ANALELE UNIVERSITĂȚII
"EFTIMIE MURGU" REȘIȚA
ANUL XX, NR. 1, 2013, ISSN 1453 - 7397

Florica Novăcescu

Big Data in High Performance Scientific Computing

The main concern of this study is to find the bottlenecks that are caused by the deluge of data, in the supercomputing community. To be more precise, the research targets the big data community, and is concentrated around identifying the new challenges posed by the huge amounts of data resulted from the numerical simulations using supercomputers.

***Keywords**: supercomputing, big data, high performance scientific computing, big data challenges*

1. Introduction

Computation in science and engineering can be regarded as a focal point around which "revolves" the need to have knowledge of computer architecture, operating system, network technology, programming language, numerical algorithms, mathematical model, physical phenomena and theory of computation, respectively.

If we discuss about the cluster or the computing center, and not about the supercomputer as a finished product, come into question issues of operating systems, parallel file systems, large databases (multi-index and/or multi-resolution VLDB - Very Large Database), job scheduler, profiling, storage solutions (network file systems, and so on). Here, the issues are far from being solved.

High performance scientific computing is located at the intersection of a number of scientific disciplines and skills sets, in the sense that it requires at least basic knowledge and skills in these scientific fields.

Computations come from an applicative context, therefore, are required some knowledge of physics and involved engineering sciences, hence the name "Computing in Science and Engineering". Further, problems in these applicative areas be transformed into linear algebra problems, and sometimes, even in combinatorial problems, be so subjected to implementation, being obvious that are required knowledge of numerical analysis, linear algebra, and discrete mathematics.

In order to expand aspects of the implications of this area in addition to those already mentioned, it requires an understanding of computer architecture both at the CPU and the parallel computing level.

2. The parallelism. Scope of application

In general, conventional architectures contain a processor, the memory system, and the functional data processing unit. Each of these components involves major performance bottlenecks. The parallelism addresses each of these components in significant ways.

Different applications use different aspects of parallelism, for example, data applications using high transfer, server application using high bandwidth network, and scientific applications that typically uses higher processing and memory system performance [2]. Therefore it is important to understand each of these performance bottlenecks.

2.1. Data parallelism

Many problems in scientific computing involve processing of large quantities of data stored on a computer. If this manipulation can be performed in parallel, i.e., by multiple processors working on different parts of the data, we speak about data parallelism [1].

Data parallelism is a structural parallelism and it applies to those programs which define the regular data structures over which runs the same operations. The parallelism can be obtained by applying the same operation simultaneously on some or on all elements of a set of data.

This type of parallelism is suitable for SIMD architectures and those vectorial, but, also, it can be implemented on MIMD architectures.

Data parallelism is recommended for implementation of algorithms which involves the synchronous handling of distributed data structures such as multidimensional arrays and graphs [2].

Processing of array data by loops or loop nests is a central component in most scientific codes. The typical examples are linear algebra operations on vectors or matrices. Often the computations performed on individual array elements are independent of each other and are hence typical candidates for parallel execution by several processors in shared memory as shown in Figure 1 (an example of medium-grained parallelism) in which the iterations of a loop are distributed to two processors P1 and P2 (in shared memory) for concurrent execution [1].

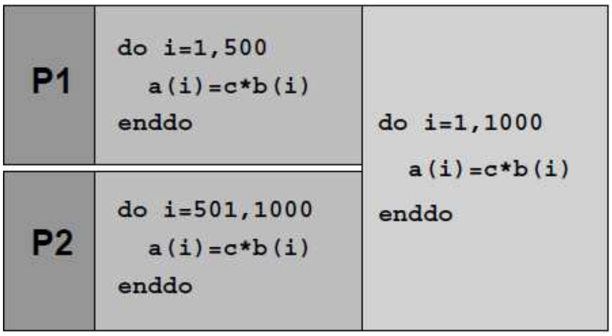

Figure 1. An example of medium-grained parallelism [1]

2.2. Functional parallelism

It is characterized by the fact that each processor performs a different task computation and communicates with the rest of the processors to achieve the final outcome of the application.

