INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 4, Issue 6)
Available online at: www.ijariit.com

High performance computing v/s big data

Nikita Mutreja
nikita.mutreja@hotmail.com
Amity University, Noida, Uttar Pradesh

Sanyam Jhamb
sanyam.jhamb@gmail.com
Amity University, Noida, Uttar Pradesh

ABSTRACT

Simulation has become a "must have" item in the technology toolbox for manufacturers who wish to optimize the product development process, reduce production costs, and speed-time-to market. Along with Big Data insights and HPC solutions, simulation can enhance the product design process by leveraging to drive product innovation, improve time to a time value. These models (Big data and HPC) provide the advanced capabilities that are needed by the manufacturers to get to the market faster than their competition. In this paper, we analyze the ecosystems of the two prominent paradigms for data-intensive applications, hereafter referred to as the high-performance computing and the Big Data paradigm. Further, the characteristics of the two paradigms have been discussed, along with comparisons and contrasts of the two approaches. It also covers the scope of these paradigms and sheds light upon the specific workloads that utilize them. At last, we discuss the convergence of both paradigms; the best of both world's approach.

Keywords— *High performance computing, Big data, Job scheduling, Hadoop, YARN, Converging paradigms, MapReduce, HDFS*

1.HIGH-PERFORMANCE COMPUTING

Historically, it means using supercomputers for large scientific applications. In today's world, it means using clusters and the cloud.

1.1 Benefits

Time: As the name suggests, High-Performance Computing, it gives high performance, which means that it's quicker, and hence it saves the engineers' time which in turn saves the company's money.

It helps in the 'what if' scenarios. Basically, we notice the effect of changes in material or slight changes in the geometry, without much more additional time.

1.2 HPC and simulation

The high-performance computing helps to solve large models using shared memory and parallel support, which means that it will take full advantage of the multi-core systems. Plus, it is used with GPU for better 3D graphics, which is beneficial for simulation.

HPC has become the only way to increase design productivity, as simulation increasingly requires optimized computing tools which provide the technology required to generate, analyze and manipulate the product data efficiently.

2.BIG DATA

So, we are constantly producing data for example, via social media, GPS etc. But it is way beyond that. On a daily basis, we upload approximately 55 million pictures, 300 million tweets, and 1 billion documents. In total, we produce 2 quintillion bytes of data. This is Big Data.

Further, the cloud or networks of multiple servers are used for the processing, which happens in minutes. For example, Netflix is an application which analyses the big data of its viewers, like popular shows and watching patterns through which they produce the perfect shows for their viewers. Hence, we can conclude Big Data as an evolving term that is referring to any sorts of the voluminous amount of data that has the potential to be mined for information.

Any small data cannot be regarded as big data. Small data is another term that's often used to describe data whose volume and the format is easily used for self-service analytics. A very common phrase concludes it all: "big data is for machines; small data is for people.

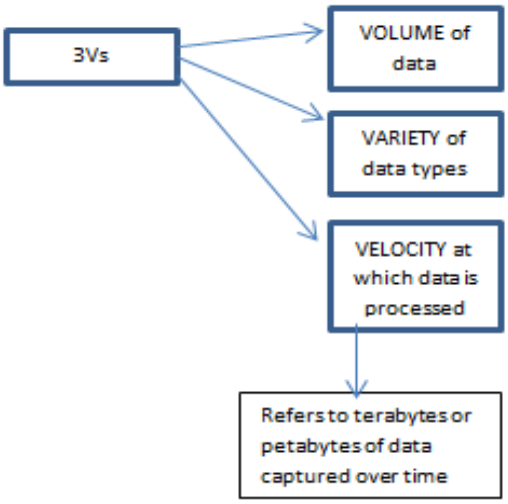

Fig. 1: Diagrammatic representation of 3V's

2.1 Big data and simulation

For example, a company develops new dishwashing liquids, predictive analytics and modeling are done to predict how moisture will excite certain fragrance particles so that the correct aromas are released during the dishwashing procedure. This was an example wherein Big Data analytics works with simulation driven product design.

Simulation and Big Data, together, produce the most effective results. Big Data analytics processes the simulation data rapidly, which enables the engineers to gain valuable information and convert it into something better.

2.2 What is the hype all about?

Much of the hype has been about extracting meaning: From large unstructured and semi-structured datasets, using scalable analysis and storage tools like Hadoop, HBase, Cassandra etc.

