DATA LAKES FOR DATA SCIENCE

Integrating Analytics Tools with Shared Infrastructure for Big Data

ABSTRACT

This paper examines the relationship between three primary domains of an enterprise big data program: data science, analytics frameworks, and IT infrastructure. A decision about tools or infrastructure in one domain can affect, and potentially limit, what can be done in the other domains. This paper shows how the practices of data science and the use of analytics frameworks, such as Hadoop and Spark, generate a set of requirements that a big data storage system must fulfill in order to serve the needs of data scientists, application developers, and infrastructure managers.

May 2015

REDEFINE

EMC WHITE PAPER

EMC²

To learn more about how EMC products, services, and solutions can help solve your business and IT challenges, contact your local representative or authorized reseller, visit www.emc.com, or explore and compare products in the EMC Store

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

The information in this publication is provided “as is.” EMC Corporation makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose.

Part Number H14214

TABLE OF CONTENTS

INTRODUCTION

The burgeoning field of data science is fusing with the new business requirement to store and analyze big data, resulting in a conceptual gulf between the practice of data science and the complexities of information technology. The gap leaves both data scientists and solution architects unsure of how to implement a big data storage system that serves the needs of the business, the analytics applications, the data scientists, and the system administrators.

To help bridge that conceptual gulf, this white paper examines the relationship between three primary domains of an enterprise big data program: data science, analytics frameworks, and IT infrastructure. The data science pipeline spans all three of these domains as engineers and data scientists collect, clean, aggregate, model, and interpret data. Big data complicates the pipeline by introducing several problems: the heterogeneity, scale, and timeliness of data as well privacy and collaboration.

The data science pipeline and the problems that accompany it shape the technology that must be applied in each domain. A decision about tools or infrastructure in one domain can affect, and potentially limit, what can be done in the other domains.

This paper demonstrates how data-management tasks and analytics tools generate a set of requirements that a big data storage system must fulfill in order to serve the needs of data scientists, application developers, and infrastructure managers. A distributed multitenant storage system delivers the scalability and flexibility that an enterprise big data program demands.

DATA SCIENCE

After setting the stage with a brief discussion of the paradigm shift to big data, this section presents a brief overview of the data science pipeline, the processing that supports it, and the problems that underlie it. In particular, this section highlights some aspects of data science and data management that shape decisions about the informational technology that will support the science. These aspects of data science and data processing have a direct bearing on the technologies, applications, and options discussed in this paper.

PARADIGM SHIFT

The paradigm shift to big data and data science is characterized in part by moving from monitoring the present to predicting the future. Although monitoring remains a best practice for many businesses, accurate forecasting can expand the business: Modeling and analyzing the past behavior of your customers, for instance, can identify the customers likely to buy a product during a given time so that you can proactively cultivate their business.

> *"The big data of this revolution is far more powerful than the analytics that were used in the past. We can measure and therefore manage more precisely than ever before. We can make better predictions and smarter decisions. We can target more-effective interventions, and can do so in areas that so far have been dominated by gut and intuition rather than by data and rigor," Andrew McAfee and Erik Brynjolfsson say in the Harvard Business Review.*[1]

In fact, the research of Andrew McAfee, Erik Brynjolfsson, and their colleagues found that

> *"the more companies characterized themselves as data-driven, the better they performed on objective measures of financial and operational results. In particular, companies in the top third of their industry in the use of data-driven decision making were, on average, 5% more productive and 6% more profitable than their competitors. This performance difference remained robust after accounting for the contributions of labor, capital, purchased services, and traditional IT investment. It was statistically significant and economically important and was reflected in measurable increases in stock market valuations."*

BARRIERS TO CHANGE

Much of the information technologies and infrastructure—such as business intelligence tools and data warehouses—that monitor and report on business activity fall short of fulfilling the promise of big data in two ways:

[1] Big Data: The Management Revolution, by Andrew McAfee and Erik Brynjolfsson, October 2012 Issue of Harvard Business Review. https://hbr.org/2012/10/big-data-the-management-revolution/ar

1. A data warehouse or relational database management system is not capable of scaling to handle the volume and velocity of big data and does not satisfy some key requirements of a big data program, such as handling unstructured data. The schema-on-read requirements of an RDMS impede the storage of a variety of data.
2. BI tools like SQL can become costly ways to analyze big data with the intention of predicting the future or prescribing actions.

Another challenge impedes businesses from progressing to the big data model: The mindset of business stakeholders.

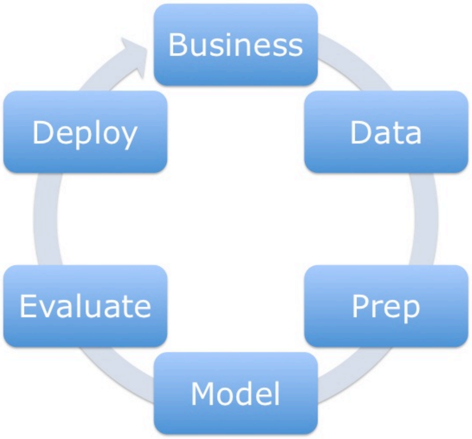
> *"Organizations, especially the business stakeholders, are having a hard time transitioning from 'monitoring the business' to 'predicting the business.' ... But now we need the business stakeholders to start thinking about predictive questions (e.g., What will happen?) and prescriptive questions (e.g., What should I do?)," Bill Schmarzo writes in his blog on big data, where he provides a road map to shift from the old mindset of business intelligence to the new model of data science.*[2]

Although a shift in the mindset of business stakeholders is important, predictive and prescriptive analysis depends on big data—datasets so large that they overwhelm the scale of traditional data processing systems and tools, such as data warehouses and business intelligence software.

MATURITY OF DATA COLLECTION AND INDUSTRY STANDARDS

At a basic level, the maturity of an enterprise's big data program can be measured at the intersection of data, analysis, and technology. An organization that does not collect data is obviously not in a position to analyze it. An immature organization has started to collect large datasets but has yet to begin analyzing them. A somewhat mature organization collects its data and analyzes it by using big data technologies such as Apache Hadoop. A mature organization collects data, cleans and integrates it, applies statistical modeling techniques to it with scientific rigor, analyzes it with big data technologies, and exploits the results to optimize the business. Industry standards offer a benchmark for determining the maturity of an enterprise’s big data program.

Standards can not only help a business gauge its maturity level but also prompt business stakeholders to adapt their mindset and processes to new methodologies. One such standard that can help business stakeholders understand the kinds of shifts in thinking that data science requires is the Cross Industry Standard Process for Data Mining (CRISP-DM). There are six high-level phases in the CRISP-DM model[3], which, as will become apparent in the next section, are similar to those of the data science pipeline:



1. Business understanding
2. Data understanding
3. Data preparation
4. Modeling
5. Evaluation
6. Deployment

Each of these phases includes generic tasks and outputs. In the first phase, determining your business objectives and your data-mining goals are two important starting points—the analysis of a dataset starts with a goal to achieve or a problem to solve.

[2] https://infocus.emc.com/william_schmarzo/dont-think-better-think-different/

[3] CRISP-DM 1.0: Step-by-Step Data Mining Guide. Pete Chapman (NCR), Julian Clinton (SPSS), Randy Kerber (NCR), Thomas Khabaza (SPSS), Thomas Reinartz (DaimlerChrysler), Colin Shearer (SPSS) and Rüdiger Wirth (DaimlerChrysler). Page 12. At http://the-modeling-agency.com/crisp-dm.pdf.

Another standard is the IDC Big Data and Analytics (BDA) MaturityScape Framework.[4] In "IDC MaturityScape Benchmark: Big Data and Analytics in North America," IDC presents research that can help businesses assess the maturity of their big data analytics programs against industry benchmarks—benchmarks that can prioritize technology and investments to improve decision making.[5]

The success with which an organization can implement and bring to maturity an enterprise data science program separates it from competitors. According to *Data Science for Business*, a study conducted by economist Prasanna Tambe of NYU's Stern School examined the extent to which big data technologies seem to help firms.

Tambe's research found that,

> *"...after controlling for various possible confounding factors, using big data technologies is associated with significant additional productivity growth. Specifically, one standard deviation higher utilization of big data technologies is associated with 1%–3% higher productivity than the average firm; one standard deviation lower in terms of big data utilization is associated with 1%–3% lower productivity. This leads to potentially very large productivity differences between the firms at the extremes."*[6]

Maturity level and technology investments aside, once you have a business objective to fulfill and problem to solve, there are two basic points from which to start a data science program for big data:

1. Analyzing data that has already been collected and stored.
2. Collecting and storing the data that you want to analyze.

Much of the work of data scientists focuses on acquiring and cleaning up large datasets, modeling them, and then analyzing them in such a way that their predictions and prescriptions are relevant, accurate, repeatable, and actionable. The nature of this process gives rise to the data science pipeline and the problems that accompany it.

