Accenture Technology Labs

Hadoop Deployment Comparison Study

Price-performance comparison between a bare-metal Hadoop cluster and Hadoop-as-a-Service

accenture

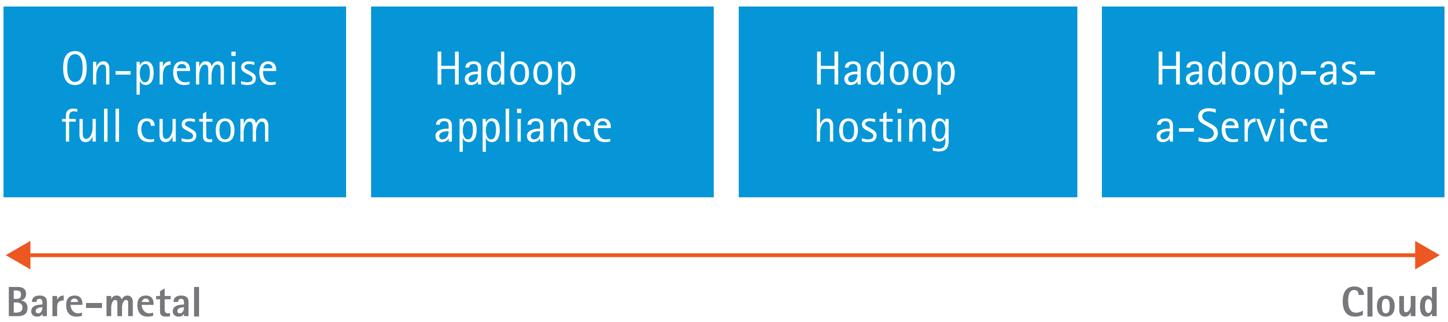
consulting | technology | outsourcing

Introduction

Big data technology changes many business organizations' perspective on data. Traditionally, a data infrastructure was something of a gatekeeper for data access; such infrastructures were built as self-contained, monolithic "scale up" appliances. Any additional scale required additional resources, which often exponentially increased cost. Big data platforms, on the other hand, can increase capacity and performance by adding nodes at a linear cost increase. This shift unlocks a world of potential, allowing businesses to mine their data for greater insights by combining different sets of data to gain a broader view of consumer behaviors and interests as well as operational improvements such as supply chain optimization. For that reason, big data has become a buzzword, and many business organizations list "big data rollouts" among their top-priority IT initiatives.

One of the key technologies that have been at the center of the big data technology landscape is Apache Hadoop™. This product provides an ecosystem with multiple tools—notably, Hadoop Distributed File System (HDFS™) and Hadoop MapReduce—that store and process large datasets in a scalable and cost-effective way.

Figure 1. The spectrum of Hadoop deployment options



When enterprises adopt Hadoop, one of the decisions they must make is the deployment model. There are four options as illustrated in Figure 1:

- **On-premise full custom.** With this option, businesses purchase commodity hardware, then they install software and operate it themselves. This option gives businesses full control of the Hadoop cluster.
- **Hadoop appliance.** This preconfigured Hadoop cluster allows businesses to bypass detailed technical configuration decisions and jumpstart data analysis.
- **Hadoop hosting.** Much as with a traditional ISP model, organizations rely on a service provider to deploy and operate Hadoop clusters on their behalf.
- **Hadoop-as-a-Service.** This option gives businesses instant access to Hadoop clusters with a pay-per-use consumption model, providing greater business agility.

To determine which of these options presents the right deployment model, organizations must consider five key areas. The first is the price-performance ratio, and it is the focus of this paper. The Hadoop-as-a-service model is typically cloud-based and uses virtualization technology to automate deployment and operation processes (in comparison, the other models typically use physical machines directly).

There have existed two divergent views related to the price-performance ratio for Hadoop deployments. One view is that a virtualized Hadoop cluster is slower because Hadoop's workload has intensive I/O operations, which tend to run slowly on virtualized environments. The other view is that the cloud-based model provides compelling cost savings because its individual server node tends to be less expensive; furthermore, Hadoop is horizontally scalable.

The second area of consideration is data privacy, which is a common concern when storing data outside of corporate-owned infrastructure. Cloud-based deployment requires a comprehensive cloud-data privacy strategy that encompasses areas such as proper implementation of legal requirements, well-orchestrated data-protection technologies, as well as the organization's culture with regard to adopting emerging technologies. Accenture Cloud Data Privacy Framework outlines a detailed approach to help clients address this issue.

The third area is data gravity. Once data volume reaches a certain point, physical data migration becomes prohibitively slow, which means that many organizations are locked into their current data platform. Therefore, the portability of data, the anticipated future growth of data, and the location of data must all be carefully considered.

A related and fourth area is data enrichment, which involves leveraging multiple datasets to uncover new insights. For example, combining a consumer's purchase history and social-networking activities can yield a deeper understanding of the consumer's lifestyle and key personal events and therefore enable companies to introduce new services and products of interest. The primary challenge is that the storage of these multiple datasets increases the volume of data, resulting in slow connectivity. Therefore, many organizations choose to co-locate these datasets. Given volume and portability considerations, most organizations choose to move the smaller datasets to the location of the larger ones. Thus, thinking strategically about where to house your data, considering both current and future needs, is key.

The fifth area is the productivity of developers and data scientists. They tap into the datasets, create a "sandbox" environment, explore the data analysis ideas, and deploy them into production. Cloud's self-service deployment model tends to expedite this process.

Out of these five key areas, Accenture assessed the price-performance ratio between bare-metal Hadoop clusters and Hadoop-as-a-Service on Amazon Web Services™. (A bare-metal Hadoop cluster refers to a Hadoop cluster deployed on top of physical servers without a virtualization layer. Currently, it is the most common Hadoop deployment option in production environments.)

