Emerging capabilities to process vast quantities of data are bringing about changes in technology and business landscapes. This article examines the drivers, the new landscape, and the opportunities available to analytics with Apache Hadoop.

**Market drivers**

In recent years, both the web and the enterprise have seen an explosion of data. There is a variety of contributing factors to this phenomenon, such as the commoditization of inexpensive terabyte-scale storage hardware, enterprise data having approached critical mass over time, and standards allowing the easy provision and exchange of information.

From an enterprise standpoint, all of this information has been getting increasingly difficult to store in traditional relational databases and even data warehouses. These issues are asking some difficult questions of practices that have been around for years. For instance: How does one query a table with a billion rows? How can one run a query across all of the logs on all of the servers in a data center? Further compounding the issue is that a lot of the information needing to be processed is either unstructured or semi-structured text, which is difficult to query.

When data exists in this quantity, one of the processing limitations is that it takes a significant amount of time to move the data. Apache Hadoop has emerged to address these concerns with its unique approach of moving the work to the data and not the other way around. Hadoop is a cluster technology comprising two separate but integrated runtimes: the Hadoop Distributed File System (HDFS), which provides redundant storage of data; and map/reduce, which allows user-submitted jobs to run in parallel, processing the data stored in the HDFS. Although Hadoop is not well suited to every scenario, it provides clear performance benefits. In using Hadoop, the community discovered that it not only allowed the processing of data at scale but opened the door for the potential of all kinds of interesting analytics.

With Hadoop, we can linearly scale clusters running on commodity hardware to incorporate larger and richer datasets. These datasets are providing new perspectives, first by running analytics over heterogeneous sources of data that one could not previously incorporate, then running analytics over that same data at scale. This results in somewhat of a paradigm shift, described by Flip Kromer, a co-founder of InfoChimps: "The web is evolving from a place where one could find out something about everything, to a place where one can find out everything about something."
Kromer goes on to pose the scenario that, over time, baseball fans have captured detailed results (player details, score, locations of ballparks) for every baseball game in the past 100 years. If one were to join that dataset with the shared location values for all the weather stations during the same time period, one could statistically predict how a 38-year-old pitcher would perform at Wrigley Field when the temperature is over 95 degrees.

The Big Data ecosystem

It is important to point out that the Big Data space is still relatively new and that there are some technological barriers to taking advantage of these opportunities. As mentioned, data is processed in Hadoop in the form of "jobs," which are written in the Java™ programming language using a paradigm known as map/reduce. While there are contributions to Hadoop that allow the use of other languages, it is still a non-trivial process to properly understand how to analyze business problems and decompose them into solutions that run as map/reduce jobs.

To truly take advantage of the opportunities around Hadoop, there needs to be a wide array of enabling technologies that move Hadoop out of the purview of the developer and make it more approachable to a wider audience.

Figure 1. Overview of the Big Data ecosystem

An ecosystem has emerged to provide much-needed tooling and support around Hadoop. Each component works with the others to provide a swathe of approaches, as shown below, to achieve most user scenarios.

Load tooling

In order to use Hadoop to analyze your data, you need to get the data into HDFS. To do this, you need load tooling. Hadoop itself provides the ability to copy files from the file system into the HDFS and vice-versa. For more complex scenarios, you can take advantage of the likes of Sqoop (see Resources), a SQL-to-HDFS database import tool. Another form of load tooling would be a web crawler, such as Apache Nutch, that crawls specific websites and stores the web pages on HDFS so the web content is available for any analytics you may wish to perform.
Real-time data is another potential source of information. You can use technologies like Twitter4J to connect to the Twitter Streaming API and persist the tweets directly onto the HDFS in JSON format.

Typical Big Data analytics use cases usually involve querying a variety of datasets together. The datasets often come from different sources, usually a mixture of data already available within the enterprise (internal) and data obtained from the web (external). An example of internal information might be log files from a data center, and external information might be several crawled websites or a dataset downloaded from a data catalog.

