DZone

DZONE'S 2019 GUIDE TO

Big Data

BROUGHT TO YOU IN PARTICIPATION WITH

LexisNexis®

unravel™

Dear Reader,

This is DZone's sixth research guide on big data technologies, and we've seen several changes in the ecosystem since 2014. What originally began as a pressing need to store terabytes of data has evolved into a pressing need to effectively analyze the data that are collected from a variety of sources, with more and more being deployed to production or manufactured and sold every day. The challenge of analyzing that data has become greater in the years since. Finding the most relevant data in a sea of numbers is already difficult enough, but that same data needs to be sanitized and delivered to an analytics platform in a timely manner. The importance of transmitting that data becomes even more important as critical, life-altering data needs to be considered. The development of autonomous vehicles and robotics is creating a greater and greater demand for these kinds of analytics.

With the greater need to analyze a wide variety of big data being generated every second, developers, architects, and data scientists will have to learn the best practices to do so, and to learn the ecosystem of tools that exist to make these tasks easier. Older technologies, such as Python, are constantly being improved by the addition of frameworks that expand its reach beyond application development into data processing and analysis. Newer technologies, such as streaming data tools like Spark Streaming, are quickly appearing and dominating their particular niches as the market finds that they need to address the issues that arise from device data. In this guide, we will look at these and other technologies, such as Samza, Storm, Lua, and Kafka. We will also explore several scenarios that call for a variety of options. Twitter sentiment analysis, autonomous cars, and environmental sustainability are all use cases for big data and analytics.

We hope DZone's *2019 Guide to Big Data: Volume, Variety, and Velocity* helps you learn more about new trends in the big data ecosystem, gain insight into what fellow developers and architects are doing in the space, and gives you some new ideas of how to incorporate these into your work or home projects. Happy reading!

WRITTEN BY MATT WERNER
PUBLICATIONS COORDINATOR, DEVADA

Table of Contents

DZone is...

Executive Summary

BY KARA PHELPS
CONTENT & COMMUNITY MANAGER, DEVADA

The research group IDC predicted in 2017 that the world will create 163 zettabytes of data per year by 2025 — around 10 times the current rate of data creation. (For a bit of perspective, a zettabyte is one trillion gigabytes.) "Big data" is now reaching a scale that was unimaginable just a few years ago, and the technologies designed to handle it are beginning to mature. The world's vast reserves of data have enormous value for those with the skills and resources to properly mine it.

We asked 459 tech professionals to tell us about the challenges they face in working with big data, trends they're seeing, and which of the latest tools and frameworks they've found to be the most helpful. Here are a few of the stronger currents we noticed this year.

Python and R: Top Choices for Big Data

DATA

79 percent of survey respondents said they use the Python ecosystem for data science and machine learning, increasing from 68 percent in 2018. 51 percent said they use R, up from 47 percent last year. Beyond R and Python, 47 percent of this year's respondents said they use Spark, 46 percent reported they use TensorFlow, and 25 percent said they use the Java Machine Learning Library (Java-ML).

IMPLICATIONS

Python and R continue to dominate the landscape when it comes to data science and machine learning. Both languages also feature wide varieties of repositories, supportive open-source communities, and sophisticated ways to analyze, visualize, and predict outcomes. Other languages, libraries, and frameworks are also growing in their use cases and popularity as data scientists build new products and features for their organizations.

RECOMMENDATIONS

R code was built for statisticians and research scientists for data analysis, while Python is a multipurpose programming language that can easily be integrated with other applications. Both are powerful and flexible, and many data scientists use both.

Cleanliness Is Next to Data Science

DATA

71 percent of survey respondents named unsanitized data as their biggest data science challenge. That's an increase from 63 percent who called it their most difficult challenge in 2018. The percentage of those who called limited training or talent their biggest challenge decreased by a negligible amount, from 41 percent to 40 percent. "Project timelines" also decreased as a number-one hurdle, from 38 percent in 2018 to just 31 percent this year. Those who said "no defined standard practice or methodology" was their biggest data science challenge increased from 27 percent to 30 percent.

IMPLICATIONS

Data cleansing and validation are crucial, yet time-consuming, steps in data analysis. At the same time, the return on investment for analytics projects focuses on getting results quickly. In the survey, 41 percent of respondents said they measure ROI in data analytics by the speed at which decisions are made, and 36 percent said they measure it by the speed at which people get access to the data they need. This conflict can set up challenges for data scientists.

RECOMMENDATIONS

As the number of data sources increases on all fronts, maintaining high data quality becomes a bigger and bigger task. It's important to account for data cleansing in project timelines — and when calculating the ROI of an analytics implementation.

"Spark" a Streaming Explosion

DATA

When asked which computational frameworks they use for streaming data processing, 49 percent of survey respondents cited Spark Streaming, compared to 35 percent of survey respondents last year — a 14 percent increase. While 49 percent of last year's survey respondents called the question "not applicable," just 32 percent of this year's respondents did so.

IMPLICATIONS

It's always been important to assimilate new data as soon as possible, but real-time (streaming) data analytics is notoriously difficult. More teams decided to take on streaming data processing this year, however, as the technology and the teams themselves mature. Spark Streaming has seen rapid adoption since Apache Spark added it in 2013, at least partially because it offers a single framework for diverse aspects of data processing. As the need for real-time data processing grows for more organizations, Spark Streaming is a popular choice.

RECOMMENDATIONS

To get started, check out DZone's Refcard on Apache Spark, which also covers Spark Streaming. It's a cheat sheet to learn more about Spark Streaming's core abstraction, Discretized Streams (DStreams), as well as an overview of the open-source Spark ecosystem.

DZONE.COM/GUIDES

Key Research Findings

BY JORDAN BAKER
CONTENT COORDINATOR, DEVADA

Demographics

For this year's big data survey, we received 459 responses with a 78% completion rating. Based on this response rate, we have calculated the margin of error for this survey to be 5%. Below are some of the basic demographics of the respondents.

- Respondents reported working in four main industry verticals:
  - 17% work in finance/banking.
  - 17% work for software vendors.
  - 11% work in consulting firms.
  - 8% work in e-commerce/internet organizations.

37% work for organizations headquartered in the United States, 34% work for Europe-based organizations, and 5% work for companies HQ-ed in South Central Asia.

- A majority of survey-takers work in enterprise-level organizations:
  - 29% work in organizations sized 500-9,999.
  - 22% work in organizations sized 10,000+.
  - 18% work in organizations sized 100-499.

Over half (58%) of respondents have 15+ years of experience in IT, 22% have 10-15 years of experience, and 12% have been in the industry for six to nine years.

- Respondents fill three main roles in their organization:
  - 33% are developers/engineers.
  - 22% are architects.
  - 22% are developer team leads.
- Respondents work to develop three main types of software:
  - 80% create web applications/services.
  - 45% develop enterprise business applications.
  - 22% work on native mobile applications.

Big Data on the Rise

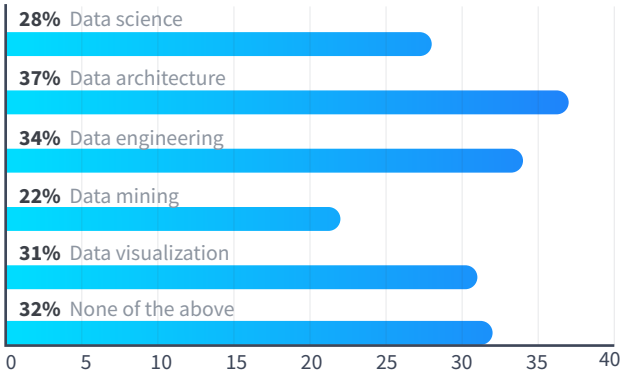
Over the course of the past year, our respondents have reported becoming much more data-driven. When asked what types of big data they tend to work with, 78% reported working with large volumes of data, 51% with a large variety of data, and 42% with data at high velocity. While the year-over-year change in the percentage of respondents working with large volumes of data and high velocity data fell within the margin of error for this report (a 4% increase and 2% decrease, respectively), these numbers, nonetheless, remain rather high. And those working with a large variety of data increased 7% year-over-year. Additionally, respondents' experience in all areas of big data increased considerably from our 2018 big data survey. Here's a quick breakdown of the percentages of respondents who had experience with a certain topic, comparing our 2018 survey data to the 2019 results:

- Data architecture
  - 2018: 26%
  - 2019: 37%
- Data engineering
  - 2018: 21%
  - 2019: 34%
- Data visualization
  - 2018: 21%
  - 2019: 31%
- Data science
  - 2018: 19%
  - 2019: 28%
- Data mining
  - 2018: 17%
  - 2019: 22%

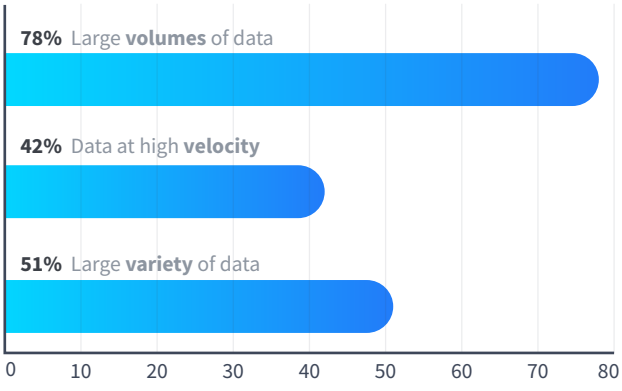
In addition to these impressive increases, the percentage of respondents

SURVEY RESPONSES

In which of the following areas of big data do you have experience?



What kind of big data do you work with?



reporting to have experience in none of those big data sub-fields fell from 45% in 2018 to 32% in 2019.

Given this increased interest in and experience with big data practices, it comes as no surprise that the adoption rates for big data-focused languages and frameworks also saw an increase over last year's survey. In 2018, 68% of respondents reported using Python for data science and machine learning; in this year's survey, 79% of respondents reported using Python. Spark and TensorFlow adoption also increased by 6%, bringing them to a 47% and 46% use rate, respectively, among our survey-takers. And, while we didn't see a dramatic increase over last year's survey, 51% of respondents told us they use R for the data science and machine learning projects.

For the rest of this report, we'll look at the various processes associated with each step of the big data pipeline (ingestion, management, and analysis), and see how they've changed over the past year.

Ingesting Data as High Velocity

When we asked respondents what data types give them the most issues regarding data velocity, two types saw noticeable increases over last year: relational (flat tables) and event data. In 2018, 33% of respondents reported relational data as an issue with regards to velocity; this year, that rose to 38%. For event logs, we saw the percentage of respondents reporting this data type as an issue go from 23% to 30%. Interestingly, relational data types seem to be a far bigger issue for users of R than for Python developers. Among those who use Python for data science, only 8% reported relational data types to be an issue when it came to data velocity. 30% of R users, however, told us they've had problems with relational data.

We also asked respondents which data sources gave them trouble when dealing with high-velocity data. Two of the issues reported fell drastically from our 2018 survey. The number of survey-takers reporting server logs as an issue fell by 10%, and those reporting user-generated data fell from 39% in 2018 to 20% in this year's survey. Despite these positive trends, respondents who said files (i.e. documents, media, etc.) give them trouble rose from 26% last year to 36%.

The tools and frameworks that data professionals and developers use to deal with data ingestions processes also witnessed interesting fluctuations over the past year. To perform data ingestion, 66% of survey-takers reported using Apache Kafka, up from 61% last year. While Kafka has been the most popular data ingestion framework for a while now, its popularity only continues to climb. For streaming data processing, Spark Streaming came out on top, with 49% of respondents telling us they use this framework (a 14% increase over last year). For performing data serialization processes, however, respondents were split between two popular choices. 36% told us they work with Avro (up from 18% in 2018) and 30% reported using Parquet (also up from 18% in 2018).

Managing a Large Volume of Data

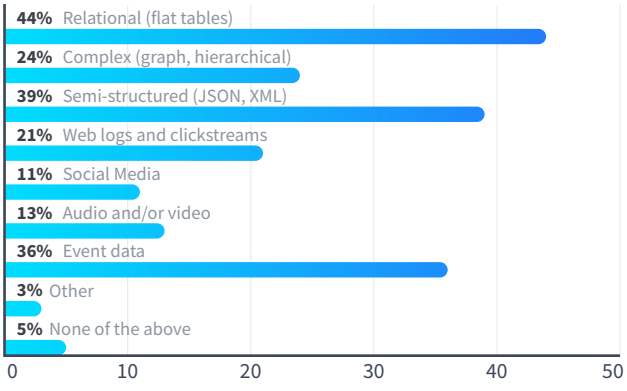
The basis of any data management plan is data storage. According to our respondents, there is a shift going on from cloud-based solutions to on-premise and hybrid solutions. 29% of respondents reported that their data typically resided in the cloud (down 10% from 2018), 31% told us they use a hybrid solution (up 7% over 2018's report), and 40% use on-premise data storage (another 7% year-over-year increase). In terms of the actual databases used to house this data, MySQL proved the most popular in both production (51%) and non-production (61%) environments, though its year-over-year adoption rate stayed rather static. PostgreSQL could be an interesting database to keep an eye on in the coming year, as its adoption rose in both production (42% in 2018 to 47% in 2019) and non-production (40% in 2018 to 48% in 2019) environments.

For filing big datasets, a vast majority of respondents told us they prefer the Hadoop Distributed Files System (HDFS). In fact, 80% of survey-takers reported using HDFS as their big data file system. While this large of a majority among respondents is impressive in its own right, HDFS also saw a 16% increase in adoption over our 2018 Big Data survey. The second most popular response to this question, Parquet, had a 36% adoption rate in our 2019 survey, up from 17% last year. Interestingly, even the least popular of the file systems reported, (O)RC File, saw an 11% year-over-year increase, rising to a 17% adoption rate.

