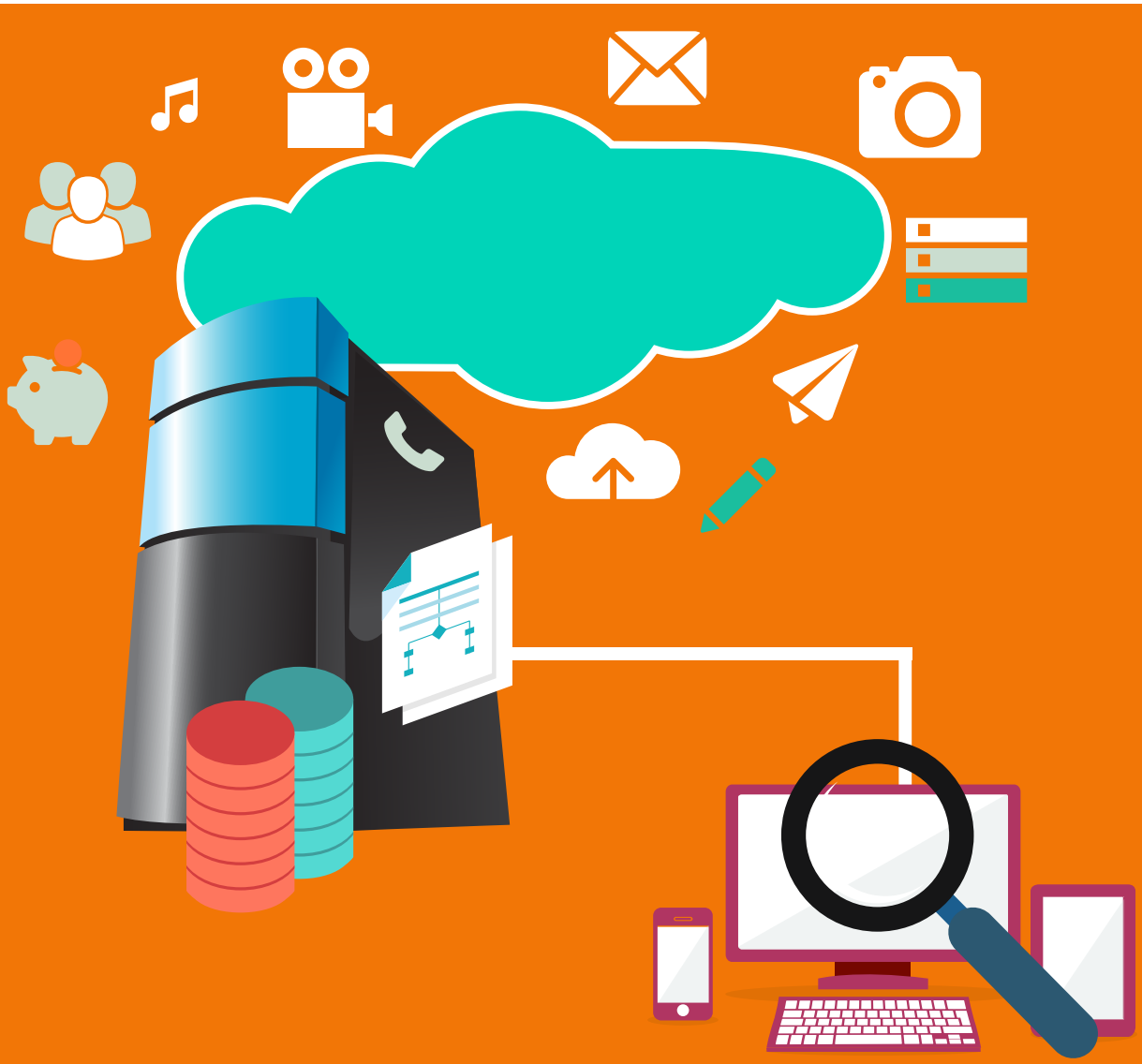
happiest minds

Building Big Data Applications: Getting started with Apache Spark



Happiest People Happiest Customers

Contents

Brief history of big data:

The rise of petabyte-scale data sets

These days business organizations are churning out huge amounts of transactional data, capturing trillions of bytes of infor-mation about their customers, suppliers, and operations. While this may have once concerned only a few data geeks, big data is now relevant across business sectors; consumers of products and services. Everyone stands to benefit from its application. The combination of massive data sets and the rapid development of new technologies capable of storing and processing infor-mation out of them have already transformed the way businesses operate.

Over the last decades, internet giants like Amazon, Google, Yahoo!, eBay and Twitter have invented and used tools for work-ing with colossal data sets that was beyond the realm of traditional data management tools.

