Distributed Systems Architecture

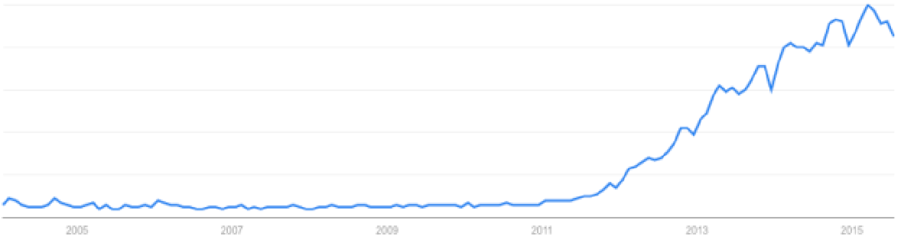
brought to you by Alexey Grishchenko

Hadoop vs MPP

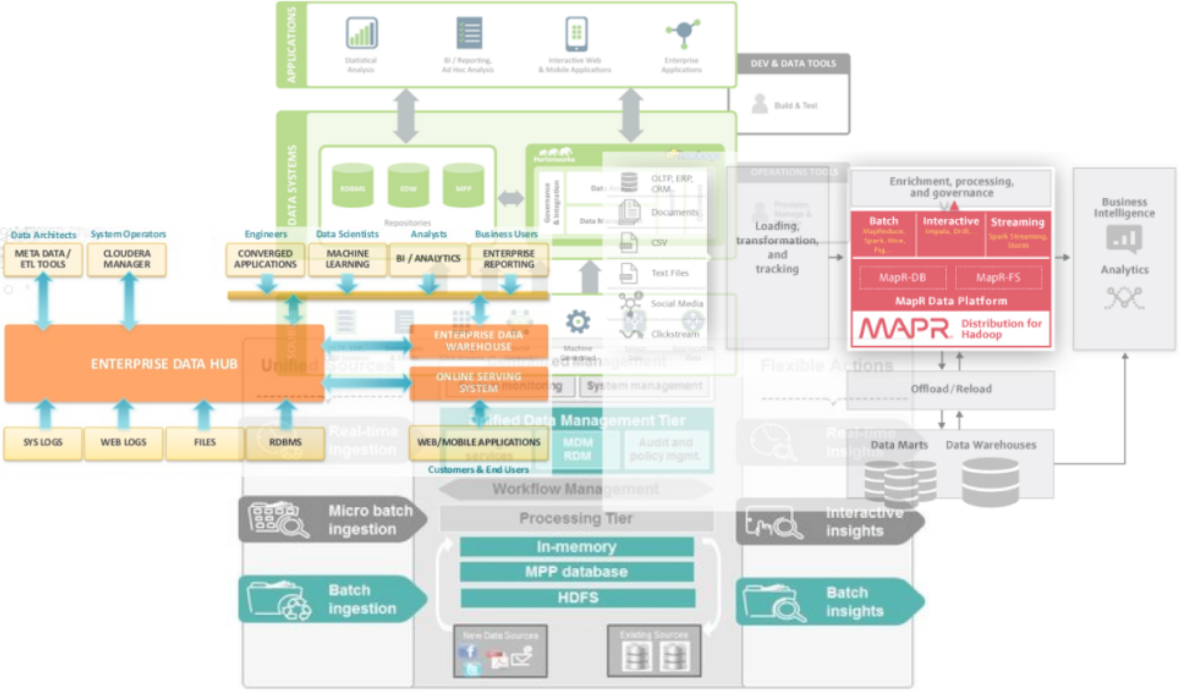
Over the latest time I've heard many discussions on this topic. Also this is a very popular question asked by the customers with not much experience in the field of "big data". In fact, I dislike this buzzword for ambiguity, but this is what the customers are usually coming to us with, so I got to use it.

If we take a look 5 years back, that was the time when Hadoop was not an option for most of the companies, especially for the enterprises that ask for stable and mature platforms. At that very moment the choice was very simple: when your analytical database grow beyond 5-7 terabytes in size you just initiate an MPP migration project and move to one of the proven enterprise MPP solutions. No one heard about the "unstructured" data – if you got to analyze logs just parse them with Perl/Python/Java/C++ and load into you analytical DBMS. And no one heard about high velocity data – simply use traditional OLTP RDBMS for frequent updates and chunk them for insertion into the analytical DWH.

