

Conclusion

Big data analytics can boost your organization by growing your revenue and customer base, improving products or operational efficiency, and in a host of other ways. The public cloud is a popular platform to analyze large sets of data, but it might not be the right option for your organization.

A Dell EMC PowerEdge FX2 solution with FC640 server nodes powered by Intel Xeon Scalable processors can cost up to $106,111 less than the public cloud AWS solution we tested while delivering all the advantages on-premises solutions have over the public cloud. Considering the Dell EMC solution also categorized, characterized, and clustered data faster than the AWS solution, an on-premises Dell EMC solution might be the better choice for your company's big data initiatives.

On November 19, 2017, we finalized the hardware and software configurations we tested. Updates for current and recently released hardware and software appear often, so unavoidably these configurations may not represent the latest versions available when this report appears. We concluded hands-on testing on December 4, 2017.

Appendix A: Server configuration information

Server configuration information	4 x Dell EMC PowerEdge FC640 (systems under test)
BIOS name and version	Dell® 1.0.1
Non-default BIOS settings	None
Operating system name and version/build number	Red Hat® Enterprise Linux® 7.4 / 3.10.0-693.2.2.el7.x86_64
Date of last OS updates/patches applied	11/15/2017
Power management policy	Performance
Processor	
Number of processors	2
Vendor and model	Intel Xeon Gold 5120
Core count (per processor)	14
Core frequency (GHz)	2.20
Stepping	1
Memory module(s)	
Total memory in system (GB)	192
Number of memory modules	12
Vendor and model	Hynix HMA82GR7AFR8N-VK
Size (GB)	16
Type	PC4-21300R
Speed (MHz)	2,666
Speed running in the server (MHz)	2,444
Storage controller	
Vendor and model	Dell PERC H330 Mini
Cache size (GB)	N/A
Firmware version	25.5.0.0019
Driver version	3.10.0-693.5.2.el7.x86_64
Local storage (type A)	
Number of drives	2
Drive vendor and model	Dell 0R7YTT
Drive size (GB)	16
Drive information (speed, interface, type)	MicroSD card

Server configuration information	4 x Dell EMC PowerEdge FC640 (systems under test)
Local storage (type B)	
Number of drives	2
Drive vendor and model	Samsung™ PM1635a SSD
Drive size (GB)	800
Drive information (speed, interface, type)	12Gb SAS, SSD
Network adapter	
Vendor and model	Intel Ethernet 10G 2P X710-k bND
Number and type of ports	2 x 10GbE
Driver version	1.6.27-k

Server enclosure configuration

Server enclosure configuration information	Dell EMC PowerEdge FX2s (system under test)
Number of management modules	1
Management module firmware revision	2.0
CMC module firmware	2.00
Midplane version	1.0
Virtual	
Vendor and model number	Dell PowerEdge FN 410S IOM
I/O module firmware revision	9.10
Number of modules	2
Occupied Slots	A1, A2
Power supplies	
Vendor and model number	Dell 0W1R7VA00
Number of power supplies	2
Wattage of each (W)	2000
Cooling fans	
Number of fans	8

Virtual machine configuration

VM configuration information	AWS m4.16xlarge VMs (systems under test)
OS	Amazon Linux AMI release 2017.03 / 4.9.27-14.31.amzn1.x86_64
vCPU count	64
Memory (GB)	256
Virtual disk (GB)	30 (OS), 2 x 600 (HDFS)

