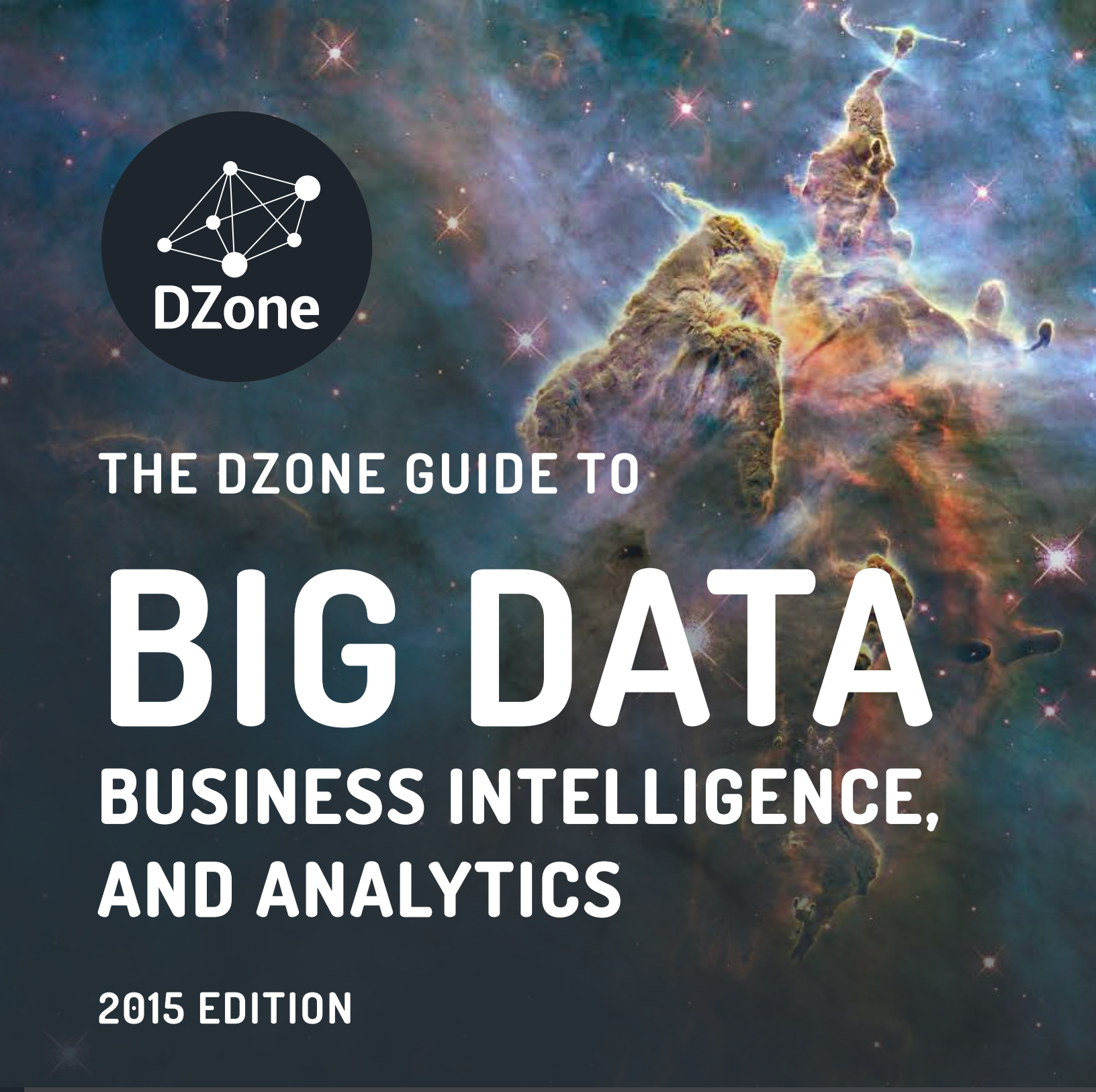


DZone

THE DZONE GUIDE TO

BIG DATA

BUSINESS INTELLIGENCE, AND ANALYTICS

2015 EDITION

BROUGHT TO YOU IN PARTNERSHIP WITH

Hortonworks

informatica

MAMMOTH DATA

Objectivity

ZOOMDATA

Dear Reader,

Analysts make hype; developers make software. Our perspective on Big Data is pragmatic. If I have to worry about the size of the data, then the data is big. OK, definition operationalized, concept clarified—hot-air throat-clearing is over. Let's spin up a Hadoop cluster.

Well, not so fast. The 'how big is Big Data' question is cliché, but it can help developers, data scientists, and business users work in the same bounded context. Yes, say the devs, application code cares about those 'three V' dimensions of Big Data—volume, velocity, and variety. Of course, say data scientists, that picture of Big Data doesn't care about the meaning of the data, so we need another two V's—maybe veracity and value. The first three describe the data itself. The second two are about how that data relates to reality.

So 'Big Data' for developers is in a bit of a funny headspace. On the one hand, from a purely engineering perspective, data is only big when it causes headaches—otherwise the job is plain old ETL, OLAP, CEP, whatever. On the other, the bombastic term 'Big Data' thrills engineers too, because it captures what makes code exciting in the first place: doing massively valuable macro-scale things simply by manipulating symbols according to formal rules.

The tools to capture and transform the world as data are young but multiplying quickly. We're finally getting beyond plain old MapReduce as tools like Storm, Spark, Tez (and Hadoop 2.0 itself) gain traction and mature. Data stores are getting more sophisticated by venturing beyond the relational paradigm and improving storage, retrieval, and caching algorithms. The result, according to our research: more than 80% of developers are already beginning work on projects for large-scale data gathering and analysis. 'Big Data' is here; the landscape is growing increasingly complex.

So in this *2015 DZone Guide to Big Data* we're focusing on post-Hadoop tools and techniques, with special emphasis on streaming data processing and visualization. We want to help reduce devs' Big Data pain while mapping analysts' Big Data enthusiasm to reality.

Enjoy the Guide and let us know what you think.

JOHN ESPOSITO
EDITOR-IN-CHIEF, DZONE RESEARCH
RESEARCH@DZONE.COM

TABLE OF CONTENTS

CREDITS

EDITORIAL

John Esposito
research@dzone.com
EDITOR-IN-CHIEF

G. Ryan Spain
DIRECTOR OF PUBLICATIONS

Mitch Pronschinske
SR. RESEARCH ANALYST

Benjamin Ball
RESEARCH ANALYST

Matt Werner
MARKET RESEARCHER

Moe Long
MARKET RESEARCHER

John Walter
EDITOR

Lauren Clapper
EDITOR

Allen Coin
EDITOR

BUSINESS

Rick Ross
CEO

Matt Schmidt
PRESIDENT & CTO

Kellet Atkinson
VP & PUBLISHER

Matt O'Brian
DIRECTOR OF BUSINESS DEVELOPMENT

Jane Foreman
VP OF MARKETING

Alex Crafts
sales@dzone.com
DIRECTOR OF MAJOR ACCOUNTS

Chelsea Bosworth
MARKETING ASSOCIATE

Chris Smith
PRODUCTION ADVISOR

Jillian Poore
SALES ASSOCIATE

Jim Howard
SALES ASSOCIATE

Chris Brumfield
CUSTOMER SUCCESS ASSOCIATE

ART

Ashley Slate
DESIGN DIRECTOR

Yassee Mohebbi
GRAPHIC DESIGNER

Special thanks to our topic experts Alexander Podelko, Alex Rosemblat, David Farley, Sergey Chernyshev, Colin Scott, and our trusted DZone Most Valuable Bloggers for all their help and feedback in making this report a great success.

WANT YOUR SOLUTION TO BE FEATURED IN COMING GUIDES?
Please contact research@dzone.com for submission information.

LIKE TO CONTRIBUTE CONTENT TO COMING GUIDES?
Please contact research@dzone.com for consideration.

INTERESTED IN BECOMING A DZONE RESEARCH PARTNER?
Please contact sales@dzone.com for information.

Executive Summary

DZone surveyed over 400 IT professionals to better understand how individuals and organizations are currently managing the collection, storage, and analysis of Big Data. The results provide a snapshot of the landscape of Big Data in the IT industry and, set against results from last year's Big Data survey, reveal recent trends and trajectories of Big Data technologies and strategies.

RESEARCH TAKEAWAYS

01. BIG DATA IS INCREASINGLY ABOUT MORE THAN JUST HADOOP

Data: While Apache Hadoop remains popular (31% of respondents reported that their company uses Hadoop), its usage has dropped 5% from last year's survey. Of the tools being used with Hadoop, the two leading technologies—Hive and Pig—show usage decreases of 8% and 11% respectively from last year's results. Other tools being used with Hadoop, particularly Spark and Flume, have increased in popularity (with respective increases of 15% and 8%). Overall, the standard deviation of usages among these tools (Hive, Pig, Spark, Flume, Sqoop, Drill, Tez, and Impala) has decreased incrementally from last year (4%).

Implications: Organizational data needs are changing, and tools once considered useful for all Big Data applications no longer suffice in every use case. When batch operations were predominant, Hadoop could handle most organizations' needs. Advances in other areas of the IT world (think IoT) have changed the ways in which data needs to be collected, distributed, stored, and analyzed. Real-time data complicates these tasks and requires new tools to handle these complications efficiently. While Big Data seemed, for a while, to emphasize the *volume* of data involved, velocity and variety are becoming increasingly vital to many Big Data applications.

Recommendations: Don't rely on hype to make decisions regarding how you work with Big Data. As discussed in this guide's Key Findings, many organizations use Hadoop even when the volume of data they work with doesn't require Hadoop's distributed storage or processing. There are a lot of tools out there you can use for your particular Big Data needs; to learn more about some of these tools, take a look at the article "Workload and Resource Management: YARN, Mesos, and Myriad" by Adam Diaz, in this guide. Or take a look at our Solutions Directory for a more comprehensive listing of platforms and tools.

02. DATA STORAGE VOLUMES NOT REACHING EXPECTED LEVELS

Data: Last year we asked Big Data survey respondents to estimate the volume of data their organizations stored and used (<1 TB, 1-9 TB, 10-49 TB, etc.), as well as to estimate storage and usage volumes for the coming year. We asked the same of respondents this year. Results from last year to this year, in almost every case, were within about 1% of each other. For example: last year, 23.6% of respondents estimated their organization used less than one terabyte of data, while only 9.4% of respondents estimated that they would be using less than one terabyte of data this year. In this year's results, 24% of respondents estimated they are currently using less than one terabyte of data, and only 10% estimate they will be using that little data next year.

Implications: Some organizations may be interested in (and ambitious about) increasing the amount of data they are able to analyze and use for their business applications but unable to devote resources to the tools and people necessary to store and use that data properly. Projections for future data storage may be reconsidered when the cost of analysis is taken into account. It's also possible that a greater number of respondents last year focused on how they would deal with larger volumes of data, and that increasingly those people tasked with planning Big Data applications within their organizations are appreciating the other V's.

Recommendations: First, reconsider the volume of data you actually need to store. It's possible that storing and analyzing data at petabyte levels could provide great value to your organization, but you don't want to waste resources on storage for data you will never use. If you do need the volume, and storage is an issue, consider how best to scale your current datastores—or look into other types of datastores completely. Michael Hausenblas's article in this guide, "Five Ways Big Data Has Changed IT Operations," discusses these options.

03. STREAMING DATA, SPARK USAGE INCREASE

Data: Spark is catching fire. In last year's survey, 24% of Hadoop users reported using Apache Spark, an open-source, in-memory cluster computing framework. This year's results show a full 15% increase of that figure, the largest year-over-year growth in Big Data tool usage by far.

Implications: Spark offers the first successful post-Hadoop general-purpose distributed computing platform, with a persistence model that facilitates massive performance gains over Hadoop for many job types. As real-time analytics (velocity) and flexibility of data models (variety) grow increasingly important, more developers are turning to Spark to handle large-scale data processing. (Read the DZone Apache Spark Refcard to get started with Spark right away.)

Recommendations: The impressive growth of Spark usage within the past year does not, of course, make it the only tool you need. In fact, as Dean Wampler discusses in his article "Why Developers Are Flocking to Fast Data and the Spark, Kafka & Cassandra Stack," it's one tool of many that can be combined for high performance in data storage, management, and analysis in certain use cases. As new opportunities arise within Big Data, new tools will become available, and new stacks created . Use multiple tools to get the most out of your data.