Some successes in scientific computing for unstructured data: Petascale machines are solving important scientific problems through the generation and analysis of large datasets. For example Biological conversion of cellulosic feedstock for biofuels, 3D simulation of blood flow, Combustion of turbulent lean fuel mixtures, Chemical structure of HIV virus, Prediction of Ice and Climate Evolution at extreme scales.

3. BIG DATA V/S HPC

3.1 Infrastructure

HPC infrastructure was traditionally built for scientific applications which required high-end computations. In the HPC cluster, 'compute' and 'data' infrastructure is separate.

Fig. 2: Working of HPC cluster infrastructure

Hadoop, on the contrary, was mainly built to process large amounts of data It introduced an integrated 'compute' and 'data' infrastructure.

Fig. 3: Working of Hadoop infrastructure

3.2 Data Management

In HPC, it all happens through files. It means that the resource manager, runtime systems and application are very tightly integrated.

MapReduce was the Hadoop runtime layer for processing data, but as per the requirements of the application like: run times for record based, random access data, iterative processing (Spark), stream (Spark Streaming) and graph processing came into the scene. YARN played a major role for these frameworks as it provided support for multi-level scheduling. This enabled the frameworks to deploy their own application-level scheduling, like a layer, on top of the Hadoop managed storage and compute resources. YARN takes control of the lower level resources, whereas the higher level resources use an application level scheduler to optimize resource usage for the framework. Hence, an application uses the abstraction provided by the runtime system (e.g. MapReduce) and does not interact with the resource management directly.

Hence, In Big data, the data management happens through higher-level abstractions, as found in the Hadoop environment.

3.3 Resource planning

In terms of resource management, there are 2 main types of resource managers: Monolithic and Modular. Monolithic are the ones wherein the components are interconnected and interdependent rather than loosely coupled as in the case of modular. In a tightly coupled architecture, all the components must be present in order for the code to compiled or executed. These can be further classified into 3 types of architectures: Centralized, Multi-level and Decentralized. Nowadays the monolithic resource managers are being replaced by application-level scheduling (Multi-level and Decentralized).

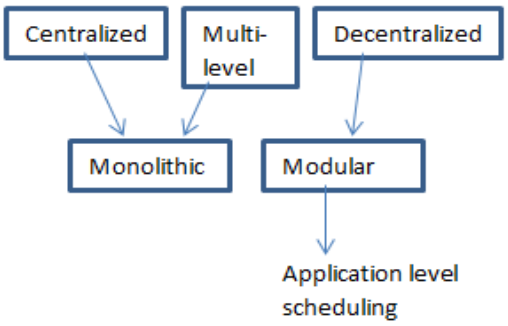

Fig. 4: Categorization of resource managers

In HPC Data locality does not take place. This means that large data is moved towards computation instead of moving the computation close to where the actual data is residing. Data locality is one of the major features provided by the Hadoop environment. HPC schedulers are designed for rigid frameworks with regular resource requirements. This means that the resource demands of the frameworks are fixed, with regard to the number of cores. Also, HPC makes use of MPI, also known as Message Passing Interface, which is Monolithic software. Hence MPI frameworks are tightly coupled. Further, the scheduling has to happen in such a way, where the execution happens simultaneously on the system. Scheduling happens on a job level, which means that the data is taken towards the compute nodes. For, heterogeneous workloads (or loosely coupled tasks), which include small, short-run and long-running batch-oriented tasks, scheduling seems like a challenge. The solution is Pilot Jobs. It refers to the concept of providing multi-level or application level scheduling on top of the system provided schedulers. YARN and Mesos are examples of multi-level schedulers. Google's Omega and Sparrow belong to the same class of schedulers. They also reside under the class of decentralized schedulers. At the time of Hadoop-1, it belonged to the class of Centralized schedulers wherein job tracker was used as a resource manager. This constricted the flexibility of the framework. Hence, Hadoop was used on top of the HPC clusters, but the problem of Data Locality persisted and was a big limitation. Further, YARN, which is a part of Hadoop-2, was designed as a solution to the limitation. It was meant to support the processing of heterogeneous workloads. Hence, YARN enabled the scheduling of processes on a task level rather than on a job level. This helped in improving the overall utilization of the HPC cluster due to the simultaneous shifting resources between frameworks.

