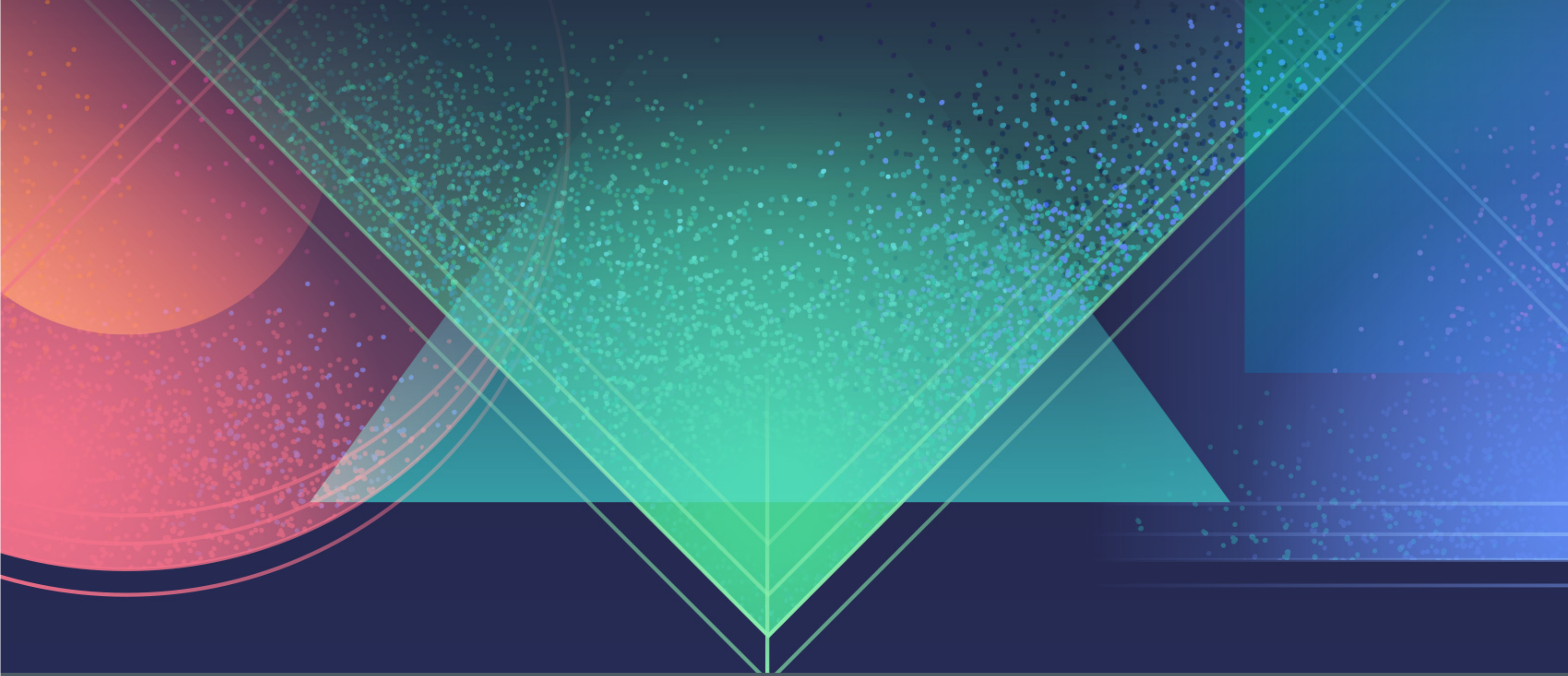
DZone

THE DZONE GUIDE TO

Big Data

Data Science & Advanced Analytics

VOLUME IV



BROUGHT TO YOU IN PARTNERSHIP WITH

DATAWATCH

HPCC SYSTEMS®
hpccsystems.com

InterSystems®
Health | Business | Government

Qubole

DEAR READER,

Looking back over the last 15 years, Big Data has often been framed as a problem, or perhaps an opportunity, that hasn't yet been fulfilled. The classic "3Vs" model Gartner first used to describe Big Data in 2001 is still the most widely accepted way of defining it; and while Volume, Velocity, and Variety tell us about the challenging aspects of dealing with this data, they do little to suggest the opportunities that have arisen from it. The 3V definition has evolved a little over the years, with some opting to include a fourth V — Veracity — and others pointing to machine and deep learning as complementary byproducts of the Big Data era. But, in essence, we're still framing things much the same way we have been since the beginning. So, what do we need to do to transform this outlook so that we no longer describe Big Data in terms of it being a problem, but rather a solution?

For starters, we need more people who understand how to work with data. It's been nearly five years since the Harvard Business Review referred to "data scientist" as the sexiest job of the 21st century, launching the budding career into buzzword fame while generating big expectations of Big Data professionals. But what exactly is a data scientist, or even data science for that matter? We know that there is massive potential in the massive amounts of data we've been gathering. We know that we can use analytics tooling to uncover this potential. We also know that we can wield algorithms to ingest this data and make sense of it in real time, even teaching software how to piece together its own insights and trigger automated processes based on this reasoning. We can feel this explosion of data-driven growth right around the corner, but it all hinges on having the right people and tools to execute the job, as well as a clear idea about what specific problems we want to solve with our data.

The truth is, we're still in the early stages. While our ability to collect and store data has greatly improved, the way we in which we use algorithms for analysis and the tools available to us are constantly evolving. We're still learning how to pair domain-specific experts, who understand a certain industry or have knowledge in a particular field, with the data implementation-specific experts, who understand how to build out the software to help with analysis. And alongside the progression of machine learning and natural language processing, we've seen a major push for more real-time analysis of data. With each step forward, new challenges are presented and old tooling is thrown out.

All of this may lead you to feel that we're still right where we started with Big Data. But, in reality, we've made great progress. Reading through the articles in this guide, you'll see that progress reflected in the advanced topics we're talking about—we'll show you how to use specific tools like Spark and R, while also introducing you to concepts like machine learning and data lake governance.

With any science, you must constantly question, experiment, analyze results, and adapt your methods accordingly. Data science is no different; there are lots of brilliant minds out there working on the 3Vs problem, refining their knowledge and practices for working with data while uncovering analytics to bring about yet another V — Value.

BY MICHAEL THARRINGTON

CONTENT AND COMMUNITY MANAGER, DZONE

TABLE OF CONTENTS

PRODUCTION
CHRIS SMITH
DIRECTOR OF PRODUCTION
ANDRE POWELL
SR. PRODUCTION COORDINATOR
G. RYAN SPAIN
PRODUCTION PUBLICATIONS EDITOR
ASHLEY SLATE
DESIGN DIRECTOR

MARKETING
KELLET ATKINSON
DIRECTOR OF MARKETING
LAUREN CURATOLA
MARKETING SPECIALIST
KRISTEN PAGÀN
MARKETING SPECIALIST
NATALIE IANNELLO
MARKETING SPECIALIST
MIRANDA CASEY
MARKETING SPECIALIST

EDITORIAL
CAITLIN CANDELMO
DIRECTOR OF CONTENT + COMMUNITY
MATT WERNER
PUBLICATIONS MANAGER
MICHAEL THARRINGTON
CONTENT + COMMUNITY MANAGER
MIKE GATES
SR. CONTENT COORDINATOR
SARAH DAVIS
CONTENT COORDINATOR
TOM SMITH
RESEARCH ANALYST
JORDAN BAKER
CONTENT COORDINATOR
ANNE MARIE GLEN
CONTENT COORDINATOR

BUSINESS
RICK ROSS
CEO
MATT SCHMIDT
PRESIDENT + CTO
JESSE DAVIS
EVP & COO
MATT O'BRIAN
DIRECTOR OF BUSINESS DEV.
ALEX CRAFTS
DIRECTOR OF MAJOR ACCOUNTS
JIM HOWARD
SR ACCOUNT EXECUTIVE
JIM DYER
ACCOUNT EXECUTIVE
ANDREW BARKER
ACCOUNT EXECUTIVE
CHRIS BRUMFIELD
SALES MANAGER
ANA JONES
ACCOUNT MANAGER
TOM MARTIN
ACCOUNT MANAGER

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

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

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

SPECIAL THANKS to our topic experts, Zone Leaders, trusted DZone Most Valuable Bloggers, and dedicated users for all their help and feedback in making this guide a great success.

Executive Summary

BY MATT WERNER
CONTENT AND COMMUNITY MANAGER, DZONE

It seems like every other month, a news piece comes out that reminds everyone of just how much data is collected by the tools and websites they use every day. Social media sites have transformed from dorm room experiments into wealthy empires due to the potential value of the data they collect, marketers use data to target their ads to niche audiences that may be more likely to buy, and Netflix looks at your viewing history to decide which original content they should invest in. For a few years, the challenge of dealing with the 3 Vs of Velocity, Volume, and Variety was, “How do we store this?” However, the biggest question that enterprises must answer now is, “How do we use the data we’ve collected?” We surveyed 734 IT professionals with experience in data science, data architecture, data engineering, data mining, and visualization to learn about how they’re transforming their data into better decisions and better software.

OPEN SOURCE PUSHES DATA SCIENCE FORWARD

DATA 71% of respondents use open-source tools for data science, while only 16% use licensed or commercial tools. Of particular interest is the adoption of Apache Spark, which has increased to 45% as compared to 31% last year.

IMPLICATIONS Spark is only a three-year-old tool, but it has seen a huge jump in popularity after an already surprising adoption rate in 2016. Another tool of note is TensorFlow, which is already being used by 17% of respondents after being released only a year and a half ago by Google. The rapid growth of these tools suggests that in the Big Data space, it’s very possible for open-source tools to catch on and become established quickly in both personal development projects and enterprise applications.

RECOMMENDATIONS Developers should stay on top of cutting-edge, open-source tools that could help make their jobs easier. Considering how quickly technologies like Spark and TensorFlow have been adopted by a significant number of developers, failing to adapt to new technologies could mean unnecessary work down the line for data scientists. For an example of how to use DataFrames for analytics in a Spark environment, refer to Frank Evans’ article on page 12.

HADOOP STILL LEADS THE PACK

DATA 65% of those with data engineering experience have used Apache Hadoop. 47% use Yarn for cluster resource management, 62% use Apache Zookeeper (once part of Hadoop, but now a standalone project) for node coordination, and 55% use Hive (built on Hadoop) for data warehousing.

IMPLICATIONS Since its release in 2011, Apache Hadoop has been a major factor in the growth of Big Data technology and use thanks to its ability to store and process data using MapReduce. Almost all of the various tools listed above are either part of or built on top of Hadoop, and all except for Yarn were used by a majority of respondents. When it comes to storing and processing data, Hadoop is the standard-bearer right now.

RECOMMENDATIONS As of this writing, there are nearly 12,000 jobs on LinkedIn looking for experience with Hadoop. There is a clear demand for developers and data scientists with relevant experience — so experimentation with Hadoop and technologies built on it will be useful for pursuing future careers in Big Data. Also consider exploring Apache Spark, which was developed to overcome the limitations of MapReduce, and becoming familiar with its various components like Spark SQL, GraphX, MLib, and Spark Streaming.

DATABASES AND PRODUCTION

DATA MySQL was used by 60% of respondents in both production and non-production environments. MongoDB was used by 47% of users in production and 48% outside of production. PostgreSQL was used by 41% of developers in production and 40% outside of production. With commercial databases, Oracle was used by 53% professionals in production and 38% outside of production, while MS SQL Server was used by 45% of people in production and 34% of people outside of production.

IMPLICATIONS While MySQL, MongoDB, and PostgreSQL are all popular solutions both in and out of production, Microsoft SQL Server and Oracle are not as popular outside of production, which is likely due to the cost of a license. Other databases, like SQLite, are being experimented with outside of production (28%) rather than in production (17%). NoSQL databases are steadily becoming more popular; a majority of those with data science experience reported using them (56%).

RECOMMENDATIONS While neither are popular outside of production environments, several enterprises are still using Oracle and MS SQL server, so it’s still worthwhile to know these tools for larger companies or legacy applications. SQLite seems to be gaining steam outside of production, which could translate to increased use outside of production down the road (depending on developers’ experiences with it). Regardless of the environment, MySQL is still the de facto database of choice for most developers and enterprises.

Key Research Findings

BY G. RYAN SPAIN
PRODUCTION COORDINATOR, DZONE

In the 2017 DZone Big Data survey, we collected responses from 734 IT professionals who identified as having experience with the following Big Data areas: 252 in data science; 358 in data architecture; 289 in data engineering; 278 in data mining; and 316 in data visualization. Demographics of these respondents include the following:

- 17% work at organizations with more than 10,000 employees; 25% at organizations with 500 to 10,000 employees; and 19% with between 100 and 500 employees.
- 35% work at organizations based in Europe, and another 35% work at organizations based in North America.
- 39% have 15 years of experience or more as an IT professional; 76% have 5 years of experience or more.
- 33% identify as a developer/engineer; 22% identify as a developer team lead.
- 77% work at organizations that use Java; 69% at organizations that use JavaScript; and 48% at organizations that use Python. 47% use Java primarily at work, and 15% use Python primarily.

DATA SCIENCE

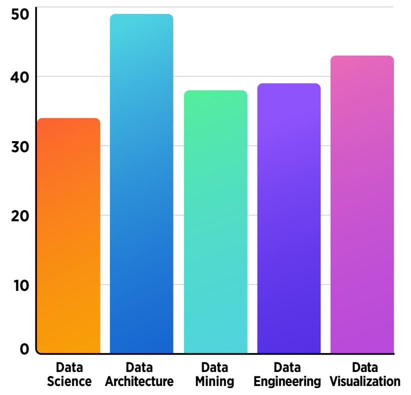
Since at least 2010, data science has relied heavily on open-source languages, tools, libraries, and frameworks. The flexibility and community around these tools help them grow in vastly varied ways, making the near-endless use cases for data science seem somewhat less expansive and unexplored. Our survey this year showed that developers experienced in data science continue to follow this trend, with 71% of respondents saying they use open-source data science tools at work, versus only 16% who use licensed tools. Another staple among data scientists is the relational database: 70% of our data science respondents said they use relational databases in their work. Still, NoSQL databases seem to continue to prove valuable in some use cases, and a majority (56%) of our data science respondents said they use non-relational databases. As far as specific languages go, Python and R both came out on top for data science, with the majority of respondents using one or both of these. Python has now taken the top spot regarding data science languages, gaining a 63% response over R's 61%, and while this margin isn't statistically significant, it does match with other trends seen in data science over the past few years, as Python's growth has been outpacing R's.

Apache Spark has also seen increased adoption again this year, up to 45% from the 31% usage we saw in last year's survey, which is impressive considering its relative newness. Still, its components, like Spark SQL and MLlib, which help with manipulating and analyzing distributed data sets such as those on Hadoop, have made it increasingly easier to gather valuable information from the kinds of new systems that have been required to process and store the ever-growing amount of data that applications generate.

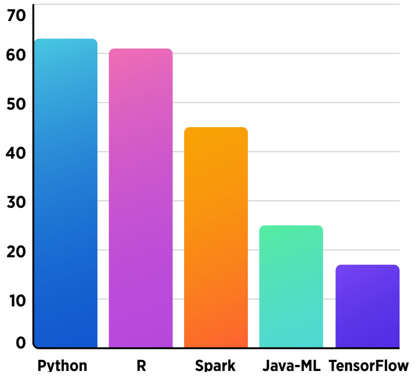
DATA ARCHITECTURE & ENGINEERING

Of the five "Big Data" categories of experience we asked of our respondents, the most popular choice was data

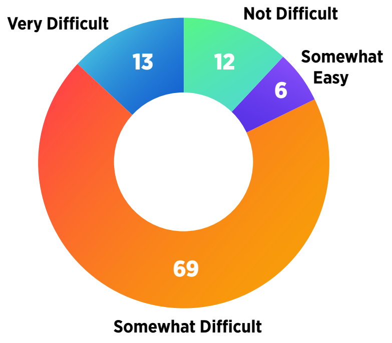
In which of the following areas of Big Data do you have experience?



What languages/libraries/frameworks do you use for data science and machine learning?



How would you describe the process for getting the necessary data to create models from your source systems?



architecture: 49% of total survey respondents said they had experience in this field. Regarding database usage in and out of production, responses were mostly consistent: MySQL was used by 60% of respondents in production and in non-production environments; MongoDB was used by 47% of developers in production and 48% outside of production; and PostgreSQL was used by 41% of users in production and 40% outside of production. But a few databases had considerable, if expected, gaps between production and non-production usage. For example, the two most popular commercial databases, Oracle and MS SQL Server, each dropped considerably in non-production use: 53% used Oracle in production to 38% use in non-production; and MS SQL Server from 45% to 34%. SQLite, on the other hand, popular for its lightweight and embedded nature, was more likely to be used in non-production environments (28%) instead of in production (17%).

