A Review of Data Intensive Computing

Yanhui Wu [*1,2], Guoqing Li[1], Lizhe Wang [*1], Yan Ma[1], Joanna Kołodziej[3] and Samee U. Khan[4]
[1] Center for Earth Observation and Digital Earth, Chinese Academy of Sciences, Beijing 100094, P. R. China
[2] School of Information Science and Engineering, NingBo University, Ningbo 315211, P. R. China
[3] Institute of Computer Science; Cracow University of Technology, Cracow, Poland
[4] North Dakota State University, ND, USA

*Corresponding author: wu_yanhui@hotmail.com, Lizhe Wang@gmail.com

***Abstract*—Data intensive computing is a common research problem in science, industry and computer academia. In recent twenty years, the explosive growth of science data has appeared all over the world. Typical data intensive computing applications include Internet text data processing, scientific research data processing, large scale graph computing, inverse and perspective problems. Data intensive computing research faces challenges of scalability of massive data management and processing, integrated data processing technology, system management, new transaction demand, unstructured data processing, service mode, good fault tolerance and high availability. Appropriate data intensive computing programming model needs to be suitable for large scale data sets parallel computing, multiple virtual machine task scheduling and constructing new data intensive computing applications. As a typical data intensive application, aerosol inverse computing I/O data volume is analyzed. Although industry and academia has brought some methods for data intensive applications, scientific discovery problems with both data intensive and computation intensive features are still no appropriate solution at present.**

Keywords- data intensive computing, Big data, parallel processing

I. INTRODUCTION

Data intensive computing is a proposed research problem in science, industry, computer academia almost at the same time. Nature journal 2008 September specials "Big Data" has introduced the technical challenges of massive data, the existing solutions to technology, as well as the foreseeable future development direction from the Internet technologies, Internet economics, supercomputing, environmental science, biology and medicine and other aspects [1]. This marks that large volumes of data management and processing have become a core problem in scientific research, commercial activities, and daily life. Data intensive computing has become one of the most important research fields in Internet and computer science.

In recent twenty years, the explosive growth of science data has appeared all over the world and set off another wave of data. According IDC digital universe study, global data reaches 1.8ZB (1ZB=220GB) in 2011 [2]. In the past five years, global data grew 9 times. It is expected that by 2020, global data volume will reach 50 times in 2011. This data volume increase was partly due to scientific observation instrument capability continued to improve. Millions of various sensors throughout the world at all times on the universe, earth, ecological, biological, physical and chemical processes were observed and recorded. On the other hand, large numbers of computing equipments around the world were running on various scientific models for large-scale simulation calculation, at the same time to produce huge amounts of scientific data.

Pacific Northwest National Laboratory (PNNL) of US Department of Energy gives the definition: data intensive computing is capturing, managing, analyzing and understanding data at volumes and rates that push the frontiers of current technologies [3]. This definition has three meanings. Firstly, data intensive computing needs to deal with massive data that change rapidly. These data sources are often distributed, heterogeneous and unstructured. The current data volume currency is Petabyte even greater. Therefore, traditional database management system cannot meet the requirements of data intensive computing. Secondly, data intensive computing is different from the traditional high performance computing and scientific simulation calculation. It passes through the data acquisition, management, and then to the analysis and comprehension of the whole process. Data intensive computing is not a simple calculation process, but a whole process combination of traditional data management, high performance computing, data analysis, data mining and visualization. Finally, data intensive computing aims to promote the frontiers of current technologies and becomes a new generation of scientific research mode, a data centered scientific discovery mode.