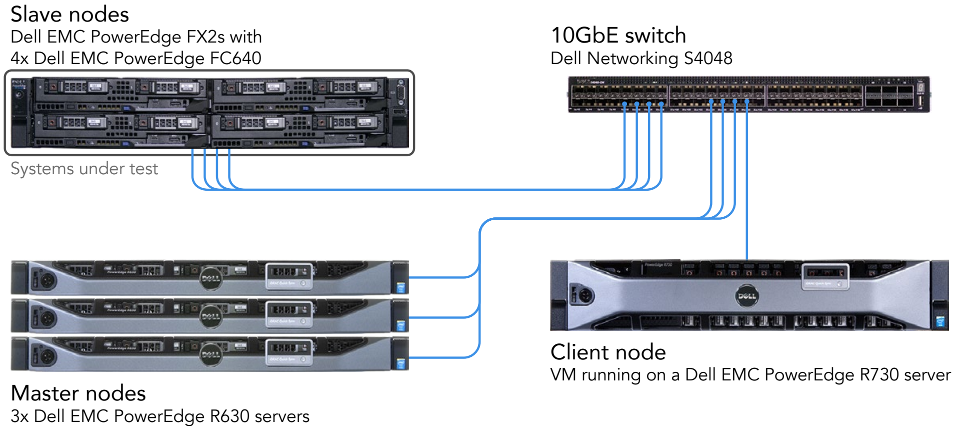
Appendix B: How we tested

We performed the following steps to install and configure the testbed for Hortonworks® Data Platform and HiBench. Your environment may differ slightly, which can affect the setup of IPs and naming of hosts.

Setting up the Dell EMC PowerEdge FX2s solution

Dell EMC solution configuration

In addition to our systems under test, we used Dell EMC PowerEdge R630 and R730 servers for infrastructure purposes.



Installing the operating system

1. Insert the Red Hat Enterprise Linux 7.4 installation media, and power on the system.
2. Select Install Red Hat Enterprise Linux 7.4, and press Return.
3. Select English, and click Continue.
4. Click NETWORK & HOST NAME.
5. Click ON for the 10Gb Ethernet connection you are using.
6. Select the 10Gb Ethernet connection you are using, and click Configure.
7. Click IPv4 Settings.
8. Change Method to Manual.
9. Click Add, and enter the IP address, netmask, and gateway.
10. Click Save.
11. Change the host name based on the role of your system, and click Apply.
12. To exit the network configuration, click Done.
13. Click INSTALLATION DESTINATION.
14. Select I will configure partitioning, and click Done.
15. Select Standard Partition, and click Click here to create them automatically.
16. Eliminate the swap, and click Done.
17. When the summary of changes appears, click Accept Changes.
18. Click Begin Installation.
19. Click ROOT PASSWORD.
20. Enter the root password, and confirm it. Click Done.
21. When the installation finishes, click Reboot.
22. For the remaining servers in the test bed, complete steps 1 through 21.

Configuring the operating system

Most actions in this section require running commands.

1. Log into the Red Hat console.
2. Disable SELinux:

```
setenforce 0
sed -i 's/SELINUX=.*/SELINUX=disabled/' /etc/selinux/config
```

3. Disable the firewall:

```
systemctl stop firewalld
systemctl disable firewalld
```

4. Configure default file permissions:

```
echo umask 0022 >> /etc/profile
```

5. Either register the system with Red Hat or configure yum patch repositories to operate from a locally accessible repository.
6. Remove chrony from the system:

```
yum remove -y chrony
```

7. Update the system:

```
yum update -y
```

8. Reboot the system.
9. Install the prerequisites for Ambari and HDP:

```
yum install -y ntp time xfsprogs tuned wget vim nfs-utils openssh-clients man zip unzip numactl
sysstat bc lzop xz libhugetlbfs python numpy blas64 lapack64 gtk2 atk cairo gcc-gfortran tcsh lsof
tcl tk java-1.8.0-openjdk-devel perl-Data-Dumper.x86_64
```

10. Enable NTP:

```
sed -i '/^server [^ ]* iburst/d' /etc/ntp.conf
echo "server [your NTP server IP address] iburst" >> /etc/ntp.conf
systemctl enable ntpd
systemctl start ntpd
```

11. Reboot the system.
12. Clear the SSH settings:

```
mkdir -p /root/.ssh
chmod 700 /root/.ssh
cd /root/.ssh
mv * *.orig
```