Some of the leading open-source tools for big data include Hadoop and NoSQL databases like Cassandra and MongoDB. Hadoop has been a spectacular success, offering cheap storage in HDFS (Hadoop distributed file system) of datasets up to Petabytes in size, with batch-mode ("offline") analysis using Map Reduce jobs.

Trends in big data:

In the past, emerging technologies took years to mature. In the case of big data, while effective tools are still emerging, the analytics requirements are changing rapidly resulting in businesses to either make it or be left behind.

Various tools have emerged in the big data space to address different use-cases for building NRT (near real time), advance analytics, recommendation applications and many more making it complex to integrate these various tools and different storage layers. At some point during this evolution, a powerful yet simple, open source framework called Apache Spark was introduced to address various common use cases of big data under one umbrella.

Besides Apache Spark, another next generation tool called Apache Flink, formerly known as Stratosphere, is also available. Apache Flink is almost similar to Apache Spark except in the way it handles streaming data; however it is still not as mature as Apache Spark as a big data tool.

Both Apache Spark and Apache Flink have the capability to build interactive, real time applications.

Since stable version of Apache Flink was released only on Nov 15th and there is no commercial vendor support, we will discuss this in our subsequent papers.

How Spark addresses the big data trends

Various projects like MLlib, Spark Streaming, Spark Sql, GraphX, Spark DataFrames, and MLPipelines are built on the core Spark API.

Spark:
Spark provides core API to process data in a distributed environment. It offers various high level API like map, reduce, filter, reduce by key to build distributed applications. Businesses can use these APIs for ETL pipelines and Data exploration. All projects in the Spark ecosystem use these core APIs to build specific systems for machine learning, interactive queries and graph based analytics.

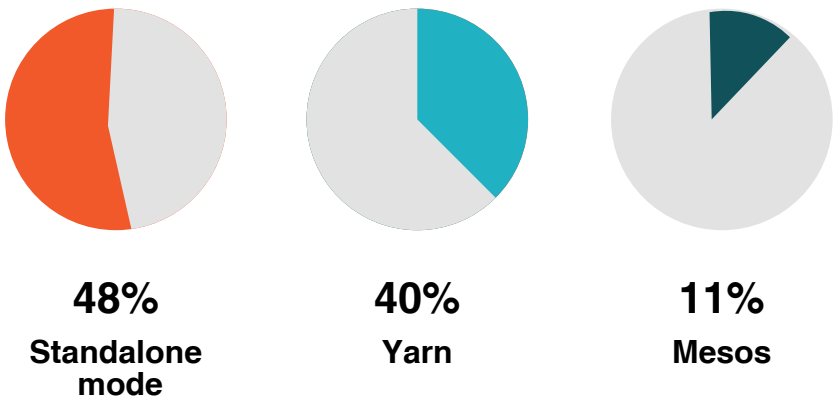
Spark Sql:
Spark Sql is a spark module which provides SQL and data frame interfaces for structured data. This is an interesting capability which allows use of Sql or Data frames API over any structured intermediate data. It also allows usage from any underlying data source like hive, parquet, ORC, NoSQL and SQL databases. Use of data frames / SQL helps spark optimize the performance since it is aware of the underlying schema. It is preferable to use Data frame API over the core API as one gains performance improvement in that process.

Spark MLlib:
MLlib module provides help with scalable machine learning algorithms for supervised learning, unsupervised learning and data mining. It is a fast growing package of distributed machine learning algorithms that has its own advantages.

Spark Streaming:
Spark streaming module helps in building applications to process real time data with a latency of above 0.5ms. Based on micro batch style computing, it allows windows function and has inbuilt capability to process data from Apache Kafka, Flume and many other such data sources.

Spark GraphX:
GraphX is a new component for graph parallel computation. GraphX includes a growing collection of graph algorithms and builders to simplify graph analytic tasks.

Most common spark development environments (cluster Managers)



-

Image courtesy Spark-Survey-2015-Infographic

Common Misconceptions:

As with everything there are some common misconceptions about Spark which can be dispelled.
a) Spark expects the data to sit in memory: Not true. Spark can use memory for holding the data sets if required. It can work on disk too. There is a huge performance boost when data is cached, particularly for iterative machine learning algorithms and interactive queries.
b) Do I need to invest in new hardware: No. One can use existing Hadoop/NoSQL deployment. Spark runs as another service in the existing deployment. It can work independently too when starting with distributed systems.

c) Can I use my existing skill set: Yes. Spark exposes its APIs in 4 different languages (Scala, Java, Python and R). In that sense, small learning curve is required to get started with Spark and some extensive training if one is well versed is any of the above mentioned languages.

Preferable Languages: Of all the languages, Scala and Python is the most preferable when to comes to using Spark. Scala and Python are the most preferred as they provide ease while building data products. All the new features on Spark s are generally made available initially in Scala and Java as Spark itself is built in Scala.