II. DATA INTENSIVE COMPUTING APPLICATIONS

The content of Internet search application, such as Google, Baidu and other text search and data mining service, is not the only data intensive computing applications. With the "big data problem" increasingly development, individual, enterprise and industry will appear a large number of data and data dependent applications. Data intensive computing is one of the key technologies to meet these applications. Typical data intensive computing applications including:

A. Internet Text Data Processing

Internet consists of large amounts of data. As the world's largest database, it provides various types of data, also contains a variety of applications. Only Google search engines index webpage amount in 2008 has been more than one trillion. But this number is only a small portion of the web. At the same time, some webpage is behind the huge

database which is called “deep web”. Its size may be hundreds of times the webpage itself. In addition, business intelligence is the old problems of data management and data mining, but it has new technology requirements in the current era. With the application development, the original production data, enterprise internal technical documents, client email and other information will become a business intelligence analysis basis. It is also a massive data processing service on the types of complex data. Internet provides applications include web search, research assistant, enterprise search, customer relationship management. Because traditional data processing and management technologies are incapable for these massive data, Internet applications have become the most important driving force for data intensive computing technology development.

B. Data Intensive Scientific Discovery

Scientific research in the amount of data generated is also amazing. These data include types of experimental data, observational data, papers, technical reports, project reports, patents and scientific literature. Scientists and technologists need theoretical research and experiment according to these data. In 2009, Jim Gray has described e-Science research mode change in “The Fourth Paradigm – Data Intensive Scientific Discovery” [4]. Today, this description is a reality. On one hand, with the data acquisition technology progress, human’s ability in data production and data collection develop rapidly, these increasing quantity of data are becoming fundamental power of science research mode revolution. On the other hand, computing technology is in the wave of innovation, multi-core architecture, cloud computing and virtualization technologies bring revolutionary change for scientific discoveries. A public information research of e-science service platform is one of the keys to improve the level of scientific research. Therefore, scientific research has also become a data intensive computing important application. Pacific Northwest National Laboratory (PNNL) of US Department of Energy and the US National Natural Science Foundation (NSF) set up scientific research projects specially, in order to promote the development of scientific research.

C. Large Scale Graph

Graph is a commonly used data structure in computer science. Compared with the linear table and tree, it has more general representation capability. In the real world, graph can represent many traditional application scenes, such as selection of the optimal path, science and technology literature citation relations. Graph can also describe a lot of emerging applications, such as social network analysis, web semantic analysis. With the real world data scale growing, graph scale is also growing. How to efficiently process large scale graph data is becoming a new data intensive computing application. A single large scale graph usually contains billion or more vertices, such as calculation of PageRank algorithm used in search engine [5]. A webpage PageRank score can be calculated according to the hyperlinks relation between this webpage and other ones. By representing webpage as a graph vertex, the hyperlinks relation as a directed edge, we can get a large scale graph. In accordance with the adjacency list stored in the form of a graph of ten billion vertices and sixty billion edges, assuming that each vertex and edge storage space is 100 bytes, the whole graph storage space is more than 1TB. In view of large scale graph storage, update, search operations, its time and space overhead is far beyond the traditional graph data management capacity. How to efficiently manage large scale graph time and space overhead has become an urgent problem.

D. Inverse Problems

Earth sciences, oceanography, atmospheric sciences and seismology are common inverse problem application fields. Scientists often need to rebuild the 3D or 3D+t field based on the actual measurement or captured data, complemented by a complex model. Typical applications include “Ocean state estimation (MITgcm 4DVar)” [6], “Atmospheric data assimilation” and “Full 3D seismic tomography”. For example, in MIT ocean general circulation model (MITgcm) simulation experiment, it is necessary to solve nonlinear minimization problem, through with iterative calculation of the former point and the adjacent point equations. In the solving process, it is needed to save three layers check point data. In this inverse application, scientists will encounter the following performance problems:

- Adjoint sweep requires state data at every time step
- Multi-level checkpointing is required that stores state data on disk or flash at checkpoints
- Recomputes state data between checkpoints

Solving these performance problems need more RAM and better disk or flash I/O bandwidth.

III. DATA INTENSIVE COMPUTING APPLICATION CHARACTERISTICS

Data intensive computing is different from traditional high performance computing. It not only needs to store large scale data sets, high speed I/O transmission, but also the need for complex calculation, analysis and visualization of the result. Compared with traditional data management problems, data intensive computing has essential difference in application environment, data size and application requirements. Its features are embodied in the data, processing technology, complex application development and application mode.