THE DATA SCIENCE PIPELINE

A community white paper on big data written by a group of researchers and published by the Computing Research Association describes the phases of a data science pipeline[7] like this:

1. Data acquisition and recording.
2. Information extraction and cleaning.
3. Data integration, aggregation, and representation.
4. Query processing, data modeling, and analysis.
5. Interpretation of the results.

Here is an example of how these phases come together to solve a problem. Let's say you've obtained corporate sponsorship to travel the United States to visit the ski resorts with the best snow. Your objective is to ski the resort with the best conditions on each weekend of the season. The problem, then, is predicting by Thursday which ski resort will have the best conditions over the weekend. Making the prediction by Thursday gives you enough time to get there by Saturday.

The first step to making data-driven decisions is acquiring data. You can obtain all the current weather data as well as the forecasts from the National Oceanic and Atmospheric Administration. In addition, you can get climate data from the U.S. government's publicly accessible datasets at Data.gov. Although both datasets can help predict the snowfall and other weather factors, the two sets fall short of being able to predict the best skiing conditions. The best conditions depend on other factors, like the amount and type of snow, whether it's icy or soft, and so forth. So

[4] The standard is described in IDC MaturityScape: Big Data and Analytics—A Guide to Unlocking Information Assets (IDC Doc # 239771, March 2013).

[5] IDC MaturityScape Benchmark: Big Data and Analytics in North America, Dec. 2013, IDC Doc # 245197. http://www.idc.com/getdoc.jsp?containerId=245197

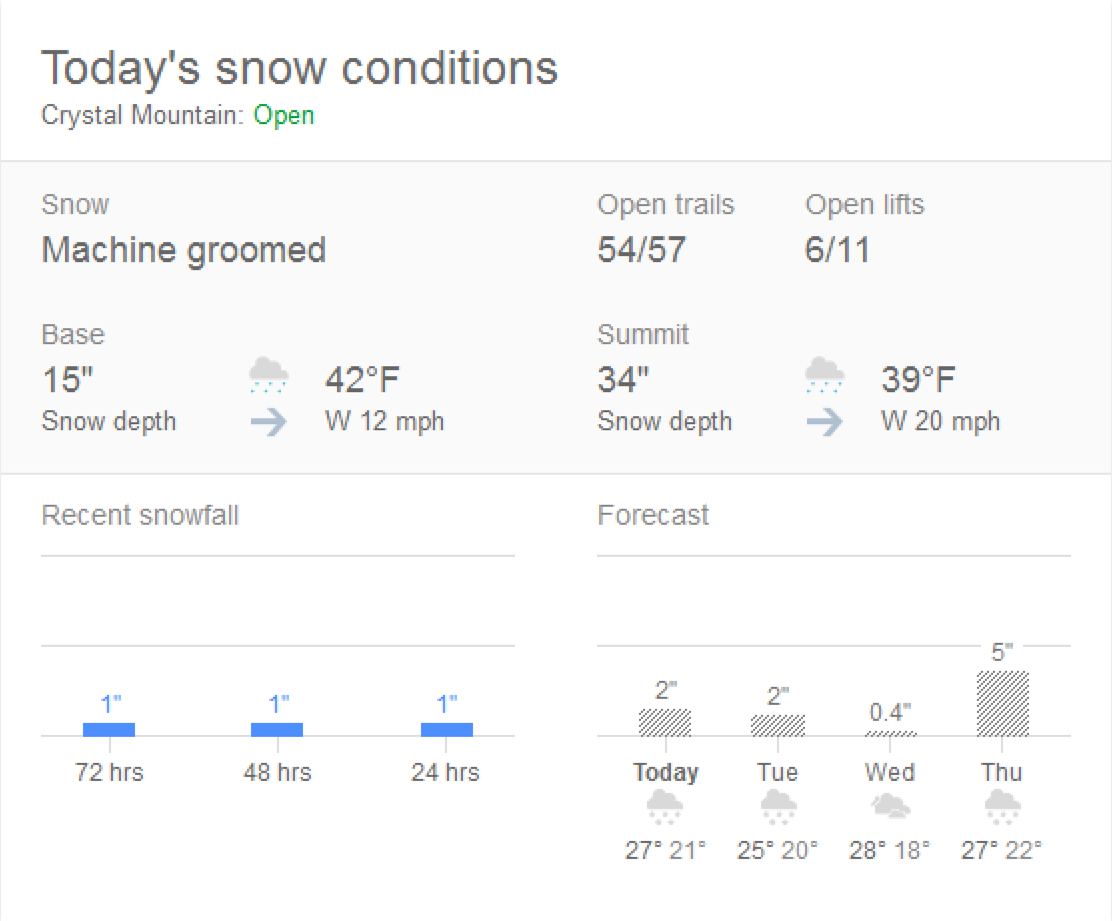
[6] Data Science for Business, by Foster Provost and Tom Fawcett, (O'Reilly), 2013.

[7] Challenges and Opportunities with Big Data. A community white paper developed by leading researchers across the United States. Divyakant Agrawal, UC Santa Barbara Philip Bernstein, Microsoft Elisa Bertino, Purdue Univ. Susan Davidson, Univ. of Pennsylvania, et. al., Feb. 2012, published on the Computing Research Association web site at http://www.cra.org/ccc/files/docs/init/bigdatawhitepaper.pdf.

a third dataset is necessary: You must obtain through automation the current ski reports from all the resorts in the nation. It is the confluence of the current conditions, current weather, previous weather trends, and the weather forecast that will help predict the best conditions. All this data must be stored in a large storage system that can easily and cost-effectively scale to handle the new data that comes in every day.

Once the data is acquired, you must extract the information relevant to your problem and clean it up so you can work with it productively. According to a recent article in The New York Times, the janitorial work of cleaning up and wrangling data into a usable form is a critical barrier to obtaining insights. "Data scientists, according to interviews and expert estimates, spend from 50 percent to 80 percent of their time mired in this more mundane labor of collecting and preparing unruly digital data, before it can be explored," the article says.[8]

Another key, then, is that the storage system housing the data must be readily accessible by data scientists so that they can visualize, manipulate, and automate the gathering and cleaning of disparate datasets. During nearly every stage of the data science pipeline, data scientists must be able to explore and modify the dataset with data management tools.