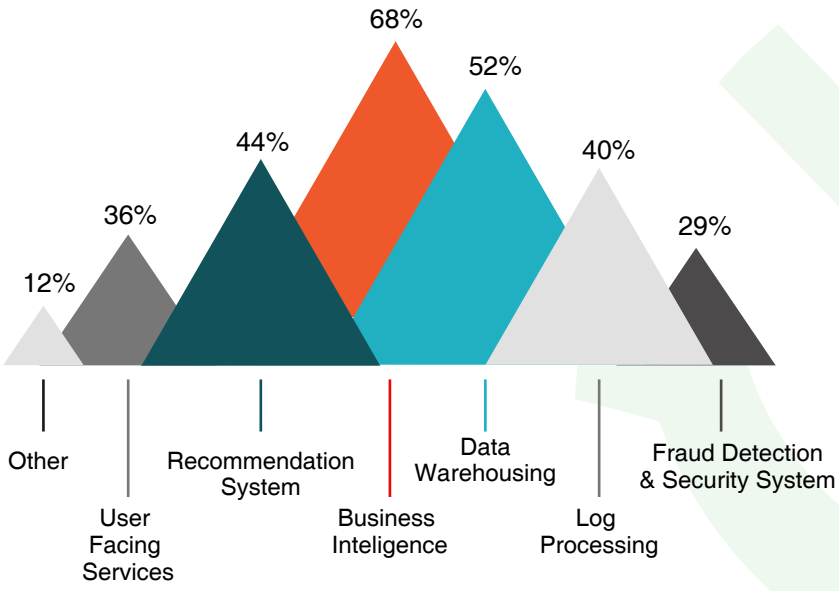
Hadoop and Spark:

If you already have a Hadoop deployment, Spark perfectly sits in your ecosystem. It is well integrated with Yarn, HBase, Hdfs and Hive. It can leverage existing infrastructure and resource management capabilities of Yarn. Major Hadoop distributions like Cloudera, Hortonworks, and MapR currently support Apache Spark and promote it.

Besides Hadoop:

Spark works closely with many of the NoSQL and storage layers like Cassandra, MongoDB, Elastic Search and any database which provides a JDBC driver. It is well integrated with Apache Mesos which works similar to YARN but not confined to only Hadoop ecosystem.

spark is used to create many types of products inside different organizations



http://cdn2.hubspot.net/hubfs/438089/DataBricks_Surveys_-_Content/Spark-Survey-2015-Infographic.pdf?t=1443057549926

Future of Apache Spark:

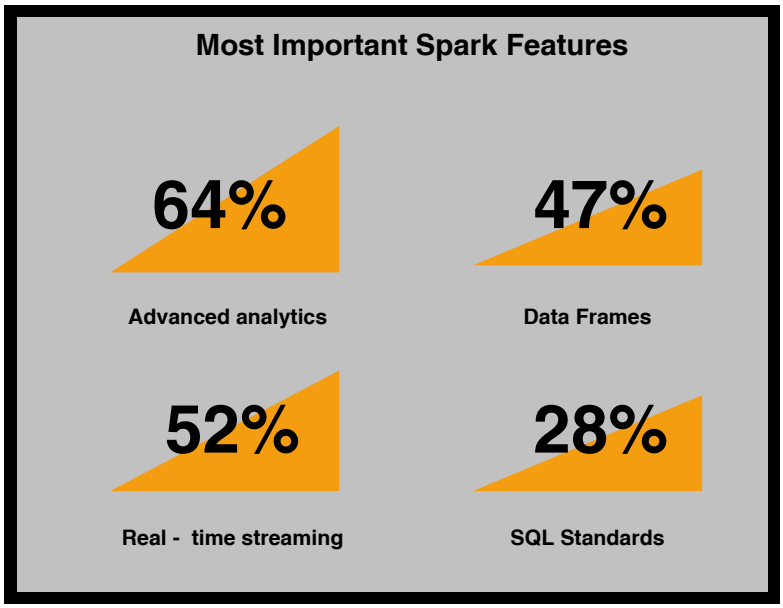
Apache Spark has some promising projects in the ecosystem. They are as follows:

DataFrames:

DataFrames API is inspired by DataFrames in R and Pandas in Python, it is being built from scratch on Big Data/Distributed systems. It exposes APIs for Python, Java, Scala and R. It uses state of the art optimizations and code generation through SQL catalyst optimizer.

MLPipelines:

Spark MLPipelines provides a uniform set of high level API built on top of DataFrame. It provides a set of reusable pipelines for various data science activities like data preparation, feature extraction, model training and validation.



Another interesting project which is not part of Apache Spark is KeystoneML which provides reusable Pipelines for various tasks like audio, text and images. It provides various examples which reproduce state-of-the-art results on public data sets. This project is also from Amp Labs.