A. Data intensive computing processing object is massive, rapid change, distributed and heterogeneous data.

Data size usually in PB level, so the traditional data storage, indexing technique is not applicable. For a computational task, time in getting data from a variety of sources is unbearable. Even very simple query operation, executive will be so complex. For example, in the 80MB/s sequential scanning 1 GB data, only 12.5 seconds, but in the sequential scanning of 1 PB data, it needs 145 days. Data in geographic distributed, heterogeneous model and representation bring some difficulties in data access and integration. A variety of different data sources lead to data

format diversification and heterogeneity. In addition to the traditional structured data, there are semi-structured and unstructured data. Data rapid dynamic change characteristics require that data processing must be real-time or have a strong timeliness. The traditional static database management technology is incapable of action.

B. *Computing has a variety of meanings.*

Unlike a computational intensive task processing, simple data block, parallel execution has been unable to meet the data intensive computing task demand. In many scientific research fields, such as Earth science, astronomy, they have the complex calculation model. The computational complexity for the local optimization and data management proposes new challenges. Data intensive computing includes search, query and other traditional data processing, but also includes smart processing, such as analysis and understanding. Note that, data intensive computing analysis and understanding is not just a single data analysis or data mining algorithm, the algorithm must be able to achieve efficiently in massive, distributed, heterogeneous data management platform. At the same time, the characteristic of the data makes it impossible to develop new algorithm for every data analysis and comprehension task ranging from storage to indexing. Therefore, data intensive computing need is associated with the storage and management platform, and combined with a high degree of flexibility and customization ability, easy to use search, query and analysis tools. By using these tools, users can construct complex data analysis or understanding application.

C. *Complex programming model*

As a general purpose computing system, data intensive computing system needs generic programming model and programming method. The current popular MapReduce programming model simplifies the programming work, improves data processing efficiency [7-8]. It is widely used in indexing system, machine learning, statistical machine translation and log analysis. But when the application calculation process is very complex, MapReduce model usually cannot get reasonable results. In science research inverse problem, this simple programming model does not converge because of its Map and Reduce decomposition method.

D. *Data intensive computing usually cannot be achieved locally.*

Because data intensive computing needs the massive storage, high performance computing platform, it usually cannot be achieved locally. Web service interface is an effective and natural way. Different form traditional high performance computing, user requirements may include data acquisition, preprocessing, data analysis process. In this complex procedure, data intensive computing service interface must provide full description function and favorable web service interaction between client and server.

IV. DATA INTENSIVE COMPUTING CHALLENGES

Just because of these new features in data intensive computing data management, the traditional data management technology is no longer applied. Data intensive computing research faces the following challenges:

A. *Keep scalability of massive data management and processing.*

In data intensive computing, the amount of data is growing faster than the growth rate of a single main memory or disk capacity. Traditional centralized or small scale distributed and parallel system data management technology is not suitable for data intensive computing. The corresponding storage and indexing technology must make fundamental changes. It may be to satisfy the rapid increasing applications in response time and throughput scalability requirements.

B. *Integrated data processing technology including search, query, analysis and mining features.*

Data intensive computing is often built in the related data based on a series of application. Any single data management techniques are not suitable for this environment. The traditional data management technology has supported from the initial database management, query processing to indexing, query, multidimensional analysis and even simple mining processing engine. But structured data query is still in the fundamental and key role. Other data processing techniques is usually through large objects, user defined types, functions and stored procedures to achieve. Traditional data management function and performance still cannot meet the need of data intensive computing.

C. *System management challenge.*

The traditional data management research has been aimed at a single application of safety management, user management, system configuration and achieved rich results. But data intensive computing service is often a series of application, sometimes even multi-tenant different application. At this time, application, user, session management model will be more complex, and more applications on the system configuration, load balancing, performance requirements are completely different.

D. *New transaction demand*

The new transaction demand is a fundamental problem. Traditional transaction model with atomic properties is designed for transactional data management. It is no longer fit for data intensive computing with a large number of analysis demands. At the same time, because widely used transaction management technologies based on the lock mechanism have very high cost in large scale distributed system implementation, so data intensive computing environment is not possible using these techniques. In data intensive computing environment, the traditional transaction processing technology both theoretical models and implementation techniques need to be reconsidered.

