A Guide to Data Analytics Toolbox

Mohammad Soltanieh-ha
Information Systems Department
Boston University Questrom School of Business

This short document intends to provide a brief overview of tools that different analytics roles use throughout a project. The document is organized into the following sections:

1. Storage and Compute Platforms
    A. On-premise data center
    B. Public cloud
    C. Hybrid or multi-cloud
2. Programming Languages
3. Databases
4. Big Data Platforms
5. Business intelligence dashboards
6. Software development tools

1. Storage and compute platforms

Any analytics project has particular computation and storage needs. These requirements could include infrastructural services such as access to computing and storage, managed platforms, or managed services. The platforms that are used today vary among different organizations but generally can be divided into three major categories:

A. On-premise data center
B. Public cloud
C. Hybrid or multi-cloud

Below, we will review these models to demonstrate the available options for running a data analytics pipeline.

A. On-premise data center

The on-premise data center model, also known as a private cloud, used to be the only possible way for companies and organizations to manage their data and computation needs. Since companies have made significant infrastructure investments in the past decades, many are still wholly or partially relying on their in-home data centers. These data centers are expensive and require a large IT team to maintain. The typical budgets for such facilities encompass the building and its security, hardware and upgrades, maintenance, and IT experts to keep the data center up. Depending on the company's size, having an on-premise data center may or may not be the right choice. With the availability of public clouds, the barrier has been significantly lowered for startups and smaller companies since they no longer need to invest in large servers while the company is still growing. This equation, however, may change by the time the company is grown enough and more computation is needed, in which case a private cloud might be more beneficial. After the appearance of the public cloud, many companies and organizations have been shifting to the cloud model due to the numerous benefits it offers. [3]

B. Public cloud

The public cloud computing platforms, or in short, the cloud, were first introduced in the late 20th century, but they didn't become widely available until 2002 when Amazon premiered its web services (AWS). While these few early cloud providers faced lots of resistance from businesses, mainly concerning security and cost models, some smaller companies such as Netflix and Facebook saw this compute model as an opportunity to grow swiftly and efficiently. Initially, services offered in the cloud were as simple as

renting hardware, servers, storage, and databases, which enabled businesses to focus on their IT operations. This allowed companies to outsource hardware upgrades and facility and server maintenance. Facility maintenance could include anything from security and cooling to backup power and power supply. The cloud providers would handle upgrading to the latest CPUs or storage technology available on the market. The users could simply migrate to the newest platform, in some cases automatically and in most cases at little or no cost. Over time, the number and variety of cloud-provided services proliferated.

[1]

Even though Amazon Web Services (AWS) initially dominated this fast-growing market, other giant tech companies entered the game, and the competition generated has benefitted the cloud users ever since. Nowadays, most cloud providers offer over 10s of services, far beyond the infrastructure-as-a-service (IaaS) model that initially was the only service. These services can include new scalable database solutions, elastic and durable storage, support for various machine learning and AI problems, and big data frameworks. In addition, multi-regional storage redundancies ensure a durable storage solution immune to natural disasters, cyber-attacks, and hardware/software failures. As the popularity of public clouds grows, the cloud providers will invest more in developing new services to allow the companies to focus on their business problems and development. Since many institutions are discovering the benefits of the cloud, this market is far from saturation.

[1] Image source: GeeksforGeeks

Bibi et al [1] compared the long-term cost of running software on-premise and on the cloud. They concluded that the upfront and annual expenditures of cloud computing are significantly less than on-premise. Thus, in addition to other advantages discussed above, cost-efficiency could be a benefit of cloud computing if utilized appropriately.

C. Hybrid or multi-cloud[2]

Over the past few decades, the companies that invested significantly in their computing infrastructure have acquired large data centers with ample or abundant compute capacity. Although cloud services provide convenience and speed when operating at a large scale, they can be very costly. It is not uncommon for the private sector and governmental agencies to run on both the public and private cloud. Besides the cost benefits of such a hybrid model, it can bring scalability and support for their peak hours of computing demands. Additionally, a hybrid approach could lower the chances of system downtime in cases of emergencies, such as power outages, national disasters, or cyberattacks. The cloud providers have begun to realize the attractiveness and flexibility of the hybrid approach and are adjusting their service models accordingly. An example of this in the analytics space would be the hybrid Kubernetes solution.