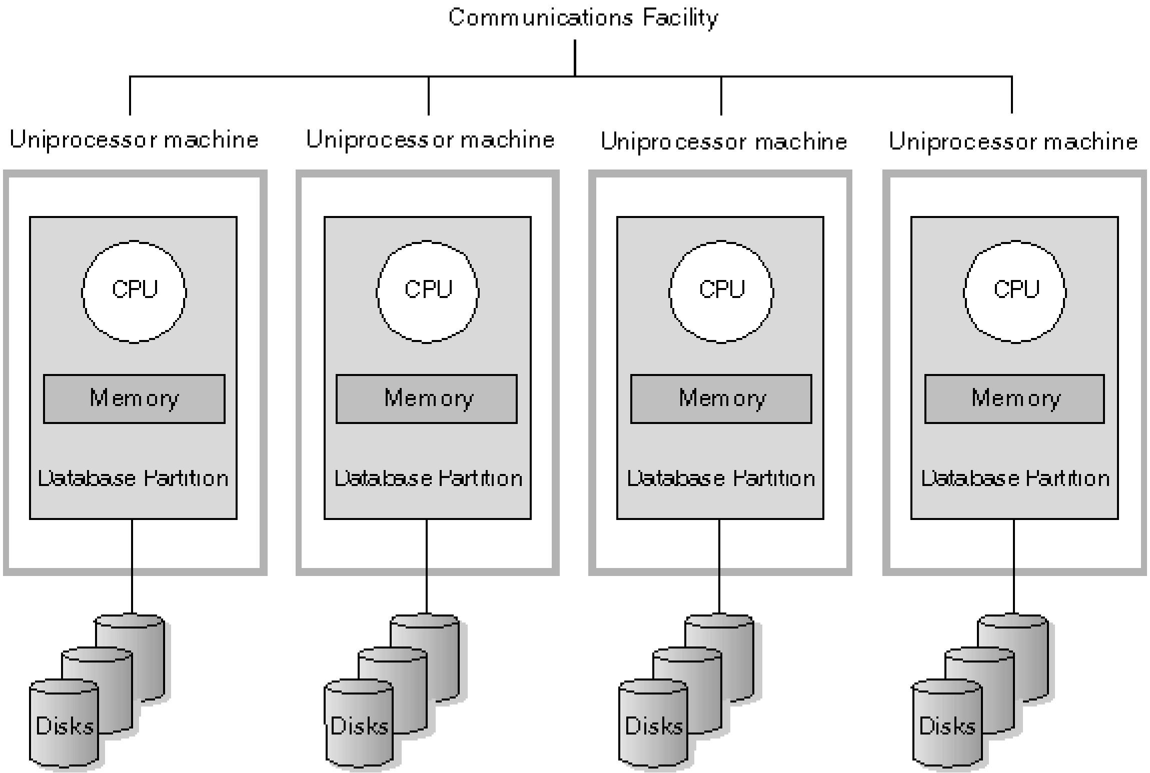
But the time passed and the buzzword "big data" started to flow around in the air, and also in the mass media and social networks. Here's google trends graph for "big data":



People were discussing the “three V” and the approaches to handle these huge amounts of data. Hadoop has emerged from the niche technology to one of the top-notch tools for data processing, getting more popular with more big companies investing into it, either by starting the broad Hadoop implementation, or by investing into one of the Hadoop vendors, or by becoming a Hadoop vendor by themselves. And as Hadoop became more and more popular, MPP databases entered their descent. You can take a look at the Teradata stocks as an example of this – for the last 3 years they are constantly falling down, and the main reason for this is that the new player has entered their market, and it is Hadoop.
So the question regarding “whether I should choose MPP solution or Hadoop-based solution?” asked by the newcomers is becoming really popular. Many of the vendors are positioning Hadoop as a replacement of the traditional data warehouse, meaning by this the replacement of the MPP solutions. Some of them are more conservative in the messaging and pushing the Data Lake / Data Hub concept, when Hadoop and MPP leave beside each other and integrating together in a single solution.