E. *Unstructured data processing*

The traditional data management techniques are based on the structured data. Even for semi-structured data, structure extraction is still a premise, complemented by special structure management technology. But in data intensive computing applications, data may be structured, semi-structured, or unstructured. Different applications may require different model for treatment of the same data.

F. *Service mode challenge*

Traditional data management service model is client/server model. In data intensive computing environment, the way of contact between client and service provider is usually the Internet. Their communication protocol is often based on HTTP. Due to the bandwidth and stability constraints, coupled with the requested service usually need to analyze massive data, so data intensive computing service mode must realize high speed response and incremental processing. This demand is absolutely different from traditional "all or nothing" data management requirement.

G. *Good fault tolerance*

Data intensive computing application must provide good fault tolerance, reduce the system maintenance and query cost, improve the system availability. In order to achieve favorable fault tolerance, application even may bear acceptable results in some degree of error. Compared with traditional transaction processing model, this is also a fundamental distinction.

V. PROGRAMMING MODEL

Data intensive computing is a parallel computing technology that processes large scale intensive data sets. Data intensive computing system end users do not need to care about parallel processing details. But, in order to allow program developers to fully utilize the data intensive computing convenience and availability, design and realization for suitable programming model is urgently needed. Appropriate data intensive computing programming model needs to meet the following conditions:

A. *Suitable for large scale data sets parallel computing*

The object of data intensive computing is massive, heterogeneous, rapidly changed data. The existing MPI (Message Passing Interface) model is not suitable for large scale data [9]. MPI is usually used for computationally intensive and less communication volume applications.

B. *Suitable for multiple virtual machine task scheduling*

One of data intensive computing supporting technologies is virtualization. Mapping from virtual machine to physical machine and task scheduling are very complex technologies. The existing openMP model does not meet this requirement [10]. It is a fine grained parallel and shared memory model.

C. *Suitable for constructing new data intensive computing applications*

According various application computing models, program developers should be able to construct new data intensive computing applications and provide rich experience for end user on the network.

The current representative programming models include MapReduce and Dryad [11]. Other models are their variants. Google MapReduce is used in large scale data sets (more than 1TB) parallel operation. The concepts of Map and Reduce are referenced from the functional programming language. MapReduce also references some features of vector programming language. It is greatly convenient for the programmers running their own program on a distributed system in the case of not understanding parallel programming. Map operation can be highly parallel. It is very useful in the area of high performance requirements and parallel computing. Reduce operation is relatively independent in large scale computing, and is very useful in highly parallel environment. However, when MapReduce faces complex computing process tasks, its efficiency is low. Even some complex computing tasks cannot get results. Microsoft Dryad model is a kind of parallel computing model based on pipeline computation. It uses a directed acyclic graph (DAG) to represent computational task decomposition. Each node in DAG represents a scheduled task. Through runtime system scheduling, Dryad can get good performance, and its scope of applications is superior to MapReduce. Dryad and MapReduce are not only programming models, they are also effective task scheduling models. Although Google discloses the principle of MapReduce, it is not open source. Hadoop open source project implements MapReduce, where HDFS is GFS open source implementation [12]. The Hadoop is widely used in industry and academia. Yahoo, Facebook and other Internet companies have been deployed Hadoop in their large production system. For example, Yahoo WebMap is a Hadoop large scale application [13]. For the data warehouse, Facebook has developed Hive system base on Hadoop [14].

These programming models have their own application domain and range. They are not suitable for all application problems. MapReduce provides a pair of key value abstraction. Although Dryad DAG abstraction is more flexible than MapReduce, it is not suitable for iterative calculation and large scale graph. From developer's view, a good programming model should have common structure and properties of programming languages, for example, iterative loop and abstract data structure.