For the experiment, we first built the total cost of ownership (TCO) model to control two environments at the matched cost level. Then, using Accenture Platform Benchmark as real-world workloads, we compared the performance of both a bare-metal Hadoop cluster and Amazon ElasticMapReduce (Amazon EMR™).

Employing empirical and systemic analyses, Accenture's study revealed that Hadoop-as-a-Service offers better price-performance ratio. Thus, this result debunks the idea that the cloud is not suitable for Hadoop MapReduce workloads, with their heavy I/O requirements. Moreover, the benefit of performance tuning is so huge that cloud's virtualization layer overhead is a worthy investment as it expands performance tuning opportunities. Lastly, despite of the sizable benefit, the performance tuning process is complex and time-consuming, thus requires automated tuning tools.

In our study, we conducted the price-performance comparison of a bare-metal Hadoop cluster and Hadoop-as-a-Service at a matched TCO and using real-world applications. The following sections provide a wealth of information: they detail the TCO model we developed for both a bare-metal Hadoop cluster and Hadoop-as-a-Service; illustrate the Accenture Data Platform Benchmark, which is a suite of three real-world Hadoop MapReduce applications; explain the experiment setup and result; and discuss the findings.

Total Cost of Ownership

For businesses, it is more meaningful to compare the performance at the matched budget than at the matched hardware specification. Therefore, it is important to understand the TCO of the Hadoop deployments that we compared.

In this TCO analysis, we list the TCO components along with various factors needed to calculate the cost of each component. Calculating the TCO of a generic Hadoop cluster is a challenging—perhaps even impossible—task, because it involves factors that are unknown or that vary based on time. Given that, we put our best efforts into including representative numbers and being specific about the assumptions we made. Moreover, for comparison, we calculated the monthly TCO and translated capital expenditures into monthly operating expenses.

As stated earlier, we compared two Hadoop deployment options at the matched TCO budget. Table 1 illustrates the methodology we used to match the TCO. We first picked a bare-metal Hadoop cluster as a reference and calculated its TCO, which was $21,845.04 per month. Then, using $21,845.04 as the monthly TCO for Hadoop-as-a-Service, we allocated that budget to the necessary components and derived the resulting cloud-based capacity so that we could compare the performance of the two deployment options.

We excluded from comparison components that are difficult to quantify and agnostic to the deployment type, such as the staff personnel cost of data scientists and business analysts.

Table 1. TCO matching on bare-metal Hadoop cluster and Amazon EMR

Bare-metal	Monthly TCO $21,845.04		Hadoop-as-a-Service
Staff for operation	$9,274.46	$3,091.49	Staff for operation
Technical support (third-party vendors)	$6,656.00	$1,372.27	Technical support (service providers)
Data center facility and electricity	$2,914.58	$2,063.00	Storage services
Server hardware	$3,000.00	$15,318.28	Hadoop service

Bare-metal Hadoop Cluster

The left half of Table 1 shows the monthly TCO breakdown of bare-metal Hadoop clusters. We picked the cluster size with 24 nodes and 50 TB of HDFS capacity. In practice, it is a reference point for small-scale initial production deployment. The following subsections explain each cost component and the assumptions we used.

Server hardware

In the TCO calculation, we estimated the hardware cost at $4,500 per node based on retail server hardware vendors. The modeled server node assumes four 2 TB hard disk drives, 24 GB memory, and 12 CPU cores. Also, this pricing includes a server rack chassis and a top-of-rack switch. Of course, multiple factors could change the given pricing, such as a different hardware configuration, a volume discount on a purchase, or regional or seasonal price discrimination.

To calculate the monthly TCO, we had to translate the one-time capital expense of the hardware purchase into the monthly operating cost. This translation typically uses a straight line depreciation method—even distribution of the capital cost across a period of time. For the sake of comparison, we chose three years as the distribution period, which is one of the most commonly used periods for server hardware. However, the best period to use is debatable because of many influential factors, such as the expected lifetime of the hardware as well as the organization's asset-depreciation policy and its technology-refresh strategy.

Data center facility and electricity

We budgeted $2,914.58 for the data center facility and electricity. For the data center facility, we assumed a tier-3 grade data center with a 10,000-square-foot building space including 4,000 square feet of IT space at the construction cost of $7,892,230. We used a 25-year straight line depreciation method to translate it to operating cost. For electricity, we budgeted $252,565 per year, assuming 720 kW total power load at $0.09 per kWh. It includes both power and cooling of the entire facility, such as servers, storage and network, and failover power sources. Also, we budgeted $701,546 per year for building maintenance. In total, the annual facility TCO was $1,269,800.

We further assumed that 40 percent of the IT space is allocated for server racks, which is 70 percent actively occupied. With a rack with a capacity of 24 1U servers and a 30-square-foot footprint, the annual facility TCO is shared by 1,344 servers, the cost of which is $944.79. We also budgeted $1,500 per rack hardware that is shared by 24 servers with a five-year depreciation cycle, and $500 per data center switch cost per server per year. Taking all these factors into account, we budgeted $1,457.29 per node per year, which translates into $2,914.58 per month for the targeting 24-node cluster.

The assumptions we made above are heavily based on Gartner Inc. reports.[1]

Technical support from third-party vendors

Hadoop is an open-source product. Users may run into bugs or technical issues, or they may desire custom features. Even though anyone can patch the Hadoop project in theory, doing so requires a deep understanding of Hadoop architecture and implementation. For enterprise customers seeking production deployment of Hadoop, this is a significant risk. To meet the need for troubleshooting, Hadoop distributors typically offer technical support in the form of annual subscriptions per server node.

The retail pricing of an annual subscription is typically not publicly shared. However, Cloudera, Inc. has shared its retail pricing, and we used it in our study, with Cloudera's permission. In particular, we used the retail pricing of Cloudera Enterprise Core: $3,382 per node per year with 24/7 support.