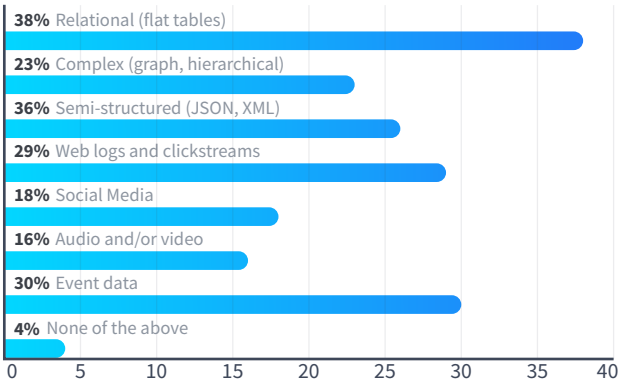
We also asked respondents about the issues they encounter when dealing with such large volumes of data. It turns out that normal files (such as documents, media files, etc.) cause the most headaches, with 49% of respondents selecting this option. Server logs also proved a popular answer, garnering 42% of responses. Data collected from IoT devices, however, saw the largest increase in developer frustrations. In 2018, 20% of respondents reported data from sensors or remote hardware as an issue;

SURVEY RESPONSES

What data types give you the most issues regarding the volume of data?



What data types give you the most issues regarding the velocity of data?



this year, 32% of survey-takers reported this type of data as a pain point. Surprisingly, despite user-generated data (i.e. social media, games, blogs, etc.) being one of the largest means of creating and ingesting new data, the difficulty this type of data gives to developers and data scientist seems to be decreasing. In 2018, 33% of respondents said user-generated data was a pain point in their big data operations; in 2019, this fell to 20%.

The types of data that gives developers issue when it comes to large volumes of data also witnessed a good deal of variability over last year. The data type that, according to respondents, causes that most issues — relational data — fell by 8%. Despite this decrease, it still registered 44% of respondents' votes. Event data also underwent a large swing, only in the opposite direction. In our 2018 survey, 25% of respondents said they had issues with event data; in 2019, this number rose to 36%. This increase in the number of respondents having trouble with event data is intriguing, given that user-generated data was reported as less of an issue than last year, yet much of the event data there is to be collected can be categorized as user-generated.

Analyzing a Variety of Data

Data mining is one of the most effective ways to sort through the immense variety of data an organization takes in. The two most popular tools for working with data wining, were, in fact, languages, specifically Python and R. Unsurprisingly, Python proved the most popular tool for data mining operations, garnering 80% of responses. Over half of respondents (51%) also selected R as a preferred data mining tool. Both of these numbers are fairly significant increases over our 2018 survey. Last year, 62% of respondents reported using Python and 41% said they used R for data mining.

One of the reasons these languages prove so popular in the data science community is the ability that their syntax gives to data professionals and developers to write complex algorithms. Interestingly, despite the extreme fluctuations we've covered in this report thus far, the preferred data mining algorithms stayed relatively stable year-over-year. Classification algorithms remained on top, selected by 62% of respondents, followed by clustering (61%) and regressions (52%). The only data mining algorithm which underwent a marked increase in adoption rate was times series algorithms. In 2018, 41% of respondents said they use time series algorithms in their data mining, which grew to 48% in this year's survey.

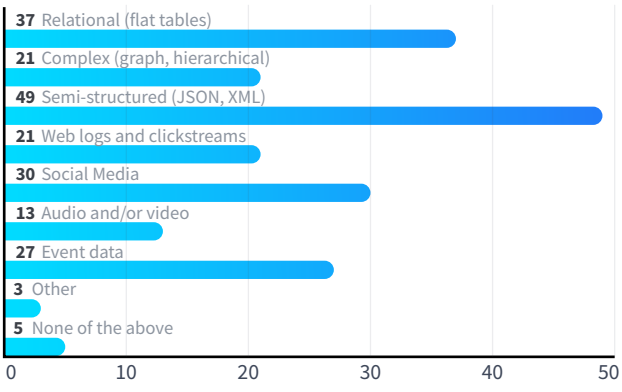
Among those respondents who work with data mining, 66% said that unsanitized data presents the largest challenge, up from 51% in 2018. This echoes the overall survey population, where 71% told us that data mining is the biggest issue in data science in general. Other important speedbumps in the data mining process that respondents delineated were the variety of data from different sources (65%) and a lack of knowledge or training (47%).

Given that well over half of all respondents reported that data variety is an issue, let's take a moment to examine the specific difficulties that come with highly variable datasets. When we asked survey-takers what data sources give them the most issues regarding the variety of data available, 52% said files (i.e. documents, media, etc.); 36% told us user-generated data; 25% reported ERP, CRM, or data from other enterprise systems; and 21% said sensor or remote hardware data. Of these five answers, only one saw any significant change over last year's survey (all the others falling within the margin of error for this report). When compared to 2018, respondents who are having issues with data from enterprise systems fell by 6%. While 40% told us that limited training and/or talent is an issue in the data science field, we nonetheless see an encouraging trend here, as the respondents who have issues with data sources due to data variety seems to have fallen over the course of the past year.

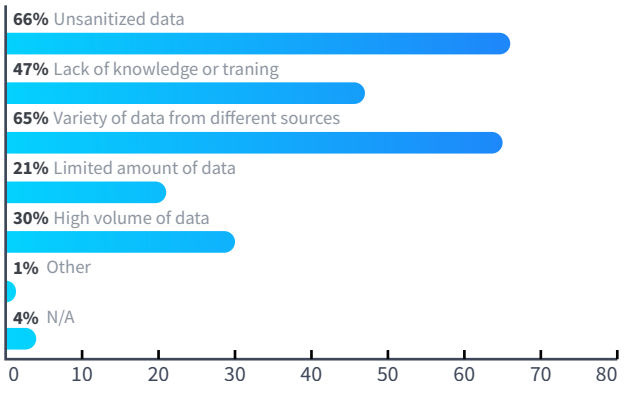
When we asked respondents about the data types that give them issues when it comes to the variety of data, we see a similar, positive trend. Respondents listed six main data types that prove challenging when working with data variety: semi-structured data, such as JSON or XML (49%); relation data (37%); data generated via social media (30%); event data (27%); complex data, such as graph or hierarchical data (21%); and web logs and clickstreams (21%). Of these six, only two saw a significant year-over-year change. In 2018, 41% of respondents reported complex data as an issue, while, as noted above, in 2019, only 21% of respondents reported this data type. This is a huge drop in the percentage of engineers and developers who have difficulty working with this data type. The second data type that underwent noticeable year-over-year change among our survey population was relational, which fell from 46% in 2018 to the aforementioned 37% in 2019. The fact that we are seeing stagnant or diminishing difficulty ratings across these six data types is, again, a positive sign. If developers and engineers continue to gain increased access to the proper training, expect these numbers to continue to fall.

SURVEY RESPONSES

What data types give you the most issues regarding the variety of data?



Which of the following challenges do you experience with data mining?



Navigating the Distributed Data Pipelines:

An Overview and Guide for Your Performance Management Strategy

QUICK VIEW

01. Modern data applications are being adopted by a wide range of verticals who depend on data for critical business decisions.

02. Data pipelines are complex and require full stack visibility of applications, users, data, and resources to optimize performance.

03. AI-driven automation is a critical capability to scaling Operations and Applications teams' ability to deal with the complexity and frequently of change in the data ecosystem.

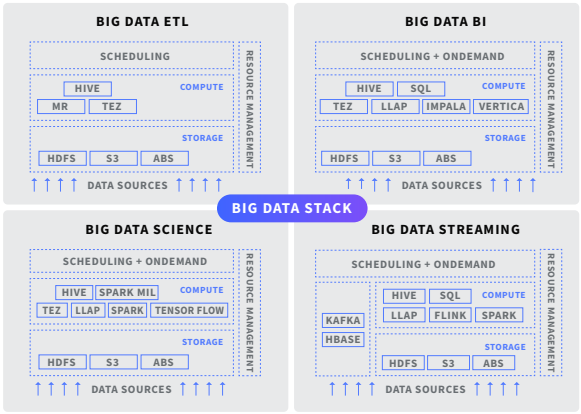
BY SHIVNATH BABU
CTO, UNRAVEL DATA SYSTEMS

There are more than 10,000 enterprises across the globe that rely on a data stack that is made up of multiple distributed systems. While these enterprises, which span a wide range of verticals — finance, healthcare, technology, and more — build applications on a distributed big data stack, some are not fully aware of the performance management challenges that often arise. This piece will provide an overview of what a modern big data stack looks like, then address the requirements at both the individual application level of these stacks (as well as holistic clusters and workloads), and explore what type of architecture can provide automated solutions for these complex environments.

The Big Data Stack and Operational Performance Requirements

Areas like health care, genomics, financial services, self-driving technology, government, and media are building mission critical applications in what's known as the big data stack. The big data stack is unique in that it is composed of multiple distributed systems. While every organization varies in how they deploy the technology, the big data stack in most enterprises goes through the following evolution:

- **ETL:** Storage systems, such as HDFS, S3, and Azure Blob Store (ABS), house the large volumes of structured, semi-structured, and unstructured data. Distributed processing engines, like MapReduce, come in for the extraction, cleaning, and transformation of the data.
- **BI:** SQL systems like Impala, Presto, LLAP, Drill, BigQuery, RedShift, or Azure SQL DW are added to the stack; sometimes alongside incumbent MPP SQL systems like Teradata and Vertica. Compared to the traditional MPP systems, the newer ones have been built to deal with data stored in a different distributed storage system like HDFS, S3, or ABS. These systems power the interactive SQL queries that are common in BI workloads.
- **Data science:** With the continued maturation of the big data stack, it starts bringing in more data science workloads that leverage machine learning and AI. This stage is usually when the Spark distributed system starts to be used more and more.
- **Data streaming:** Over time, enterprises begin to understand the importance of making data-driven decisions in near real-time, as well as how to overcome the challenges in implementing them. Usually at this point in the evolution, systems like Kafka, Cassandra, and HBase are added to the big data stack to support applications that ingest and process data in a continuous streaming fashion.



Evolution of the big data stack in an enterprise.

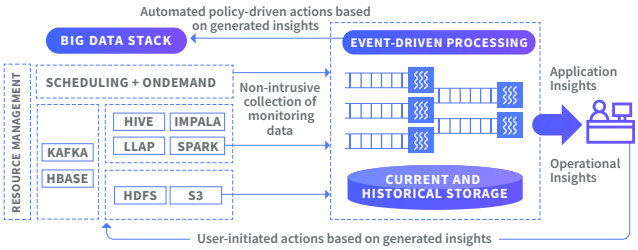
With so many enterprises worldwide running applications in production on a distributed big data stack (comprised of three or more distributed systems), performance challenges are no surprise. The stacks have many moving parts, which makes it very hard to get any answers in the event that something goes wrong. When there's a breakdown, organizations often find themselves scrambling to understand the following:

- **Failure:** What caused this application to fail, and how can I fix it?
- **Stuck:** This application seems to have made little progress in the last hour. Where is it stuck?
- **Runaway:** Will this application ever finish, or will it finish in a reasonable amount of time?
- **SLA:** Will this application meet its SLA?
- **Change:** Is the behavior (e.g., performance, resource usage) of this application very different from the past? If so, in what way and why?
- **Rogue/victim:** Is this application causing problems on my cluster or is the performance of this application being affected by one or more other applications?

Many operational performance requirements are needed at the "macro" level compared to the level of individual applications. These include:

- Configuring resource allocation policies to meet SLAs in multi-tenant clusters.
- Detecting rogue applications that can affect the performance of SLA-bound applications through a variety of low-level resource interactions.
- Configuring the hundreds of configuration settings that distributed systems are notoriously known for having to get the desired performance.
- Tuning data partitioning and storage layout.
- Optimizing dollar costs on the cloud.
- Capacity planning using predictive analysis to account for workload growth proactively.

Overview of a Performance Management Strategy



Architecture of a performance management platform for the big data stack.

Suffice to say, distributed big data stacks come with many inherent challenges. To address these challenges and tame highly distributed big data deployments, organizations need an approach to application performance management that delivers all of the following:

- **Full data stack collection:** To answer questions — such as, “what caused this application to fail?” or "will this application ever meet its SLA?" — monitoring data from every level of the stack will be necessary. This includes data from SQL queries, execution plans, data pipeline dependency graphs, and logs from the application level; resource allocation and wait-time metrics from the resource management and scheduling level; and actual CPU, memory, and network usage metrics from the infrastructure level, among other sources. Collecting such data in a non-intrusive or low-overhead manner from production clusters remains a major technical challenge, but this challenge is being addressed by the database and systems community.
- **Event-driven data processing:** Large big data stacks can include over 500 nodes and run hundreds of thousands of applications every day across ETL, BI, data science, and streaming systems. These deployments generate tens of terabytes of logs and metrics every day. This data introduces two unique challenges — variety and consistency — which are further outlined below. To solve these two problems, the data processing layer has to be based on event-driven processing algorithms whose outputs converge to the same final state, irrespective of the timeliness and order in which the monitoring data arrives. The end user should get the same insights irrespective of the timeliness and order in which the monitoring data arrives.
- **The variety challenge:** The monitoring data collected from the big data stack covers the full spectrum from unstructured logs to semi-structured data pipeline dependency DAGs to structured time-series metrics. Stitching this data together to create meaningful and useable representations of application performance is a nontrivial challenge.
- **The consistency challenge:** Monitoring data has to be collected independently and in real-time from various moving parts of the multiple distributed systems that comprise the big data stack. Thus, no prior assumptions can be made about the timeliness or order in which the monitoring data arrives at the processing layer.
- **Machine learning-driven insights and policy-driven actions:** Enabling all of the monitoring data to be collected and stored in a single place opens up interesting opportunities to apply statistical analysis and learning algorithms to this data. These algorithms can generate insights that, in turn, can be applied manually by the user or automatically based on configured policies to address the performance requirements identified earlier.