13. On the hosts that will be data nodes, format and prepare the drives for HDFS, making sure to modify the SKIP_LETTER variable for the operating system drive:

```
#!/bin/bash
FS=xfs
COUNT=0
SKIP_LETTER="w"
umount /grid/*
rmdir /grid/*
rmdir /grid
mkdir -p /grid
for i in {a..x};
do
  if [ "$SKIP_LETTER" = "$i" ]; then
    continue
  fi
  DEV=/dev/sd$i
  GRID=/grid/$COUNT
  echo $DEV $COUNT
  mkdir -p $GRID
  #dd if=/dev/zero of=$DEV bs=1M count=10 oflag=direct
  sync
  mkfs.${FS} $DEV
  echo -e "'blkid -p $DEV | awk '{print $2}''\t${GRID} \t${FS}\tdefaults,noatime,nodiratime,nofail,x-
systemd.device-timeout=60\t0 0" >> /etc/fstab
  ((COUNT++))
done
```

14. On the data nodes, run `vim /etc/default/grub` and modify the GRUB_CMDLINE_LINUX setting by adding `iommu=pt` to the end of the line.
15. Configure the data nodes to run in high throughput performance mode:

```
tuned-adm profile throughput-performance
```

16. Configure the data nodes to run the Mellanox cards in high performance mode. Repeat this action each time you reboot.
17. For the remaining servers in the test bed, complete steps 1 through 16.
18. On one host, create an SSH private key for all hosts:

```
ssh-keygen -t rsa -q
cp id_rsa.pub authorized_keys
echo "StrictHostKeyChecking=no" > config
```

19. Copy the key to the other hosts.
20. Create a hosts file containing a FQDN, nickname, and IP address of every host in the cluster. Copy the file to all of your hosts.

Installing Apache Ambari

1. Log into the Ambari host.
2. Add the Ambari repository to yum repositories:

```
wget -nv http://public-repo-1.hortonworks.com/ambari/centos7/2.x/updates/2.5.0.3/ambari.repo -O /etc/
yum.repos.d/ambari.repo
```

3. Navigate to the OpenJDK directory, and start the Ambari server setup with the following command:

```
ambari-server setup
```

4. To accept the default of not customizing a user account for the ambari-server daemon, press Return.
5. When prompted, choose a custom JDK by typing `3`, and pressing Return.
6. To accept the default of not entering advanced database configuration, press Return.
7. To start up the Ambari host, type `ambari-server start`.

Installing Hortonworks Data Platform on the cluster

1. Open a web browser, and navigate to the Ambari host's website.
2. Log into the Ambari host website.
3. In Welcome to Apache Ambari, click Launch Install Wizard.
4. In the Get Started page, name your cluster, and click Next.
5. In Select Version, select the latest version, check Use Public Repository, and click Next.
6. In Install Options, enter the hosts you plan to use. For example:

```
namenode[1-3].hdp.local
spark[1-4].hdp.local
client1.hdp.local
```

7. Copy the contents of the SSH private key you created previously into the host registration information. Type root for the User Account and 22 for the port number.
8. Click Register and Confirm.
9. When the Host name pattern expressions window appears, verify that all of the hosts are there, and click OK.
10. In the Confirm Hosts window, verify that all of the hosts for installation are there, and click Next.
11. In Choose Services, uncheck the following services:
    - Accumulo
    - Atlas
    - Falcon
    - Flume
    - HBase
    - Sqoop
    - Oozie

    Leave the rest of the services at their defaults, and click Next.
12. In Assign Masters, assign services to the following master servers:
    - namenode1
        - NameNode
        - ZooKeeper Server
        - DRPC Server
        - Storm UI Server
        - Nimbus
        - Infra Solr Instance

- Metrics Collector
    - Grafana
    - Kafka Broker
    - Knox Gateway
    - HST Server
    - Activity Explorer
    - Activity Analyzer
    - Spark History Server
    - Spark2 History Server
- namenode2
    - SNameNode
    - App Timeline Server
    - ResourceManager
    - History Server
    - WebHCat Server
    - Hive Metastore
    - HiveServer2
    - ZooKeeper Server

13. In Assign Slaves and Clients, apply the slave and client components to the following hosts:
    - spark1-4
        - DataNode
        - NodeManager
        - Supervisor
    - client1
        - Client
14. In Customize Services, perform the following actions:
    a. Enter a database password for the Hive Metastore.
    b. Enter a password for the Grafana Admin.
    c. Enter a password for the Knox Master Secret.
    d. Enter a password for the admin in SmartSense.
    e. In the HDFS section, add the following line to the DataNode directories textbox:
    ```
    /grid/0,/grid/1
    ```
15. When warned in the Configurations window, click Proceed Anyway.
16. In Review, verify your settings, and click Deploy.
17. After the deployment completes, in Install, Start and Test, click Next.
18. In Summary, click Complete.

