Hadoop's Limitations for Big Data Analytics

Make sure you have all the tools to do the job: Solving big data analytic challenges requires a complete ecosystem.

Hadoop's Limitations for Big Data Analytics

For more information, visit
ParAccel.com/technology
or contact us at 866.903.0335.

Executive Summary

The era of 'big data' represents new challenges to businesses. Incoming data volumes are exploding in complexity, variety, speed and volume, while legacy tools have not kept pace. In recent years, a new tool – Apache Hadoop – has appeared on the scene. And while it solves some big data problems, it is not magic. In order to act effectively on big data, businesses must be able to assimilate data quickly, but also must be able to explore this data for value, allowing analysts to ask and iterate their business questions quickly. Hadoop – purpose built to facilitate certain forms of batch-oriented distributed data processing – lends itself readily to the assimilation process. But it was built on fundamentals which severely limit its ability to act as an analytic database.

With the rise of big data has come the rise of the analytic database platform. Even five years ago, a company could leverage a DBMS such as Oracle for a data warehouse. However, Oracle was built in a time when databases rarely exceeded a few gigabytes in size. Along with other legacy DBMSs, it cannot perform at the scale now required. Enter the analytic platform. The analytic platform allows analysts to use their existing tools and skillsets to ask new questions of big data quickly, easily, and at scales unseen previously.

The de facto best practice infrastructure for big data today often consists of a processing infrastructure of systems such as Hadoop to acquire and archive the data, and an analytic platform to enable the highly iterative analysis process. But because Hadoop is still relatively new, there is a great deal of confusion about its strengths and weaknesses. This paper will discuss those topics, and concludes with guidance on how to build the complete ecosystem for big data analytics.

Hadoop's Limitations for Big Data Analytics

What is Hadoop?

If you'd have asked that question in 2010, you probably would've gotten nothing but blank stares! But the fascination level with Hadoop has skyrocketed in the last few years such that today there are an array of products out there which call themselves "Hadoop", many of which do different things, but all of which use the same basic platform. So let's start by taking a moment to clarify what Hadoop really is…and isn't.

In the most literal definition, Hadoop is a collection of open-source projects originated by Doug Cutting in 2006 to apply the Google MapReduce programming framework across a distributed system. At its core are two components: the Hadoop Distributed File System (HDFS); and the MapReduce programming and job management framework. Because Hadoop provided an easily obtained framework for distributed processing, a number of open-source projects quickly emerged which leveraged this to solve very specific problems.

Here are a few of the better known examples:

Project Name	Purpose
Hive	Puts a partial SQL interface in front of Hadoop
Pig	A scripting language on top of Java for MapReduce programming
HBase	Applies a partial columnar scheme on top of Hadoop
Mahout	A set of data mining algorithms
HCatalog	A metadata layer to simplify access to data stored in Hadoop
Impala	A database-like SQL layer on top of Hadoop

Each of these was developed to address a gap in Hadoop: Hive, Impala and HBase to make Hadoop look something like a database; Pig to lower the cost of developing MapReduce programs; and, Mahout to allow programmers to avoid re-inventing statistical algorithms every time they author a new MapReduce program.

If you have trouble relating that to your business needs, you are not alone. While Hadoop has been very exciting to hands-on data technicians, it's been something of a mystery to the business world. Today it has high name recognition, but most people aren't clear on what it is actually used for.

If you ask a room full of people what they use Hadoop for today, you'll very likely get these responses:

> "ETL"
>
> "Archiving"
>
> "Basic queries where we don't care about latency"
>
> "We've tried it for analytics, but haven't gotten very far"

The blessing – and the curse – of Hadoop is that it is a collection of projects, developed by different people at different points in time, to solve tactical problems. The core of Hadoop makes this possible – a distributed file system and programming framework to execute distributed MapReduce programs. The challenge is that these open-source technologies evolve organically rather than being purposefully designed. The overall capabilities of Hadoop continue to be shaped by this evolving core. Furthermore, critics of MapReduce say it is not accepted to be a solution to the kinds of problems for which analytic platforms are purpose-built.

Before going on, let's review the things which make Hadoop appealing:

1) The software is "free" (If we don't count the costs of hardware – which Hadoop consumes generously or of Hadoop developers, who today command salaries in excess of $200k.)
2) It is a technology platform for distributed MapReduce programs – which can dramatically speed up certain types of data processing operations.
3) It is a distributed filesystem that lends itself to low cost storage.