Today's snow conditions

Crystal Mountain: Open

Snow
Machine groomed

Open trails
54/57

Open lifts
6/11

Base
15"
Snow depth
42°F
W 12 mph

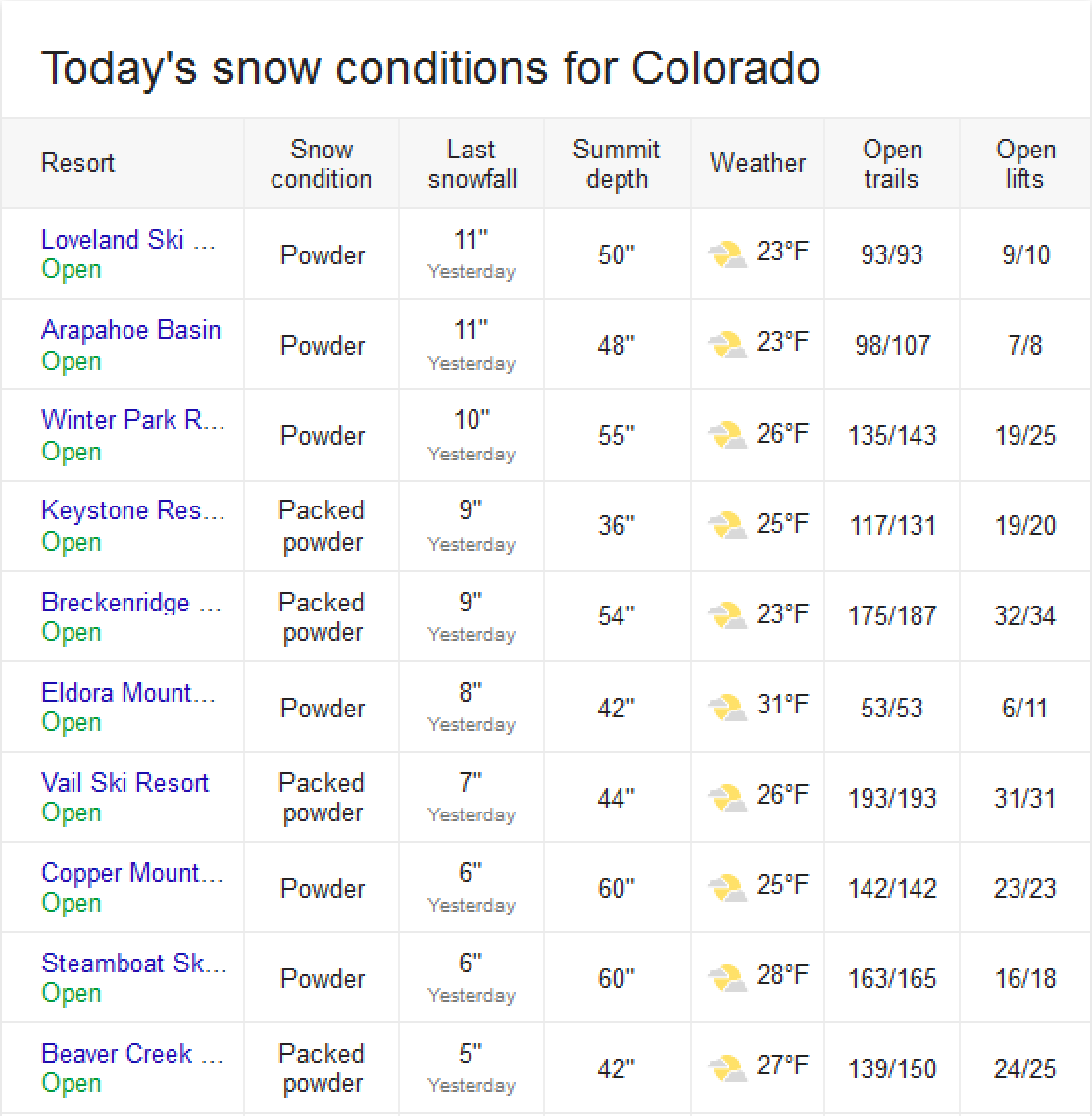
Summit
34"
Snow depth
39°F
W 20 mph

The next step in the pipeline is aggregation. The datasets must be combined and aggregated in a way that is likely to solve the problem. The system in which the data is stored must provide the flexibility to combine, recombine, aggregate, and represent data in different ways.

Once aggregated, the interesting work begins: modeling the data and analyzing it with analytics tools capable of using statistical methods, machine learning, and other advanced algorithms. The scale of the data dictates that

[8] For Big-Data Scientists, 'Janitor Work' Is Key Hurdle to Insights, By STEVE LOHRAUG. 17, 2014, http://www.nytimes.com/2014/08/18/technology/for-big-data-scientists-hurdle-to-insights-is-janitor-work.html?_r=0.

these tasks will be much easier if the data does not need to be copied or migrated from the storage system into another system for analysis—workflows that fall under the umbrella of extract, transform, and load (ETL). The most efficient and cost-effective place to analyze the data is in a storage system that works with big data technologies and performs well enough to make in-place analytics feasible.



Today's snow conditions for Colorado

Resort	Snow condition	Last snowfall	Summit depth	Weather	Open trails	Open lifts
Loveland Ski ... Open	Powder	11" Yesterday	50"	⛅ 23°F	93/93	9/10
Arapahoe Basin Open	Powder	11" Yesterday	48"	⛅ 23°F	98/107	7/8
Winter Park R... Open	Powder	10" Yesterday	55"	⛅ 26°F	135/143	19/25
Keystone Res... Open	Packed powder	9" Yesterday	36"	⛅ 25°F	117/131	19/20
Breckenridge ... Open	Packed powder	9" Yesterday	54"	⛅ 23°F	175/187	32/34
Eldora Mount... Open	Powder	8" Yesterday	42"	⛅ 31°F	53/53	6/11
Vail Ski Resort Open	Packed powder	7" Yesterday	44"	⛅ 26°F	193/193	31/31
Copper Mount... Open	Powder	6" Yesterday	60"	⛅ 25°F	142/142	23/23
Steamboat Sk... Open	Powder	6" Yesterday	60"	⛅ 28°F	163/165	16/18
Beaver Creek ... Open	Packed powder	5" Yesterday	42"	⛅ 27°F	139/150	24/25

The final step is the interpretation of the results—which requires distributing the results to others and collaborating (or competing) in understanding them.

Although the data science pipeline bears some similarity to the scientific method—become curious enough about something to ask a question about it, conduct preliminary research, formulate a hypothesis, test it, analyze your results, and present them if they are newsworthy—there are a set of problems specific to big data that set the data science pipeline apart from the scientific method.

PROBLEMS WITH THE DATA SCIENCE PIPELINE

According to the Computing Research Association white paper on big data cited earlier, five key problems accompany the phases of the data science pipeline. These problems shape the analytics frameworks that you apply to data and the IT infrastructure in which you manage and store the data.

1. The heterogeneity of data.
2. The scale of data.
3. The timeliness of data.
4. Privacy, security, and compliance.
5. Human collaboration.

The collection and storage of disparate datasets, even when the datasets relate to the same phenomenon, ultimately results in data from different sources. Meantime, the volume of data and the increasing rate at which it is growing can overwhelm a storage system or data warehouse that does not easily scale.

The timeliness of the data can be important in at least two ways, as use cases like preventing credit card fraud demonstrate: Ingestion should occur as the data is generated, and analysis and interpretation should immediately follow. Timeliness requires high-performance data ingestion.

Finally, during the phases of the data science pipeline, the privacy of sensitive or personally identifiable information must be protected; in some use cases, such as analyzing health information or credit card purchases, privacy is mandated by compliance regulations like the Health Insurance Portability and Accountability Act (HIPAA) and the Payment Card Industry Data Security Standard (PCI DSS).

COLLECTING AND STORING DATA

With data streams and large datasets, which can grow from terabytes to petabytes, ingesting data into a scalable storage system is often a prerequisite to formal analysis. For a big data program driven by a research question or a well-defined problem, it will likely be necessary to rapidly ingest data from many sources—the Internet, social media, devices, applications, data repositories, and data warehouses. To capture large streams of data from multiple sources simultaneously, the storage system must ingest the data at high-performance speeds.

The data, too, is likely to be heterogeneous: It will reside in formats that range from structured data in traditional databases and semistructured data in XML or JSON files to unstructured data in text or log files. And the datasets might be augmented with multimedia: video, images, and audio files.

The heterogeneity of data and its sources yields a requirement for the system that ingests the data: interoperability. The system must be flexible enough to let data scientists ingest data from different sources by using common data transfer protocols, including HTTP, FTP, SMB, and NFS. In addition, the Hadoop Distributed File System, HDFS, is becoming increasingly important, especially for working with large datasets.

Here is an example of the kind of interoperability that acquiring data might entail. The Natural Language Toolkit is a platform for building Python programs to work with human language data. The toolkit's web site includes dozens of corpora and trained models that can be downloaded with HTTP at http://www.nltk.org/nltk_data/.

One corpus is selections from Project Gutenberg, which digitizes books in the public domain as plain text and ebooks and makes them available to the public on the Internet. The Natural Language Toolkit web site offers about 4 MB of the selections for download with HTTP:

```
http://www.nltk.org/nltk_data/packages/corpora/gutenberg.zip
```

To store the dataset in a location that provides access to other users and applications, an SMB share can be created on an EMC Isilon network-attached storage system. You can, for example, connect to the system by SSH as an administrator and run the following commands:

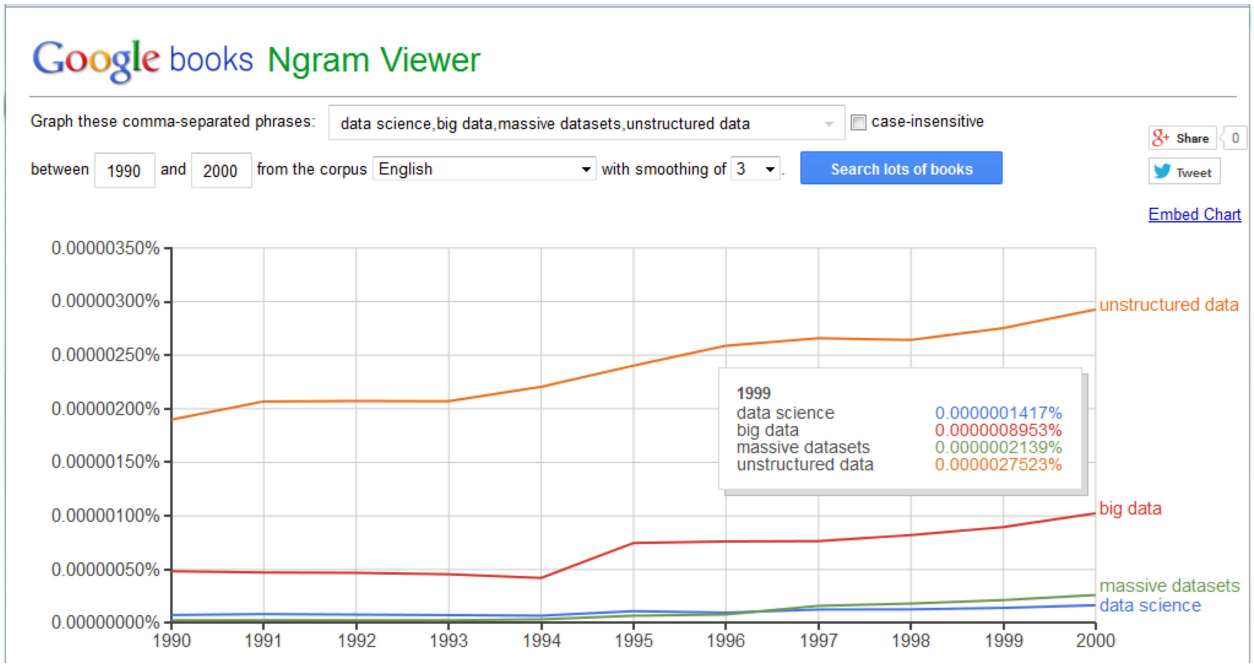
```
mkdir /ifs/isiloncluster01/zone1/analytics
isi smb shares create --name=analytics --path=/ifs/isiloncluster01/zone1/analytics --
browsable=true --description="Analytics Share"
```