Installing HiBench

Most actions in this section require running commands.

1. Log into the client host.
2. Install the prerequisite packages to the client:
```
yum install -y maven git vim numpy blas64 lapack64
echo "/usr/hdp/current/hadoop-client/lib/native" > /etc/ld.so.conf.d/hadoop-client-native.conf
```
3. Change to the HDFS user, create the HiBench directory in the dfs, and set permissions:
```
su - hdfs
hdfs dfs -mkdir /HiBench
hdfs dfs -chown -R root:hadoop /HiBench
hdfs dfs -mkdir /user/root
hdfs dfs -chown root /user/root
exit
```
4. Edit the bash profile to automatically include the HiBench prerequisites:
```
vim ~/.bash_profile
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk
```

```
export HADOOP_HOME=/usr/hdp/current/hadoop-client
export SPARK_HOME=/usr/hdp/current/spark2-client
export KAFKA_HOME=/usr/hdp/current/kafka-broker
export LD_LIBRARY_PATH=/usr/hdp/current/hadoop-client/lib/native
```

5. Download the HiBench benchmark from GitHub:

```
cd /root
git clone https://github.com/intel-hadoop/HiBench.git
```

6. Change to the newly downloaded HiBench directory, and type the following command:

```
mvn -Dspark=2.1 -Dscala=2.11 clean package | tee hibench_build.log
```

7. Modify the `99-user_defined_properties.conf` file by changing the following lines:

```
hibench.hadoop.home               /usr/hdp/current/hadoop-client
hibench.spark.home                /usr/hdp/current/spark2-client
hibench.hadoop.mapreduce.home     /usr/hdp/current/hadoop-mapreduce-client
```

Configuring Hortonworks Data Platform

Using the Ambari console, we made the following changes to the settings to the services:

- HDFS
    - DataNode
    - DataNode directories - each of the drives on the server
    - DataNode failed disk tolerance - 2
- YARN
    - Memory
        - Memory allocated for all YARN containers on a node - 192 GB
        - Maximum Container Size (Memory) - 196608 MB
- CPU
    - Percentage of physical CPU allocated for all containers on a node - 100%
        - Number of virtual cores - 56
        - Maximum Container Size (VCores) - 56

Configuring HiBench

We used the following settings in testing:

hibench.conf	Dell testbed	AWS testbed
hibench.default.map.parallelism	832	1024
hibench.default.shuffle.parallelism	832	1024
hibench.scale.profile	bigdata	bigdata
spark.conf	Dell testbed	AWS testbed
hibench.yarn.executor.num	52	64
hibench.yarn.executor.cores	4	4
spark.executor.memory	12g	14g
spark.driver.memory	12g	14g

Running HiBench

1. Log into the client host.
2. In the HiBench directory, navigate to the bin/workload/[benchmark] directory.
3. Initialize the data:

```
./prepare/prepare.sh
```

4. Log into the first data node.
5. Clear the PageCache, dentries, and inodes:

```
echo 3 > /proc/sys/vm/drop_caches
```