The resulting economics of a Hadoop infrastructure make it seem like an attractive solution in situations which are known to have high costs today – namely managing and analyzing big data to get real value. However, there are some serious limitations in its analytic functionality, which may not be immediately obvious.

Hadoop's Limitations for Big Data Analytics

First, let's back up and consider why big data is such a big deal. As the world becomes more instrumented, the volumes of data available to the enterprise are growing by orders of magnitude. Those data volumes often hold critical insight for organizations – if only it can be efficiently analyzed, which is no easy task. Big data has truly changed the requirements for data management and analysis technologies. Five or ten years ago, the accepted model for data analysis and reporting was to bring the data in to an ETL platform, prepare it, then load it to a data warehouse. This was a daily batch process, all very orderly and planned. Changes and projects were often predictable and known weeks to months in advance. Data warehouse vendors responded to this with products designed for this world – such as the Netezza, Teradata, and Oracle approach, primarily based on appliances.

This dramatic growth in incoming big data has created an economic impedance mismatch in infrastructure: there is 10x, 100x, or even 1000x the volume of data flowing in daily, which can (and does) change constantly. Firms know that the new, incoming data is not without value, but until they analyze it sufficiently they can't make a business case to extend their proprietary ETL infrastructure or purchase another expensive appliance. And because they know the data may have value, they don't want to dispose of it either.

This is where Hadoop has caught on. Businesses are discovering that they can deploy a Hadoop cluster to ingest their data and perform ETL operations. And because some open-source Hadoop tools offer simple query functionality, organizations can do some assessment of the value of the data. Further, once the data is loaded, it can simply be retained, solving the archival problem.

If you've researched Hadoop, and maybe attended a few meetups, you've probably heard that it can also be used as a database or for analytics. There are users who believe that once Hadoop is in place, a data warehouse isn't required for analytics. But the database-like qualities of Hadoop are not a replacement for a true analytic platform. The fundamentals of Hadoop were not designed to facilitate highly interactive analytics. This is why the Hadoop community today is using Hadoop largely for ETL and archival.

Hadoop's Gaps in Analytic Functionality

Limited database functionality is not the only reason Hadoop hasn't taken over the world. First, open-source software is notorious for being highly variable in quality, with some of it being just plain unusable. This is because the economics of open-source development provide no incentive for software suitability or quality. Instead open-source is frequently an avenue for new software engineers to try their hands at software authoring. And while there is a small community of experienced developers who contribute to open-source code projects, they are incented to do so only by goodwill. In fact, firms which "sell" open-source solutions often base their business model on providing implementation services, so they have few incentives to make the software easier to setup and use. Finally, due to the distributed nature of open-source development, quality assurance is difficult or impossible. The end-result is that only some needs are met, and when they are, it is with a solution of unpredictable usability and quality.

Specifically with regard to Hadoop, it was conceived to solve a very specific problem: enabling distributed MapReduce processing on arbitrary sized clusters of low-cost hardware. To enable this, the Hadoop contributors built a distributed filesystem – the Hadoop Distributed File System (HDFS); and a set of components to execute distributed MapReduce java programs.

For some insight into the MapReduce programming framework, have a look here. But while it is very good at certain forms of distributed data processing, it is not well suited to be a platform for highly interactive analytics. For commentary and background, see here.

The fundamentals of Hadoop were not designed to facilitate highly interactive analytics.

Hadoop’s Limitations for Big Data Analytics

Another consideration is that HDFS was purpose-designed with one thing in mind – to speed the processing of various web documents, and to apply the MapReduce framework to this processing. To this end, it is first and foremost a filesystem. This means that it does not require a schema. And while it designs for redundancy, it also does not constrain itself – after all, why bother? It was purpose built to operate on clusters of arbitrary size, so there was no reason to design efficient storage into the mix. The downsides of HDFS come from its strengths. It has no optimizer – so your developers will need to be sure to optimize their own data flow. Because it was built to be a filesystem, there is no notion of transaction consistency or recovery checkpoints. This means that the answer you get from a Hadoop cluster may or may not be 100% accurate, depending on the nature of the job.

The answer you get from a Hadoop cluster may or may not be 100% accurate, depending on the nature of the job.