After being downloaded to a personal computer and then transferred to the share on the EMC Isilon storage system by using the SMB protocol, the small corpus now looks like this:

```
al-hfhb-1# ls /ifs/isiloncluster01/zone1/analytics
austen-emma.txt         bryant-stories.txt      chesterton-thursday.txt shakespeare-hamlet.txt
austen-persuasion.txt   burgess-busterbrown.txt edgeworth-parents.txt   shakespeare-macbeth.txt
austen-sense.txt        carroll-alice.txt       melville-moby_dick.txt  whitman-leaves.txt
blake-poems.txt         chesterton-ball.txt     milton-paradise.txt      chesterton-brown.txt
```

This dataset forms the humble beginning of a corpus of digitized books for analyzing linguistic and cultural trends. In 2011, a team of researchers published a ground-breaking paper in "culturomics"—the collection and analysis of massive amounts of data to study culture. The paper, “Quantitative Analysis Of Culture Using Millions Of Digitized Books,” helped establish a new type of evidence in the social sciences and humanities. The paper surveyed the linguistic and cultural phenomena reflected in English between 1800 and 2000 to show how analytics can reveal insights about such diverse fields as lexicography, grammar, collective memory, and the adoption of technology.[9]

The researchers created a corpus of 5,195,769 digitized books containing about 4 percent of all books ever published. The Google Books Ngrams dataset, as it has come to be known, is freely available on Amazon S3 in a file format suitable for Hadoop. The original dataset is available from http://books.google.com/ngrams/. The size of the corpus from Google Books, as stored on Amazon, is 2.2 terabytes (TB).[10]



Graph these comma-separated phrases: data science,big data,massive datasets,unstructured data ☐ case-insensitive

between 1990 and 2000 from the corpus English with smoothing of 3.

1999	
data science	0.0000001417%
big data	0.0000008953%
massive datasets	0.0000002139%
unstructured data	0.0000027523%

But if this dataset is just the beginning of a corpus of cultural texts for big data analysis, the size of the data will eventually yield a requirement for the storage system: scalability. Because an EMC Isilon cluster for analytics data can scale out to store as much as 50 petabytes, it is particularly well suited for storing big data.[11]

BUILDING A DATA SET

Project Gutenberg includes mirrors of the Gutenberg repository so that the books can be downloaded with FTP, which makes building the dataset easier. The Seattle mirror furnishes access to all the books at the following URL:

[9] Michel, Jean-Baptiste, Yuan Kui Shen, Aviva P. Aiden, Adrian Veres, Matthew K. Gray, The Google Books Team, Joseph P. Pickett, et al. 2011. Quantitative analysis of cutlure using millions of digitized books. Science 14 January 2011: Vol. 331 no. 6014 pp. 176-182. The full dataset, which comprises over two billion culturomic trajectories, can be explored at www.culturomics.org.

[10] See the datasets on the AWS web site; they are licensed under a Creative Commons Attribution 3.0 Unported License. https://aws.amazon.com/datasets/8172056142375670.

[11] Although an EMC Isilon cluster with HD400 nodes can store as much as 50 petabytes of data, the primary use case of the Isilon HD400 nodes is archiving, not analytics.

ftp://gutenberg.readingroo.ms/gutenberg/

But a faster method of building out the dataset with all the books in the Project Gutenberg collection is rsync, a program that can transfer and synchronize files between two computer systems. With the rsync transfer algorithm, a type of delta encoding minimizes network usage so that the entire Project Gutenberg corpus can be transferred to the storage system with minimum network disruption. The Open Science Data Cloud has stored the entire collection of Project Gutenberg—about 42,000 ebooks—and made it available for download through rsync as a 742 GB corpus of text data.[12] Because the EMC Isilon storage cluster on which the dataset is being stored supports rsync, the Gutenberg corpus can be transferred directly to the storage system:

```
rsync -avzuP globus.opensciencedatacloud.org::public/gutenberg/ /path/to/local_copy
rsync -avzuP globus.opensciencedatacloud.org::public/gutenberg/
/ifs/isiloncluster01/zone1/analytics
```

AUGMENTING A DATA SET

Bowling Alone: The Collapse and Revival of American Community, by Robert D. Putnam, a professor of public policy at Harvard University, is a landmark treatise in data-driven social science. For the book, which was published in 2001, Putnam and his team of researchers collected vast data about the social behavior of Americans. Three primary datasets were used extensively in the research reported in the book, and it is unlikely that one of the datasets alone would have sufficed to draw strong conclusions. In data science, augmenting a dataset with additional data can often help improve the accuracy and relevance of the results of an analysis.

The Project Gutenberg dataset can be augmented with 28 MB of documents that appeared on the Reuters news wire in 1987.[13] A tar.gz file containing the files was downloaded to a Linux computer and decompressed. To add files to the dataset on the storage system, an NFS export was created on the storage system:

```
isi nfs exports create /ifs/isiloncluster01/zone1/analytics --all-dirs=yes
```

The export was then mounted on a Linux workstation and used to transfer the files over NFS to the storage system:

```
sudo mount 192.0.2.11:/ifs/isiloncluster01/zone1/analytics /mnt
```

CONTINUOUSLY EXPANDING THE DATA SET WITH THE SWIFT PROTOCOL

The Cornell University Library furnishes open access to 1,009,394 e-prints in physics, mathematics, computer science, quantitative biology, quantitative finance, and statistics at arXiv.org. As a document submission and retrieval system that serves academic communities, arXiv disseminates manuscripts on cutting-edge research. The web site presents an API through which you can access all the manuscripts and their metadata.

By connecting to arXiv through its API, you can automatically query the publications by keyword and instantly download the metadata for each manuscript that matches the query. You can then add the metadata to the analytics dataset on the Isilon storage cluster by using the OpenStack Object Storage protocol, which is code-named Swift. The Swift protocol, coupled with a scale-out data lake, automates the collection of vast amounts of data for analysis.

The following code could be one of many automated queries that you run once a day to check for the latest research on a topic of interest and then ingest the metadata or manuscript into a dataset. This example, which was run on an Ubuntu Linux workstation, submits an API request to arXiv.org with Curl and then pipes the output into a Swift PUT command that stores the results in the analytics container on the Isilon cluster. (In the command, the -d @- syntax instructs the second curl command to read its input from the output of the first curl command.)

[12] See https://www.opensciencedatacloud.org/publicdata/ and https://www.opensciencedatacloud.org/publicdata/gutenberg/.
[13] The Reuters dataset is available at http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html.

```
curl "http://export.arxiv.org/api/query?search_query=all:culturomics&start=0&max_results=100" |
curl -d @- http://192.0.2.11:28080/v1/AUTH_admin/ifs/isiloncluster01/zone1/analytics/obj1.txt -H
"X-Auth-Token: AUTH_tka7f7e356r138a936fb35772xr0075a12c"  -v -X PUT
```

The query returned one result in the form of the following metadata[14] in XML , which is now stored on the Isilon cluster as obj1.txt:

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom">
<link href="http://arxiv.org/api/query?search_query%3Dall%3Aculturomics
      %26id_list%3D%26start%3D0%26max_results%3D100"
      rel="self" type="application/atom+xml"/>
<title type="html">ArXiv Query: search_query=all:culturomics&amp;
      id_list=&amp;start=0&amp;max_results=100</title> ...
<updated>2015-02-03T00:00:00-05:00</updated>
<opensearch:totalResults xmlns:opensearch="http://a9.com/-/spec/opensearch/1.1/">1
</opensearch:totalResults> ...
<entry> ...
<published>2012-02-23T20:54:23Z</published>
<title>Culturomics meets random fractal theory: Insights into long-range correlations of social
and natural phenomena over the past two centuries</title>
<summary>Culturomics was recently introduced as the application of high-throughput data
collection and analysis to the study of human culture. Here we make use of this data by
investigating fluctuations in yearly usage frequencies of specific words that describe social
and natural phenomena, as derived from books that were published over the course of the past two
centuries. We show that the  determination of the Hurst parameter by means of fractal analysis
provides fundamental insights into the nature of long-range correlations contained in the
culturomic trajectories, and by doing so, offers new interpretations as to what might be the main
driving forces behind the examined phenomena. ... </summary> ...
<arxiv:journal_ref xmlns:arxiv="http://arxiv.org/schemas/atom">
    J. R. Soc. Interface 9 (2012) 1956-1964</arxiv:journal_ref>
<link href="http://arxiv.org/abs/1202.5299v1" rel="alternate" type="text/html"/>
... </entry></feed>
```

Adding automated streams of data to the original Project Gutenberg dataset from other sources carries a requirement that the IT infrastructure must address well in advance of expanding the dataset: Putting in place a scale-out storage system that can handle the throughput of multiple streams of incoming data and the volume of stored data that those streams will generate.

Managing the ingestion of such data streams might also call for a centralized service for managing data streams. Apache Kafka partitions data streams and spreads them over a distributed cluster of machines to coordinate the ingestion of vast amounts of data for analysis.

ANALYTICS FRAMEWORKS

This section examines analytics frameworks and tools for analyzing large datasets. The technical characteristics of the frameworks and tools influence the kind of IT infrastructure that supports data science.

UNKNOWN PROBLEMS AND VARYING DATA SETS DEMAND FLEXIBLE APPROACHES

Implementing a data science program—especially in a context of multiple tenants analyzing various datasets to solve different problems—produces two overarching, interrelated requirements:

[14] The example is an abridged version of the metadata.

1. The flexibility to use the analytics tool that works best with the dataset on hand.
2. The flexibility to use the analytics tool that best serves your analytical objectives.

Several variables cement these requirements:

1. When you begin to collect data to solve a problem, you might not know the characteristics of the dataset, and those characteristics might influence the analytics framework that you select.
2. When you have a dataset but have not yet identified a problem to solve or an objective to fulfill, you might not know which analytics tool or method will best serve your purpose.

In the traditional domain of the data warehouse and business intelligence system, these requirements are well known, as the following passage from *The Data Warehouse Toolkit* demonstrates:

> *"The DW/BI system must adapt to change. User needs, business conditions, data, and technology are all subject to change. The DW/BI system must be designed to handle this inevitable change gracefully so that it doesn't invalidate existing data or applications. Existing data and applications should not be changed or disrupted when the business community asks new questions or new data is added to the warehouse."*[15]

The fact of the matter is that unknown business problems and varying datasets demand a flexible approach to choosing the analytics framework that will work best for a given project or situation. The sheer variety of data requires a variety of tools—and different tools are likely to be used during the different phases of the data science pipeline. Common tools include Python, the statistical computing language R, and visualization software Tableau. But the framework that many businesses are rapidly adopting is Apache Hadoop.

HADOOP: MAPREDUCE AND HDFS

A key big data analytics platform, Apache Hadoop comprises the Hadoop Distributed File System, or HDFS, a storage system for vast amounts of data, and MapReduce, a processing paradigm for data-intensive computational analysis. YARN separates resource management from computational processing to create the next-generation MapReduce framework.

You can create MapReduce applications that analyze files stored in a Hadoop Distributed File System. Because the Isilon cluster storing the Project Gutenberg data supports HDFS, a MapReduce application can access the data directly on the cluster and analyze its content in place—that is, without moving it. Here is an example of a simple MapReduce application, run from a compute cluster of Linux machines, that counts the words in a file on an Isilon cluster and outputs the result:[16]

```
$ bin/hadoop jar wc.jar WordCount /ifs/isiloncluster01/zone1/analytics/inputfile
/ifs/isiloncluster01/zone1/analytics/outputfile
```

COMPUTING PROBLEMS IN ANALYZING BIG DATA

In addition to unknown business problems and varying datasets, other problems give rise to new frameworks for big data analytics.

> *"Modern data analysis faces a confluence of growing challenges. First, data volumes are expanding dramatically, creating the need to scale out across clusters of hundreds of commodity machines. Second, this new scale increases the incidence of faults and stragglers (slow tasks), complicating parallel database design. Third, the complexity of data analysis has also grown: modern data analysis employs sophisticated statistical methods, such as machine learning algorithms, that go well beyond the roll-up and drill-down capabilities of traditional enterprise data warehouse systems. Finally, despite these increases in scale and complexity, users*

[15] *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*, 3rd Edition, by Ralph Kimball; Margy Ross, John Wiley & Sons, 2013.

[16] This example is adapted from the documentation on the Apache Hadoop web site at http://hadoop.apache.org/docs/current/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html#Example:_WordCount_v1.0.

> *still expect to be able to query data at interactive speeds,"*[17] *several researchers write in a paper titled "Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing."*

This passage highlights four key requirements for the information technology that forms a data platform:

1. A scale-out storage system.
2. The use of robust high-performing analytics tools capable of processing huge volumes of data.
3. The ability to embed statistical analysis, machine learning, and other advanced methods in the analytics tools.
4. The ability to query large datasets in near real time when immediacy is a concern.

Assuming a scale-out storage system is in place, the last three of these requirements helped form the impetus for Apache Spark.

DISTRIBUTED DATA ANALYSIS WITH SPARK

Apache Spark is a fast engine for large-scale data processing that runs, according to the Apache Spark web site, as much as 100 times faster than MapReduce in memory and 10 times faster on disk. It differs from MapReduce in another significant ways, however:

- It can be used interactively from the Python or Scala shell.
- It combines streaming, SQL, and complex analytics by powering a stack of tools that can be combined in the same application.
- It runs on Hadoop YARN or standalone, and it can access diverse data sources, including not only HDFS but also Cassandra, MongoDB, and HBase.

The following Python code, which is adapted from the Apache Spark web site at https://spark.apache.org/, demonstrates how to connect to a file stored on an Isilon cluster by using the Isilon HDFS interface. In the following example, the name of the Isilon SmartConnect zone associated with the access zone named az1 is isiloncluster1.lab.example.com.

```
file = spark.textFile("hdfs://isiloncluster1.lab.example.com/cluster001/az1/hadoop/logfile")
counts = file.flatMap(lambda line: line.split(" ")) \
.map(lambda word: (word, 1)) \
.reduceByKey(lambda a, b: a + b)
counts.saveAsTextFile("hdfs://isiloncluster1.lab.example.com/cluster001/az1/hadoop/result")
```

The Project Gutenberg dataset can be expanded with a streaming workflow that exploits Spark to filter salient content from the stream. The relevant content is then added to the dataset on the storage cluster.

Even though a streaming analytics workflow adds a layer of complexity to a data processing platform, adding information in near real-time from Twitter can add a layer of contemporary voices to the dataset. A streaming workflow can originate as a real-time Twitter feed, for example, and be distributed to a computational task in memory on a Spark cluster. The streaming task filters and counts the number of references to *fiction* and *culturomics*. The information can be persisted to the storage system for additional analysis later. A data analyst can later run a Hadoop MapReduce job to produce a report on such tweets over the past 6 months.

The following examples tap the set of streaming APIs offered by Twitter to obtain low-latency access to Twitter's global stream of Tweet data:[18]