Many other categories of tools for data architecture were fragmented. While Spark Streaming was popular among computational frameworks for streaming data (38%), and Kafka for data ingestion (54%), no other framework or system had more than a 25% response rate. Honorable mentions go to GraphX for garnering 24% of responses in the iterative graph processing category, and to protocol buffers, which are used by 20% of data architecture respondents for data serialization.

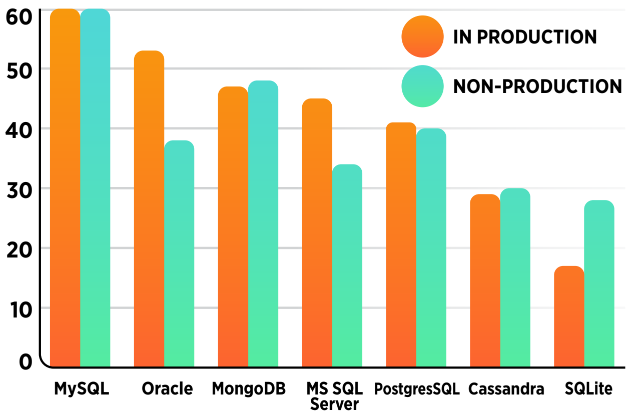
Respondents with experience in data engineering seemed to have some clear favorites (Big Data classics, in their own right). As far as file systems go, Hadoop's popularity was apparent, with 65% of data-engineering respondents using it (only 5% of respondents who did not answer "Not applicable" also did not say they use HDFS). 47% of respondents said they use Hadoop's Yarn for cluster resource management. Apache ZooKeeper, once a part of Hadoop, now its own top-level Apache project, had a 62% response rate for node coordination. And Apache Hive, built on Hadoop, had a 55% response rate for data warehousing. 60% of data engineering respondents said they do not use any particular tool for benchmarking cluster performance, but those who did predominately favored homegrown solutions (27%).

DATA MINING & VISUALIZATION

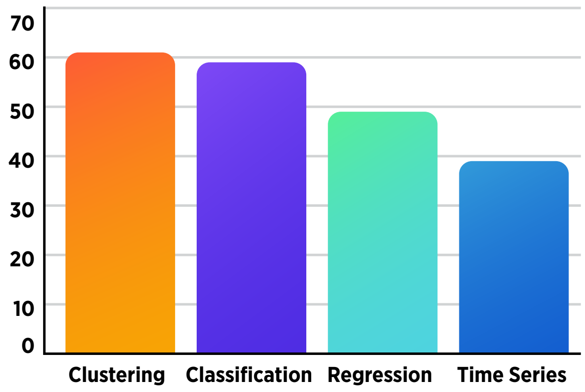
Python won out over R again with regards to data mining, with 57% of data mining-experienced respondents using Python for mining, as opposed to 48% who said they use R. The majority of these respondents (69%) said that it is "somewhat difficult" to attain adequate training data from their source systems, with other respondents being fairly fragmented among the options of "very difficult" (13%), "somewhat easy" (6%), and "not difficult" (12%). In order to judge the performance of a data mining classifier, most respondents looked at the accuracy (52%) or balanced accuracy (22%) of the result set, followed by receiver operating characteristics (ROC). Most miners said they used clustering (61%) and classification (59%) algorithms, followed by regression (49%), and finally time series (39%).

In the visualization category, a handful of tools stood out above the rest. Regarding overall visualization tooling, 42% of respondents said they use D3.js, the front-runner, 29% of respondents said they use Tableau; and 28% said they use Chart.js. Tableau was also popular specifically for "Big Data" visualization (39%) and real-time visualization (39%); Chart.js was recommended for "Big Data" and open-source visualization (32% and 37%, respectively); and D3.js was most popular again in the open source category (53%). or high-risk software (i.e. software in which failure could lead to significant financial loss or loss of life) were much more likely to regularly design for parallel execution, with over half of these respondents (54% each) answering this question positively.

What database management systems do you use?



What data mining algorithms do you use at work?



Data Lake Governance Best Practices

BY PARTH PATEL FIELD ENGINEER, ZALONI
GREG WOOD FIELD ENGINEER, ZALONI
AND ADAM DIAZ DIRECTOR OF FIELD ENGINEERING, ZALONI

QUICK VIEW

01 Without proper application of best practices any type of storage can become unmaintainable.

02 Data Governance is the key to avoiding a data swap with Hadoop.

03 The proper use of zones and associated permissions is first key to an organized Data Lake.

04 Automation of activities like data quality, data lifecycle, and data privacy provide ongoing cleansing and movement of the data in your lake.

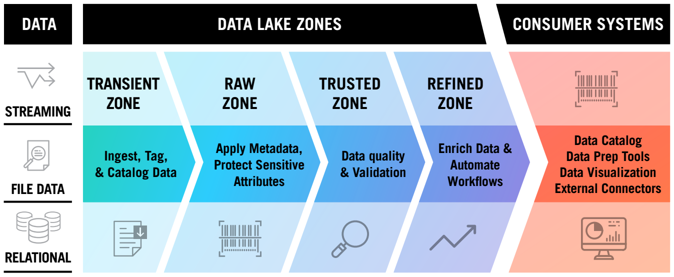
Data Lakes are emerging as an increasingly viable solution for extracting value from Big Data at the enterprise level, and represent the logical next step for early adopters and newcomers alike. The flexibility, agility, and security of having structured, unstructured, and historical data readily available in segregated logical zones brings a bevy of transformational capabilities to businesses. What many potential users fail to understand, however, is what defines a usable Data Lake. Often, those new to Big Data, and even well-versed Hadoop veterans, will attempt to stand up a few clusters and piece them together with different scripts, tools, and third-party vendors; this is neither cost-effective nor sustainable. In this article, we'll describe how a Data Lake is much more than a few servers cobbled together: it takes planning, discipline, and governance to make an effective Data Lake.

ZONES

Within a Data Lake, zones allow the logical and/or physical separation of data that keeps the environment secure, organized, and Agile. Typically, the use of 3 or 4 zones is encouraged, but fewer or more may be leveraged. A generic 4-zone system might include the following:

- **Transient Zone** – Used to hold ephemeral data, such as temporary copies, streaming spools, or other short-lived data before being ingested.
- **Raw Zone** – The zone in which raw data will be maintained. This is also the zone where sensitive data must be encrypted, tokenized, or otherwise secured.
- **Trusted Zone** – After Data Quality, Validation, or other processing is performed on data in the Raw Zone, it becomes the "source of truth" in this zone for downstream systems.
- **Refined Zone** – Manipulated and enriched data is kept in this zone. This is used to store output from tools like Hive or external tools that will write into to the Data Lake.

This arrangement can be adapted to the size, maturity, and unique use cases of the business as necessary, but will leverage physical separation via exclusive servers/clusters, logical separation through deliberate structuring of directories and access privileges, or some combination of both. Visually, this architecture is similar to the one below.



Establishing and maintaining well-defined zones is the most important activity to create a healthy Lake, and promotes the rest of the concepts in this article. At the same time, it is important to understand what zones do not provide—namely, zones are not a Disaster Recovery or Data Redundancy policy. Although zones may be considered in DR, it's still important to invest in a solid underlying infrastructure to ensure redundancy and resilience.

LINEAGE

As new data sources are added, and existing data sources updated or modified, maintaining a record of the relationships within and between datasets becomes more important. These relationships might be as simple as a renaming of a column, or as complex as joining multiple tables from different sources, each of which might have several upstream transformations themselves. In this context, lineage helps to provide both traceability to understand where a field or dataset originates and an audit trail to understand where, when, and why a change was made. This may sound simple, but capturing details about data as it moves through the Lake is exceedingly hard, even with some of the purpose-built software being deployed today. The entire process of tracking lineage involves aggregating logs at both a transactional level (who accessed the data and what did they do?) and at a structural or filesystem level (what are the relationships between datasets and fields?). In the context of the Data Lake, this will include any batch and streaming tools that touch the data (such as MapReduce and Spark), but also any external systems that may manipulate the data, such as RDBMS systems. This is a daunting task, but even a partial lineage graph can fill the gaps of traditional systems, especially as new regulations such as GDPR emerge; flexibility and extensibility is key to manage future change.

DATA QUALITY

In a Data Lake, all data is welcome, but not all data is equal. Therefore, it is critical to define the source of the data and how it will be managed and consumed. Stringent cleansing and data quality rules might need to be applied to data that requires regulatory compliance, heavy end-user consumption, or auditability. On the other hand, not much value can be gained by cleansing social media data or data coming from various IoT devices. One can also make a case to consider applying the data quality checks on the consumption side rather than on the acquisition side. Hence, a single Data Quality architecture might not apply for all types of data. One has to be mindful of the fact that the results used for analytics could have an impact if the data is 'cleansed.' A field-level data quality rule that fixes values in the datasets can sway the outcomes of predictive models as those fixes can impact the outliers. Data quality rules to measure the usability of the dataset by comparing the 'expected vs. received size of the dataset' or 'NULL Value Threshold' might be more suitable in such scenarios. Often the level of required validation is influenced by legacy restrictions or internal processes that already are in place, so it's a good idea to evaluate your company's existing processes before setting new rules.

PRIVACY/SECURITY

A key component of a healthy Data Lake is privacy and security, including topics such as role based access control, authentication, authorization, as well as encryption of data at rest and in motion. From a pure Data Lake and data management perspective the main topic tends to be data obfuscation including tokenization and masking of data. These two concepts should be used to help the data itself adhere to the security concept of least privilege. Restricting access to data also has legal implications for many businesses looking to comply with national and international regulations for their vertical. Restriction access takes several forms; the most obvious is the prodigious use of zones within the storage layer. In short, permissions in the storage layer can be configured such that access to the data in its most raw format is extremely limited. As that data is later transformed through tokenization and masking (i.e., hiding PII data) access to data in later zones can be expanded to larger groups of users.

DLM

Enterprises must work hard to develop the focus of their data management strategy to more effectively protect, preserve, and serve their digital assets. This involves investing in time and resources to fully create a lifecycle management strategy and to determine whether to use a flat structure, or to leverage tiered protection. The traditional premise of a Data Lifecycle Management was based around the fact that data was created, utilized, and then archived. Today, this premise might hold true for some transactional data, but many data sources now remain active from a read perspective, either on a sustained basis or during semi-predictable intervals. Enterprises that know and understand the similarities and differences across their information, data and storage media, and are able to leverage this understanding to maximize usage of different storage tiers, can unlock value while removing complexity and costs.

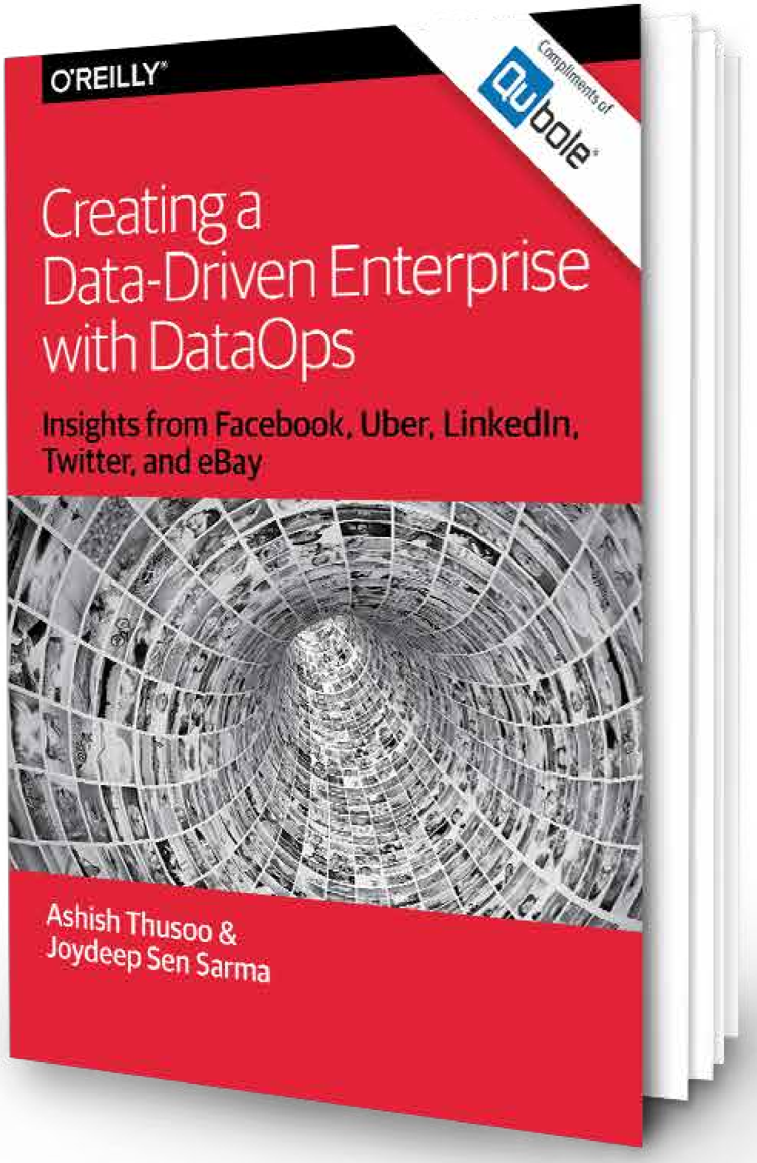
CONCLUSION

Much like relational databases in the days of their infancy, some implementations of Hadoop in recent years have suffered from a lack of best practices. When considering using Hadoop as a Data Lake there are many best practices to consider. Utilizing zones and proper authorization as a part of a data workflow framework provides a highly scalable and parallel system for data transformation.

PARTH PATEL is a Field Engineer at Zaloni. Previously, Parth has worked as a Systems Engineer and as an entrepreneur and member of the National Advisory Board of Dunkin Brands. He can be reached via email or LinkedIn.

GREG WOOD is a Field Engineer at Zaloni, and previously held positions at companies focusing on analytics and system design. He is happy to join the world of Data Lakes, and can be reached via email or LinkedIn.

ADAM DIAZ is a longtime technologist working in the software industry for roughly twenty years. Adam has spent many years with Hadoop enabling high performance and parallel computing solutions at scale at companies like SAS, Teradata, Hortonworks, and MapR. Adam is the Director of Field Engineering at Zaloni, where he enables Data Lake solutions utilizing Bedrock. He can be reached via email or LinkedIn.


O'REILLY
Compliments of
Qubole
Creating a Data-Driven Enterprise with DataOps
Insights from Facebook, Uber, LinkedIn, Twitter, and eBay
Ashish Thusoo & Joydeep Sen Sarma

How Did LinkedIn, Twitter, and eBay Harness Big Data to Create World-Class Companies?

Simple: they each built a modern big data platform - one that democratizes their data and provides real-time, always-on insights to everyone across the business. Read their stories in this exciting new book that takes a deep look into what they did, how they did it, and what they learned along the way.

The book, *Creating a Data-Driven Enterprise with DataOps*, published by O'Reilly Media, Inc., was written by myself, Ashish Thusoo, and Qubole co-founder, Joydeep Sen Sarma. We were inspired by what we have been able to accomplish here at Qubole, enabling our customers to make data accessible across organizations for increased insights at every level, that we decided to share our story. As we began to write, we reached out to our counterparts in the industry and found that many of them were willing to share their tales from the front lines of the data wars, and this book is the outcome of that collaboration.

We all know that the effective use of Big Data is the key to gaining a competitive advantage and outperforming the competition. The reality is that today's Big Data goals are not any different than the rest of your information management goals – it's just that now the economics and technology are mature enough to process and analyze this data. But there is one catch: how do you change your organization to really become a data-driven culture, and how do you make the data team the core of access to information and not a barrier to analytic success?

We hope you read our stories of building self-service data infrastructures and are inspired to create your own data-driven enterprises with DataOps.

Register here for a copy of the ebook.

WRITTEN BY ASHISH THUSOO
SR DIRECTOR, TECHNOLOGY DEVELOPMENT, **QUBOLE**

PARTNER SPOTLIGHT

Qubole Data Service By Qubole

The Qubole Data Service is the leading Big Data-as-a-Service platform for the cloud.