Conclusion

The modern big data stack faces many unique performance management challenges. These challenges exist at the individual application level, as well as at the workload and cluster levels. To solve these problems at every level, a performance management strategy needs to offer full stack data collection, event-driven data processing, and AI-driven insights and policy-driven actions. As organizations look to get more and more out of their big data stack — including the use of artificial intelligence and machine learning — it's imperative that they adopt a performance management approach that provides these key pillars.

SHIVNATH BABU is the CTO at Unravel Data Systems and an adjunct professor of computer science at Duke University. His research focuses on ease-of-use and manageability of data-intensive systems, automated problem diagnosis, and cluster sizing for applications running on cloud platforms. Shivnath co-founded Unravel to solve the application management challenges that companies face when they adopt systems like Hadoop and Spark. Unravel originated from the Starfish platform built at Duke, which has been downloaded by over 100 companies. Shivnathhas won a US National Science Foundation CAREER Award, three IBM Faculty Awards, and an HP Labs Innovation Research Award. LinkedIn

Big Data Building Blocks:

Selecting Architectures and Open-Source Frameworks

QUICK VIEW

01. It's important to start with a clear definition of your business and solution/product needs.

02. There are many frameworks to use; adapt the right tool for the right scenario.

03. Frameworks are added and improved every day. Continue learning and evolving with them.

BY ADI POLAK
SR. CLOUD DEVELOPER ADVOCATE, MICROSOFT

From bare metal servers to the cloud, data at scale is everywhere. What we don't talk about as much, however, is that it too often ends up as tedious series of tasks — from complex real-time processing, event processing, data analyzing, data streaming, and building machine learning models, all the way down to simpler activities like cleaning and preparing data. Developers, DevOps teams, and data scientists across industries and organizations are searching for ways to automate these processes, but they still face another issue: finding the best solutions to big data's biggest challenges, like high availability, horizontal scaling, and fault tolerance. Many open-source tools, like Apache Kafka and Apache Spark, claim to solve these challenges, but that leaves teams asking questions like:

- How do we choose the best tool for our organization and products?
- Is there a generic architecture for data at scale?
- Can we use a basic architecture to start?

In this article, we'll use examples of common business scenarios, walking through our requirements and database decision process to help you see how to evaluate various architectures and open-source tools to find the best solution for *your* scenario.

The Scenario: Twitter Sentiment Analysis

Many customers use social media to talk about products and services. Twitter is no different. Opinion-filled tweets can go viral and dramatically impact your product's (and company's) reputation. So, in our sample scenario, let's imagine we're a regional retail company. We would like to track and analyze Twitter posts in real-time so that we can take action when necessary, showing our appreciation for positive feedback and mitigating customer dissatisfaction quickly.

- **The problem**: We have many products and services, and across all business units, our customers generate 100,000 tweets each hour, seven days a week.
- **The business goal**: An easy, automated way to understand social sentiment so that we catch issues before they escalate.
- **The solution requirements**: Real-time social media monitoring, horizontal scaling to accommodate future growth, and a real-time summary and alert system that pings customer success/operations teams (based on certain criteria).

WHAT WE'LL USE:

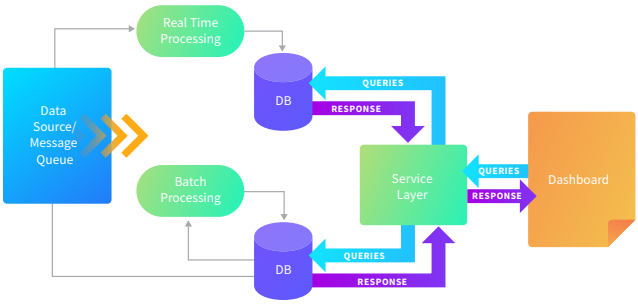
- **Real-time subscriber** that will subscribe and pull data from the Twitter API
- **Message queue**
- **Text parser** that will adjust tweets into a format that our sentiment analysis engine can consume, as well as add elements such as likes, retweets, etc.
- **Sentiment analysis engine** that will evaluate the tweet "feeling" and return its sentiment (positive/negative)
- **Score engine** that will receive data from the sentiment analysis

engine and the parser, run an algorithm that defines tweet severity, and evaluate whether or not to fire an alert to our teams

- **Database** that will store our data, which the service layer will use to transfer to a dashboard in real-time

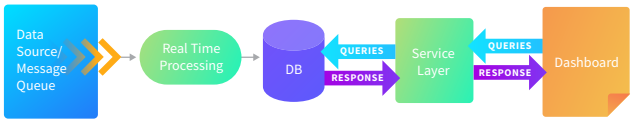
GENERIC ARCHITECTURE OPTIONS

A Kappa architecture consists of a message queue, a real-time processing layer, and a service layer. It is designed for both parallel processing and asynchronous or synchronous pipelines.



Simple Kappa architecture diagram

A Lambda architecture is similar to the Kappa, with one extra layer, the batch layer, which combines output from all ingested data according to the product needs. In our example, the service layer is in charge of creating our dashboard view, where we combine output from the real-time processing and batch processing input to create a view that consists of insights from both.



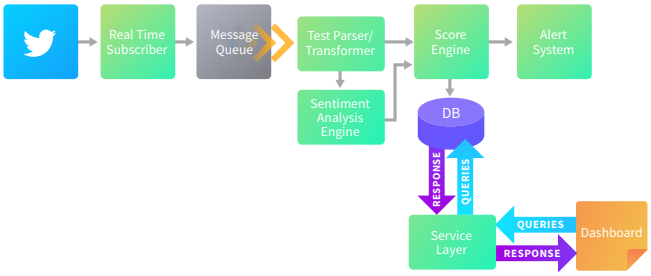
Simple Lambda architecture diagram

OUR SOLUTION CHOICE: KAPPA ARCHITECTURE

The Lambda architecture can be somewhat complex due to the need to maintain both batch and real-time processing codebases. Since we would like to start with a simplified solution, **we'll use the Kappa architecture option.**

Our goals:

- **Minimum latency**: Evaluate new tweets as quickly as possible (i.e. a few seconds).
- **High throughput**: Digest many tweets in parallel.
- **Horizontal scalability**: Accommodate increase in loads.
- **Integrations**: Allow different components in our system to integrate with other systems, such as alerting systems, web services, and so on. This will help us evaluate future changes in the architecture to support new features.



Our solution architecture diagram

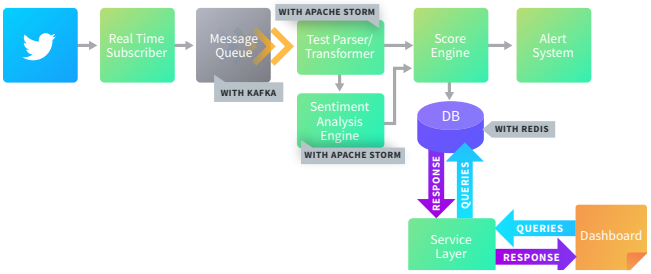
OUR DECISION PROCESS

For real-time subscribers, many commercial products exist, both in the cloud (e.g. Azure Logic Apps) and on-premise. The same goes for the alert system. One option is Graphite, which works well with Grafana. Although these two are popular, we should use **Prometheus** instead. Prometheus works better for us since it provides an out-of-the-box full alert management system (which Graphite and Grafana lack), as well as data collection and basic visualization.

As for our message queue, there are multiple products (e.g. RabbitMQ, Apache ActiveMQ, and Apache Kafka). Many frameworks, like Apache Spark and Apache Storm, have built-in integration with Kafka. We selected **Kafka** for its scalability, high throughput, easy integration, and community support. On top of that, Kafka Stream clients are supported in various programming languages, which reduces the learning curve of a dedicated programming language.

Our text parser, sentiment analysis engine, and score engine can be built with Apache Storm or Apache Spark Streaming. Apache Storm focuses on stream processing, performs task-parallel computations, and can be used with any programming languages. Apache Spark includes a built-in machine learning library MLlib , which can help us with the sentiment engine. But Spark Streaming processes data in micro-batches, which might limit the latency capabilities. Given this, and since MLlib doesn't support *all* machine learning algorithms, we chose to work with **Apache Storm**. With Storm as the streaming layer, we can process each tweet as one task, which ends up being faster than working with data-parallel-driven Spark. This meets our scalability, low latency, and high throughput goals.

Finally, we need to evaluate our database options. Since we want to see a summary of our analyses in real-time on a dashboard, we'll use an indexed, in-memory DB like **Redis** for faster queries and minimum latency.



Our solution architecture diagram with our technology selections

It's important to note that our architecture setup allows changes. As our system develops and we add statistical features, we can add a batch layer, turning the Kappa architecture into a Lambda one. The architecture also supports adding multiple inputs from various social networks, like LinkedIn. If there's a good chance that you'll add a batch layer in the future, consider using Apache Spark instead of Apache Storm. This way, you'll maintain one framework for both stream and batch processing.

Big Data Landscape at a Glance: Overview and Consideration

Apache Kafka provides a unified, high-throughput, low-latency platform for handling real-time data feeds, and most clouds support a managed Kafka. Kafka also provides Kafka Stream for streaming applications, as well as Kafka Connect, with which we can create a database connector that reads data from Kafka and writes it to a desired database, like PostgreSQL, MongoDB, and more. Kafka delivers an in-order, persistent, and scalable messaging system, supports microservices, and has a thriving open-source community — making it an extremely popular message queue framework.

Redis is an open-source in-memory data structure project that implements a distributed, in-memory key-value database with optional durability. Redis supports different kinds of abstract data structures, such as strings, lists, maps, sets, sorted sets, HyperLogLogs, bitmaps, streams, and spatial indexes. In the Stack Overflow 2018 Developer Survey, Redis ranks as developers' "most-loved database," but it's important to note that Redis doesn't support join operation or query language out-of-the-box. You'll also need to learn Lua for creating your own stored procedure; the learning process is longer and perhaps harder.

Apache Spark is an open-source distributed general-purpose cluster computing framework used most often for machine learning, stream processing, data integration, and interactive analytics. Spark Streaming is built on top of Apache Spark. It's a scalable fault-tolerant streaming processing system that natively supports both batch and streaming workloads, with out-of-the box support for graphs and machine learning libraries. Spark is relatively simple to grasp, is extremely fast, and has vast community support.

Apache Storm is a distributed stream processing computation framework written predominantly in the Clojure programming language. Usually compared with Spark Streaming, Storm is focused on stream processing and performs task-parallel computations. Storm is often used with Kafka, where Storm processes the data and Kafka is used for event stream or message bus purposes. Storm has huge support in the cloud and can be used with any programming language since the communication is over JSON-based protocol. Current adapters exist in Python, Ruby, JavaScript, Perl, and more.

Compared to Apache Samza, yet another stream processing framework, Storm shines in high-speed event processing (which is good for synchronous systems), while Samza shines in processing many gigabytes. Both are well-suited to work with massive amounts of real-time data.

What's Next?

There are endless topics, technologies, scenarios, and requirements, so it's impossible to cover them all. This article gives you a foundation to evaluate your needs, understand the broader database architecture and technology landscape, and select from popular open-source tools and services.

However, it's important to remember that, as with all technology, new products and updates are released at a rapid pace. When you build your solution, think about how it'll handle future improvements, and continually evaluate if and how your solution meets your needs.

In summary: continue learning, always.

And I'm a lifelong learner who will be learning right along with you. If you'd like to see what I'm discovering, I write about all things big data on my personal blog.

If you'd like to build your own sentiment analysis tool, check out this Azure Stream Analytics tutorial.

ADI POLAK You will often find Adi writing code, investigating software engineering problems, reading, talking, and playing with tech. She believes in sharing knowledge and learning from others, and helped found FlipCon to support the functional programming community. Adi is active in big data, security, and JVM-based languages communities, including Java, Scala, and Kotlin, and has years of experience working with high-scale data and machine learning. LinkedIn - Twitter

Spark as a Fundamental Open-Source Big Data Technology

QUICK VIEW

01. Apache Spark is in widespread commercial use as a big data analytics engine.

02. Spark is faster, more flexible, and easier to adopt than Hadoop.

03. Spark adoption still remains in an early though explosive phase.

BY CAMERON LAIRD
VP, PHASEIT

Spark as a Fundamental Open-Source Big Data Technology

Big data — the analysis of datasets beyond the capacity of conventional tools — has been a necessity for many industries for many years. The particular unconventional tools that make big data feasible have changed through the years, however. As this article explains, Spark is the platform whose adoption for solving big data problems currently appears most explosive.

Time Before Spark

To understand Spark's potential, it helps to recall the shape of big data one decade ago. In 2008-2009, the big data-as-a-business concept was often conflated with Hadoop technology. Hadoop is an open-source framework for managing clusters (networks of multiple computers) operating on MapReduce programming tasks. MapReduce is a programming model popularized by Google in 2004 that structures the collection and analysis of large datasets. A decade ago, paradigmatic big data projects were coded as MapReduce batches applied to the data of a particular domain and then executed on Hadoop-managed clusters. Big data and Hadoop were so closely identified then and for several years after that departments unfamiliar with big data (e.g. venture capitalists, PR firms, HR departments) notoriously confused the two in their advertisements and other writing. It's fair to summarize, as Kaushik Pal does, that Hadoop is "the basic data platform for all big data-related offerings."

Hadoop's emphasis on batch processing is clumsy for iterative and interactive jobs. Even more than that, Hadoop's MapReduce interpretation assumes that datasets reside in the Hadoop Distributed File System (HDFS). Many (perhaps the majority of) datasets fit this model uncomfortably. High-performance machine learning, for instance, emphasizes in-memory processing with relatively infrequent recourse to filesystem mass storage.