Staff for operation

A Hadoop cluster requires various operational tasks because it comes with the complexity of distributed systems. The Hadoop cluster should be deployed on reasonably chosen hardware and tuned with appropriate configuration parameters. It also requires cluster health monitoring and failure recovery and repair.[2] In addition, as workload characteristics change over time, the cluster also needs to be re-tuned. Also, the job schedulers should be controlled and configured to keep the cluster productive. Furthermore, because Hadoop is an evolving product, users must keep up with current Hadoop versions and integrate new tools in the Hadoop ecosystem as needed. Finally, the underlying infrastructure itself should be managed and kept available, which typically requires IT and system administration support.

There is no publically available data point for Hadoop operation staff FTE cost, yet. The closest one we could find was Linux Server FTE cost data published by Gartner.[3] Based on the data, one Linux Server FTE personnel can manage 28.3 servers, and the associated cost is $130,567 on average. Based on these assumptions, we budgeted $ 9,274.46 for operation staff personnel cost.

Hadoop-as-a-Service (Amazon EMR)

Hadoop-as-a-Service refers to a cloud service that allows users to deploy Hadoop clusters on demand, run MapReduce jobs, and tear down the clusters when the jobs are completed. Of the multiple providers in the space, we chose Amazon EMR™ for its popularity and maturity. In this section, we explain the TCO breakdown of our Hadoop-as-a-Service deployment using Amazon EMR, along with the assumptions we used.

Staff for operation (cloud administrator)

We budgeted $3,091.49 for cloud-related internal operation staff personnel cost, which is one-third of its bare-metal counterpart. Using a service provider like Amazon EMR shifts a large portion of operational burden to that provider. For example, Amazon EMR deploys a fully configured Hadoop cluster with a few inputs from users such as the cluster's instance type and count.

However, based on our experience with Amazon EMR and other cloud vendors, we recognize the need for an internal role: a cloud administrator. A cloud administrator monitors the health of a company's assets that are deployed to the cloud as well as the cloud itself, makes trouble-shooting decisions, tunes the Hadoop cluster parameters for performance, owns the technical relationship with cloud service providers, and keeps up with newly offered features, to name a few of the responsibilities of the role.

Technical support from service providers

Amazon EMR has lower but similar technical risks that the enterprise may need to address. Amazon Web Services (AWS) provides four levels of premium support. Among them, we found the "business" grade to be most comparable to the support level we assumed for its bare-metal counterpart. Instead of an annual per node subscription-pricing model, AWS charges an amount proportionate to the account's monthly spending. Given the remaining budget and based on Amazon premium support pricing, this support costs $1,372.27.

Storage services (Amazon S3)

When using cloud, storing original input data and final output data in the cloud, rather than in HDFS, has multiple benefits. First, users can reduce the server instance cost by tearing down the Hadoop cluster when not running MapReduce jobs. Second, multiple Hadoop clusters can easily run analyses on the dataset in parallel without interfering with one another's performance. This approach, however, limits the data locality and thus may slow the job execution.

In calculating the required volume for cloud storage, we assumed 50 percent occupancy of the available HDFS space in the bare-metal cluster, because users need spare room when planning the capacity of bare-metal clusters. First of all, a bare-metal cluster needs extra HDFS space to hold temporary data between cascaded jobs. Second, it needs extra local temporary storage to buffer the intermediate data shuffled between map tasks and reduce tasks. Lastly, it needs to reserve room for future data growth given that a bare-metal cluster does not expand instantaneously. An Amazon EMR cluster, on the other hand, comes with local storage and HDFS and thus does not need space in Amazon S3 for temporary storage. Also, Amazon EMR clusters do not need to be overprovisioned for future growth. Based on Amazon S3 pricing, this storage costs $2,063.00.

Hadoop service (Amazon EMR)

After considering all the above components, the given budget leaves $15,318.28 for Amazon EMR spending.

Currently, Amazon EMR supports 14 EC2 instance types and three pricing options. Because it is both time- and cost-prohibitive to run the benchmark on all 42 combinations, we selected a few representative ones. First, we picked three instance types: standard extra large (m1.xlarge), high-memory quadruple extra large (m2.4xlarge), and cluster compute eight extra large (cc2.8xlarge). Each of them represents its instance family in that they are the largest instance type of each instance family and a multiple of smaller ones in the family. We excluded three instance families for specific reasons: high-CPU instances have a CPU-memory resource ratio that is not balanced for our MapReduce workloads; cluster GPU instances are expensive when GPU resources are not used; high-storage instances are expensive when 48 TB of local storage space on each instance is not needed.

In calculating the number of affordable instances, we used the following assumptions. First, we assumed 50 percent cluster utilization, which means the percentage of time that the EMR cluster is on. In the context of a pay-per-use model, a higher utilization assumption leads to a smaller number of affordable instances for a given budget. Second, we assumed a three-year heavy utilization contract for reserved instances; this is the least expensive given the aforementioned cluster-utilization assumptions. Third, we used a snapshot of spot-price history (March 2013) to calculate the average savings from using spot instances, which are listed in Table 2. Fourth, we assumed that we would allocate only one-third of the Amazon EMR budget for spot instances and the rest for reserved instances. The last assumption is based on our earlier experience in provisioning Amazon EMR clusters with all spot instances: the clusters were often so large that AWS ran out of available spot slots. It led AWS to increase the price of the affected instance type, which resulted in diminished cost savings or job failures due to the termination of a significant portion of the cluster. Using these assumptions, we calculated the number of affordable instances for nine different deployment options, which are listed in Table 3.