CATEGORY

Big Data Platform

NEW RELEASES

Continuously updated SaaS

OPEN SOURCE

No

CASE STUDY

The Qubole Data Service (QDS) provides data teams the control, automation, and orchestration to deploy and run any kind of Big Data workload, using any analytic engine for an unlimited number of users on their choice of Amazon AWS, Microsoft Azure, Oracle Bare Metal Cloud, or Google Cloud Platform. It supports a full range of data processing engines: Hadoop, Spark, Presto, Hive and more. Analysts and Data Scientists use the Workbench features to run ad hoc queries and machine learning algorithms, while data engineers use QDS for ETL and other batch workloads. QDS is the only cloud-native Big Data platform, and provides cluster management through a control panel or API to make administrative tasks easy, performance optimization via auto-scaling, and cost control through policy-based spot instance support. QDS is used by firms such Pinterest, Oracle, and Under Armour to process over 600 petabytes of data monthly.

STRENGTHS

- Cloud-native platform that can run on Amazon AWS, Microsoft Azure, Oracle Bare Metal Cloud, and Google Cloud Platforms
- Supports most big data processing engines: Hadoop, Spark, Presto, Hive, Pig, Airflow, and more
- Easy to deploy SaaS, which requires no hardware or software installation so you can run workloads immediately
- Optimizes cost and performance with auto-scaling and spot-instance features
- Automates cluster management through policy-based configuration management

NOTABLE CUSTOMERS

- Oracle
- Autodesk
- Under Armour
- Lyft
- Flipboard

WEBSITE www.qubole.com

TWITTER @Qubole

BLOG qubole.com/big-data-blog

How Smart is Your Big Data Platform?

BY **ARJUNA CHALA**

SR. DIRECTOR OF INNOVATION AND EMERGING TECHNOLOGY, **LEXISNEXIS RISK SOLUTIONS**

QUICK VIEW

01 Batch processing of big data is too slow to accommodate applications requiring real-time performance.

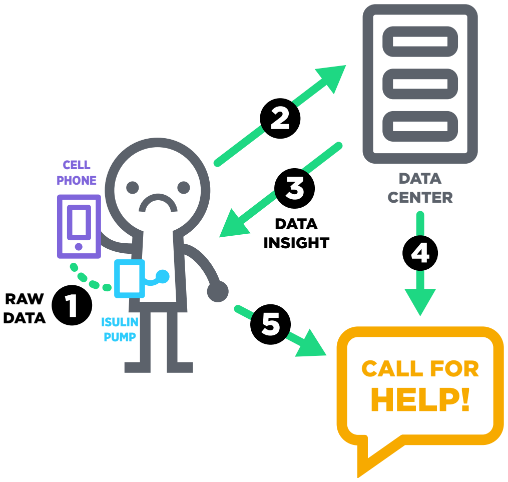
02 Business analytics platforms supporting smart data allow analytics to happen at the endpoint for faster implementation.

03 Smart data leverages local compute resources to power analytics, enabling connected endpoint devices to improve performance through machine learning.

Big Data, business analytics platforms, and IoT have upended the business and IT worlds. Organizations have realized the significant financial benefits to be gained by analyzing the massive amounts of data their systems generate to see what improvements or changes can be made to their product or processes. While the benefits of data analysis are compelling, the exponential growth in the amount of data generated by connected systems makes handling that data a complex proposition.

With organizations generating terabytes of data every day, the time required to upload, categorize, clean, and analyze that data to gain any insight from it can be significant. In light of this, organizations have implemented batch data processing schedules in which new data from client devices is uploaded to the data center for analysis at scheduled intervals rather than continuously. The data is then analyzed, the appropriate action identified, and instructions to implement that action are sent back to the device. Batch data processing can take anywhere from a few hours to a few days to complete. In applications requiring faster performance, real-time scoring of data based on identifying a few attributes and uploading them for immediate analysis can shorten the time required between data analysis and action, but accelerating the process is not without cost. Data scoring is typically limited to attributes gathered in a single transaction, so it can't incorporate any findings from previous transactions.

While batch processing and real-time scoring are sufficient for applications that don't require real-time performance adjustments or machine learning to automate those adjustments, there are vertical markets (healthcare, for example) where decisions need to be made and implemented in seconds to avoid potentially catastrophic problems. Additionally, after the data is analyzed and the analytics platform recommends a change to a client device or process, if that change requires a human to implement it, the time between collecting data and taking action based on that data can grow even longer.



In this IoT-enabled insulin pump example, (1) data is uploaded from the device via a Bluetooth connection to a smartphone which then (2) sends the data via WiFi or cellular connection to the data center for analysis. After analysis, the data center sends new instructions back to the device (3). If the datacenter receives data indicating a health emergency requiring immediate assistance, it can then call 911 (4) and/or instruct the user's phone to also call 911 (5).

DELIVERING ANALYSIS IN REAL-TIME

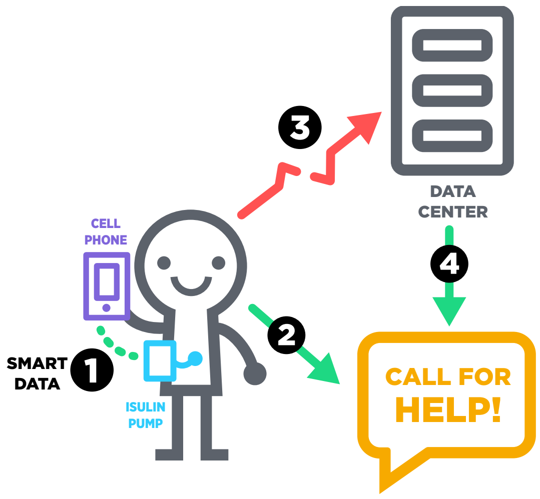
The need to deliver real-time analytics becomes more challenging when the connected nature of IoT is added to the mix. Specifically, endpoint devices in an IoT network are not always guaranteed to be connected to the Internet. Imagine a combination blood sugar monitor/personal insulin pump with a built-in cellular or WiFi modem to provide connectivity. Throughout the day, the device measures the user's blood sugar and uploads the data to a healthcare service provider's data center for analysis. When data from the device is analyzed, if blood sugar levels are too high, the healthcare provider's analytics platform can instruct the pump to administer the correct injection dosage. But what if the insulin pump is unable to connect? What if the user is traveling and can't get the pump online due to poor signal quality or technical issues? The insulin pump wouldn't be able to provide real-time blood sugar updates to the data center or receive instructions about the dosage and time until the next injection. It's easy to see the significant health risks such a scenario presents to patients, as well as the potential liability exposure the device OEM and/or healthcare provider could face.

In the world of real-time Big Data analytics, changes to devices and processes need to happen quickly, locally, and automatically. Furthermore, automation can't simply be a pre-defined list of rules that instructs an endpoint device to take action when a condition is met. Trying to anticipate any potential problem and developing a rule to handle it locally requires too much time and effort. So what's an organization to do?

SMART DATA EMPOWERS THE ENDPOINT

The solution to the problem is smart data. The "smart" adjective has been used to describe a host of connected devices and services, but what does it mean when applied to data? Let's define smart data as digital information that is formatted so it can be acted upon at the endpoint before being sent to a downstream analytics platform for further consolidation and analytics. Furthermore, the action taken is not merely a pre-defined list of instructions already in place that is implemented automatically when certain parameters are met. Rather, the data analysis (artificial intelligence) that determines what action to take is performed by the endpoint itself, using its own processing resources and machine learning to identify the proper course of action. Another way to look at smart data is to see how similar it behaves in comparison to the human brain. The human brain is constantly learning from the data it receives over time, and can determine the proper course of action when it sees a familiar pattern.

Let's apply smart data to our hypothetical insulin pump to see how it could enable the pump to self-correct without the need to access a centralized data pool for analysis and action. The pump detects elevated blood sugar levels in the patient, and recognizes that this is a critical situation. In addition to contacting the data center, the device immediately contacts the patient's doctor using the patient's cell phone, and notifies the patient and people around him.



With Smart Data (1), if the medical device is unable to connect to the datacenter (2), the endpoint can conduct analysis on the smart data independently (3) to determine if action is required (in this case, calling 911). Additionally, the datacenter can also call for help to notify the patient or 911 if it detects an abnormal reading or a lost connection (4).

In addition, the device would recommend the right course of action. For example, upon determining the patient's blood sugar is too high, the pump could determine on its own that a dose of insulin should be injected and extra blood sugar tests conducted over the next few hours to confirm that levels return to normal. This analysis and decision-making process would run in the background, and require no involvement from the patient, device OEM, or healthcare provider. Over time, if blood sugar levels continue to fluctuate, it could be an indicator of a more serious health problem. The device could then alert the patient and advise them to seek immediate medical attention.

So while Big Data, business analytics, and IoT have revolutionized our relationship with data, the benefits they provide will remain beyond the reach of organizations requiring real-time analytics until analysis and decisions can happen at the endpoint. Smart data makes that possible.

ARJUNA CHALA is the Sr. Director of Technology Development 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.

Using DataFrames for Analytics in the Spark Environment

BY **FRANK D. EVANS**

DATA SCIENTIST, **EXAPTIVE**

QUICK VIEW

01 A DataFrame is a clean and clear way to maintain the organization of structured data.

02 Spark DataFrames are a great way to maintain the relational nature of your data as you work with it.

03 Spark SQL on DataFrames lets you interact directly with your data using the power of the Spark engine.

Making the jump from static data to some kind of actionable model is often the real trick of any analytic technique. As data gets bigger and bigger, just handling the "small" stuff can become its own class of problem. Introducing intelligent organizational structure as early as possible can help frame how problems should be approached — as well as moving the small problems to the side where a developer or analyst can focus on the substance of the real problems they want to solve. This article will look at the organizational structure of DataFrames and how they can be leveraged within the Apache Spark environment to fully take advantage of the power of the Spark engine and ecosystem while staying organized within familiar frames of reference.

DATAFRAMES IN GENERAL

A DataFrame is analogous to a table in a relational database or spreadsheet. It has rows and columns, and data in the cells at the intersection of each row/column. A row is a single set of data that all conceptually goes together. A column represents a certain aspect or attribute of each row, and is often composed of data of the same data type across each row. Resultantly, any given portion of data can be coordinated via the row and column in which it lives.

A DataFrame works the same way. While data is often thought of as a collection of many individual items or objects, a DataFrame is a single object of organization around multiple pieces of data. Because of this, it can be worked with as a cohesive and singular object. One DataFrame can be filtered into a subset of its rows. It can create a new column based on operations on existing columns. Multiple DataFrames can be combined in different ways to make new DataFrames.

DataFrames have long been a staple in other analytical frameworks like R and Python (via Pandas). In those environments, organizing data into a DataFrame opens up a whole host of direct (reference indexing, native methods) and indirect (SQL, machine learning integration) functionalities on data to be used in an analytical pipeline. Spark is no different; DataFrames are now a first-class citizen of the Apache Spark engine.

DATAFRAMES IN SPARK

At the heart of Spark is a dataset. All Spark operations take place against a distributed dataset. A DataFrame in Spark is just an additional layer of organization overlaid on a dataset. In a DataFrame, the dataset contains named columns. Just like a table in a relational database or a spreadsheet, a given column can now be indexed and referenced by its name — either directly in programming a Spark job or indirectly when using the data in another component of the Spark ecosystem like a machine learning function or SQL.

GETTING DATA INTO A DATAFRAME

If data is already stored and is to be loaded into a Spark session from an existing relational style storage (such as Hive, Impala, or MySQL tables), then a DataFrame structure can be inferred that will clone the data's existing structure. If not, then the schema of the DataFrame can be specifically mapped to an existing dataset by providing the column name and datatype as metadata for each attribute of the underlying data.

Spark has compatibility for other common data formats that can be logically mapped to a DataFrame easily. Common flat formats like JSON and CSV can be read directly. More complex formats that preserve metadata like Avro and Parquet can also be read into a Spark DataFrame without having to re-specify the organizational structure of the underlying data.

SPARK SQL AND DATAFRAMES

The biggest instance of similarity between a Spark DataFrame and a table in a relational table is the ability to write SQL directly against the data. Once a dataset is organized into a DataFrame, Spark SQL allows a user to write SQL that can be executed by the Spark engine against that data. The organizational structure that the DataFrame creates allows the same powerful execution engine to be used and to take advantage of built-in processing optimizations. To the user, the data can be interacted with as though in a database, but with the full parallelizable capabilities of the Spark environment.

DATAFRAME METHODS AND PIPELINE CHAINING

One of the most powerful capabilities of traditional DataFrames is creating pipelines of chained operations on data that are arbitrarily complex. Unlike SQL, which is compiled into a single set of operations, chained pipelines of operations make data operations more intuitive and natural to conceive without having to nest multiple subqueries over and over.

A normal complex SQL statement normally has up to one of each major operation type: a filter, a group by, a windowed function, an aggregation, and some joins. However, it can take some mental gymnastics to think of a problem in this way, and may take multiple SQL statements and sub-queries to accomplish. A DataFrame has a number of native methods accessible. And since the output of many native methods on a DataFrame is another DataFrame — they can be chained together into arbitrarily complex pipelines.

Imagine a processing pipeline of functions that started with a filter, then aggregated those results, then joined in another data set based on that aggregation, then reordered the results of that and carried out a window function…and so on. The capabilities are practically endless and the logic of the analysis is free to follow the thought process of the analyst. Yet still, under the hood, the Spark engine will process and optimize the underlying operations necessary to accomplish this arbitrarily complex series. This combination becomes an incredibly powerful tool in the box of an analyst to bend data to their will.

UNDER THE HOOD

While a normal Spark RDD dataset is a collection of individual data elements, each of which may be constructed by one or many objects, a DataFrame is, at its core, a single object in memory reference space. It serves both as a cohesive object as well as a container object for the elements that make it up. This means that full parallelization is still a first-class feature. Most operations can still be processed across an arbitrarily large number of nodes in a cluster, meaning the size of a DataFrame has no practical limit. However, a user still gets the good aspects of treating a DataFrame like a singular object, similar to using R or Python.

Lower-level than the DataFrame as a whole is the Row object that makes up each cohesive component of a DataFrame. Like a row in a relational database, the Row object in a Spark DataFrame keeps the column attributes and data available as a cohesive single object for given "observation" of data. Rows have their own methods and magic methods, as well; for example, a Row can convert itself into a hashmap/dictionary and can evaluate whether it contains a given column or not.

DIFFERENCES FROM TRADITIONAL DATAFRAMES

This is the root of biggest difference between a DataFrame in R/Python Pandas and a Spark DataFrame from a technical perspective. A traditional DataFrame is a columnar structure. For a DataFrame having 10 rows of data with 3 columns, under the hood, the frame is stored as 3 arrays (vectors) of length 10 each (along with some meta-data). A given "row" in the DataFrame is achieved by indexing into each column array to the same position. This is achievable since the entire DataFrame is held in memory while processing.

That is not the case in a Spark DataFrame. A Spark DataFrame needs to be able to process its operations in parallel, often on different servers that cannot communicate their state to one another during the processing stages. Thus, the data is stored in a relational format rather than a columnar one; where the natural division of objects is along a given row rather than a cohesive column. Because it is common to have far more rows than columns as data gets very large, this organizational structure means that operations are parallelized from the row level instead of the column level. To the user, almost all of this is handled in the engine layer where interactions are not substantially different.

TAKEAWAYS

Apache Spark is highly effective not because it reinvents the wheel but because it amplifies existing analytical methods to be as highly effective across massive data sets, as they are to comparably small ones you can process on a single machine. Spark DataFrames are revolutionary precisely because they are at heart not a whole new thing but rather a bridge from the tried and true analytical methods of the past to the scalable nature necessary for the future. Everything old is new again!

FRANK D. EVANS is a Data Scientist with Exaptive. He primarily works with machine learning and feature engineering, specializing in unstructured and semi-structured data. His interests span natural language processing, network graph analysis, and building semi-supervised applications. Frank has a B.S. from St. Gregory's University and a Master's Specialization in Data Science from Johns Hopkins University.

Understanding Machine Learning

BY CHARLES-ANTOINE RICHARD
MARKETING DIRECTOR, ARCBEES

QUICK VIEW

01 There are two functions in particular that are trailblazing businesses' adoption of machine learning: logistics and production, and sales and marketing.

02 Three conditions for success emerge in order to obtain an observable return on investment when using machine learning: identifying the right business problem, having sufficient data, and having qualified talent.

03 Collaboration is key. Artificial intelligence is disruptive and its adoption can prove arduous at times, but members of the AI field are deeply passionate about it and are gladly willing to share their knowledge and know-how.

WHAT EXACTLY IS MACHINE LEARNING?

Here's the simplest definition I came across, from Berkeley: Machine learning is "[...] the branch of AI that explores ways to get computers to improve their performance based on experience".

Let's break that down to set some foundations on which to build our machine learning knowledge.