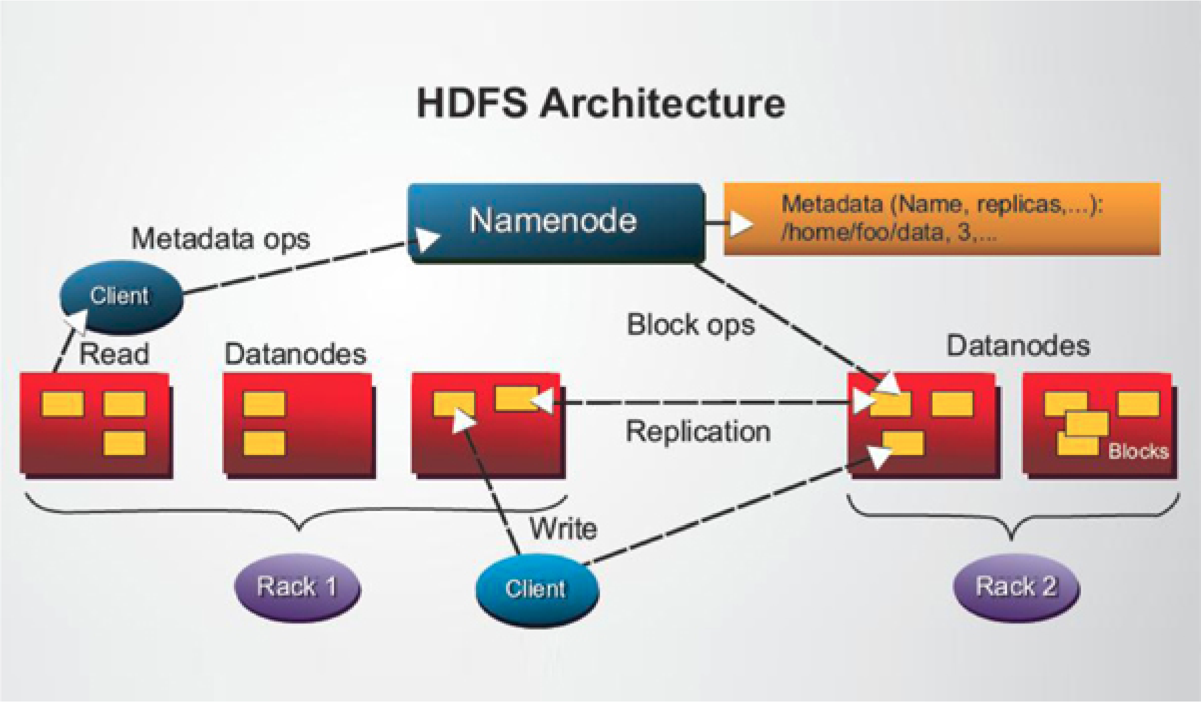


So what MPP is? MPP stands for Massive Parallel Processing, this is the approach in grid computing when all the separate nodes of your grid are participating in the coordinated computations. MPP DBMSs are the database management systems built on top of this approach. In these systems each query you are staring is split into a set of coordinated processes executed by the nodes of your MPP grid in parallel, splitting the computations the way they are running times faster than in traditional SMP RDBMS systems. One additional advantage that this architecture delivers to you is the scalability, because you can easily scale the grid by adding new nodes into it. To be able to handle huge amounts of data, the data in these solutions is usually split between nodes (sharded) the way that each node processes only its local data. This further speeds up the processing of the data, because using shared storage for this kind of design would be a huge overkill – more complex, more expensive, less scalable, higher network utilization, less parallelism. This is why most of the MPP DBMS solutions are shared-nothing and work on DAS storage or the set of storage shelves shared between small groups of servers. This approach is used by solutions like Teradata, Greenplum, Vertica, Netezza and other similar ones. All of them have a complex and mature SQL optimizers developed specifically for MPP solutions. All of them are extensible in terms of built-in languages and the toolset around these solutions supporting almost any of the customer wish, whether it is geospatial analytics, full-text search of data mining. All of them are closed-source complex enterprise solutions (but FYI, Greenplum would be open sourced in 2015'Q4) being around in this industry for years and they are stable enough to run the mission-critical workloads of their users.