Tungsten:

Project Tungsten brings in drastic change to the Spark's execution engine from the starting of the project.
Let us look into some factors that partake in achieving this.

- Memory Management and Binary Processing ► Used to maximize the usage of application semantics for memory man agement exclusively, along with removing the disadvantages of garbage collection and JVM object model.
- Cache-aware processing ► Helps in exploiting the memory hierarchy for data structures and algorithm.
- Code generation ► Helps in exploiting the compilers and CPUs

Spark applications, while using this will be benefited with efficient CPU and memory utilization.

Spark for mobiles:

As processing is moving more each day towards mobile based technology , the contributors of Spark are re-architecting Spark to fit the scenario and this ability is expected to be available from Spark version 2.0 . With this feature, mobiles can act as nodes for the spark cluster. This can change the way big data is currently being processed.

Use Cases:

Let us discuss some use case scenarios:

Interactive exploratory analysis:

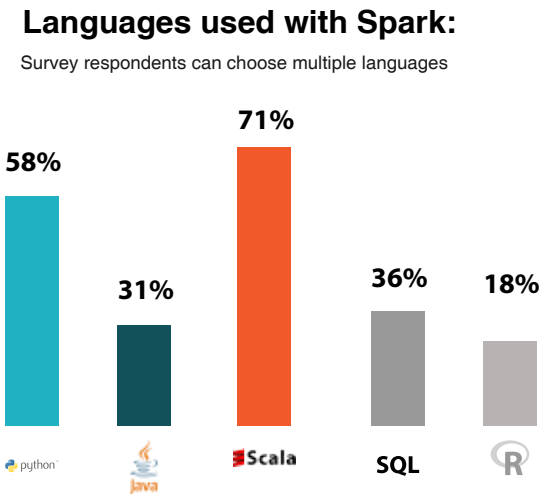
- Leverage Spark's in memory caching and powerful execution capabilities to explore large datasets.
- Use Spark's rich SQL, DataFrame and core API to explore various kind of structured, semi structured and unstructured datasets.
- Connect external applications like Tableau to power existing exploration through SQL drivers.

Faster ETL:

- Help port existing Hive scripts by changing the underlying execution engine from map-reduce to spark for an instant boost.
- Help migrate your pig scripts to Spark.
- Leverage Spark's optimized scheduling for efficient IO and in-memory processing capabilities on large data sets.

Real time dash boards:

- Use Spark streaming to perform near real time aggregations and window based aggregations.
- Integrate Spark SQL on real time data for interactive processing on real time data.
- Apply machine learning algorithms like clustering, classification, recommendations in NRT for identifying anomalies, recommending products.
- Since the same core is used for Spark API for batch and real time, most of the code can be reused instead of re-engi neering it separately for batch and streaming.



http://cdn2.hubspot.net/hubfs/438089/DataBricks_Surveys_-_Content/Spark-Survey-2015-

Machine learning:

Use out of box algorithms from Spark MLlib for supervised, unsupervised Data mining.

Use Spark's caching capabilities for improving performance of iterative algorithms.

Can take the advantage of some advance algorithms which can be learned as the data trickles in.

Conclusion

With the rate of information growth, businesses across verticals are challenged to finding actionable insight from the massive data that they have accumulated. Big data is here to stay and using the right tool can give any business a boost and help charter its growth.

We have seen a drastic improvement in the performance and a decrease in number of calendar days across various projects executed in Spark. Many applications are being migrated to Spark for the ease it offers to developers. Apache Spark is worth it.

About the Author

Vishnu Subramanian works as solution architect for Happiest minds with years of experience in building distributed systems using Hadoop, Spark, ElasticSearch , Cassandra, Machine Learning.A Databricks certified spark developer and having experience in building Data Products. His interests are in IOT, Data Science , BigData Security.

Vishnu Subramanian

Happiest Minds

Happiest Minds enables Digital Transformation for enterprises and technology providers by delivering seamless customer experience, business efficiency and actionable insights through an integrated set of disruptive technologies: big data analyt-ics, internet of things, mobility, cloud, security, unified communications, etc. Happiest Minds offers domain centric solutions applying skills, IPs and functional expertise in IT Services, Product Engineering, Infrastructure Management and Security. These services have applicability across industry sectors such as retail, consumer packaged goods, e-commerce, banking, insurance, hi-tech, engineering R&D, manufacturing, automotive and travel/transportation/ hospitality.Headquartered in Bangalore, India, Happiest Minds has operations in the US, UK, Singapore, Australia and has secured $ 52.5 million Series-A funding. Its investors are JPMorgan Private Equity Group, Intel Capital and Ashok Soota.

E-mail: Business@happiestminds.com

Visit us: www.happiestminds.com

Follow us on