Data catalogs
Data catalogs fulfill a much-needed capability for the user to search for datasets. Until you've tried looking for them, you won't realize just how hard it is to find large datasets, especially ones that fit the particular analytic scenarios you're trying to run. Usually, users are forced to crawl the web or mine social-media sites to build their own. It saves an enormous amount of time if you can simply find and download an existing structured dataset on a topic of interest. Companies like InfoChimps provide a consolidated catalogs where you can locate specific datasets by category or search. Another example of a data catalog is the Amazon Public Data Sets.

Analytics tooling
If you were to use Hadoop by itself to run analytics over the stored data on the HDFS, it would typically require the skill of a developer to write the jobs in the Java language using the Hadoop map/reduce API. For those working directly with the APIs, you can use tools like KarmaSphere Studio in Eclipse (see Resources) to make use of the productivity gains from a Hadoop-specific IDE. There are alternatives for using other languages with Hadoop Streaming and Hadoop Pipes, but these still require the skill of a developer. This has created an opportunity for the creation of less-complex approaches to defining and running map/reduce jobs.

Those familiar with shell scripting and UNIX® Pipes might consider looking at WuKong (see Resources), which allows you to construct and run jobs using Ruby and shell scripts. Apache Pig and Hive (see Resources) are two technologies that would be of interest to a data analyst, as they provide a SQL-like query interface where the user can express how to construct and run the analytics for a given job through higher-order languages. Another approach that is more targeted to the business analyst is IBM® BigSheets (see Resources), which provides a browser-based visual metaphor akin to a spreadsheet for defining, running, and visualizing analytics jobs.

All approaches typically make use of extensions, often called user-defined functions or macros, that take the input data and inject a measure of structure into it (semantically or explicitly) so the information can be addressed and queried in a manner similar to traditional analytical methods. Analytics tooling is joined at the hip with export tooling, as the latter is intrinsic in actually being able to do something useful with your data once you've run your analytics.

Export tooling
When you ask a Big Data question (your analytics), you tend to get a Big Data answer (the resultant dataset). Often, the answer is so big that it is not human-readable or comprehensible.
If this is the case, providing the results as a visualization might be one solution. For instance, a tag cloud can filter a large set of results to allow a human to instantly recognize certain areas of value in the data. Another approach might be to export the dataset into a particular format, such as JSON, CSV, TSV, or ATOM, so it can be consumed by an application for use. Interesting visualizations are fairly common, but they typically are not pluggable into the existing tooling around Hadoop. This is an emerging space in which we will likely be seeing some innovation over the coming months.

**Apache Hadoop**

Apache Hadoop is the nucleus of the ecosystem. This is where all the data resides. A constraint unique to this ecosystem is the fact that Big Data likes to be at rest. It can add serious latency to any computation efforts to move large amounts of data around and that is why map/reduce is so efficient in that it moves the work to the data. Due to its ability to scale horizontally and linearly, Hadoop is also a viable option in the cloud as one can provision a Hadoop cluster, copy the data up, run the job, retrieve the output, and decommission the cluster once the work is complete. This has significant cost savings over purchasing and maintaining the hardware for jobs run intermittently.

**BigSheets**

For the non-programmer interested in analytics, InfoSphere BigInsights is a new IBM product portfolio that contains a technology preview called BigSheets (see Resources). BigSheets provides a compelling visual interface to gather, analyze, and explore data at scale. BigSheets is a fairly all-encompassing tool in that it provides easily configurable load, analytics, and export tooling on top of Apache Hadoop.

**Conclusion**

We've looked at the current deluge of data and how the open source community is addressing these issues with the Apache Hadoop project. We've also examined the exciting opportunities to derive new insights with Big Data and some of the open source and proprietary tooling within the ecosystem that has sprung up around Apache Hadoop.

For those wanting a more granular understanding of Hadoop, don't miss "Distributed computing with Linux and Hadoop" (see Resources) and try out the WordCount example (the Hello World equivalent for map/reduce) described in detail on the Apache Hadoop project Wiki.

For those wanting a simpler on-ramp into the analytics, try out Apache Pig (see Resources) and run through the tutorials on the project wiki.