With regard to using Hadoop to solve business problems, a recent experience by the ParAccel team illustrates the experience of the new Hadoop user (as told by Rick Glick, our Vice President of Customer & Partner Development):

> A group of us at ParAccel enjoy brewing our own beer. One weekend, not too long ago, we were preparing to make a batch. We’d bought a carboy, four sizes of tubing, a wort chiller, bottles and caps, a capper, four types of malt, three kinds of hops, some additives to insure clarity in the beer and good foam (key to a good beer experience!), and a half dozen other items.
>
> After two hours of connecting, sorting, locating, relocating, two more trips to the store, white boarding, and logistics planning, we had everything set up. The kitchen looked like Frankenstein’s laboratory with tubing everywhere and good smells in the air. One of us jokingly suggested that they felt like they’d just set up Hadoop.

It’s a bit of humor intended to illustrate a point – that if you plan to develop with Hadoop you’d better roll up your sleeves and learn Java, MPP architectures, and distributed data algorithms. And in the end, you may find that despite your efforts, it still doesn’t quite deliver what your business needs. While Hadoop is a powerful framework for certain types of distributed problems, it requires specialized expertise to use it effectively. If you’re more interested in the end result, you may be better served buying purpose built software, rather than doing it yourself.

That said, the pluses of Hadoop are sufficiently interesting such that many businesses today at least are experimenting with it. In some areas they’re putting it to use – for ETL and data archival. These are relatively simple uses for which they can find staff, and for which good Hadoop functionality already exists.

There are a number of startups today built with the goal of extending Hadoop to make it more enterprise-friendly. It’s beyond the scope of this paper to list them, and it may not even be possible – it seems that every day a new Hadoop startup is announced. They’re all trying to solve different problems on the platform. At the end of the day, however, they all have to live with the fundamental limitations of the platform.

Hadoop's Limitations for Big Data Analytics

So what, specifically, are the limitations of a Hadoop platform?

Hadoop limitations for analytics:

- **Multiple copies of already big data:** Because HDFS was built without the notion of efficiency, it results in multiple copies of the data. At a minimum, there are generally three copies of the data. And because of the need for data locality in maintaining performance, we very often see six copies of the data required…and that's for data that's already "big" by definition.
- **Very limited SQL support:** There are open source components which attempt to set up Hadoop as a queryable data warehouse, but these offer very limited SQL support. Typically they lack such basic SQL functions such as subqueries, 'group by' analytics, etc.
- **Inefficient execution:** HDFS has no notion of a query optimizer, so cannot pick an efficient cost-based plan for execution. Because of this, Hadoop clusters are generally significantly larger than would be required for a similar database.
- **Challenging framework:** The MapReduce framework is notoriously difficult to leverage for more than simple transformational logic. There are open source components which attempt to simplify this, but they also use proprietary languages.
- **Lack of required skills:** The intriguing data mining libraries which are part of the Hadoop project – Mahout – are inconsistently implemented, and in any event require both knowledge of the algorithms themselves as well as the skills for distributed MapReduce development. Try finding *that* combination of skills!

At the time this paper was being written, Cloudera announced a new product for use with Hadoop named Impala. It is being positioned as a SQL-like engine which bypasses the Hadoop MapReduce framework and allows business intelligence (BI) tools to execute queries against data in HDFS and HBase. On the surface, it looks like an incremental step forward over Hive. Hive relies on MapReduce to execute queries, which degrades query performance significantly. Impala, on the other hand, deploys a separate set of processes which bypass MapReduce to read directly from HDFS and HBase data. Early commentary on Impala hints that in the future, it will add a columnar storage engine, cost-based optimizer and other distinctly database-like features. But based on DBMS development cycles, this will be a long way out.

Since Cloudera is a leading vendor for Hadoop, the announcement carries implications for the larger Hadoop world. First, this move implies that MapReduce does not make sense as an engine for querying. And secondly, it's clear that in order to do analytics on big data, it's natural and efficient to use SQL to query a column-oriented MPP database with columnar-based storage, a cost-based optimizer and other database-like functions. There is no reason to reinvent the database on Hadoop, especially when platforms already exist that can be an extension of Hadoop for analytics. For example, ParAccel has spent years developing just that – a purpose-built, column oriented MPP database for analytics, with full SQL support and in-database libraries of sophisticated analytic functions.

While new market entrants like Impala may offer simple query functionality, it will be a long time before they are on par with purpose-built analytic platforms. Until then – lightweight analytic tools for Hadoop are best used in complement with a fully-baked analytic platform.