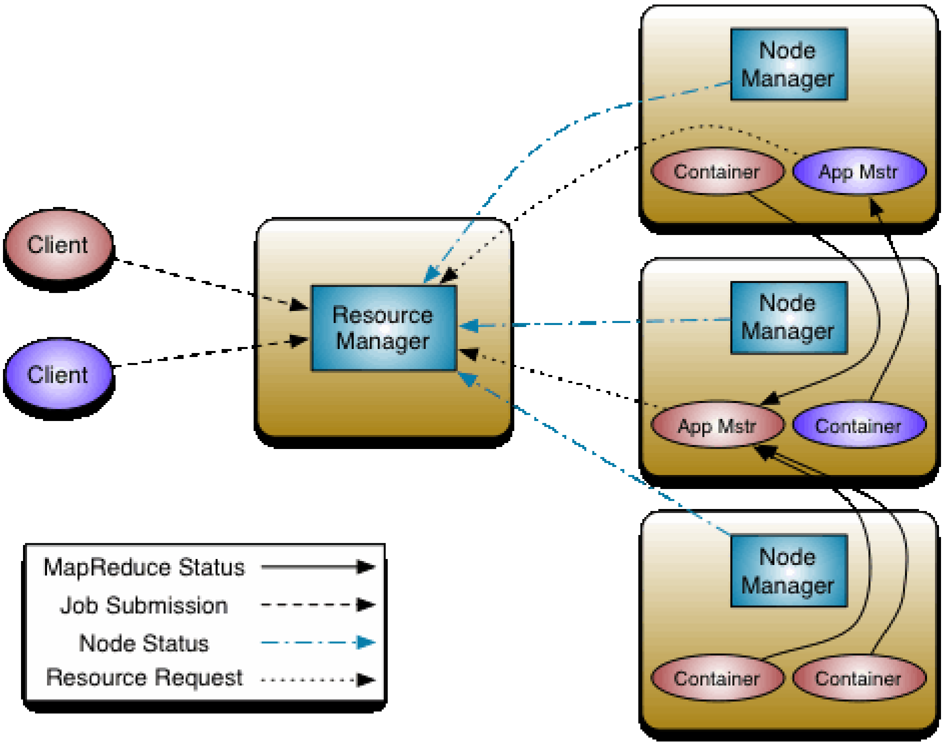


What about Hadoop? This is not a single technology, it is an ecosystem of related projects, which has its pros and cons. The biggest pro is extensibility – many new components arise (like Spark some time ago) and they are kept integrated with the core technologies of the base Hadoop, which prevents you from the lock-in and allows to further grow your cluster use cases. As a con I can put the fact that building the platform of a separate technologies by yourself is a hell lot of work and no one is doing it manually now, most of the companies are running pre-built platforms like the ones provided by Cloudera and Hortonworks.

Hadoop storage technology is built on a completely different approach. Instead of sharding the data based on some kind of a key, it chunks the data into blocks of a fixed (configurable) size and splits them between the nodes. The chunks are big and they are read-only as well as the overall filesystem (HDFS). To put it simple, loading small 100-row table into MPP would cause the engine to shard the data based on the key of your table, this way in a big enough cluster there is a huge probability that each of the nodes will store only one row. In contrast, in HDFS the whole small table would be written in a single block, which would be represented as a single file on the datanode's filesystems.

