DZone
RESEARCH

DZONE RESEARCH PRESENTS

2014 GUIDE TO BIG DATA

BROUGHT TO YOU IN PARTNERSHIP WITH

MAPR Typesafe

GOGRID New Relic.

WELCOME

Dear Reader,

Welcome to our fifth DZone Research Guide and welcome to The Age of Big Data. It's fitting that this guide follows our Internet of Things guide, as by all accounts, IoT is driving the creation of more data than ever before. In the blink of an eye we can fill more storage, in a smaller form factor, than the largest hard drives of 15 years ago. Through the never-ending march of technology and the broad availability of cloud, available storage and computing power is now effectively limitless. The combination of these technologies gives developers and analysts such as ourselves a wealth of new possibilities to draw conclusions from our data and make better business decisions.

Just as our fourth guide was focused around the platforms and devices that are driving this amazing creation of new data, this guide is focused around the tools that developers and architects will use to gather and analyze data more effectively. We've covered a wide spectrum of the tools: from NoSQL databases like MongoDB and Hadoop to business intelligence (BI) tools like Actuate BIRT, down to traditional relational databases like Oracle, MySQL, and PostgreSQL. Gathering the data is easy, it's what you do with it after the fact that makes it interesting.

As you'll find while you read through our findings from nearly 1,000 developers, architects, and executives, Big Data is no longer a passing fad or something that people are just beginning to explore. Nearly 89% of all respondents told us that they are either exploring a Big Data implementation or have already rolled out at least one project. This is amazing growth for an industry that barely existed even 5 years ago. So, welcome to the DZone Big Data Guide, and we hope you enjoy the data and the resources that we've collected.

MATT SCHMIDT
CTO, PRESIDENT
research@dzone.com

TABLE OF CONTENTS

SUMMARY & KEY TAKEAWAYS 3

KEY RESEARCH FINDINGS 4

THE NO FLUFF INTRODUCTION TO BIG DATA
BY BENJAMIN BALL 6

THE EVOLUTION OF MAPREDUCE AND HADOOP
BY SRINATH PERERA & ADAM DIAZ 10

THE DEVELOPER'S GUIDE TO DATA SCIENCE
BY SANDER MAK 14

THE DIY BIG DATA CLUSTER
BY CHANWIT KAEWKASI 20

FINDING THE DATABASE FOR YOUR USE CASE 22

BIG DATA SOLUTIONS DIRECTORY 25

GLOSSARY 35

CREDITS

DZONE RESEARCH

JAYASHREE GOPALAKRISHNAN
DIRECTOR OF RESEARCH

MITCH PRONSCHINSKE
SENIOR RESEARCH ANALYST

BENJAMIN BALL
RESEARCH ANALYST, AUTHOR

MATT WERNER
MARKET RESEARCHER

JOHN ESPOSITO
REFCARDZ EDITOR/COORDINATOR

ALEC NOLLER
SENIOR CONTENT CURATOR

MIKE LENTO
GRAPHIC DESIGNER

Special thanks to our topic experts David Rosenthal, Oren Eini, Otis Gospodnetic, Daniel Bryant, Marvin Froeder, Isaac Sacolick, Arnon Rotem-Gal-Oz, Kate Borger, and our trusted DZone Most Valuable Bloggers for all their help and feedback in making this report a great success.

DZONE MARKETING AND SALES

ALEX CRAFTS
SENIOR ACCOUNT MANAGER

KELLET ATKINSON
DIRECTOR OF MARKETING

ASHLEY SLATE
DIRECTOR OF DESIGN

CHRIS SMITH
PRODUCTION ADVISOR

DZONE CORPORATE

RICK ROSS
CEO

MATT SCHMIDT
CTO, PRESIDENT

BRANDON NOKES
VP OF OPERATIONS

HERNÂNI CERQUEIRA
LEAD SOFTWARE ENGINEER

SUMMARY & KEY TAKEAWAYS

Big Data, NoSQL, and NewSQL—these are the high-level concepts relating to the new, unprecedented data management and analysis challenges that enterprises and startups are now facing. Some estimates expect the amount of digital data in the world to double every two years [1], while other estimates suggest that 90% of the world's current data was created in the last two years [2]. The predictions for data growth are staggering no matter where you look, but what does that mean practically for you, the developer, the sysadmin, the product manager, or C-level leader? DZone's *2014 Guide to Big Data* is the definitive resource for learning how industry experts are handling the massive growth and diversity of data. It contains resources that will help you navigate and excel in the world of Big Data management. These resources include:

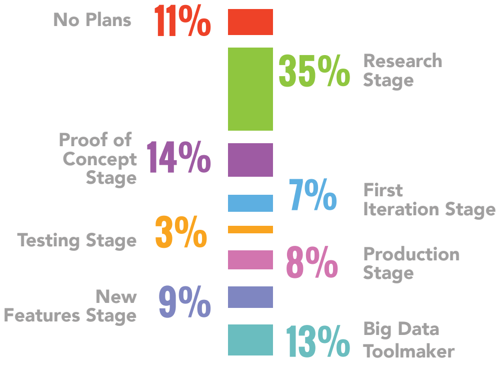
- Side-by-side feature comparison of the best analytics tools, databases, and data processing platforms (selected based on several criteria including solution maturity, technical innovativeness, relevance, and data availability).
- Comprehensive data sourced from 850+ IT professionals on data management tools, strategies, and experiences.
- A database selection tool for discovering the strong and weak use cases of each database type.
- A guide to the emerging field of data science.
- Forecasts of the future challenges in processing and creating business value from Big Data.

KEY TAKEAWAYS

BIG DATA PLANS ARE UNDERWAY FOR MOST ORGANIZATIONS

Big Data analysis is a key project for many organizations, with only 11% of survey respondents saying they have no plans to add large-scale data gathering and analysis to their systems. A large portion, however, are only in the exploratory stages with new data management technologies (35%). 24% are currently building or testing their first solution, and 30% have either deployed a solution or are improving a solution that they've already built. As you'll discover in this guide, this final group is finding plenty of new data correlations, but their biggest challenge is finding the causes behind these observations.

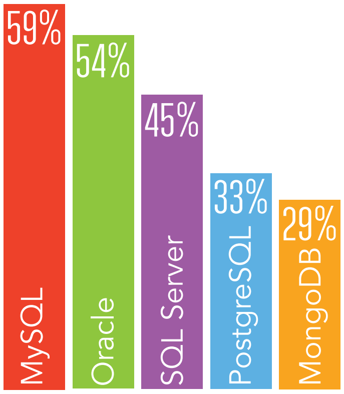
BIG DATA PROJECT STATUS



HADOOP IS NOW PREVALENT IN MODERN IT

Apache Hadoop isn't the sole focus of Big Data management, but it certainly opened the door for more cost-effective batch processing. Today the project has a comprehensive arsenal of data processing tools and compatible projects. 53% of respondents have used Hadoop and 35% of respondents' organizations use Hadoop, so even though many organizations still don't use Hadoop, a majority of developers have taken the initiative to familiarize themselves with this influential tool. Looking forward, YARN and Apache Spark are likely to become highly influential projects as well.

WHICH DATABASES DOES YOUR ORGANIZATION USE?



RDBMS STILL DOMINATES THE BROADER IT INDUSTRY

Relational database mainstays including MySQL and Oracle are being used at 59% and 54% of respondents' organizations respectively. SQL Server (45%) and PostgreSQL (33%) are also popular choices. Even though NoSQL databases gained a rabid following from developers who were fed up with certain aspects of RDBMS, today relational data stores are still a dominant part of the IT industry, and the strengths of SQL databases have been reiterated by many key players in the space. A multi-database solution, or polyglot persistence approach, is a popular pattern among today's experts. NoSQL databases are certainly making inroads at companies besides the high-profile web companies that created them. Currently, MongoDB is the most popular NoSQL database with 29% of respondents' organizations currently using it. For a look at the strong and weak use cases of each database type, see the *Finding the Database For Your Use Case* section of this guide.

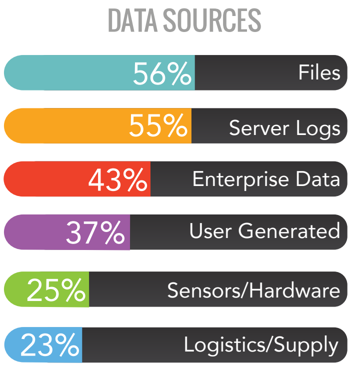
[1] http://www.emc.com/about/news/press/2014/20140409-01.htm

[2] http://www.sciencedaily.com/releases/2013/05/130522085217.htm

KEY RESEARCH FINDINGS

More than 850 IT professionals responded to DZone's 2014 Big Data Survey. Here are the demographics for this survey:

- Developers (43%) and development team leads (26%) were the most common roles.
- 60% of respondents come from large organizations (100 or more employees) and 40% come from small organizations (under 100 employees).
- The majority of respondents are headquartered in the US (35%) or Europe (38%).
- Over half of the respondents (63%) have over 10 years of experience as IT professionals.
- A large majority of respondents' organizations use Java (86%). Python is the next highest (37%).

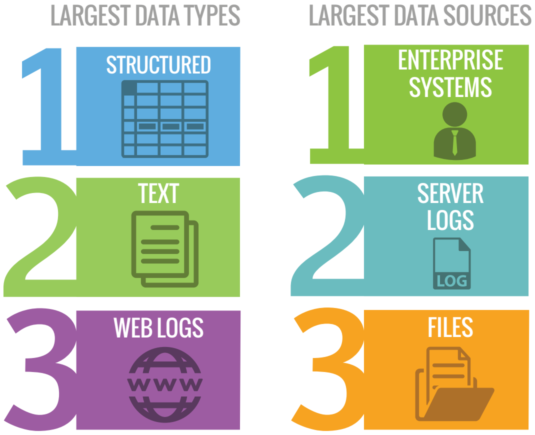


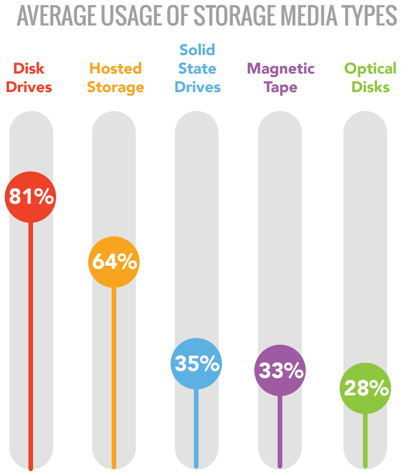
FILES AND LOGS ARE THE MOST COMMON DATA SOURCES

The first step to getting a handle on your data is understanding the sources and types of data that your systems record and generate. In the three Vs of Big Data, sources and types of data represent your data variety. After asking the respondents about the data sources they deal with, we found that files such as documents and media are the most common (56%), while server logs from applications are close behind (55%). Even though the Big Data explosion is being driven mainly by user generated data (37%), most of the companies in this survey aren't dealing with it as much as files, server logs, and enterprise system data such as ERP and CRM entries. Much of the exploding user generated data is in the hands of the high-profile web companies like Facebook, Google, and Twitter.

UNSTRUCTURED DATA IS MORE COMMON IN HADOOP AND NoSQL ORGS

Next, users were asked about the types of data they analyze. Structured data (76%) and semi-structured data (53%) are the two most common types of data. Among unstructured data types, event messaging data (34%), text (38%), and clickstreams (32%) are the most common. For most organizations, their largest set of data being analyzed is structured data (39%). However, when filtering just for organizations that have a high number of data processing nodes or that use NoSQL, unstructured data types and user generated data sources are more common than the overall survey pool.





CLOUD STORAGE CUSTOMERS TAKE FULL ADVANTAGE

Another important aspect of Big Data is volume. Sometimes the easiest way to deal with data volume is more hardware. The most common storage medium for respondents is disk-drives (84%), which is not surprising; however, solid-state drives (34%) and third-party hosted storage (22%) also have significant traction. The average disk-drive-using respondent uses it for 81% of their storage needs. If a respondent uses hosted storage, they often make heavy usage of it, averaging about 64% of their storage.

ALMOST ALL ORGS EXPECT THEIR STORAGE NEEDS TO GROW EXPONENTIALLY

Just how much volume do today's IT organizations have? The majority of respondents surveyed don't have more than 9 TBs in their entire organization, but in just the next year, only 10% think they will have less than 1 TB. The bulk of organizations expect to have between 1 and 50 TBs. Since one year predictions are often more accurate than 2-5 year predictions, this is strong evidence that most organizations will experience exponential data growth. The past trends of the IT industry also provide clear evidence that exponential data growth is common and a constant factor in technology.

ESTIMATED ORGANIZATION DATA STORAGE AND USAGE

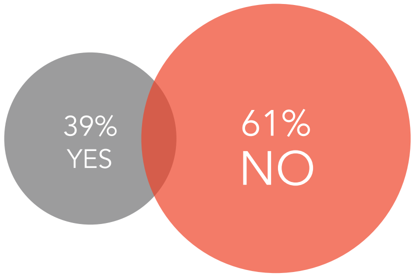
<1 TB	1-9 TB	10-49 TB	50-99 TB	100-499 TB	500 TB - 1 PB	> 1 PB
24%	29%	16%	9%	11%	6%	5%
10%	25%	20%	13%	17%	8%	7%

● CURRENTLY ● IN THE NEXT YEAR

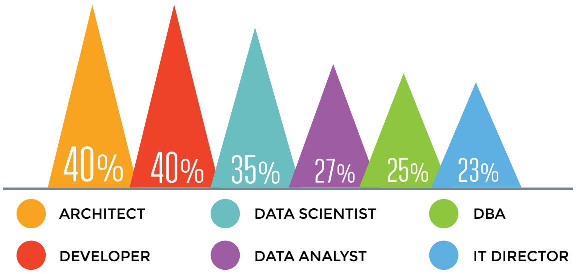
HADOOP USAGE IS HIGH DESPITE THE LEARNING CURVE

53% of respondents have used Apache Hadoop. However, only 29% have used it at their work, while 38% have used it for personal projects. This indicates that the respondents are preparing themselves for the tool's increasing ubiquity in modern data processing. The top three uses for Hadoop among respondents were statistics and pattern analysis (63%), data transformation and preparation (60%), and reporting/BI (53%). When asked about their experience with Hadoop, 39% said it was difficult to use. This is reflected in another question that asked about the most challenging aspects of Hadoop. The three biggest are the learning curve (68%), development effort (59%), and hiring experienced developers (44%). Finally, users were asked about the Hadoop distribution they use. The most commonly used distribution is the basic Apache distribution (48%), but close behind is Cloudera (40%).