VI. COMPUTING AND STORAGE COUPLED ARCHITECTURE

The biggest challenge in data intensive computing architecture is to meet I/O bandwidth requirement between computing system and storage system when processing large amount of data. In traditional high performance computer, data need a relatively long read and write path from memory to compute nodes. In order to complete computing

procedure, a number of read and write operations is essential. When the volume of data exceeds 1PB, storage system overhead of its architecture is unable to bear. It is difficult to meet the need of massive data processing. It is necessary to put forward a computing and storage coupled architecture. This architecture can efficiently shorten read and write path between computing and storage; reduce the data flow of intermediate links; cut down the system overhead; provide rapid data access capacity; realize storage balance between computing and read-write operation. Google's oriented Internet text search server cluster has been using this structure. In this architecture, data are distributed at each node that contains computational and storage function. Data computing procedure is executed at nodes that greatly reduce the system overhead of data movement. Of course, this kind of architecture is not fit for all scientific problems, there are many aspects need to be optimized and redesigned.

A. Data file system

Data intensive computing data objects are often not true relational data or structured data. These data may be text, webpage and XML documents with unstructured or semi-structured data. These data management and processing tasks are also referred to as NoSQL or non-schematic data management. These technologies have used object-oriented database achievements, especially object-oriented data mode management, as well as complex data type management. For unstructured data storage, general data management mode is file system. In business data analysis field, such as text search, there are Google file system GFS and Apache open source Hadoop file system HDFS. In scientific computing field with large scale data flow characteristics, such as global change, remote sensing image processing and high energy physics, these applications have high demand for scalability and concurrent I/O. In these fields, parallel file system occupies a dominant position, such as LustreFS file system and PVFS parallel virtual file system.

For structured data, primary storage mode is database and distributed table. In business analysis field, with rapid growth of data volume, traditional database system cannot satisfy users' requirements for storage system scalability. Many Internet enterprises prefer NoSQL system, such as Google's BigTable system, Apache open source HBase system. In scientific computing field, scientific database data format is different from business data file format. Scientific database needs time-space data format, uncertain and imprecise data format and image storage data format. In addition, scientific database also need to be able to manage multiple data sources, and provide a complex query operation, such as remote sensing image database with NASA (National Aeronautics and Space Administration) and ESA (European Space Agency) data.

B. Storage system

In Earth sciences, oceanography, atmospheric sciences and seismology, inverse or predictive problems are common applications. In global change aerosol computing, scientists in CEODE (Center for Earth Observation and Digital Earth, Chinese Academy of Sciences) inverse SYNTAM model by using TERRA and AQUA data system [15-16]. For the MODIS LIB first level data inverse problem, in resolution of one kilometer, the amount of data of covering the whole Asia for one month is 1.8TB. In MIT ocean general circulation model (MITgcm) simulation experiment, scientists need to solve nonlinear minimization problem through former point and adjacent point equations iterative calculation. In the solving process, storage system needs to save three layer checkpoint data. In the scientific computing, storage system will read massive data files and put them into memory. Storage system also needs to keep a number of temporary result data into memory. Data intensive computing has very high demand for system memory size and memory scalability in order to adapt to a variety of scientific applications. San Diego Supercomputer Center (SDSC) has realized a data intensive computer called Gordon and its prototype system called DASH [17]. Its design idea is to add flash drives and remote memory in traditional storage architecture. The effect of this two level memory is to ease the speed gap between memory and disk. Gordon system uses the ScaleMP's vSMP software to achieve memory unified addressing in every super node in order to realize global shared memory space. Gordon's architecture reaches two orders improvement of I/O access time.

C. Programming model

Programming model includes execution model, programming language, programming framework, run time system and development tool set. Data intensive computing puts forward new demands on parallel programming model, mainly from the balance between programming simplicity and performance optimization. First of all, programming model needs to provide basic semantic abstract representation in data intensive computing system, for example, different parallel levels expression, simple synchronous operation and resource management access. Secondly, important performance factors include large scale data movement, complex computing locality and multi-task scheduling operation. In order to take into account the simplicity and performance optimization, programming model needs to provide transparency for the above factors.