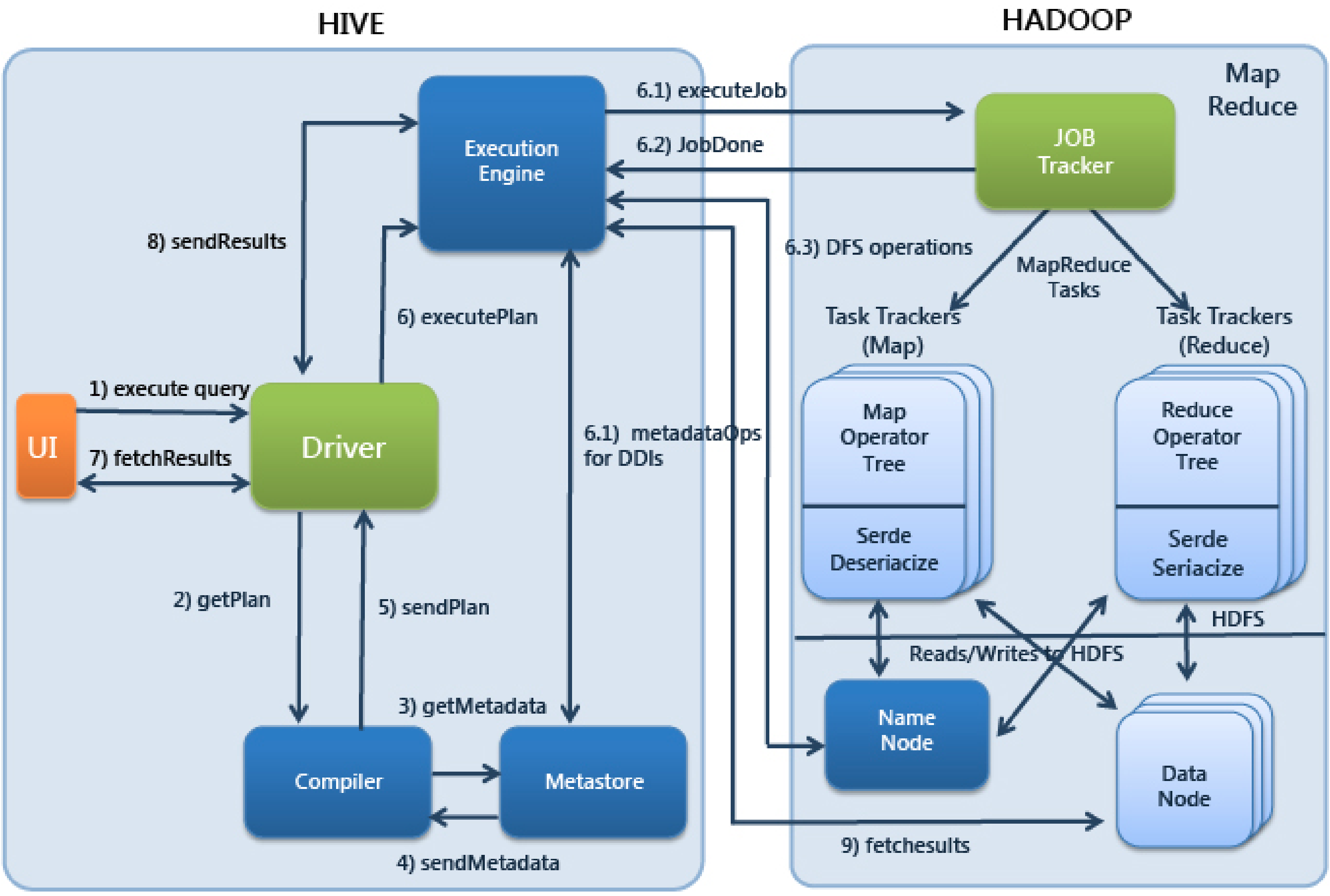


Next, what about the cluster resource management? In contrast to MPP design, Hadoop resource manager (YARN) is giving you more fine-grained resource management – compared to MPP, the MapReduce jobs does not require all its computational tasks to run in parallel, so you can even process a huge amounts of data within a set of tasks running on a single node if the other part of your cluster is completely utilized. It also has a series of nice features like extensibility, support for long-living containers and so on. But in fact it is slower than MPP resource manager and sometimes not that good in managing concurrency.