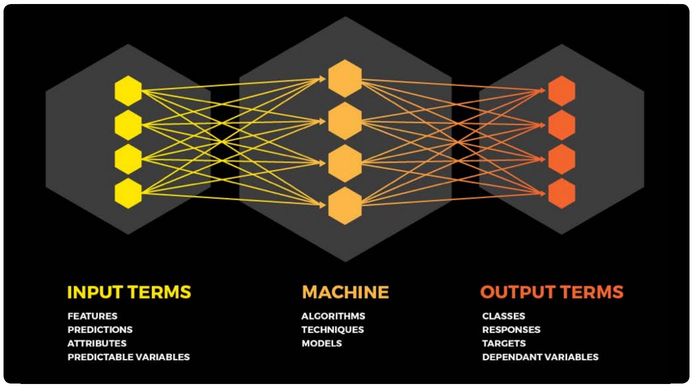
Branch of AI: Artificial intelligence is the study and development by which a computer and its systems are given the ability to successfully accomplish tasks that would typically require a human's intelligent behavior. Machine learning is a part of that process. It's the technology and process by which we train the computer to accomplish a given task.

Explores models: Machine learning techniques are still emerging. Some models for training a computer are already recognized and used (as we will see below), but it is expected that more will be developed with time. The idea to remember here is that different models can be used when training a computer. Different business problems require different models.

Get computers to improve their performance: For a computer to accomplish a task with AI, it needs practice and adaptation. A machine learning model needs to be trained using data and in most cases, a little human help.

Based on experience: Providing an AI with experience is another way of saying "to provide it with data." As more data is fed into the system, the more accurately the computer can respond to it and to future data that it will encounter. More accuracy in understanding the data means a better chance to successfully accomplish its given task or to increase its degree of confidence when providing predictive insight.

Quick Example:



1. Entry data is chosen and prepared along with input conditions (e.g. credit card transactions).
2. The machine learning algorithm is built and trained to accomplish a specific task (e.g. detect fraudulent transactions).
3. The training data is augmented with the desired output information (e.g. these transactions appear fraudulent, these do not).

HOW DOES MACHINE LEARNING WORK?

Machine learning is often referred to as magical or a black box:

Insert data ➔ magic black box ➔ mission accomplished.

Let's look at the training process itself to better understand how machine learning can create value with data.

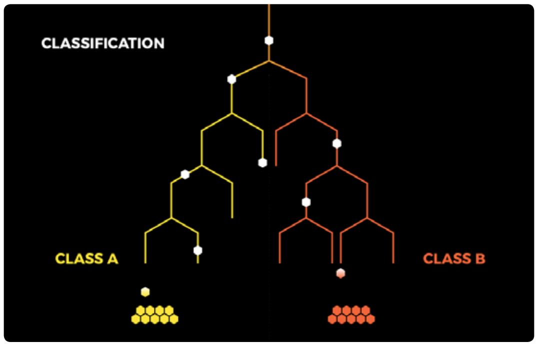
- **Collect**: Machine learning is dependent on data. The first step is to make sure you have the right data as dictated by the problem you are trying to solve. Consider your ability to collect it, its source, the required format, and so on.
- **Clean**: Data can be generated by different sources, contained in different file formats, and expressed in different languages. It might be required to add or remove information from your data set, as some instances might be missing information while others might contain undesired or irrelevant entries. Its preparation will impact its usability and the reliability of the outcome.
- **Split**: Depending on the size of your data set, only a portion of it may be required. This is usually referred to as sampling. From the chosen sample, your data should be split into two groups: one to train the algorithm and the other to evaluate it.
- **Train**: As commonly seen with neural networks, this stage aims to find the mathematical function that will accurately accomplish the chosen goal. Using a portion of your data set, the algorithm will attempt to process the data, measure its own performance and auto-adjust its parameters (also called backpropagation) until it can consistently produce the desired outcome with sufficient reliability.
- **Evaluate**: Once the algorithm performs well with the training data, its performance is measured again with data that it has not yet seen. Additional adjustments are made when needed. This process allows you to prevent overfitting, which happens when the learning algorithm performs well but only with your training data.
- **Optimize**: The model is optimized before integration within the destined application to ensure it is as lightweight and fast as possible.

ARE THERE DIFFERENT TYPES OF MACHINE LEARNING?

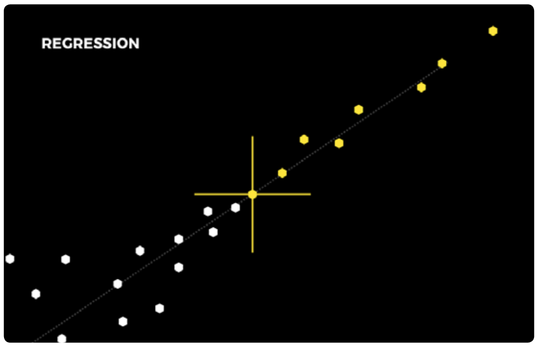
There are many different models that can be used in machine learning, but they are typically grouped into three different types of learning: supervised, unsupervised, and reinforcement. Depending on the task, some models are more appropriate than others.

Supervised learning: With this type of learning, the correct outcome for each data point is explicitly labelled when training the model. This means the learning algorithm is already given the answer when reading the data. Rather than finding the answer, it aims to find the relationship so that when unassigned data points are introduced, it can correctly classify or predict them.

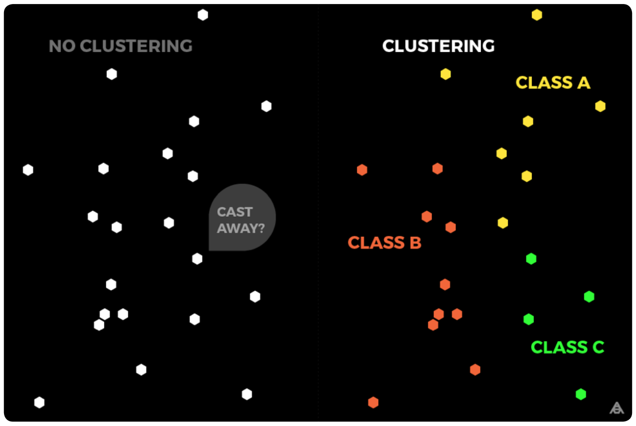
In a classification context, the learning algorithm could be, for example, fed with historic credit card transactions, each labelled as safe or suspicious. It would learn the relationship between these two classifications and could then label new transactions appropriately, according to the classification parameters (e.g. purchase location, time between transactions, etc.).



In a context where data points are continuous in relation to one another, like a stock's price through time, a regression learning algorithm can be used to predict the following data point.



Unsupervised learning (graph shown on following page): In this case, the learning algorithm is not given the answer during training. Its objective is to find meaningful



relationships between the data points. Its value lies in discovering patterns and correlations. For example, clustering is a common use of unsupervised learning in recommender systems (e.g. people who liked this bottle of wine also enjoyed this one).

Reinforcement learning: This type of learning is a blend between supervised and unsupervised learning. It is usually used to solve more complex problems and requires interaction with an environment. Data is provided by the environment and allows the agent to respond and learn. In practice, this ranges from controlling robotic arms to find the most efficient motor combination, to robot navigation where collision avoidance behavior can be learned by negative feedback from bumping into obstacles. Logic games are also well-suited to reinforcement learning, as they are traditionally defined as a sequence of decisions, such as poker, backgammon, and more recently Go with the success of AlphaGo from Google. Other applications of reinforcement learning are common in logistics, scheduling, and tactical planning of tasks.

WHAT CAN MACHINE LEARNING BE USED FOR?

To help you identify what situations can be tackled with machine learning, start with your data. Look for areas in your business that are capable of producing data (in large quantities) and what value can be derived from it.

Machine learning is different from other technological advancements; it is not a plug and play solution, at least not yet. Machine learning can be used to tackle a lot of situations and each situation requires a specific data set, model, and parameters to produce valuable results.

This means you need a clearly defined objective when starting out. Machine learning is making considerable advances in many fields, and all functions within an organization are likely to see disruptive advancements in the future. Nonetheless, some fields are riper than others to pursue its adoption.

I believe there are two functions in particular that are trailblazing businesses' adoption of machine learning:

- Logistics and production;
- Sales and marketing.

The reason why these two areas are leading the way to a more widespread integration of machine learning within daily practices is simple: they promise a direct influence on ROI.

Most gains from its use can be categorized into two major fields: predictive insight and process automation, both of which can be used in ways that can either lower costs or increase revenue.

Predictive insight:

- Predictive insight into customers' behavior will provide you with more opportunities for sales;
- Anticipating medicine effectiveness can reduce time to market;
- Forecasting when a user is about to churn can improve retention.

In this context, machine learning has the potential to increase your reactivity by providing you with the tools and information to make decisions faster and more accurately.

Process automation and efficiency:

- Augmenting investment management decisions with machine learning powered software can provide better margins and help mitigate costly mistakes;
- Robotic arm movement training can increase your production line's precision and alleviate your need for quality control;
- Resource distribution according to user demand can save time and costs during delivery.

When machine learning is used in this context, your business becomes smarter. Your processes and systems augment your value proposition, and your resources are used more efficiently.

CHARLES-ANTOINE RICHARD is the Marketing Director at Arcbees. He has extensive experience putting into market innovative digital products and services. He has a passion for merging the bonds between business strategy and emerging technological solutions like artificial intelligence. He focuses his efforts on incorporating commercial insight within the development process, through all stages of a product's lifecycle. Charles is a Bachelor of Business Administration from University Laval in Quebec, having specialized in marketing and entrepreneurship.

Critical Capabilities in Next Generation Self-service Data Preparation Tools

BY FRANK MORENO
VP PRODUCT MARKETING, DATAWATCH

Data preparation is *the process of collecting, cleaning, and consolidating data into one file or data table, primarily for use in analysis. It is most often used when handling messy, inconsistent, or un-standardized data; trying to combine data from multiple sources; reporting on data that was entered manually; or dealing with data that was scraped from an unstructured source such as PDF documents.*

In a world that has become increasingly dependent on spreading and sharing data rapidly, the need for smart businesses to adapt is clear. Commodities, including enterprise data, need to be managed and governed in a way that allows access to the right people at the right time, enabling business to happen at ever-increasing speeds. If important information is not available in a timely and convenient manner, then the opportunity for taking full advantage of that data has already passed.

Effort must be made to empower businesses to access their data faster and more easily, and share that data across the enterprise in a secure and dependable way. The quick sorting, cleaning, and blending of that critical data is becoming the base expectation for successful insights, thus proving the need for smart, evolving self-service data preparation. **Here are the critical components of a next-gen data prep tool:**

□ **Visual data preparation built for the cloud**
Web-based self-service data preparation providing access to everyone, everywhere. Available onsite or in the cloud.

□ **Certified, centralized data**
Any employee can easily find and use ANY data that has been made accessible to them within a data ecosystem.

□ **Collaboration**
Understand the relevancy of data in relation to how it's utilized by different user roles in the organization (e.g., sales operations or internal auditing), follow key users and data sets, and collaborate to better harness the "tribal knowledge" that too often goes unshared.

□ **Gamification**
Leverage motivational concepts and techniques to encourage decision makers to engage and collaborate with one another – both to drive participation and to better their ability to make more informed decisions

□ **Crowdsourcing**
Leverage user ratings, recommendations, discussions, comments, and popularity to make better decisions about which data to use. Share and comment on workspaces and data sources.

□ **Import and export connectivity**
Packaged connectors for import and exports.

□ **Information marketplace**
Browse a centralized catalog of all relevant internal and external data. Organize and manage content resources (workspaces and data sources).

□ **Intuitive search**
Search cataloged data, metadata, and data preparation models indexed by user, type, application, and unique data values to quickly find the right information.

□ **Machine learning**
Benefit from machine learning capabilities, which identify patterns of use and success, perform data quality scoring, suggest relevant sources, and automatically recommend likely data preparation actions based on user persona.

□ **Data quality and governance**
Provide sanctioned, curated data sets to promote reuse and consistency. Comprehensive governance features, including data masking, data retention, data lineage, and role-based permissions, are necessary to uphold corporate and regulatory compliance and enhance trust in data, analytics processes, and results.

□ **Automated operations**
Define, schedule, and execute workflows and exports.

DATAWATCH

BI BLUEPRINT FOR DATA PREPARATION

WHAT IT IS, WHO NEEDS IT AND WHY

WHAT IS DATA PREP?

Data Prep: \dā-tə prep\ *verb*: the process of collecting, cleaning, and consolidating data for use in analysis, involving error correction — human and/or machine input — filling in nulls and incomplete data, and merging data from multiple sources and formats; often considered an arduous task due to difficulty resulting from wrangling of the data.

Data prep includes data cleaning, data masking, data extraction, and data blending.

WHO NEEDS DATA PREP?

Not all data roles are the same. Not only do the responsibilities of each vary, but so do the groups they serve.

Self-service data prep simplifies the data prep process itself by transforming how analysts and everyday business users acquire, manipulate and blend data.

DATA SCIENTIST

Leveraging data to tell the whole story

DATA ANALYST

Using multiple data sources for complete analysis

DATA ARCHITECT

Defining how data is stored, consumed and managed.

DATA ENGINEER

Designing and building an organization's big data infrastructure

DATABASE ADMINISTRATOR

Managing database software usage

DATA AND ANALYTICS MANAGER

Designing, implementing and supporting data analysis solution

WHY DATA PREP IS NEEDED

No matter which data role individuals in your organization fall into, each can benefit from self-service data prep. Just take a look at the issues those in your company face without data prep and how they can benefit from it. Think cost-effective, scalable and comprehensive.

WITHOUT DATA PREP:

- Reliance on IT for data access
- Risky data can't be masked
- Hand-keying data from PDFs, HTML or text documents
- Data is outdated by the time it's been prepped for analysis

WITH DATA PREP:

- Quickly leverage data from any source
- More time spent on analysis than on preparation
- Reduce compliance risk with data governance
- Automate and streamline manual data prep tasks

Ready to try data prep for yourself? Download a free trial of Datawatch Monarch self-service data prep software at

datawatch.com/try-now

Data Socialization: How to Achieve "Data Utopia" and Expedite Outcomes

In the two years since Gartner proclaimed "self-service data preparation" to be essential, the competitive landscape has changed dramatically. Now, dozens of vendors offer stand-alone data preparation tools or have integrated data preparation features into existing solutions. Data types are evolving just as quickly, with big data, streaming data, and machine data adding to the ongoing challenge of analysis.

At the same time, social technology has transformed the way we live and work, increasing expectations about the availability and timeliness of information, yielding similar expectations for businesses.

The data-prep market has seen rapid innovation. However, for many, data access is restricted to personal data sources, historical reports or information controlled by IT. Plus, teams are limited to sharing what little data they have via Excel spreadsheets, increasing compliance concerns and diminishing trust in their analysis.

Enter data socialization, a concept that is reshaping the way organizations think about and interact with their data. It involves a data management platform that unites self-service data preparation, data discovery, and cataloging, automation, and governance features with key attributes common to social media, such as leveraging ratings and comments to make better decisions about which data to use. It enables users to search for, share, and reuse data to achieve true enterprise collaboration.

> Enter data socialization, a concept that is reshaping the way organizations think about, and interact with, their data.

In this "data utopia," anyone can find and use ANY data that has been made accessible to them, creating a social network of curated and raw data sets. Users can learn from each other and be more productive as they source, cleanse, and prepare data. Ultimately, it enables organizations to expedite analytics outcomes to drive better and faster business decisions.

WRITTEN BY MICHAEL MORRISON
CEO, **DATAWATCH CORP**

PARTNER SPOTLIGHT

Datawatch Monarch Self-Service Data Preparation

DATAWATCH

> "It became clear that we needed to automate the reconciliation process."

NEW RELEASES
Bi-Annually

OPEN SOURCE
No

CASE STUDY
IBERIABANK struggled with reconciling general ledger accounts — a manual and very time-consuming process. Denny Pagnelli, Controller at IBERIABANK explains, "There are millions and millions of rows in the general ledger...It became clear that we needed to automate the reconciliation process." The Bank began using Monarch for its reconciliation process. Now, they can extract the data from various databases and PDFs, fully automating the process and cutting reconciliation time from 2 hours to 2.5 minutes.