Control parallelism applies when program data structures are irregular and operations are varied and different. In this case, parallelization is achieved by dividing the program into multiple computing tasks that are executed in parallel on different processors independently and asynchronously. This type of parallelism is suitable for MIMD architectures and is present for concurrent programming languages [2].

The fact that the processes can interact with each other involves the explicit expression of the interaction by the programmer using some synchronization and communication primitive. One of the most popular models is the communication through messages (message passing) in which the processes communicate with each other by exchanging messages (sending and receiving messages).

Creating libraries of functions for communicating through messages is another approach, and languages such as Fortran, C, C++ contains such functions libraries to communicate through messages such as MPI, MPL, PVM [3].

The simplest form of functional parallelism is to implement pipelined algorithms, the algorithm is divided into a number of stages of execution, and each phase is assigned to a processor system and the output from one stage being input for the next stage, similar process with the pipeline execution of instructions [4].

3. Big Data

Big Data comprise a large and complex set of different structured and non-structured data sets (> 10 terabytes) that are difficult to be processed using the traditional tools and data management practices.

The "big data" term refers to the huge amounts of unstructured data produced by high-performance applications that fit into a very large and heterogeneous family of application scenarios: from scientific computing applications to social networks, from e-governance to health information systems, and so on [5]. Data resulted from these applications have some common characteristics, among which may be mentioned:

- *large-scale data* - feature that refers to the size and distribution of data warehouses;
- scalability issues - which refers to the ability to run applications on huge large-scale data repositories (for example, to scale rapidly over inputs with increasing size);
- support for advanced ETL processes - (*Extraction-Transformation-Loading*) from raw data to somewhat structured information;
- design and development of easy and interpretable analysis over large data warehouses, to obtain information and to extract useful knowledge from them;

Big Data is something that exceeds the processing capacity of conventional database systems. The data is too big, moves too fast, or do not fit with certain database architectures. So, to get a value from these data, or to effectively give significance of these data, an alternative method must be chosen to process them. Big Data analytics represents developed tools and capabilities to retrieve a data perspective. These big data analyzes take place if advanced analysis techniques operate on large data sets [8].

At the same time, big data refers to issues of capturing, storing, managing, analyzing massive quantities of different types of data. Most commonly, these massive amounts of different types of data refers to terabytes or petabytes of data stored in multiple formats resulted from different internal and external sources, with strict requirements for speed and complexity analysis [8].

Thus, the concept of "big data" involves collecting large amounts of data and then looking for patterns and new revelations. So, the quality of data resulted from various new sources, is still a topical issue [9].

Therefore, we can distinguish three different types of scientific data:

- Observational data - uncontrolled events that happens and about which we could record data;
  - The examples include astronomy, Earth observation, geophysics, medicine, commerce, social data, Internet of things;

- Experimental data – we design controlled events in order to record data about them;
    - The examples include particle physics, photon sources, neutron sources, bioinformatics, and product development;
- Data derived from simulation – we create a model, simulate something and then we record the resulted data.
    - The examples include weather and climatology, nuclear and fusion energy, high-energy physics, materials, chemistry, biology, fluid dynamics, renewable energies [9].

4. Big Data challenges

Significant improvements in the management of scientific data would increase research productivity in solving complex scientific problems.

The next generation of e-Science infrastructures (running on millions of cores) will start from the premise that exascale HPC applications (exascale computing refers to computing systems capable of at least one exaFLOPS, and one exaFlop is a thousand petaflops or a quintillion, 10^{18} floating point operations per second) will generate scientific data at very high frequency (terabytes/s).

Hundreds of Exabyte of data (distributed over many computing centers) are expected until 2020 to be available through heterogeneous storage resources for access, analysis, post-processing and other scientific purposes [11].

Data mining and knowledge discovery in databases is a research direction.