Kubernetes is a parallel computing framework that can simultaneously orchestrate tens of thousands of computers to solve big data problems. This solution is practical when the user needs a more significant number of CPUs than what is available on the premises, in which case, the additional CPUs and related hardware can be rented out from a public cloud to create a unified Kubernetes cluster. This elasticity allows the user to expand the size of their private cloud on-demand without requiring the organization to acquire hardware that would sit idle the majority of the time.

[2]Image source: Information Age

2. Programming languages[3]

Programming and querying languages are the primary tools in the data scientist's arsenal. Although many elementary analyses can be performed in Excel, there are severe limitations in advanced analytics. Also, programming languages enable a reproducible and collaborative framework. The most popular programming languages used for data analytics are R and Python, with Python being the most universal. The primary reason for their popularity is the open-source packages and libraries, which provide the data science community with free resources and support. This allows data scientists to reuse the open-source algorithms which have been vetted by hundreds if not thousands of programmers, enabling advanced coding pipelines to be written with minimal effort. Some of the everyday operations in R or Python are data cleaning, summarizing, visualization report generating, statistical analysis, statistical modeling, machine learning, optimization.

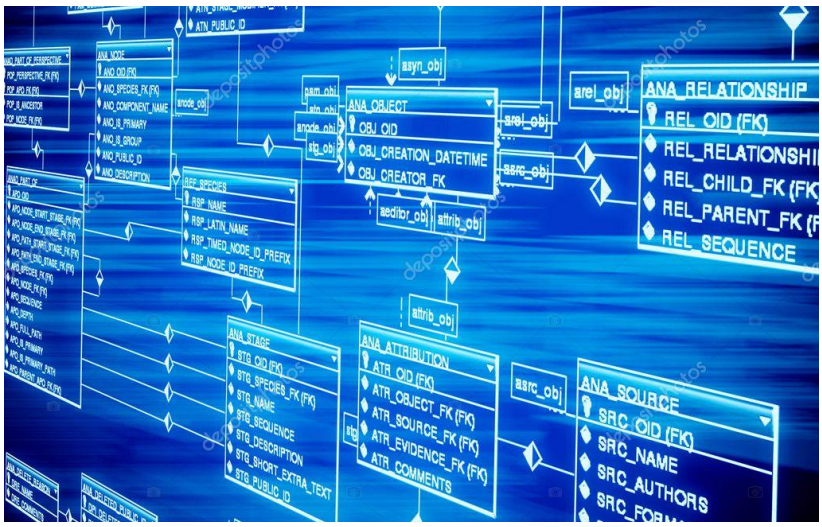
Many open-source libraries are written in more efficient programming languages such as C, C++, and Java. The user can access these libraries from a more friendly environment such as R or Python without any prior knowledge in the native language in which the library was written. These lower-level programming languages are beneficial because they can run code up to 100 times faster while remaining accessible to people with minimal computer science training.

The most common way to store tabular data is using a database. Standard querying languages (SQL) allow interaction with the data and perform operations in the database. These databases are also used as the backend of business dashboards.

[3] Image source: www.simplilearn.com

3. Databases

Generally, databases can be categorized as SQL or NoSQL, according to whether they contain structured or unstructured data, respectively. Most of the data owned by businesses is in an unstructured or semi-structured format. These data types include documents, audio files, and text, such as chatbot conversations. Processing these kinds of unstructured data is labor-intensive or requires sophisticated algorithms. To keep it manageable and accessible, companies strive to maintain data in a structured format whenever possible. SQL databases are the standard tool to store structured data, and their computing capability can also be used to perform elementary to intermediate analysis.