Table 2. Average spot instance cost compared with on-demand instances

Instance type	Average cost compared with on-demand instances
m1.xlarge	20%
m2.4xlarge	14%
cc2.8xlarge	14%

Table 3. Affordable number of instances

Instance type	On-demand instances (ODI)	Reserved instances (RI)	Reserved + Spot instances (RI + SI)
m1.xlarge	68	112	192
m2.4xlarge	20	41	77
cc2.8xlarge	13	28	53

Accenture Data Platform Benchmark

The Accenture Data Platform Benchmark suite comprises multiple real-world Hadoop MapReduce applications. Within Accenture Technology Labs, we have been fortunate to directly monitor enterprise clients' business needs and to solve their real-world business problems by leveraging big data platform technologies, including Hadoop. On the basis of such client experience, our internal road map, and published literature, we assembled the suite of Hadoop MapReduce applications, which we named Accenture Data Platform Benchmark. We used the following selection process. First, we categorized and selected common use cases of Hadoop MapReduce applications: log management, customer preference prediction, and text analytics. Then, from each category, we implemented a representative and baseline workload with publicly available software packages and public data. This strategy makes the benchmark agnostic to any of our clients' custom design and thus easier to share, while keeping it relevant. The rest of this section introduces three workloads in the benchmark suite.

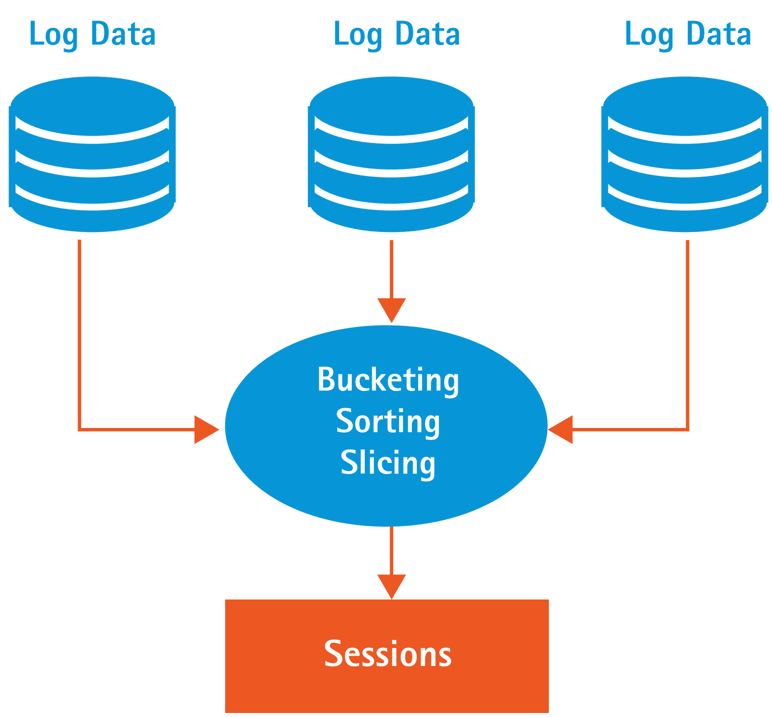
Sessionization

In the context of log analysis, a session is a sequence of related interactions that are useful to analyze as a group. The sequence of web pages through which a user navigated is an example of a session. Sessionization is one of the first steps in many types of log analysis and management, such as personalized website optimization, infrastructure operation optimization, and security analytics.

Sessionization is a process of constructing a session, taking raw log datasets from multiple sources. It reads a large number of compressed log files, decompresses and parses them, and buckets the log entries by a session holder identifier (for example, by user ID). Then, within each bucket, it sorts the entries by time order, and finally slices them into sessions based on a time gap between two consecutive logged activities.

We used synthesized large log datasets whose entries had a timestamp, user ID, and the log information content (140 random characters, in this case). The application relies on user ID for bucketing, timestamp for sorting, and 60 seconds as an implied session boundary threshold. For the study, we used about 150 billion log entries (~24 TB) from 1 million users and produced 1.6 billion sessions.

Figure 2. Sessionization



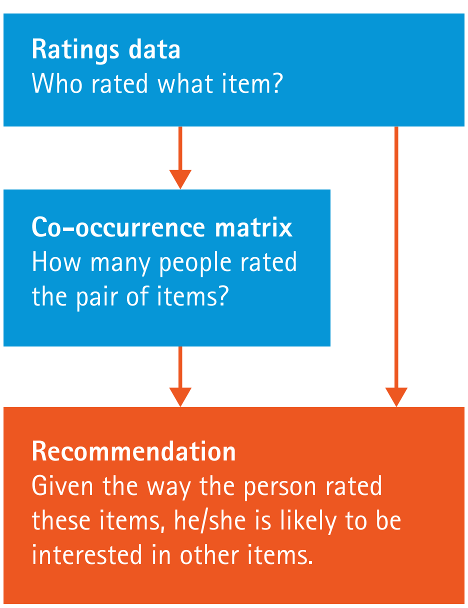
Recommendation engine

A recommendation engine is one of the most popular instantiations of customer preference prediction. Many industries—including retail, media content providers, and advertising—use recommendation engines to predict the unexpressed preference of customers and further stretch revenue potential.

While there are many algorithms and use cases for recommendation engines, we used an item-based collaborative filtering algorithm and a movie recommendation engine as a reference. It reads a history of movie ratings from multiple users regarding multiple movies. Then, it builds a co-occurrence matrix that scores the similarity of each pair of movies. Combining the matrix and each user's movie-rating history, the engine predicts a given user's preference on unrated movies.

We used the collaborative filtering example in the Apache Mahout project. Moreover, we used synthesized movie ratings data from 3 million users on 50,000 items, with 100 ratings per user on average.