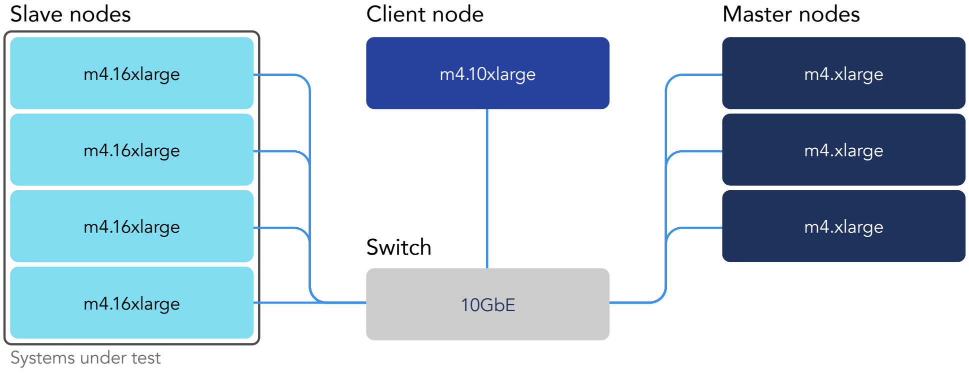
6. For the remaining data nodes, complete steps 4 and 5.
7. After clearing the cache, wait 5 minutes.
8. Log into the client host.
9. In the HiBench directory, navigate to the bin/workload/[benchmark] directory.
10. Run the benchmark:

```
./spark/run.sh
```

11. When the test completes, record the results.

Setting up the AWS solution

AWS solution configuration



Deploying Cloudbreak

1. Log into the AWS console.
2. In the Instance view, click Launch Instance.
3. In the Choose an Amazon Machine Image screen, select Community AMIs, enter the appropriate Cloudbreak AMI into the search field (the Cloudbreak AMI for the datacenter can be found at this link), and press Return.
4. Verify that the image is cloudbreak-deployer-[version-name-here], and click Select.
5. Select m3.large as the instance type, and click Next.
6. Leave instance details at defaults, and click Next.
7. Leave the storage at defaults, and click Next.
8. Add tags if desired, and click Next.
9. Configure the security group, and click Next.

Configuring Cloudbreak

1. Using the saved SSH key from creating the VM, SSH into the newly deployed Cloudbreak VM (default username: cloudbreak).
2. Navigate to the cloudbreak-deployment directory:

```
cd /var/lib/cloudbreak-deployment/
```

3. Create a Profile file:

```
vi Profile
```

4. In the Profile file, add the following lines:

```
export UAA_DEFAULT_SECRET='[SECRET]'
export UAA_DEFAULT_USER_PW='[PASSWORD]'
```

5. Save, and exit vi.
6. Start cloudbreak:

```
cbd start
```

7. Using a web browser, navigate to the Cloudbreak instance's web address.
8. Log in (default username: admin@example.com) with the password.
9. In the top right, click the dropdown list, and select Manage credentials.
10. Making sure that AWS is selected, choose Key Based credentials, put in the AWS access key and secret access key, and click Create credential.
11. Expand Manage blueprints, select hdp-spark-cluster, and click Copy & edit.

12. Replace the values on the left with the values on the right:

Original value	New value
"name": "SPARK_CLIENT"	"name": "SPARK2_CLIENT"
"name": "SPARK_JOBHISTORYSERVER"	"name": "SPARK2_JOBHISTORYSERVER"
"name": "SPARK_CLIENT"	"name": "SPARK2_CLIENT"

13. In the same node as the SPARK2_JOBHISTORYSERVER and SPARK2_CLIENT, add the follow lines (under SPARK2_CLIENT):

```
{
    "name": "SPARK2_THRIFTSERVER"
},
{
    "name": "LIVY2_SERVER"
},
```

14. At the bottom, change stack_version to 2.6, rename the blueprint, and click create blueprint.
15. For Slave node configuration, click Manage templates, and choose the following options (Note: At the time of test planning, m4.16xlarge was not an option in the Cloudbreak templates, so we created the instances using m4.10xlarge and then modified the instances to 16x after deployment):
    - Name: m4-10xlarge
    - Instance Type: m4.10xlarge (After deploying the instances, change them to m4.16xlrage.)
    - Volume Type: Provisioned IOPS (SSD)
    - IOPS Provisioned: 4,500
    - Attached Volumes Per Instance: 2
    - Volume Size (GB): 600
16. Click Create template.
17. For Master node configuration, click Manage templates, and choose the following options:
    - Name: m4-4xlarge
    - Instance Type: m4.4xlarge
    - Volume Type: General Purpose (SSD)
    - Attached Volumes Per Instance: 1
    - Volume Size (GB): 30
18. Click Create template.
19. For client node configuration, click Manage templates, and choose the following options:
    - Name: client-node
    - Instance Type: m4.10xlarge
    - Volume Type: General Purpose (SSD)
    - Attached Volumes Per Instance: 1
    - Volume Size (GB): 30