Key Research Findings

01. MOST DEVELOPERS ARE OR SOON WILL BE GATHERING AND ANALYZING BIG DATA

Real-world implementation challenges deflate hype quickly. As a result, excitement at the industry level does not always translate to success or even efficiency at the technical level. Fortunately, the buzz-term 'Big Data' is already seen by C-levels in terms that directly affect both developers and data scientists. Each of the 'three V's' of Volume, Velocity, and Variety is significant at both application-level data management and business-level analysis.

Accordingly, 82% of developers have already implemented or are actively seeking to implement large-scale data gathering and analysis. This represents a 7% increase since our last annual survey on Big Data. These numbers indicate that data whose scale is significant in some dimension is already impacting a large majority of software developers, and that the substance-to-hype ratio for Big Data applications continues to grow.

02. MOST BIG DATA INITIATIVES ARE NOT FULLY IMPLEMENTED YET

While most developers are actively dealing with Big Data, the outcomes of most of these projects remain on the horizon. Of the developers working on Big Data projects right now, 74% are either exploring and learning tools and technology for large-scale data gathering and analysis or currently building a proof-of-concept. Furthermore, of the remaining 26% (developers whose Big Data projects are beyond proof-of-concept), just over half are beyond the testing stage. These numbers suggest that development patterns have considerable room to mature in order to shrink the pre-productive R&D cycle for applications that handle Big Data.

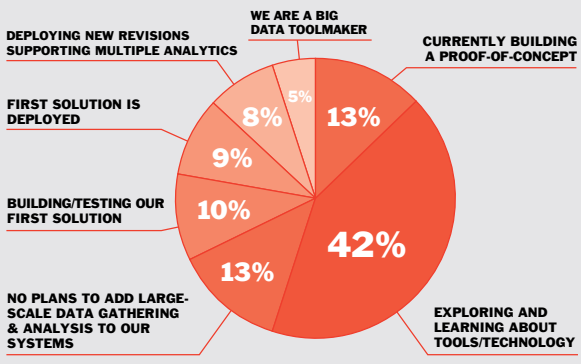
Nevertheless, it is significant that over half of developers surveyed are exploring and/or prototyping systems for large-scale data gathering and analysis. This is the first time a majority of respondents reported practical, forward-looking embrace of Big Data, although the change from last year is incremental (up 6%). Big Data is now less hype than substance, even for development initiatives not yet yielding business value.

03. AUTOMATICALLY GENERATED DATA IS MORE LIKELY TO BE ANALYZED THAN USER GENERATED OR ENTERPRISE GENERATED DATA

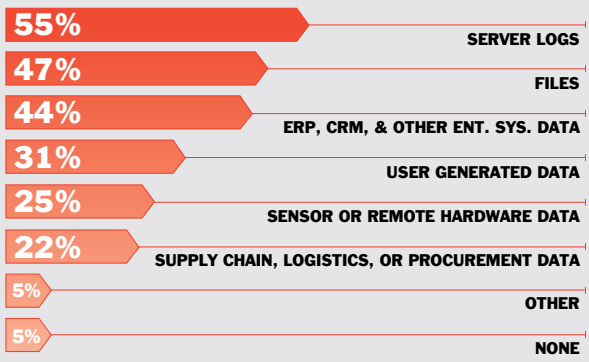
It can be misleading to observe that 90% of data over all time has been created in the past two years alone, insofar as much of this data has become available only because the cost of distributed, high-performance storage and computation resources has dropped to current levels. For example, tools originally developed by Google and Facebook (MapReduce, GFS, BigTable, Cassandra, etc.) to facilitate large-scale search and real-time message availability have allowed business analysts to process, query, and drill down into increasingly large datasets more quickly and with less infrastructure cost. Once these high-capacity storage, query, and workload distribution engines were developed, users began to record data in volumes that could not easily have been analyzed before.

It is not surprising, therefore, that 80% of developers work at organizations that analyze automatically generated data, while only 44% work at organizations that analyze ERP, CRM, and other system data, and a still-smaller share of 31% analyze user-generated data. Furthermore, a large majority of automatically generated data consists of server logs (69%), perhaps the 'thinnest'—but most interesting to developers and IT operations—kind of data available for analysis. Developers are apparently growing more savvy at handling voluminous logs, insofar as usage of Flume, a tool for large-

01. **WHAT IS THE STATUS OF YOUR ORGANIZATION'S LARGE-SCALE DATA GATHERING AND ANALYSIS EFFORTS?**



02. **WHAT DATA SOURCES DOES YOUR ORGANIZATION ANALYZE?**



scale collection, aggregation, and transport of log data in particular, has grown more quickly (up 8% this year) than any other tool for Big Data—except for the more general-purpose Spark framework.

Interestingly, the share of organizations that analyze sensor data did not change from last year, despite evidence of increased adoption of IoT (as surveyed in our *2015 Guide to the Internet of Things*, released just last month).

04. DEVELOPERS ARE MOVING AWAY FROM HADOOP FOR REAL-TIME (BI, SEARCH, OPTIMIZATION) APPLICATIONS

As developers build new Big Data tools optimized for non-batch operations, Hadoop usage is becoming more specialized. One of Hadoop's strengths is its ease of use: the MapReduce abstraction is conceptually simple and suitable for many existing applications, and it saves many hours of costly re-implementation of the same logic. Where real-time streaming and iterative algorithms are not needed, Hadoop usage remains strong. In fact, the percentage of users who use Hadoop for ETL/ELT and data preparation has actually increased over the past year (65% versus 59% last year).

On the other hand, where real-time and (sometimes) iterative processing provides significant value, Hadoop usage has dropped. Developers are now using Hadoop less for reporting/BI (down 4%), search and pattern analysis (down 6%), and optimization analytics (down 3%) than a year ago—all use cases that benefit from the real-time capabilities of frameworks that avoid MapReduce and/or persist distributed datasets in memory.

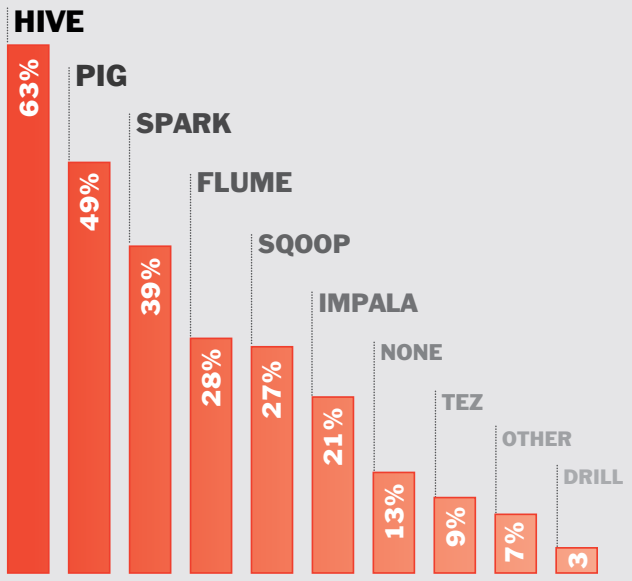
While the refocusing of Hadoop correlates with the increase in Spark usage, additional research is needed to discover which tools and techniques developers are actually using for real-time work. It would be particularly interesting to see how Spark and Storm carve up the set of developers leaving Hadoop for developing real-time applications.

05. MOST DATA PROCESSING CLUSTERS REMAIN SMALL, PERHAPS TOO SMALL TO NEED HADOOP

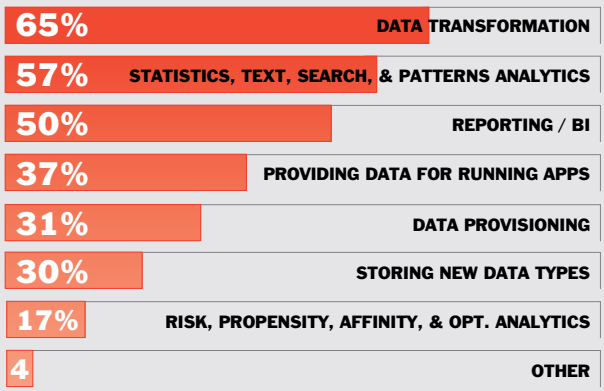
Presumably the need for distributed computation decreases as per-node capabilities (hardware, DBMS technology, load-balancing algorithms, point-of-origin data processing, data management practices, etc.) increase, all other things being equal. But as distributed computing frameworks grow increasingly powerful and easy to use, developers and systems engineers are encountering fewer and fewer difficulties in running and building software for distributed systems. The fundamental complexities generated by process and resource isolation, however, do not disappear. This makes the ease-of-use offered by modern frameworks for distributed computing a potentially hazardous temptation where a single node will do.

In fact, cluster sizes remain typically far below the scales for which Hadoop was originally built. 51% of respondents' organizations process data on clusters containing fewer than five nodes, well below the eight-node floor in Hortonworks' Hadoop cluster sizing guide. Moreover, only 9% of respondents' clusters exceed 9 nodes. While Hadoop MapReduce and HDFS are perfectly able to run on a single node, developers and IT managers should weigh the drawbacks of multi-node complexity against the advantages offered by lower-end commodity hardware.

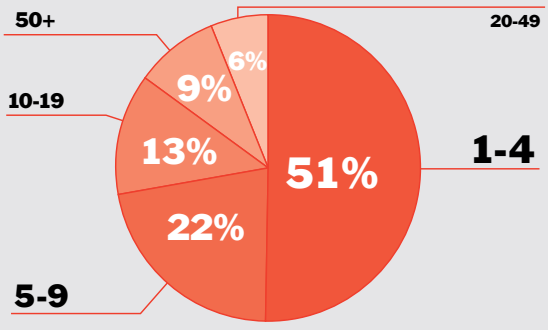
03. **WHICH TOOLS DOES YOUR COMPANY USE TO HELP MANAGE DATA WITH HADOOP?**



04. **WHAT DOES YOUR COMPANY USE HADOOP FOR?**



05. **HOW MANY NODES ARE TYPICALLY IN YOUR ORGANZATION'S DATA PROCESSING CLUSTERS?**



Workload and Resource Management: YARN, Mesos, and Myriad

BY ADAM DIAZ

QUICK VIEW

01 YARN and Mesos both have a container-based architecture.

02 Mesos was meant as a more generic solution inclusive of Hadoop and YARN.

03 Myriad is meant for scheduling of YARN jobs via Mesos.

04 All these technologies have to do with building a distributed architecture.

Workload management on distributed systems tends to be an afterthought in implementation. Anyone who has used a compute cluster for a job-dependent function quickly discovers that placement of work and its prioritization are paramount not only in daily operations but also in individual success. Furthermore, many organizations quickly find out that even with robust job placement policies to allow for dynamic resource sharing, multiple clusters become a requirement for individual lines of business based upon their requirements. As these clusters grow, so does so-called data siloing.

Over time, the amount of company computer power acquired could eclipse any reasonable argument for individual systems if used in a programmatically shareable way. A properly shared technology allows for greater utilization across all business units, allowing some organizations to consume beyond their reasonable share during global lulls in workloads across the organization. The unique requirements of people, business units, and business as a whole make the development of such global sharing difficult—if not impossible. This brings the need for workload and resource management into sharp relief. This article will describe the latest advances in Big Data-based workload and resource management.

YARN

Much has been written about YARN, and it is well described in many places. For context, I offer a high-level overview: YARN is essentially a container system and scheduler designed primarily for use with a Hadoop-based cluster. The containers in YARN are capable of running many types of tasks including MPI, web servers, and virtually anything else one would like. YARN containers might be seen as difficult to write, giving rise to other projects like what was once called HOYA (Hbase on YARN, which was eventually coined Apache Slider) as an attempt at providing a generic implementation of easy-to-use YARN containers. This allowed for a much easier entry point for those wishing to use YARN over a distributed file system. This has been used mainly for longer running services like HBase and Accumulo, but will likely support other services as it moves out of incubator status.

YARN, then, was expected to be a cornerstone of Hadoop as an operating system for data. This would include HDFS as the storage layer along with YARN scheduling processes for a wider variety of applications well beyond MapReduce.