Key of success in the context of "big data" is represented of analysis. Data archiving, provenance, cleaning, protection and movement are serious problems, but they are known today, and will probably be addressed in a more or less similar manner throughout the world [5].

Some of the significant challenges of Big Data field are listed below:

- Big data will "live" in the cloud - either in a cloud as the currently existing or in a developed cloud, formed to meet the needs of large data [9];
- Visual analysis will be a necessity for large data;
- While structured searches will remain the main, the unstructured searches and graphical analysis applications can reach to cover those structures [14];
- Although there are software like Hadoop and MapReduce [12], and other similar software new approaches and programming models for large data analysis will be developed and implemented for many applications - especially those involving unstructured queries and graphical analysis [13].
- Since the data will be impossible to move, the analysis technique may need to be "moved" to data, rather than "bringing" the data for analysis, as is now a common practice.

- The calculation engines may need to be and live inside the data (and thus inside the cloud). In fact, depending on the nature application as end user, this could turn a large number of processors computer in some great, dedicated and working on data using in-situ analysis [15].
- Computing architectures for the big day will have to adapt to unique and effective non-local memory and execute code that is difficult to predict [11].

Below analysis shows the evolution of modern data infrastructure, in terms of big data appearance. So, it follows that the data is in a rapid expansion, and Nathaniel Rowe in [16] shows this trend for several years. The average annual growth rate of data in 2012 is two times higher than the rate reported three years ago, i.e. in 2009, as shown in Figure 2.

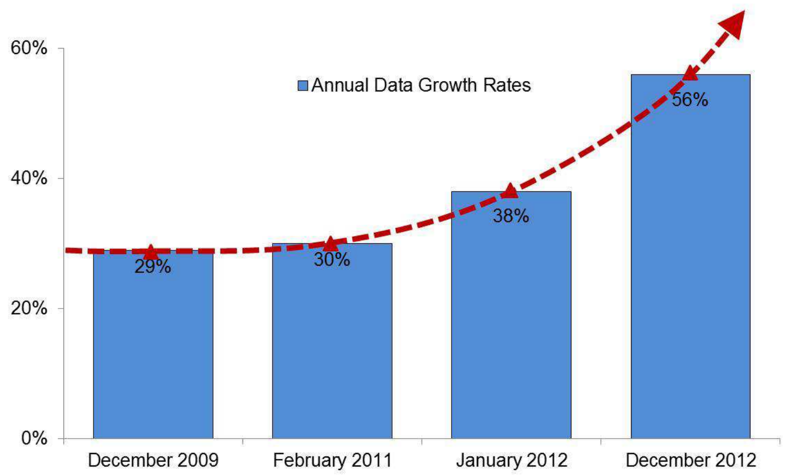

Figure 2. The explosion of data in the period 2009 - 2012 [16]

At a growth rate of 56% every year, all storage requirements and implicit associated costs, doubles at every 19 months, however, this data expansion show no sign of stopping.

Not only the amount of data is growing, but also it expands the diversity of information sources both internally and externally. In 2011, it recorded an average of 26 unique data sources feeding the analytical applications and systems, but this number of data sources increased by 19% in 2012, reaching 31 sources [16].

A substantial amount of this data is unstructured, i.e. 48% according to a study presented in. Therefore, the biggest challenge is to find a resource that can help manage these unstructured data.

Currently, the need for unstructured data management system is well known. For organizations with more than five terabytes of data, it was found that only a quarter (24%) of them could manage and use most of their unstructured data.

Finally, the whole complexity given by increasing amount of data, data sources and data formats not only has an impact on IT, but also on daily business operations. Finding the reliable information and it's supplying as quickly both to the managers and the decision makers, has become a daunting task with which many companies are struggling.

With staggering data growth rates, conclusion is that there have never been more available data, but nor possibilities to access, analyze and share these data more effectively. Especially in the case high-performance computing environments (HPC), the data warehouses can grow very quickly and although the computing server technology has kept pace, the storage possibilities have not achieved this, thus creating a barrier between scientists and their data [5].