Figure 3. Recommendation engine using item-based collaborative filtering



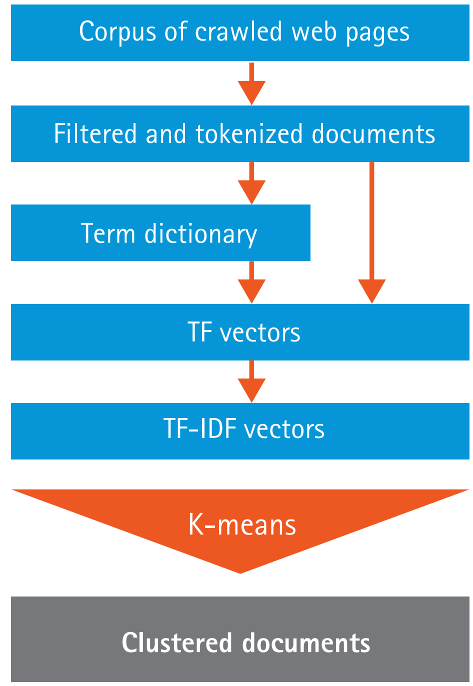
Document clustering

Document clustering is one of the important areas in unstructured text analysis. It groups a corpus of documents into a few clusters. The document clustering, as well as its building blocks, has been popularly used in many areas, such as search engines and e-commerce site optimization.

The application starts with a corpus of compressed crawled web pages. After decompression, it reads out and parses each html document, then extracts tokenized terms but filters out unnecessary words ("stop words"). Next, it builds a term dictionary: a set of pairings of a distinct term and its numerical index. Using this term dictionary, it maps each tokenized document to its corresponding term frequency (TF) vector, which lists the occurrence of each term in the document. To enhance the precision of the clustering, it normalizes these TF vectors into term frequency—inverse document frequency (TF-IDF) vectors. Finally, taking these TF-IDF vectors, it runs a k-means clustering algorithm to cluster the documents.

We used a crawled web page dataset publicly available from the Common Crawl project hosted in Amazon S3. Given the size of clusters undergoing testing, we used 3 TB of compressed data (10 TB uncompressed) or 300 million web pages.

Figure 4. Document clustering



Experiment Setup

For the bare-metal Hadoop deployment, we used the Hadoop hosting service from MetaScale[4], a Sears Holdings subsidiary. The cluster we used for study has a client node, a primary NameNode, a secondary NameNode, a JobTracker node, and 22 worker-node servers, each of which runs a DataNode for HDFS as well as TaskTracker for MapReduce. Table 4 lists the detailed hardware specification of the cluster. We preloaded the input data to its HDFS and stored the output and the intermediate data between cascaded jobs, also in HDFS.

As mentioned, for cloud-based deployment, we used Amazon EMR. We ran the benchmark workloads on nine different cluster configurations, as listed in Table 3. We preloaded the input data to Amazon S3, and directed output data to Amazon S3, while keeping the intermediate data between cascaded jobs in HDFS.

This resulted in 10 cluster setups with 3 benchmark workloads to run on each. We did performance tuning for each combination of the workload and the cluster setup. We tuned them by both applying manual tuning techniques and getting help from an automated performance-tuning tool. The tuned performance results are shown in Figure 5, Figure 6, and Figure 7 and discussed in the next section.

Table 4. Hardware specification of the bare-metal cluster

Cluster in total	
Client node	1
Master nodes	3
Worker/Data nodes	22
Cores	264
memory (GB)	528
Raw TB	176
HDFS TB Available	148.7
Usable TB (w/ 3 replicas)	49.6
Wokers/Data node summary	
Model	Dell R415
CPU type	Opteron 4180
# of cores	12
Clock speed (GHz)	2.6
Memory (GB)	24
Memory bus speed (MHz)	1333
# of disks	4
Each disk's capacity (TB)	2
Total capacity (TB)	8

Experiment Result

In our study, we conducted the price-performance comparison of a bare-metal Hadoop cluster and Hadoop-as-a-Service at the matched TCO using real-world applications.

Figure 5, Figure 6, and Figure 7 show the execution time comparison of a bare-metal Hadoop cluster and nine different options from Amazon EMR. In most cases, we could find at least one AWS instance type that resulted in better price-performance ratio with an on-demand instance (ODI) pricing option, which is the most expensive one. Moreover, clusters with two other cheaper pricing options (a reserved instance option only or combined with spot instance option) significantly outperformed the bare-metal cluster across all workloads. In practice, companies start to use reserved instances and spot instances heavily as they transition from a pilot mode into a production rollout.

This result debunks the idea that the cloud is not suitable for Hadoop MapReduce workloads given their heavy I/O requirements. Hadoop-as-a-Service provides a better price-performance ratio than the bare-metal counterpart.