STRENGTHS

- Automatically extract from reports, PDF documents, and web pages
- Combine, clean, and use with your favorite tools
- Easily combine disparate data automatically using powerful join analysis
- Automate data preparation processes for maximum time saving

NOTABLE CUSTOMERS

- Dana-Farber Cancer Institute
- Mastercard
- Unitil
- Au Bon Pain
- Equifax
- Public Service Credit Union

WEBSITE datawatch.com

TWITTER @Datawatch

BLOG datawatch.com/blog

MACHINE LEARNING

at the Big Data Buffet

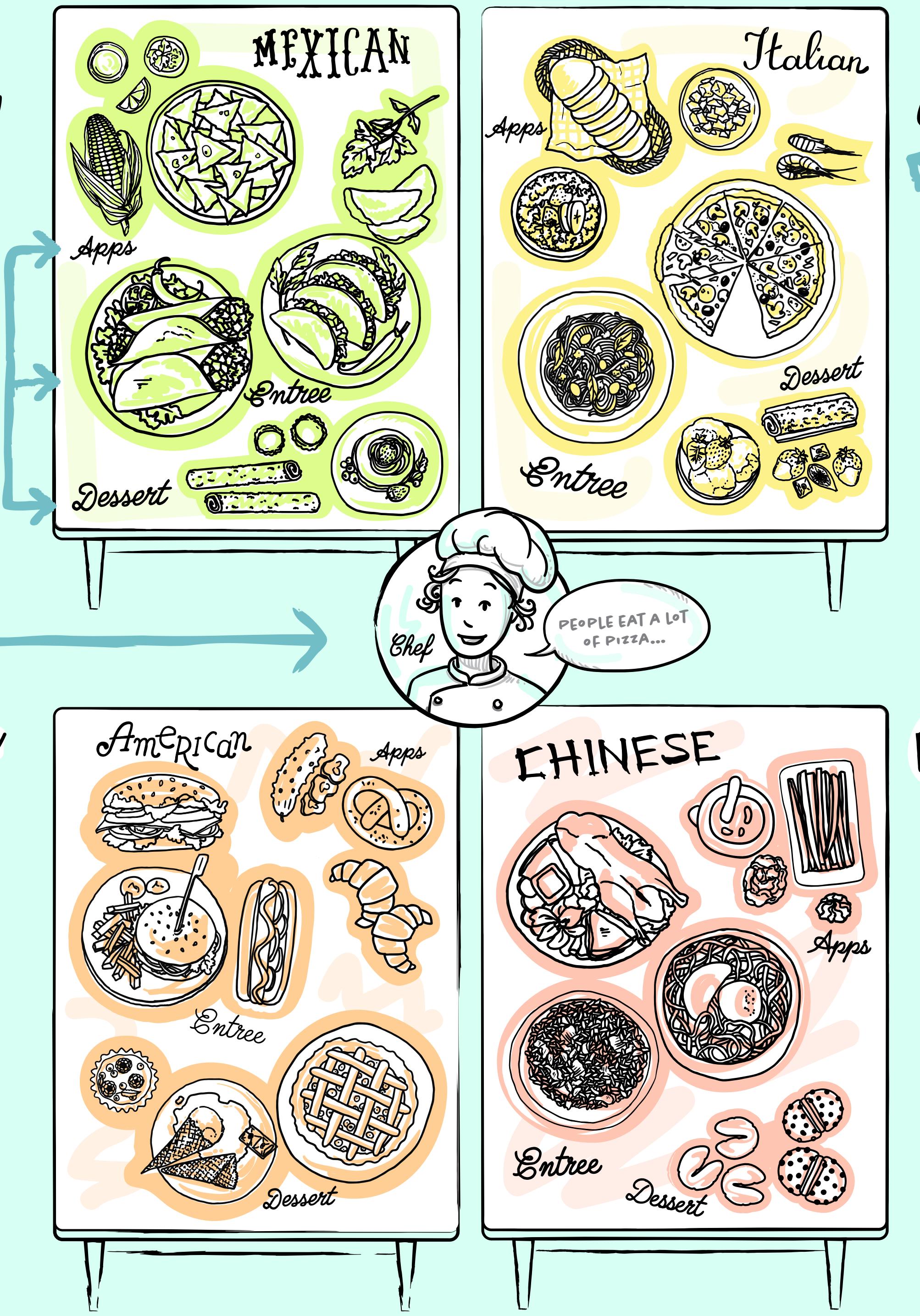
As everyone collects more data to analyze, more and more work is created for data scientists and analysts. Enter the field of machine learning and AI, a field designed for computers to "learn" based on samples of data and make predictions on how data will change and move. Here, we've explored some of the key machine learning algorithms that help organizations gorge themselves on historical data and make decisions quickly. **Feast your eyes on our examples below!**

CLUSTERING

The task of grouping a set of data points into a group where all objects are more similar to each other than those in other groups. For example, in the Mexican section, chips, guacamole, and queso are all clustered together as appetizers, since they're all small portions that are meant to start a meal, as opposed to entrees or desserts.

Classification

The act of assigning a category to a new data point based on a set of training data. For example, if we introduce a new dish to the buffet, shrimp scampi, and we know that the dish uses a lot of garlic and lemon juice, it can be accurately classified as Italian station.

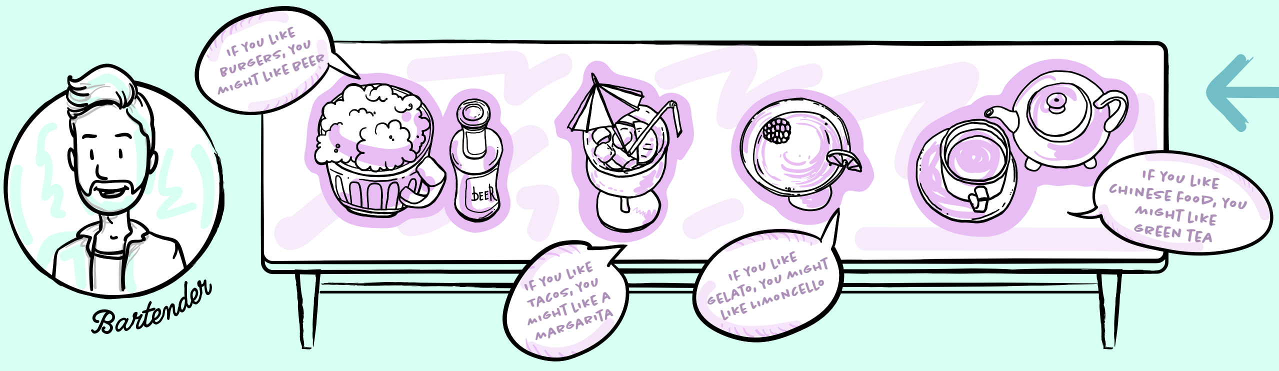


Regression

The prediction of what value an item would have in a continuous series of data. For example, the head chef knows that during the weekends, pizza flies off the stations, so based on what they've sold in past weekends, she orders 127 pounds of dough, 53 jars of sauce, and 44 bags of cheese.

RECOMMENDATION

A task in which a suggested choice or piece of data is chosen by a machine based on what that choice has in common with a set of historical data. For example, if a waiter sees that you've picked up a hamburger, they would suggest you pair it with a beer.


IF YOU LIKE BURGERS, YOU MIGHT LIKE BEER
BEER
IF YOU LIKE TACOS, YOU MIGHT LIKE A MARGARITA
IF YOU LIKE GELATO, YOU MIGHT LIKE LIMONCELLO
IF YOU LIKE CHINESE FOOD, YOU MIGHT LIKE GREEN TEA
Bartender

A Big Data Reference Architecture for IoT

BY **TIM SPANN** SOLUTIONS ENGINEER, **HORTONWORKS**
AND **VAMSI CHEMITIGANTI** GENERAL MANAGER (FINANCIAL SERVICES), **HORTONWORKS**

QUICK VIEW

01 IoT is rapidly transitioning from a consumer technology to its enterprise disruptor cousin — IIOT. Government and diverse industries have begun creating industrial applications that provide a better customer experience, reduce waste, and cost while driving better financial outcomes.

02 IIoT Platforms that solve specific business challenges are being built on a combination of Big Data, advanced analytics, and cloud computing.

03 We discuss such an industrial strength architecture that is built from the above technology elements and enables flexible deployment for new capabilities while reducing total cost of ownership.

The Industrial Internet of Things (IIOT) is emerging as a key business and technology trend in society. IIOT enables various entities such as municipalities, industrial manufacturing, utilities, telecom, and insurance to address key customer and operational challenges. The evolution of technological innovation in areas such as Big Data, predictive analytics, and cloud computing now enables the integration and analysis of massive amounts of device data at scale while performing a range of analytics and business process workflows on this data.

This article aims to discuss a vendor- and product-agnostic reference architecture that covers an end-to-end IIOT implementation, covering various layers of such an architecture. The end goal is to enable the creation of enterprise business applications that are data-driven.

DATAFRAMES IN GENERAL

The first requirement of IIoT implementations is to support connectivity from the Things themselves or the Device layer. The Device layer includes a whole range of sensors, actuators, smartphones, gateways, industrial equipment, etc. The ability to connect with devices and edge devices like routers and smart gateways using a variety of protocols is key. These network protocols include Ethernet, Wi-Fi, and cellular, which can all directly connect to the Internet. Other protocols that need a gateway device to connect include Bluetooth, RFID, NFC, Zigbee, et al. Devices can connect directly with the data ingest layer, shown above, but it is preferred that they connect via a gateway, which can perform a range of edge processing. This is important from a business standpoint. For example, in certain verticals like healthcare and financial services, there exist stringent regulations that govern when certain identifying data elements (e.g. video feeds) can leave the premises of a hospital or bank. A gateway cannot just perform intelligent edge processing, but can also connect thousands of device endpoints and facilitate bidirectional communication with the core IIoT architecture. For remote locations, more powerful devices like the Arrow BeagleBone Black Industrial and MyPi Industrial, you can run a tiny Java or C++ MiniFi agent for your secure connectivity needs. These agents will send the data to an Apache NiFi gateway or directly into your enterprise HDF cluster in the cloud or on-premise.

The data sent by the device endpoints are modeled into an appropriate domain representation based on the actual content of the messages. The data sent over also includes metadata around the message. A canonical model can optionally be developed (based on the actual business domain) which can support a variety of applications from a business intelligence standpoint.

The ideal tool for these constantly evolving devices, metadata, protocols, data formats, and types is Apache NiFi. Apache NiFi supports the flexibility of ingesting changing file formats, sizes, data types, and schemas. Whether your devices send XML today and send JSON tomorrow, Apache NiFi supports ingesting any file type you may have.   Once inside Apache NiFi, it is enveloped in security, with every touch to each flow file controlled, secured, and audited. You will have full data provenance for each file, packet, or chunk of data sent through the system. Apache NiFi can work with specific schemas if you have special requirements for file types, but it can also work with unstructured or semi-structured data just as well. NiFi can ingest 50,000 streams concurrently on a zero-master, shared-nothing cluster that horizontally scales via easy administration with Apache Ambari.

DATA AND MIDDLEWARE LAYER

The IIoT Architecture recommends a Big Data platform with native message-oriented middleware (MOM) capabilities to ingest device mesh data. This layer will also process device data in such a fashion — batch or real-time — as the business needs demand.

Application protocols such as AMQP, MQTT, CoAP, WebSockets, etc. are all deployed by many device gateways to communicate application-specific messages. The reason for recommending a Big Data/NoSQL dominated data architecture for IIoT is quite simple. These systems provide Schema on Read, which is an innovative data-handling technique. In this model, a format or schema is applied to data as it is accessed from a storage location, as opposed to doing the same while it is ingested. From an IIoT standpoint, one must not just deal with the data itself but also metadata such as timestamps, device ID, other firmware data, such as software version, device manufactured data, etc. The data sent from the device layer will consist of time series data and individual measurements.

The IIoT data stream can be visualized as a constantly running data pump, which is handled by a Big Data pipeline that takes the raw telemetry data from the gateways, decides which ones are of interest, and discards the ones not deemed significant from a business standpoint. Apache NiFi is your gateway and gate keeper. It ingests the raw data, manages the flow of thousands of producers and consumers, does basic data enrichment, sentiment analysis in stream, aggregation, splitting, schema translation, format conversion, and other initial steps to prepare the data. It does that all with a user-friendly web UI and easily extendible architecture. It will then send raw or processed data to Kafka for further processing by Apache Storm, Apache Spark, or other consumers. Apache Storm is a distributed real-time computation engine that reliably processes unbounded streams of data. Storm excels at handling complex streams of data that require windowing and other complex event processing. While Storm processes stream data at scale, Apache Kafka distributes messages at scale. Kafka is a distributed pub-sub real-time messaging system that provides strong durability and fault tolerance guarantees. NiFi, Storm, and Kafka naturally complement each other, and their powerful cooperation enables real-time streaming analytics for fast-moving big data. All the stream processing is handled by NiFi-Storm-Kafka combination. Consider it the Avengers of streaming.

Appropriate logic is built into the higher layers to support device identification, ID lookup, secure authentication, and transformation of the data. This layer will process data (cleanse, transform, apply a canonical representation) to support business automation (BPM), BI (business intelligence), and visualization for a variety of consumers. The data ingest layer will also provide notification and alerts via Apache NiFi.

Here are some typical uses for this event processing pipeline:

a. Real-time data filtering and pattern matching
b. Enrichment based on business context
c. Real-time analytics such as KPIs, complex event processing, etc.
d. Predictive analytics
e. Business workflow with decision nodes and human task nodes

APPLICATION TIER

Once the device data has been ingested into a modern data lake, key functions that need to be performed include data aggregation, transformation, enriching, filtering, sorting, etc. As one can see, this can get very complex very quick — both from a data storage and processing standpoint. A cloud-based infrastructure, with its ability to provide highly scalable compute, network, and storage resources, is a natural fit to handle bursty IIoT applications. However, IIoT applications add their own diverse requirements of computing infrastructure, namely the ability to accommodate hundreds of kinds of devices and network gateways, which means that IT must be prepared to support a large diversity of operating systems and storage types.

The business integration and presentation layer is responsible for the integration of the IIoT environment into the business processes of an enterprise. The IIoT solution ties into existing line-of-business applications and standard software solutions through adapters or enterprise application integration (EAI) and business-to-business (B2B) gateway capabilities. End users in business-to-business or business-to-consumer scenarios will interact with the IIoT solution and the special-purpose IIoT devices through this layer. They may use the IIoT solution or line-of-business system UIs, including apps on personal mobile devices, such as smartphones and tablets.

Once IIoT knowledge has become part of the Hadoop-based data lake, all the rich analytics, machine learning, and deep learning frameworks, tools, and libraries now become available to data scientists and analysts. They can easily produce insights, dashboards, reports, and real-time analytics with IIoT data joined with existing data in the lake, including social media data, EDW data, and log data. All your data can be queried with familiar SQL through a variety of interfaces such as Apache Phoenix on HBase, Apache Hive LLAP and Apache Spark SQL. Using your existing BI tools or the open sourced Apache Zeppelin, you can produce and share live reports. You can run TensorFlow in containers on YARN for deep learning insights on your images, videos, and text data while running YARN-clustered Spark ML pipelines fed by Kafka and NiFi to run streaming machine learning algorithms on trained models.

VAMSI CHEMITIGANTI is responsible for driving Hortonwork's business and technology vision from a global banking standpoint. The clients Vamsi engages with on a daily basis span marquee financial services names across major banking centers on Wall Street, in Toronto, London, and Asia, including businesses in capital markets, core banking, wealth management, and IT operations. He is also a regular speaker at industry events. Vamsi blogs on financial services business and the industry landscape at www.vamsitalkstech.com

TIM SPANN is a Big Data Solution Engineer. He helps educate and disseminate performant open source solutions for Big Data initiatives to customers and the community. With over 15 years of experience in various technical leadership, architecture, sales engineering, and development roles, he is well-experienced in all facets of Big Data, cloud, IoT, and microservices. As part of his community efforts, he also runs the Future of Data Meetup in Princeton.

HPCC Systems:

A powerful, open source Big Data analytics platform.

Two integrated clusters, a declarative programming language, and a standards-based web services platform form the basis of this comprehensive, massively scalable Big Data solution.

Open source.
Easy to use.
Proven.

HPCC Systems Platform Features:

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.

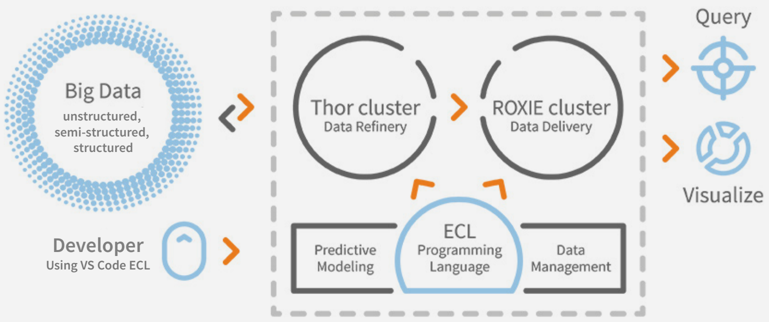
Query and Search
An indexed based search engine to perform real-time queries. SOAP, XML, REST, and SQL are all supported interfaces.