IS HADOOP DIFFICULT TO USE?



WHICH ROLES MANAGE ANALYTICS IN YOUR ORGANIZATION?



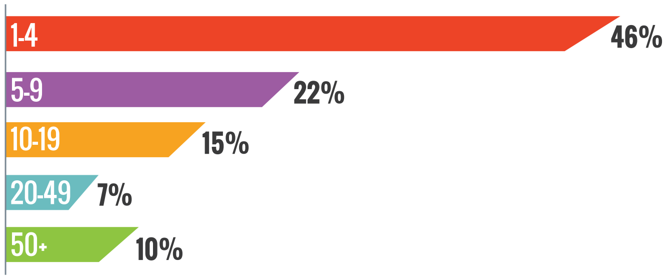
BIG DATA IS VERY MUCH IN THE HANDS OF THE DEVELOPERS

The three departments in respondents' organizations that are most commonly responsible for providing a data analytics environment are operations, research or analysis, and the application group. When it comes to developing and managing data analytics, application architects and developers are called upon most often in the surveyed organizations (40%). Data scientists were also common stewards of data analytics with 35% of organizations utilizing them. The least likely manager of data analytics was the CIO or similar executive (8%).

TEAMS RUNNING HADOOP AND COLUMN STORES TEND TO HAVE BIGGER ANALYTICS CLUSTERS

Almost half of organizations (46%) have just one to four data processing nodes in a data processing cluster, but there is a wide distribution of respondents using various node amounts. Companies that gather sensor data, user data, or logistics/supply chain data tend to have higher node counts among respondents, which means they are probably more data-heavy sources. Organizations with audio/video data or scientific data are more likely than other segments to have over four data nodes. Also, companies that use Hadoop often had more data processing nodes than non-Hadoop companies. This is where it becomes clear how much Hadoop helps with handling multi-node data processing clusters. Teams with data scientist titles and teams running HBase or Cassandra also tend to have more nodes.

HOW MANY NODES ARE TYPICALLY IN YOUR DATA PROCESSING CLUSTERS?



THE NO FLUFF INTRODUCTION TO BIG DATA

by Benjamin Ball

Big Data traditionally has referred to a collection of data too massive to be handled efficiently by traditional database tools and methods. This original definition has expanded over the years to identify tools (Big Data tools) that tackle extremely large datasets (NoSQL databases, MapReduce, Hadoop, NewSQL, etc.), and to describe the industry challenge posed by having data harvesting abilities that far outstrip the ability to process, interpret, and act on that data. Technologists knew that those huge batches of user data and other data types were full of insights that could be extracted by analyzing the data in large aggregates. They just didn't have any cheap, simple technology for organizing and querying these large batches of raw, unstructured data.

The term quickly became a buzzword for every sort of data processing product's marketing team. Big Data became a catchall term for anything that handled non-trivial sizes of data. Sean Owen, a data scientist at Cloudera, has suggested that Big Data is a stage where individual data points are irrelevant and only aggregate analysis matters [1]. But this is true for a 400 person survey as well, and most people wouldn't consider that very big. The key part missing from that definition is the transformation of unstructured data batches into structured datasets. It doesn't matter if the database is relational or non-relational. Big Data is not defined by a number of terabytes: it's rooted in the push to discover hidden insights in data that companies used to disregard or throw away.

> *Big Data became a catchall term for anything that handled non-trivial sizes of data.*

Due to the obstacles presented by large scale data management, the goal for developers and data scientists is two-fold: first, systems must be created to handle large-scale data, and second, business intelligence and insights should be acquired from analysis of the data. Acquiring the tools and methods to meet these goals is a major focus in the data science industry, but it's a landscape where needs and goals are still shifting.

WHAT ARE THE CHARACTERISTICS OF BIG DATA?

Tech companies are constantly amassing data from a variety of digital sources that is almost without end—everything from email addresses to digital images, MP3s, social media communication, server traffic logs, purchase history, and demographics. And it's not just the data itself, but data about the data (metadata). It is a barrage of information on every level. What is it that makes this mountain of data Big Data?

One of the most helpful models for understanding the nature of Big Data is "the three Vs": volume, velocity, and variety.

Data Volume

Volume is the sheer size of the data being collected. There was a point in not-so-distant history where managing gigabytes of data was considered a serious task—now we have web giants like Google and Facebook handling petabytes of information about users' digital activities. The size of the data is often seen as the first challenge of characterizing Big Data storage, but even beyond that is the capability of programs to provide architecture that can not only store but query these massive datasets. One of the most popular models for Big Data architecture comes from Google's MapReduce concept, which was the basis for Apache Hadoop, a popular data management solution.

Data Velocity

Velocity is a problem that flows naturally from the volume characteristics of Big Data. Data velocity is the speed at which data is flowing into a business's infrastructure and the ability of software solutions to receive and ingest that data quickly. Certain types of high-velocity data, such as streaming data, needs to be moved into storage and processed on the fly. This is often referred to as complex event processing (CEP). The ability to intercept and analyze data that has a lifespan of milliseconds is widely sought after. This kind of quick-fire data processing has long been the cornerstone of digital financial transactions, but it is also used to track live consumer behavior or to bring instant updates to social media feeds.

Data Variety

Variety refers to the source and type of data collected. This data could be anything from raw image data to sensor readings, audio recordings, social media communication, and metadata. The challenge of data variety is being able to take raw, unstructured data and organize it so that an application can use it. This kind of structure can be achieved through architectural models that traditionally favor relational databases—but there is often a need to tidy up this data before it will even be useful to store in a refined form. Sometimes a better option is to use a schemaless, non-relational database.

HOW DO YOU MANAGE BIG DATA?

The Three Vs is a great model for getting an initial understanding of what makes Big Data a challenge for businesses. However, Big Data is not just about the data itself, but the way that it is handled. A popular way of thinking about these challenges is to look at how a business stores, processes, and accesses their data.

- **Store:** Can you store the vast amounts of data being collected?
- **Process:** Can you organize, clean, and analyze the data collected?
- **Access:** Can you search and query this data in an organized manner?

The Store, Process, and Access model is useful for two reasons: it reminds businesses that Big Data is largely about managing data, and it demonstrates the problem of scale within Big Data management. "Big" is relative. The data batches that challenge some companies could be moved through a single Google datacenter in under a minute. The only question a company needs to ask itself is how it will store and access increasingly massive amounts of data for its particular use case. There are several high level approaches that companies have turned to in the last few years.

THE TRADITIONAL APPROACH

The traditional method for handling most data is to use relational databases. Data warehouses are then used to integrate and analyze data from many sources. These databases are structured according to the concept of "early structure binding"—essentially, the database has predetermined "questions" that can be asked based on a schema. Relational databases are highly functional, and the goal with this type of data processing is for the database to be fully transactional. Although relational databases are the most common persistence type by a large margin (see *Key Findings* pg. 4-5), a growing number of use cases are not well-suited for relational schemas. Relational architectures tend to have difficulty when dealing with the velocity and variety of Big Data, since their structure is very rigid. When you perform functions such as JOIN on many complex data sets, the volume can be a problem as well. Instead, businesses are looking to non-relational databases, or a mixture of both types, to meet data demand.

THE NEWER APPROACH - MAPREDUCE, HADOOP, AND NoSQL DATABASES

In the early 2000s, web giant Google released two helpful web technologies: Google File System (GFS) and MapReduce. Both were new and unique approaches to the growing problem of Big Data, but MapReduce was chief among them, especially when it comes to its role as a major influencer of later solution models. MapReduce is a programming paradigm that allows for low cost data analysis and clustered scale-out processing.

MapReduce became the primary architectural influence for the next big thing in Big Data: the creation of the Big Data management infrastructure known as Hadoop. Hadoop's open source ecosystem and ease of use for handling large-scale data processing operations have secured a large part of the Big Data marketplace.

Besides Hadoop, there was a host of non-relational (NoSQL) databases that emerged around 2009 to meet a different set of demands for processing Big Data. Whereas Hadoop is used for its massive scalability and parallel processing, NoSQL databases are especially useful for handling data stored within large multi-structured datasets. This kind of discrete data handling is not traditionally seen as a strong point of relational databases, but it's also not the same kind of data operations that Hadoop runs. The solution for many businesses ends up being a combination of these approaches to data management.

FINDING HIDDEN DATA INSIGHTS

Once you get beyond storage and management, you still have the enormous task of creating actionable business intelligence (BI) from the datasets you've collected. This problem of processing and analyzing data is maybe one of the trickiest in the data management lifecycle. The best options for data analytics will favor an approach that is predictive and adaptable to changing data streams. The thing is, there are so many types of analytic models, and different ways of providing infrastructure for this process. Your analytics solution should scale, but to what degree? Scalability can be an enormous pain in your analytical neck, due to the problem of decreasing performance returns when scaling out many algorithms.

> *Data insight means nothing for a business if they can't then create actionable intelligence.*

Ultimately, analytics tools rely on a great deal of reasoning and analysis to extract data patterns and data insights, but this capacity means nothing for a business if they can't then create actionable intelligence. Part of this problem is that many businesses have the infrastructure to accommodate Big Data, but they aren't asking questions about what problems they're going to solve with the data. Implementing a Big Data-ready infrastructure before knowing what questions you want to ask is like putting the cart before the horse.

But even if we do know the questions we want to ask, data analysis can always reveal many correlations with no clear causes. As organizations get better at processing and analyzing Big Data, the next major hurdle will be pinpointing the causes behind the trends by asking the right questions and embracing the complexity of our answers.

[1] http://www.quora.com/What-is-big-data

WRITTEN BY
Benjamin Ball
Benjamin Ball is a Research Analyst and Content Curator at DZone. When he's not producing technical content or tweeting about someone else's (**@bendzone**), he is an avid reader, writer, and gadget collector.

Being data-driven has never been easier

New Relic Insights™ is a real-time analytics platform that collects and analyzes billions of events directly from your software to provide instant, actionable intelligence about your applications, customers, and business.

Start making better decisions today
www.newrelic.com/insights

Quizlet

"We've come to rely on New Relic Insights so much that we've basically stopped using all of our other analytics tools."

Andrew Sutherland
Founder and CTO, Quizlet

New Relic® INSIGHTS™

TACKLING BIG DATA CHALLENGES WITH SOFTWARE ANALYTICS

Companies today are amassing a tidal wave of data created from their production software, websites, mobile applications, and back-end systems. They recognize the value that lies in this big data, yet many struggle to make data-driven decisions. Businesses need answers to their data questions now—not in days or weeks.

While there are various analytics tools out there that can help address such challenges—anything from web analytics, business intelligence tools, log search, NoSQL, and Hadoop—each falls short in some capacity. When evaluating solutions for big data analysis, consider the following criteria:

- **Real-time data collection and analysis.** Does the solution allow you to collect and ingest data in real time or is there a lag between collection and analysis? Many analytics tools don't provide data in real time, often resulting in up to 24-hour lag times between collection and reporting. For example, you might have visibility into the front-end user experience, but not into back-end technical performance issues that might be impacting the frontend.
- **Lightning-fast querying.** In addition to real-time data collection, can you query the database and get an immediate answer? Do you have to wait for a nightly batch process to occur for an existing query to finish running before you can see the results? Many open source technologies have attempted to make progress on this front (Hadoop, Hive, Tez, Impala, Spark), but they are still not delivering the real-time analysis businesses need to succeed.
- **Flexibility and granularity of data.** Can you easily add custom data types and associated attributes using your analytics toolset? Can you add them at volume without hitting some arbitrary limit or needing to upgrade to a higher service tier? Companies need the ability to add data into their analytics tool and get user-level analyses out of their datasets. This means being able to go beyond simple aggregations to get useful segments of your data to generate insights that impact your bottom line.
- **Ease of setup and use.** What does it require to set up and get your analytics environment up and running? It shouldn't take a team of developers weeks or months to build out your data pipeline, nor should it require advanced data scientists to get the answers to your questions. Instead, look for a tool that's easy for both technical and business users to run queries.

Software analytics combines these key capabilities, gathering metrics in real time from live production software and transforming them into actionable data. It allows you to ask your software questions and get immediate answers. In short, software analytics is all about removing the time, complexity, and high cost of analytics, so that companies both big and small can make fast and informed data decisions that help propel the business forward. To learn more, visit: www.newrelic.com/insights.

by **Ankush Rustagi**
Product Marketing Manager, **New Relic**

New Relic Insights BY NEW RELIC

DATA PLATFORM | OPERATIONAL INTELLIGENCE

New Relic's analytics platform provides real-time data collection and querying capabilities based on closed-source database technologies.

DATABASE INTEGRATIONS
NEW RELIC QUERY LANGUAGE

HOSTING
SaaS

HADOOP
No Hadoop Support

INTEGRATION SUPPORT
None

NOTABLE CUSTOMERS
- Microsoft
- Nike
- Groupon
- Sony
- NBC
- Intuit

✓ High Availability
✓ Load Balancing
✓ Automatic Failover

STRENGTHS
- Provides actionable, real-time business insights from the billions of metrics your software is producing
- Collects every event automatically, as it happens, directly from the source
- Stores data in a cloud-hosted database - no installation, provisioning or configuration required
- Queries billions of real-time events in milliseconds using SQL-like query language
- Enables fast and informed decision making about your software, customers, and business

FULL PROFILE LINK	dzone.com/r/pdHQ	WEBSITE newrelic.com	TWITTER @newrelic	PROPRIETARY

THE EVOLUTION OF MAPREDUCE AND HADOOP

by Srinath Perera & Adam Diaz

With its Google pedigree, MapReduce has had a far-ranging impact on the computing industry [1]. It is built on the simple concept of mapping (i.e. filtering and sorting) and then reducing data (i.e. running a formula for summarization), but the true value of MapReduce lies with its ability to run these processes in parallel on commodity servers while balancing disk, CPU, and I/O evenly across each node in a computing cluster. When used alongside a distributed storage architecture, this horizontally scalable system is cheap enough for a fledgling startup. It is also a cost-effective alternative for large organizations that were previously forced to use expensive high-performance computing methods and complicated tools such as MPI (the Message Passing Interface library). With MapReduce, companies no longer need to delete old logs that are ripe with insights—or dump them onto unmanageable tape storage—before they've had a chance to analyze them.