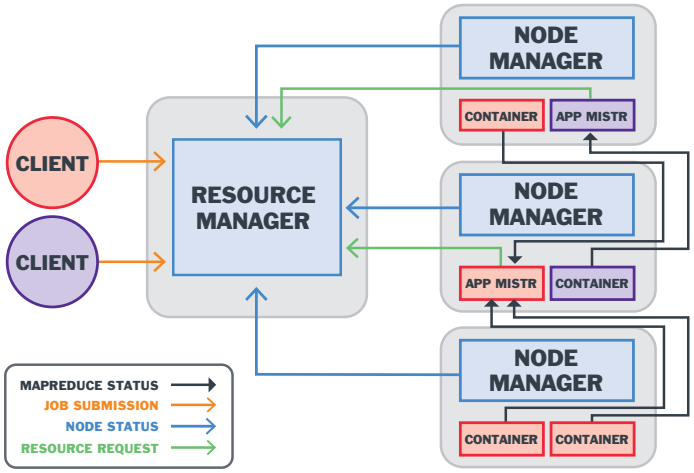


Figure 1: YARN Architecture—Basic Structure of the YARN Job Submission Process

The implementation, while novel and liberating, leaves room for improvement in several ways. In order to build a novel distributed computer one needs more than just a CPU/scheduler and data storage. This process in many ways mirrors the development of an operating system. In Linux it requires security, distributed configuration management, and workload management—all done with the intention of harnessing the power of multiple hardware systems into a single virtual pool of resources. As luck would have it, the Open Source community offers the benefit of multiple solutions.

MESOS

Enter Apache Mesos. While some have described Mesos as a "meta scheduler," or a scheduler of schedulers, its creators have more aptly called Mesos a "distributed systems kernel." Mesos essentially uses a container architecture but is abstracted enough to allow seamless execution of multiple, sometimes identical, distributed systems on the same architecture, minus the resource overhead of virtualization systems. This includes appropriate resource isolation while still allowing for data locality needed for frameworks like MapReduce.

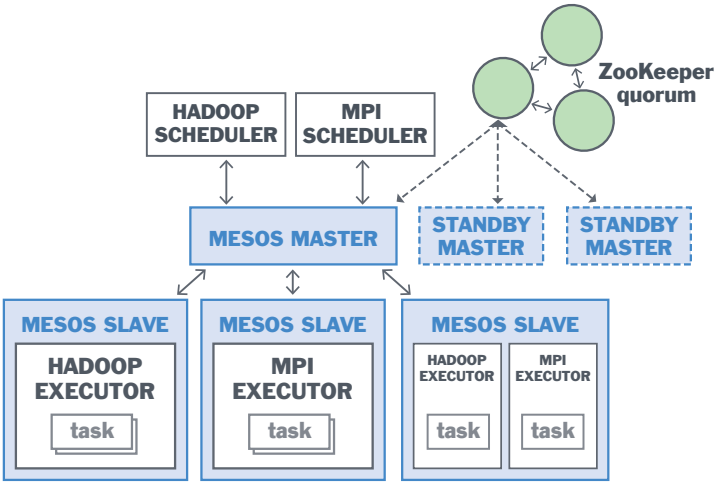


Figure 2: Mesos Architecture—High-level Representation of Mesos Daemons Running Both Hadoop- and MPI-based Jobs

As one might expect, Mesos uses a very familiar nomenclature and includes container-based architecture. It should be noted that in function, Mesos assigns jobs very differently. Mesos is considered a two-level scheduler and is described by its creators as a distributed kernel used to abstract system resources. Mesos itself is really an additional layer of scheduling on top of application frameworks that each bring their own brand of scheduling. Application schedulers interface with a Mesos master setup in a familiar Zookeeper-coordinated active-passive architecture, which passes jobs down to compute slaves to run the application of choice.

Mesos is written in C, not Java, and includes support for Docker along with other frameworks. Mesos, then, is the core of the Mesos Data Center Operating System, or DCOS, as it was coined by Mesosphere.

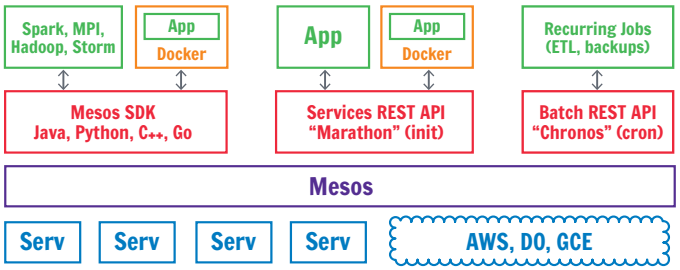


Figure 3: Mesosphere DCOS Architecture—Representation of the DCOS From Mesosphere

This Operating System includes other handy components such as Marathon and Chronos. Marathon provides cluster-wide "init" capabilities for applications in containers like Docker or cgroups. This allows one to programmatically automate the launching of large cluster-based applications. Chronos acts as a Mesos API for longer-running batch-type jobs while the core Mesos SDK provides an entry point for other applications like Hadoop and Spark.

In many of these architectures there still can exist hard partitioning of resources even though scheduling may be centralized. The true goal is a full shared, generic and reusable on-demand distributed architecture. Announced during the authoring of this article is a new offering from Mesosphere using DCOS called Infinity, which is used to package and integrate the deployment of clusters in just such a way. Out of the box it will include Cassandra, Kafka, Spark, and Akka. This is current available as an early access project.

MYRIAD

In order to directly integrate YARN with Mesos, the Apache Myriad project was formed. The initial goals included making the execution of YARN work on Mesos scheduled systems transparent, multi-tenant, and smoothly managed. The high-level architecture promises to allow Mesos to centrally schedule YARN work via a Mesos-based

framework, including a REST API for scaling up or down. The system also includes a Mesos executor for launching the node manager as shown in the following diagram.

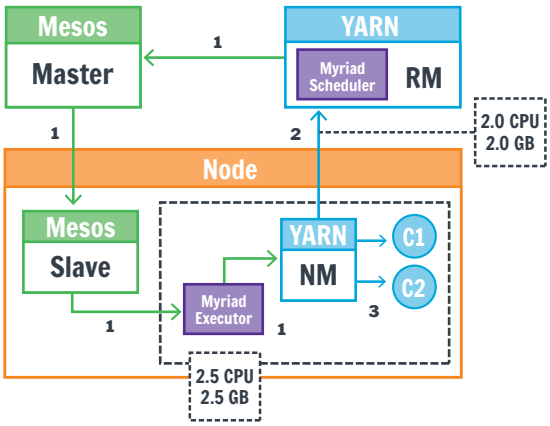


Figure 5: Mesos/YARN Interaction—YARN Container Launching via Mesos Management

From a high level, this type of architectural abstraction might seem like common sense. It should be noted, however, that it has taken some time to evolve and mature each of these systems, which in themselves are fields of study worthy of extensive analysis. This new Myriad (Mesos-based) architecture allows for multiple benefits over YARN-based resource management. It makes resource management generic and, therefore, the use of the overall system more flexible. This includes running multiple versions of Hadoop and other applications using the same hardware as well as operational flexibility for sizing. It also allows for use cases such as the same hardware for development, testing, and production. Ultimately, this type of flexibility has the greatest promise to fulfill the ever-changing needs of the modern data-driven architecture.

A generic, multi-use architecture, encompassing a distributed persistence layer along with the ability to engage in use cases from batch to streaming (real time/near real time) scheduled dynamically yet discretely over a commodity compute infrastructure seems to be the holy grail of many a project today. Mesos and Myriad seem to be two projects well on their way to fulfilling that dream. There is still a great deal of work to be done, including use cases of jobs that span geography, along with associated challenges such as cross-geography replication and high availability.

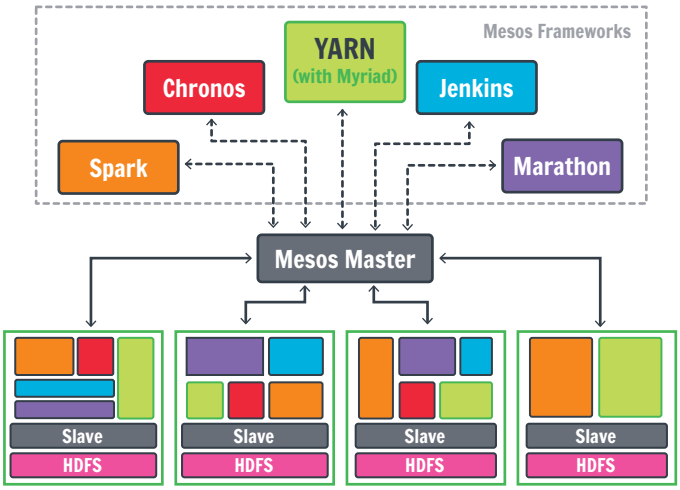


Figure 6: Myriad Architecture—Multi-tenant Mesos Architecture Including YARN-based Hadoop Workloads

> **In order to build a novel distributed computer, one needs more than just a CPU/scheduler and data storage.**

CONCLUSION

The discussion of high-level architectures really starts to bring into play the concept's overall solution architectures. Candidates include the Lambda and its more modern successor, the Zeta Architecture. Generic components like a distributed persistence layer, containerization, and the handling of both solution and enterprise architecture are hallmarks of these advanced architectures. The question of how to best use resources in terms of first principal componentry is being formed and reformed by a daily onslaught of new technology. What is the best storage layer tech? What is the best tech for streaming applications? All of these questions are commonly asked and hotly debated. This author would argue that "best" in this case is the tool or technology that fits the selection criteria specific to your use case. Sometimes there is such a thing as "good enough" when engineering a solution for any problem. Not all the technologies described above are needed by every organization, but building upon a solid foundational framework that is dynamic and pluggable in its higher layers will be the solution that eventually wins the day.

REFERENCES:
mesosphere.github.io/marathon
nerds.airbnb.com/introducing-chronos
mesos.apache.org
apache-myriad.org
events.linuxfoundation.org/sites/events/files/slides/aconna15_bordelon.pdf
mapr.com/developer-preview/apache-myriad

ADAM DIAZ is a long-time Linux geek and fan of parallel and distributed systems. Adam cut his teeth working for companies like Platform Computing and Hortonworks. Adam also has a deep background in analytic tooling on distributed systems including SAS, R and now Spark. His current endeavor is working for MapR Technologies in the rapidly evolving and highly competitive field of Hadoop. He can be reached at www.techtonka.com.

THE LEADER IN BIG DATA CONSULTING.

By combining new data technologies like Hadoop and Spark with a high-level strategy, we turn unstructured information into real business intelligence.

OUR SERVICES

- CREATING MODERN INFRASTRUCTURES FOR DATA-DRIVEN ORGANIZATIONS
- DESIGNING SYSTEMS TO SCALE FOR TODAY'S EVER-GROWING AMOUNT OF DATA
- ENABLING COMPANIES TO PERFORM COMPREHENSIVE ANALYSIS ON LARGE AMOUNTS OF DATA

MAMMOTHDATA.COM | (919) 321-0119

Why Developers Are Flocking to Fast Data and the Spark, Kafka & Cassandra Stack

BY DEAN WAMPLER

QUICK VIEW

01 The first generation of Big Data was primarily focused on data capture and offline batch mode analysis.

02 Spark Streaming, Apache Kafka, and Apache Cassandra have emerged as a very powerful "stack" for mini-batch processing.

03 Fast Data is a new movement that describes new systems and approaches focused on timely, cost-efficient data processing, as well as higher developer productivity.

One of the most noteworthy trends for Big Data developers today is the growing importance of speed and flexibility for data pipelines in the enterprise.

Big Data got its start in the late 1990s when the largest Internet companies were forced to invent new ways to manage data of unprecedented volumes. Today, when most people think of Big Data, they think of Hadoop [1] or NoSQL databases. However, the original core components of Hadoop—HDFS (Hadoop Distributed File System for storage), MapReduce (the compute engine), and the resource manager now called YARN (Yet Another Resource Negotiator)—were, until recently, rooted in the "batch mode" or "offline" processing commonplace. Data was captured to storage and then processed periodically with batch jobs. Most search engines worked this way in the beginning; the data gathered by web crawlers was periodically processed into updated search results.

The first generation of Big Data was primarily focused on data capture and offline batch mode analysis. But the new "Fast Data" trend has concentrated attention on narrowing the time gap between data arriving and value being extracted out of that data.