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

High Performance Computing Cluster (HPCC)

"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



Big Data
unstructured, semi-structured, structured

Thor cluster
Data Refinery

ROXIE cluster
Data Delivery

Query

Visualize

Developer
Using VS Code ECL

Predictive Modeling

ECL Programming Language

Data Management

Visit: hpccsystems.com

LexisNexis®
RISK SOLUTIONS

HPCC SYSTEMS®

The HPCC Platform's ECL Roots Make it the Ideal Solution for Data Scientists

Most Big Data solutions built with tools like Hadoop and Spark are built using programming languages such as Java, Scala, and Python. As these tools are well known in the Big Data software development community, this reduces many of the traditional barriers to platform development (not enough developers, proprietary software, lack of support resources, etc.) and helps create a strong software ecosystem. However, has any thought been given to who will ultimately use these tools? The data scientists who help their organizations get the most value from their data may be experts in analysis, but they're not programmers. So when they need to view data in a different way, IT needs to develop a script that pulls the required data and presents it in a dashboard, wasting valuable time and manpower resources. The HPCC Systems Big Data platform by LexisNexis Risk Solutions uses ECL, an easy to use, data-centric programming language that's optimized for data processing and queries. In other words, the platform uses the native language of data scientists, allowing them to easily build the search queries and visualizations they need to get the best information from their data, without support from IT. The HPCC platform also includes a large library of built-in modules of common data manipulation tasks so data scientists can begin analyzing their data immediately. The platform offers exceptional performance as well, generating code five times more efficiently than SQL-based platforms.

A Big Data Analysis Platform that is powerful, extensible, maintainable, and completely homogeneous for easy setup and customization

WRITTEN BY ARJUNA CHALA
SR DIRECTOR, TECHNOLOGY DEVELOPMENT, **LEXISNEXIS**

PARTNER SPOTLIGHT

HPCC Systems® By LexisNexis® Risk Solutions

HPCC SYSTEMS®
hpccsystems.com

End-to-end Big Data in a massively scalable supercomputing platform

CATEGORY
Big Data

NEW RELEASES
Large releases annually, updates throughout the year

OPEN SOURCE
Yes

CASE STUDY
ProAgrica drives growth and improves efficiency to global food production by delivering high-value insight and data, critical tools, and advanced technology solutions. ProAgrica estimates that the world will need to double its current food production output to feed a population estimated to reach 9.6 billion people by 2050. There is a vast amount of data available across the agricultural landscape that could provide valuable insights into more efficient global food production. ProAgrica is leveraging the High-Performance Computing Cluster (HPCC), an open-source Big Data platform developed by LexisNexis Risk Solutions, to consolidate, organize, and enhance agricultural data to help deliver value across the entire industry.

STRENGTHS
- Fast performance
- Easy to deploy and use
- Scale based on the amount of data
- Rich API for data preparation, integration, quality checking, duplicate checking, etc.
- Parallelized machine learning algorithms for distributed data
- Real-time query and search with support for SQL, JSON, SOAP, and XML
- Free support via detailed documentation, video tutorials, forums, and direct contact

NOTABLE CUSTOMERS
- InfoSys
- CPL Online
- Flight Global
- Comrise
- Cognizant
- ClearFunnel
- RNET Technologies
- ProAgrica

WEBSITE hpccsystems.com

TWITTER @hpccsystems

BLOG hpccsystems.com/blog

Executive Insights on the State of Big Data

BY **TOM SMITH**
RESEARCH ANALYST, **DZONE**

QUICK VIEW

01 The key to a successful Big Data strategy is knowing what problem you're trying to solve before you begin investing in software and tools.

02 Companies can get more out of Big Data by knowing what they're going to do with the data they're collecting, as well as the insights they'd like to discern.

03 The future of Big Data is real-time decision making with machine learning and natural language processing.

To gather insights on the state of Big Data today, we spoke with 22 executives from 20 companies who are working with Big Data themselves or providing Big Data solutions to clients. Here's who we talked to:

NITIN TYAGI, Vice President Enterprise Solutions, Cambridge Technology Enterprises

RYAN LIPPERT, Senior Marketing Manager and **SEAN ANDERSON**, Senior Product Marketing Manager, Cloudera

SANJAY JAGAD, Senior Manager, Product Marketing, Coho Data

AMY WILLIAMS, COO, Data Conversion Laboratory (DCL)

ANDREW BRUST, Senior Director Market Strategy and Intelligence, Datameer

ERIC HALLER, Executive Vice President, Experian DataLabs

JULIE LOCKNER, Global Product Marketing, Data Platforms, Intersystems

ERIC MIZELL, Vice President Global Engineering, Kinetica

JIM FREY, V.P. Strategic Alliances, Kentik

ROB CONSOLI, Chief Revenue Officer, Liaison

DALE KIM, Senior Director of Industrial Solutions, MapR

CHRIS CHENEY, CTO, MPP Global

AMIT SATOOR, Senior Director, Product and Solution Marketing, SAP

GUY LEVY-YURISTA, Head of Product, Sisense

JON BOCK, Vice President of Product and Marketing, Snowflake Computing

BOB BRODIE, CTO, SUMOHeavy

KIM HANMARK, Director of Professional Services EMEA, TARGIT

DENNIS DUCKWORTH, Director of Product Marketing, VoltDB

ALEX GORELIK, Founder and CEO and **TODD GOLDMAN**, CMO, Waterline Data

OLIVER ROBINSON, Director and Co-Founder, World Programming

KEY FINDINGS

01 The key to a successful Big Data strategy is **knowing what problem you are trying to solve** before you begin investing in software and tools. Without knowing the problem you are trying to solve and the metrics that will define success, you don't know how to specify the software and tools that will help you achieve your goals. The second key, closely related to the first, is knowing what insights you are looking for and the value you are attempting to bring your business. The more specific you are about the business need and the problem, the more likely you are to solve it. Pursuing a "Big Data" strategy because you are collecting a lot of data will end up wasting a lot of money and time. A Big Data initiative is not an inexpensive proposition, so identify the specific use case, execute the solution, show the value provided, and then move to the next use case.

02 The 80 percent of companies that aren't getting more out of Big Data can start with strategic planning. **Know what you're going to do with the information you've collected**, the insights you want to uncover, the source of the data, and understand the need for it to be cleaned and prepped before it can be integrated with other data. Empower others in the organization to access the data. Ultimately you want to be able to provide real-time decision-making at every level of the business; however, you need to implement several successful use cases before you can achieve this goal. Crawl, walk, then run.

03 The biggest change in Big Data over the past year has been the **uptick in real-time streaming of data** and ingestion engines that can handle the volume and scale of that data. Streams are part of a Big Data strategy and help to break down siloes. With machine learning and natural language processing, Big Data is available for humans everywhere. At least one company is already enabling their clients to use Alexa and natural language processing to run queries on their data and obtain insights from oceans of data.

04 **Hadoop, Spark, and Tableau** were the most frequently mentioned solutions for collecting and analyzing data with several other tools sitting atop Hadoop. Open source solutions like Kafka, Nitti, and Storm were mentioned by a couple of respondents, with Python and R being mentioned as useful languages for data analysis. SAS used to have a monopoly on analytics tools, but that has changed in the last 12 months with more people using R and H2O. However, this is not the end of SAS, since it's well-engrained in 99 percent of the Fortune 500.

05 **Retail, healthcare, media, and telecommunications** are the four most frequently mentioned industries where Big Data is solving real-world problems. However, examples were also provided in financial services, government, IT, and fleet management. In healthcare and financial services, Big Data is being used to identify patient/customer care, fraud, and abuse. Natural language processing is enabling the monitoring and reporting of sentiment on social media channels to help telcos, retailers, CPG manufacturers, and pharmaceutical companies understand consumer sentiment, predict trends, and churn. Retailers are focused on personalization across multiple devices and brick-and-mortar stores to provide a better customer experience.

06 **Lack of skilled data professionals** is the most frequently mentioned issue preventing companies from realizing the benefits of Big Data. Having the right people to build out a Big Data team is key, but there's currently a huge talent gap. Data scientists must keep their skills sharp and know what tools are evolving to tackle the problems their companies are attempting to solve. The Big Data ecosystem is moving very quickly, and it takes time to learn about what tools are available, what their best use cases are, and to determine if they'll still be relevant in a year. People underestimate the difficulty of implementing a fully-functioning Big Data system. In addition to data scientists, you need product owners, a data engineering team, and other professionals familiar with data preparation, integration, and operationalization.

07 **The future of Big Data is real-time decision-making with machine learning and natural language processing.** This will provide insights everywhere for everyone – not just the data elite. We will be collecting more data and getting actionable insights with automated processes to get near-term value from data. Big Data analytics will be integrated into day-to-day operations.

08 **The proliferation of data and tools is on par with privacy and security as the biggest concerns around the state of Big Data today.** There is confusion around the technologies and there's too much data to ingest. We have a separate tool for every problem, the tools are complex, some only have minor differences, and they are changing daily. A year ago, MapReduce was the "big thing." Today it's Spark. How do I know where to invest my money and my time? Security and privacy continue to be secondary concerns, with more emphasis on widgets than where the data is coming from and how to keep it safe. Google, Apple, and telcos are collecting data on everyone, and we don't know what they're doing with it. Companies are collecting more data than they can protect. The black hats are ahead of the white hats.

09 The skills developers need to work on for Big Data projects fall into two areas: languages and business skills. The most frequently recommended languages were **Java and Python, and knowing Apache Spark was also highly encouraged**. The most frequently mentioned business skills were 1) understanding the business and business problem; 2) collaboration; and, 3) understanding machine learning and natural language processing.

10 Additional considerations from the respondents were varied:

- Does Big Data technology include relational databases? What are the types of data and speeds that it includes? Can it scale in different formats and different engines? Can it integrate with disparate data?
- We talk about Big Data but we don't talk about the need to clean the data and put it in a searchable format.
- We need to help people find the faster path to build solutions and how to project the time to deliver projects.
- Where is Big Data going as an industry to produce tangible value?
- Specific industries, such as healthcare and financial services, are seeing the need for a very specific set of tools. What technologies and trends are emerging for particular industries with particular needs?
- Voice search is a massive opportunity for data and it's going to get hotter.
- How do others see cloud playing into Big Data? Playgrounds in the cloud are great for developers, but how to we bring what they've done back on premise?
- Focus on machine learning and natural language processing thereafter (Alexa and Echo).
- How can companies who aren't big enough to invest in Big Data solutions find a place to host their data that lets them analyze it and then move it to a larger platform when they're ready? I know about Mixpanel, but are their others?

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

Streamline the Path From Big Data to Value Creation

Enterprises creating the next generation of applications need to consider the most efficient path to business value. In our services-based economy, customer experiences matter most. Business value comes from developing applications that create the ultimate experience.

The ultimate experience, depending on your industry, requires knowing as much as you can about your customers, your patients, and your citizens. Having that knowledge when they are about to make a decision that matters is critical in cementing positive impressions that will last. Weaving together multiple data sources with unique storage systems and processing platforms required can slow application development and create scalability challenges. It also complicates governance, risk management, and compliance.

If developers are not building applications that leverage all possible sources of information, Big Data, and real-time analytics to create experiences that can be quickly adapted to new engagement models, they will fall short of their targets. Enterprises need to consider using a comprehensive, consolidated data platform to build transformational applications. The alternative — deploying multiple disparate technologies and systems — adds complexity and application development costs.

InterSystems has been delivering reliable, scalable, and innovative technology to thousands of customers for more than 30 years. The InterSystems Data Platform reduces complexity without sacrificing functionality, performance, or flexibility. By delivering world-class customer support, we focus on our customers' needs so they can focus on building applications that matter – elevating the bar for the ultimate in customer experience.

WRITTEN BY JULIE LOCKNER
DIRECTOR, PRODUCT MARKETING AND PARTNER PROGRAMS, **INTERSYSTEMS**

PARTNER SPOTLIGHT

InterSystem Caché by Intersystems

InterSystems
Health | Business | Government

> "With Caché, we obtain considerably superior performance and scalability than is possible with other databases." - **WILLIAM O'MULLANE**, SCIENTIFIC OPERATIONS MANAGER OF THE GAIA MISSION, EUROPEAN SPACE AGENCY

NEW RELEASES

Major release annually, 2 minor releases annually

OPEN SOURCE

No

CASE STUDY

The European Space Agency (ESA) launched an ambitious mission to chart a three-dimensional map of the Milky Way. Because an enormous amount of data will need to be quickly stored and analyzed, ESA has selected Caché as the advanced database technology to support the scientific processing of the Gaia mission.

Gaia will spend five years monitoring a billion stars in our galaxy. In the course of its operation, AGIS must be able to insert up to 50 billion Java objects into a database within seven days. Caché is the only database ESA found that could provide the necessary performance and scalability with only moderate hardware requirements. The information Gaia collects will allow scientists to learn a great deal about the origin, structure, and evolutionary history of our galaxy.

PRODUCT

InterSystem Caché is the industry's only multi-workload, multi-model NoSQL and relational database, with embedded data and application interoperability via Ensemble, built-in structured and unstructured data analytics, and a comprehensive rapid application development environment without sacrificing high performance, scalability, reliability, and security.

STRENGTH

Complete data platform for reliable, complex applications

NOTABLE CUSTOMERS

- Ontario Systems
- MFS
- TD Ameritrade
- ESA
- Netsmart
- WS Trends

WEBSITE www.intersystems.com

TWITTER @InterSystems

BLOG intersystems.com/intersystems-blog/data-matters

Improved R Implementation of Collaborative Filtering for Recommender Systems

BY **STEFAN NIKOLIC**
DATA SCIENTIST, **SMARTCAT**

QUICK VIEW

01 In this article, we focus on memory-based CF algorithms and showcase some of our recent work on improving the classic implementation.

02 We achieved significant algorithm speedup by taking advantage of rating matrix sparsity when calculating similarities and k nearest neighbors.

03 Additionally, we present an approach to deal with large matrices on which classic implementation may run out of memory.

04 The approach is to divide matrices into parts and calculate predictions part-by-part.

Collaborative filtering (CF) is one of the most popular techniques for building recommender systems. It is a method of making automatic predictions about the interests of a user by collecting preference or taste information from many users (collaborating). In this article, we will focus on memory-based algorithms and showcase some of our recent work on improving the classic CF implementation, thus making it applicable to large-scale datasets and reducing the training time by several magnitudes.

Memory-based algorithms can be divided into:

1. User-based CF: If we want to predict how user U will rate item I, we can check how other users who are similar to user U have rated that item. It is probable that the user will rate items similarly to users with similar tastes.

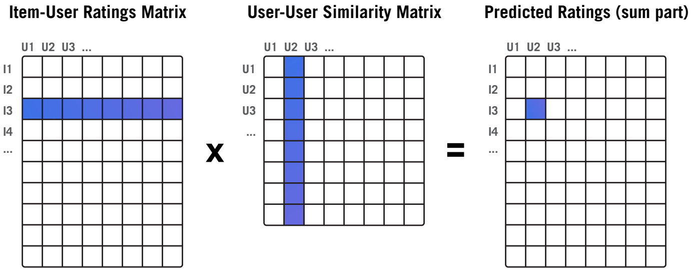
2. Item-based CF: If we want to predict how user U will rate item I, we can check how they have rated other items that are similar to item I. It is probable that the user will rate similar items similarly.

Let's go through an example and see how user-based CF can be implemented. The following formulas show how to calculate rating ru,i, the prediction about how user **u** will rate item i. We aggregate over ratings that users similar to u have given to item i (the set of similar users is marked with S and the similarity function is marked with sim). The more similar a user is, the more influence his rating has on the overall prediction. The value of w is the weighting factor used to scale the sum down to a single rating.

$$r_{u,i} = w \sum_{u' \in S} sim(u,u') r_{u',i}$$

$$w = \frac{1}{\sum_{u' \in S} |sim(u,u')r|}$$

In order to evaluate a recommender system, we need to calculate the predictions for all ratings in a test set. Calculations are usually not done in a loop, but rather using matrix multiplication, since it is a much faster operation. The following picture shows the matrices being used:



Let's focus on user U2 and item I3 for a moment. For predicting how user U2 will rate item I3, we need to know how other users have rated I3 (the blue row in the first matrix) and how similar other users are to U2 (the blue column in the second matrix; note that the similarity of U2 to itself can be ignored by setting it to zero). In this case, the formula for the sum from above can be written as follows (user and item are marked as u2 and i3, and set S covers all users):

$$\sum_{u'\in S} sim(u_2,u')r_{u',i_3}$$

The result is stored in the blue cell of the rightmost matrix. In order to find the final prediction, we also need the coefficient w (as explained above), which is calculated in a similar manner. Finally, by multiplying two matrices, we get instant results for all predictions (not only for U2, I3).