```
https://stream.twitter.com/1.1/statuses/filter.json?track=fiction
https://stream.twitter.com/1.1/statuses/filter.json?track=culturomics
```

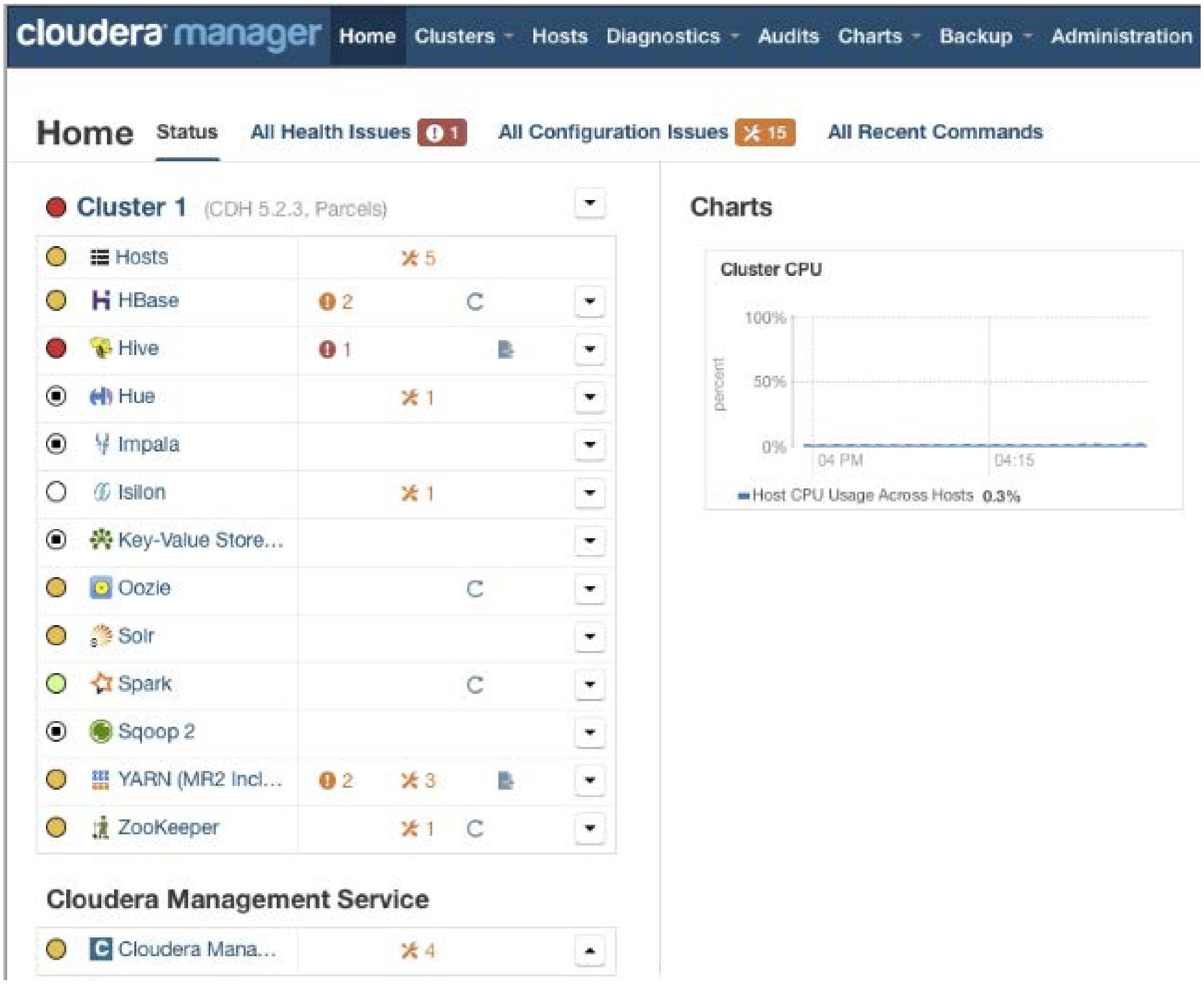
[17] Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing; Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy McCauley, Michael J. Franklin, Scott Shenker, Ion Stoica, University of California, Berkeley. https://www.cs.berkeley.edu/~matei/papers/2012/nsdi_spark.pdf

[18] See https://dev.twitter.com/streaming/public and https://dev.twitter.com/streaming/reference/post/statuses/filter.

OTHER HADOOP-RELATED TOOLS

Apache Hadoop is supplemented by projects that expand the options for distributed data analytics; the projects include Apache Hive, Apache Pig, Impala, and Apache Solr.

By using Cloudera, the dataset stored on the Isilon cluster can be accessed and queried with these tools, and you can orchestrate the workflows by using Apache Oozie. You can also access stored data and process it with other Hadoop distributions, such as Pivotal and Hortonworks.



Importantly, the storage system should not limit the Hadoop distributions with which you analyze data. The flexibility to run any Hadoop distribution and its accompanying tools is paramount. Putting in place IT infrastructure that restricts your choice of tools can limit your ability to respond to new analytics use cases later.

THE TOOL SUITS THE PROBLEM AND THE DATA (OR THE USER)

Analytics tools such as Apache Hadoop and its related projects underscore the data science pipeline. At each stage of the workflow, data scientists or data engineers are working to clean their data, extract aspects of it, aggregate it, explore it, model it, sample it, test it, and analyze it. With such work comes many use cases, and each use case demands the tool that best fits the task. During the stages of the pipeline, different tools may be put to use. The tools include Hadoop, git, Avro, Impala, Apache NiFi, Flume, Kafka, Solr, Tableau, Pivotal HAWQ, and MongoDB.

According to O'Reilly Media's 2014 Data Science Salary Survey, the most common tools remain SQL, Excel, R, and Python. But the tools that correspond to higher median salaries are the newer, highly scalable ones: Apache Hadoop, Hive, Pig, Cassandra, Cloudera, Spark, and HBase.[19]

In general, the data scientists run their tools on Linux, Mac OS X, and Microsoft Windows, and the survey found a sign that some analysts try different tools with the same function. An interesting observation came out of the survey's interpretation of its clustering analysis of tools: "individuals and companies have not necessarily settled on their choice of tools and are trying different combinations among the many available options."

The plethora of tools run on different operating systems necessitates that the IT infrastructure that supports the data scientists is flexible. The infrastructure must support data access over multiple protocols so that many tools running on different operating systems, whether on a compute cluster or a user's workstation, can access the stored data. A key concept that is motivated by the need to support different datasets, teams, tools, and access methods is the data lake.

IMPLEMENTING A DATA LAKE FOR FRAMEWORK FLEXIBILITY

A data lake is a storage strategy to collect data in its native format in a shared infrastructure from disparate workflows and make it available to different analytics applications, teams, and devices over common protocols. As a central, highly scalable repository that stores disparate sources of information without requiring schema-on-write or ETL, a data lake is an operational foundation that drives the shift toward a data-driven enterprise. A data lake consolidates all the data into a single location for cross-correlation and analysis with any tool.

To handle the volume and velocity of big data, a data lake must be an order of magnitude more scalable than existing approaches for data warehousing and business intelligence.

To deliver seamless access to big data without needing to migrate or export data, the data lake must support multiprotocol data access, including HDFS, as business units and applications deposit information through SMB, NFS, rsync, FTP, HTTP, REST, and Swift. The data lake must support existing applications and workflows as well as emerging workloads and relatively new applications, like Spark. The data lake must also provide the means to protect, back up, and secure the data.

The flexibility of a data lake empowers the IT infrastructure to serve the rapidly changing needs of the business, the data scientists, the datasets, and the big data tools. Importantly, the flexibility of a data lake fosters data analysis with any analytics application, from new frameworks like Spark to established languages like Python and R. Decisions about requirements need no longer focus on the analytics application but on secure, universal access.

When business units pool their data for storage and analysis, however, the data lake must be able to securely segregate the business units and their datasets. Implicit in the concept of an enterprise data lake, then, is a requirement for secure *multitenancy*—the storage platform can securely segregate business units and datasets while working with different analytics applications. To support multitenancy, the storage platform should fulfill the following requirements:

- Support many tenants—business units or distinct groups of data scientists—and their datasets on a shared infrastructure.
- Store a variety of datasets.
- Isolate tenants and their datasets with robust security.
- Work with different analytics applications and different versions of same applications simultaneously.
- Run many analytics jobs concurrently.
- Work with multiple data access protocols simultaneously to support data scientists who use different applications or different connection methods, such as NFS, SMB, and HDFS.

[19] *2014 Data Science Salary Survey: Tools, Trends, What Pays (and What Doesn't) for Data Professionals*, Publisher: O'Reilly; released November 2014 at http://www.oreilly.com/data/free/2014-data-science-salary-survey.csp.

These requirements combine with the problems outlined in the previous sections to form a matrix of issues that a storage solution for big data analytics must address.

If the storage solution is flexible enough to support many big data activities, it can yield a sizable return on the investment. As Foster Provost and Tom Fawcett write in *Data Science for Business*:

> *"We can think of ourselves as being in the era of Big Data 1.0. Firms are busying themselves with building the capabilities to process large data, largely in support of their current operations—for example, to improve efficiency. ... We should expect a Big Data 2.0 phase to follow Big Data 1.0. Once firms have become capable of processing massive data in a flexible fashion, they should begin asking: 'What can I now do that I couldn't do before, or do better than I could do before?' This is likely to be the golden era of data science."*[20]

The notion of a multitenant data lake sets the stage for a discussion of the IT infrastructure that best supports the data science pipeline and the newer, highly scalable big data tools while addressing the technical problems that underlie the pipeline.

STORAGE INFRASTRUCTURE FOR A DATA PLATFORM

Most organizations have yet to implement a single, scalable storage system that exposes all their data to analytics applications and interoperates with their compute clusters. The data is often stuck in silos, inefficient scale-up storage systems that are costly to manage and difficult to expand. The inability to scale such traditional storage systems hinders the progression to a platform that makes a data-driven enterprise possible.

For instance, according to McAfee and Brynjolfsson:

> *"Sears required about eight weeks to generate personalized promotions, at which point many of them were no longer optimal for the company. It took so long mainly because the data required for these large-scale analyses were both voluminous and highly fragmented—housed in many databases and 'data warehouses' maintained by the various brands."*[21]

Although being able to handle the velocity, volume, and variety of data are basic requirements for a storage system, there are other problems that must be solved and additional requirements that must be fulfilled. The next section summarizes these requirements. Subsequent sections address such problems as privacy, multitenancy, fault tolerance, availability, and scalability.

STORAGE REQUIREMENTS FOR BIG DATA ANALYTICS

The data science pipeline, its problems, and the variety of analytics tools give rise to the following IT infrastructure requirements:

- Sheds schema-on-read requirements to support the original format of a wide range of media types and data formats, including not only semi-structured and unstructured data but also audio files, videos, and images.
- Performs high-performance data ingestion from many sources, devices, and applications.
- Scales out simply and easily to store petabytes of data.
- Supports a range of use cases, mixed workflows, and next-generation workloads.
- Lets data engineers combine data from different sources into a dataset.
- Delivers input-output performance to handle large-scale analytics workloads.
- Gives analytics applications and data scientists access to data over standard protocols, such as NFS and SMB, from different operating systems, such as Mac OS X and Microsoft Windows.
- Gives the highly scalable computational frameworks, such as Hadoop and Spark, access to data over the HDFS protocol so that data scientists can analyze data without migrating it to a compute cluster.

[20] *Data Science for Business*, Foster Provost and Tom Fawcett (O'Reilly), 2013, 978-1-449-36132-7.
[21] Big Data: The Management Revolution, by Andrew McAfee and Erik Brynjolfsson, Harvard Business Review, October 2012. https://hbr.org/2012/10/big-data-the-management-revolution/ar.

- Works with many analytics tools and standard programming languages that can embed machine learning and statistical analysis, such as Python and R, so that data scientists can analyze unstructured data with the tools they prefer.
- Works with Apache Hive, Cloudera Implala, and Pivotal HAWQ to let data scientists structure and query large datasets in a SQL-like manner with massively parallel processing even though the data resides in a distributed storage system.
- Interoperates with an orchestration framework, such as Apache Oozie, to manage complex workflows and concurrent jobs.
- Provides fault tolerance for large-scale parallel, distributed processing with no single point of failure to ensure high availability for data access.
- Secures data with robust security capabilities, including authentication and granular access control, to help meet compliance regulations.
- Stores data with an efficient data protection scheme to minimize storage costs and maximize capacity utilization.
- Manages data with enterprise storage features like deduplication, snapshots, and backups.
- Balances cost, capacity, and performance.
- Minimizes operation expenditures (OPEX) for data management and scalability.
- Lets system administrators scale capacity and performance independently to meet fluid requirements.

Setting aside the question of a distributed compute cluster for a moment, these are the requirements against which storage infrastructure for big data must be measured.

MULTIPROTOCOL DATA ACCESS

An early requirement in the list—multiprotocol data access—rules out most storage systems. Many of the highly scalable analytics tools access data with a relatively new protocol: HDFS. Most scale-up storage architectures, however, do not support native HDFS access. Those that do, usually through a specialized HDFS connector, often lack the scalability to support big data or introduce a layer of complexity.[22] A storage system without HDFS access effectively bars the use of highly scalable analytics tools like Hadoop and Spark, which either limits the choice of analytics tools or forces data engineers to move the data to another system.