The opposite end of the spectrum from batch data is real-time event processing, where individual events are processed as soon as they arrive with tight time constraints, often microseconds to milliseconds. High-frequency trading systems are one example, where market prices move quickly, and real-time adjustments control who wins and who loses. Between the extremes of batch and real-time are more general stream processing models with less stringent responsiveness guarantees. A popular example is the mini-batch model, where data is captured in short time intervals and then processed as small batches, usually within time frames of seconds to minutes.

FAST DATA DEFINED

The phrase "fast data" captures this range of new systems and approaches, which balance various tradeoffs to deliver timely, cost-efficient data processing, as well as higher developer productivity. Let's begin by discussing an emerging architecture for fast data.

What high-level requirements must a Fast Data architecture satisfy? They form a triad:

- Reliable data ingestion
- Flexible storage and query options
- Sophisticated analytics tools

The components that meet these requirements must also be reactive (meaning they scale up and down with demand); resilient against failures that are inevitable in large distributed systems; responsive to service requests (even if

failures limit the ability to deliver services); and driven by events from the world around them.

THE NEW STACK EMERGES FOR FAST DATA

For applications where real-time, per-event processing is not needed—where mini-batch streaming is all that's required—research [3] shows that the following core combination of tools is emerging as very popular: Spark Streaming, Kafka, and Cassandra. According to a recent Typesafe survey, 65% of respondents use or plan to use Spark Streaming, 40% use Kafka, and over 20% use Cassandra.

Spark Streaming [4] ingests data from Kafka, databases, and sometimes directly from incoming streams and file systems. The data is captured in mini-batches, which have fixed time intervals on the order of seconds to minutes. At the end of each interval, the data is processed with Spark's full suite of APIs, from simple ETL to sophisticated queries, even machine learning algorithms. These other APIs are essentially batch APIs, but the mini-batch model allows them to be used in a streaming context.

> **The new "Fast Data" trend has concentrated attention on narrowing the time gap between data arriving and value being extracted out of that data.**

This architecture makes Spark a great tool for implementing the Lambda Architecture [5], where separate batch and streaming pipelines are used. The batch pipeline processes historical data periodically, while the streaming pipeline processes incoming events. The result sets are integrated in a view that provides an up-to-date picture of the data. A common problem with this architecture is that domain logic is implemented twice [6], once for the streaming pipeline and once for the batch pipeline. But code written with Spark for the batch pipeline can also be used in the streaming pipeline, using Spark Streaming, thereby eliminating the duplication.

Kafka [7] provides very scalable and reliable ingestion of streaming data organized into user-defined topics. By focusing on a relatively narrow range of capabilities, it does what it does very well. Hence, it makes a great buffer between downstream tools like Spark and upstream sources of data, especially those sources that can't be queried again in the event that data is, for some reason, lost downstream.

Finally, records can be written to a scalable, resilient database, like Cassandra, Riak, or HBase; or to a distributed filesystem, like HDFS or S3. Kafka might also be used as a temporary store of processed data, depending on the downstream access requirements.

THE BUSINESS CASE FOR FAST DATA

Most enterprises today are really wrangling data sets in the multi-terabyte size range, rather than the petabyte size range typical of the large, well-known Internet companies. They want to manipulate and integrate different data sources in a wide variety of formats, a strength of Big Data technologies. Overnight batch processing of large datasets was the start, but it only touched a subset of the market requirements for processing data.

Now, speed is a strong driver for an even broader range of use cases. For most enterprises, that generally means reducing the time between receiving data and when it can be processed into information. This can include traditional techniques like joining different datasets, and creating visualizations for users, but in a more "real-time" setting. Spark thrives in working with data from a wide variety of sources and in a wide variety of formats, from small to large sizes, and with great efficiency at all size scales.

You tend to think of open source disrupting existing markets, but this streaming data / Fast Data movement was really born outside of commercial, closed-source data tools. First you had large Internet companies like Google solving the fast data problem at a scale never seen before, and now you have open-source projects meeting the same needs for a larger community. For developers diving into Fast Data, the Spark / Kafka / Cassandra "stack" is the best place to start.

RESOURCES

[1] hadoop.apache.org
[2] reactivemanifesto.org
[3] typesafe.com/blog/apache-spark-preparing-for-the-next-wave-of-reactive-big-data
[4] spark.apache.org/streaming
[5] lambda-architecture.net
[6] radar.oreilly.com/2014/07/questioning-the-lambda-architecture.html
[7] kafka.apache.org
[8] cassandra.apache.org
[9] github.com/basho/riak
[10] hbase.apache.org
[11] hadoop.apache.org/docs/r1.2.1/hdfs_design.html
[12] aws.amazon.com/s3

DEAN WAMPLER, PH.D., is the Architect for Big Data Products and Services in the Office of the CTO at Typesafe. He specializes in scalable, distributed, "Big Data" application development using Spark, Mesos, Hadoop, Scala, and other tools. Dean is a co-organizer and frequent speaker at conferences worldwide, the co-organizer of several Chicago-area Meetup groups, and a contributor to several open source projects. He is the author of "Programming Scala, 2nd Edition" and "Functional Programming for Java Developers," and the co-author of "Programming Hive," all from O'Reilly.

FUSE BIG DATA AND FAST DATA

with THINGSPAN®

Objectivity's ThingSpan is a database platform for developing and managing advanced information fusion solutions. It combines the power of object data modeling technology with the high-performance, parallel processing of Hadoop and Spark to deliver an easier, more effective way of supporting mission-critical applications at Big Data scale.

CGG

"A technically superior approach to application development.

Dräger

"High availability with zero administration effort.

Objectivity®

3099 North First Street, Suite 200
San Jose, CA 95134 USA
408-992-7100

twitter.com/objectivitydb
facebook.com/ObjectivityInc
linkedin.com/company/objectivity

A New Approach to Analyzing Time-Series Data

These days, most large organizations have a plan to collect and analyze their Big Data assets from many sources. One area of interest for many companies is time-series (T-S) data, which refers to data related to specific points or durations of time.

T-S data can be analyzed through both structured queries to assess how two discrete data points relate to one another, and through unstructured analysis that aggregates a spectrum of data that doesn't lend itself to simple parameters.

The Industrial Internet of Things has made it possible to accumulate large volumes of streaming T-S data. Consider, for instance, a seismic data survey for oil and gas exploration. A typical seismic survey can collect as much as 35 petabytes of data, which can be used to develop highly precise 3D maps. The goal is to gather and analyze this data as quickly and accurately as possible.

Big Data, including T-S data, can be analyzed through the Hadoop Distributed File System (HDFS) with a traditional or NoSQL database management system, but because such systems typically rely on batch processing of data, this type of system is not ideal for assessing real-time, streaming data.

Instead, it can be beneficial to use an Object Database Management System (ODBMS), which can store data about real-world "objects," such as physical locations, keeping relationships between data points intact, so that such data does not need to be queried separately for each use.

> **Using an ODBMS can save up to ⅓ of the development effort over a traditional or NoSQL database management system.**

Using an ODBMS can save up to one-third of the development effort over a traditional or NoSQL database management system, and can also dramatically reduce the storage overheads needed to maintain indices and links between objects.

Organizations with real-time time-series data analysis requirements should investigate whether an ODBMS is right for their business needs. Visit objectivity.com to learn more.

WRITTEN BY LEON GUZENDA
CHIEF TECHNOLOGY OFFICER, **OBJECTIVITY, INC**

ThingSpan by Objectivity

ThingSpan combines object data modeling with the parallel processing of Hadoop and Spark to support applications at Big Data scale.

CASE STUDY

CGG, a leader in fully integrated geoscience services, has been working with Objectivity to develop a common platform for its major geoscience analytical software. The data challenge comes from having to integrate diverse geoscience data, and then providing reservoir modeling and simulation data on top of this integrated data set. This common data model provided by CGG and Objectivity enables analysts to work on thousands of wells with data physically located anywhere in the world, thereby resulting in quick collaboration on important drilling decisions. The scale-out computing model of ThingSpan and native support of Hadoop allows higher performance at a lower cost for analysis of large, multi-dimensional data associated with geoscience.

CLASSIFICATION

Data Management, Data Integration

HOSTING

On-Premise

FEATURES

- Built on Hadoop
- Native Stream Processing Capabilities
- Monitoring Tool Included
- Supports Spark

NOTABLE CUSTOMERS

- CGG
- Siemens
- Drager

BLOG objectivity.com/about-us/blog | **TWITTER** @objectivitydb | **WEBSITE** objectivity.com

5 Ways Big Data Has Changed IT Operations

QUICK VIEW

01

Today's hottest Big Data technologies each carry their own specific operational considerations that must be mastered.

02

Almost all Big Data solutions assume a scale-out architecture that leads enterprises into advanced cluster scheduling and orchestration.

03

There are five key trends that operations teams should anticipate related to packaging, testing, and evolving Big Data platforms.

BY MICHAEL HAUSENBLAS

With the uptick of Big Data technologies such as Hadoop, Spark, Kafka, and Cassandra, we are witnessing a fundamental change in how IT operations are carried out. Most, if not all, of said Big Data technologies are inherently distributed systems, and many of them have their roots in one of the nowadays dominating Web players, especially Google. But how does using these Big Data technologies impact the daily IT operations of a company aiming to benefit from them? To address this question, we take a deeper look at five trends you and your Ops team should be aware of when employing Big Data technologies. These trends emerged in the past 15 years and are of a technological as well as organizational nature.

01. FROM SCALE-UP TO SCALE-OUT

There is a strong tendency throughout all verticals to deploy clusters of commodity machines connected with low-cost networking gear rather than the specialized, proprietary, and typically expensive supercomputers. While likely older than 15 years, Google has spearheaded this movement with its Warehouse-Scale Computing Study. Almost all of the currently available Big Data solutions (especially those that are open source, but more on this point below) implicitly assume a scale-out architecture. Need to crunch more data? Add a few machines. Want to process the data faster? Add a few machines.

The adoption of the 'commodity cluster' paradigm, however, has two implications that are sometimes overlooked by organizations starting to roll out solutions:

1. With the ever-growing number of machines, sooner or later the question arises if a pure on-premise deployment is sustainable as you will need the space and pay hefty energy bills while typically seeing cluster utilizations less than 10%.

2. The current best practice is to effectively create a dedicated cluster for each technology. This means you have a Hadoop cluster, a Kafka cluster, a Storm cluster, a Cassandra cluster, etc.—not only because this siloes issues (in terms of being able to swiftly react to business needs; for example, to accommodate different seasons), but also because the overall TCO tends to increase.

The issues discussed above do not mean you can't successfully deploy Big Data solutions in your organization at scale; it simply means that you need to be prepared for the long-term operational consequences, such as opex vs. capex, as well as migration scenarios.

02. OPEN SOURCE RULEZ

Open Source plays a fundamental role in Big Data technologies. Organizations adopt it to avoid vendor lock-in

and to be less dependent on external entities for bug fixes, or simply to adapt software to their specific needs. The open and usually community-defined APIs ensure transparency; and various bodies, such as the Apache Software Foundation or the Eclipse Foundation, provide guidelines, infrastructure, and tooling for the fair and sustainable advancement of these technologies. Lately, we have also witnessed the rise of foundations such as the Open Data Platform, the Open Container Initiative, or the Cloud Native Computing Foundation, aiming to harmonize and standardize the interplay and packaging of infrastructure and components.

> **While the software might be open source and free to use, one still needs the expertise to efficiently and effectively use it.**

As in the previous case of the commodity clusters, there is a gotcha here: there ain't no such thing as a free lunch. That is, while the software might be open source and free to use, one still needs the expertise to efficiently and effectively use it. You'll find yourself in one of two camps: either you're willing to invest the time and money to build this expertise in-house—for example, hire data engineers and roll your own Hadoop stack—or you externalize it by paying a commercial entity (such as a Hadoop vendor) for packaging, testing, and evolving your Big Data platform.

03. THE DIVERSIFICATION OF DATASTORES

When Martin Fowler started to talk about polyglot persistence in 2011, the topic was still a rather abstract one for many people—although Turing Award recipient Michael Stonebraker made this point already in his 2005 paper "'One Size Fits All': An Idea Whose Time Has Come and Gone." The omnipotent and dominant era of the relational database is over, and we see more and more NoSQL systems gaining mainstream traction.