[4]

Several SQL-based database solutions are on the market, some of which are freely accessible as part of an open project, e.g., MySQL and Postgres. Many companies also offer their own database solutions; examples include Microsoft SQL Server and Oracle databases. Although these more traditional databases have minor differences, they share a standard syntax and have similar functionalities.

Modern database solutions come with additional features that make them more attractive for certain business problems. These features include more powerful data processing capabilities, storage elasticity, serverless, fault tolerance, and advanced analytics functionality. Examples of such databases include MySQL, Amazon's Redshift, Google's BigQuery for data warehousing, and Google's BigTable for low latency querying. MySQL Cluster scales horizontally on commodity hardware which makes it very popular. Other proprietary solutions mentioned above operate on a similar concept.

[4] Image source: Depositphotos

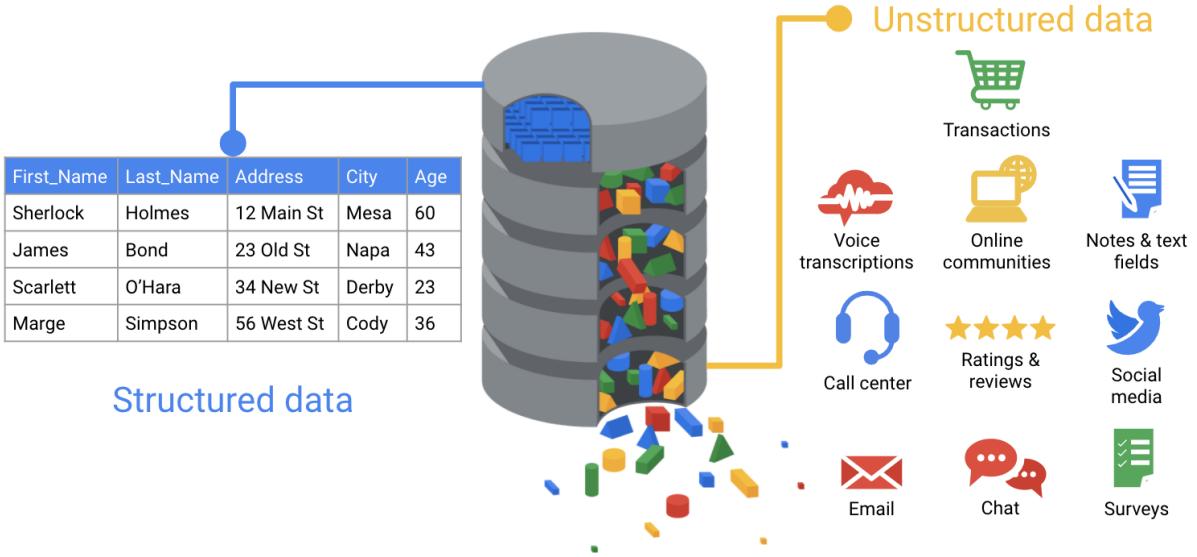
These products are designed for specialized tasks and are highly efficient when delegated to the right job.

Frequently, data scientists use databases in conjunction with Python or other programming languages. The data is stored and most likely pre-processed and aggregated in the database. Then it's transferred over to a Python environment for visualization, reporting, statistical modeling, and advanced analytics. The output of Python's analysis can be written back to a database table. This allows a database to act as a central repository for the raw and processed data, which different teams can use. Connecting Python to a database makes it possible to obtain an aggregated portion or a subset of the data rather than the entire dataset, which may be too big for a Python environment to handle.

4. Big Data Platforms

Loosely speaking, big data refers to datasets that are too large to be handled by one computer. Depending on the task at hand, the solution complexity can vary significantly. For example, if the only challenge is storing and archiving the data, then a bigger hard drive can solve the issue with little effort, assuming the data is smaller than the biggest hard drive on the market. However, suppose this data needs to be processed. In that case, there are more complicated challenges to overcome, such as limited memory of the computer and the hard limits on increasing the memory of one machine. In those

cases, we would need to distribute the data into a cluster of computers that consists of many machines orchestrated by a central master node. Depending on the task and whether we want to manage this cluster or use a managed service, there are multiple options that companies can choose to solve their big data problems. Below we will discuss an array of big data solutions for structured and unstructured datasets available to the users.