HADOOP TAKES OVER

Today, the Apache Hadoop project is the most widely used implementation of MapReduce. It handles all the details required to scale MapReduce operations. The industry support and community contributions have been so strong over the years that Hadoop has become a fully-featured, extensible data-processing platform. There are scores of other open source projects designed specifically to work with Hadoop. Apache Pig and Cascading, for instance, provide high-level languages and abstractions for data manipulation. Apache Hive provides a data warehouse on top of Hadoop.

As the Hadoop ecosystem left the competition behind, companies like Microsoft, who were trying to build their own MapReduce platform, eventually gave up and decided to support Hadoop under the pressure of customer demand [2]. Other tech powerhouses like Netflix, LinkedIn, Facebook, and Yahoo (where the project originated) have been using Hadoop for years. A new Hadoop user in the industry, TRUECar, recently reported having a cost of $0.23 per GB with Hadoop. Before Hadoop, they were spending $19 per GB [3]. Smaller shops looking to keep costs even lower have tried to run virtual Hadoop instances. However, virtualizing Hadoop is the subject of some controversy amongst Hadoop vendors and architects. The cost and performance of virtualized Hadoop is fiercely debated.

Hadoop's strengths are more clearly visible in use cases such as clickstream and server log analytics. Analytics like financial risk scores, sensor-based mechanical failure predictions, and vehicle fleet route analysis are just some of the areas where Hadoop is making an impact. With some of these industries having 60 to 90 day time limits on data retention, Hadoop is unlocking insights that were once extremely difficult to obtain in time. If an organization is allowed to store data longer, the Hadoop File System (HDFS) can save data in its raw, unstructured form while it waits to be processed, just like the NoSQL databases that have broadened our options for managing massive data.

> *We don't really use MapReduce anymore.*
>
> *- Urs Hölzle, Google*

WHERE MAPREDUCE FALLS SHORT

- It usually doesn't make sense to use Hadoop and MapReduce if you're not dealing with large datasets like high-traffic web logs or clickstreams.
- Joining two large datasets with complex conditions—a problem that has baffled database people for decades—is also difficult for MapReduce.
- Machine learning algorithms such as KMeans and Support Vector Machines (SVM) are often too complex for MapReduce.
- When the map phase generates too many keys (e.g. taking the cross product of two datasets), then the mapping phase will take a very long time.
- If processing is highly stateful (e.g. evaluating a state machine), MapReduce won't be as efficient.

As the software industry starts to encounter these harder use cases, MapReduce will not be the right tool for the job, but Hadoop might be.

HADOOP ADAPTS

Long before Google's dropping of MapReduce, software vendors and communities were building new technologies to handle some of the technologies described above. The Hadoop project made significant changes just last year and now has a cluster resource management platform called YARN that allows developers to use many other non-MapReduce technologies on top of it. The Hadoop project contributors were already thinking about a resource manager for Hadoop back in early 2008 [4].

With YARN, developers can run a variety of jobs in a YARN container. Instead of scheduling the jobs, the whole YARN container is scheduled.

With YARN, developers can run a variety of jobs in a YARN container. Instead of scheduling the jobs, the whole YARN container is scheduled. The code inside that container can be any normal programming construct, so MapReduce is just one of many application types that Hadoop can harness. Even the MPI library from the pre-MapReduce days can run on Hadoop. The number of products and projects that the YARN ecosystem enables is too large to list here, but this table will give you an idea of the wide ranging capabilities YARN can support:

CATEGORY	PROJECT
Search	Solr, Elasticsearch
NoSQL	HBase, Accumulo
Streaming	Storm, Spark Streaming
In-Memory	Impala, Spark
Proprietary Apps and Vendors	Microsoft, SAS, SAP, Informatica, HP etc.

THREE WAYS TO START USING YARN

Below are three basic options for using YARN (but not the only options). The complexity decreases as you go down the list but the granular control over the project also decreases:

1. Directly code a YARN application master to create a YARN application. This will give you more control over the behavior of the application, but it will be the most challenging to program.
2. Use Apache Tez, which has a number of features including more complex directed acyclic graphs than MapReduce, Tez sessions, and the ability to express data processing flows through a simple Java API.
3. Use Apache Slider, which provides a client to submit JAR files for launching work on YARN-based clusters. Slider provides the least programmatic control out of these three options, but it also has the lowest cost of entry for trying out new code on YARN because it provides a ready to use application master.

For organizations migrating from Hadoop 1.x (pre-YARN) to Hadoop 2, the migration shouldn't be too difficult since the APIs are fully compatible between the two versions. Most legacy code should just work, but in certain very specific cases custom source code may need to simply be recompiled against newer Hadoop 2 JARs.
As you saw in the table, there are plenty of technologies that take full advantage of the YARN model to expand Hadoop's analysis capabilities far beyond the limits of the original Hadoop. Apache Tez greatly improves Hive query times. Cloudera's Impala project is a massively parallel processing (MPP) SQL query engine. And then there's Apache Spark, which is close to doubling its contributors in less than a year [5].

APACHE SPARK STARTS A FIRE

Spark is built specifically for YARN. In addition to supporting MapReduce, Spark lets you point to a large dataset and define a virtual variable to represent the large dataset. Then you can apply functions to each element in the dataset and create a new dataset. So you can pick the right functions for the right kinds of data manipulation. But that's not even the best part.

The real power of Spark comes from performing operations on top of virtual variables. Virtual variables enable data flow optimization across one execution step to the other, and they should optimize common data processing challenges (e.g. cascading tasks and iterations). Spark streaming uses a technology called "micro-batching" while Storm uses an event driven system to analyze data.

Apache Spark is close to doubling its contributors in less than a year.

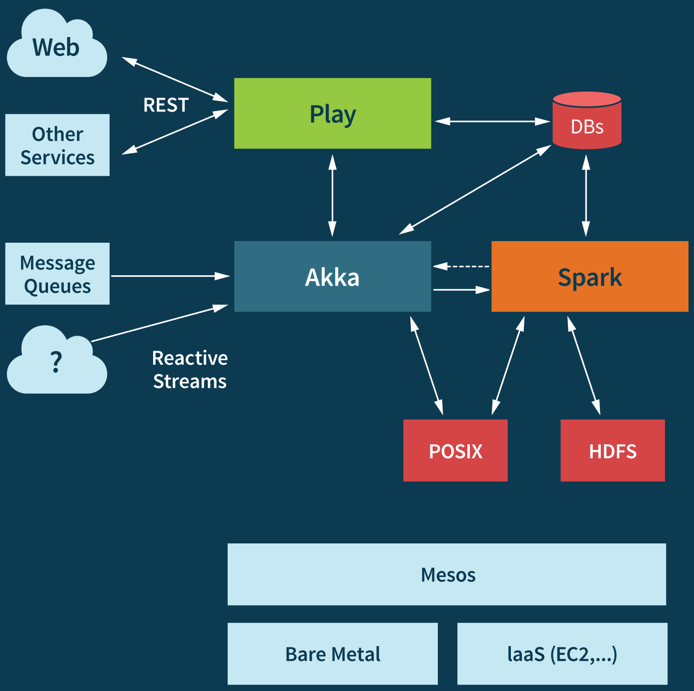
JUST ONE TOOL IN THE TOOLBOX

MapReduce's main strength is simplicity. When it first emerged in the software industry, it was widely adopted and soon became synonymous with Big Data, along with Hadoop. Hadoop is still the toolbox most commonly associated with Big Data, but now more organizations are realizing that MapReduce is not always the best tool in the box.

[1] http://research.google.com/archive/mapreduce.html
[2] http://www.zdnet.com/blog/microsoft/microsoft-drops-dryad-puts-its-big-data-bets-on-hadoop/11226
[3] http://blogs.wsj.com/cio/2014/06/04/hadoop-hits-the-big-time/
[4] https://issues.apache.org/jira/browse/MAPREDUCE-279
[5] http://inside-bigdata.com/2014/07/15/theres-spark-theres-fire-state-apache-spark-2014/

WRITTEN BY
Adam Diaz
Adam Diaz is a Hadoop Architect at Teradata. He has previously worked at big data powerhouses like IBM, SAS, and HortonWorks.

WRITTEN BY
Srinath Perera
Srinath Perera is a Research Director and architect at WSO2. He is a member of the Apache Software foundation, a PMC member of Apache Web Service project, a committer on Apache Axis, Axis2, and Geronimo, and a co-founder of Apache Axis2.


Web
REST
Other Services
Play
DBs
Message Queues
Akka
Spark
?
Reactive Streams
POSIX
HDFS
Mesos
Bare Metal
IaaS (EC2,...)

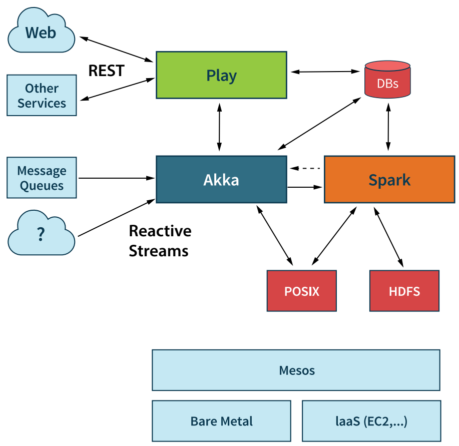
BUILDING BIG DATA APPLICATIONS WITH SPARK & THE TYPESAFE REACTIVE PLATFORM

WHY SPARK?

In the Hadoop community, an emerging consensus is forming around Apache Spark as the next-generation, multi-purpose compute engine for Big Data applications. Spark improves upon the venerable MapReduce compute engine in several key areas:

- Spark provides a more flexible and concise programming model for developers, and features significantly better performance in most production scenarios.
- Spark supports traditional batch-mode applications, but it also provides a streaming model.
- The functional programming foundation of Spark and its support for iterative algorithms provide the basis for a wide range of libraries—including SparkSQL for integrated SQL-based queries over data with defined schemas, Spark Streaming for handling incoming events in near-real time, GraphX for computations over graphs, and MLlib for machine-learning.
- Spark scales down to a single machine and up to large clusters. Spark jobs can run in Hadoop using the YARN resource manager, on a Mesos cluster, or in small standalone clusters.

Spark's greater flexibility offers new opportunities for integrating data analytics in event streaming reactive applications built with the Typesafe Reactive Platform. A possible architecture is shown in the following figure.



Services can be deployed and managed by Mesos, providing efficient allocation of cluster resources running on bare hardware or Infrastructure-as-a-Service (IaaS). Play and Akka implement web services and ingest reactive streams of data from message queues and other sources. Database access is also shown. Akka streams data to Spark, whose streaming model works on time slices of event traffic.

Spark is used to perform analytics: anything from running aggregations to machine-learning algorithms. Spark and Akka can read and write data in local or distributed file systems. Additional batch-mode Spark jobs would run periodically to perform large-scale data analysis, like aggregations over long time frames, and ETL (extract, transform, and load) tasks like data cleansing, reformatting, and archiving.

SPARK AND SCALA

Functional programming provides a set of operations on data that are broadly applicable and work together to build non-trivial transformations of data. The Scala library implements these operations for data sets that fit into memory for a single JVM process. Scala developers can write concise, expressive code, with high productivity.

The Spark API scales up the idioms of the Scala library to the size of clusters. Therefore, developers using Spark enjoy the same productivity benefits that other Scala developers enjoy. The Spark API is also remarkably similar to the Scala Collections API. Let's look at a simple example, the famous Word Count algorithm, where we read in one or more documents, tokenize them into words, then count the occurrences of every word.

The following listing shows an implementation using the Scala Collections API, where we read all the text from a single file.

SCALA CODE

```
import java.io._
import scala.io._

val wordsCounted = Source.fromFile(...)
// Read from a file,
      .getLines.map(line => line.toLowerCase)
      // convert to lower case,
      .flatMap(line => line.split(“””\W+”””)).toSeq
      // split into words,
      .groupBy(word => word)
      // group words together,
      .map { case (word, group) => (word, group.size) }
      // count group sizes,
val out = new PrintStream(new File(...))
// write results.
wordsCounted foreach (word_count => out.println(word_count)
```

The comments provide the essential details. Note how concise this source code is!

The Spark implementation looks almost the same. There are differences in handling the input and output, and in how the environment is set up and torn down, but the core logic is identical. The same idioms work for small data in a single process all the way up to a massive cluster.

SPARK CODE

```
import org.apache.spark.SparkContext
import org.apache.spark.SparkContext._

val sc = new SparkContext(“local”, “Word Count (2)”)
// “Context” driver
// Except for how input/output, the sequence of calls is identical.
val wordsCounted = sc.textFile(args(0)).map(line => line.toLowerCase)
      .flatMap(line => line.split(“””\W+”””))
      .groupBy(word => word)
      .map { case (word, group) => (word, group.size) }
wordsCounted.saveAsTextFile(args(1))
sc.stop()
```

Apache Spark is a natural extension to the Typesafe Reactive Platform for adding sophisticated data analytics.

by **Dean Wampler** Consultant, **Typesafe**

THE DEVELOPER'S GUIDE TO DATA SCIENCE

BY SANDER MAK

When developers talk about using data, they are usually concerned with ACID, scalability, and other operational aspects of managing data. But data science is not just about making fancy business intelligence reports for management. Data drives the user experience directly, not after the fact.

Large scale analysis and adaptive features are being built into the fabric of many of today's applications. The world is already full of applications that learn what we like. Gmail sorts our priority inbox for us. Facebook decides what's important in our newsfeed on our behalf. E-commerce sites are full of recommendations, sometimes eerily accurate. We see automatic tagging and classification of natural language resources. Ad-targeting systems predict how likely you are to click on a given ad. The list goes on and on.