What this means for your operations: anticipate the increased usage of different kinds of NoSQL datastores throughout the datacenter, and be ready to deal with the consequences. Challenges that typically come up include:

- Determining the system of record
- Synchronizing different stores
- Selecting the best fit for the datastore to use for a certain use case, for example a multimodal database like ArangoDB for rich relationship analysis, or a key-value store such as Redis for holding shopping basket data.

04. DATA GRAVITY & LOCALITY

In your IT operations, you'll usually find two sorts of services: stateless and stateful. The former includes things like a Web server while the latter almost always is, or at least contains, a datastore. Now, the insight that data has gravity is especially relevant for stateful services. The implication here is to consider the overall cost associated with transferring data, both in terms of volume and in tooling, if you were to migrate for disaster recovery reasons or to a new datastore altogether (ever tried to restore 700TB of backup from S3?).

Another aspect of data gravity in the context of crunching data is known as data locality: the idea of bringing the computation to the data rather than the other way round. Making sure your Big Data technology of choice benefits from data locality (e.g. Hadoop, Spark, HBase) is a step in the right direction; using appropriate networking gear (like 10GE) is another. As a general note: the more you can multiplex your cluster (that is, running different services on the same machines), the better you're prepared.

05. DEVOPS IS THE NEW BLACK

The last trend here is not necessarily Big Data specific, but surprisingly often overlooked in a Big Data context: DevOps. As it was aptly described in the book *The Phoenix Project*, DevOps refers to the best practices for collaboration between the software development and operational sides of an organization. But what does this mean for Big Data technologies?

It means that you need to ensure that your data engineer and data scientist teams use the same environment for local testing as is used in production. For example, Spark does a great job allowing you to go from testing to cluster submission. In addition, for the mid-to-long run, you should containerize the entire production pipeline.

CONCLUSION

With the introduction of Big Data technologies in your organization, you can quickly gain actionable business insights from raw data. However, there are a few things you should plan for from the IT operations point of view, including where to operate the cluster (on-premise, public cloud, hybrid), what your strategy is concerning open source, how to deal with different datastores as well as data gravity, and, last but not least, how to set up the pipeline and the organization in a way developers, data engineers, data scientists, and operations folks can and will want to work together to reap the benefits of Big Data.

MICHAEL HAUSENBLAS is a Datacenter Application Architect at Mesosphere. His background is in large-scale data integration research, the Internet of Things, and Web applications. He's experienced in advocacy and standardization (World Wide Web Consortium, IETF). Michael frequently shares his experiences with the Lambda Architecture and distributed systems through blog posts and public speaking engagements and is a contributor to Apache Drill. Prior to Mesosphere, Michael was Chief Data Engineer EMEA at MapR Technologies, and prior to that a Research Fellow at the National University of Ireland, Galway, where he acquired research funding totalling over €4M, working with multinational corporations such as Fujitsu, Samsung and Microsoft as well as governmental agencies in Ireland.

MINING THE BIG DATA
GOLD RUSH

The Big Data pipeline is not a linear process—it is a complex system of interconnecting components, platforms, and feedback loops. To better understand how these practices and tools connect, we've illustrated a basic model of the Big Data pipeline. And, we know... there's so much more than what we can show here on the page! Beyond the segment of the pipeline we show here is a whole other territory of predictive analytics and business intelligence worth exploring. For now, let's take a look at five important stages for dealing with Big Data.

START

01 COLLECT

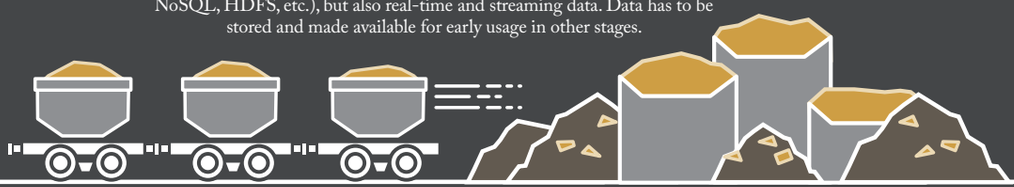
This is where data is defined and collected. The collection stage deals with streaming, real-time, and bulk import data (data that has been previously collected). Data volume and velocity are major concerns at this stage.

DARK DATA

Data that's been previously collected, stored, and processed, and which occurs at many stages. Dark data is often thought to be useless, or just not being used. New technologies and methodologies are allowing for this data to be processed and made useful.

02 STORE & TRANSPORT

This stage deals with scalable and extensible data storage systems (RDBMS, NoSQL, HDFS, etc.), but also real-time and streaming data. Data has to be stored and made available for early usage in other stages.



03 TRANSFORM & ACCESS

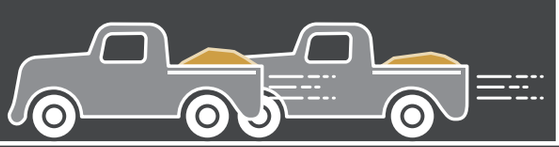
At this point, data is being made accessible for cleaning, sorting, and staging. The data is being transformed into useable data sets, file formats are changing, and compression algorithms are being applied. Data is being catalogued and made accessible.

04 MODEL & ANALYZE

Information about the data itself—and the way in which it was processed and collected—is being used to validate sets of data, model further subsets of data, build aggregations for reporting data, and perform complex event processing. Machine learning and NLP practices are being applied at this stage, as well as various forms of in-memory analytics.

05 VISUALIZE & PREDICT

Processed, modeled, and analyzed data is now becoming available for visualization and advanced analytics. Data is being interpreted and acted upon, and further studied for predictive analytics. Business intelligence tools are creating actionable information for organizations and teams.



Executive Insights on Big Data

BY TOM SMITH

QUICK VIEW

01 Big Data isn't going anywhere. It's just going to get bigger. Unfortunately so are expectations.

02 Ask plenty of questions to understand what needs to be accomplished with the data.

03 Failure to set realistic expectations upfront can lead to wasted effort and disappointed clients.

To more thoroughly understand the state of Big Data, and where it's going, we interviewed 14 executives with diverse backgrounds and experience with Big Data technologies, projects, and clients.

Specifically, we spoke to:

***Margaret Roth**, Co-Founder and CMO, Yet Analytics · **Dr. Greg Curtin**, CEO and Founder, Civic Resource Group · **Guy Kol**, Founder and V.P. R&D, NRGene, Ness Ziona, Israel · **Gena Rotstein**, CEO and Founder, Dexterity Ventures, Inc. · **Scott Sundvor**, CTO, 6SensorLabs · **Ray Kingman**, CEO, Semcasting · **Puneet Pandit**, Founder and CEO, Glassbeam · **Mikko Jarva**, CTO Intelligent Data, Comptel Corporation · **Vikram Gaitonde**, Vice President Products, Solix Technologies · **Dan Potter**, CMO, Datawatch · **Paul Kent**, SVP Big Data, SAS · **Matt Pfeil**, CCO Co-Founder, DataStax · **Philip Rathle**, VP Products, Neo Technology, Inc. · **Hari Sankar**, VP of Product Management, Oracle*

There is alignment with regards to what Big Data is, how it can be used, and its future. Discrepancy lies in the perception of the state of Big Data today. Some companies have been working with Big Data for years, others feel unable to perform "real" analytics work due to the data hygiene required, as well as the necessary integration of disparate databases.

Here's what we learned from the conversations.

01

The definition of Big Data is consistent across executives and industries—volume, velocity, and variety of data that is always changing and growing exponentially beyond what companies can traditionally handle.

Scalability is critical, as are data management and retention policies. Data collection requires a more strategic approach. Companies will evolve from collecting/storing every piece of data to collecting and storing data based on need.

02

Executives stay abreast of industry trends by **meeting with clients and prospects**, learning pain points, and determining which data is available to solve the problem.

Just as there's a tsunami of data, there's also a tsunami of information and hype about Big Data. Stay above the noise by having a "big picture" perspective of the problem you are solving.

03

Real world problems solved by Big Data are myriad. I spoke with companies sequencing the wheat genome; enabling smart cities; and evolving healthcare, automotive, retail, education, media, and beyond. Every initiative is using data to help clients move from being *reactive* to *proactive.*

Accessing data, integrating multiple sources, and providing the analysis to solve problems requires patience, vision, and knowledge. You will gain all three by working in a real-world environment solving real problems. None will come from contemplating Big Data in the abstract.

Once you appreciate the amount of time spent on data hygiene—an absolute requirement before any analysis can take place—you'll structure data collection and integration so hygiene is less tedious and time-consuming.

04

The composition of a **data analytics team requires a number of skills**: development of algorithms and software

implementations, data science, design, engineering, and input from *analysts with domain expertise*. The most important qualities for team members are creativity, collaboration, and curiosity. No one person or skill-set is the solution to every Big Data project.

Big Data provides an opportunity for developers to contribute beyond their typical scope of influence. It's best for developers to have a broad range of interests and expertise. The more perspectives they can bring to bear on the problem, the better.

05

According to Ginni Rometty, CEO of IBM, **Big Data is the "next oil."** Several executives pointed out that this oil will be "unrefined" for the next 10 to 20 years. The future of Big Data is in providing real-time data to connect people, machines, experiences, and environments to improve life in a more personal way—from fewer traffic jams to more sustainable agriculture.

Some executives I spoke with believe no one is really dealing with Big Data yet. There is so much data in repositories that the challenge is to figure out how to aggregate data so it can be analyzed. We also need to determine the right questions to ask, and the right data to store, to transform business and the customer experience. Other executives are already doing these things for their clients; however, even these executives see unrealized possibilities.

Demand for Big Data services is growing quickly as the business world sees the possibilities. Once you empower business people, they ask for more information. They ask smarter questions. Speed and agility gain importance. Real-time operational and business data allows people to make well-informed decisions quickly, thereby saving time and money. Effective use of Big Data is becoming an expectation.

06

Hadoop was the most frequently mentioned software, with Cloudera and Hortonworks being the most frequently mentioned management applications. However, many other solutions—including Cassandra, Clojure, Datomic, Hype, NoSQL, PostgreSQL, SQL Server, and Tableau for visualization—were discussed.

There's enormous demand for Big Data developers, so you don't need to know all of the software and applications. Pick what you want to become an expert in and write your ticket with that software. Taking the time to learn Hadoop is a good place to start.

07

Executives identified a **broad range of obstacles for success** with clients. The only obstacle for success mentioned by multiple executives was the lack of sufficiently knowledgeable and experienced people. Other concerns included: legacy software systems, fear of what's in legacy data, lack of understanding of the value Big Data can provide, knowing the right questions to ask, knowing who owns the data in the cloud, vendors making unsubstantiated claims, and too much hype around Big Data.

These obstacles are not unexpected given how early we are in the development and execution of Big Data projects. However, be aware of them as you get involved with specific projects. Ask the right questions up front, set the right expectations, and save a lot of time and rework.

08

Concerns around Big Data are similar to IoT except **privacy is more important than security**. Industrial data is one thing, personal data is a whole other animal. As long as personal data is used for good, people will get comfortable as they benefit from Big Data. While Big Data will result in greater knowledge, it should also result in greater transparency—by governments, companies, and advertisers—and help prevent fraud and identity theft.

"Data lakes" should be weighed against the danger to privacy posed by centralized data stores. Before we build huge data repositories, we need to know how we're going to use and safeguard that data. As the data infrastructure matures, these problems will become increasingly easy to solve.

09

The future of Big Data is the ability to make well-informed decisions quicker and easier than ever before. People will not be doing what a machine can do, so they'll be free to use their minds to do creative things. The blue-sky vision is: Big Data will be the central technology to human existence, since it will affect all aspects of life (e.g., weather, transportation, healthcare, nutrition, energy, etc.).

Based on the vision above, following are three takeaways for developers.

- **Big Data is evolutionary, not revolutionary.** Big Data problems are similar to problems you've faced before. Leverage and improve upon the knowledge you already have by learning new architectures and languages.
- Be prepared to **be part of the bigger picture.** Become more well-rounded and more prepared to collaborate with, and contribute to, your team.
- Understand the real-world problems you are solving. (It may help to think in terms of Domain-Driven Design.) Think about creating the next destructive idea that's lurking in the open source community. Share more, borrow more, be more open-minded about the possibilities of what you are working on.