A storage system that uses a scaling methodology to create a dynamic storage environment will support balanced development of data as needed, resulting the scale-out storage concept.

Implementation of the scale-out storage can eliminate the HPC storage obstacles and also "put" the data immediately into the hands of those who need of them.

When using the scale-out storage technology, occurs the problem of adding storage nodes working in tandem according to need of additional storage resources. Scale-out storage architecture is designed to scale both capacity and performance.

Torsten Hoefler et all in [6] support their idea through which the data transfer can be improved if it is possible to reduce the impact of high latency by introducing of level - 4 relays along the route through which will pass data. These relays allow a cascade of TCP connections, each with a lower latency, resulting in better data transfer [6]. This type of TCP connections would have a positive result on some typical applications such as Big Data in HPC.

Computer security is another challenge of Big Data area, which increases in complexity and also involves the increasing complexity of network connection. So, in [7], it supports the idea of developing new algorithms for computer security suitable to work with big data applications.

Jun Zhang et all, in [10], found that increasing the number of applications for large data, as web search engines, shall have a high full-time tracking availability, storage and analysis of a large number of logins for real-time access users. Therefore, researchers have presented an approach to save this archive of data streams within a cluster of databases for a speedy recovery. This method is based on a simple replication protocol with a high performance data loading and a query strategy.

Big Data is data that exceeds the processing capacity of conventional database systems, data that is too big, moves too fast, or do not match to database architecture structures, and to get a value from this data, an alternative method must be chosen for their processing [9].

In general, the scientists avoid working with databases because the database software generally lacks functionality and features which could make it more attractive, and fast enough for science.

Current approaches relate to techniques for multi-scale and multi-indexing of metadata for efficient handling of scientific data storage.

For example, an important feature that is not yet fully developed is the ability to manage data varying in time at high resolution.

The time-traveling query processing at arbitrary levels of resolution is the key to the development of mature and dedicated databases in CSE (Computational Science and Engineering).

Ideal would be, that a database system meet features such as:

- Parallel query processing in distributed and high performance;
- Have a object-relational oriented structure;
- Allows a huge number of levels for compressed metadata;
- Provide high multi-indexing capabilities;
- Provide a time-travelling multi-resolution query processing.

In a typical supercomputing center, some nodes are dedicated for making computing, being called processing elements, while the others are used for storage. All traffic to and from the supercomputing center must pass through the so-called gateway level. This already creates a huge obstacle for uploading and downloading data to and from the supercomputing center. At the gateway level, the users are identified, logged, and assigned with specific security clearance and logged for audit.

Internet connection can vary from a very good connection, with limited bandwidth, to one very poor, with data losses, duplication of packets and fluctuations in response times. This, typically, puts the most serious obstacle to the user who tries to access remote data from the supercomputing center.

The Internet connection gateway is usually very good, but it has limited bandwidth and hardware resources, so that the number of simultaneously clients is limited. The third obstacle arises when the gateway attempts to access data from storage areas or computing nodes. So there are three levels of data transport bottlenecks which paralyzes the user's ability to manipulate large numerical simulation results.

Other trends in the big data field can be formulated as follows: assess the volumes of structured, unstructured or semi-structured data, effective storage management and data extraction, real time processing of different types of large data resulted from different data sources, data classification based on the context, as well as real time evaluation and results analyze of data.

5. Conclusion

Indeed, big data overlaps HPC, especially when the customer is ready to spend as little time for data analysis.

Scientific computing is ascending dominated by the data. For an increasing number of scientific disciplines is desired to perform numerical simulations of concerned complex systems. Once these simulations are run, many of them become a "standard reference" that others around the world wanting to use it, or to compare. These large data sets generated on the largest supercomputers must therefore be stored and "published". This creates a series of new challenges.