The lack of support for HDFS among infrastructure options leaves two proven architectures for a data lake:

1. Hadoop, which combines MapReduce and HDFS into a single cluster for compute and storage.
2. An EMC Isilon storage cluster, which separates storage from compute.

SHARED STORAGE VS. HADOOP DAS

Some enterprises aim to fulfill their big data storage requirements with a single monolithic system—a large Hadoop cluster—that handles storage and compute. But as a multi-purpose enterprise data store, Hadoop carries several major flaws: inefficiency, immature security, and inaccessibility.

Hadoop's data protection scheme, which replicates data three times, is inefficient. Compared with an efficient protection scheme, such as erasure coding, triple replication results in a costly decrease in capacity that can undermine the savings from using commodity hardware. HDFS also lacks the enterprise capabilities—such as snapshots, tiering, and disaster recovery—that are necessary to store data efficiently and manage it effectively.

Because many storage architects recognize the inherent limitations of HDFS as a data lake, they implement other storage systems to hold enterprise data, but then analyzing the data with Hadoop and other tools that use HDFS entails a costly and inefficient workflow—known as extract, transform, load (ETL)— to move the data into HDFS for

[22] Although there are specialized HDFS connectors for some systems, connectors are suboptimal, nonstandard workarounds for accessing data.

analysis and then export the results. All these inefficiencies take their toll, a Hadoop data lake increasingly becomes a highly complex undertaking that increases management costs and operating expenses risks.

The inefficiencies of HDFS are compounded by Hadoop's lack of mature security capabilities to safeguard data and to isolate users. To protect an HDFS system and the data it contains, system administrators must build and manage many layers of security software.

Although HDFS works with more analytics applications beyond MapReduce, including HAWQ, Impala, Spark, and Storm—it impedes seamless access by applications outside the Apache Hadoop project, such as end users or analytics applications that use NFS or SMB. Serving the data to other applications requires an ETL workflow that complicates the data science pipeline and stifles rapid insights because it takes too long time to load the data. The same problem applies the other way, too: Application data that is stored by an NFS or SMB workflow in a storage silo must be extracted and loaded into HDFS for analysis, which increases the time to actionable results.

SEPARATING STORAGE FROM COMPUTE

In contrast to a Hadoop cluster, an Isilon storage cluster separates data from compute. As Hadoop clients run MapReduce jobs, the clients access the data stored on an Isilon cluster over HDFS. The Isilon cluster becomes the native HDFS storage system for MapReduce clients.

EMC Isilon scale-out NAS fosters the convergence of data analytics with stored data. As you work to extract value from stored data, you can use an Isilon cluster's HDFS implementation to point your data analytics tools at the storage system. Instead of requiring application developers to move data to the compute grid, you can take the compute function to where the data already resides—in storage.

The convergence of stored data and data analysis helps streamline the entire analytics workflow. The convergence eliminates the need to extract the data from a storage system and load it into a traditional Hadoop deployment with an ETL workflow. The convergence also eliminates the need to export the data after it is analyzed. Streamlining the analytics workflow cost-effectively speeds the transition to a data-focused enterprise: You not only increase the ease and flexibility with which you can analyze data but also reduce your capital expenditures and operating expenses.

Separating storage from compute brings significant benefits, as Gartner's January 2015 report on critical capabilities for scale-out storage systems makes clear:

> *"I&O leaders are embracing scale-out file system storage for its added benefits. First and foremost, the technology includes embedded functionality for storage management, resiliency and security at the software level, easing the tasks related to those functions in the I&O organization. The technology also offers nearly linear horizontal scaling and delivers highly aggregated performance through parallelism. This means that scale-out file system storage enables pay-as-you-grow storage capacity and performance ... "*[23]

The Gartner report cites big data analytics as an emerging use case for scale-out storage. With its native support for the HDFS protocol, an EMC Isilon cluster uniquely serves as a data lake to support such next-generation workloads with the following advantages:

- Scale storage independently of compute as your datasets grow
- Store data in a POSIX-compliant file system with existing SMB, NFS, and HDFS workflows
- Protect data reliably and efficiently with erasure coding
- Preserve system availability with fault-tolerance
- Improve security and compliance for stored data
- Support multitenancy
- Manage data with enterprise storage features like snapshots and tiering

[23] Critical Capabilities for Scale-Out File System Storage, Gartner, 27 January 2015, www.gartner.com/technology/reprints.do?id=1-28XVMOC&ct=150130&st=sb. In the quotation, I&O stands for *infrastructure and operations*.

For more information on how separating storage from compute improves efficiency, scalability, and workflow flexibility, see EMC Isilon Scale-Out NAS for In-Place Hadoop Data Analytics.

FAULT-TOLERANT PERFORMANCE FOR HDFS WORKLOADS

A storage system for MapReduce must gracefully tolerate node and disk failures. To process data in parallel for MapReduce jobs with fault tolerance, Isilon's clustered architecture supports the following availability objectives with no single point of failure:

- Redundant namenodes—every node in an Isilon cluster serves as a namenode and a datanode simultaneously
- Tolerance for multi-failure scenarios
- Fully distributed single file system
- Pro-active failure detection and preemptive drive rebuilds
- Fast, scalable drive rebuilds
- Flexible, efficient data protection with erasure codes
- Fully journaled file system

An EMC Isilon cluster can continue to perform when components fail. Because an Isilon cluster is a pure scale-out architecture coupled with a distributed operating system, called OneFS, that gracefully handles component failures, the file system can continue to support input-output operations for analytics jobs while drives or nodes are being repaired or replaced.

Isilon does not contain HA pairs of controllers and thus does not experience a performance bottleneck on node failure. Because every node in effect acts as a controller, the failure of a node results in the distribution of the node's workload to other nodes. If a node goes offline, the cluster's overall performance degrades only by a small percentage. If one node in a 10-node cluster goes offline, for example, the cluster's performance diminishes only by about 10 percent. Similarly, drive rebuild times are inherently low because the OneFS operating system distributes the work of rebuilding the data by using all the disks and CPUs in a set of nodes. For more information, see EMC Isilon Scale-out NAS: An Architecture For Resiliency, High Availability, And Data Protection.

PRIVACY, SECURITY, AND COMPLIANCE

A significant problem that underlies the data science pipeline is privacy—safeguarding the stored data with robust security settings that help meet internal governance policies and external compliance regulations.

In its native state, the HDFS file system fulfills few compliance requirements for information security. Without separating Hadoop compute clients from the Hadoop Distributed File System, Hadoop is difficult to secure because it is a complex, distributed system with a dual purpose—data storage and data-intensive computational analysis. Because Hadoop is often implemented as a multi-tenant service without client-server interactions, there is no single point of access where the system can be secured. The distributed nature of the system, coupled with many jobs running on many clients, makes Hadoop's native security capabilities difficult to implement and time-consuming to manage.

With a Hadoop cluster, ensuring compliance with a regulation or corporate data policies can require as many as 20 additional layers of security software, all of which must interoperate seamlessly. Even with a layered-approach to securing a native Hadoop system, however, compliance problems linger. Connecting Hadoop to Active Directory with Apache Knox, for instance, controls access only to the system, not to directories or files. Sensitive data can be accessed by personnel without a business need to know.

In contrast, a separate storage cluster can help secure the data in a data lake with the following capabilities:

- Role-based access control for system administration
- Identity management with an external directory service like Active Directory or LDAP
- Authentication with the Kerberos security protocol

- Fine-grained access control to the file system
- Permissions and ACL policies that persist across the data access protocols and operating systems
- User and ID mapping to associate one user with one ID to help meet compliance requirements
- Write-once, read-many storage (WORM)
- Encryption of data at rest
- Auditing of file system events and administrative changes

For more information, see Compliance and Security for Hadoop Scale-Out Data Lakes.

MULTITENANCY

If you make the wrong decisions in selecting a storage infrastructure for big data, the system's security may not keep pace with the requirements associated with the data science pipeline. A primary example of such a mismatch is implementing a storage system that does not support multitenancy only to find that multitenancy is needed when new teams of data scientists are added to the organization or new datasets are added to the data lake. Multitenancy means that the storage platform can, among other requirements, securely segregate business units and datasets while working with different data access protocols and analytics applications.

With multitenancy, a data lake securely extends the opportunity to perform big data analytics to each group in an enterprise. All teams can be represented equally, each with its own virtual space and security context to store and process data for analytics. Multitenancy, in effect, democratizes big data analytics.

An Isilon storage cluster implements multitenancy through access zones. An access zone segregates a portion of the cluster and presents it as a secure virtual storage region. Each access zone can connect to directory services like Active Directory, control the access of tenants, consolidate SMB file shares and NFS exports, and contain an autonomous HDFS root directory for each tenant. Isilon access zones work with most of the Hadoop projects and applications as well as YARN.

An Isilon data lake fulfills the following multitenancy requirements:

- Isolates tenants and their datasets with access zones.
- Provides multitenancy for identity management, authentication, and access control.
- Handles mixed workflows in each access zone, including Hadoop, R, HAWQ, PIG, HIVE, and other applications.
- Simultaneously runs multiple Hadoop distributions—including Cloudera, Pivotal HD, Apache Hadoop, and Hortonworks—against the same dataset at the same time. The OneFS distributed operating system supports HDFS 1.0 and 2.0 simultaneously without migrating data or modifying metadata.
- Runs many Hadoop jobs concurrently and lets you allocate resources with YARN.
- Secures each access zone with Kerberos authentication and other security mechanisms to protect the confidentiality of data.
- Manages tenants and their data with quotas, storage pools, and other enterprise features.

The following sample listing shows how an access zone encapsulates directory services, authentication, auditing, an HDFS root directory, and other features in a security context that contains a virtual storage area:

```
test-cluster-1# isi zone zones list -v
                     Name: z1 ❶
                     Path: /ifs/z1
               Cache Size: 9.54M
            Map Untrusted:
           Auth Providers: lsa-ldap-provider:krbldap,
                           lsa-krb5-provider:SOUTH.EXAMPLE.COM ❷
             NetBIOS Name:
```

```
       All Auth Providers: Yes
       User Mapping Rules: -
     Home Directory Umask: 0077
       Skeleton Directory: /usr/share/skel
            Audit Success: create, delete, rename, set_security, close ❸
            Audit Failure: create, delete, rename, set_security, close
      HDFS Authentication: all
      HDFS Root Directory: /ifs/z1 ❹
          WebHDFS Enabled: Yes
       HDFS Ambari Server:
     HDFS Ambari Namenode:
Syslog Forwarding Enabled: No
      Syslog Audit Events: create, delete, rename, set_security
                  Zone ID: 2
```

❶ The name of this access zone.

❷ The directory services with which OneFS authenticates users and groups in the access zone.

❸ The audit settings that are turned on for the access zone.

❹ The unique HDFS root directory for the access zone. Each access zone can have a different HDFS root directory that serves only the zone's tenants.

An access zone can simultaneously contain an HDFS namespace and namespaces for Swift, NFS, and SMB files. Within the access zone, you can authenticate users by simultaneously using Active Directory, NIS, and LDAP.

By placing authentication, access control, directory structures, and enterprise management features in the context of an access zone, OneFS lets you target the following high-level use cases with relative ease:

- Isolate directories and files in virtual storage regions without the overhead of managing individual volumes
- Consolidate Multiple HDFS workflows on a single cluster
- Migrate a scale-up storage system to an Isilon cluster to ease management, reduce operational expenses, and improve storage efficiency
- Set upper limits on resource usage by managing quotas, collections of disks, and network interfaces
- Provision and manage a different Hadoop root directory for different tenants

Perhaps most importantly for Hadoop, each access zone can contain a unique HDFS root directory for the tenant in that zone. For HDFS, the ability to enforce unique HDFS root directories in each access zone allows you to point multiple Hadoop compute clusters at the same HDFS storage system. For more information, see EMC Isilon Multitenancy for Hadoop Big Data Analytics.

The flexibility of multitenancy—especially when it is coupled with the additional flexibility of virtualization—empowers the IT infrastructure team to deliver Hadoop as a service (HDaaS) to internal teams of data scientists.

VIRTUALIZING HADOOP WITH LARGE-SCALE INFRASTRUCTURE TO DELIVER HDAAS

Traditional Hadoop clusters have proved inefficient for handling large-scale analytics jobs sized at hundreds of terabytes or even petabytes.

Adobe's Digital Marketing organization, which operates data analytic jobs on a scale of hundreds of terabytes, was encountering increased demand internally to use Hadoop to analyze the company's eight-petabyte data repository.

Adobe has a goal of running analytics jobs against datasets that are hundreds of terabytes in size. Simply adding commodity servers to Hadoop clusters would become highly inefficient, especially since traditional Hadoop clusters require three copies of the data to ensure availability. Adobe also was concerned that current Hadoop versions lack high availability features. For example, Hadoop has only has two namenodes, which tracks where data resides in Hadoop environments. If both namenodes fail, the entire Hadoop cluster would collapse.

To address the problems and requirements, Adobe explored an innovative approach to Hadoop. Rather than running traditional Hadoop clusters on commodity servers with locally attached storage, Adobe virtualized the Hadoop computing environment and used its existing EMC Isilon storage—where the eight-petabyte data repository resides—as a central location for Hadoop data. Rather than moving data from a large data repository to the Hadoop clusters—a time-consuming task—Adobe Technical Operations determined it would be most efficient to simply use Hadoop to access datasets on the existing Isilon-based data repository.

Technical Operations proposed separating the Hadoop elements and placing them where they can scale more efficiently and reliably. This meant using Isilon, where Adobe's file-based data repository is stored, for centralized Hadoop storage and virtualizing the Hadoop compute nodes to obtain more flexible scalability and lower compute costs.

Adobe enlisted resources, technologies, and expertise of EMC, VMware, and Cisco to build a reference architecture for virtualized Hadoop-as-a-Service (HDaaS) and perform a comprehensive proof of concept. After configuring, refining, and testing, Adobe successfully ran a 65-terabyte Hadoop job—one of the industry's largest known Hadoop workloads to date in a virtualized environment. In addition, it was the largest workload ever tested by EMC in a virtual Hadoop environment on an EMC Isilon storage cluster.

The POC refutes claims by some in the industry that suggest shared storage will cause problems with Hadoop. To the contrary, Isilon had no adverse effects and even contributed superior results in a virtualized HDaaS environment compared to traditional Hadoop clusters. These advantages apply to many aspects of Hadoop, including performance, storage efficiency, data protection, and flexibility.

For more information—including architectural details, configuration tuning, and performance results—see Virtualizing Hadoop In Large-Scale Infrastructures.

Separating the infrastructure into a virtualized Hadoop compute cluster and a scale-out centralized storage system optimally balanced performance, capacity, and cost. Adobe plans to bring virtual HDaaS to production for its business users and data scientists.

For more information on how separating the Hadoop compute function from the enterprise storage function improves scalability, efficiency, and workflow flexibility, see EMC Isilon Scale-Out NAS for In-Place Hadoop Data Analytics.

CONCLUSION

The three domains of an enterprise big data program—data science, analytics frameworks, and IT infrastructure—should complement one another without imposing functional limitations. A multiprotocol, multitenant data lake supports the data science pipeline while solving the pipeline's problems. When an enterprise stores all its data, regardless of type, in a data lake, data scientists can analyze the data with the tools of their choice, including such big data frameworks as Hadoop, Hive, and Spark. Meanwhile, as the scale of an enterprise's data expands, infrastructure managers can scale out the data lake to match increases in the volume of data. With multiprotocol data access to support all users, applications, and devices, a data lake gives an enterprise the flexibility to adapt to emerging workloads and new use cases. An EMC Isilon storage cluster, with its HDFS interface and secure multitenancy, uniquely delivers the scalability and flexibility that a data lake for data science demands.