Spark, "a unified analytics engine for large-scale data processing" that began as a Berkeley class project in 2009, emphasizes:

- Compatibility with Hadoop through the reuse of HDFS as a storage layer
- Interactive querying
- Support of machine learning
- Pipelining (i.e. ease of connection of different execution units so that a complex calculation can be achieved as a "bucket brigade" that passes data through successive stages of computation)

Spark also features flexibility in several aspects, including the different programming languages it serves, the clouds in which it can be rented, and the big data libraries it integrates.

Spark vs. Hadoop

Spark is typically faster than Hadoop, with a factor of up to 100+ jobs for fitting Spark's in-memory model better. Spark is tuned for typical ML tasks like Naive Bayes and K-Means computations, and can also help save time and alleviate hardware constraints. Early Spark projects, however, had a reputation for leaking memory, at least in the hands of novices. Additionally, long-running batch MapReduce jobs appear to be easier to get correct with Hadoop.

Spark is also a more general-purpose programming framework, as mentioned above and as the examples below show in more detail. Hadoop conceives big data rather inflexibly as Java-coded MapReduce operations; in contrast, the learning curve for Spark is far less steep. A conventional programmer in Python, Java, Scala, R, or even SQL can almost immediately start to write familiar-looking programs on a conventional desktop that simultaneously leverage Spark's power. Spark's official site has several evocative examples. Consider this word counter in Python:

```
import pyspark

source = "file://..."
result = "file://..."
with pyspark.SparkContext("local", "WordCount") as sc:
    text_file = sc.textFile(source)
    counts = text_file.flatMap(lambda line: line.split("
    "))
        .map(lambda word: (word, 1))
        .reduceByKey(lambda a, b: a + b)
    counts.saveAsTextFile(result)
```

Any Python programmer can read this. While it runs on a low-powered development host, it also runs unchanged on Docker-ized Spark, with Spark on industrial-strength cloud clusters, experimental supercomputers, high up-time mainframes, and so on. Also, it's easy to refine such an example with conventional Python programming; a follow-up example might be:

```
import re
 import pyspark

 source = "file://..."
 result = "file://..."

 def better_word_splitter(line):
      '''
         Use negative look-behind to split on all
         whitespace, but only once per whitespace
         sequence.
      '''
      return re.split("(?<!\\s)\s", line.strip())

with pyspark.SparkContext("local", "WordCount2") as sc:
  text_file = sc.textFile(source)
  counts = text_file.flatMap(better_word_splitter)\
    .map(lambda word: (word, 1))\
    .reduceByKey(lambda a, b: a + b)
  counts.saveAsTextFile(result
```

Spark is certainly newer than Hadoop and has a reputation of being less widely understood. At the same time, Spark complements and generalizes Hadoop so existing specialists in programming domains like ETL transformations, ML, graph analysis, OLAP, dataset streaming, time-series analysis, or interactive and experimental queries can adopt Spark incrementally. Also, Spark's incorporation of these distinct domains simplifies architectural design; everything needed for a particular result can be written within a single pipeline and computed on a standard Spark cluster.

Another example from the official Spark Apache site — this time in Scala — illustrates some of the power of Spark's integration. Not long ago, predictive analysis was an undertaking for graduate school; now, the power of Spark makes it a one-liner:

```
// Every record of this DataFrame contains the label and
// features represented by a vector.
val df = sqlContext.createDataFrame(data).
toDF("label", "features")

// Set parameters for the algorithm.
// Here, we limit the number of iterations to 10.
val lr = new LogisticRegression().setMaxIter(10)

// Fit the model to the data.
val model = lr.fit(df)

// Inspect the model: get the feature weights.
val weights = model.weights

// Given a dataset, predict each point's label, and
show the results.
model.transform(df).show()
```

Spark's exposure in general-purpose programming languages such as Scala means that it's easy to extend, adapt, and integrate such powerful results with other organizational assets. Too often in the past, big data was an isolated specialization. Spark's strengths bring big data to a wider range of programmers and projects.

Keep in mind what Spark brings operationally: Once a program is correct, it will be *fast* and it will be able to be scaled heroically with Spark's ability to manage a range of clusters.

Ready to Go

All these capabilities sound nice. But is Spark truly safe for projects that rely on million-dollar hardware, not to mention the value and security of proprietary data? Yes! Billion-dollar companies including GoDaddy, Alibaba, and Shopify rely on Spark for crucial services and results.

As an intellectual property, Hadoop found a home in 2011 with the Apache Software Foundation in an innovative ownership arrangement. Spark later followed that same path. Interestingly enough, for the last four of those years, activity at the Spark repository exceeded that of the older and generally more prominent Hadoop repository. While that comparison means little in isolation, it at least hints at the large number of organizations that regard Spark as a fundamental open-source technology.

If anything, Spark's flexibility and integration make it a safer choice than Hadoop or other alternatives. While Hadoop itself behaves reliably, too many Hadoop-based projects have stumbled in interfacing to a MapReduce-focused kernel; the MapReduce part is correct, but the wrappers around it connecting to other organizational assets end up being novel and correspondingly shaky. In contrast, Spark's more general framework invites the kind of convenient, trustworthy interfaces that contribute to the success of a project as a whole.

Conclusion

Derrick Harris was right when he summarized Spark for a business audience over three years ago: "Spark is faster, more flexible, and easier to use than Hadoop MapReduce." Spark's sophisticated in-memory processing makes it faster — sometimes by orders of magnitude. Spark maintains a rich array of APIs for graphs, streaming, ML, and more that even manage Spark's own in-memory acceleration. Spark builds in pipelines and supports multiple clustering facilities. Programmers can work in any of five languages rather than just Hadoop's Java basis.

For all these reasons, Spark's growth will only increase in the next few years. Spark is the one technology that big data practitioners most need to know.

CAMERON LAIRD is an award-winning programmer and author, and a frequent contributor to DZone. Applications he's written in a variety of languages are in operation under the surface of the sea, in low earth orbit, in planes, trains, and automobiles, and in plenty of points between. In recent years, he's shifted part of his focus to coaching the most recent cohort of developers to launch their careers smarter and more effectively.

DZONE.COM/GUIDES

unravel™

HELP YOUR APPLICATIONS DO DATA BETTER

Your Modern Data Applications, ETL, IoT, Machine Learning, Customer 360 and more, need to perform reliably. With Big Data, that's not always easy.

UNRAVEL MAKES DATA WORK

Unravel removes the blind spots in your data pipelines, providing AI-powered recommendations to drive more reliable performance in your modern data applications.

Greater Productivity

98% reduction in troubleshooting time

Guaranteed Reliability

100% of apps delivered on time

Lower Costs

60% reduction in cost

Don't just monitor performance – optimize it. LEARN HOW → UNRAVELDATA.COM

Will AI Finally Make Big Data Doable for the Enterprise?

Evidence of the success of big data is all around us. Amazon knows what we want based on the likes and wants of our million closest friends. Facebook will not let you forget that you once looked for a new lawn mower. Companies like LinkedIn, Amazon, and Google are all doing fine with big data and enjoying market caps that reflect the successful monetization of their data and associated algorithms.

But what about everyone else? The results are mixed and there is a palpable sense of disillusionment around the considerable cost, effort, and time needed to get a big data program off the ground. For data operations teams (DataOps), the margin between success and failure often hinges on two essential challenges:

- **Poor performance** achieving predictable peak performance of big data applications and their data pipeline components like Spark, Hadoop, and Kafka.
- **Scarce skillsets** acquiring and retaining the skills necessary to achieve and sustain success.

Unravel addresses both of these challenges

The key to solving both of these problems lies in the use of AI. Achieving peak performance on systems running hundreds of jobs across thousands of cluster nodes has become impossibly hard. So, we over-provision our datacenters or cloud resources and plan for the worst case. But we can apply AI algorithms to the operational metadata created during application runs to understand and predict application slowdowns and failures, and then prescribe corrective action.

To empower data operations (DataOps) teams to be able to scale their capabilities, AI can use operational metadata to diagnose underperforming applications and triage failures in a fraction of the time it would take if done manually.

These capabilities are built into Unravel.

WRITTEN BY DR. SHIVNATH BABU - CTO AT UNRAVEL DATA

PARTNER SPOTLIGHT

Unravel Data

Unravel radically simplifies the way you understand and optimize the performance of your data pipelines, with full-stack visibility and AI-powered recommendations.

Category Data Operations

New Release Unravel 4.5

Open Source? No

Case Study Unravel helps a top-20 global bank reduce big data application troubleshooting and tuning time, and speeds up response times for support teams serving application development and IT operations. The bank's multi-tenant data analytics platform is a very complex environment, and tooling has been an ongoing challenge for both app dev and operations teams. Unravel provides a 360° view of their modern data application portfolio, so that those teams can be proactive in their troubleshooting efforts and can support the SLAs required by the business. As a result of using Unravel's AI-powered auto-tuning and automated troubleshooting, the bank has seen a 70% reduction in support tickets and a 98% reduction in troubleshooting time for issues with their data pipelines. They have also seen a 60% reduction in resource costs through optimization and performance tuning.

Strengths

1. Captures and correlates performance and status data for Spark, Kafka, Hadoop, Hive, Hbase, Tez, Impala, and other data systems
2. For IT Operations - Reduce MTTI and MTTR with a unified, full stack view of your complex data pipelines with AI-powered alerts and recommendations
3. For App Developers - Write better code and optimize performance for cloud and hybrid environments with actionable advice, insights, and recommendations.
4. For Architects - Design production-ready data pipelines for current data needs and future hybrid cloud deployments
5. Cross-platform architecture supports Amazon, Microsoft, and Google clouds, as well as on-premises operations, hybrid, and multi-cloud environments.

Notable Customers

- Kaiser Permanente
- Autodesk
- Neustar
- TIAA
- Leidos
- Wayfair

Website	unraveldata.com
Twitter	@unraveldata
Blog	unraveldata.com/blog

DZONE.COM/GUIDES

Big Data in the Renewable Energy Sector

QUICK VIEW

01. Renewable energy sources need to be scaled up in order to replace traditional sources that are responsible for greenhouse gas emissions. A combination of Big Data and artificial intelligence can help the industry become more efficient.

02. Big Data is used for accurate prediction of meteorological variables that generate solar and wind power, using computational intelligence techniques for real-time analysis.

03. Big Data can also help domestic and industrial electricity consumers to use less power. However, Big Data itself is a huge consumer of electricity, so it also needs to be used to cool down more efficiently.

BY JO STICHBURY
FREELANCE TECHNICAL WRITER

In this article, I will look at how big data and AI can be used to improve efficiency of renewable energy production and offer opportunities for reducing electricity consumption.

Introduction

We have seen a global revolution in the use of big data to improve efficiency in manufacturing, security, and healthcare, to name just a few industries. In recent years, environmental issues, particularly climate change, have attracted concern and been widely discussed. Can the same approach be used for energy monitoring, modeling, analysis, and prediction to achieve sustainable energy objectives and to reduce the volume of carbon dioxide emissions that are causing global warming?

Clean and Efficient Electricity Generation

In the US, renewable energy sources generate 17 percent of electricity used, and calculations suggest that solar, wind, hydroelectric, and other renewable sources are the world's fastest-growing energy source according to the Energy Information Administration of the U.S. Department of Energy.

Renewable energy sources need to be scaled up in order to replace the traditional energy sources that are responsible for greenhouse gas emissions before it is too late to reverse the impacts on our ever-warming climate. In order to scale, they need to be as efficient as possible, and a combination of Big Data and artificial intelligence can help. Blending renewable energy into existing utility grids requires estimation of the power that will come from solar-, wind- and hydro-electricity sources in order for the infrastructure to function with appropriate estimation, planning, pricing, and real-time operations.

PREDICTING AND MAXIMIZING SOLAR ELECTRICITY PRODUCTION

Power generation from distributed solar photovoltaic (PV) arrays has grown rapidly in recent years, with global photovoltaic capacity estimated to reach over 1 terawatt of solar capacity within the next 5 years, according to the latest global data.

Big data is used for accurate prediction of meteorological variables, pulling in disparate observational data sources and models, then using computational intelligence techniques for real-time analysis. For example, SunCast is a system from the National Center for Atmospheric Research (NCAR) that is used to provide solar energy forecasting. It is based on real-time measurements in the field and satellite data for cloud patterns. The forecast blends a number of models and tunes them according to historic observations using statistical learning and a host of artificial intelligence algorithms.

You may have driven or taken a train past a large, rural, photovoltaic array. If you live in a town, you have probably seen PV panels on rooftops. In the urban environment, where is the best place to locate PV arrays? A recent paper illustrated the use of image recognition and machine learning to determine the best sites to place rooftop-based PV arrays, allowing local decision makers to assess the potential solar-power capacity within their jurisdiction. The approach does not require the use of 3D city models and instead uses public geographical building data and aerial images. The AI takes the geodata and outputs irradiance simulation and power generation potentials, which can be used to determine the best sites for PV panels.

Since solar panels may be placed in inaccessible areas, their owners need

to be aware of environmental factors that can have negative effects on their efficiency and cause a loss of power generation, such as shading, fallen leaves, dust, snow, and bird damage, among others. Machine learning can be used to monitor the output from individual panels as a set of time series data, with the model trained to detect anomalous outputs and classify them. The AI can then indicate a problem on a particular panel's surface, which can then be scheduled for inspection and repair.

PREDICTING WIND-TURBINE PRODUCTION

Wind power provides a significant opportunity for future power generation and is growing substantially each year. One report suggests that wind power could reach nearly 2,000 GW by 2030, supplying between 16.7-18.8 percent of global electricity and help save over 3 billion tons of CO2 emissions, although this is an ambitious prediction, and I'd recommend that you consult the full report if you are interested in the nuances involved.