Now What? Designing an Efficient Big Data Analytics Architecture

This paper has been focused on Hadoop so far, but Hadoop is just a tool for a business need. The business need today is to manage big data, and deliver rich analytics at scale, with agility and a cost equation a business can afford. Traditional infrastructures were built for a world with orders of magnitude less data. As discussed, while Hadoop is useful in ingesting and preparing big data, it does not meet the need of analytics. As data grows in size and scope, the business opportunities to differentiate with big data analytics grow. Companies know this, and are looking to analytic platforms as complements to Hadoop's scalability and data processing capabilities. Analytic platforms, such as ParAccel's, are purpose built to enable the preparation and execution of rich big data analyses with tools that are familiar to analysts – typically SQL, the lingua franca of analytics.

Real-world deployments of a modern infrastructure today rely on cooperative analytic processing, with acquisition platforms (such as Hadoop) and analytic platforms, such as ParAccel's, working together to deliver big data analytics capabilities. These architectures often consist of Hadoop clusters for data acquisition and archival, with an analytic platform in front of the Hadoop cluster. This allows analysts to interact with big data quickly and easily, without the need to author Java programs or wait on a Hadoop batch process.

This cooperative processing relies on integration of Hadoop and analytic platforms. But data integration between Hadoop and the other databases has historically been slow and difficult due to the parallel nature of Hadoop. Many so-called Hadoop connectors are slow and inefficient.

Acquisition platforms (such as Hadoop) and analytic platforms work together to deliver big data analytics capabilities.

That is why ParAccel has built an On Demand Integration module specifically for bi-directional data interchange between Hadoop and the ParAccel analytic platform. This module automatically determines optimal parallelism between the Hadoop cluster and the ParAccel platform for optimal throughput, and greatly simplifies the process of moving the data. Furthermore, ParAccel also offers similar On Demand Integration modules for Teradata, Oracle and SQL Server, since most businesses use more than just Hadoop. ParAccel's analytic platform fits seamlessly into IT environments, leveraging and extending the analytic capabilities of the existing technologies.

Hadoop’s Limitations for Big Data Analytics

ParAccel and Hadoop Together

More organizations are interested in using Hadoop for low-cost processing and storage, and ParAccel for analytics across Hadoop data and other corporate data sources. This approach leverages the strengths of both technologies.

ParAccel Strengths

- In-database analytic functions for time series, clustering, linear and logistic regression, matrix operations, financial analysis, spatial analysis, and more
- Full ANSI SQL support
- Parallel Hadoop integration
- Bi-directional Hadoop integration
- Platform openness to allow for interaction with other systems

Hadoop Strengths

- Ingest big data
- Store data inexpensively
- Low software cost
- Published APIs

Implementing the ParAccel-Hadoop solution is simple, due to the prebuilt On Demand Integration module for Hadoop. Spin up both clusters, run the On Demand Integration module to connect them and start running complex, SQL-based analytics on big data immediately.

Once you have your clusters up, you’re ready to go. Leverage ParAccel’s extensive library of analytic, statistical and data mining functions, or create, store and share your own custom algorithms. With a platform purpose-built for analytics and a high-performance Hadoop integration module, ParAccel will help you get real value from big data analytics. ParAccel makes it easier for your analysts to interact with big data, enabling your organization to accelerate, innovate and complete.

Sizing ParAccel and Hadoop Clusters

Based on experience and feedback from the marketplace, at ParAccel we’ve got a pretty good idea of how to estimate the ratio between Hadoop and ParAccel clusters. The challenge in doing this is that your Hadoop workload may not be the same as your analytic workload! So please consider this as a guideline only. That said, in tests, we’ve observed repeatedly that due to its optimizer and storage scheme, ParAccel requires fewer nodes than a given Hadoop cluster storing the same data and doing the same work.

The real world ratio is typically seven Hadoop nodes to one ParAccel node. If your analytic workload is simple, this ratio may go as high as 10:1. If the workload is more complex, it may go as low as 5:1.

Next Steps

Watch or share a short whiteboard video on how ParAccel uniquely enables big data analytics

Read more about ParAccel’s On Demand Integration Module for Hadoop or learn more about Cooperative Analytic Processing Architectures

Learn Colin White’s criteria for analytic platforms for big data

Read how ParAccel uniquely addresses all expert criteria for analytic platforms

For more information, visit www.ParAccel.com, or call your local sales director.

ParAccel can also be reached at 866.903.0335.