One of the main disadvantages of memory-based CF is related to its scalability and performance when matrices are large. We tried to address these issues using a new implementation of CF in the R programming language (this can be applied to other languages as well). Our implementation was compared to one of the most commonly used packages for recommender systems in R, 'recommenderlab'. The comparison was performed on a single computer with a 4-core i5 and 16GB of RAM using well-known and freely available datasets (MovieLens 1m and MovieLens 10m). It will be shown that our implementation:

1. Is significantly faster.
2. Can support building recommender systems on large datasets, for which the 'recommenderlab' implementation runs out of memory.

EXECUTION TIME IMPROVEMENT

Ratings matrices are usually both large (there are a lot of users and items) and sparse (users typically rate only few items, if any). In R, there is a special representation for sparse matrices, such that missing values (ratings) are not stored into memory. Very often, over 90% of ratings are missing, so this saves a lot of memory. Our implementation, as well as 'recommenderlab', uses this sparse form of matrices.

The main steps used in our implementation of user-based CF are as follows (the same approach is used for item-based CF):

Take a ratings matrix.

1. A user specifies whether to normalize ratings. This step usually increases accuracy. Normalization is used to remove individual ratings bias, which is introduced by users who consistently give lower or higher ratings compared to other users.
2. Calculate similarities between users.
3. Use the k nearest neighbor approach (keep only k most similar users by keeping only k highest values per columns in the user-user similarity matrix). The user needs to specify the value of k.
4. Calculate predictions and denormalize them in case normalization is performed in step one.

The implementation in 'recommenderlab' follows the same procedure. However, we have introduced optimizations that have resulted in significant speed improvements. The two main optimization steps are summarized below:

1. Similarities are calculated using R functions that operate on sparse matrices.
2. k-nearest neighbors on similarity matrices were not calculated in a loop, but rather using an optimized implementation. First, we grouped all the values from the similarity matrix by column. In each group (column), we applied a function that finds the k-th highest value. This was implemented using the R 'data.table' package. Finally, we used this information to keep only the k highest values per column in the similarity matrix.

EVALUATION

We compared our implementation vs. 'recommenderlab' using the following setup:

- 10-fold cross validation. In each iteration, 90% of the ratings were used to create the model and calculate similarities and 10% were used for testing. All users and items were considered for both training and testing.
- Center normalization, where user's average rating is subtracted from his actual ratings.
- Cosine measure to calculate similarities.
- k: the number of nearest neighbors was set to 100 and 300.

	Our implementation (Execution time)		Recommenderlab (Execution time)	
	Item-based CF	User-based CF	Item-based CF	User-based CF
k=100	69.7s	116.2s	3301.5s	6321.1s
k=300	77.1s	132.7s	3300.3s	6408.4s

The evaluation was performed on a popular MovieLens 1m dataset. This dataset contains 6,040 users and 3,706

Continued on next page

movies (items), with 1,000,209 ratings. The results can be found in the table above, showing execution time of the algorithm in seconds.

As it can be seen, we have achieved a significant speed-up. However, the speed is only one side of the problem with this classic implementation. As we have already mentioned, another concern is space, i.e. what to do when we run out of memory in case matrices become too large. In the next section, we will introduce a new approach to make it feasible to train CF recommender even on large datasets, on which classic implementation might run out of memory.

BUILD A RECOMMENDER ON LARGE DATASETS

In this test, we used the MovieLens 10m data set. Just to recall, all algorithms were run on a single machine with 16 GB of RAM, and evaluated using 10-fold cross validation. In such a setup, the 'recommenderlab' implementation cannot be used on this data set (at least for user-based CF, since it runs out of memory when the similarities matrix needs to be calculated).

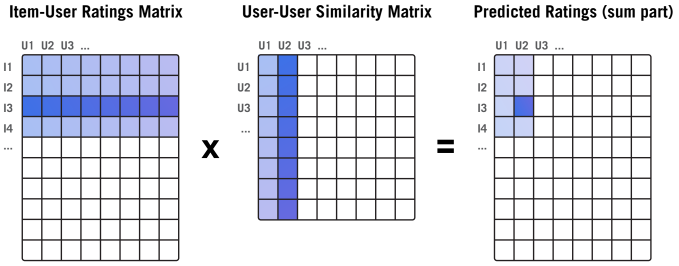
In our implementation, we tried to solve the problem of large matrices by dividing them into parts, i.e. we did not calculate all predictions at once, but in chunks. Here is the procedure for user-based CF (the same approach is used for item-based CF):

1. Take N rows (items) of the item-user matrix. In the picture, we took rows of indices [I1:I4]

2. Take M users and calculate similarities between them and all other users. In the picture, we calculated similarities for users [U1:U2].

3. Calculate the product of N rows from step 1 and M columns from step 2. The result is the MXN matrix, which contains the sums used to find the predictions. In our example, those will be predictions about how items I1 to I4 will be rated by users U1 and U2.

4. Repeat the first three steps for different N and M chunks until the result matrix is fully covered.



RESULTS ON MOVIELENS 10M DATASET

This dataset contains 69,878 users and 10,677 movies with around 10,000,054 ratings. Here is the evaluation setup:

- 10-fold cross validation, as explained above
- Center normalization
- Cosine measure to calculate similarities
- k: The number of nearest neighbors was set to 100 and 1000.
- Chunk size: The numbers of rows and columns per chunk are the parameters of the algorithm. We used some values that we found to be nearly optimal for our hardware setup.

The results of the comparison can be found in the following table, showing the execution time of the algorithm in minutes.

	Our implementation (Execution time)	
	Item-based CF	User-based CF
k=100	10.53m	178.43m
k=1000	23.33m	191.86m

As we can observe, the algorithm was executed successfully. With this current implementation, when we need to find recommendations in real-time for one or several users, the calculation of similarities and predictions will be much faster, since we will operate on a small number of users. On the MovieLens 10m dataset, user-based CF takes a second to find predictions for one or several users, while item-based CF takes around 30 seconds. This can be optimized further by storing the similarity matrix as a model, rather than calculating it on the fly. Additionally, an obvious advantage of this algorithm is that it is scalable. Since we calculate predictions on chunks of matrices independently, it is suitable to be parallelized. One of the next steps is to implement and test this approach on some distributed framework. The code is freely available as a public GitHub project. In the near future, we plan to work on this implementation further, extend the project with new algorithms, and possibly publish it as an R package.

STEFAN NIKOLIC is a data scientist in SmartCat (Serbia) where his current focus is on user behavior analytics and recommender systems. His general areas of interest are machine learning, data analysis, and algorithms. Besides data science, he has a strong background and experience in software development and big data technologies.

Diving Deeper

INTO BIG DATA

TOP #BIGDATA TWITTER FEEDS

To follow right away

@BigDataGal

@AndrewYNg

@KirkDBorne

@drob

@aditdeshpande3

@Sarahetodd

@data_nerd

@karpathy

@craigbrownphd

@Ronald_vanLoon

TOP BIG DATA REFCARDZ

Machine Learning: Patterns for Predictive Analytics

dzone.com/refcardz/machine-learning-predictive

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

Apache Spark: An Engine for Large-Scale Data Processing

dzone.com/refcardz/apache-spark

Introduces Spark, explains its place in big data, walks through setup and creation of a Spark application, and explains commonly used actions and operations.

R Essentials

dzone.com/refcardz/r-essentials-1

R is a highly extensible, open-source programming language used mainly for statistical analysis and graphics. R has become a widely popular language because of its varying data structures, which can be more intuitive than data storage in other languages; its built-in statistical and graphical functions; and its large collection of useful plugins that can enhance the language's abilities in many different ways.

BIG DATA ZONES

Learn more & engage your peers in our big data-related topic portals

Big Data dzone.com/bigdata

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

Database dzone.com/database

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

IoT dzone.com/iot

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

BIG DATA WEBSITES

Dataversity dataversity.net

Revolution Analytics blog.revolutionanalytics.com

DataTau datatau.com

BIG DATA PODCASTS

Linear Digressions lineardigressions.com

Data Skeptic dataskeptic.com

Partially Derivative partiallyderivative.com

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.

FRAMEWORKS

COMPANY NAME	PRODUCT	DESCRIPTION	WEBSITE
Alluxio Open Foundation	Alluxio	Distributed storage system across all store types	alluxio.org
Apache Foundation	Ambari	Hadoop cluster provisioning, management, and monitoring	ambari.apache.org
Apache Foundation	Apex	Stream+batch processing on YARN	apex.apache.org
Apache Foundation	Avro	Data serialization system (data structure, binary format, container, RPC)	avro.apache.org
Apache Foundation	Beam	Programming model for batch and streaming data processing	beam.apache.org
Apache Foundation	Crunch	Java library for writing, testing, running MapReduce pipelines	crunch.apache.org
Apache Foundation	Drill	Distributed queries on multiple data stores and formats	drill.apache.org
Apache Foundation	Falcon	Data governance engine for Hadoop clusters	falcon.apache.org
Apache Foundation	Flink	Streaming dataflow engine for Java	flink.apache.org
Apache Foundation	Flume	Streaming data ingestion for Hadoop	flume.apache.org
Apache Foundation	Giraph	Iterative distributed graph processing framework	giraph.apache.org
Apache Foundation	GraphX	Graph and collection processing on Spark	spark.apache.org/graphx
Apache Foundation	GridMix	Benchmark for Hadoop clusters	hadoop.apache.org/docs/r1.2.1/gridmix.html
Apache Foundation	Hadoop	MapReduce implementation	hadoop.apache.org
Apache Foundation	Hama	Bulk synchronous parallel (BSP) implementation for big data analytics	hama.apache.org
Apache Foundation	HAWQ	Massively parallel SQL on Hadoop	hawq.incubator.apache.org
Apache Foundation	HDFS	Distributed file system (Java-based, used by Hadoop)	hadoop.apache.org
Apache Foundation	Hive	Data warehousing framework on YARN	hive.apache.org

SOLUTIONS DIRECTORY: FRAMEWORKS

COMPANY NAME	PRODUCT	DESCRIPTION	WEBSITE
Apache Foundation	Ignite	In-memory data fabric	ignite.apache.org
Apache Foundation	Impala	Distributed SQL on YARN	impala.apache.org
Apache Foundation	Kafka	Distributed pub-sub messaging	kafka.apache.org
Apache Foundation	MADlib	Big data machine learning w/SQL	madlib.incubator.apache.org
Apache Foundation	Mahout	Machine learning and data mining on Hadoop	mahout.apache.org
Apache Foundation	Mesos	Distributed systems kernel (all compute resources abstracted)	mesos.apache.org
Apache Foundation	Oozie	Workflow scheduler (DAGs) for Hadoop	oozie.apache.org
Apache Foundation	ORC	Columnar storage format	orc.apache.org
Apache Foundation	Parquet	Columnar storage format	parquet.apache.org
Apache Foundation	Phoenix	SQL->HBase scans->JDBC result sets	phoenix.apache.org
Apache Foundation	Pig	Turns high-level data analysis language into MapReduce programs	pig.apache.org
Apache Foundation	Samza	Distributed stream processing framework	samza.apache.org
Apache Foundation	Spark	General-purpose cluster computing framework	spark.apache.org
Apache Foundation	Spark Streaming	Discretized stream processing with Spark's RDDs	spark.apache.org/streaming
Apache Foundation	Sqoop	Bulk data transfer between Hadoop and structured datastores	sqoop.apache.org
Apache Foundation	Storm	Distributed realtime (streaming) computing framework	storm.apache.org
Apache Foundation	Tez	Dataflow (DAG) framework on YARN	tez.apache.org
Apache Foundation	Thrift	Data serialization framework (full-stack)	thrift.apache.org
Apache Foundation	YARN	Resource manager (distinguishes global and per-app resource management)	hadoop.apache.org/docs/r2.7.1/hadoop-yarn/hadoop-yarn-site/YARN.html
Apache Foundation	Zeppelin	Interactive data visualization	zeppelin.apache.org
Apache Foundation	Zookeeper	Coordination and State Management	zookeeper.apache.org
Chart.js	Chart.js	Simple JavaScript charting library	chartjs.org
D3.js	D3.js	Declarative-flavored JavaScript visualization library	d3js.org
Disco Project	Disco	MapReduce framework for Python	discoproject.org
Druid	Druid	Columnar distributed data store w/realtime queries	druid.io
Eclipse Foundation	BIRT	Visualization and reporting library for Java	eclipse.org/birt
EnThought	SciPy	Scientific computing ecosystem (multi-dimensional arrays, interactive console, plotting, symbolic math, data analysis) for Python	scipy.org

SOLUTIONS DIRECTORY: FRAMEWORKS

COMPANY NAME	PRODUCT	DESCRIPTION	WEBSITE
Facebook	Presto	Distributed interactive SQL on HDFS	prestodb.io
GFS2 Group	GFS	(Global File System) Shared-disk file system for Linux clusters	git.kernel.org/cgit/linux/kernel/git/gfs2/linux-gfs2.git/?h=for-next
Google	Protocol Buffers	Data serialization format & compiler	developers.google.com/protocol-buffers/docs/overview
Google	TensorFlow	Computation using dataflow graphs (nodes are math operations, edges are tensors between operations)	tensorflow.org
GraphViz	GraphViz	Graph(nodes+edges) visualization toolkit	graphviz.org
H2O	H2O	Stats, machine learning, math runtime for big data	h2o.ai
JavaML	Java-ML	Various machine learning algorithms for Java	java-ml.sourceforge.net
JUNG Framework	JUNG Framework	Graph(nodes+edges) framework (model, analyze, visualize) for Java	jung.sourceforge.net
LinkedIn	Pinot	Realtime OLAP distributed data store	github.com/linkedin/pinot
LISA Lab	Theano	Python library for multi-dimensional array processing w/GPU optimizations	deeplearning.net/software/theano
Microsoft	SSRS	SQL Server reporting (server-side)	microsoft.com/en-us/sql-server/default.aspx
Misco	Misco	MapReduce Framework	alumni.cs.ucr.edu/~jdou/misco
NumFocus	Julia	Dynamic programming language for scientific computing	julialang.org
NumFocus	Matplotlib	Plotting library on top of NumPy (like parts of MATLAB)	matplotlib.org
NumFocus	NumPy	Mathematical computing library (multi-dimensional arrays, linear algebra, Fourier transforms, more) for Python	numpy.org
NumFocus	Pandas	Data analysis and modeling for Python	pandas.pydata.org
OpenTSDB Authors	OpenTSDB	Time-series database on Hadoop	opentsdb.net
Project Jupyter	Jupyter	Interactive data visualization and scientific computing on Spark and Hadoop	jupyter.org
Red Hat	Ceph	Distributed object and block store and file system	ceph.com
Sencha	InfoVis Toolkit	JavaScript visualization library	philogb.github.io/jit
Tableau	Tableau	Interactive data visualization for BI	tableau.com
The R Foundation	R	Language and environment for statistical computing and graphics	r-project.org
University of Waikato	Weka	Machine learning and data mining for Java	cs.waikato.ac.nz/ml/weka
Wolfram	Wolfram Language	Knowledge-based programming language w/many domain-specific libraries	wolfram.com/language
Xplenty	Cascading	Platform to develop big data applications on Hadoop	cascading.org
YCSB	YCSB	General-purpose benchmarking spec	github.com/brianfrankcooper/YCSB/wiki/Getting-Started