Wind power predictions are needed for turbine control, load tracking, power system management, and energy trading. Many different wind power prediction models have been used in combination with data mining. There are a number of approaches, such as a physical (deterministic) approach, based on lower atmosphere or numerical weather predictions using weather forecast data like temperature, pressure, surface roughness, and obstacles. An alternative statistical approach uses vast amounts of historical data without considering meteorological conditions and relies upon artificial intelligence (neural networks, neuro-fuzzy networks) and time series analysis approaches. A final approach is a hybrid model that combines both physical and statistical methods.

Reducing Electricity Consumption

A number of households are now familiar with the concept of home energy monitors, which consist of a sensor, a transmitter, and a handheld display. The sensor clips onto a power cable connected to your electricity meter box and monitors the magnetic field around the power cable to measure the electrical current passing through it. The transmitter takes data from the sensor and sends it to the handheld display unit, which calculates your power usage, the costs, and the greenhouse gas emissions (tons of CO2), assuming the electricity is from a non-renewable source. By collecting and analyzing the data from a sufficiently large number of homes, it is possible to determine where energy savings can be made or where there is flexibility in usage outside of peak hours. Consumers can then be advised on how to reduce their consumption, cut their bills, integrate renewable energy, and reduce emissions.

For example, in some states in the US where energy markets are deregulated, customers can choose between different energy providers, but each offers a different tariff and promotional rate, which complicates selection. Machine learning can be used within a web platform to help consumers minimize their bills. When they sign up, the customers state their energy preferences (limiting themselves to sustainable sources, for example) and the machine-learning model uses a smart meter to inspect their usage pattern and match it against the best supplier, automatically switching them to different suppliers and energy plans as better deals arise. The aim is to encourage uptake of renewable energy by offering it to consumers who are most willing to do the right thing and limit their use of non-sustainable sources as long as they are not penalized by significantly higher prices.

DATA CENTER ENERGY CONSUMPTION

While Big Data is helping in a myriad of ways to increase the generation of sustainable energy and reduce consumption, it is, itself, responsible for consuming an increasing amount of energy. As Nature News reported recently, data center energy usage in 2018 exceeded the national energy consumption of some countries. Currently, data centers account for approximately 1 percent of global electricity demand, but usage is predicted to rise rapidly within the coming years — particularly if computationally intensive cryptocurrency mining continues to grow. Data center usage will make a significant contribution to global carbon emissions, since only approximately 20 percent of the electricity used in them comes from renewable sources, according to Greenpeace.

The main cause of energy consumption in a data center is cooling, which is typically performed by pumps, chillers, and cooling towers. Traditionally, it has been difficult to optimize the cooling process manually because of the complexity of the interactions between the combinations of necessary equipment. The rules and heuristics needed for every scenario have been difficult to define, particularly when interactions with the surrounding environment (such as the weather) are also considered. The result was that human operators were unable to calculate changes to settings that could respond sufficiently quickly to variations within the data center environment in order to optimize electricity efficiency.

To investigate whether AI could do better, Google turned to DeepMind, and in 2016, the team blogged about a deep learning model trained with sensor data that was able predict the impact of environmental factors on performance and energy consumption. The model makes recommendations to human operators to suggest optimization settings to improve cooling efficiency and thus reduce power consumption. In one particular Google data center, the model effected a 40 percent drop in energy usage for cooling.

In Conclusion

Big data and AI are fundamentally changing the models of power generation, pricing, and consumption, causing significant disruption in the energy sector. New, smarter ways of monitoring, modeling, analyzing, and predicting energy generation and usage are helping us to achieve sustainable energy objectives as the global population faces an unprecedented environmental challenge.

JO STICHBURY is a freelance technical writer with over 20 years' experience in the software industry, including 8 years of low-level mobile development. Jo typically writes about machine intelligence, high performance computing, electric and driverless vehicles, and the future of transport. She holds an MA and a PhD in Natural Sciences from the University of Cambridge. LinkedIn - Twitter

Python and HDF5 for Machine Learning

QUICK VIEW

01. Python is a key platform for machine learning today

02. Pickle files are a common, but inappropriate medium for saving datasets for machine learning applications

03. HDF5 is a more performant, easy to use, and pythonic alternative

BY CHRIS LAMB
CYBERSECURITY RESEARCH SCIENTIST AT SANDIA NATIONAL LABORATORIES

The Python platform is an incredibly powerful alternative to closed source (and expensive!) platforms like Matlab or Mathmatica. Over the years, with the active development of packages like Numpy and Scipy (for general scientific computing) and platforms like Tensorflow, Keras, Theano, and Pytorch, the power available to everyone today via the Python environment is staggering. Add things like Jupyter notebooks, and for most of us, the deal is sealed.

Personally, I stopped using Matlab almost five years ago. Matlab has an incredible array of software modules available in just about any discipline you can imagine, granted, and Python doesn't have that magnitude of modules available (well, not yet at least). But for the deep learning work I do every day, the Python platform has been phenomenal.

I use a couple of tools for machine learning today. When I'm working on cybersecurity, I tend to use **pip** as my module manager, and **virtualenvwrapper** (with virtualenv, duh) as my environment manager. For machine learning, I use **Anaconda**. I appreciate Anaconda because it provides both module management and environment management in a single tool. I would use it for cybersecurity work too, but it is scientific computing focused, and many of the system-oriented modules I use aren't available via Anaconda and need to be installed via Pip.

I also install **NumPy**, **scikit-learn**, **Jupyter**, **IPython**, and **ipdb**. I use the base functionality of these for machine learning projects. I'll usually install some combination of **TensorFlow**, **Keras**, or **PyTorch**, depending on what I'm working on. I use **tmux** and **powerline** too, but these aren't Python modules (well, **powerline** is, via **powerline-status**). They are pretty though, and I really like how they integrate with IPython. Finally, I install **H5py**.

H5py is what I wanted to talk to you about today. A surprising number of people aren't familiar with it, nor are they familiar with the underlying data storage format, HDF5. They should be.

Python has its own fully functional data serialization format. Everyone who's worked with python for any amount of time knows and loves pickle files. They're convenient, built-in, and easy to save and load. But they can be BIG. And I don't mean kind of big. I mean many gigabytes (terabytes?) big, especially when using imagery. And let's not even think about video.

HDF5 (Hierarchical Data Format 5) is a data storage system originally designed for use with large geospatial datasets. It evolved from HDF4, another storage format created by the HDF Group. It solves some significant drawbacks associated with the use of pickle files to store large datasets — not only does it help control the size of stored datasets, it eliminates load lag, and has a much smaller memory footprint.

Storage Size

HDF5, via H5py, provides you with the same kind of flexibility with regard to stored data types as NumPy and SciPy. This provides you with the ability to be very specific when specifying the size of elements of a tensor. When you have millions of individual data elements, there's a pretty significant difference between using a 16-bit or 32-bit data width.

You can also specify compression algorithms and options when creating and saving a dataset, including LZF, LZO, GZIP, and SZIP. You can specify the aggression of the compression algorithm too. This is a huge deal — with sparse datasets, the ability to compress elements in those datasets provides huge space savings. I typically use GZIP with the highest level of compression, and it's remarkable how much room you can save. On one image dataset I recently created, I was forced to use an int64 to store a binary value because of the model I was using. Compression allowed me to eliminate almost all the empty overhead on these binary values, shrinking the archive by 40% from the previous int8 implementation (which was saving the binary value as ASCII, using the entire width of the field).

Load Lag

Pickle files need to be completely loaded into the process address space to be used. They are serialized memory resident objects, and to be accessed they need to be, well, memory resident, right? HDF5 files just don't care about that.

HDF5 is a hierarchical set of data objects (Big shock, right? Since hierarchical is the first word in the name?). So, it's more like a filesystem than a single file. This is important.

Because it's more of a filesystem than a single data file, you don't need to load all the contents of the file at once. HDF5 and H5py load a small driver into memory, and that driver is responsible for accessing data from the HDF5 data file. This way, you only load what you need. If you've ever tried to load large pickle files, you know what a big deal this is. And not only can you load the data quickly, you can access it quickly via comfortable Pythonic data access interfaces, like indexing, slicing, and list comprehensions.

Data Footprint

Not having to load all your data every time you need to use it also gives you a much smaller data footprint in runtime memory. When you're training deep networks using high resolution true color imagery, where your pixel depth is on the order of 32 bits, you are using A LOT of memory. You need to free up as much memory as you can to train your model so you can finish in a few days instead of a few weeks. Setting aside terabytes (or gigabytes, for that matter) of memory just to store data is a waste of resources, using HDF5 you don't have to.

HDF5 is essentially a key/value store, stored as a tree. You have access to either **Datasets** or **Groups**. Datasets are, well, datasets. Groups are collections of datasets, and you access them both via keys. Datasets are leaf elements in the storage graph, groups are internal nodes. Groups can hold other groups or datasets; datasets can only contain data. Both groups and datasets can have arbitrary metadata (again stored as key value pairs) associated with them. In HDF5-ese, this metadata are called **attributes**. Accessing a dataset is as easy as this:

```
import h5py as h5
with h5.File('filename.h5', 'r') as f:
group = f['images']
dataset = group['my dataset']
# Go ahead, use the dataset! I dare you!
```

Figure 1: Getting your HDF5 on, Python style.

H5py, the python interface to HDF5 files, is easy to use. It supports modern **with** semantics, as well as traditional **open/close** semantics. And with attributes, you don't have to depend on naming conventions to provide metadata on stored datasets (like image resolution, or provenance, or time of creation). You store that data as attributes on the object itself.

Python is frequently used in data analysis today, both in statistical data analysis and machine learning. Many of us use native serialization formats for our data as well. While pickle files are easy to use, they bog down when dealing with large amounts of data. HDF5 is a data storage system designed for huge geospatial data sets and picks up perfectly where pickle files leave off.

CHRIS LAMB currently serves as a Cybersecurity Research Scientist with Sandia National Laboratories. He is also a Research Assistant Professor affiliated with the Electrical and Computer Engineering department at the University of New Mexico. Currently, his research interests center around industrial control system cybersecurity, machine learning, artificial intelligence, and their intersections. He is a TOGAF 9 Certified Enterprise Architect and a Certified Information Systems Security Professional (CISSP) through the International Information Systems Security Certification Consortium. LinkedIn

HPCC SYSTEMS®

Open source big data analytics made easy

Discover HPCC Systems – the truly open source big data solution that allows you to quickly process, analyze and understand large data sets, even data stored in massive, mixed-scheme data lakes. Designed by data scientists, HPCC Systems is a complete integrated solution from data ingestion and data processing to data delivery. **The free online introductory courses and a robust developer community allow you to get started quickly.**

EASY TO USE. | PROVEN. | SCALABLE. | COMPLETE.

HPCC SYSTEMS PLATFORM COMPONENTS

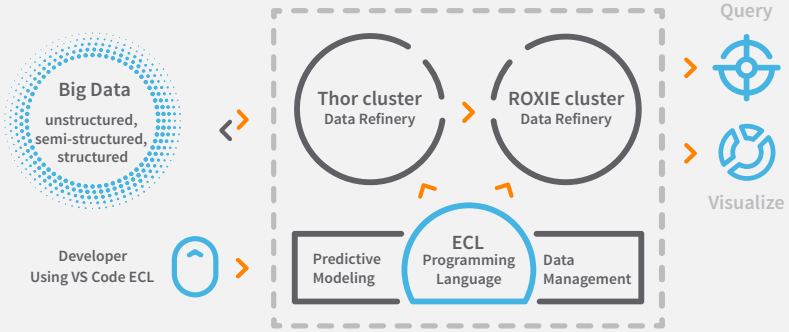
ETL
Extract, Transform, and Load your data using a powerful programming language (ECL) specifically developed to work with data.

Data Management Tools
Data profiling, data cleansing, snapshot data updates and consolidation, job scheduling, and automation are some of the key features.

Query Engine
An indexed based search engine to perform real-time queries. SOAP, XML, REST, and SQL are all supported interfaces.

Predictive Modeling Tools
In place (supporting distributed linear algebra) predictive modeling functionality to perform linear Regression, Logistic Regression, Decision Trees, and Random Forests.

HPCC SYSTEMS



"With other Big Data technologies, we need to use many different open-source modules; it's a lot of work to make them work together. With HPCC Systems, there is just one language, ECL, that can do almost everything."

MIKE YANG
Principal Technology Architect, Infosys

VISIT: **HPCCSYSTEMS.COM**

LexisNexis® RISK SOLUTIONS

HPCC SYSTEMS®

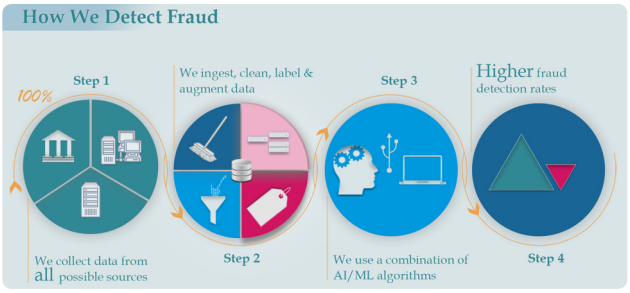
DataSeers and HPCC Systems Bring Big Data Analytics to Regional Banks

DataSeers Helps Regional Banks Analyze Huge Prepaid Card Data Sets with Ease, Speed and Flexibility

Prepaid cards are a hit with consumers, and BFSI Research forecasts the global market for them will reach $3.65 billion by 2022. Accordingly, prepaid card services are an increasingly popular offering for financial institutions, provided they can keep up with the steep data storage and analysis requirements. In the past, these requirements precluded smaller, regional banks from getting into the prepaid card market; the requisite IT staff, hardware setup, and software licenses were too expensive for their smaller budgets. But this barrier to entry is lowering thanks to the rise of integrated big data analysis platforms and open source software.