The scheduling unit is also known as Container. And these units are returned from the resource managers. In Big Data, the requested number of resources is not returned by the managers. YARN can also request de-allocation of resources as per the resources available at the moment. In HPC, this does not happen.

3.4 Speed difference

To get a better idea, a specific analysis in a construction firm was reviewed. The R&D department of the company shifted to the

use of HPC systems after the amount of procedural data increased. We had the test results and time-related details for the analysis using traditional data analytics and also using HPC systems to analyse the data.

Table 1: Specifications of the case study

Job	Cast in place mechanical anchor concrete anchorage pull out capacity analysis.
Materials	Steel + Concrete
Procedure	3D Non-Linear contact + Damage analysis
Number of elements	1626338
Number of DOF	1937301

Table 2: This table demonstrates the runtime

System	Elapsed time
Single 12 core system	29 hours 3min 41 sec
HPC- based InfiniBand enabled 72 core to a parallel computing system	5 hours 30 mins

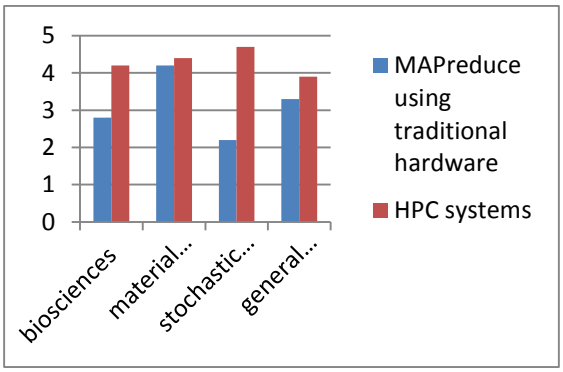

Fig. 5: Outcome of Table 1 and Table 2

The bioscience sector involves a critical analysis of severe and complicated simulations which were among the reasons that pushed the engineers to build what today is HPC. But again with increasing issues and also with increasing availability of health-related information for a very wide part of the population it has become important to run that data in the available simulations, now running such massive data over such complicated simulations results in a lot of time being used, if the data is analysed traditionally but using HPC systems makes all the difference. For instance, Schrödinger which is based in Germany is currently using public cloud resources which are based upon HPC systems to identify and separate potential candidates for new medications to tackle fatal diseases like cancer, leukaemia etc.

3.5 Job scheduling

Talking about the traditional approach when dealing with data analytics, the procedure is simple.

Basically, to process such a huge amount of data, we simply distribute it into chunks and have a number of different systems processing it. This method is called MAPREDUCE where we are mapping a lot of data into chunks to a number of systems and after it is processed we are reducing it back.

This method makes failover trivial and avoids any sort of communications between the processors so that if one process fails only that process has to be restarted not the whole.

Yet, MapReduce being powerful cannot fit all problems and has a lot of disadvantages in terms of scheduling.

Hadoop uses an integrated scheduler consisting of a master job tracker which supervises the MapReduce work. The job track is aware of all the node, data location, and job placement so if a node fails it reassigns that specific task to some other node. The problem arises not when the number of data Increases but when the rate of flow increases.

Here HPC solves the issue as it requires fine grain control of resources like cores, accelerators, memory etc. Providing parallel computing capabilities HPC provides what Hadoop couldn't. In contrast to HPC, Big data schedulers are more focused. They conduct array processing of jobs. HPC enables parallel processing of jobs.

Why does array processing of jobs happen? The answer lies within Flynn's classification of Computer Architectures. This classification refers to the 'stream concept'. It's divided into SISD (Single instruction single data), SIMD (single instruction multiple data), MISD (multiple instruction single data) and MIMD (multiple instructions multiple data). In SIMD, each instruction is executed on a different set of data by different processors, that is, multiple processing units of the same type process on multiple data streams. Hence, this group is dedicated to array processing machines. The same concept is applied to big data.