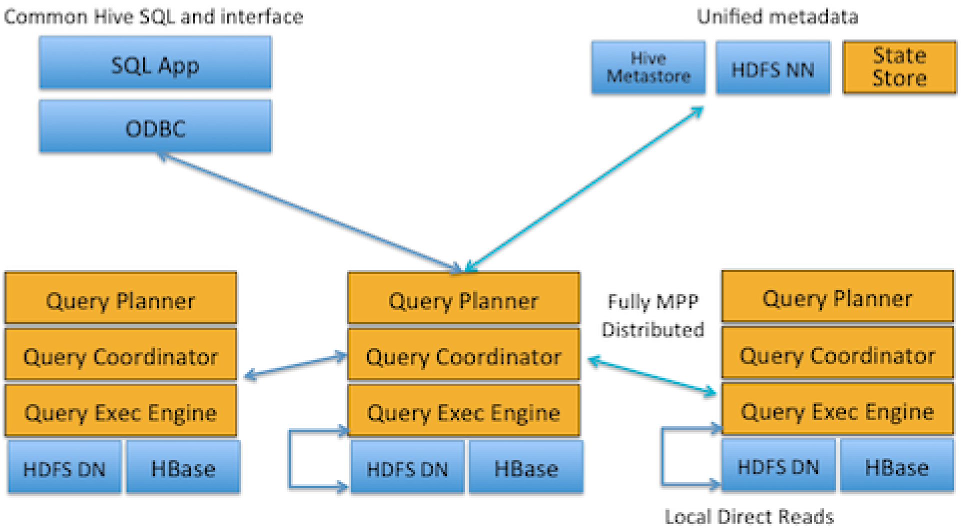


Next is the SQL interface for Hadoop. Here you have a wide choice of tools: it might be Hive running on MR/Tez/Spark, it might be SparkSQL, it might be Impala or HAWQ or IBM BigSQL, it might be something completely different like Splice Machine. You've got a wide choice and it's very easy to get lost in the technologies like this.

First option is Hive, it is an engine that translate SQL queries into MR/Tez/Spark jobs and executes them on the cluster. All the jobs are built on top of the same MapReduce concept and give you good cluster utilization options and good integration with other Hadoop stack. But the cons are big as well – big latency in executing the queries, lower performance especially for table joins, no query optimizer (at least for now) so the engine executes what you ask it to, even if it is the worst option. This picture covers the obsolete MR1 design but it is not important in our context:



Solutions like Impala and HAWQ are on the other side of this edge, they are MPP execution engines on top of Hadoop working with the data stored in HDFS. They can offer you much lower latency and lower processing time for the queries at the cost of less scalability and less stability, just like the other MPP engines:



SparkSQL is a different beast sitting between the MapReduce and MPP-over-Hadoop approaches, trying to get the best of both worlds and having its own drawbacks. Similarly to MR, it splits the job into a set of tasks scheduled separately giving better stability. Like MPP, it tries to stream the data between execution stages to speed up the processing. Also it uses fixed executors concept which is familiar to MPP (with its impalad and HAWQ segments) to reduce the latency of the queries. But it also combines the drawbacks of these solutions – not that fast as MPP, not that stable and scalable as MapReduce.

As I covered all of the technologies separately, here it the table bringing it all together:

	MPP	**Hadoop**
Platform Openness	Closed and proprietary. For some technologies even documentation download is not possible for non-customers	Completely open source with both vendor and community resources freely available over the internet
Hardware Options	Many solutions are Appliance-only, you cannot deploy the software on your own cluster. All the solutions require specific enterprise-grade hardware like fast disks, servers with high amounts of ECC RAM, 10GbE/Infiniband, etc.	Any HW would work, some guidelines on configurations are provided by vendors. Mostly recommendations are to use cheap commodity HW with DAS
Scalability (nodes)	Tens of nodes in average, 100-200 is a max	100 nodes in average, a number of thousands is a max
Scalability (user data)	Tens of terabytes in average, petabyte is a max	Hundreds of terabytes in average, tens of petabytes is a max
Query Latency	10-20 milliseconds	10-20 seconds
Query Average Runtime	5-7 seconds	10-15 minutes
Query Maximum Runtime	1-2 hours	1-2 weeks
Query Optimization	Complex enterprise query optimizer engines kept as one of the most valuable corporate secrets	No optimizer or the optimizer with really limited functionality, sometimes not even cost-based
Query Debugging and Profiling	Representative query execution plan and query execution statistics, explainatory error messages	OOM issues and Java heap dump analysis, GC pauses on the cluster components, separate logs for each task give you lots of fun time
Technology Price	Tens to hundreds thousand dollars per node	Free or up to thousands dollars per node
Accessibility for End Users	Simple friendly SQL interface and simple interpretable in-database functions	SQL is not completely ANSI-compliant, user should care about the execution logic,