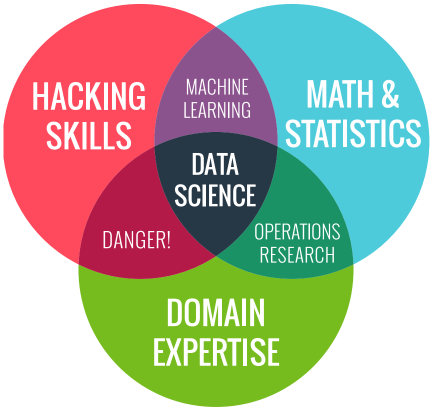
Many of the applications discussed above emerged from web giants like Google, Yahoo, and Facebook and other successful startups. Yes, these places are filled to the brim with very smart people, working on the bleeding edge. But make no mistake, this trend will trickle down into "regular" application development too. In fact, it already has. When users interact with slick and intelligent apps every day, their expectations for business applications rise as well. For enterprise applications it's not a matter of if, but when.

This is why many enterprise developers will need to familiarize themselves with data science. Granted, the term is incredibly hyped, but there's a lot of substance behind the hype. So we might as well give it a name and try to figure out what it means for us as developers.

FROM DEVELOPER TO DATA SCIENTIST

How do we cope with these increased expectations? It's not just a software engineering problem. You can't just throw libraries at it and hope for the best. Yes, there are great machine learning libraries, like Apache Mahout (Java) and scikit-learn (Python). There are even programming languages squarely aimed at doing data science, such as the R language. But it's not just about that. There is a more fundamental level of understanding you need to attain before you can properly wield these tools.

This article will not be enough to gain the required level of understanding. It can, however, show you the landmarks along the road to data science. This diagram (adapted from Drew Conway's original) shows the lay of the land [1]:



As software engineers, we can relate to hacking skills. It's our bread and butter. And that's good, because from that solid foundation you can branch out into the other fields and become more well-rounded.

Let's tackle domain expertise first. It may sound obvious, but if you want to create good models for your data, then you need to know what you're talking about. This is not strictly true for all approaches. For example, deep learning and other machine learning techniques might be viewed as an exception. In general though, having more domain-specific knowledge is better. So start looking beyond the user-stories in your backlog and talk to your domain experts about what really makes the clock tick. Beware though: if you only know your domain and can churn out decent code, you're in the danger zone. This means you're at risk of re-inventing the wheel, misapplying techniques, and shooting yourself in the foot in a myriad of other ways.

Of course, the elephant in the room here is "math & statistics." The link between math and the implementation of features such as recommendation or classification is very strong. Even if you're not building a recommender algorithm from scratch (which hopefully you wouldn't have to), you need to know what goes on under the hood in order to select the right one and to tune it correctly. As the diagram points out, the combination of domain expertise and math and statistics knowledge is traditionally the expertise area of researchers and analysts within companies. But when you combine these skills with software engineering prowess, many new doors will open.

> "HOW DO WE COPE WITH THESE INCREASED EXPECTATIONS? YOU CAN'T JUST THROW LIBRARIES AT IT AND HOPE FOR THE BEST."

What can you do as developer if you don't want to miss the bus? Before diving head-first into libraries and tools, there are several areas where you can focus your energy:

- Data management
- Statistics
- Math

We'll look at each of them in the remainder of this article. Think of these items as the major stops on the road to data science.

DATA MANAGEMENT

Recommendation, classification, and prediction engines cannot be coded in a vacuum. You need data to drive the process of creating/tuning a good recommender engine for your application, in your specific context. It all starts with gathering relevant data, which might already be in your databases. If you don't already have the data, you might have to set up new ways of capturing relevant data. Then comes the act of combining and cleaning data. This is also known as data wrangling or munging. Different algorithms have different pre-conditions on input data. You'll have to develop a strong intuition for good data versus messy data.

Typically, this phase of a data science project is very experimental. You'll need tools that help you quickly process lots of heterogeneous data and iterate on different strategies. Real world data is ugly and lacks structure. Dynamic scripting languages are often used to filter and organize data because they fit this challenge perfectly. A popular choice is Python with Pandas or the R language.

"THE DATA SCIENCE FIELD IS CURRENTLY DOMINATED BY PhDs. ON THE FLIPSIDE, WE LIVE IN AN AGE WHERE EDUCATION HAS NEVER BEEN MORE ACCESSIBLE."

It's important to keep a close eye on everything related to data munging. Just because it's not production code, doesn't mean it's not important. There won't be any compiler errors or test failures when you silently omit or distort data, but it will influence the validity of all subsequent steps. Make sure you keep all your data management scripts, and keep both mangled and unmangled data. That way you can always trace your steps. Garbage in, garbage out applies as always.

STATISTICS

Once you have data in the appropriate format, the time has come to do something useful with it. Much of the time you'll be working with sample data to create models that handle yet unseen data. How can you infer valid information from this sample? How do you even know your data is representative? This is where we enter the domain of statistics, a vitally important part of data science. I've heard it said: "a Data Scientist is a person who is better at statistics than any software engineer and better at software engineering than any statistician."

What should you know? Start by mastering the basics. Understand probabilities and probability distributions. When is a sample large enough to be representative? Know about common assumptions such as independence of probabilities, or that values are expected to follow a normal distribution. Many statistical procedures only make sense in the context of these assumptions. How do you test the significance of your findings? How do you select promising features from your data as input for algorithms? Any introductory material on statistics can teach you this. After that, move on the Bayesian statistics. It will pop up more and more in the context of machine learning.

It's not just theory. Did you notice how we conveniently glossed over the "science" part of data science up till now? Doing data science is essentially setting up experiments with data. Fortunately, the world of statistics knows a thing or two about experimental setup. You'll learn that you should always divide your data into a training set (to build your model) and a test set (to validate your model). Otherwise, your model won't work for real-world data: you'll end up with an overfitting model. Even then, you're still susceptible to pitfalls like multiple testing. There's a lot to take into account.

MATH

Statistics tells you about the when and why, but for the how, math is unavoidable. Many popular algorithms such as linear regression, neural networks, and various recommendation algorithms all boil down to math. Linear algebra, to be more precise. So brushing up on vector and matrix manipulations is a must. Again, many libraries abstract over the details for you, but it is essential to know what is going on behind the scenes in order to know which knobs to turn. When results are different than you expected, you need to know how to debug the algorithm.

It's also very instructive to try and code at least one algorithm from scratch. Take linear regression for example, implemented with gradient descent. You will experience the intimate connection between optimization, derivatives, and linear algebra when researching and implementing it. Andrew Ng's Machine Learning class on Coursera takes you through this journey in a surprisingly accessible way.

BUT WAIT, THERE'S MORE...

Besides the fundamentals discussed so far, getting good at data science includes many other skills, such as clearly communicating the results of data-driven experiments, or scaling whatever algorithm or data munging method you selected across a cluster for large datasets. Also, many algorithms in data science are "batch-oriented," requiring expensive recalculations. Translation into online versions of these algorithms is often necessary. Fortunately, many (open source) products and libraries can help with the last two challenges.

Data science is a fascinating combination between real-world software engineering, math, and statistics. This explains why the field is currently dominated by PhDs. On the flipside, we live in an age where education has never been more accessible, be it through MOOCs, websites, or books. If you want read a hands-on book to get started, read Machine Learning for Hackers, then move on to a more rigorous book like Elements of Statistical Learning. There are no shortcuts on the road to data science. Broadening your view from software engineering to data science will be hard, but certainly rewarding.

[1] http://drewconway.com/zia/2013/3/26/the-data-science-venn-diagram

WRITTEN BY

Sander Mak

Sander Mak works as Senior Software Engineer at Luminis Technologies. He has been awarded a JavaOne Rockstar award for his talk *Data Science with R for Java Developers* and is giving three talks at this year's JavaOne conference. Sander speaks regularly at various international developer conferences, sharing his passion for Java and JVM languages.

A MODERN APPROACH TO SQL-ON-HADOOP

Modern applications for social, mobile, and sensor data are generating an order of magnitude more data than ever before. It's not just the scale, but the variety and variability of these datasets that are a challenge. These datasets are often self-describing, include complex content and evolve rapidly, making it difficult for traditional DBAs to maintain the schemas required in SQL RDBMSs. This delays time to insight from data for business analysts.

> DRILL BRINGS THE SQL ECOSYSTEM AND RELATIONAL SYSTEM PERFORMANCE TO BIG DATA WITHOUT COMPROMISING ON HADOOP/NOSQL FLEXIBILITY

One such example is JSON, the lingua franca of data for APIs, data exchange, data storage, and data processing. HBase, another example, is a highly scalable NoSQL database capable of storing 1000s of columns in a single row, and every row has its own schema. Other formats/systems include Parquet, AVRO, and Protobuf.

New projects from the Apache Hadoop community such as Apache Drill take a different approach to SQL-on-Hadoop. The goal: perform self-service data exploration by bringing the SQL ecosystem and performance of the relational systems to Big Data scale without compromising on Hadoop/NoSQL flexibility.

The core elements of Drill include:

- **Agility:** Perform direct queries on self-describing, semi-structured data in files and HBase tables without needing to specify metadata definitions in a centralized store; this saves weeks or months on data preparation, modeling, and subsequent schema management.
- **Flexibility:** Drill provides a JSON-like internal data model to represent and process data, so you can query both simple and complex/nested data types.
- **Familiarity:** Use Drill to leverage familiar ANSI SQL syntax and BI/analytics tools through JDBC/ODBC drivers.

To learn more, read the quick start guide and visit the Apache Drill web site.

by **Neeraja Rentachintala**
Director of Product Managment, **MapR**

MapR Distribution including Apache Hadoop BY MAPR TECHNOLOGIES

DATA PLATFORM DATA MANAGEMENT, DATA INTEGRATION, ANALYTICS

MapR's distribution features true built-in enterprise-grade features like high availability, full multi-tenancy, integrated optimized NoSQL, and full NFS access.

DATABASE INTEGRATIONS

ORACLE HBASE IBM DB2
SQL SERVER MONGODB MAPR-DB

HOSTING

SaaS, PaaS, On-Premise

HADOOP

Built on Hadoop

INTEGRATION SUPPORT

ETL

NOTABLE CUSTOMERS

· Cisco
· comScore
· HP
· Samsung
· Beats Music
· TransUnion

✓ High Availability
✓ Load Balancing
✓ Automatic Failover

STRENGTHS

· Built-in enterprise grade capabilities to ensure no data or work loss despite disasters
· Full multi-tenancy to manage distinct user groups, data sets, or jobs in a single cluster
· Higher performance to do more work with less hardware, resulting in lower TCO
· Integrated security to ensure enterprise data is only accessed by authorized users
· Integrated NoSQL to run combined operational and analytical workloads in one cluster

FULL PROFILE LINK	dzone.com/r/7u6h	WEBSITE mapr.com	TWITTER @mapr	PROPRIETARY & OPEN SOURCE

This blog post was written by DZone's in-house content curator Alec Noller, who actively writes for DZone's NoSQL Zone. You can stop by DZone's NoSQL Zone for daily news and articles: http://dzone.com/mz/nosql

NoSQL'S OVERSATURATION PROBLEM

BY ALEC NOLLER

It's a familiar story at this point - trying out NoSQL, then moving back to relational databases - and the response is generally consistent as well: NoSQL will only be useful if you understand your individual problem and choose the appropriate solution.

According to Matthew Mombrea at IT World, though, that doesn't quite cover it. In this recent article, he shares his own "NoSQL and back again" thoughts, which hinge on the idea that there are simply too many NoSQL solutions out there, preventing newcomers from jumping right in.

Mombrea acknowledges that there are use cases where NoSQL is ideal. However, he argues that there are some major drawbacks that require additional effort:

> *It's helpful to think of NoSQL as a flat file storage system where the filename is the key and the file contents are the value. You can store whatever you want in these files and you can read/write to them very quickly, but . . . the brains of a relational database are gone and you're left to implement everything you've taken for granted with SQL in your code . . . for every application. The overhead is not justifiable for most applications.*

Beyond that, he argues that the advantages of NoSQL don't even apply to most use cases:

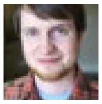
> *The big draw to NoSQL is it's ability to scale out with ease and to provide very high throughput. While it would be really nice to have the same scalability with an RDBMS, the real world fact is that 99% of applications will never operate at a scale where it matters. Look at Stack Exchange. They are one of the most trafficked sites on the planet and they run on MSSQL using commodity servers.*

Given these drawbacks, how is one to decide what solution is appropriate? If nothing else, NoSQL demands quite a bit more research, which is hard to justify when it's not even clear that it's necessary or beneficial. This is potentially an oversimplification, though, of the use cases that call for NoSQL solutions. According to Moshe Kaplan's list of times when one ought to choose MongoDB over MySQL, for example, there are quite a few other scenarios. Just a few ideas:

- If you need high availability in an unreliable environment
- If your data is location-based
- If you don't have a DBA

Mombrea's conclusion, though, still hits an interesting point: NoSQL is young, and adoption may become more practical as it matures, given that such a central aspect is understanding which solution is appropriate for any given job. That may be a more manageable task when the market has thinned a bit, leaving behind a handful of well-tested solutions to well-defined problems.

FIND THIS ARTICLE ONLINE:
http://bit.ly/1z45kCY



Alec Noller is the Senior Content Curator at DZone. When he's not creating and curating content for DZone, he spends his time writing and developing Java and Android applications.

FIND MORE AT DZONE'S NOSQL AND BIG DATA ZONES:

NoSQL: dzone.com/mz/nosql
Big Data: dzone.com/mz/big-data

DIVING DEEPER INTO BIG DATA

TOP TEN #BIGDATA TWITTER FEEDS

@KirkDBorne @kdnuggets @marcusborba @data_nerd @BigDataGal

@jameskobielus @medriscoll @spyced @IBMbigdata @InformaticaCorp

TOP 6 BIG DATA WEBSITES

Big Data University *bit.ly/1pdcuxt* A collection of courses that teach users about a wide variety of Big Data concepts, frameworks, and uses, including Hadoop, analytics, and relational databases.

Big Data and the History of Information Storage *bit.ly/1AHocGZ* A timeline of Big Data concepts, from the 1880 US census to the modern digital data explosion.

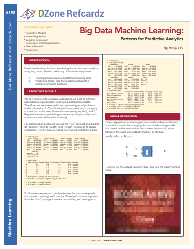
The Data Store: on Big Data *bit.ly/1oDxZYW* Current news and information on Big Data concepts from The Guardian.