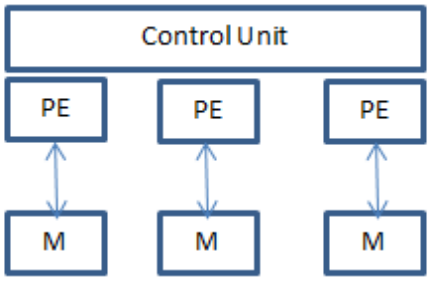

Fig. 6: Representation of execution of jobs

In Fig-6, PE stands for processing element and M stands for memory

3.6 Case study

Affecting the prominent sectors of the market like cybersecurity, online transactions, social media and whatnot, HPC holding hands with data analytic techniques has become a major tool for players to make the revenue flow in. HPDA-focused servers gave approximately a 14% growth rate in the last 5 year period which is incredible for a tool that was being used as an apparatus in the academic sector.

So basically HPC has increased the possibilities of what can be achieved with the humongous amount of data produced each day.

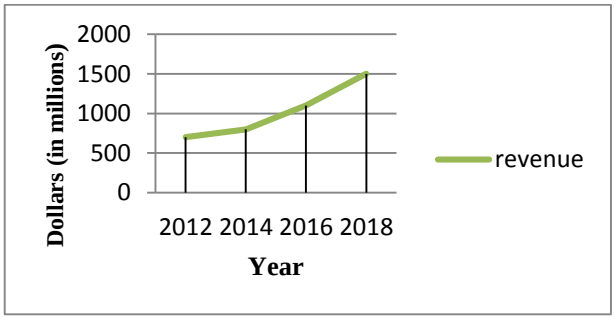

Fig. 7: Revenue generated by the Big Data industry

4. ANALYSIS

On Hadoop running, HDFS modules data is stored on compute nodes and is accessed locally. On the other hand, HPC clusters employ separate nodes to store the data which gives memory a dedicated retrieval approach.

Clemson Palmetto HPC cluster is one of the leading in the top US systems available. A plus point of Clemson clusters is that they allow the use of a number of file systems like the orange fs. Now, we can use ofs as a memory platform and use it to run big data applications. Orange fs can run Hadoop without any modifications so a number of applications can be executed and a number of tests can be done using this combination. A major component of the structure is the use of java native interface. Hadoop MapReduce is written in Java whereas the orange fs client libraries are in C. JNI allows the ofs file to run the Hadoop applications upon the client libraries. To assess the execution time based on the scheduling of the two systems we use grep and word count. Grep and word count are data-intensive benchmarks which on a simple scale give vivid comparative results.

The sample size of the file is taken to be 500mb and the number of files reaches until 4096.

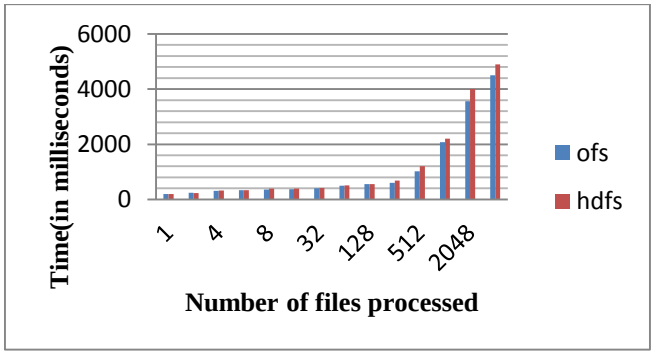

Fig. 8: HPC processes files faster than big data

As per figure 8, the comparison between the two approaches is an average for both the grep and the word count test wherein we can evidently see that for smaller files the speed and efficiency of scheduling are similar but as the quantum increases, OFS proves to be significantly faster.

This assessment depicts how the use of HPC improves what Hadoop as an analytic tool lacks and that is scheduling that can be used with high-speed transfers and parallel computing. Traditional scheduling like FIFO and capacity can be exchanged with efficient HPC schedulers like muoi and torque which when combined with data flow techniques of HPC can make tools like Hadoop become much more productive.

5. CAN HPC AND BIG DATA CO-EXIST?

HPC and Big Data, both were initially created for different sets of workloads. In case of HPC, parallel computations whereas data storage in case of big data. Since the time YARN was designed, big data has been able to support heterogeneous workloads effectively. At the same time, new developments in HPC came up, wherein Pilot Jobs enabled HPC to support loosely coupled workloads. The big data consists of a very high level of abstractions in terms of processing and storage. Contrary to HPC, YARN can conduct data-aware scheduling, because of which it requires an advanced level of integration between the framework and the system schedulers.