The executives we spoke with are working on their own products or serving clients. We're interested in hearing from developers, and other IT professionals, to see if these insights offer real value. Is it helpful to see what other companies are working on from a more industry-level perspective? We welcome your feedback at research@dzone.com.

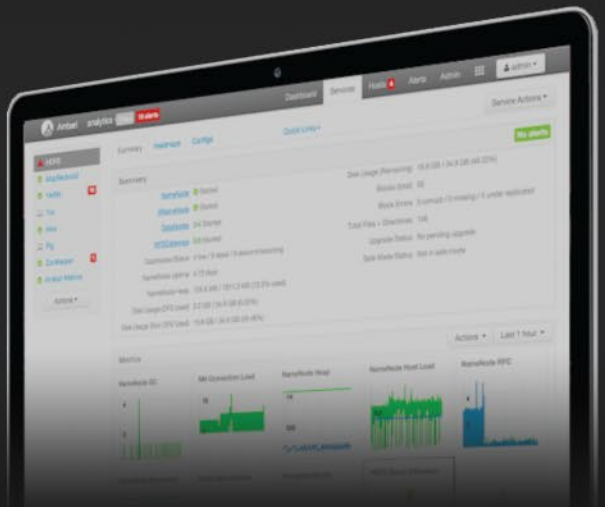
TOM SMITH is a Research Analyst at DZone who excels at gathering insights from analytics—both quantitative and qualitative—to drive business results. His passion is sharing information of value to help people succeed. In his spare time, you can find him either eating at Chipotle or working out at the gym.

Get Started With Hadoop Development

Hortonworks Sandbox: A personal, portable Hadoop environment that comes with a dozen interactive Hadoop tutorials

Get up and running with
Big Data in 15 minutes!

hortonworks.com/sandbox



Hadoop for Developers

HADOOP IS TRANSFORMING EVERY INDUSTRY, ANY DATA, MANY APPLICATIONS

Mass adoption of Apache Hadoop and its ecosystem of technologies in the enterprise has spurred an insatiable demand for data scientists and Hadoop developers. These enterprise projects are being driven by both public and private sector initiatives across an increasingly wide range of use cases. Predictive analytics (what is going to happen based on history), single view of customer (connecting all of a customer's touch points with an organization to provide the best possible service) and enabling emerging Internet of All Things applications are three of the key innovation areas being driven by developers today with Hadoop. At Hortonworks, we are committed to supporting these use cases for enterprises through our Hortonworks Data Platform.

INSIDE THE HORTONWORKS DATA PLATFORM

The Hortonworks Data Platform consists of core Apache Hadoop plus many Apache ecosystem technologies like Hive, Spark, NiFi, Storm, Kafka, Falcon, Pig, Ranger, and more.

At Hortonworks, we believe in supporting a distribution of Hadoop that works with these technologies in a coordinated and structured way, so developers can feel confident that they will benefit from ongoing innovation in a way that is 100% open by design.

After a few weeks of experience with core Hadoop, developers tend to specialize in ELT (Extract Load Transform), Streaming, Data Science, Data Pipeline, Visualization, Online Application, Security, Data Management, etc., based on their job requirements and personal interest.

MOVING FORWARD WITH HADOOP

To help get you started with Hadoop, we have created a wide range of tutorials at hortonworks.com/tutorials.
Java developers should start with the tutorial "Introducing Apache Hadoop to Java Developers." If you prefer Python, head over to "Hands-on Tour of Apache Spark in 5 Minutes" or if Scala is your programming language of choice, dive in with "Interacting with Data on HDP using Scala and Apache Spark". SQL Devs can learn Hive with "How to Process Data with Hive".

All of these resources and more can be found at our one-stop shop for Hadoop developers at developer.hortonworks.com. Also check out the Hortonworks Gallery on GitHub for code, apps, and extensions for developers to use in their own projects.
Saptak Sen is a Developer Advocate with Hortonworks. You can always chat him up at @saptak on Twitter.

WRITTEN BY SAPTAK SEN
DEVELOPER ADVOCATE, **HORTONWORKS**

Hortonworks Data Platform

Apache Hadoop enables businesses to quickly gain insight from massive amounts of structured and unstructured data. The Hortonworks Data Platform (HDP) brings Apache Hadoop to enterprises developing predictive analytics apps, reducing the cost of data warehousing, and enabling innovation in the Internet of Anything.

CASE STUDY

Progressive Insurance is one of the largest U.S. auto insurance companies. The team turned to Hortonworks Data Platform to transform its business with massive ingestion of new types of data. Progressive uses HDP for ad placement and to store driving data for its usage-based insurance products that take sensor data from customer vehicles to customize rates based on customer behavior, creating a new category of offering for the company.

CLASSIFICATION

Data Management, Big Data Analytics, Data Integration

HOSTING

SaaS, PaaS, or On-Premise

FEATURES

- Built on Hadoop
- Native Stream Processing
- On-Demand Job Documentation
- MapReduce Job Designer

NOTABLE CUSTOMERS

- Microsoft
- T-Mobile
- eBay
- Bloomberg
- Cardinal Health
- TrueCar
- Symantec
- Spotify

BLOG **hortonworks.com/blog** | TWITTER **@hortonworks** | WEBSITE **hortonworks.com**

How Streaming Is Shaking Up the Data Analytics Landscape

BY JUSTIN LANGSETH

QUICK VIEW

01 Innovation in open source and at the network and I/O layers has broken down previous speed and performance barriers for visualization and analytics on big data sets.

02 Analytics are being pushed into the stream (via Spark), which is emerging as the de facto approach for sub-second query response times across billions of rows of data.

03 Working with data streams ensures the timely and accurate analysis that enables enterprises to harness the value of the data they work so hard to collect.

The rise of Apache Spark and the general shift from batch to real-time has disrupted the traditional data stack over the last two years. But one of the last hurdles to getting actual value out of Big Data is on the analytics side, where the speed to querying and visualizing Big Data (and the effectiveness of those visualizations translated into actual business value) is still a relatively young conversation, despite the fact that 87% of enterprises believe Big Data analytics will redefine the competitive landscape of their industries within the next three years [1].

Most engineers who are using legacy business intelligence tools are finding them woefully unprepared to handle the performance load of Big Data, while others who may be writing their own analytics with D3.js or similar tools are wrestling with the new backend challenges of fusing real-time data with other datastores.

Let's take a look at the megatrend toward streaming architectures, and how this is shaking up analytics requirements for developers.

DATA NATURALLY EXISTS IN STREAMS

All commerce, whether conducted online or in person, takes place as a stream of events and transactions. In the beginning, the stream was recorded in a book—an actual book that held inventories and sales, with each transaction penned in on its own line on the page. Over time, this practice evolved. Books yielded to computers and databases, but practical limitations still constrained data processing to local operations. Later on, data was packaged, written to disk, and shipped between locations for further processing and analysis. Grouping the data stream into batches made it easier to store and transport.

Technology marches on, and it has now evolved to the point that, in many cases, batching is no longer necessary. Systems are faster, networks are faster and more reliable, and programming languages and databases have evolved to accommodate a more distributed streaming architecture. For example, physical retail stores used to close for a day each quarter to conduct inventory. Then, they evolved to batch analysis of various locations on a weekly basis, and then a daily basis. Now, they keep a running inventory that is accurate through the most recent transaction. There are countless similar examples across every industry.

SO WHY ARE ANALYTICS AND VISUALIZATIONS STILL IN BATCH MODE?

Traditional, batch-oriented data warehouses pull data from multiple sources at regular periods, bringing it to a central location and assembling it for analysis. This practice causes data management and security headaches that grow larger over time as the number of data sources and the size of each batch grows. It takes a lot of time to export batches from the data source and import them into the data warehouse. In very large organizations, for which time is of the essence, batching can cause conflicts with backup operations. And the

process of batching, transporting, and analysis often takes so much time that it becomes impossible for a complex business to know what happened yesterday or even last week.

By contrast, with streaming-data analysis, organizations know they are working with the most recent—and timely—version of data because they stream the data on demand. By tapping into data sources only when they need the data, organizations eliminate the problems presented by storing and managing multiple versions of data. Data governance and security are simplified; working with streaming data means not having to track and secure multiple batches.

We live in an on-demand world. It's time to leave behind the model of the monolithic, complex, batch-oriented data warehouse and move toward a flexible architecture built for streaming-data analysis. Working with data streams ensures the timely and accurate analysis that enables enterprises to harness the value of the data they work so hard to collect, and tap into it to build competitive advantage.

BREAKING DOWN THE BARRIERS TO REAL-TIME DATA ANALYSIS

Previously, building streaming-data analysis environments was complex and costly. It took months or even years to deploy. It required expensive, dedicated infrastructure; it suffered from a lack of interoperability; it required specialized developers and data architects; and it failed to adapt to rapid changes in the database world, such as the rise of unstructured data.

In the past few years we have witnessed a flurry of activity in the streaming-data analysis space, both in terms of the development of new software and in the evolution of hardware and networking technology. Always-on, low-latency, high-bandwidth networks are less expensive and more reliable than ever before. Inexpensive and fast memory and storage allow for more efficient data analysis. In the past few years, we've witnessed the rise of many easy-to-use, inexpensive, and open-source streaming-data platform components. Apache Storm [2], a Hadoop-compatible add-on (developed by Twitter) for rapid data transformation, has been implemented by The Weather Channel, Spotify, WebMD, and Alibaba.com. Apache Spark [3], a fast and general engine for large-scale data processing, supports SQL, machine learning, and streaming-data analysis. Apache Kafka [4], an open-source message broker, is widely used for consumption of streaming data. And Amazon Kinesis [5], a fully managed, cloud-based service for real-time data processing over large, distributed data streams, can continuously capture large volumes of data from streaming sources.

CHECKLIST FOR DEVELOPERS BUILDING ANALYTICS AND VISUALIZATIONS ON TOP OF BIG DATA

The past few years have been witness to explosive growth in the number of streaming data sources and the volume of streaming data. It's no longer enough to look to historical data for business insight. Organizations require timely analysis of streaming data from such sources as the Internet of Things (IoT), social media, location, market feeds, news feeds, weather feeds, website clickstream analysis, and live transactional data. Examples of streaming-data analytics include telecommunications companies optimizing mobile networks on the fly using network device log and subscriber location data, hospitals decreasing the risk of nosocomial (hospital-originated) infections by capturing and analyzing real-time data from monitors on newborn babies, and office equipment vendors alerting service technicians to respond to impending equipment failures.

As the charge to streaming analytics continues and the focus becomes the "time to analytics gap" (how long it takes from arrival of data to business value being realized), I see three primary ways that developers should rethink how they embed analytics into their applications:

- **Simplicity of Use:** Analytics are evolving beyond the data scientist workbench, and must be accessible to broad business users. With streaming data, the visual interface is critical to make the data more accessible to a non-developer audience. From allowing them to join different data sources, to interacting with that data at the speed of thought—any developer bringing analytics into a business application is being forced to deal with the "consumerization of IT" trend such that business users get the same convenience layers and intuitiveness that any mobile or web application affords.
- **Speed / Performance First:** With visualization come requirements to bring the query results to the users in near real-time. Business users won't tolerate spinning pinwheels while the queries get resolved (as was the case with old approaches to running Big Data queries against JDBC connectors). Today we're seeing analytics pushed into the stream (via Spark), which is emerging as the de facto approach for sub-second query response times across billions of rows of data, and not having to move data before it's queried.
- **Data Fusion:** Embedded analytics capabilities must make multiple data sources appear as one. Businesses shouldn't have to distinguish between "Big Data" versus other forms of data. There's just data, period (including non-streaming and static data)—and it needs to be visualized, and available within business applications for sub-second, interactive consumption.

[1] forbes.com/sites/louiscolumbus/2014/10/19/84-of-enterprises-see-big-data-analytics-changing-their-industries-competitive-landscapes-in-the-next-year
[2] storm.apache.org
[3] spark.apache.org
[4] kafka.apache.org
[5] aws.amazon.com/kinesis

JUSTIN LANGSETH is an expert in Big Data, business intelligence, text analytics, sentiment analytics, and real-time data processing. He graduated from MIT with a degree in Management of Information Technology, and holds 14 patents. Zoomdata is Justin's 5th startup—he previously founded Strategy.com, Claraview, Clarabridge, and Augaroo. He is eagerly awaiting the singularity to increase his personal I/O rate which is currently frustratingly pegged at about 300 baud.