5

Structured Data

Although it may be unimaginable for some to have a dataset that cannot be opened by Excel due to the limited computer memory, for most companies, it's the nature of their business. These live datasets are too big and keep growing every day with every transaction and event. This is the reason why we have many database solutions, which were discussed above. Databases are powerful and some have been built with big data in mind, e.g., MySQL, Amazon Redshift, and Google BigQuery. Some, on the other hand, can get obsolete as the data grows beyond a threshold.

SQL Databases

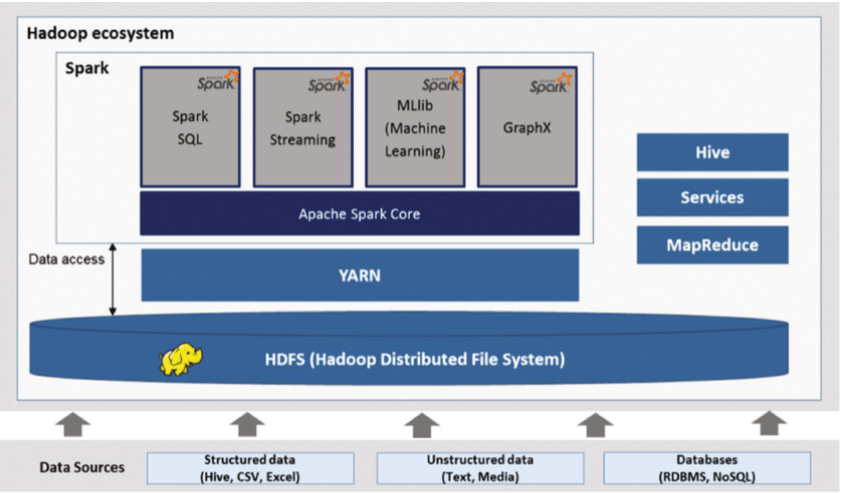
A scalable and elastic database could be the solution if the task at hand is record-keeping and basic analytics. For an open-source solution like MySQL, one can horizontally scale up the size of the cluster, meaning that additional computers can be added to handle more data. This is great as the cost of such a database increases linearly, unlike vertical scaling, which is far more expensive and has hard limits. The

[5] Image source: Google Cloud Computing Foundations for Higher Ed

process is similar for the proprietary databases such as BigQuery, but auto-scaling is automated, and the details are not publicly available.

Hadoop and Spark Ecosystems[6]

Database engines cannot complete certain tasks. Examples are advanced analytics and machine learning. In these cases, we would need to leverage architectures that can support advanced algorithms. In the early 2000s, the de facto solution for these scenarios became Apache Hadoop. Hadoop is an open-source software platform for distributed storage and distributed processing of very large datasets on computer clusters built from commodity hardware. Hadoop File System (HDFS) would distribute the files across a cluster that can be scaled up horizontally. Hadoop MapReduce allows the user to process this data in a parallel and scalable fashion. HDFS creates multiple copies of each file and distributes them across the cluster to ensure fault-tolerance. Because of this architecture design, we won't lose any data n the event of a node failure as there are still exact replicas of the file available in the cluster. Although some companies are still using HDFS and MapReduce, there are other solutions on the market that can perform big data analytics far more efficiently; in some cases, such as Apache Spark, it could be up to 100x faster than the MapReduce framework.



Apache Spark is a unified computing engine and a set of libraries for parallel data processing on computer clusters. The main advantage of the Spark framework is the in-memory processing capability with additional support for advanced analytics, machine learning, and real-time data streaming. Spark has several different APIs that allow users to interact with it via Java, Python, R, and even SQL. The Spark library also includes the Spark MILib that offers the most common machine learning algorithms that can run at scale.