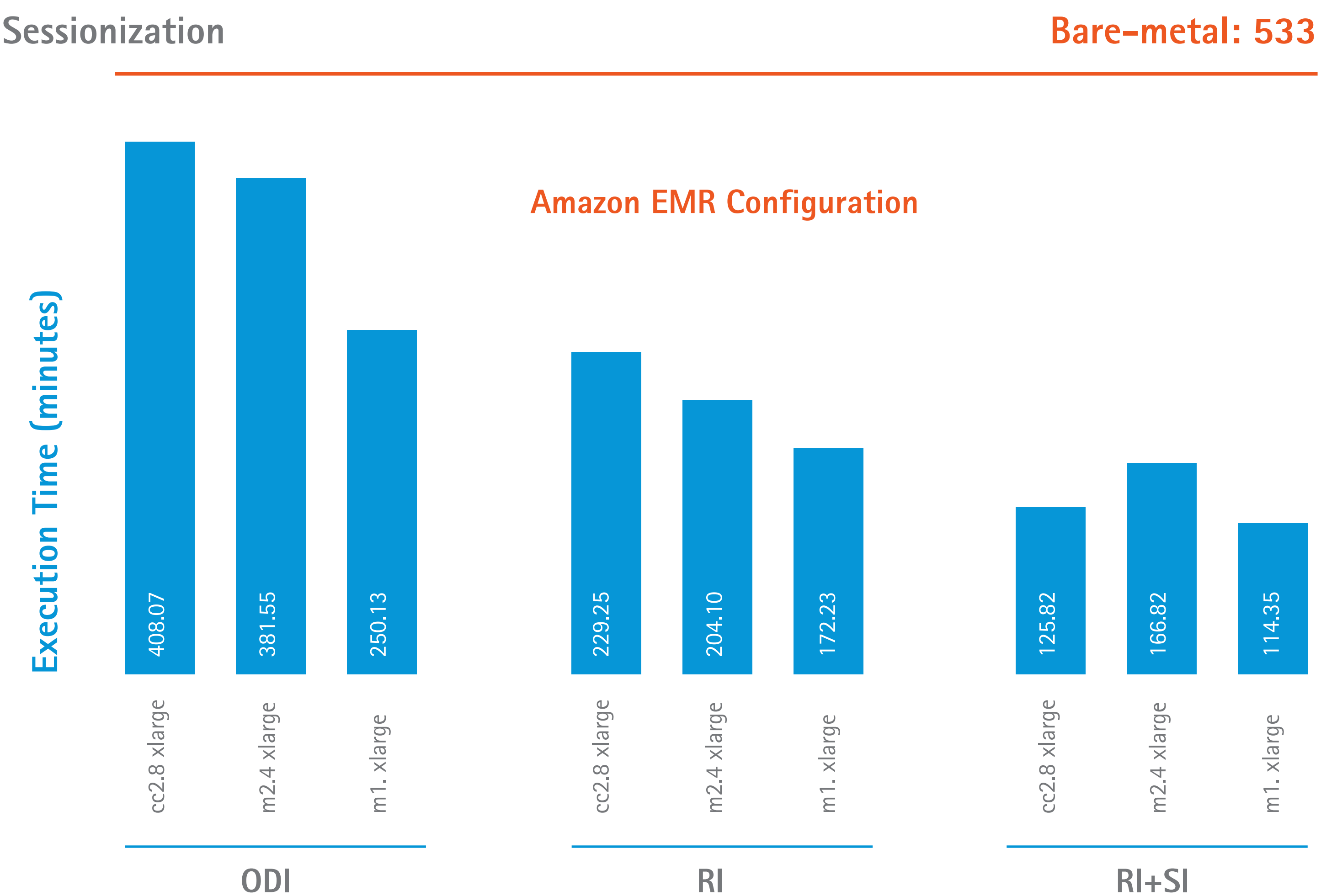
Sessionization

As shown in Figure 5, all nine Amazon EMR clusters outperformed the bare-metal Hadoop cluster. The result is notable for two reasons.

First, it demonstrates that the impact of S3, the remote storage, is not a critical factor in real-world applications. S3 impacts Sessionization the most because Sessionization has only one MapReduce job, whose output data size is almost identical to the input data size.

Second, it shows the potential benefits of choosing a job-specific hardware configuration. Because Sessionization only rearranges the dataset, it requires large heap size per task slot and benefits from data compression. An m1.xlarge instance type outperformed other types because it provided the best memory per CPU core ratio and CPU core per dollar ratio.