DataSeers, an HPCC Systems partner, is an Atlanta-based company specializing in data solutions for the financial industry. DataSeers developed a big data appliance, called FinanSeer, based on the open-source HPCC Systems platform. The FinanSeer appliance helps banks improve and accelerate their big data capabilities so they can:

1. Increase the speed at which they onboard new client data
2. Improve fraud oversight and reporting to keep overhead low and comply with regulatory requirements
3. Deliver data in whatever visual format is required and with minimal support from IT, so business managers can empower themselves to make better strategic decisions



DataSeers' FinanSeer appliance uses the big data processing and analytics capabilities of HPCC Systems to help banks spot fraud more quickly, improving profitability and easing regulatory compliance.

To illustrate, one of DataSeers clients is a regional bank in the Midwest. The bank reported that after implementing FinanSeer, their compliance team discovered a customer was submitting transaction data that included suspiciously similar billing and email addresses; a red flag for possible fraud. Thanks to the speed of the FinanSeer appliance, the bank searched a data set of over 400 million records to identify all of the suspicious transactions in less than 10 minutes. Prior to FinanSeer, reviewing those banking records was a manual process involving hard copy spreadsheets that could take weeks to complete.

WRITTEN BY ARJUNA CHALA
SR DIRECTOR TECHNOLOGY INNOVATION, HPCC SYSTEMS

PARTNER SPOTLIGHT

HPCC Systems

End-to-end Big Data in a massively scalable supercomputing platform. Open source. Easy to use. Proven.

HPCC SYSTEMS®

Category End-to-end Big Data **New Release** Major release annually, minor releases throughout the year **Open Source?** Yes

Case Study Proagrica is a leader in the agriculture industry, driving growth and improving efficiency by delivering high-value insight and data, critical tools, and advanced technology solutions. Proagrica needed a massively scalable data refinery environment for ingesting and transforming structured and unstructured data from its various data lakes, including extremely complicated and diverse data sets. Additional requirements included being able deliver quality/clean data from multiple data sets, provide data security, and deliver real-time analytics. Proagrica found the answer to their big data needs with HPCC Systems: a proven and enterprise-tested platform for manipulating, transforming, querying, and warehousing big data.

Strengths

1. ETL Engine: Extract and transform your data using a powerful scripting language
2. Query Engine: An index-based search engine to perform realtime queries
3. Data Management Tools: Data profiling and cleansing, job scheduling, and automation
4. Predictive Modeling Tools: Linear/logistic regression, decision trees, random forests
5. SOAP, XML, REST, and SQL are all supported interfaces

Notable Customers

- LexisNexis Risk Solutions
- RELX Group
- Infosys
- ClearFunnel
- CPL Online

Website	hpccsystems.com
Twitter	@hpccsystems
Blog	hpccsystems.com/blog

Autonomous Cars, Big Data, and Edge Computing:

What You Need to Know

BY ARJUNA CHALA
SR. DIRECTOR OF TECHNOLOGY INNOVATION, HPCC SYSTEMS

QUICK VIEW

01. Autonomous cars generate massive amounts of data; up to 4 terabytes a day.

02. Data that is critical to safely navigate the car must be analyzed locally by the car itself.

03. Today's non-critical car data can be invaluable to future autonomous driving applications.

04. Autonomous car vendors need legacy data storage strategy to avoid dark data.

The driverless car has been a high-tech dream for decades. Now that broadband connectivity, cloud computing, and artificial intelligence are increasingly available, autonomous cars should go mainstream in the near future, provided certain technical and regulatory milestones are reached. But another issue that must be addressed before self-driving cars can reach critical mass is the issue of data. Specifically, the data analysis and storage requirements of autonomous cars present challenges beyond the capabilities of most current big data solutions.

Autonomous cars generate a staggering amount of data. Intel fourestimated one car generates terabytes of data in eight hours of operation. Multiple image, radar/lidar, time-of-flight, accelerometers, telemetry, and gyroscope sensors generate data streams that must be analyzed in order to perform the calculations and adjustments required to safely navigate a car. That analysis needs to happen in real time if the car is to keep up with constantly changing driving conditions (other cars or pedestrians moving around the vehicle, changing weather and light conditions, traffic signs, and so on). These real-time performance requirements mean there's no time to upload data to a central server, conduct the necessary analytics, and then send instructions back to the car for execution. Data that is critical to safely navigate the car must be analyzed locally by the car itself — essentially, the car is an edge device in a cloud network.

Not only does the car need to analyze data on its own, it must also learn to pick and choose between different data streams to identify the ones best suited for analysis at any given moment to keep the car driving safely.

That last requirement — the need to determine what data is required to perform an analysis — is tricky. While predefined filters can help a car's machine learning routines learn what data to use and when to use it, those filters are generated by human engineers, so they can't be updated in real time. Accordingly, an autonomous car will need to run machine learning and analytics engines powerful enough to recognize mission-critical data requiring immediate analysis and action on their own, without involving a human in the analysis. Once input from a person is required, decision-making based on data analysis in real time is simply not possible.

We need analytics and machine learning algorithms for autonomous cars that can:

- Identify data in all formats.
- Recognize what data is required for mission critical operations and perform analysis of that data locally.
- Compress or aggregate non-critical data for uploading to the cloud for future use.

- Schedule uploads of non-critical data from the car to the cloud when less expensive communications are available (for example, when the car is parked overnight at home and can access the owner's Wi-Fi instead of a metered cellular network).
- Know how to call for historical data from the cloud so the AI can use it for future analytics.

Today's non-critical data can be useful for future applications, provided the data is properly stored and easily accessible. If they don't make plans in advance for how to make data available whenever necessary, autonomous car vendors run the risk of creating a "dark data" problem.

The last bullet is particularly important. An autonomous car manufacturer will be responsible for storing vast amounts of data generated by cars operating around the world, and much of that data will likely have no real value when initially captured. However, that data's value may be revealed in the future as the manufacturer's autonomous driving applications evolve and improve. Today's non-critical data can be useful for future applications, provided the data is properly stored and easily accessible. If they don't make plans in advance for how to make data available whenever necessary, autonomous car vendors run the risk of creating a "dark data" problem. Dark data is the term used to describe data assets an organization collects, but fails to take advantage of — because they don't know how to, or perhaps forgot they have. This will be a particularly significant problem for self-driving cars because of the sheer volume of data they generate.

To address the dark data problem, autonomous car vendors need to move their data storage strategies away from data warehouse models and adopt emerging data storage models like data lakes. While a detailed examination of the difference between a data warehouse and a data lake is beyond the scope of this article, to illustrate the difference between the two, compare a book with a library. With a book (data warehouse), someone has already determined what content is contained in that book and how it is formatted, while a library (data lake) allows you to store whatever content you want in almost any format. In other words, a data warehouse is a centralized platform for basic importing, exporting, and preprocessing of data gathered from a collection of linked systems using one data schema. A data lake is a distributed yet integrated data platform that supports schema-less (including unstructured and structured) data and performs queries of data in real time by leveraging metadata to quickly find, transform, and load data between systems. Data lakes' support for both structured and unstructured data on the same platform is important, as autonomous car sensors generate data streams in very different formats that can't easily be stored in the same schema. Other key differences that distinguish a data lake from a data warehouse include:

- Schema on read
- Unlimited storage
- The ability to access both raw and processed data
- The ability to link data from many individual clusters

Linking data between clusters is particularly important for autonomous cars, as it allows for integration of different data sets from different geographic locations. Car OEMs are global companies with multiple offices and datacenters scattered around the world. As more countries move to support autonomous cars, autonomous car vendors will want to use all the data generated by cars driving locally in the self-driving AI and ML algorithms they use to power their cars globally. As we see more vendors enter the autonomous driving market, the ones who will ultimately win out over others will be those vendors best prepared to analyze data at the local level and those who have catalogued their databases properly — so future autonomous applications can find the data they need, when they need it.

ARJUNA CHALA is Sr. Director of Technology Innovation for the HPCC Systems® platform at LexisNexis® Risk Solutions. With almost 20 years of experience in software design, Arjuna leads the development of next generation big data capabilities including creating tools around exploratory data analysis, data streaming and business intelligence. Arjuna has a BS in Computer Science from RVCE, Bangalore University. LinkedIn - Twitter

DZONE.COM/GUIDES

2019 Executive Insights on Big Data

BY TOM SMITH
RESEARCH ANALYST, DEVADA

QUICK VIEW

01. The keys to a successful big data initiative are defining the business problem to be solved, having the data to do so, and having the right tools.

02. Failure of big data initiatives is a function of not having the right skills, not having a clear definition of the business problem, and a lack of vision for how the project will scale.

03. The future of big data is artificial intelligence (AI) and machine learning (ML), along with streaming data and more mature toolsets.

To understand the current and future state of big data, we spoke to 31 IT executives from 28 organizations. Here's who we spoke to:

- Cheryl Martin, V.P. Research Chief Data Scientist, Alegion
- Adam Smith, COO, Automated Insights
- Amy O'Connor, Chief Data and Information Officer, Cloudera
- Colin Britton, Chief Strategy Officer, Devo
- OJ Ngo, CTO and Co-founder, DH2i
- Alan Weintraub, Office of the CTO, DocAuthority
- Kelly Stirman, CMO and V.P. of Strategy, Dremio
- Dennis Duckworth, Director of Product Marketing, Fauna
- Nikita Ivanov, founder and CTO, GridGain Systems
- Tom Zawacki, Chief Digital Officer, Infogroup
- Ramesh Menon, Vice President, Product, Infoworks
- Ben Slater, Chief Product Officer, Instaclustr
- Jeff Fried, Director of Product Management, InterSystems
- Ilya Pupko, Chief Architect, Jitterbit
- Bob Eve, Senior Director, TIBCO
- Bob Hollander, Senior Vice President, Services & Business Development, InterVision
- Rosaria Silipo, Principal Data Scientist, and Tobias Koetter, Big Data Manager and Head of Berlin Office, KNIME
- Bill Peterson, V.P. Industry Solutions, MapR
- Jeff Healey, Vertica Product Marketing, Micro Focus
- Derek Smith, CTO and Co-founder and Katie Horvath, CEO, Naveego
- Michael LaFleur, Global Head of Solution Architecture, Provenir
- Stephen Blum, CTO, PubNub
- Scott Parker, Director of Product Marketing, Sinequa
- Clarke Patterson, Head of Product Marketing, StreamSets
- Bob Eve, Senior Director, TIBCO
- Yu Xu, Founder and CEO, and Todd Blaschka, CTO, TigerGraph
- Bala Venkatrao, V.P. of Product, Unravel Data
- Madhup Mishra, V.P. of Product Marketing, VoltDB
- Alex Gorelik, Founder and CTO, Waterline Data

Key Findings

1. While there were more than two dozen elements identified as being important for successful big data initiatives, identifying the use case, having quality data, and having the right tools were mentioned most frequently. Choose the right project – a well-defined problem that's a pain point. Define the most critical use cases and identify where data is holding you back from solving those use cases. Have a clear set of goals and know the business decisions you are trying to drive.

Have reliable and valid data since the level of trust in your work will be a function of the level of reliability and the level of use. Data accuracy and correctness are critical for every data project. Inventory data sources and assess data quality. The ability for the data-driven organization to take action with complete accuracy by relying on a purpose-built, high-performance, open data analytical platform. Without the highest level of data accuracy and integrity, analysis and targeting will not be effective.

Leverage tooling to simplify data processing and analysis and to make more progress faster. Have the right tools to ingest, transform, analyze, and visualize the results. It's important to have the flexibility to look at the data using multiple tools and data models.

2. The two most popular ways to secure data are encryption and controlling authorization and access. We encrypt data when transferred and store encrypted data on disk. We never have unencrypted data so it's never at risk. Data is encrypted in transit and at rest using industry

standard encryption ciphers. Self-encrypted disk drives can be used on database servers for that level of protection within the data centers themselves.

Control data access so only users with proper permissions have access. Enterprise authorization and authentication frameworks enforce that.

3. The most frequently mentioned languages, tools, and frameworks were Python, Spark, and Kafka. TensorFlow, Tableau, and PowerBI were also mentioned by several respondents.

4. Use cases span a dozen industries and use cases with the most frequently mentioned industries being financial services, retail, and healthcare, and the most frequent use cases were around security/fraud prevention and customer insight/understanding. Fraud is a major issue, and big data is being used for anomaly detection in healthcare and financial services. Financial services is a predominant industry because money is involved. Look at large credit customers for fraud detection with two millisecond response time.

Financial services companies have new regulations with real-time reporting requirements. The Fundamental Review of the Trading Book (FRTB) regulations require financial services companies to calculate their portfolio value and risk exposure in real-time.

5. Failure of big data initiatives are a function of lack of security, skills, definition of the business problem, and inability to scale. If you don't know the lineage of the data and cannot ensure its secure, you are asking for trouble. Security needs to enforce policy. Do not put security policy definitions in the hands of developers. Start by standardizing data governance, security, and access control.

The biggest challenge of big data initiatives, like all data analytics projects, is the recruitment of qualified employees. Not having the people with the right skills can lead to a complex path or failure. Know what your people are capable of. Organizations don't have the technical expertise or engineering capacity to keep up with all the changes today's data-driven economy require.

Figure out the problem to be solved before deciding on the technology you choose. Have a clear business objective. Not having clear, precise goals for any data project is a common failure. Organizations don't spend the time to understand, categorize, and tag their information. People just take short cuts and dump or keep information en masse. They are mismanaging the process because they do not know what they have.

You need to understand how the technology scales. To achieve scalability, you will need to build your application a certain way. The challenges a lot of projects have is the difficulty of testing scale, volume, and variety with a PLC. Underestimating demand for data and the challenges of trying to scale up a compromised architecture after the fact to deal with larger demand and a broader user basis is a common failure.