[6] Image source: A Comparison of Predictive Analytics Solutions on Hadoop [2]

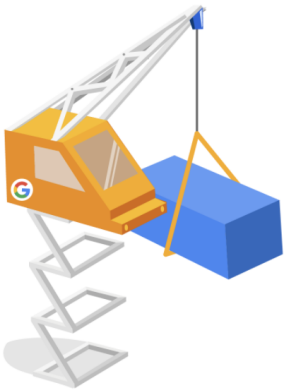
Kubernetes

One of the other big data scenarios is where we want to process a large amount of data, but the dataset consists of small, independent chunks of data. As an example, take a company like FICO. FICO focuses on credit monitoring and credit scoring. Typically, they update a customer's credit score weekly. The challenge is that they might need to do this task for 10's of millions of customers every week. This involves pulling data from different sources and reviewing the last seven years of financial activities to provide a credit score used by the lenders. If performed in a series of tasks, it could take a very long time to finish; perhaps not enough time until next week's update should be released. This is an embarrassingly parallel task, meaning that since individual clients are independent of each other, the jobs can be run in parallel without needing any information from other jobs. While a platform like Spark can be utilized for this job, it might be overkill. All that needs to be done is to distribute these tasks to different computers on a cluster and let them run. The speed can linearly increase compared to the size of the cluster.

Kubernetes is an open-source container orchestration system that can automate applications and scale the cluster up or down depending on the load. Kubernetes duplicates the software and distributes it throughout the cluster using containers. Containers are isolated units of computing that can run individually on the same machine, allowing them to run many operations simultaneously using a single operating system. Imagine a cluster with 10,000 machines, each containing 20 containers: we will be able to run 200,000 jobs in parallel. This can significantly speed up the processing time. Being able to scale up or down the cluster, particularly in a cloud environment, can reduce the compute time or save money when excess resources are not needed. When the need arises, the cluster can spin up a million machines to perform the task, and when there is nothing scheduled, it can go down to zero. We end up paying the same amount of money for the job at hand, but in a shorter time, which is highly desirable. Many governmental and private entities use this approach in a hybrid scenario for their peak hours, leveraging a large pool of resources in the public cloud only when extra resources are needed.

Docker is the tool that puts the application and everything it needs in the container. Once the application is in a container, it can be moved anywhere that will run Docker containers — any laptop, server, or cloud provider. Kubernetes is the orchestration tool for managing a cluster of Docker containers as a single system. It can be run in the

cloud and on-premises environments. It was initially designed by Google and released in 2014 for the first time. The Cloud Native Computing Foundation now maintains it.



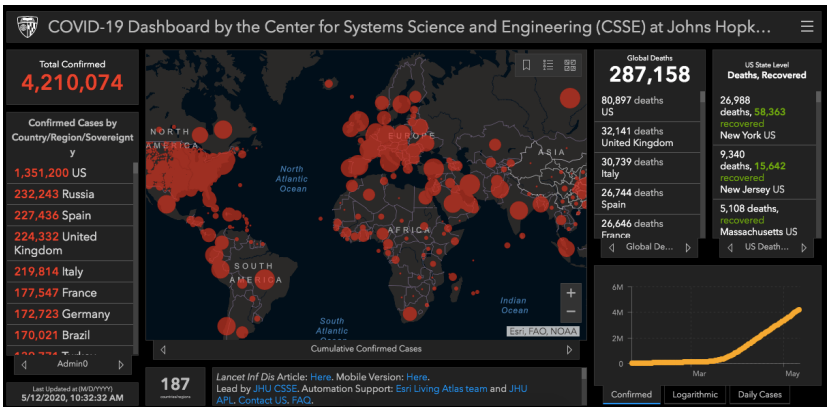
Unstructured Data

Most of the data kept by companies is in an “unstructured” format. Unstructured refers to the fact that each data point might have a slightly different shape than the other ones and is not organized by rows and columns. Examples are PDF documents, images, videos, and audio files. Typically, these files are kept in an archive and won’t be used for analysis since working with them requires a highly specialized workforce and is computationally more expensive than working with structured data. There are many different solutions to store these data types, including open-source HDFS or cloud services such as Google Cloud Storage or Amazon S3. These solutions are highly scalable and fault-tolerant, but only the cloud solutions offer further benefits such as geo-redundancy and high durability.