Deploying the Apache Spark cluster

1. Near the top right, click Create cluster.
2. Name the cluster, add tags if desired, and click Setup Network and Security.
3. Choose network options, and click Choose Blueprint.
4. Choose the newly created hdp-spark21-cluster template, and change the following settings:
    - host_group_client_1
        - Group Size: 1
        - Template: m4.10xlarge
        - Security Group: default-aws-only-ssh-and-ssl
        - Ambari Server: checked
    - host_group_master_3
        - Group Size: 1
        - Template: m4.4xlarge
        - Security Group: default-aws-only-ssh-and-ssl
        - Ambari Server: unchecked

- host_group_slave_1
    - Group Size: 4
    - Template: m4.10xlarge
    - Security Group: default-aws-only-ssh-and-ssl
    - Ambari Server: unchecked
- host_group_master_2
    - Group Size: 1
    - Template: m4.4xlarge
    - Security Group: default-aws-only-ssh-and-ssl
    - Ambari Server: unchecked
- host_group_client_1
    - Group Size: 1
    - Template: m4.4xlarge
    - Security Group: default-aws-only-ssh-and-ssl
    - Ambari Server: unchecked

5. Click Review and Launch.
6. Click Create and start cluster.

Installing HiBench

1. Log into your client host.
2. Change to the root user:

```
sudo su -
```

3. Install the prerequisite packages to the client:

```
yum install -y git vim numpy blas64 lapack64
echo "/usr/hdp/current/hadoop-client/lib/native" > /etc/ld.so.conf.d/hadoop-client-native.conf
```

4. Install maven to the client:

```
sudo sed -i s/\$releasever/6/g /etc/yum.repos.d/epel-apache-maven.repo
sudo yum install -y apache-maven
```

5. Make sure to change Java to OpenJDK 8 by typing the following commands and selecting the OpenJDK version:

```
update-alternatives --config java
update-alternatives --config javac
```

6. Change to the HDFS user, create the HiBench directory in the dfs, and set permissions:

```
su - hdfs
hdfs dfs -mkdir /HiBench
hdfs dfs -chown -R root:hadoop /HiBench
hdfs dfs -mkdir /user/root
hdfs dfs -chown root /user/root
exit
```

7. Edit the bash profile to automatically include the HiBench prerequisites:

```
vim ~/.bash_profile
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk
export HADOOP_HOME=/usr/hdp/current/hadoop-client
export SPARK_HOME=/usr/hdp/current/spark2-client
export KAFKA_HOME=/usr/hdp/current/kafka-broker
export LD_LIBRARY_PATH=/usr/hdp/current/hadoop-client/lib/native
```

8. Download the HiBench benchmark from GitHub:

```
cd /root
git clone https://github.com/intel-hadoop/HiBench.git
```

9. Change to the newly downloaded HiBench directory, and type the following command:

```
mvn -Dspark=2.1 -Dscala=2.11 clean package | tee hibench_build.log
```

10. Modify the 99-user_defined_properties.conf file by changing the following lines:

```
hibench.hadoop.home              /usr/hdp/current/hadoop-client
hibench.spark.home               /usr/hdp/current/spark2-client
hibench.hadoop.mapreduce.home    /usr/hdp/current/hadoop-mapreduce-client
```

Configuring Hortonworks Data Platform

Using the Ambari console, we made the following changes to the settings to the services:

- HDFS
  - DataNode
  - DataNode directories - each of the drives on the server
  - DataNode failed disk tolerance - 2
- YARN
  - Memory
    - Memory allocated for all YARN containers on a node – 256 GB
    - Maximum container Size (Memory) – 262144 MB
- CPU
  - Percentage of physical CPU allocated for all containers on a node - 100%
    - Number of virtual cores – 64
    - Maximum Container Size (VCores) – 64