Figure 5. Execution time comparison for the Sessionization workload

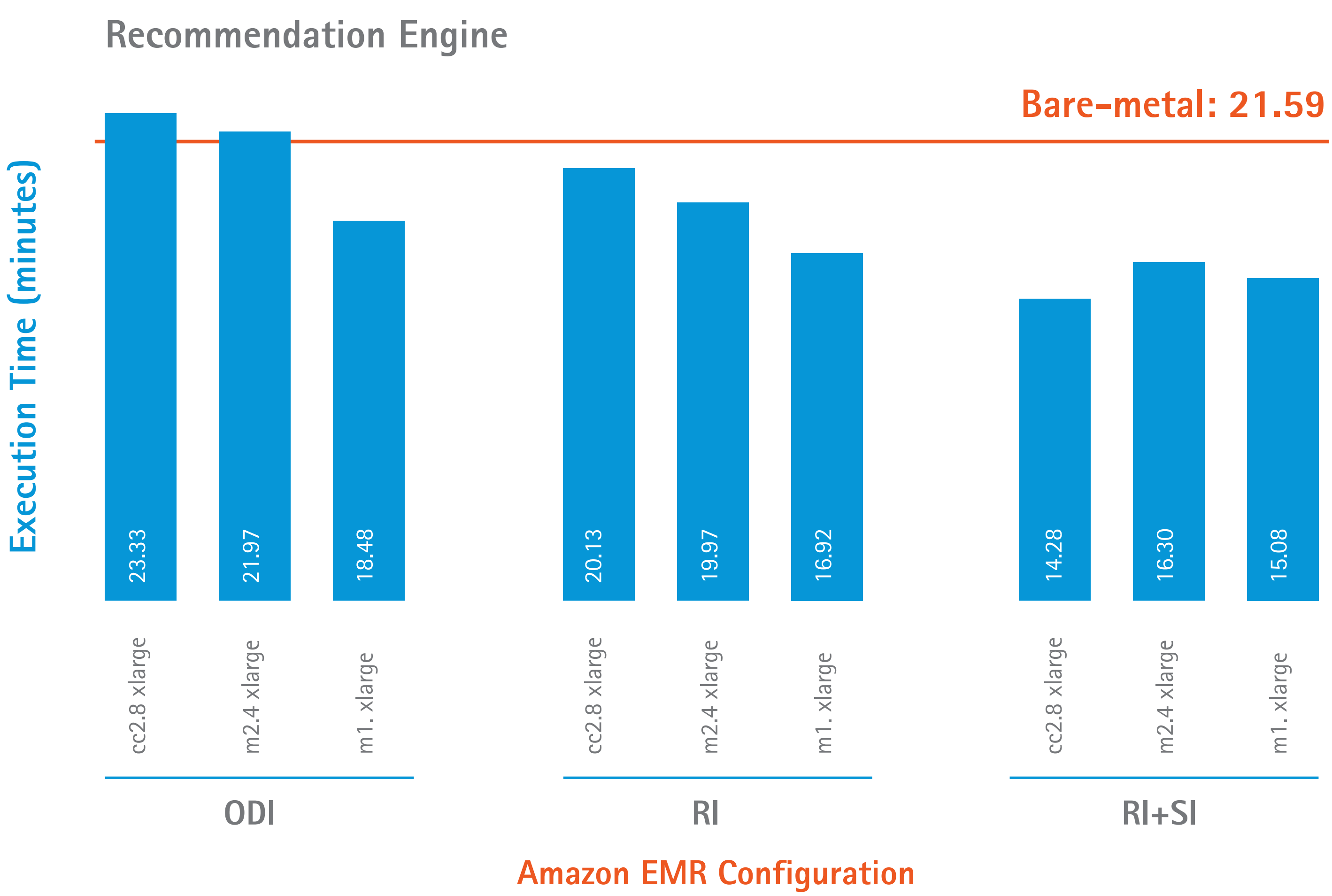


Recommendation Engine

Figure 6 shows that there are one or more instance types per pricing option that outperform the bare-metal Hadoop cluster. With the ODI option, Amazon EMR outperforms the bare-metal cluster by 17 percent with the m1.xlarge instance type.

The Recommendation Engine workload is affected by the MapReduce runtime framework overhead because it goes through 11 cascaded MapReduce jobs in about 20 minutes and processes the relatively small dataset.

Figure 6. Execution time comparison for the Recommendation Engine workload

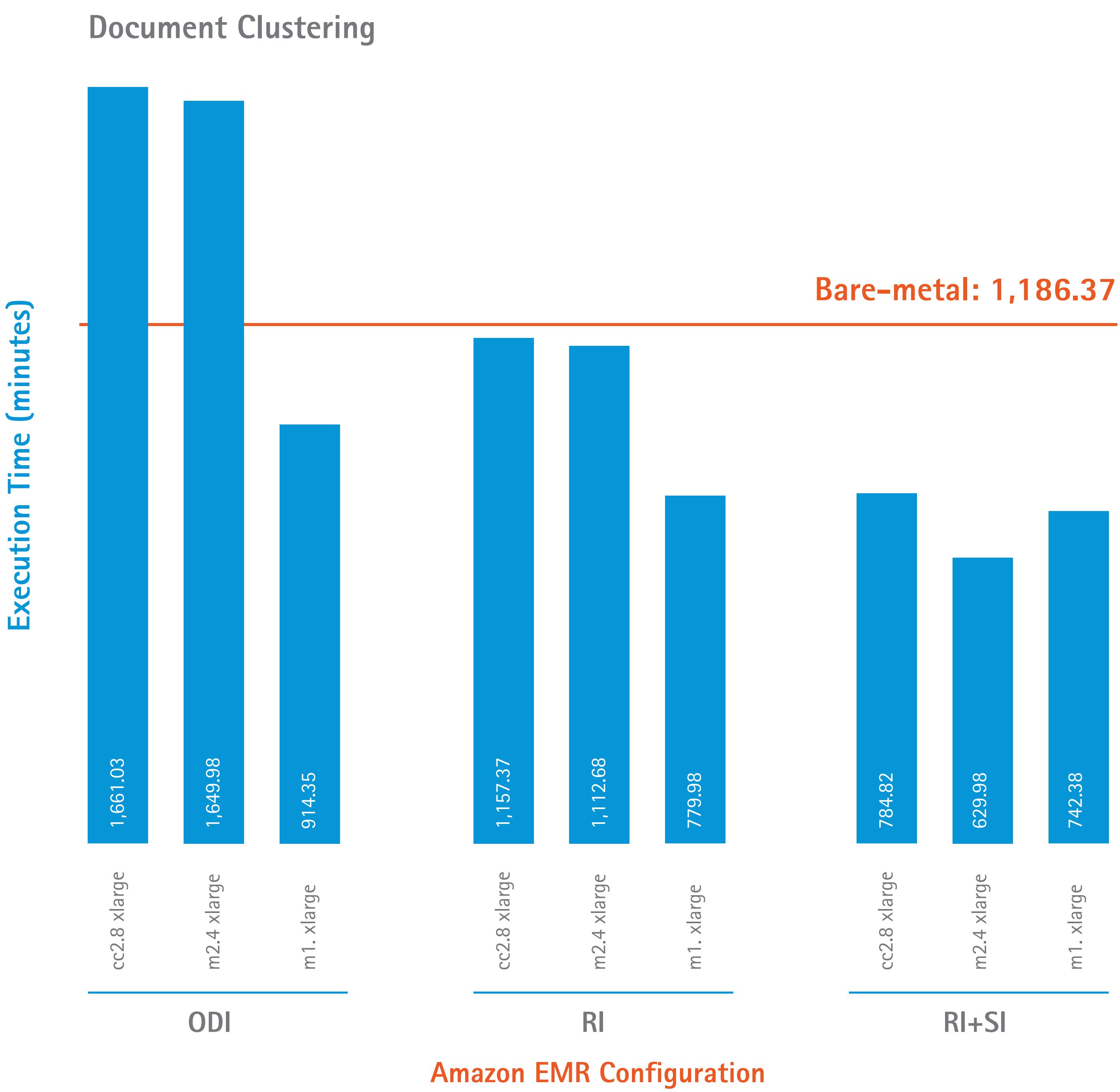


Document Clustering

We observed the same trend again. Figure 7 shows that there are one or more instance types per pricing option that outperform the bare-metal Hadoop cluster. With the ODI option, Amazon EMR outperforms the bare-metal cluster by 30 percent with the m1.xlarge instance type.