If, in addition to storage, fast data retrieval is also essential, then a NoSQL database can be utilized. NoSQL stands for “Not only SQL” which emphasizes that they may support SQL-like query languages or sit alongside SQL databases in polyglot-persistent architectures. Examples of NoSQL databases are MongoDB and Elasticsearch for document storage, Cassandra, HBase, Amazon DynamoDB, Google Bigtable for wide-column storage, and Redis for key-value storage.

As illustrated above, different solutions should be considered for different types of big data problems. Finding the right solution can simplify future tasks, reduce cost, and improve the chances of success.

5. Business intelligence dashboards[7]



BI dashboards are interactive visualization tools that present the key performance indicators (KPIs), raw or processed data, and the corresponding charts related to the business. These dashboards typically combine data from different sources and present them in a visual format. They allow business users to make data-driven decisions by consolidating and analyzing data. Dashboards are a vital component of the BI software and are the front end of the analytics pipeline. The data to be visualized in a dashboard has to go through many different steps, including cleaning, joining, aggregation, and modeling. Typically, many of the steps mentioned above are involved before the data arrives at a dashboard to be presented to the stakeholders. For instance, most advanced dashboards use a robust database as their back-end engine to speed up the calculations and data retrieval. Tableau and Power BI are among the most popular dashboards.

6. Software development tools

To achieve an end-to-end analytics pipeline, several different pieces of software development are involved. Commonly, a collection of open-source and/or proprietary software is developed to address the requirements of a project. Developers make use of many tools to achieve this, some of which were mentioned earlier. Version control tools are among the most crucial tools that are used for collaboration and code sharing. This allows a team to distribute tasks, perform code reviews, and collaborate on the same code. Among the widely used tools are GitHub and BitBucket, both of which use the open-source software Git.

Version control tools allow the analysts to access the same version of the code and databases synchronize the data that they use. These two components are essential but not entirely enough to grant a seamless collaboration. The operating system,

[7] Image from the John Hopkins covid tracker

environmental variables, and versions of packages used in the application also play an indispensable role. If not compatible, the code can be vulnerable to unpredictable environmental changes. To solve this problem, software developers use containerization solutions. As briefly mentioned in the big data section, containers simulate the exact same execution environment regardless of the operating system and other changeable parameters that can make the pipeline volatile. By sharing the same code, data, and execution environment, it is guaranteed that everyone will get to the same conclusion by running the code, assuming there are no stochastic elements involved. This is particularly important in a collaborative setting where reproducible code is the key to success.

Further Reading

Articles:

- Business Application Acquisition: On-Premise or SaaS-Based Solutions?
- Different Types of Cloud Service Models
- SQL vs. NoSQL Databases: What's the Difference?
- Database — Introduction (Part 1 & Part 2)
- A beginner's guide to the basics of what cloud computing is about
- How To Build An Effective Business Intelligence Dashboard

Books:

- Google BigQuery: The Definitive Guide
- Data Science on the Google Cloud Platform
- The Analytics Edge
- Data Science for Business
- The Big Book of Dashboards

References

1. S. Bibi, D. Katsaros and P. Bozanis, "Business Application Acquisition: On-Premise or SaaS-Based Solutions?," in IEEE Software, vol. 29, no. 3, pp. 86-93, May-June 2012, doi: 10.1109/MS.2011.119.
2. R. Norousi et al, "A Comparison of Predictive Analytics Solutions on Hadoop", Conference: International Conference on Intelligent Decision Technologies
3. Dresner Advisory Services' 2018 Big Data Analytics Market Study