6. Concerns regarding the state of big data revolve around security and governance, data quality, the amount of data, and the need to have a specific business case. There are huge security challenges to moving so much data around – fake data generation, insider attacks, and API vulnerabilities. Employees often have access to data they should not have access to which enhance the human-error factor. Internal breaches are more common and worrisome than external ones.

There's not enough emphasis on data quality and contextual relevance. We need to think about the lifecycle of information for quality, proper governance, and enforcement of governance. The rate and new sources of data is growing. Be forward thinking about the business case for the data. The biggest challenge for big data today is identifying how we will derive value from the data fast enough to inform real-time decision making.

7. The future of big data is artificial intelligence (AI) and machine learning (ML) along with streaming data and more mature toolsets. We'll see higher adoption of AI/ML using it to filter through data and enabling more people to get involved with data science. AI/ML is becoming less hype and more of a trend. Big data is not very useful by itself. The use of AI/ML technologies like TensorFlow provide great opportunities uncovering pattern a human cannot see. AI/ML will focus on making sensible answers for people.

We'll see the continued emergence of streaming, always-on technology. More tools for visualizing and reporting on big data. More mature tools with the ability to handle more data, more data types, and streaming data will arise quickly.

8. The primary thing developers need to keep in mind when working on big data projects is the business problem they are trying to solve and how what they are doing will add value to the business and improve the user experience of the customer. Focus on what matters and partner with business to solve problems. Think about the business context of what you are doing. Understand the regulations and constraints around the data. Understand the business outcome you're working on and identify the business partner to help realize the value.

Developers need to focus on how they can provide value to their specific business in response to their particular industry rather than spending all of their time trying to build functionality they can get from the market.

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

DZONE.COM/GUIDES

Big Data Solutions Directory

This directory of big data and analytics frameworks, languages, platforms, and services provides comprehensive, factual comparisons of 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.

Company	Product	Product Type	Free Trial	Website
1010data	1010edge	Data analytics, orchestration, & modeling	Available by request	1010data.com/products/1010edge
Actian	Vector	DBMS, column store, & analytics platform	Free tier available	actian.com
Aginity	Aginity Amp	Data analytics mgmt platform	Demo available by request	aginity.com/main/products
Alation	Alation	Enterprise data collaboration & analytics platform	Demo available by request	alation.com/product
Alluxio Open Foundation	Alluxio	Distributed storage system across all store types	Open source	alluxio.org
Alteryx	Alteryx Designer	ETL, predictive & spatial analytics, automated workflows, reporting, & visualization	Demo available by request	alteryx.com/products/alteryx-designer
Amazon Web Services	Amazon Kinesis	Stream data ingestion, storage, query, & analytics PaaS	N/A	aws.amazon.com/kinesis
Amazon Web Services	Amazon Machine Learning	ML algorithms-as-a-service, ETL, data visualization, modeling & management APIs, & batch & real-time predictive analytics	N/A	aws.amazon.com/machine-learning
Apache Foundation	Ambari	Hadoop cluster provisioning, mgmt, & monitoring	Open source	ambari.apache.org
Apache Foundation	Apex	Stream & batch processing on YARN	Open source	apex.apache.org
Apache Foundation	Avro	Data serialization system (data structure, binary format, container, RPC)	Open source	avro.apache.org
Apache Foundation	Beam	Programming model for batch & streaming data processing	Open source	beam.apache.org
Apache Foundation	Crunch	Java library for writing, testing, & running MapReduce pipelines	Open source	crunch.apache.org

Company	Product	Product Type	Free Trial	Website
Apache Foundation	Drill	Distributed queries on multiple data stores & formats	Open source	drill.apache.org
Apache Foundation	Falcon	Data governance engine for Hadoop clusters	Open source	falcon.apache.org
Apache Foundation	Flink	Streaming dataflow engine for Java	Open source	flink.apache.org
Apache Foundation	Flume	Streaming data ingestion for Hadoop	Open source	flume.apache.org
Apache Foundation	Giraph	Iterative distributed graph processing framework	Open source	giraph.apache.org
Apache Foundation	GraphX	Graph & collection processing on Spark	Open source	spark.apache.org/graphx
Apache Foundation	GridMix	Benchmark for Hadoop clusters	Open source	hadoop.apache.org/docs/r1.2.1/gridmix.html
Apache Foundation	Hadoop	MapReduce implementation	Open source	hadoop.apache.org
Apache Foundation	Hama	Bulk synchronous parallel (BSP) implementation for big data analytics	Open source	hama.apache.org
Apache Foundation	HAWQ	Massively parallel SQL on Hadoop	Open source	hawq.apache.org
Apache Foundation	HDFS	Distributed file system (Java-based, used by Hadoop)	Open source	hadoop.apache.org
Apache Foundation	Hive	Data warehousing framework on YARN	Open source	hive.apache.org
Apache Foundation	Ignite	In-memory data fabric	Open source	ignite.apache.org
Apache Foundation	Impala	Distributed SQL on YARN	Open source	impala.apache.org
Apache Foundation	Kafka	Distributed pub-sub messaging	Open source	kafka.apache.org
Apache Foundation	MADlib	Big data machine learning in SQL	Open source	madlib.apache.org
Apache Foundation	Mahout	Machine learning & data mining on Hadoop	Open source	mahout.apache.org
Apache Foundation	Mesos	Distributed systems kernel (all compute resources abstracted)	Open source	mesos.apache.org

DZONE.COM/GUIDES

Company	Product	Product Type	Free Trial	Website
Apache Foundation	Oozie	Workflow scheduler (DAGs) for Hadoop	Open source	oozie.apache.org
Apache Foundation	ORC	Columnar storage format	Open source	orc.apache.org
Apache Foundation	Parquet	Columnar storage format	Open source	parquet.apache.org
Apache Foundation	Phoenix	OLTP & operational analytics for Apache Hadoop	Open source	phoenix.apache.org
Apache Foundation	Pig	Turns high-level data analysis language into MapReduce programs	Open source	pig.apache.org
Apache Foundation	Samza	Distributed stream processing framework	Open source	samza.apache.org
Apache Foundation	Spark	General-purpose cluster computing framework	Open source	spark.apache.org
Apache Foundation	Spark Streaming	Discretized stream processing w/ Spark RDDs	Open source	spark.apache.org/streaming
Apache Foundation	Sqoop	Bulk data transfer between Hadoop & structured datastores	Open source	sqoop.apache.org
Apache Foundation	Storm	Distributed real-time (streaming) computing framework	Open source	storm.apache.org
Apache Foundation	Tez	Dataflow (DAG) framework on YARN	Open source	tez.apache.org
Apache Foundation	Thrift	Data serialization framework (full-stack)	Open source	thrift.apache.org
Apache Foundation	YARN	Resource manager (distinguishes global & per-app resource management)	Open source	hadoop.apache.org/docs/r2.7.1/hadoop-yarn/hadoop-yarn-site/YARN.html
Apache Foundation	Zeppelin	Interactive data visualization	Open source	zeppelin.apache.org
Apache Foundation	ZooKeeper	Coordination & state mgmt	Open source	zookeeper.apache.org
Attunity	Attunity Visibility	Data warehouse & Hadoop data usage analytics	Demo available by request	attunity.com/products/visibility
Attunity	Attunity Replicate	Data replication, ingestion, & streaming platform	Available by request	attunity.com/products/replicate
BigML	BigML	Predictive analytics server & development platform	Free tier available	bigml.com

Company	Product	Product Type	Free Trial	Website
Bitam	Artus	Business intelligence platform	Available by request	bitam.com/artus.php
Board	BOARD All in One	BI, analytics, & corporate performance mgmt platform	Demo available by request	board.com/en/product
Capsenta	Ultrawrap	Database wrapper for lightweight data integration	Available by request	capsenta.com
Cask	CDAP	Data integration & app development on Hadoop & Spark	Open source	cask.co
Cazena	Cazena	Cloud-based data science platform	N/A	cazena.com/what-is-cazena
Chart.js	Chart.js	Simple JavaScript charting library	Open source	chartjs.org
Cirro	Cirro Data Puppy	Database mgmt system	Available by request	cirro.com
Cisco	Cisco Edge Fog Fabric	IoT & streaming data analytics	N/A	cisco.com/c/en/us/products/cloud-systems-management/edge-fog-fabric
Cloudera	Cloudera Enterprise Data Hub	Predictive analytics, analytic database, & Hadoop distribution	Open source version available	cloudera.com/products/enterprise-data-hub.html
Confluent	Confluent Platform	Data integration, streaming data platform	Open source version available	confluent.io/product
D3.js	D3.js	Declarative-flavored JavaScript visualization library	Open source	d3js.org
Databricks	Databricks	Data science (ingestion, processing, collaboration, exploration, & visualization) on Spark	Open source version available	databricks.com/product/databricks
Dataguise	Dataguise DgSecure	Big data security monitoring	Available by request	dataguise.com/dgsecure-ondemand
Dataiku	Dataiku DSS	Collaborative data science platform	Free tier available	dataiku.com/dss/features/connectivity
Datameer	Datameer	BI, data integration, ETL, & data visualization on Hadoop	Demo available by request	datameer.com/product
DataRobot	DataRobot	Machine learning model-building platform	Demo available by request	datarobot.com/product
DataRPM	DataRPM	Cognitive predictive maintenance for industrial IoT	Demo available by request	datarpm.com/platform
DataWatch	DataWatch Monarch	Data extraction & wrangling, self-service analytics, & streaming visualization	Available by request	datawatch.com/in-action/monarch-desktop

DZONE.COM/GUIDES

Company	Product	Product Type	Free Trial	Website
Disco Project	Disco	MapReduce framework for Python	Open source	discoproject.org
Domo	Domo	Data integration, preparation, & visualization	Available by request	domo.com/product
Druid	Druid	Columnar distributed data store w/real-time queries	Open source	druid.io
Eclipse Foundation	BIRT	Visualization & reporting library for Java	Open source	eclipse.org/birt
EngineRoom.io	EngineRoom	Geospatial, data transformation & discovery, modeling, predictive analytics, & visualization	N/A	engineroom.io
Enthought	SciPy	Scientific computing ecosystem (multi-dimensional arrays, interactive console, plotting, symbolic math, data analysis) for Python	Open source	scipy.org
Exaptive	Exaptive	RAD & application marketplace for data science	Demo available by request	exaptive.com
Exasol	Exasol	In-memory analytics database	Free tier available	exasol.com
Facebook	Presto	Distributed interactive SQL on HDFS	Open source	prestodb.github.io
Fair Isaac Corporation	FICO Decision Management Suite	Data integration, analytics, & decision mgmt	N/A	fico.com/en/products/fico-decision-management-suite
GoodData	GoodData Platform	Data distribution, visualization, analytics (R, MAQL), BI, & warehousing	30 days	gooddata.com/platform
Google	Protocol Buffers	Data serialization format & compiler	Open source	developers.google.com/protocol-buffers/docs/overview
Google	TensorFlow	OSS library for machine intelligence	Open source	tensorflow.org
Graphviz	Graphviz	Graph visualization toolkit	Open source	graphviz.org
H2O.ai	H2O	Open-source prediction engine on Hadoop & Spark	Open source	h2o.ai/products/h2o
Hitachi Group	Pentaho	Data integration layer for big data analytics	30 days	hitachivantara.com/go/pentaho.html
Hortonworks	Hortonworks Data Platform	Hadoop distribution based on YARN	N/A	hortonworks.com/products/data-platforms/hdp

Company	Product	Product Type	Free Trial	Website
Hortonworks	Hortonworks DataFlow	Streaming data collection, curation, analytics, & delivery	N/A	hortonworks.com/products/data-platforms/hdf/
IBM	IBM BigInsights	Scalable data processing & analytics on Hadoop & Spark	Available by request	ibm.com/analytics/us/en/technology/biginsights
IBM	IBM Streaming Analytics	Streaming data app development & analytics platform	Free tier available	ibm.com/cloud/streaming-analytics
IBM	IBM InfoSphere Information Server	Data integration, data quality, & data governance	Demo available by request	ibm.com/analytics/information-server
Ignite Technologies	Infobright DB	Column-oriented store w/semantic indexing & approximation engine for analytics	N/A	ignitetech.com/solutions/information-technology/infobrightdb
Infor	Birst	Enterprise & embedded BI and analytics platform	Available by request	birst.com/product
Informatica	Big Data Management	Data integration platform on Hadoop	N/A	informatica.com
Informatica	Big Data Streaming	Event processing & streaming data mgmt for IoT	N/A	informatica.com
Informatica	Enterprise Data Lake	Collaborative, centralized data lake & data governance	N/A	informatica.com
Informatica	MDM—Relate 360	Big data analytics, visualization, search, & BI	N/A	informatica.com
Information Builders	iWay 8	Data modernization	90 days	informationbuilders.com/learn/ibtv/iway-8-iit-overview
Information Builders	WebFOCUS	BI & analytics	Available by request	informationbuilders.com/products/bi-and-analytics-platform
InterSystems	IRIS	Data mangement, interoperability, & analytics	N/A	intersystems.com/products/intersystems-iris/#technology
Java-ML	Java-ML	Various ML algorithms for Java	Open source	java-ml.sourceforge.net
Jinfonet	JReport	Visualization & embedded analytics for web apps	Available by request	jinfonet.com/product
JUNG Framework	JUNG Framework	Graph framework for Java & data modeling, analyzing, & visualizing	Open source	jung.sourceforge.net
Kognitio	Kognitio on Hadoop	In-memory, MPP, SQL, & NoSQL analytics on Hadoop	Free tier available	kognitio.com/products/kognitio-on-hadoop