Configuring HiBench

We used the following settings in testing:

hibench.conf	Dell testbed	AWS testbed
hibench.default.map.parallelism	832	1024
hibench.default.shuffle.parallelism	832	1024
hibench.scale.profile	bigdata	bigdata
spark.conf	Dell testbed	AWS testbed
hibench.yarn.executor.num	52	64
hibench.yarn.executor.cores	4	4
spark.executor.memory	12g	14g
spark.driver.memory	12g	14g

Running HiBench

1. Log into the client host.
2. Change to the root user:
```
sudo su -
```
3. In the HiBench directory, navigate to the bin/workload/[benchmark] directory.
4. Initialize the data:
```
./prepare/prepare.sh
```
5. Log into the first data node.
6. Clear the PageCache, dentries, and inodes:
```
sudo echo 3 > /proc/sys/vm/drop_caches
```
7. For the remaining data nodes, complete steps 5 and 6.
8. After clearing the cache, wait five minutes.
9. Log into the client host.
10. Change to the root user:
```
sudo su -
```
11. In the HiBench directory, navigate to the bin/workload/[benchmark] directory.
12. Run the benchmark:
```
./spark/run.sh
```
13. When the test completes, record the results.

Appendix C: Cost analysis

Overview

We collected cost data between December 8 and December 12, 2017.

We estimated the cost to support running the tested workload for three years on an in-house Dell EMC PowerEdge FX2s chassis with four Dell EMC PowerEdge FC640 compute modules and on all-upfront, reserved AWS EC2 instances with EBS storage volumes. We include costs only for the systems under test as defined in Appendix A. The following tables show the costs for the two solutions and the savings for the Dell EMC solution. The section following the tables provides more information and assumptions.

Dell EMC PowerEdge FX2s solution	
Item	Price
Hardware list price	$118,391.67
Red Hat Enterprise Linux Server software	$15,588.00
Datacenter power and cooling	$6,983.38
Datacenter space	$1,118.57
Server administration	$3,262.59
Total estimated costs for Dell EMC solution	$145,344.21

AWS solution	
Item	Price
AWS per instance (three-year estimate)	$251,455.60

Comparison	
Savings for Dell EMC solution	**$106,111.39**
Savings (percent) for Dell EMC solution	**42.19%**

By changing the drives of the AWS instances from io1 storage to gp2 storage and expanding them to 1,500 GB each, it's possible for a gp2-backed AWS solution to cost less than the io1-backed AWS solution we tested and offer a similar IOPS SLA. For our analysis, choosing this configuration would change the three-year estimated price of an AWS solution to $182,432.80, meaning an organization that chooses the Dell EMC solution could save $37,088.59 over three years, a 20.33 percent savings.

Description of costs

Dell EMC PowerEdge FX2s solution costs

Hardware

Item	Quantity	Unit Price	Total
PowerEdge FN I/O module, factory installed	2	$2,376.83	$4,753.66
PowerEdge FX2s chassis configuration with Flexible IO (up to 8 PCIe Slots)	1	$9,662.01	$9,662.01
PowerEdge FC640 compute modules	4	$25,994.00	$103,976.00
Total			**$118,391.67**

We used the list prices for the Dell EMC PowerEdge FX2s chassis with two PowerEdge FN I/O modules, factory installed, and four Dell EMC PowerEdge FC640 compute modules. Those prices included Dell ProSupport™ Mission Critical subscriptions for the compute modules and Dell ProSupport Plus for the chassis and I/O modules. We included Dell EMC iDRAC9 Enterprise with OpenManage™ Essentials Server Configuration Management.

We tried to match our on-premises infrastructure hardware and AWS infrastructure as closely as possible. However, AWS did not offer the exact same configuration. We assumed infrastructure machines would not affect the big data capabilities of either solution, so we did not include their costs in the analysis. The solutions had the same number of infrastructure machines.