As compared to the functionalities of both the paradigms, big data's ecosystem exceeds that of HPC. Many proposals are being put forward with respect to the convergence of the two models, but one problem still lies. Convergence leads to loss of data locality-aware scheduling, which is one of the major functions of big data. The ideal approach is the extension of Pilot job abstraction to YARN and the usage of Pilot Data for data locality-aware scheduling.

Currently, there is a need to merge the two paradigms and try to produce ecosystems that have the performance of HPC and the usability and flexibility of the big data. This ecosystem will bring out the 'best of both the worlds'.

6. REFERENCES

[1] https://searchdatacenter.techtarget.com/definition/high-performance-computing-HPC
[2] https://www.nimbix.net/3-ways-big-data-hpc-converging/
[3] http://hibd.cse.ohio-state.edu/
[4] http://hbase.apache.org/
[5] https://ai.google/research/pubs/pub27898
[6] https://www.nimbix.net/why-is-infiniband-support-important/
[7] https://www.datanami.com/2015/01/26/rethinking-hadoop-for-hpc/
[8] https://www.nextplatform.com/2015/03/10/hpc-clusters-lead-a-double-life-with-hadoop/
[9] http://www.adminmagazine.com/HPC/Articles/Is-Hadoop-the-New-HPC
[10] https://www.hpcwire.com/2014/02/14/adapting-hadoop-hpc-environments/
[11] https://insidehpc.com/2017/03/accelerating-hadoop-spark-memcached-hpc-technologies/
[12] https://www.scads.de/en/news-en/blog-en/226-big-data-frameworks-on-hpc-machines
[13] http://dsc.soic.indiana.edu/publications/HPC-ABDSDescribed_final.pdf
[14] https://hal.archives-ouvertes.fr/hal-01633507/file/bigdata_hpc_colocation.pdf
[15] https://www.rdmag.com/article/2014/03/big-data-meets-hpc
[16] http://dsc.soic.indiana.edu/publications/HPC-ABDSDescribed_final.pdf
[17] https://www.sciencedirect.com/science/article/pii/S2405918815000045
[18] https://www.hpcwire.com/solution_content/hpe/financial-services/big-data-hpc-speeding-innovation-high-frequency-trading/
[19] https://tudresden.de/zih/hochleistungsrechnen?set_language=en
[20] https://searchdatacenter.techtarget.com/definition/high-performance-computing-HPC
[21] https://www.nimbix.net/3-ways-big-data-hpc-converging/
[22] http://hibd.cse.ohio-state.edu/
[23] http://hbase.apache.org/
[24] https://ai.google/research/pubs/pub27898
[25] https://www.nimbix.net/why-is-infiniband-support-important/
[26] https://www.datanami.com/2015/01/26/rethinking-hadoop-for-hpc/
[27] https://www.nextplatform.com/2015/03/10/hpc-clusters-lead-a-double-life-with-hadoop/
[28] http://www.admin-magazine.com/HPC/Articles/Is-Hadoop-the-New-HPC
[29] https://www.hpcwire.com/2014/02/14/adapting-hadoop-hpc-environments/
[30] https://insidehpc.com/2017/03/accelerating-hadoop-spark-memcached-hpc-technologies/

[31] https://www.scads.de/en/news-en/blog-en/226-big-data-frameworks-on-hpc-machines
[32] http://dsc.soic.indiana.edu/publications/HPC-ABDSDescribed_final.pdf
[33] https://hal.archives-ouvertes.fr/hal-01633507/file/bigdata_hpc_colocation.pdf
[34] https://www.rdmag.com/article/2014/03/big-data-meets-hpc
[35] http://dsc.soic.indiana.edu/publications/HPC-ABDSDescribed_final.pdf
[36] https://www.sciencedirect.com/science/article/pii/S2405918815000045
[37] https://www.hpcwire.com/solution_content/hpe/financial-services/big-data-hpc-speeding-innovation-high-frequency-trading/
[38] https://tu-dresden.de/zih/hochleistungsrechnen?set_language=en
[39] http://www.cs.virginia.edu/~hs6ms/publishedPaper/Conference/2016/Hadoop-cameraready-BigData2016.pdf
[40] https://digitalcommons.unf.edu/cgi/viewcontent.cgi?article=1381&context=etd
[41] https://www.informs-sim.org/wsc17papers/includes/files/069.pdf
[42] http://www.sersc.org/journals/IJFGCN/vol8_no2/33.pdf