		underlying data layout. Functions are usually required to be written in Java, compiled and put on the cluster
Target End User Audience	Business Analysts	Java Developers and experienced DBAs
Single Job Redundancy	Low, job fails when MPP node fails	High, job fails only if the node manages the job execution will fail
Target Systems	General DWH and analytical systems	Purpose-built data processing engines
Vendor Lock-in	Typical case	Rare case usually caused by technology misuse
Minimal recommended collection size	Any	Gigabytes
Maximal Concurrency	Tens to hundreds of queries	Up to 10-20 of jobs
Technological Extensibility	Use only vendor-provided tools	Mix up with any brand-new open source tools introduced (Spark, Samza, Tachyon, etc.)
DBA Skill Level Requirement	Average RDBMS DBA	Top-notch with good Java and RDBMS background
Solutions Implementation Complexity	Moderate	High

Given all this information, you can conclude why Hadoop cannot be used as a complete replacement of the traditional enterprise data warehouse, but it can be used as an engine for processing huge amounts data in a distributed way and getting important insights from your data. Facebook has a 300PB Hadoop installation and they still use a small 50TB Vertica cluster, LinkedIn has a huge Hadoop cluster and still they use Aster Data cluster (MPP bought by Teradata), and you can continue with this list.

Share this:

✉ in 232 f 🐦 G+

This entry was posted in Enterprises, Hadoop, SQL-on-Hadoop and tagged big data, enterprise, hadoop, mpp, Spark on July 13, 2015 [https://0x0fff.com/hadoop-vs-mpp/] .

12 thoughts on “Hadoop vs MPP”

Saurabh
July 13, 2015 at 7:47 pm

superbly informative !! thanks

Sulccc
July 13, 2015 at 9:01 pm

It would be interesting to see how the other new technologies in that area like HANA from SAP would be compared

---

0x0FFF Post author
July 14, 2015 at 7:15 am

In fact SAP HANA is mostly the same MPP. It uses RAM more efficiently and it has efficient columnar store, but many other solutions have similar approaches used. Here is a nice overview of its architecture: click and click.
Based on my experience MPP does not play well at OLTP field and primary use case for HANA is analytics, like the other MPP and Hadoop solutions.

DurhamDesi
September 29, 2015 at 12:20 pm

Fantastic article...you explained it like a Boss 🙂

John
March 1, 2016 at 10:41 pm

Thoughts about HAWQ ? Greenplum hadoop engine?

0x0FFF Post author
March 2, 2016 at 4:15 pm

I know about HAWQ as I'm working at Pivotal 🙂 In fact, HAWQ is a mixture of MPP approach with Hadoop approach, so you can consider it in the middle between these two worlds. Soon my article will out on blog.pivotal.io discussing design principles of HAWQ and comparing it with MPP and Spark, so stay tuned

Femi Anthony
March 18, 2016 at 8:27 am

Fantastic article.

kenlf0508
March 30, 2016 at 8:58 am

This article contains so much information that I needed. And the analyzation of different between MPP and Hadoop is very accuracy as well as completion. I will reference this article when I setting up my database. Thank you very much to the author.

Gurneet
March 30, 2016 at 9:47 pm

Yet another insightful article! The SparkSQL section was my favorite

Pingback: Hadoop, MPP | gerleim

Timmy
November 4, 2016 at 8:37 pm

very good article.

Rajesh
December 6, 2016 at 11:02 am

Concisely written, well articulated and informative article! These days we have so much information but something like this is rare! Thank you author!