Top 5 Big Data Resources *bit.ly/1qKSg3i* A list of five top articles about various Big Data concepts, from Hadoop to data mining to machine learning.

DB-Engines Ranking *bit.ly/1rZY37n* Database rankings according to popularity and industry prevalance.

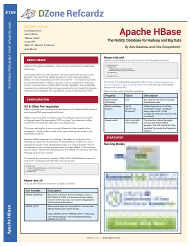
What is Big Data? *oreil.ly/1tQ62E1* O'Reilly's introduction to Big Data's basic concepts.

DZONE REFCARDZ



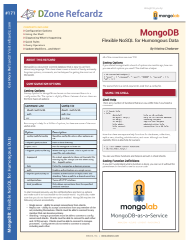
Big Data Machine Learning: *Patterns for Predictive Analytics* *bit.ly/WUFymk*

This Refcard covers machine learning for predictive analytics, a powerful instrument in the developer's Big Data toolbox.



Apache HBase: *The NoSQL Database for Hadoop and Big Data* *bit.ly/1nRgDYh*

HBase is the Hadoop database: a distributed, scalable Big Data store that lets you host very large tables — billions of rows multiplied by millions of columns — on clusters built with commodity hardware.



MongoDB: *Flexible NoSQL for Humongous Data* *bit.ly/YEgXEi*

This cheat sheet covers a bunch of handy and easily forgotten options, commands, and techniques for getting the most out of MongoDB.

Getting Started with Apache Hadoop *bit.ly/1wnFbQI*

This Refcard presents Apache Hadoop, a software framework that enables distributed storage and processing of large datasets using simple high-level programming models.

DZONE BIG DATA ZONES

Big Data Zone *http://dzone.com/mz/big-data*

We're on top of all the best tips and news for Hadoop, R, and data visualization technologies. We also give you advice from data science experts on how to understand and present that data.

NoSQL Zone *http://dzone.com/mz/nosql*

DZone's portal for following the news and trends of the non-relational database ecosystem, including solutions such as MongoDB, Cassandra, Redis, and many others.

SQL Zone *http://www.dzone.com/mz/sql*

DZone's portal for following the news and trends of the relational database ecosystem, which includes solutions such as MySQL, PostgreSQL, SQL Server, and many others.

TOP 6 BIG DATA TUTORIALS

Hadoop Tutorial Modules *yhoo.it/1wiJhH3*
The Yahoo! Hadoop tutorial, with an introduction to Hadoop and Hadoop tutorial modules.

Hadoop, Hive, and Pig *bit.ly/1wiJbiM*
A tutorial from Hortonworks on using Hadoop, Hive, and Pig.

R Introduction *bit.ly/1rVhyrM*
An in-depth introduction to the R language.

Hive Tutorial *bit.ly/1qKRlQ8*
A tutorial on getting started with Hive.

Using R in Hive *bit.ly/1pXZtl7*
A tutorial on using R in MapReduce and Hive.

91 R Tutorials *bit.ly/1sD047U*
91 tutorials to help you explore R.

THE DIY BIG DATA CLUSTER

BY CHANWIT KAEWKASI

Hadoop has been widely adopted to run on both public and private clouds. Unfortunately, the public cloud is not always a safe place for your sensitive data.

The Heartbleed exploit is one recent example of a major security bug in many public infrastructures that went undiscovered for years. For some organizations, it makes more sense to use a private cloud. Unfortunately, private clouds can be costly both in terms of building and operating, especially with x86-class processors. To save on operating costs, Baidu (the largest Chinese search engine) has recently changed its servers from x86-based to custom ARM servers. After this transition, they reported 25% savings in total cost of ownership.

An on-premise Hadoop cluster built using system-on-a-chip (SoC) boards is the perfect answer for a cash-strapped startup or low-budget departmental experiment. It removes the need for a data center and gives you more control over security and tuning than a public cloud service. This article will introduce a method for building your own ARM-based Hadoop cluster. Even if you aren't interested in actually building one of these clusters, you can still learn a lot about how Hadoop clusters are modeled, and what the possibilities are.

THE AIYARA CLUSTER

Readers are always amazed when I show them that they can have their own Hadoop cluster for data analytics without a data center. My colleagues and I call ours the Aiyara cluster model for Big Data.

The first Aiyara cluster was presented on DZone as an ARM cluster consisting of 22 Cubieboards (Cubieboard A10). These ARM-based SoC boards make it fully modular so that each node can be easily replaced or upgraded. If you're familiar with ARM architecture, you know that these devices will produce less heat (no cooling system required) and consume less power than other chip architectures like x86. In Thailand, it costs our group $0.13 a day to run our 22-board cluster.

Twenty boards are used as Spark worker boards and the other two are the master and driver boards. All of the worker nodes are connected to a solid-state drive (SSD) via SATA port, and two of the workers are also running Hadoop data nodes.

As far as performance goes, our cluster is even faster than we expected. The software stack we use is simply Apache Spark over HDFS, and in our benchmarks we are able to batch process 34GB of Wikipedia articles in 38 minutes. After batch processing, each ad hoc query can be executed in 9-14 seconds.

> "WE ARE ABLE TO BATCH PROCESS 34GB OF WIKIPEDIA ARTICLES IN 38 MINUTES. AFTER BATCH PROCESSING, EACH AD HOC QUERY CAN BE EXECUTED IN 9-14 SECONDS."

BUILDING YOUR OWN CLUSTER

If you'd like to try making your own ARM cluster, I'd recommend starting with a 10-node cluster. You can pick an alternate board or use the same one that we did. Eight of them will be workers and the other two will be your master and driver boards. Your board can be any ARM SoC that has:

1. At least 1 GHz CPU
2. At least 1 GB memory
3. At least 4 GB of built-in NAND flash
4. A SATA connector
5. A 100 Mbps or greater ethernet connector

You'll also need an ethernet switch with enough ports. The size of each SSD depends on how much storage you want. 120 GB for each board would be fine for a 10-node cluster, but the default replication level of Hadoop is three, so your usable capacity will be one-third of the total capacity. Finally, you'll need a power supply large enough for the cluster and you'll need a power splitter to send power to all the boards.

These are the minimum requirements for building an Aiyara cluster.

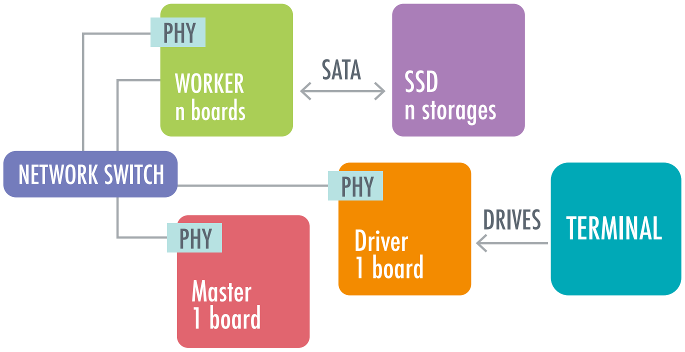


Fig 1. Block Diagram for Physical Cluster Model

SOFTWARE STACK

To build an Aiyara cluster, you need the software packages listed below.

- Linux 3.4+ (distribution of your choice)
- Java Development Kit 1.7+ for ARM
- Apache Hadoop 2.3+ and Apache Spark 0.9+ from CDH5
- Python 2.4 and Ansible 1.6+ for cluster-wide management
- Ganglia or JMX for cluster monitoring

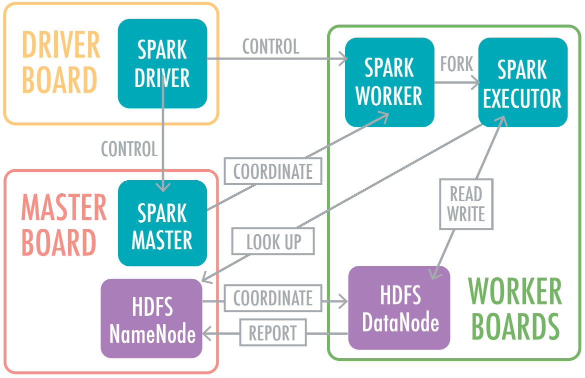


Fig 2. Overview of the logical clusters, HDFS, and Spark, laid atop the hardware boards.

MANAGING PERFORMANCE

We normally use the internal NAND flash to store the operating system, Hadoop, and Spark on each node. External SSDs should only be used to store data written by HDFS. When something goes wrong with an SSD, we can just replace it with a new one. Then, we just ask Hadoop to re-balance data to the newly added storage. In contrast, when a Worker board fails, we can just re-flash the whole file system to its internal NAND flash. If this recovery process does not help, just throw the board away and buy a new one (Cubieboards cost us $49 each). That's one of the big advantages of having a commodity cluster.

Because of memory limitations on each node, we configure Spark to spill intermediate results to internal NAND flash when the JVM heap is full. But writing a large set of small files to the ext4fs over a NAND device is not a good idea. Many write failures occurred and the file system became unstable.

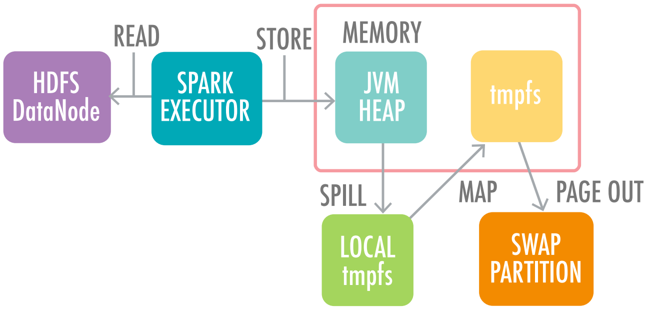


Fig 3. The cluster architecture to solve spilling problems on ext4fs over the NAND flash device.

For an Aiyara cluster, we solve this problem by setting up a swap partition up for each Worker node, then mounting a tmpfs (an in-memory file system) to use as the spill directory for Spark. In our Aiyara Cluster Mk-I, we have a 2 GB swap partition on each node. When a Spark Executor spills intermediate results to the tmpfs and later paging out to the disk, small files in the tmpfs will be grouped as a larger block.

Performance degradation from paging out generally does not affect a Big Data cluster for batch processing, so we do not need to worry about this issue. The only requirement is that we tune network parameters to prevent dissociation of Spark's executors.

MANAGE THE CLUSTER

The host name and IP address on each node is set through DHCP during starting up via the DHCP host name script. We have found that it is good to map the host name to the node's IP address rather than allow it to be assigned randomly. For example, we map 192.168.0.11 to node01 and 192.168.0.22 to node12. It is the "node${ip -10}" pattern. This technique allows us to scale up to 200 nodes per a logical rack. It will work fine in most cases.

We write Bash scripts to perform cluster-wide operations through SSH. For example, we have a script to perform disk checks on every worker node using the fsck command via SSH before starting the cluster.

The following are steps for properly starting the cluster:

- Perform file system check
- Mount SSD on every node. Rather than having fstab to auto-mount on each node, we mount the storage manually via the master node.
- Start HDFS, NameNode, and DataNode
- Start Spark, master, and worker

When it comes to maintaining software packages, doing so using plain scripts and SSH becomes harder to manage. To make cluster management easier, Ansible is a natural choice for us. Ansible is compatible with the armhf (ARM) architecture on Debian and it uses agent-less architecture, so we only need to install it on the master board.

For monitoring, there are several tools you can use. If you would like to use Ganglia to monitor the whole cluster, you need to install Ganglia's agent on each node. My team chose a more convenient option; we just use the JMX to monitor all Spark nodes with VisualVM, a tool that comes with the JDK. Techniques for monitoring Linux and Java servers can be generally applied to this kind of cluster. With JMX, we can observe not only CPU usage, but also JVM-related resources such as garbage collection and thread behavior. Logging from HDFS and Spark are also important for troubleshooting.

START BUILDING YOUR OWN!

My team believes that the Aiyara cluster model is a viable solution for batch processing, stream processing, and interactive ad-hoc querying on a shoestring budget. All of the components and techniques have been thoroughly tested during our research. In the near future, ARM SoC boards will become cheaper and even more powerful. I believe that having this kind of low-cost Big Data cluster in a small or medium size company will become a more compelling alternative to managing a data center or outsourcing your data processing.

WRITTEN BY
Chanwit Kaewkasi

Chanwit Kaewkasi is an Assistant Professor at the Suranaree University of Technology's School of Computer Engineering in Thailand. He currently co-develops a series of low-cost Big Data clusters with Spark and Hadoop. He is also a contributor to the Grails framework, and leads development of the ZK plugin for Grails.

Big Data in 10 Minutes?

It’s true. With 1-Button Deploy™ Big Data is easy.

Don’t get bogged down with the boring side of Big Data.

Our technology lets you get started today without a long deployment.

Choose from Hadoop, MongoDB, Cassandra and more.

Visit

gogrid.com/ods

to learn more and get started with a free trial.

GOGRID

BIG DATA & MULTI-CLOUD GO HAND-IN-HAND

In IT organizations around the globe, CTOs, CIOs, and CEOs are asking the same question: "how can we use Big Data technologies to improve our platform operations?" Whether you're responsible for overcoming real-time monitoring and alerting obstacles, or providing solutions to platform operations analysis, behavioral targeting, and marketing operations, the solutions for each of these use cases can vary widely.

When it comes to Big Data and the cloud, variety is the key. There isn't a single one-size-fits-all solution for every one of your use cases and assuming a single solution fits all use cases is a pitfall that could cost you your job. As a result, companies are frequently using three to five Big Data solutions, and their platform infrastructure now spans a mix of cloud and dedicated servers.

With the freedom to use multiple solutions, the challenge is how to use them effectively. Whether you are choosing a cloud provider or a Big Data technology, you never want to be locked into a single vendor. When you're evaluating solutions, it makes sense to try out a few options, run some tests, and ensure you have the right solution for your particular use case.

Given the wide variety of new and complex solutions, however, it's no surprise that a recent survey of IT professionals showed that more than 55% of Big Data projects fail to achieve their goals. The most significant challenge cited was a lack of understanding of and the ability to pilot the range of technologies on the market.

BUSINESSES NO LONGER HAVE TO TIE THEMSELVES TO A SINGLE SOLUTION OR VENDOR.