CHECKLIST

BIG DATA MANAGEMENT REQUIREMENTS

Big Data management is the genus to which Big Data processing and analytics are species. For developers, data scientists, and other data professionals, keeping every data management requirement in mind is not an easy task. This checklist will help you navigate critical requirements for consistently and flexibly curating raw sources of Big Data into trusted assets for next-generation analytics.

BIG DATA INTEGRATION

HIGH PERFORMANCE INGESTION	SCALABLE PROCESSING	FLEXIBLE DEPLOYMENT
☐ Purpose-built adapters for wide variety of data sources	☐ Parameterized pipeline development	☐ Deployment-agnostic pipeline abstraction
☐ Mass ingestion of high-volume and changed data	☐ Purpose-built components for data parsing and transformation	☐ Resource-optimized pipeline execution
☐ Real-time streaming for low-latency data	☐ End-to-end pipeline processing and data provisioning	☐ Deployment on cloud or on-premise

BIG DATA GOVERNANCE + QUALITY

COLLABORATIVE GOVERNANCE	360 VIEW OF RELATIONSHIPS	FIT-FOR-PURPOSE DATA CONFIDENCE
☐ Role-based data stewardship	☐ Universal metadata catalog	☐ Anomaly detection through data profiling
☐ Keyword and faceted data search	☐ Data matching and record linking at scale	☐ Data quality using purpose-built components
☐ Team-based data publishing and sharing	☐ Data relationship tracking and inference	☐ Data provenance through end-to-end lineage

BIG DATA SECURITY

360 VIEW OF SENSITIVE DATA	RISK ANALYTICS OF SENSITIVE DATA	POLICY-BASED PROTECTION OF SENSITIVE DATA
☐ Understanding of sensitive data protection and cost	☐ Risk modeling and score trending by residency, usage, and proliferation	☐ Non-intrusive de-identification of sensitive data
☐ Automated policy-based profiling and discovery	☐ Data proliferation tracking and monitoring	☐ Centralized security policy management
☐ Visibility of sensitive data by residency, usage, and proliferation	☐ Detection, analysis, and alerting of high risk users	☐ Support for on-premise and cloud

diving deeper
INTO BIG DATA

TOP 10 #BIGDATA TWITTER FEEDS

 @BIGDATAGAL

 @MEDRISCOLL

 @JAMESKOBIELUS

 @SPYCED

 @KDNUGGETS

 @KIRKDBORNE

 @MARCUSBORBA

 @DATA_NERD

 @HORTONWORKS

 @IBMBIGDATA

BIG DATA ZONES

Big Data
dzone.com/bigdata

The Big Data/Analytics Zone is a prime resource and community for Big Data professionals of all types. We're on top of all the best tips and news for Hadoop, R, and data visualization technologies. Not only that, but we also give you advice from data science experts on how to understand and present that data.

Database
dzone.com/database

The Database Zone is DZone's portal for following the news and trends of the database ecosystems, which include relational (SQL) and non-relational (NoSQL) solutions such as MySQL, PostgreSQL, SQL Server, NuoDB, Neo4j, MongoDB, CouchDB, Cassandra and many others.

Internet of Things
dzone.com/iot

The Internet of Things (IoT) Zone features all aspects of this multifaceted technology movement. Here you'll find information related to IoT, including Machine to Machine (M2M), real-time data, fog computing, haptics, open distributed computing, and other hot topics. The IoT Zone goes beyond home automation to include wearables, business-oriented technology, and more.

TOP BIG DATA REFCARDZ

Practical Data Mining with Python
Covers the tools used in practical Data Mining for finding and describing structural patterns in data using Python.

Machine Learning
Covers machine learning for predictive analytics, explains setting up training and testing data, and offers machine learning model snippets.

Getting Started with Apache Hadoop
Covers the most important concepts of Hadoop, describes its architecture, and explains how to start using it.

TOP BIG DATA WEBSITES

Dataversity.net
Daily updates of the latest in Big Data news and research, including cognitive computing, business intelligence, and NoSQL.

PlanetBigData.com
Content aggregator for Big Data blogs with frequent updates.

DataScienceCentral.com
A site by practitioners for practitioners that focuses on quality information over marketing ploys.

TOP BIG DATA TUTORIALS

R Introduction bit.ly/R-Intro
An in-depth introduction to the R language.

Introducing Apache Hadoop to Developers bit.ly/Hadoop-Dev
A developer-focused intro to Hadoop from Hortonworks.

Spark Stack: Getting Started
bit.ly/SparkStack
Get started with Spark with these overviews from the Spark Stack community

Solutions Directory

This directory contains two solution types: (1) Big Data platforms that provide analytics, data management, data visualization, business intelligence, and more; and (2) open-source frameworks for a variety of lower-level Big Data needs. It provides feature data and product category information gathered from vendor websites and project pages. Solutions are selected for inclusion based on several impartial criteria, including solution maturity, technical innovativeness, relevance, and data availability.

BIG DATA PLATFORMS

PRODUCT	CLASSIFICATION	REAL-TIME ANALYSIS	HADOOP SUPPORT	WEBSITE
1010Data Big Data Discovery	Big Data Analytics PaaS	No	No	1010data.com
Actian Analytics Platform	Big Data Analytics	Yes	Hadoop Integrations Available	actian.com
Alpine Chorus	Predictive Analytics	No	Yes	alpinenow.com/use-case-big-data-business
Alteryx Designer	Predictive Analytics	No	Yes	alteryx.com
Amazon Kinesis	Big Data Analytics PaaS	Yes	Hadoop Integrations Available	aws.amazon.com
Amazon Machine Learning	Predictive Analytics	No	No	aws.amazon.com
Appfluent by Attunity	Data Management, Business Intelligence	No	Hadoop Integrations Available	appfluent.com
Argyle Data	Big Data Analytics	Yes	Hadoop Integrations Available	argyledata.com
Azure Machine Learning by Microsoft	Predictive Analytics	No	Yes	azure.microsoft.com
BI Office by Pyramid Analytics	Big Data Analytics	No	Yes	pyramidanalytics.com
BigML	Machine Learning Platform	No	No	bigml.com
Birst	Big Data Analytics	Yes	Yes	birst.com
Bitam Artus	Data Managment	No	No	bitam.com
BOARD All in One	Business Intelligence	No	No	board.com
Cask Data App Platform	Data Management, Data Integration	No	Built on Hadoop	cask.co
Cloudera Enterprise	Data Management, Big Data Analytics	No	Built on Hadoop	cloudera.com
Databricks	Data Managment PaaS	No	Yes	databricks.com
DataDirect Connectors by Progress Software	Data Integration	No	Hadoop Integrations Available	progress.com
Datameer	Big Data Analytics, Data Visualization	No	Yes	datameer.com
DataTorrent RTS	Data Integration	Yes	Built on Hadoop	datatorrent.com
Datawatch Designer	Data Visualization	Yes	Hadoop Integrations Available	datawatch.com

BIG DATA PLATFORMS

PRODUCT	CLASSIFICATION	REAL-TIME ANALYSIS	HADOOP SUPPORT	WEBSITE
FICO Decision Management Suite	Predictive Analytics	Yes	Yes	fico.com
Guavus	Data Management, Big Data Analytics	Yes	Yes	guavus.com
HDInsight by Microsoft	Data Management PaaS	Yes	Built on Hadoop	azure.microsoft.com
Hortonworks Data Platform	Data Processing	Yes	Built on Hadoop	hortonworks.com
HP Haven	Data Integration, Big Data Analytics, PaaS	Yes	Hadoop Integrations Available	hp.com
IBM Watson	Cognitive Computing, Machine Learning	Yes	No	ibm.com
iHub by OpenText Analytics	Data Visualization	No	Yes	birt.actuate.com
Infobright Enterprise	Big Data Analytics	No	Hadoop Integrations Available	infobright.com
Informatica BDE	Data Management, Data Integration, Big Data Analytics	Yes	Hadoop Integrations Available	informatica.com
Insights Engine by GoodData	Data Visualization	No	Yes	gooddata.com
Jaspersoft by Tibco Software	Data Management, Business Intelligence	No	Yes	jaspersoft.com
Kapow by Kofax	Data Integration	No	Yes	kofax.com
Kognitio Analytical Platform	Big Data Analytics	Yes	Yes	kognitio.com
Logentries	Data Managment, Big Data Analytics	Yes	Yes	logentries.com
Logi Info	Big Data Analytics, Business Intelligence	No	Yes	logianalytics.com
MapR Distribution	Data Management	Yes	Built on Hadoop	mapr.com
Microsoft Power BI	Data Visualization, Data Integration	No	No	powerbi.microsoft.com
MicroStrategy	Data Managment	No	Yes	microstrategy.com
Necto by Panorama Software	Business Intelligence, Consulting	No	Hadoop Connector Available	panorama.com
New Relic Insights	Big Data Analytics	Yes	Yes	newrelic.com
Oracle Big Data Cloud Service	Data Management	No	Hadoop Connectors Available	oracle.com
Palantir Gotham	Data Management, Data Integration, Business Intelligence	No	Hadoop Integrations Available	palantir.com
ParStream	Big Data Analytics	Yes	No	parstream.com
Paxata	Big Data Analytics	No	Yes	paxata.com
Pentaho	Big Data Analytics	Yes	Yes	pentaho.com
Pivotal Big Data Suite	Data Management Platform	Yes	Yes	pivotal.io

BIG DATA PLATFORMS

PRODUCT	CLASSIFICATION	REAL-TIME ANALYSIS	HADOOP SUPPORT	WEBSITE
Platfora	Big Data Analytics	Yes	Yes	platfora.com
PredicSis	Predictive Analytics	Yes	No	predicsis.com
Prognoz Platform	Data Management	No	Yes	prognoz.com
Qlik Sense Enterprise	Big Data Analytics	No	Yes	qlik.com
RapidMiner Studio	Predictive Analytics	No	Yes	rapidminer.com
RedPoint Data Management	Data Management	No	Hadoop Integrations Available	redpoint.net
RJmetrics	Big Data Analytics	No	Yes	rjmetrics.com
SAP HANA	Data Management	Yes	Yes	sap.com
SAS Intelligence Platform	Data Management	Yes	Built on Hadoop	sas.com
SiSense	Big Data Analytics	No	Yes	sisense.com
Skytree Infinity	Predictive Analytics	No	Yes	skytree.net
SpaceCurve	Data Management, Data Visualization	Yes	Hadoop Connector Available	spacecurve.com
Splunk Enterprise	Big Data Analytics, Business Intelligence	Yes	Yes	splunk.com
Spring XD	Data Management	Yes	Yes	spring.io
Sumo Logic	Data Management	Yes	Hadoop Integrations Available	sumologic.com
Tableau Desktop	Big Data Analytics	Yes	Hadoop Integrations Available	tableausoftware.com
Talend Big Data Integration	Data Integration	No	Built on Hadoop	talend.com
Targit	Data Management	Yes	Yes	targit.com
Terracotta In-Memory Data Management by Software AG	Data Management	No	Hadoop Connector Available	terracotta.org
ThingSpan by Objectivity	Data Management, Data Integration	No	Built on Hadoop	objectivity.com
Think Big by Teradata	Data Management	No	Yes	thinkbig.teradata.com
TIBCO Spotfire	Big Data Analytics	Yes	Hadoop Integrations Available	spotfire.tibco.com
Tidemark	Big Data Analytics	Yes	Built on Hadoop	tidemark.com
Treasure Data	Data Integration, Big Data Analytics	No	Hadoop Integrations Available	treasuredata.com
Trifacta Platform	Data Management, Business Intelligence	No	Hadoop Integrations Available	trifacta.com
Turbine by Engineroom.io	Data Management PaaS	No	Hadoop Integrations Available	engineroom.io

BIG DATA PLATFORMS

PRODUCT	CLASSIFICATION	REAL-TIME ANALYSIS	HADOOP SUPPORT	WEBSITE
webFOCUS by Information Builders	Data Visualization, Business Intelligence	No	Yes	informationbuilders.com
Yellowfin	Business Intelligence, Data Visualization	No	No	yellowfinbi.com
Zoomdata	Data Integration, Data Visualization	Yes	Yes	zoomdata.com

FRAMEWORKS

PRODUCT	CLASSIFICATION	WEBSITE
Ambari	System Metrics Framework	ambari.apache.org
Crunch	MapReduce Library	crunch.apache.org
Disco	Distributed Computing	discoproject.org
Drill	Query Engine	drill.apache.org
Falcon	Feed Processing	falcon.apache.org
Flink	Computational Framework	flink.apache.org
Flume	Data Integration Framework	flume.apache.org
Giraph	Graph Processing	giraph.apache.org
GraphX	Graph Processing	spark.apache.org/graphx
Hadoop	Data Processing	hadoop.apache.org
Hama	Analytics Framework	hama.apache.org
Hive	Data Warehouse Framework	hive.apache.org
Kafka	Messaging System	kafka.apache.org
Mesos	Resource Manager	mesos.apache.org
Misco	MapReduce Framework	alumni.cs.ucr.edu/~jdou/misco
Oozie	Workflow Scheduler	oozie.apache.org
OpenTSDB	System Metrics Framework	opentsdb.net
Pig	Analytics Programming Language	pig.apache.org
Samza	Stream Processing	samza.apache.org
Spark	Computational Framework	spark.apache.org
Sqoop	Data Integration Framework	sqoop.apache.org
Storm	Stream Processing	storm.apache.org
Tez	Interactive Data Processing	tez.apache.org
YARN	Resource Manager	hadoop.apache.org
Zookeeper	Coordination and State Management	zookeeper.apache.org

diving deeper

INTO FEATURED BIG DATA SOLUTIONS

Looking for more information on individual Big Data solutions providers?
Nine of our partners have shared additional details about their offerings, and we've summarized this data below.