The Document Clustering workload benefits from a large heap size to hold the word dictionary. As seen in Sessionization, m1.xlarge outperformed because it provided the best memory per CPU core ratio and CPU core per dollar ratio.

Figure 7. Execution time comparison for the Document Clustering workload



Discussion

Performance impact: I/O virtualization overhead and performance tuning

We have tried to observe whether the overhead critically impacted the overall performance when running real-world applications, and thus might influence the deployment decision. Our study did not focus on measuring the precise I/O virtualization overhead. However, multiple technical reports have been published that claim the I/O virtualization overhead can be minimized to a negligible level with techniques like pass-through mode. Also, our experiment environment is not designed to compare two clusters that are identical but for a virtualization layer.

Throughout the study, we learned that the I/O virtualization overhead that comes with the cloud environment is a worthy investment to enable optimization opportunities relevant only to the cloud.

Before we discuss the optimization opportunities, we need to highlight the significance of the performance tuning. We consistently observed that the performance improvement by applying various tuning techniques made a huge impact. For instance, the Sessionization workload did not even complete its execution with the default configuration of the Hadoop cluster. However, a minimal tuning made it complete in 21 hours and 40 minutes. Further rounds of aggressive performance tuning improved its execution time to 9 hours and 11 minutes. We experienced a speed up of eight times when we tuned the Recommendation Engine workload.

Certain optimization opportunities are relevant only to cloud. First, cloud offers a variety of hardware options from which users can choose on the basis of their job-specific requirements. For example, memory space per CPU core impacts the maximum per-task heap space allocation that Sessionization requires much more than the Recommendation Engine workload does.

Second, in the cloud-based Hadoop cluster, cluster-wide parameters (for example, the number of map task slots and reduce task slots per TaskTracker node) can be tuned specifically to the workload. Traditionally, bare-metal Hadoop cluster administrators tune these parameters not for an individual workload's performance but for cluster-wide resource scheduling and throughput because the cluster is shared by multiple workloads. However, because cloud-based Hadoop clusters are increasingly deployed on-demand and are dedicated to a specific workload, tuning these cluster-wide parameters becomes more relevant. For example, we tuned the concurrent map task slots and reduce task slots differently per application.

Automated performance tuning

Even though we can improve the performance significantly by applying various tuning techniques, doing so is typically a time-consuming and iterative process that requires deep expertise in Hadoop. For example, the manual tuning of the Sessionization workload that was mentioned above took more than two weeks overall; each iteration took about a half to a full day including performance analysis, tuning, and execution.

Moreover, the tuned parameters for one dataset may not be optimal for another dataset. In the cloud environment, the tuning is even more challenging because underlying infrastructure performance varies dynamically. When running workloads with a multiple of cascaded MapReduce jobs such as Recommendation Engine or Document Clustering, it becomes almost unmanageable.

To minimize the burden of performance tuning, we used an automated performance tuning tool called Starfish. It is a tool that collects performance logs from a profile run and automatically generates an optimized MapReduce job configuration parameter set. With Starfish, we could minimize the manual analysis and tuning iterations and achieve significant performance improvement. For instance, on the bare-metal Hadoop cluster, Starfish optimized the Recommendation Engine workload (with 11 cascaded MapReduce jobs) and improved the performance by eight times relative to the default parameter settings within a single performance tuning iteration.

Conclusion

In our study, we conducted a price-performance comparison of a bare-metal Hadoop cluster and Hadoop-as-a-Service. Based on the TCO model that we developed, we calculated the TCO of a 24-node 50 TB-capacity bare-metal Hadoop cluster and derived the capacity of nine different cloud-based Hadoop clusters at the matched TCO. Then, we ran three real-world Hadoop applications, Accenture Data Platform Benchmark, to compare their performance of these clusters.

Our study concludes with three notable insights. First, Hadoop-as-a-Service offers a better price-performance ratio. Second, the benefit of performance tuning is so huge that cloud's virtualization layer overhead is a worthy investment as it expands performance-tuning opportunities. Third, despite the sizable benefit, the performance-tuning process is complex and time-consuming and thus requires automated tuning tools.

Contact:

Scott W. Kurth
Director – Accenture Technology Labs
scott.kurth@accenture.com

Michael E. Wendt
R&D Developer –
Accenture Technology Labs
michael.e.wendt@accenture.com

References

1. David J. Cappuccio, "Use a TCO Model to Estimate the Costs of Your Data Center," June 2012. Gartner.

2. Hadoop tolerates failures but does not necessarily fix the failures.

3. Jamie K. Guevara, et al., "IT Key Metrics Data 2013: Key Infrastructure Measures: Linux Server Analysis: Current Year," December 2012, Gartner.

4. http://www.metascale.com

About Accenture

Accenture is a global management consulting, technology services and outsourcing company, with approximately 261,000 people serving clients in more than 120 countries. Combining unparalleled experience, comprehensive capabilities across all industries and business functions, and extensive research on the world's most successful companies, Accenture collaborates with clients to help them become high-performance businesses and governments. The company generated net revenues of US$27.9 billion for the fiscal year ended Aug. 31, 2012. Its home page is www.accenture.com.

About Accenture Technology Labs

Accenture Technology Labs, the dedicated technology research and development (R&D) organization within Accenture, has been turning technology innovation into business results for more than 20 years. Our R&D team explores new and emerging technologies to create a vision of how technology will shape the future and invent the next wave of cutting-edge business solutions. Working closely with Accenture's global network of specialists, Accenture Technology Labs help clients innovate to achieve high performance.

The Labs are located in Silicon Valley, California; Sophia Antipolis, France; Arlington, Virginia; Beijing, China and Bangalore, India.

For more information, please visit: www.accenture.com/accenturetechlabs.

mc429