DZONE.COM/GUIDES

Company	Product	Product Type	Free Trial	Website
Lavastorm	Lavastorm Server	Data prep, analytics app dev platform	Free tier available	lavastorm.com/product/explore-lavastorm-server
LexisNexis	HPCC Platform	Data mgmt, predictive analytics, & big data workflow	Open source	hpccsystems.com
LexisNexis	LexisNexis Customer Information Management	Data mgmt & migration	N/A	risk.lexisnexis.com/corporations-and-non-profits/customer-information-management
Liaison Technologies	Liaiason Alloy	Data mgmt & integration	Demo available by request	liaison.com/liaison-alloy-platform
Lightbend	Lightbend Reactive Platform	JVM app dev platform w/Spark	Open source version available	lightbend.com/products/reactive-platform
LinkedIn	Pinot	Real-time OLAP distributed data store	Open source	github.com/apache/incubator-pinot
LISA Lab	Theano	Python library for multi-dimensional array processing w/ GPU optimizations	Open source	deeplearning.net/software/theano
Loggly	Loggly	Cloud log management & analytics	14 days	loggly.com/product
Logi Analytics	Logi Analytics Platform	Embedded BI & data discovery	Demo available by request	logianalytics.com/analytics-platform
Looker	Looker Business Intelligence	Data analytics & BI platform	Demo available by request	looker.com/product/business-intelligence
Looker	Looker Embedded Analytics	Embedded analytics, data exploration, & data delivery	Demo available by request	looker.com/product/embedded-analytics
MapR	Analytics and Machine Learning Engines	Real-time analytics & ML at scale	Free tier available	mapr.com/products/analytics-ml
MapR	Converged Data Platform	Big data platform on enterprise-grade Hadoop distribution w/ integrated open-source tools; NoSQL DBMS	Free tier available	mapr.com/products/mapr-converged-data-platform
MapR	DataTorrent RTS	Stream & batch (based on Apache Apex) app dev platform	Open source	mapr.com/apps/datatorrent
MapR	MapR Event Streams	Global pub-sub event streaming system	Free tier available	mapr.com/products/mapr-streams
Micro Focus	ArcSight Data Platform	Data collection & log mgmt platform	Available by request	software.microfocus.com/en-us/products/siem-data-collection-log-management-platform
Micro Focus	IDOL	ML, enterprise search, & analytics platform	N/A	software.microfocus.com/en-us/products/information-data-analytics-idol/overview

Company	Product	Product Type	Free Trial	Website
Micro Focus	Vertica	Distributed analtyics DB & SQL analytics on Hadoop	Free tier available	vertica.com/overview
Microsoft	SSRS	SQL Server reporting (server-side)	Free tier available	docs.microsoft.com/en-us/sql/reporting-services/create-deploy-and-manage-mobile-and-paginated
Microsoft	Azure Machine Learning Studio	Predictive analytics & ML development platform	12 months	azure.microsoft.com/en-us/services/machine-learning-studio
Microsoft	Power BI	BI platform	Free tier available	powerbi.microsoft.com
MicroStrategy	Advanced Analytics	Predictive analytics, native analytical functions, & data mining	Free tier available	microstrategy.com/us/products/capabilities/advanced-analytics
New Relic	New Relic Insights	Real-time app performance analytics	Available by request	newrelic.com/insights
NumFOCUS	Julia	Dynamic programming language for scientific computing	Open source	julialang.org
NumFOCUS	Matplotlib	Plotting library on top of NumPy (like parts of MATLAB)	Open source	matplotlib.org
NumFOCUS	NumPy	Mathematical computing library (i.e. multi-dimensional arrays, linear algebra, Fourier transforms) for Python	Open source	numpy.org
NumFOCUS	Pandas	Data analysis & modeling for Python	Open source	pandas.pydata.org
Objectivity	ThingSpan	Graph analytics platform w/ Spark & HDFS integration	60 days	objectivity.com/products/thingspan
OpenText	OpenText Big Data Analytics	Analytics & visualization w/ analyatics server	45 days	opentext.com/products-and-solutions/products/analytics/opentext-big-data-analytics
OpenTSDB Authors	OpenTSDB	Time-series DB on Hadoop	Open source	github.com/OpenTSDB/opentsdb/releases
Oracle	Big Data Discovery	Big data analytics & visualization platform on Spark	Available by request	oracle.com/big-data/big-data-discovery
Oracle	R Advanced Analytics for Hadoop	R interface for manipulating data on Hadoop	N/A	oracle.com/technetwork/database/database-technologies/bdc/r-advanalytics-for-hadoop/overview
Palantir	Foundry	Data integration platform	Demo available by request	palantir.com/palantir-foundry

DZONE.COM/GUIDES

Company	Product	Product Type	Free Trial	Website
Palantir	Gotham	Cluster data store, on-the-fly data integration, search, in-memory DBMS, ontology, & distributed key-value store	N/A	palantir.com/palantir-gotham
Panoply	Panoply	Data mgmt & analytics platform	Available by request	panoply.io
Panorama Software	Necto	BI, visualization, & data mgmt	Available by request	panorama.com/necto
Paxata	Paxata Adaptive Information Platform	Data integration, preparation, exploration, & visualization on Spark	14 days	paxata.com/product/paxata-adaptive-information-platform
Pepperdata	Pepperdata Cluster Analyzer	Big data performance analytics	Available by request	pepperdata.com/products/cluster-analyzer
Pivotal	Pivotal Greenplum	Open-source data warehouse & analytics	Open source	pivotal.io/pivotal-greenplum
Pivotal	Spring Cloud Data Flow	Cloud platform for building streaming & batch data pipelines & analytics	Open source	cloud.spring.io/spring-cloud-dataflow
Prognoz	Prognoz Platform	BI & analytics (OLAP, time series, & predictive)	Free tier available	prognoz.com/platform
Progress Software	DataDirect Connectors	Data integration: many-source, multi-interface, & multi-deployment	Available by request	progress.com/datadirect-connectors
Project Jupyter	Jupyter	Interactive data visualization & scientific computing on Spark & Hadoop	Open source	jupyter.org
Pyramid Analytics	BI Office	Data discovery & analytics platform	Free tier available	pyramidanalytics.com/bi-office
Qlik	Qlik Analytics Platform	Data visualization platform	Free tier available	qlik.com/us/products/qlik-analytics-platform
Qlik	Qlik Sense	Data visualization, integration, & search	Free tier available	qlik.com/us/products/qlik-sense
Qlik	QlikView	BI app platform	Free tier available	qlik.com/us/products/qlikview
Qubole	Qubole Data Service	Data engines for Hive, Spark, Hadoop, Pig, Cascading, Presto on AWS, Azure, & Google Cloud	30 days	qubole.com
Rapid7	InsightOps	Log mgmt & analytics	Available by request	logentries.com
RapidMiner	RapidMiner Studio	Predictive analytics workflow & model builder	30 days	rapidminer.com/products/studio
RapidMiner	RapidMiner Radoop	Predictive analytics on Hadoop & Spark w/R & Python support	30 days	rapidminer.com/products/radoop

Company	Product	Product Type	Free Trial	Website
Red Hat	Ceph	Distributed object & block store & file system	Open source	ceph.com/get
Red Hat	GFS2	Shared-disk file system for Linux clusters	Open source	access.redhat.com/documentation/en-us/red_hat_enterprise_linux
RedPoint	RedPoint Data Management	Data mgmt, quality, & integration (also on Hadoop)	14 days	redpoint.net/products/data-management-solutions
SAP	SAP HANA	In-memory, column-oriented, relational DBMS w/text search, analytics, stream processing, R integration, & graph processing	Free tier available	sap.com/products/hana.html
SAS	SAS Platform	Analytics, BI, data mgmt, & deep statistical programming	Demo available by request	sas.com/en_us/software/sas9.html
Sencha	InfoVis Toolkit	JavaScript visualization library	Open source	philogb.github.io/jit
Sisense	Sisense	Analytics, BI, visualization, & reporting	Demo available by request	sisense.com/product
Software AG	Terracotta DB	In-memory data mgmt, job scheduler, Ehcache implementation, & enterprise messaging	Available by request	terracotta.org
Splunk	Splunk Enterprise	Operational intelligence for machine-generated data	60 days	splunk.com/en_us/products/splunk-enterprise.html
Stitch	Stitch	ETL-as-a-service	Free tier available	stitchdata.com
StreamSets	Dataflow Performance Manager	Data mgmt & analytics platform	Demo available by request	streamsets.com/products/dpm
Sumo Logic	Sumo Logic	Log & time-series mgmt & analytics	30 days	sumologic.com
Tableau	Tableau Desktop	Visualization, analytics, & exloration (self-service, server, & hosted options)	14 days	tableau.com/products/desktop
Talend	Talend Data Fabric	Real-time or batch data mgmt platform	N/A	talend.com/products/data-fabric
Talend	Talend Open Studio	ELT & ETL on Hadoop w/open-source components	Open source	talend.com/download/talend-open-studio
Tamr	Tamr	Data mgmt, sanitation, analytics, & BI	3 days	tamr.com/product
Targit	Targit Decision Suite	BI, analytics, discovery front-end w/self-service options	30 days	targit.com/en/software/decision-suite

DZONE.COM/GUIDES

Company	Product	Product Type	Free Trial	Website
Teradata	Teradata	Data warehousing, analytics, data lake, SQL on Hadoop & Cassandra, big data appliances, R integration, & workload mgmt	Open source versions available	teradata.com/products
The R Foundation	R	Language & environment for statistical computing & graphics	Open source	r-project.org
Thoughtspot	Thoughtspot	Relational search engine	Demo available by request	thoughtspot.com/product
TIBCO	Jaspersoft	BI, analytics, ETL, data integration, reporting, & visualization	Free tier available	jaspersoft.com/business-intelligence-solutions
TIBCO	Spotfire X	Data mining & visualization	30 days	spotfire.tibco.com
TIBCO	TIBCO Data Virtualization	ETL, data virtualization, & integration platform	Demo available	tibco.com/products/data-virtualization
Treasure Data	Treasure Data	Analytics IaaS	Demo available by request	treasuredata.com
Trifacta	Trifacta Wrangler	Data wrangling, exploration, & visualization on Hadoop	Free solution	trifacta.com/products/wrangler
University of Waikato	Weka	ML & data mining for Java	Open source	cs.waikato.ac.nz/ml/weka
Unravel	Unravel	Predictive analytics & ML performance monitoring	Available by request	unraveldata.com/optimize-troubleshoot-and-analyze-big-data-performance
Waterline Data	Waterline Data	Data marketplace (inventory, catalogue w/self-service) on Hadoop	Demo available by request	waterlinedata.com/product-overview
Wolfram	Wolfram Language	Knowledge-based programming language w/many domain-specific libraries	Free tier available	wolfram.com/language
Workday	Workday Prism Analytics	Data prep, discovery, & analytics on Hadoop and Spark	N/A	workday.com/en-us/applications/prism-analytics.html
Xplenty	Cascading	Platform to develop big data apps on Hadoop	Open source	cascading.org
YCSB	YCSB	General-purpose benchmarking spec	Open source	github.com/brianfrankcooper/YCSB/wiki/Getting-Started
Yellowfin	Yellowfin	BI & data visualization	30 days	yellowfinbi.com/platform
Zaloni	Zaloni	Enterprise data lake mgmt	Demo available	zaloni.com/platform
Zoomdata	Zoomdata	Analytics, visualization, & BI w/ self-service on Hadoop, Spark, & many data stores	30 days	zoomdata.com

Diving Deeper Into Big Data

Zones

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

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

AI dzone.com/ai
The Artificial Intelligence (AI) Zone features all aspects of AI pertaining to machine learning, natural language processing, and cognitive computing. The AI Zone goes beyond the buzz and provides practical applications of chatbots, deep learning, knowledge engineering, and neural networks.

Twitter

 @data_nerd
 @caroljmcdonald
 @revodavid
 @karenchurch
 @mmarie
 @evdlaar
 @JohnDCook
 @GaelVaroquaux
 @victoria_holt
 @randal_olson

Refcardz

Temporal Data Processing
Download this Refcard to learn how to handle data that varies over time in relational databases using temporal tables.

Understanding Data Quality
This Refcard will show you the key places data derives from, characteristics of high-quality data, and the five phases of a data quality strategy that you can follow.

Getting Started With Apache Hadoop
Learn how Apache Hadoop stores and processes large datasets, get a breakdown of the core components of Hadoop, and learn the most popular frameworks for processing data on Hadoop.

Podcasts

IBM Analytics Insights
Learn the latest in big data and analytics, as well as the implications of big data analytics for the enterprise from a range of experts in varying industries.

Data Stories
Get insight into the blurry lines between art and infographics through extensive coverage of data visualization projects.

O'Reilly Data Show
Dive into the opportunities and techniques driving big data and data science, as well as topics like graph databases, open source, and machine learning.

Books

Data Analytics Made Accessible
In the 2019 edition of this highly rated book, dive into big data, artificial intelligence, data science, and R through real-world examples and use cases.

Data Science (MIT Press Essential Knowledge Series)
Get a concise introduction to the evolution of data science, how data science relates to machine learning, the ethical challenges that data science presents, and more.

Big Data: A Revolution That Will Transform How We Live, Work, and Think
Learn from two leading big data experts what big data is, how it is changing lives, and how we can protect ourselves from its hazards.

INTRODUCING THE

Open Source Zone

Start Contributing to OSS Communities and Discover Practical Use Cases for Open-Source Software

Whether you are transitioning from a closed to an open community or building an OSS project from the ground up, this Zone will help you solve real-world problems with open-source software.

Learn how to make your first OSS contribution, discover best practices for maintaining and securing your community, and explore the nitty-gritty licensing and legal aspects of OSS projects.

COMMITTERS & MAINTAINERS

COMMUNITY GUIDELINES

LICENSES & GOVERNANCE

TECHNICAL DEBT

Visit the Zone

BROUGHT TO YOU IN PARTNERSHIP WITH Flexera