If you'd like to share data about these or other related solutions, please email us at research@dzone.com.

DATA PROCESSING

Hortonworks Data Platform

FEATURES
- High Availability
- Load Balancing
- Automatic Failover

HOSTING
SaaS or On-Premise

DATA INTEGRATION
ETL and ELT

DATABASE INTEGRATIONS
- MySQL
- HBase
- Oracle
- Teradata
- VoltDB
- MongoDB
- Couchbase

VISUALIZATION

iHub by OpenText Analytics

FEATURES
- JDBC Connectors
- Hadoop Connectors
- Auto-scaling Feature

HOSTING
SaaS or On-Premise

DATA INTEGRATION
n/a

DATABASE INTEGRATIONS
- PostgreSQL
- SQL Server
- Oracle
- IBM DB2

DATA INTEGRATION

Informatica Intelligent Data Platform

FEATURES
- High Availability
- Load Balancing
- Automatic Failover

HOSTING
SaaS or On-Premise

DATA INTEGRATION
ETL and ELT

DATABASE INTEGRATIONS
- MySQL
- Oracle
- MongoDB
- Cassandra
- PostgreSQL
- IBM DB2

REPORTING & DASHBOARDS

JReport Server Live by Jinfonet

FEATURES
- High Availability
- Load Balancing
- Automatic Failover

HOSTING
SaaS

DATA INTEGRATION
ETL

DATABASE INTEGRATIONS
OLAP databases that support JDBC or ODBC

CONSULTING

Mammoth Data

TECHNOLOGIES
- Hadoop
- Cassandra
- Couchbase
- MongoDB
- Neo4j

VERTICALS
- Biotech
- Education
- Finance
- Healthcare
- Retail
- Technology
- Utilities

PARTNERS
- Cloudera
- CloudBees
- Couchbase
- Datastax
- Hortonworks
- MongoDB
- Neo Technology

DATA PROCESSING

MapR Distribution

FEATURES
- High Availability
- Load Balancing
- Integrated Database

HOSTING
SaaS or On-Premise

DATA INTEGRATION
ETL

DATABASE INTEGRATIONS
- MapR-DB
- HBase
- MySQL

BIG DATA ANALYTICS

New Relic Insights

FEATURES
- High Availability
- Load Balancing
- Automatic Failover

HOSTING
SaaS

DATA INTEGRATION
n/a

DATABASE INTEGRATIONS
- n/a

ODBMS + HADOOP & SPARK

Thingspan by Objectivity

FEATURES
- High Availability
- Load Balancing
- Monitoring Solution Included

HOSTING
On-Premise

DATA INTEGRATION
ETL and ELT

DATABASE INTEGRATIONS
- None

VISUALIZATION

Zoomdata

FEATURES
- High Availability
- Automatic Failover
- Visualization Solution Included

HOSTING
SaaS or On-Premise

DATA INTEGRATION
ELT

DATABASE INTEGRATIONS
- MongoDB
- Impala
- PostgreSQL
- MySQL
- SQL Server
- Oracle
- Spark SQL

glossary

BATCH PROCESSING Doing work in chunks. More formally: the execution of a series of programs (jobs) that process sets of records as complete units (batches). Commonly used for processing large sets of data offline for fast analysis later.

BIG DATA Data whose size makes people use it differently. Describes the entire process of discovering, collecting, managing, and analyzing datasets too massive and/or unstructured to be handled efficiently by traditional database tools and methods.

BUSINESS INTELLIGENCE (BI) The use of tools and systems for the identification and analysis of business data to provide historical and predictive insights.

COMPLEX EVENT PROCESSING Treating events as sets of data points from multiple sources. For example: three simple events 'stubbed toe,' 'fell forward' & 'crushed nose' might constitute the single complex event 'injured by falling.' Useful concept to transition from 'dumb' input streams to 'smart' business analysis.

DATA ANALYTICS The process of harvesting, managing, and analyzing large sets of data to identify patterns and insights. Exploratory, confirmatory, and qualitative data analytics require different application-level feature sets.

DATA MANAGEMENT The complete lifecycle of how an organization handles storing, processing, and analyzing datasets.

DATA MINING The process of discovering patterns in large sets of data and transforming that information into an understandable format. Often involves deeper scientific and technical work than analytics.

DATA MUNGING/WRANGLING The process of converting raw mapping data into other formats using automated tools to create visualizations, aggregations, and models. Often addresses data that straight-up ETL discards.

DATA SCIENCE The field of study broadly related to the collection, management, and analysis of raw data by various means of tools, methods, and technologies.

DATA WAREHOUSE A collection of accumulated data from multiple streams within a business, aggregated for the purpose of business management—where ready-to-use data is stored.

DATABASE CLUSTERING Making the same dataset available on multiple nodes (physical or logical), often for the advantages of fault tolerance, load balancing, and parallel processing.

DISTRIBUTED SYSTEM A set of networked computers solving a single problem together. Coordination is accomplished via messages rather than shared memory. Concurrency is managed by software that functions like an OS over a set of individual machines.

EVENTUAL CONSISTENCY The idea that databases will contain data that becomes consistent over time.

EXTRACT TRANSFORM LOAD (ETL) Taking data from one source, changing it into a more useful form, and relocating it to a data warehouse, where it can be useful. Often involves extraction from heterogeneous sources and transformation functions that result in similarly structured endpoint datasets.

HADOOP An Apache Software Foundation distributed framework developed specifically for high-scalability, data-intensive, distributed computing. The most popular implementation of MapReduce.

HADOOP DISTRIBUTED FILE SYSTEM (HDFS) A distributed file system created by Apache Hadoop to handle data throughput and access from the MapReduce algorithm.

HIGH AVAILABILITY Ability of a system to keep functioning despite component failure.

MAPREDUCE A programming model created by Google for high scalability and distribution on multiple clusters for the purpose of data processing.

MESOS An open-source cluster manager developed by the Apache Software Foundation. Abstracts basic compute resources and uses a two-level scheduling mechanism. Like a kernel for distributed systems.

MYRIAD An Apache project used to integrate YARN with Mesos.

ONLINE ANALYTICAL PROCESSING (OLAP) A concept that refers to tools which aid in the processing of complex queries, often for the purpose of data mining.

ONLINE TRANSACTION PROCESSING (OLTP) A type of system that supports the efficient processing of large numbers of database transactions; used heavily for business client services.

PREDICTIVE ANALYTICS The determination of patterns and possible outcomes using existing datasets.

R (LANGUAGE) An interpreted language used in statistical computing and visualization.

REPLICATION Storing multiple copies of the same data so as to ensure consistency, availability, and fault-tolerance. Data is typically stored on multiple storage devices.

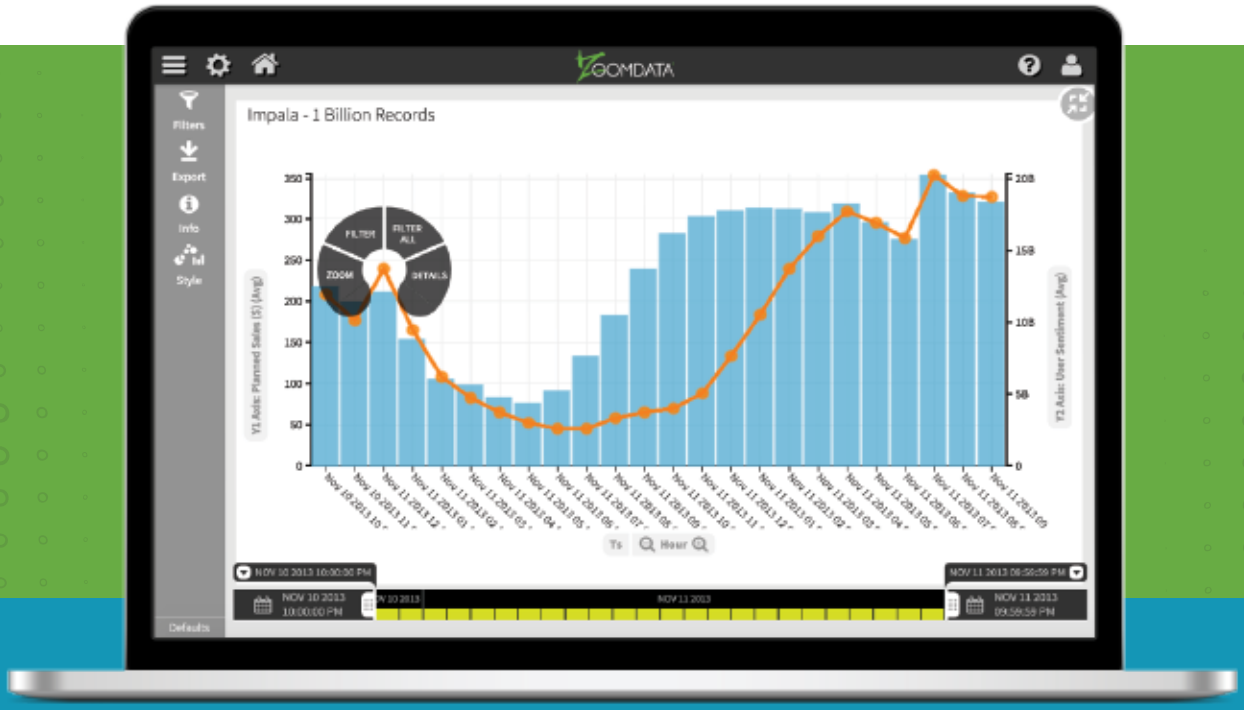
SCALABILITY Ability of a system to accommodate increasing load.

SPARK An in-memory open-source cluster computing framework by the Apache Software Foundation. Able to load data into a cluster's memory and query it numerous times.

STREAMING DATA Data that becomes available continuously rather than discretely.

YARN (YET ANOTHER RESOURCE NEGOTIATOR) Also called MapReduce 2.0. Built for Hadoop by the Apache Software Foundation for cluster resource management. Decouples job tracking from resource management.

ZOOMDATA®



Zoomdata 2.0 allows business users to visually consume and interact with all the data in the modern enterprise

- Zoomdata Fusion enables interactive analysis across disparate data sources, bridging modern and legacy data architectures, blending real-time streams and historical data, and unifying enterprise data on-premises and in the cloud.
- New Smart Connectors to Apache HBase via Apache Phoenix, MemSQL and popular cloud applications like Salesforce, Google Analytics, Marketo, Zendesk and SendGrid.
- Easily Activate Zoomdata with one-click deployments on the broadest range of cloud platforms, and bring analytics to your data, wherever it is.

Try a Free 60 Day Trial In The Cloud or On-Premise

http://bit.ly/dzone2k15