We used the Dell EMC online store to get prices for the PowerEdge FC640 compute modules and support, matching the configuration to the units we tested. We requested a quote from Dell EMC sales for prices of the PowerEdge FX2s and I/O module and their support because those prices were not available online. We used the following configuration of the PowerEdge FC640 compute modules for pricing:

Chassis	Up to two SATA or SAS hard drives (passthrough or unconfigured drives, PERC required)
Processor	2 x Intel Xeon Gold processors 5120 (2.2G, 14C/28T, 10.4GT/s 2UPI, 19M Cache, Turbo, HT (105W) DDR4-2400)
Memory	12 x 16GB RDIMM, 2666MT/s, dual rank
RAID controller	H330
Hard drives	2 x 800GB SSD SAS write-intensive 12Gb 512n 2.5" hot-plug drive (PX05SM,10 DWPD,14600 TBW)
Embedded systems management	iDRAC9 Enterprise with OpenManage Essentials, Server Configuration Management
PCIe riser	PowerEdge FC PCIE mezzanine adapter for enablement of FX2s
Internal SD module	Internal dual-SD module with 2 x 16GB SD cards
Network daughter card for fabric A	Intel X710 dual-port, 10Gb KR blade network daughter card
Hardware support services	Three years of ProSupport Plus Mission Critical 4Hr Onsite Service

Software

We included the cost of 12 one-year subscriptions (one per year for three years for the four servers) for Red Hat Enterprise Linux Server, Premium (Physical or Virtual Nodes), which are priced at $1,299 in the Red Hat store.

Datacenter power and cooling

We measured the Dell EMC solution's power consumption with a power analyzer while it ran under load. We calculated the energy cost for that usage, assuming the systems were running 24/7 for three years. We used a cost per KwH of 10.63 cents. We calculated a cost to cool the equipment at 70 percent of the power cost.

Datacenter space

We used a VMware estimate of rack space costs based on 24 square feet per rack at a cost of $270 per square foot. Space costs can vary widely based on factors such as geography, datacenter size, and datacenter density.

Server administration

Administrator effectiveness ranges widely based on level of automation, uniformity of systems, administrator experience, and the amount of hardware support provided by the vendor. We included Dell ProSupport with Mission Critical response time for the compute modules and Dell ProSupport Plus for the chassis and I/O modules. We assumed this is one FX2s among many in the datacenter, which the administrator can manage together. We also assumed that as a whole the datacenter is well-managed with automated procedures. Those assumptions would keep administrator time to a minimum. We estimated one full-time admin can support 100 fully loaded FX2s chassis and calculated administration costs based on one percent of the three-year total compensation of a Server Administrator II.

AWS solution costs

We used pricing for the US-East/US Standard (Virginia) datacenter in the AWS Simple Monthly Calculator to estimate the cost of the Amazon EC2 instances and storage instances we used in testing. We included business-level AWS support.

Amazon EC2 instances

Instances	Usage	Type	Billing option
4	100% utilized/month	Linux on M4.16xlarge	Three-year, all-upfront reserved

Amazon EBS volumes

Volumes	Volume type	Storage (GB)	IOPS
8	Provisioned IOPS SSD (Io1)	600	4,500
4	General Purpose SSD (gp2)	30	

Pricing estimate

Service type	Components	Component price	Service price (includes one-time and monthly fees)
Amazon EC2 Service (US-East)	Compute:	$0.00	
	EBS Volumes (monthly fee):	$612.00	
	EBS IOPS (monthly fee):	$2,340.00	
	Reserved Instances (one-time fee):	$126,480.00	
			$129,432.00
AWS Support (Business)	Support for all AWS services (monthly fee):	$295.20	
	Support for Reserved Instances (one-time fee):	$8,076.40	
			$8,371.60

Fee	Cost
Total one-time fee	$134,556.40
Total monthly fee	$3,247.20

We multiplied the monthly fee by 36 (i.e. three years of payments) and added that sum to the total one-time fee for a three-year cost of $251,455.60. We did not include any in-house administration costs for the AWS instances.

This project was commissioned by Dell EMC.

Principled Technologies®

Facts matter.®