This challenge systematically pushes companies toward a limited set of proprietary platforms that often reduce the choice down to a single technology, perpetuating the tendency to seek one cure-all technology solution. But this is no longer a realistic strategy.

No single technology such as a database can solve every problem, especially when it comes to Big Data. Even if such a unique solution could serve multiple needs, successful companies are always trialing new solutions in the quest to perpetually innovate and thereby achieve (or maintain) a competitive edge.

It's easy for an executive to tell you, "I want to use Hadoop," but it's your job that's on the line if Hadoop doesn't meet your specific needs. Similarly, if your cloud vendor has a vulnerability, only businesses with a multi-cloud strategy can shift workloads to another cloud with little or no impact to their customers.

Bottom line, businesses no longer have to tie themselves to a single solution or vendor and hope it's the right decision. Freedom to choose use case specific solutions on top of a reliable, multi-cloud infrastructure empowers businesses to take advantage of the best technologies without the risk.

WRITTEN BY
by Andrew Nester
Director of Marketing, **GoGrid LLC**

GoGrid Orchestration Services BY GOGRID

BIG DATA CLOUD | BIG DATA PAAS

GoGrid offers a dedicated PaaS for running big data applications, including several one-button deployments for solutions to simplify the production of big data apps.

DESCRIPTION

GoGrid offers an easier, faster way for users to take advantage of big data with a cloud platform. Businesses can benefit from lower costs, orchestrated solution deployment, and support for several open source solutions. These solutions are integrated through a 1-Button Deploy system to simplify the process of moving Big Data applications from trial to pilot project and finally to full-scale production.

STRENGTHS

- Button deploy support for several databases including Cassandra, DataStax, and HBase
- Utilizes Hadoop for predictive analytics, processing of large data sets, clickstream analysis, and managing log file data
- Support for several open source solutions means customers have no proprietary or platform lock-in
- PaaS solution designed specifically for working with big data

NOTABLE CUSTOMERS

- Condé Nast Digital
- Merkle
- Glam
- MartiniMedia
- Artizone

FREE TRIAL

14-day free trial

FULL PROFILE LINK dzone.com/r/Tvu6

WEBSITE gogrid.com

TWITTER @gogrid

PROPRIETARY

FINDING THE DATABASE FOR YOUR USE CASE

This chart will help you find the best types of databases to try testing with your software.

DB TYPES	STRONG USE CASES	WEAK USE CASES
Relational DB *Examples:* MySQL, PostgreSQL, SQL Server	○ When ACID transactions are required ○ Looking up data by different keys with secondary indexes (also a feature of several NoSQL DBs) ○ When strong consistency for results and queries is required ○ Conventional online transaction processing ○ Risk-averse projects seeking very mature technologies and widely available skills ○ Products for enterprise customers that are more familiar with relational DBs	○ Systems that need to tolerate partition failures ○ Schema-free management ○ Handling any complex / rich entities that require you to do multiple joins to get the entire entity back.
Key-Value Store *Examples:* Redis, Riak, DynamoDB	○ Handling lots of small, continuous, and potentially volatile reads and writes; also look for any DB with fast in-memory access or SSD storage ○ Storing session information, user preferences, and e-commerce carts ○ Simplifying the upgrade path of your software with the support of optional fields, adding fields, and removing fields without having to build a schema migration framework	○ Correlating data between different sets of keys ○ Saving multiple transactions (Redis is exempt from this weakness) ○ Performing well during key searches based on values (DynamoDB is exempt) ○ Operating on multiple keys (it's only possible through the client side) ○ Returning only partial values is required ○ Updates in place are necessary
Document Store *Examples:* MongoDB, Couchbase, RavenDB	○ Handling a wide variety of access patterns and data types ○ Handling reads with low latency ○ Handling frequently changing, user generated data ○ Simplifying the upgrade path of your software with the support of optional fields, adding fields, and removing fields without having to build a schema migration framework ○ Deployment on a mobile device (Mobile Couchbase)	○ Atomic cross-document operations (RavenDB is exempt) ○ Querying large aggregate data structures that frequently change ○ Returning only partial values is required ○ Joins are desired ○ Foreign key usage is desired ○ Partial updates of documents (especially child/sub-documents)
Column Store *Examples:* Cassandra, HBase, Accumulo	○ When high availability is crucial, and eventual consistency is tolerable ○ Event Sourcing ○ Logging continuous streams of data that have no consistency guarantees ○ Storing a constantly growing set of data that is accessed rarely ○ Deep visitor analytics ○ Handling frequently expiring data (Redis can also set values to expire)	○ Early prototyping or situations where there will be significant query changes (high cost for query changes compared to schema changes) ○ Referential integrity required ○ Processing many columns simultaneously
Graph Store *Examples:* Neo4j, Titan, Giraph	○ Handling entities that have a large number of relationships, such as social graphs, tag systems, or any link-rich domain. ○ Routing and location services ○ Recommendation engines or user data mapping ○ Dynamically building relationships between objects with dynamic properties ○ Allowing a very deep join depth	○ High volume write situations ○ Serving and storing binary data ○ Querying unrestricted across massive data sets

Sources: NoSQL Distilled, High Scalability

THE SOLUTIONS DIRECTORY

This directory of data management and analysis tools provides comprehensive, factual comparison data gathered from third-party sources and the tool creators' organizations. Solutions in the directory are selected based on several impartial criteria including solution maturity, technical innovativeness, relevance, and data availability. The solution summaries underneath the product titles are based on the organization's opinion of its most distinguishing features.

NOTE: The bulk of information gathered about these solutions is not present in these quarter-page profiles. For example, the Language/Drivers Supported and Database Integration sections only contain a subset of the databases and languages that these solutions support. To view an extended profile of any product, you can click the short-code link found at the bottom of each profile, or simply go to dzone.com/zb/products and enter the shortcode at the end of the link.

FULL PROFILE LINK

dzone.com/r/QM4k

Get easy access to full product profiles with this URL.

Actian Analytics Platform ACTIAN

DATA PLATFORM | DATA MANAGEMENT, DATA INTEGRATION, ANALYTICS

Actian's platform offers complete end-to-end analytics, built with next gen software architecture that can be run on commodity hardware.

HADOOP SUPPORT Hadoop integrations available	DB INTEGRATIONS ORACLE SQL SERVER IBM DB2 SAP HANA MONGODB
INTEGRATION SUPPORT • ETL • ELT	STATISTICAL LANGUAGES • R • SAS
BUILT-IN IDE No IDE	CLOUD HOSTING SaaS, PaaS, On-Premise
STREAM PROCESSING No	MAPREDUCE JOB DESIGNER No
PROPRIETARY	FULL PROFILE LINK dzone.com/r/QM4k
TWITTER @ActianCorp	WEBSITE actian.com

Aerospike AEROSPIKE

DATABASE | IN-MEMORY, UNORDERED KEY-VALUE

Aerospike is a flash-optimized database that indexes data in RAM or flash for predictable low latency, high throughput, and ACID transactions.

REPLICATION • Synchronous • Asynchronous	LANGUAGES/DRIVERS SUPPORTED C C++ JAVA NODE.JS PYTHON
SQL SUPPORT No	TRANSACTIONS SUPPORTED "Compare-and-set" type transactions
CONSISTENCY MODEL Strong consistency	INDEXING CAPABILITIES Simple indexes
AUTO-SHARDING Yes	FULL TEXT SEARCH No
OPEN SOURCE	FULL PROFILE LINK dzone.com/r/j3GR
TWITTER @aerospikedb	WEBSITE aerospike.com

Amazon Kinesis AMAZON

DATA PLATFORM | DATA MANAGEMENT, DATA INTEGRATION, ANALYTICS

Kinesis is a fully managed service for real-time processing of streaming data at massive scale.

HADOOP SUPPORT Hadoop integrations available	DB INTEGRATIONS MYSQL POSTGRESQL MONGODB
INTEGRATION SUPPORT • ETL • ELT	STATISTICAL LANGUAGES • None
BUILT-IN IDE No IDE	CLOUD HOSTING SaaS, PaaS
STREAM PROCESSING Yes	MAPREDUCE JOB DESIGNER Yes
PROPRIETARY	FULL PROFILE LINK dzone.com/r/VsbC
TWITTER @awscloud	WEBSITE aws.amazon.com

BigMemory SOFTWARE AG

DATA PLATFORM DATA MANAGEMENT, DATA INTEGRATION

BigMemory supports hundreds of terabytes of data in-memory with a predictable latency of low milliseconds regardless of scale.

HADOOP SUPPORT
No Hadoop Support

DB INTEGRATIONS
ORACLE SQL SERVER IBM DB2 MYSQL

INTEGRATION SUPPORT
- ETL

BUSINESS MODELER
Yes

BUILT-IN IDE
Eclipse

CLOUD HOSTING
SaaS, On-Premise

STREAM PROCESSING
Yes

MAPREDUCE JOB DESIGNER
No

OPEN SOURCE

FULL PROFILE LINK
dzone.com/r/LG7u

TWITTER @softwareag_NA WEBSITE terracotta.org

BIRT iHub ACTUATE

DATA PLATFORM DATA MANAGEMENT, DATA INTEGRATION, ANALYTICS

BIRT iHub has inherent extensibility at multiple levels, enabling developers to address the most complex application requirements.

HADOOP SUPPORT
Built on Hadoop

DB INTEGRATIONS
ORACLE SQL SERVER IBM DB2 SAP HANA MONGODB

INTEGRATION SUPPORT
- ELT

STATISTICAL LANGUAGES
- None

BUILT-IN IDE
Eclipse

CLOUD HOSTING
SaaS, PaaS, On-Premise

STREAM PROCESSING
Yes

MAPREDUCE JOB DESIGNER
Yes

PROPRIETARY

FULL PROFILE LINK
dzone.com/r/YyWw

TWITTER @actuate WEBSITE actuate.com

Cassandra

DATABASE COLUMNAR

Apache Cassandra is a NoSQL database originally developed at Facebook, and is currently used at tech firms like Adobe and Netflix.

REPLICATION
- Synchronous
- Asynchronous

LANGUAGES/DRIVERS SUPPORTED
C# GO JAVA NODE.JS RUBY

SQL SUPPORT
No

TRANSACTIONS SUPPORTED
No support

CONSISTENCY MODEL
Tunable per operation

INDEXING CAPABILITIES
Rich query language

AUTO-SHARDING
Yes

FULL TEXT SEARCH
Via DataStax

OPEN SOURCE

FULL PROFILE LINK
dzone.com/r/zrPN

TWITTER @cassandra WEBSITE cassandra.apache.org

Cloudera Enterprise CLOUDERA

DATA PLATFORM DATA MANAGEMENT, DATA INTEGRATION, ANALYTICS

A unified platform with compliance-ready security and governance, holistic system management, broad partner integration, and world-class support.

HADOOP SUPPORT
Built on Hadoop

DB INTEGRATIONS
ORACLE SQL SERVER IBM DB2 MONGODB HBASE

INTEGRATION SUPPORT
- ETL
- ELT

STATISTICAL LANGUAGES
- R
- SAS

BUILT-IN IDE
No IDE

CLOUD HOSTING
SaaS

STREAM PROCESSING
Yes

MAPREDUCE JOB DESIGNER
No

PROPRIETARY & OPEN SOURCE

FULL PROFILE LINK
dzone.com/r/Jf3V

TWITTER @Cloudera WEBSITE cloudera.com

Continuuity Reactor CONTINUUITY

DATA PLATFORM DATA MANAGEMENT, DATA INTEGRATION, ANALYTICS

Allows developers to build data applications quickly by enabling them to focus on business logic and value rather than infrastructure.

HADOOP SUPPORT
Built on Hadoop

DB INTEGRATIONS
MONGODB CASSANDRA HBASE

INTEGRATION SUPPORT
- ETL
- ELT

BUSINESS MODELER
No

BUILT-IN IDE
None

CLOUD HOSTING
SaaS, PaaS

STREAM PROCESSING
Yes

MAPREDUCE JOB DESIGNER
No

OPEN SOURCE

FULL PROFILE LINK
dzone.com/r/HaQ4

TWITTER @continuuity **WEBSITE** continuuity.com

Couchbase Server COUCHBASE

DATABASE IN-MEMORY, UNORDERED KEY-VALUE, DOCUMENT

Couchbase Server features an integrated cache, rack awareness, cross data center replication, and an integrated administration console.

REPLICATION
- Synchronous
- Asynchronous

SQL SUPPORT
No

CONSISTENCY MODEL
Strong consistency

AUTO-SHARDING
Yes

LANGUAGES/DRIVERS SUPPORTED
C C++ JAVA NODE.JS RUBY

TRANSACTIONS SUPPORTED
“Compare-and-set” type transactions

INDEXING CAPABILITIES
Rich query language

FULL TEXT SEARCH
Via ElasticSearch

OPEN SOURCE

FULL PROFILE LINK
dzone.com/r/RGL7

TWITTER @couchbase **WEBSITE** couchbase.com

DataTorrent RTS DATATORRENT

DATA PLATFORM DATA MANAGEMENT

DataTorrent RTS enable real-time insights via an easy to use high performance, scalable, fault-tolerant Hadoop 2.0 native-platform.

HADOOP SUPPORT
Built on Hadoop

DB INTEGRATIONS
ORACLE SQL SERVER IBM DB2 SAP HANA MONGODB

INTEGRATION SUPPORT
- ETL

BUSINESS MODELER
No

BUILT-IN IDE
None

CLOUD HOSTING
On-Premise

STREAM PROCESSING
Yes

MAPREDUCE JOB DESIGNER
Yes

PROPRIETARY

FULL PROFILE LINK
dzone.com/r/aPAt

TWITTER @datatorrent **WEBSITE** datatorrent.com

FoundationDB FOUNDATIONDB

DATABASE ORDERED KEY-VALUE

FoundationDB provides a key-value API with ordering and full ACID transactions that allow users to layer multiple data models.