PLATFORMS

COMPANY NAME	PRODUCT	DESCRIPTION	WEBSITE
1010data	Insights Platform	Data management, analysis, modeling, reporting, visualization, RAD apps	1010data.com/products/insights-platform/analysis-modeling
Actian	Vector	DBMS, column store, analytics platform	actian.com/products/actian-analytics-databases/vector-smp-analytics-database/
Aginity	Aginity Amp	Data analytics management platform	aginity.com/amp-overview
Alation	Alation	Enterprise data collaboration and analytics platform	alation.com/product
Alpine Data	Alpine Chorus 6	Data science, ETL, predictive analytics, execution workflow design and management	alpinedata.com/product
Alteryx	Alteryx Analytics Platform	ETL, predictive analytics, spatial analytics, automated workflows, reporting and visualization	alteryx.com/products/alteryx-designer
Amazon Web Services	Amazon Kinesis	Stream data ingestion, storage, query, and analytics PaaS	aws.amazon.com/kinesis
Amazon Web Services	Amazon Machine Learning	Machine learning algorithms-as-a-service, ETL, data visualization, modeling and management APIs, batch and realtime predictive analytics	aws.amazon.com/machine-learning
Attunity	Attunity Visibility	Data warehouse and Hadoop data usage analytics	attunity.com/products/visibility
Attunity	Attunity Replicate	Data replication, ingestion, and streaming platform	attunity.com/products/replicate
BigML	BigML	Predictive analytics server and development platform	bigml.com
Birst	Birst	Enterprise and embedded BI and analytics platform	birst.com
Bitam	Artus	Business intelligence platform	bitam.com/artus
Board	BOARD All in One	BI, analytics, corporate performance management platform	board.com/en/product
CAPSENTA	Ultrawrap	Database wrapper for lightweight data integration	capsenta.com
Cask Data	Cask	Containers (data, programming, application) on Hadoop for data lakes	cask.co
Cask Data	Cask Data App Platform	Analytics platform for YARN with containers on Hadoop, visual data pipelining, data lake metadata management	cask.co
Cazena	Cazena	Cloud-based data science platform	cazena.com/what-is-cazena
Cirro	Cirro Data Cloud	Data virtualization and governance platform	cirro.com/products.html#data-cloud
Cisco	Cisco Edge Analytics Fabric	IoT and streaming data analytics	cisco.com/c/en/us/products/analytics-automation-software/edge-analytics-fabric
Cisco	Cisco Data Virtualization	ETL, data virtualization and integration platform	cisco.com/c/en/us/products/analytics-automation-software/data-virtualization/
Cloudera	Cloudera Enterprise Data Hub	Predictive analytics, analytic database, and Hadoop distribution	cloudera.com/products/enterprise-data-hub.html
Confluent	Confluent Platform	Data integration, streaming data platform	confluent.io/product/

SOLUTIONS DIRECTORY: PLATFORMS

COMPANY NAME	PRODUCT	DESCRIPTION	WEBSITE
Databricks	Databricks	Data science (ingestion, processing, collaboration, exploration, and visualization) on Spark	databricks.com/product/databricks
Dataguise	Dataguise DgSecure	Big Data security monitoring	dataguise.com
Datameer	Datameer	BI, data integration, ETL, and data visualization on Hadoop	datameer.com/product/product-overview
DataRobot	DataRobot	Machine learning model-building platform	datarobot.com/product
DataRPM	DataRPM	Cognitive predictive maintenance for industrial IoT	datarpm.com/platform.php
DataTorrent	DataTorrent RTS	Stream and batch (based on Apache Apex) application development platform	datatorrent.com/products-services/datatorrent-rts
DataWatch	DataWatch Monarch	Data extraction and wrangling, self-service analytics, streaming visualization	datawatch.com/our-platform/monarch
Domo	Domo	Data integration, preparation, and visualization	domo.com/product
EngineRoom.io	EngineRoom	Geospatial, data transformation and discovery, modeling, predictive analytics, visualization	engineroom.io
Exaptive	Exaptive	RAD and application marketplace for data science	exaptive.com/platform
EXASOL	EXASOL	In-memory analytics database	exasol.com/en/product
Fair Isaac Corporation	FICO Decision Management Suite	Data integration, analytics, decision management	fico.com/en/analytics/decision-management-suite
GoodData	GoodData Platform	Data distribution, visualization, analytics (R, MAQL), BI, warehousing	gooddata.com/platform
H2O.ai	H2O	Open source prediction engine on Hadoop and Spark	h2o.ai
Hewlett Packard Enterprise	HPE Haven	Data integration, analytics (SQL and unstructured), exploration	saas.hpe.com/en-us/software/big-data-platform-haven
Hewlett Packard Enterprise	HPE IDOL	Machine learning, enterprise search, and analytics platform	saas.hpe.com/en-us/software/information-data-analytics-idol
Hewlett Packard Enterprise	HPE Vertica Analytics Platform	Distributed analtyics database and SQL analytics on Hadoop	vertica.com/overview
Hitachi Group	Pentaho	Data integration layer for Big Data analytics	pentaho.com/product/product-overview
Hortonworks	Hortonworks Data Platform	Hadoop distribution based on YARN	hortonworks.com/products/data-center/hdp
Hortonworks	Hortonworks DataFlow	Streaming data collection, curation, analytics, and delivery	hortonworks.com/products/data-center/hdf
IBM	IBM BigInsights	Scalable data processing and analytics on Hadoop and Spark	ibm.com/analytics/us/en/technology/biginsights
IBM	IBM Streams	Streaming data application development and analytics platform	ibm.com/software/products/en/ibm-streams
IBM	IBM InfoSphere	Data integration, management, governance, data warehousing	ibm.com/software/products/en/category/bigdata

SOLUTIONS DIRECTORY: PLATFORMS

COMPANY NAME	PRODUCT	DESCRIPTION	WEBSITE
Infobright	Infobright Enterprise	Column-oriented store with semantic indexing and approximation engine for analytics	infobright.com/infobright-enterprise-edition
Informatica	Intelligent Data Lake	Collaborative, centralized data lake, data governance	informatica.com/products/big-data/intelligent-data-lake.html
Informatica	Big Data Management	Data integration platform on Hadoop	informatica.com/products/big-data/big-data-edition.html
Informatica	Relate 360	Big Data analytics, visualization, search, and BI	informatica.com/products/big-data/big-data-relationship-manager.html
Informatica	Intelligent Streaming	Event processing and streaming data management for IoT	informatica.com/products/big-data/intelligent-streaming.html
Information Builders	WebFOCUS	BI and analytics	informationbuilders.com/products/intelligence
Information Builders	Omni-Gen	Data management, quality, integration platform	informationbuilders.com/products/omni
Intersystems	DeepSee	Real-time transactional data analytics	intersystems.com/our-products/embedded-technologies/deepsee
Jinfonet	JReport	Visualization, embedded analytics for web apps	jinfonet.com/product
Kognitio	Kognitio Analytical Platform	In-memory, MPP, SQL and NoSQL analytics on Hadoop	kognitio.com/analyticalplatform
Lavastorm	Lavastorm Server	Data preparation, analytics application development platform	lavastorm.com/product/explore-lavastorm-server
LexisNexis	LexisNexis Customer Data Management	Data management and migration	lexisnexis.com/risk/customer-data-management
Liaison Technologies	Liaiason Alloy	Data management and integration	liaison.com/liaison-alloy-platform
Lightbend	Lightbend Reactive Platform	JVM application development platform with Spark	lightbend.com/platform
Loggly	Loggly	Cloud log management and analytics	loggly.com/product
Logi Analytics	Logi	Embedded BI, data discovery	logianalytics.com/products/info
Looker	Looker Platform	Data integration, governance, and visualization	looker.com/product/simple-platform
MapR	MapR Converged Data Platform	Big Data platform on enterprise-grade Hadoop distribution with integrated open-source tools (Spark, Hive, Impala, Solr, etc.), NoSQL (document and wide column) DBMS	mapr.com/products
Microsoft	Cortana Intelligence Suite	Predictive analytics and machine learning development platform	azure.microsoft.com/en-us/services/machine-learning
Microsoft	Power BI	Business intelligence	powerbi.microsoft.com
MicroStrategy	MicroStrategy	Data management, analytics, BI, and MDM	microstrategy.com/us/platform
New Relic	New Relic Insights	Real-time application performance analytics	newrelic.com/insights
Objectivity	ThingSpan	Graph analytics platform with Spark and HDFS integration	objectivity.com/products/thingspan

SOLUTIONS DIRECTORY: PLATFORMS

COMPANY NAME	PRODUCT	DESCRIPTION	WEBSITE
OpenText	OpenText Big Data Analytics	Analytics and visualization (GUI->code) with analyatics server	opentext.com/what-we-do/products/analytics/opentext-big-data-analytics
Oracle	Big Data Discovery	Big Data analytics and visualization platform on Spark	oracle.com/big-data/big-data-discovery
Oracle	R Advanced Analytics for Hadoop	R interface for manipulating data on Hadoop	oracle.com/technetwork/database/database-technologies/bdc/r-advanalytics-for-hadoop/overview
Palantir	Gotham	Cluster data store, on-the-fly data integration, search, in-memory DBMS, ontology, distributed key-value store	palantir.com/palantir-gotham
Palantir	Metropolis	Big Data analytics, integration, visualization, and modeling	palantir.com/palantir-metropolis
Panoply	Panoply	Data management and analytics platform	panoply.io
Panorama Software	Necto	Business Intelligence, visualization, data management	panorama.com/necto
Paxata	Paxata Adaptive Information Platform	Data integration, preparation, exploration, visualization on Spark	paxata.com/product/paxata-adaptive-information-platform
Pepperdata	Pepperdata Cluster Analyzer	Big Data performance analytics	pepperdata.com/products/cluster-analyzer
Pivotal	Pivotal Greenplum	Open source data warehouse and analytics	pivotal.io/pivotal-greenplum
Pivotal	Spring Cloud Data Flow	Cloud platform for building streaming and batch data pipelines and analytics	cloud.spring.io/spring-cloud-dataflow
Prognoz	Prognoz Platform	BI and analytics (OLAP, time series, predictive)	prognoz.com
Progress Software	DataDirect Connectors	Data integration: many-source, multi-interface (ODBC, JDBC, ADO.NET, OData), multi-deployment	progress.com/datadirect-connectors
Pyramid Analytics	BI Office	Data discovery and analytics platform	pyramidanalytics.com/pages/bi-office.aspx
Qlik	Qlik Sense	Data visualization, integration, and search	qlik.com/us/products/qlik-sense
Qlik	QlikView Guided Analytics	Business intelligence application platform	qlik.com/us/products/qlikview
Qubole	Qubole Data Service	Data engines for Hive, Spark, Hadoop, Pig, Cascading, Presto on AWS, Azure, Google Cloud	qubole.com
Rapid7	Logentries	Log management and analytics	logentries.com
RapidMiner	RapidMiner Studio	Predictive analytics workflow and model builder	rapidminer.com/products/studio
RapidMiner	RapidMiner Radoop	Predictive analytics on Hadoop and Spark with R and Python support	rapidminer.com/products/radoop
RedPoint	RedPoint Data Management	Data management, quality, integration (also on Hadoop)	redpoint.net/products/data-management-solutions
SAP	SAP HANA	In-memory, column-oriented, relational DBMS (cloud or on-premise) with text search, analytics, stream processing, R integration, graph processing	sap.com/product/technology-platform/hana.html

SOLUTIONS DIRECTORY: PLATFORMS

COMPANY NAME	PRODUCT	DESCRIPTION	WEBSITE
SAS	SAS Platform	Analytics, BI, data management, deep statistical programming	sas.com/en_us/software/sas9.html
Sisense	Sisense	Analytics, BI, visualization, reporting	sisense.com/product
Skytree	Skytree	Machine Learning platform with self-service options	skytree.net
Software AG	Terracotta In-Memory Data Management by Software AG	In-memory data management, job scheduler, Ehcache implementation, enterprise messaging	terracotta.org
Splunk	Splunk Enterprise	Operational intelligence for machine-generated data	splunk.com/en_us/products/splunk-enterprise.html
Stitch	Stitch	ETL-as-a-service	stitchdata.com
StreamSets	Dataflow Performance Manager	Data management and analytics platform	streamsets.com/products/dpm
Sumo Logic	Sumo Logic	Log and time-series management and analytics	sumologic.com
Tableau	Tableau Desktop	Visualization, analytics, exloration (with self-service, server, hosted options)	tableau.com
Talend	Talend Data Fabric	Real-time or batch data management platform	talend.com/products/data-fabric
Talend	Talend Open Studio	ELT and ETL on Hadoop with open source components	talend.com/download/talend-open-studio
Tamr	Tamr	Data management, sanitation, analytics, BI	tamr.com/product
Targit	Targit Decision Suite	BI, analytics, discovery front-end with self-service options	targit.com/en/software/decision-suite
Teradata	Teradata	Data warehousing, analytics, lake, SQL on Hadoop and Cassandra, Big Data appliances, R integration, workload management	teradata.com
Thoughtspot	Thoughtspot	Relational search engine	thoughtspot.com/product
Tibco	Jaspersoft	BI, analytics (OLAP, in-memory), ETL, data integration (relational and non-relational), reporting, visualization	jaspersoft.com/business-intelligence-solutions
TIBCO	TIBCO Spotfire Platform	Data mining and visualization	spotfire.tibco.com
Treasure Data	Treasure Data	Analytics infrastructure as a service	treasuredata.com
Trifacta	Trifacta Wrangler	Data wrangling, exploration, visualization on Hadoop	trifacta.com
Unravel	Unravel	Predictive analytics and machine learning performance monitoring	unraveldata.com/product
Waterline Data	Waterline Data	Data marketplace (inventory, catalogue with self-service) on Hadoop	waterlinedata.com/product-overview
Workday	Workday Prism Analytics	Data preparation, discovery, and analytics on Hadoop and Spark	workday.com/en-us/applications/prism-analytics.html
Yellowfin	Yellowfin	Business Intelligence, Data Visualization	yellowfinbi.com/platform
Zaloni	Zaloni	Enterprise data lake management	zaloni.com
Zoomdata	Zoomdata	Analytics, visualization, BI with self-service on Hadoop, Spark, many data stores	zoomdata.com

GLOSSARY

ALGORITHM
A series of instructions used to solve a mathematical problem.

APACHE HADOOP
An open-source tool to process and store large distributed data sets across machines by using MapReduce.

APACHE NIFI
An open source Java server that enables the automation of data flows between systems in an extensible, pluggable, open manner. NiFi was open sourced by the NSA.

APACHE SPARK
An open-source Big Data processing engine that runs on top of Apache Hadoop, Mesos, or the cloud.

ARTIFICIAL INTELLIGENCE
The ability of a machine to recognize its environment, act in a rational way, and maximize its ability to solve problems.

BACKPROPAGATION
The process by which a neural network will auto-adjust its parameters until it can consistently produce the desired outcome with sufficient reliability.

BIG DATA
A common term for large amounts of data. To be qualified as big data, data must be coming into the system at a high velocity, with large variation, or at high volumes.

CLUSTER
A subset of data that share particular characteristics. Can also refer to several machines that work together to solve a single problem.

COAP
Constrained Application Protocol is an Internet Applicaton protocol for limited resource devices that can be translated to HTTP if needed.

COLLABORATIVE FILTERING
A technique used by recommender systems to identify information or patterns in how several users may rate objects.

DATA FLOW MANAGEMENT
The specialized process of ingesting raw device data, while managing the flow of thousands of producers and consumers. Then performing basic data enrichment, analysis in stream, aggregation, splitting, schema translation, format conversion and other initial steps to prepare the data for further business processing.

DATA GOVERNANCE
The process of managing the availability, usability, integrity, and security of data within a Data Lake.

DATA LAKE
A storage repository that holds raw data in its native format.

DATA SCIENCE
A field that explores repeatable processes and methods to derive insights from data.

DEVICE LAYER
The entire range of sensors, actuators, smartphones, gateways and industrial equipment that send data streams corresponding to their environment and performance characteristics.

IIOT
The Industrial Internet of Things refers to the business platform being built in a specific industry to take advantage of data from industrial devices.

MACHINE LEARNING
An AI that is able to learn by being exposed to new data, rather than being specifically programmed.

MQTT
Message Queue Telemetry Transport is a small common Internet messaging protocol often used within the IoT for pub-sub messaging on TCP/IP.

NEURAL NETWORK A system modeled on the brain and nervous system of humans, where processors operate parallel to each other and are organized in tiers, and each processor receives information from the processor preceding it in the system.

NOSQL DATABASE
Short for “not only SQL”, or any database that uses a system to store and search data other than just using tables and structured query languages.

PARSE
To divide data, such as a string, into smaller parts for analysis.

REAL-TIME STREAM
Processing A model for analyzing sequences of data by using machines in parallel, though with reduced functionality.

RELATIONAL DATABASE MANAGEMENT SYSTEM (RDBMS)
A system that manages, captures, and analyzes data that is grouped based on shared attributes called relations.

RECOMMENDER SYSTEMS
A machine learning system that predicts the rating that a user may give a particular item or data point.

SMART DATA
Digital information that is formatted so it can be acted upon at the collection point before being sent to a downstream analytics platform for further data consolidation and analytics.

STRUCTURED QUERY LANGUAGE (SQL)
The standard language used to retrieve information from a relational database.

ZIGBEE
A specification for building local low energy wireless networks with tiny radios often used in IoT.

ZONES
Distinct areas within a Data Lake that serve specific, well-defined purposes.