Differences between HPC developments and trends in "data-intensive computing" are felt by scientists working on their scientific problems in "trenches" where it is becoming increasingly difficult to move and to analyze large data sets that they can generate. So, the lack of specialized databases can be a topical issue in high-performance computation.

The deeply scientific computing involves working with "deeply" large amounts of data, and analysis of these data requires methods that can work with huge datasets, and can find very subtle effects overlooked in previous measurements.

References

[1] Georg Hager, Gerhard Wellein, *Introduction to High Performance Computing for Scientists and Engineers*, ISBN 978-1-4398-1192-4, CRC Press Taylor & Francis Group, 2011.

[2] Victor Eijkhout, Edmond Chow, Robert van de Geijn, *Introduction to High Performance Scientific Computing*, 1st edition, 2010.

[3] Ilie Popa, *Introducere în calculul paralel și distribuit. Algoritmi*, Editura Matrix Rom, București, 2005.

[4] Blaise Barney, *Introduction to Parallel Computing*, Lawrence Livermore National Laboratory.

[5] Alfredo, Cuzzocrea, Song Il-Yeol, and C. Davis Karen, *Analytics over Large-Scale Multidimensional Data: The Big Data Revolution!*, DOLAP '11*:* Proceedings of the ACM 14th international workshop on Data Warehousing and OLAP, 101-103, 2011.

[6] Torsten Hoefler, Umar Kalim, Mark Gardner, Eric Brown, Wu-chun Feng, *Cascaded TCP: Big Throughput for Big Data Applications in Distributed HPC*, The International Conference of High Performance Computing, Storage and Analysis, Salt Lake City, Utah, 2012.

[7] Daniel M. Best, *Cyber Security's Big Data, Graphs, and Signature*, The International Conference of High Performance Computing, Storage and Analysis, Salt Lake City, Utah, 2012.

[8] Philip Russom, *Big Data Analytics*, TDWI best practices Report, 2011.

[9] Edd Dumbill, *Big Data*, Published Quarterly, 2013.

[10] Jun Zhang, Zhijian Wang, Shuren Zhu and Xiaoming Meng, *High Performance and High Availability Archived Stream System for Big Data*, Applied Mechanics and Materials, 10.4028/www.scientific.net/AMM.263-266.2792, China, 2012.

[11] Syed, J., Ghanem, M., Guo, Y, *Supporting scientific discovery processes in Discovery Net*, *Concurrency and Computation: Practice and Experience* 19 (2): 167, doi:10.1002/cpe.1049, 2007.
[12] Colby Ranger, Ramanan Raghuraman, Arun Penmetsa, Gary Bradski, Christos Kozyrakis, *Evaluating MapReduce for Multi-core and Multiprocessor Systems*, Computer Systems Laboratory, Stanford University, 2007.
[13] Jaliya Ekanayake, Hui Li, Bingjing Zhang, Thilina Gunarathne, Seung-Hee Bae, Judy Qiu, Geoffrey Fox, *Twister: A Runtime for Iterative MapReduce*, ACM 1-58113-000-0/00/0010, 2010.
[14] Jiong Xie, Shu Yin, Xiaojun Ruan, Zhiyang Ding, Yun Tian, James Majors, Adam Man-zanares, and Xiao Qin, *Improving MapReduce Performance through Data Placement in Heterogeneous Hadoop Clusters*, Proc. 19th Int'l Heterogeneity in Computing Workshop, Atlanta, Georgia, 2010.
[15] Tom White, *Hadoop: The Definitive Guide MapReduce for the Cloud*, O'Reilly Media, ISBN 0-596-52197-9, 2009.
[16] Nathaniel Rowe, *Big Data Trends in 2013. Can You Handle Your Unstructured Data?*, Aberdeen Group, Research Report, 2013.

Address:

- Prep. Drd. Eng. Florica Novăcescu, "Eftimie Murgu" University of Reşiţa, Piaţa Traian Vuia, nr. 1-4, 320085, Reşiţa, f.novacescu@uem.ro