REPLICATION
- Synchronous

SQL SUPPORT
Full ANSI SQL

CONSISTENCY MODEL
Strong consistency

AUTO-SHARDING
Yes

LANGUAGES/DRIVERS SUPPORTED
C JAVA NODE.JS RUBY PYTHON

TRANSACTIONS SUPPORTED
Arbitrary multi-statement transactions spanning arbitrary nodes

INDEXING CAPABILITIES
Rich query language

FULL TEXT SEARCH
No

PROPRIETARY

FULL PROFILE LINK
dzone.com/r/fsVb

TWITTER @FoundationDB **WEBSITE** foundationdb.com

Hazelcast HAZELCAST

DATABASE IN-MEMORY, RELATIONAL, DOCUMENT, KEY-VALUE

Hazelcast is a small, open source, 3.1MB JAR database library with no external dependencies that is easily embeddable into database apps.

REPLICATION
- Synchronous
- Asynchronous

LANGUAGES/DRIVERS SUPPORTED
C C++ JAVA PYTHON SCALA

SQL SUPPORT
Limited subset

TRANSACTIONS SUPPORTED
Arbitrary multi-statement transactions spanning arbitrary nodes

CONSISTENCY MODEL
Tunable per database

INDEXING CAPABILITIES
Simple indexes

AUTO-SHARDING
Yes

FULL TEXT SEARCH
Via open source libraries

OPEN SOURCE

FULL PROFILE LINK
dzone.com/r/MPaA

TWITTER @hazelcast

WEBSITE hazelcast.com

HBase

DATABASE COLUMNAR

HBase excels at random, real-time read/write access to very large tables of data atop clusters of commodity hardware.

REPLICATION
- Asynchronous

LANGUAGES/DRIVERS SUPPORTED
C C++ C# JAVA PYTHON

SQL SUPPORT
No

TRANSACTIONS SUPPORTED
Arbitrary multi-statement transactions spanning arbitrary nodes

CONSISTENCY MODEL
Strong consistency

INDEXING CAPABILITIES
Via Solr

AUTO-SHARDING
Yes

FULL TEXT SEARCH
Via open source libraries

OPEN SOURCE

FULL PROFILE LINK
dzone.com/r/rNMp

TWITTER @HBase

WEBSITE hbase.apache.org

IBM DB2 IBM

DATABASE RELATIONAL

IBM DB2 is a database for Linux, Windows, UNIX, and z/OS, offering high-performing storage and analytics capabilities for distributed systems.

REPLICATION
- Synchronous
- Asynchronous

LANGUAGES/DRIVERS SUPPORTED
C C++ JAVA PHP RUBY

SQL SUPPORT
Full ANSI SQL

TRANSACTIONS SUPPORTED
Arbitrary multi-statement transactions spanning arbitrary nodes

CONSISTENCY MODEL
Eventual consistency

INDEXING CAPABILITIES
Simple indexes

AUTO-SHARDING
Yes

FULL TEXT SEARCH
Yes

PROPRIETARY

FULL PROFILE LINK
dzone.com/r/QQ4k

TWITTER @ibm

WEBSITE ibm.com

InfiniteGraph OBJECTIVITY

DATABASE GRAPH

InfiniteGraph is a database providing scalability of data and processing with performance in a distributed environment.

REPLICATION
- Synchronous

LANGUAGES/DRIVERS SUPPORTED
C++ JAVA PYTHON C#

SQL SUPPORT
Limited subset

TRANSACTIONS SUPPORTED
Arbitrary multi-statement transactions spanning arbitrary nodes

CONSISTENCY MODEL
Strong consistency

INDEXING CAPABILITIES
No

AUTO-SHARDING
No

FULL TEXT SEARCH
Via Lucene

PROPRIETARY

FULL PROFILE LINK
dzone.com/r/Qa4k

TWITTER @objectivitydb

WEBSITE objectivity.com

Informatica BDE INFORMATICA

DATA PLATFORM DATA MANAGEMENT, DATA INTEGRATION

BDE provides a safe, efficient way to integrate & process all types of data on Hadoop at any scale, without having to learn Hadoop.

HADOOP SUPPORT
Hadoop integrations available

DB INTEGRATIONS
ORACLE SQL SERVER IBM DB2 SAP HANA MONGODB

INTEGRATION SUPPORT
- ETL
- ELT

BUSINESS MODELER
Yes

BUILT-IN IDE
Informatica Developer

CLOUD HOSTING
SaaS, On-Premise

STREAM PROCESSING
Yes

MAPREDUCE JOB DESIGNER
Yes

PROPRIETARY

FULL PROFILE LINK
dzone.com/r/rWNp

TWITTER @INFA_BD

WEBSITE informatica.com

MapR MAPR TECHNOLOGIES

RESEARCH PARTNER

DATA PLATFORM DATA MANAGEMENT, DATA INTEGRATION, ANALYTICS

MapR's platform features true built-in enterprise-grade features like high availability, full multi-tenancy, integrated optimized NoSQL, and full NFS access.

HADOOP SUPPORT
Built on Hadoop

DB INTEGRATIONS
ORACLE SQL SERVER IBM DB2 SAP HANA MONGODB

INTEGRATION SUPPORT
- ETL

STATISTICAL LANGUAGES
- R
- SAS

BUILT-IN IDE
No IDE

CLOUD HOSTING
SaaS, PaaS, On-Premise

STREAM PROCESSING
Yes

MAPREDUCE JOB DESIGNER
No

PROPRIETARY & OPEN SOURCE

FULL PROFILE LINK
dzone.com/r/7u6h

TWITTER @mapr

WEBSITE mapr.com

MemSQL MEMSQL

DATABASE IN-MEMORY, RELATIONAL, COLUMNAR

MemSQL has fast data load and query execution during mixed OLTP/OLAP workloads due to compiled query plans and lock-free data structures.

REPLICATION
- Synchronous
- Asynchronous

LANGUAGES/DRIVERS SUPPORTED
C C++ JAVA NODE.JS RUBY

SQL SUPPORT
Full ANSI SQL

TRANSACTIONS SUPPORTED
Arbitrary multi-statement transactions spanning arbitrary nodes

CONSISTENCY MODEL
Strong consistency

INDEXING CAPABILITIES
Rich query language

AUTO-SHARDING
Yes

FULL TEXT SEARCH
No

PROPRIETARY

FULL PROFILE LINK
dzone.com/r/TLv6

TWITTER @memsql

WEBSITE memsql.com

MongoDB Enterprise MONGODB, INC.

DATABASE DOCUMENT

MongoDB blends linear scalability and schema flexibility with the rich query and indexing functionality of an RDBMS.

REPLICATION
- Synchronous
- Asynchronous

LANGUAGES/DRIVERS SUPPORTED
C C++ JAVA NODE.JS RUBY

SQL SUPPORT
No

TRANSACTIONS SUPPORTED
"Compare-and-set" type transactions

CONSISTENCY MODEL
Strong consistency

INDEXING CAPABILITIES
Rich query language

AUTO-SHARDING
Yes

FULL TEXT SEARCH
Yes

OPEN SOURCE

FULL PROFILE LINK
dzone.com/r/w9dP

TWITTER @mongoDBinc

WEBSITE mongodb.com

Neo4j NEO TECHNOLOGY

DATABASE GRAPH

Neo4j is a schema-optional, ACID, scalable graph database with minutes-to-milliseconds performance over RDBMS and NOSQL.

REPLICATION
- Asynchronous

LANGUAGES/DRIVERS SUPPORTED
C C++ JAVA NODE.JS RUBY

SQL SUPPORT
No

TRANSACTIONS SUPPORTED
Arbitrary multi-statement transactions spanning arbitrary nodes

CONSISTENCY MODEL
Eventual consistency

INDEXING CAPABILITIES
Rich query language

AUTO-SHARDING
No

FULL TEXT SEARCH
Via Lucene

OPEN SOURCE

FULL PROFILE LINK
dzone.com/r/rdNp

TWITTER @neo4j

WEBSITE neo4j.com

New Relic Insights NEW RELIC

RESEARCH PARTNER

DATA PLATFORM OPERATIONAL INTELLIGENCE

New Relic's analytics platform provides real-time data collection and querying capabilities based on closed-source database technologies.

HADOOP SUPPORT
No Hadoop Support

DB INTEGRATIONS
NEW RELIC QUERY LANGUAGE

INTEGRATION SUPPORT
- None

STATISTICAL LANGUAGES
- None

BUILT-IN IDE
No IDE

CLOUD HOSTING
SaaS

STREAM PROCESSING
Yes

MAPREDUCE JOB DESIGNER
No

PROPRIETARY

FULL PROFILE LINK
dzone.com/r/pdHQ

TWITTER @newrelic

WEBSITE newrelic.com/insights

NuoDB NUODB

DATABASE IN-MEMORY, DISTRIBUTED RELATIONAL OBJECT STORE

NuoDB is a distributed, peer-to-peer database with elastic scaling capabilities to store and analyze big data.

REPLICATION
- Asynchronous

LANGUAGES/DRIVERS SUPPORTED
C C++ JAVA NODE.JS RUBY

SQL SUPPORT
Full ANSI SQL

TRANSACTIONS SUPPORTED
Arbitrary multi-statement transactions on a single node

CONSISTENCY MODEL
Strong consistency

INDEXING CAPABILITIES
Rich query language

AUTO-SHARDING
No

FULL TEXT SEARCH
No

PROPRIETARY

FULL PROFILE LINK
dzone.com/r/NdpH

TWITTER @nuodb

WEBSITE nuodb.com

Oracle Database ORACLE

DATABASE RELATIONAL, DOCUMENT, COLUMNAR, GRAPH

Oracle features a multitenant architecture, automatic data optimization, defense-in-depth security, high availability, and failure protection.

REPLICATION
- Synchronous
- Asynchronous

LANGUAGES/DRIVERS SUPPORTED
C C++ JAVA NODE.JS RUBY

SQL SUPPORT
Full ANSI SQL

TRANSACTIONS SUPPORTED
Arbitrary multi-statement transactions spanning arbitrary nodes

CONSISTENCY MODEL
Strong consistency

INDEXING CAPABILITIES
Rich query language

AUTO-SHARDING
No

FULL TEXT SEARCH
Yes

PROPRIETARY

FULL PROFILE LINK
dzone.com/r/ttUJ

TWITTER @OracleDatabase

WEBSITE oracle.com/database

Pivotal Big Data Suite PIVOTAL

DATA PLATFORM DATA MANAGEMENT, DATA INTEGRATION, ANALYTICS

Pivotal Big Data Suite delivers HAWQ, GemFire, Greenplum, and Hadoop in a single integrated analytics and integration platform.

HADOOP SUPPORT
Built on Hadoop

DB INTEGRATIONS
MYSQL HBASE CASSANDRA

INTEGRATION SUPPORT
- ETL
- ELT

BUSINESS MODELER
No

BUILT-IN IDE
Spring Tools Suite

CLOUD HOSTING
SaaS, PaaS

STREAM PROCESSING
Yes

MAPREDUCE JOB DESIGNER
No

PROPRIETARY

FULL PROFILE LINK dzone.com/r/wdrP

TWITTER @Pivotal **WEBSITE** pivotal.io

Postgres Plus ENTERPRISEDB

DATABASE RELATIONAL, DOCUMENT, KEY-VALUE

Postgres Plus Advanced Server advances PostgreSQL with enterprise-grade performance, security and manageability enhancements.

REPLICATION
- Synchronous
- Asynchronous

SQL SUPPORT
Full ANSI SQL

CONSISTENCY MODEL
Strong consistency

AUTO-SHARDING
No

LANGUAGES/DRIVERS SUPPORTED
C++ JAVA NODE.JS RUBY PYTHON

TRANSACTIONS SUPPORTED
Arbitrary multi-statement transactions on a single node

INDEXING CAPABILITIES
Rich query language

FULL TEXT SEARCH
Yes

OPEN SOURCE

FULL PROFILE LINK dzone.com/r/MdaA

TWITTER @enterprisedb **WEBSITE** enterprisedb.com

RavenDB HIBERNATING RHINOS

DATABASE DOCUMENT

RavenDB is a self-optimizing ACID database with multi master replication, dynamic queries, and strong support for reporting.

REPLICATION
- Asynchronous

SQL SUPPORT
No

CONSISTENCY MODEL
Strong consistency

AUTO-SHARDING
Yes

LANGUAGES/DRIVERS SUPPORTED
JAVA NODE.JS PYTHON PHP SCALA

TRANSACTIONS SUPPORTED
Arbitrary multi-statement transactions spanning arbitrary nodes

INDEXING CAPABILITIES
Rich query language

FULL TEXT SEARCH
Yes

OPEN SOURCE

FULL PROFILE LINK dzone.com/r/U4Jf

TWITTER @ravendb **WEBSITE** ravendb.net

Redis Cloud REDIS LABS

DATABASE IN-MEMORY, UNORDERED KEY-VALUE

Redis Cloud is an infinitely scalable, highly available, and top performing hosted Redis service.

REPLICATION
- Synchronous
- Asynchronous

SQL SUPPORT
No

CONSISTENCY MODEL
Strong consistency

AUTO-SHARDING
Yes

LANGUAGES/DRIVERS SUPPORTED
C C++ JAVA NODE.JS RUBY

TRANSACTIONS SUPPORTED
Arbitrary multi-statement transactions on a single node

INDEXING CAPABILITIES
No

FULL TEXT SEARCH
No

PROPRIETARY

FULL PROFILE LINK dzone.com/r/3sjG

TWITTER @redislabsinc **WEBSITE** redislabs.com

Riak BASHO

DATABASE UNORDERED KEY-VALUE

Riak excels at high write volumes using a straight key value store and vector clocks to provide the most flexibility for data storage.

REPLICATION

- Synchronous

SQL SUPPORT

No

CONSISTENCY MODEL

Eventual consistency

AUTO-SHARDING

Yes

LANGUAGES/DRIVERS SUPPORTED

C C++ JAVA NODE.JS RUBY

TRANSACTIONS SUPPORTED

Riak 2.0 adds single object compare-and-set for strongly consistent bucket types

INDEXING CAPABILITIES

Rich query language

FULL TEXT SEARCH

Yes

OPEN SOURCE

FULL PROFILE LINK dzone.com/r/r9Np

TWITTER @basho

WEBSITE basho.com

SAP HANA SAP AG

DATA PLATFORM DATA MANAGEMENT, ANALYTICS, DATA INTEGRATION

SAP HANA is a platform with a columnar database, an app and web server, as well as predictive, spatial, graph, and text processing libraries and engines.

HADOOP SUPPORT

Hadoop integrations available

INTEGRATION SUPPORT

- ETL
- ELT

BUILT-IN IDE

No IDE

STREAM PROCESSING

Yes

DB INTEGRATIONS

SAP HANA IBM DB2 CASSANDRA

STATISTICAL LANGUAGES

- R

CLOUD HOSTING

SaaS, PaaS, On-Premise

MAPREDUCE JOB DESIGNER

Job Designer

PROPRIETARY

FULL PROFILE LINK dzone.com/r/Uksf

TWITTER @SAPInMemory

WEBSITE sap.com

SAS Platform SAS

DATA PLATFORM DATA MANAGEMENT, DATA INTEGRATION, ANALYTICS

SAS provides advanced analytics, data management, BI & visualization products for data scientists, business analysts and IT.

HADOOP SUPPORT

Built on Hadoop

INTEGRATION SUPPORT

- ETL
- ELT

BUILT-IN IDE

SAS AppDev Studio

STREAM PROCESSING

Yes

DB INTEGRATIONS

ORACLE IBM DB2 SAP HANA TERADATA NETEZZA

STATISTICAL LANGUAGES

- SAS
- R
- PMML

CLOUD HOSTING

SaaS, PaaS, On-Premise

MAPREDUCE JOB DESIGNER

Yes

PROPRIETARY

FULL PROFILE LINK dzone.com/r/L7vu

TWITTER @SASsoftware

WEBSITE sas.com

ScaleOut hServer SCALEOUT SOFTWARE

DATABASE IN-MEMORY, KEY-VALUE

Runs Hadoop MapReduce applications continuously on live, fast-changing, memory-based data with low latency and high scalability.

REPLICATION

- Synchronous
- Asynchronous

SQL SUPPORT

Limited subset

CONSISTENCY MODEL

Strong consistency

AUTO-SHARDING

Yes

LANGUAGES/DRIVERS SUPPORTED

C C++ C# JAVA REST

TRANSACTIONS SUPPORTED

"Compare-and-set" type transactions

INDEXING CAPABILITIES

Rich query language

FULL TEXT SEARCH

No

PROPRIETARY

FULL PROFILE LINK dzone.com/r/d9PM

TWITTER @ScaleOut_Inc

WEBSITE scaleoutsoftware.com

Splunk Enterprise SPLUNK

DATA PLATFORM | OPERATIONAL INTELLIGENCE

Splunk is a fully-integrated platform that supports both real-time and batch search, and treats time series data as a first-class construct.

HADOOP SUPPORT Hadoop integrations available	DB INTEGRATIONS ORACE SQL SERVER MONGODB CASSANDRA HBASE
INTEGRATION SUPPORT • None	STATISTICAL LANGUAGES • None
BUILT-IN IDE No IDE	CLOUD HOSTING SaaS, On-Premise
STREAM PROCESSING Yes	MAPREDUCE JOB DESIGNER No
PROPRIETARY	FULL PROFILE LINK dzone.com/r/jsGR
TWITTER @splunkdev	WEBSITE splunk.com/product

Spring XD PIVOTAL

DATA PLATFORM | DATA MANAGEMENT, DATA INTEGRATION, ANALYTICS

Spring XD offers a unified, distributed, highly available, and extensible runtime for Big Data ingestion, analytics, batch processing, and data export.

HADOOP SUPPORT Hadoop Integrations Available	DB INTEGRATIONS ORACE SQL SERVER IBM DB2 SAP HANA MYSQL
INTEGRATION SUPPORT • None	BUSNINESS MODELER No
BUILT-IN IDE Spring Tools Suite	CLOUD HOSTING SaaS, PaaS, On-Premise
STREAM PROCESSING Yes	MAPREDUCE JOB DESIGNER No
OPEN SOURCE	FULL PROFILE LINK dzone.com/r/dPNM
TWITTER @Pivotal	WEBSITE pivotal.io

SQL Server MICROSOFT

DATABASE | IN-MEMORY, RELATIONAL

SQL Server is considered the de facto database for .NET development. Its ecosystem is connected to numerous .NET technologies.

REPLICATION • Synchronous • Asynchronous	LANGUAGES/DRIVERS SUPPORTED C# JAVA RUBY PHP VISUAL BASIC
SQL SUPPORT Full ANSI SQL	TRANSACTIONS SUPPORTED Arbitrary multi-statement transactions spanning arbitrary nodes
CONSISTENCY MODEL Tunable per database	INDEXING CAPABILITIES Rich query language
AUTO-SHARDING No	FULL TEXT SEARCH Yes
PROPRIETARY	FULL PROFILE LINK dzone.com/r/CKLT
TWITTER @Microsoft	WEBSITE microsoft.com

Sumo Logic SUMO LOGIC

DATA PLATFORM | OPERATIONAL INTELLIGENCE

Sumo Logic features machine-learning based analytics, elastic log processing, real-time dashboards, and multi-tenant SaaS architecture.

HADOOP SUPPORT Hadoop integrations available	DB INTEGRATIONS ORACLE SQL SERVER IBM DB2 MYSQL
INTEGRATION SUPPORT • None	STATISTICAL LANGUAGES • None
BUILT-IN IDE No IDE	CLOUD HOSTING SaaS
STREAM PROCESSING Yes	MAPREDUCE JOB DESIGNER No
PROPRIETARY	FULL PROFILE LINK dzone.com/r/Xx9z
TWITTER @sumologic	WEBSITE sumologic.com

Tableau TABLEAU SOFTWARE

DATA PLATFORM DATA MANAGEMENT, ANALYTICS

Tableau has a highly-rated user experience with intuitive visual data exploration tools that make ordinary business users into data experts.

HADOOP SUPPORT
Hadoop integrations available

DB INTEGRATIONS
ORACLE SQL SERVER MYSQL POSTGRESQL TERADATA

INTEGRATION SUPPORT
- None

STATISTICAL LANGUAGES
- R

BUILT-IN IDE
No IDE

CLOUD HOSTING
SaaS, On-Premise

STREAM PROCESSING
Yes

MAPREDUCE JOB DESIGNER
Yes

PROPRIETARY

FULL PROFILE LINK
dzone.com/r/NNpH

TWITTER @tableau

WEBSITE tableausoftware.com

Tibco Spotfire TIBCO SOFTWARE

DATA PLATFORM DATA MANAGEMENT, DATA INTEGRATION, ANALYTICS

Tibco's strength lies in data discovery with real-time and bidirectional integration with business processes in an easy-to-use interface.

HADOOP SUPPORT
Hadoop integrations available

DB INTEGRATIONS
ORACLE SQL SERVER IBM DB2 SAP HANA MYSQL

INTEGRATION SUPPORT
- None

STATISTICAL LANGUAGES
- R
- S+

BUILT-IN IDE
Tibco Spotfire S+

CLOUD HOSTING
SaaS, On-Premise

STREAM PROCESSING
Yes

MAPREDUCE JOB DESIGNER
Yes

PROPRIETARY

FULL PROFILE LINK
dzone.com/r/99zr

TWITTER @TIBCO

WEBSITE spotfire.tibco.com

Teradata Aster TERADATA

DATABASE RELATIONAL

Teradata's strengths and maturity lie in the data warehouse market. Their focus is also on Hadoop capabilities and multistructured formats.

REPLICATION
- Synchronous
- Asynchronous

LANGUAGES/DRIVERS SUPPORTED
JAVA C#

SQL SUPPORT
Full ANSI SQL

TRANSACTIONS SUPPORTED
Arbitrary multi-statement transactions on a single node

CONSISTENCY MODEL
Strong consistency

INDEXING CAPABILITIES
Rich query language

AUTO-SHARDING
No

FULL TEXT SEARCH
Yes

PROPRIETARY

FULL PROFILE LINK
dzone.com/r/jGCR

TWITTER @Teradata

WEBSITE teradata.com

Vertica HEWLETT-PACKARD

DATA PLATFORM DATA MANAGEMENT, ANALYTICS

Vertica thrives at petabyte-scale, offers complete SQL on Hadoop, and queries have been found to run 50-1,000x faster with Vertica than on legacy solutions.

HADOOP SUPPORT
Hadoop integrations available

DB INTEGRATIONS
HP VERTICA

INTEGRATION SUPPORT
- ETL
- ELT

STATISTICAL LANGUAGES
- R

BUILT-IN IDE
No IDE

CLOUD HOSTING
SaaS, PaaS, On-Premise

STREAM PROCESSING
Yes

MAPREDUCE JOB DESIGNER
No

PROPRIETARY

FULL PROFILE LINK
dzone.com/r/taUJ

TWITTER @HPVertica

WEBSITE vertica.com

GLOSSARY OF TERMS

A

ACID (ATOMICITY, CONSISTENCY, ISOLATION, DURABILITY): A term that refers to the model properties of database transactions, traditionally used for SQL databases.

B

BASE (BASIC AVAILABILITY, SOFT STATE, EVENTUAL CONSISTENCY): A term that refers to the model properties of database transactions, specifically for NoSQL databases needing to manage unstructured data.

BATCH PROCESSING: The execution of a series of programs (jobs) that process sets of records as complete units (batches). This method is commonly used for processing large sets of data offline for fast analysis later.

BIG DATA: The entire process of collecting, managing, and analyzing datasets too massive to be handled efficiently by traditional database tools and methods; the industry challenge posed by the management of massive structured and unstructured datasets.

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

C

COLUMN STORE: A high-availability database deployed on multiple datacenters that is primarily used for logging continuous streams of data with few consistency guarantees.

COMPLEX EVENT PROCESSING: An organizational process for collecting data from multiple streams for the purpose of analysis and planning.

D

DATA ANALYTICS: The process of harvesting, managing, and analyzing large sets of data to identify patterns and insights.

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

DATA MINING: The process of patterns in large sets of data and transforming that information into an understandable format.

DATA MUNGING: The process of converting raw mapping data into other formats using automated tools to create visualizations, aggregations, and models.

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.

DATABASE MANAGEMENT SYSTEM (DBMS): A suite of software and tools that manages data between the end user and the database.

DOCUMENT STORE: A type of database that aggregates data from documents rather than defined tables and is used to present document data in a searchable form.

E

EXTRACT LOAD TRANSFORM (ELT): The process of preparing integrated data in a database to be used by downstream users.

EXTRACT TRANSFORM LOAD (ETL): The process of extracting, transforming, and loading data during the data storage process; often used to integrate data from multiple sources.

EVENT-STREAM PROCESSING (ESP): An organizational process for handling data that includes event visualization, event processing languages, and event-driven middleware.

EVENTUAL CONSISTENCY: The idea that databases conforming to the BASE model will contain data that becomes consistent over time.

F

FAULT TOLERANCE: A system's ability to respond to hardware or software failure without disrupting other systems.

G

GRAPH STORE: A type of database used for handling entities that have a large number of relationships, such as social graphs, tag systems, or any link-rich domain; it is also often used for routing and location services.

H

HADOOP: An Apache Software Foundation framework developed specifically for high-scalability, data-intensive, distributed computing.

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

K

KEY-VALUE STORE: A type of database that stores data in simple key-value pairs. They are used for handling lots of small, continuous, and potentially volatile reads and writes.

N

NewSQL: A shorthand descriptor for relational database systems that provide horizontal scalability and performance on par with NoSQL systems.

NoSQL: A class of database systems that incorporate other means of querying outside of traditional SQL and do not follow standard relational database rules.

M

MACHINE LEARNING: An area of study in artificial intelligence (AI) that attempts to mimic human intelligence by enabling computers to interpret situations through observation and analysis.

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

MASSIVELY PARALLEL PROCESSING (MPP): The strategy of pairing independent database processors together with a messaging interface to create cooperative clusters of processors.

MESSAGE PASSING INTERFACE (MPI): A standardized messaging interface created to govern parallel computing systems.

O

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.

R

RELATIONAL DATABASE: A database that structures interrelated datasets in tables, records, and columns.

S

STRONG CONSISTENCY: A database concept that refers to the inability to commit transactions that violate a database's rules for data validity.

STRUCTURED QUERY LANGUAGE (SQL): A programming language designed for managing and manipulating data; used primarily in relational databases.

SYSTEM-ON-A-CHIP(SOC): An integrated chip that is comprised of electronic circuits of multiple computer components to create a complete device.

DZone | *smart content for tech professionals* | dzone.com

now hiring JAVA DEVELOPERS,

WEB DESIGNERS, FRONT-END/UI BUILDERS, AND OTHER SMART PEOPLE

DZone was recently named to the Inc. 5000 as one of the fastest growing companies in the US, and we are looking for talented people to help us continue our growth. With you on our team, hopefully we can move up to the Inc. 500 next year!

AMERICA'S FASTEST-GROWING Inc. 5000 PRIVATE COMPANIES

required SKILLS

JAVA DEVELOPERS: Excellent working knowledge of Java and Java Web Architectures. Skilled in back-end technologies like Spring, Hibernate, Lucene and SQL, as well as standard web technologies.

WEB DESIGNERS: "Live and breathe" the Web with superior creative and innovative problem-solving skills. Knowledge of key web technologies like HTML, CSS, Bootstrap, as well as Adobe Creative Suite.

FRONT-END/UI BUILDER: Passion for simple and beautiful designs to help the look and feel of our web and mobile products. Knowledge of standard web technologies like HTML, CSS, Bootstrap, LESS, and Javascript as well as Adobe Creative Suite.

why work AT DZONE?

Working at DZone sets you up with a meaningful career and the ability to see your hard work make a difference on a global scale.

- Work with other smart, ambitious people who are passionate about technology
- Ability to work in our Cary, NC headquarters or telecommute from anywhere around the world
- Flexible and fun startup environment
- Opportunity for personal growth and learning experiences through a variety projects (you won't be doing the same thing every day)
- Fantastic benefits package including your choice of several comprehensive medical, life, and disability plans
- Awesome perks like catered weekly lunch, XBox games, snacks, and beer on tap

about DZONE

DZone makes online content and resources for developers, tech professionals, and smart people everywhere.

Our website, DZone.com is visited by millions of tech pros from all over the world every month, and our free resources have been downloaded millions of times.

AnswerHub, our software platform for building online communities, is used by some of the most recognizable companies in the world including LinkedIn, eBay, Epic Games and Microsoft.

DZone

Check out **dzone.com/jobs** to learn more.