VII. AEROSOL INVERSE COMPUTING: A CASE STUDY

Aerosol inverse computing is an Earth atmosphere Science problem. It is also a data intensive computing application. Technically, an aerosol is a colloid suspension of fine solid particles or liquid droplets in a gas. Examples are clouds, and air pollution such as smog and smoke. Earth's atmosphere contains aerosols of various types and concentrations, including quantities of:

- natural inorganic materials: dust, smoke, sea salt, water droplets.
- natural organic materials: pollen, spores, bacteria.
- anthropogenic products of combustion such as: smoke, ashes dusts.

Aerosol inverse computing uses SYNTAM model algorithm which consists four non-line equations. SYNTAM model is a heuristic algorithm. In aerosol inverse computing,

it needs to produce a large number of intermediate results. These intermediate results are some temporary files which need to be read from or write to storage disks. Read and write operations occupy a large amount of computer I/O bandwidth.

In aerosol inverse computing experiments, we use TERRA and AQUA system MODIS LIB first level data. Table I shows I/O data volume in various computing conditions.

With 10 kilometers low resolution, the whole of China coverage area, double satellite I/O data volume is 360MB. With 1 kilometer high resolution, the whole of China and one month, double satellite I/O data volume is 1.1TB. With 1 kilometer high resolution, the whole of Asia and one month, double satellite I/O data volume is 1.4TB. The small portion of experimental time spent in SYNTAM equation calculation. At the same time, the most of the experimental time spent with the storage device I/O operation. I/O time overhead is far greater than the calculation time overhead. Aerosol inverse computing is a typical data intensive application.

Table I Aerosol inverse computing I/O data volume

Resolution (kilometers)	10 km	1 km	1 km	1 km
Coverage area	Whole of China	Whole of China 1 day	Whole of China 1 month	whole of Asia 1 month
Single satellite input data	140MB	14GB	420GB	700GB
Single satellite output data	40MB	4GB	120GB	200GB
Double satellite input data	280MB	28GB	840GB	1.4TB
Double satellite output data	80MB	8GB	240GB	400GB
Total I/O	360MB	36GB	1.1TB	1.8TB

VIII. CONCLUSION

Data intensive computing system key technologies are massive data storage and management. This paper discusses the data intensive computing applications, as well as architecture design challenges. Aiming at these urgent demands in data intensive computing, industry and academia has brought some solutions. But there are still following problems:

- Whether MapReduce and Dryad model are suitable for data intensive computing application needs further discussion.
- Architecture requirements for data intensive tasks and computational intensive tasks are different. For scientific discovery problems with both data intensive and computation intensive features, there is still no good solution at present.

Data intensive computing research prospect is very broad. Architecture designers can cope with above challenges by using hardware technology. Integrating graphics processor units (GPUs) or other coprocessors into system can efficiently improve system nodes processing capacity; solve both data intensive and computational intensive scientific discovery problems. In typical applications, according specific data characteristics, the design of specific programming model is a good choice.

ACKNOWLEDGMENT

Dr. Yanhui Wu's work in this paper is supported by "One-hundred talent program" of Chinese Academy of Sciences and the Scientific Research Fund of Zhejiang Provincial Education Department (Y201121189).

Dr. Lizhe Wang's work in this paper is supported by "One-hundred talent program" of Chinese Academy of Sciences.

REFERENCES

[1] Big Data. Nature Specials. September 2008.

[2] Extracting Value from Chaos. The 2011 Digital Universe Study. IDC. June 2011.

[3] Data Intensive Computing. http://dicomputing.pnnl.gov/index.html

[4] T.Hey, S.Tansley, and K.Tolle. The Fourth Paradigm --- data intensive scientific discovery. Microsoft Corporation. October 2009

[5] Brin Sergey, Page Larry. The anatomy of a large-scale hyper textual web search engine. Computer Networks and ISDN Systems, 1998, 30(1-7) : 107-117

[6] Patrick Heimbach, Chris Hill, Ralf Giering. An efficient exact adjoint of the parallel MIT General Circulation Model, generated via automatic differentiation. Future Generation Computer Systems. 21(8), October 2005, Pages 1356−1371

[7] Jeffrey Dean, Sanjay Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. Proceedings of the 6th symposium on operating system design and implementation. San Francisco: USENIX Association, 2004, 137-150

[8] http://zh.wikipedia.org/wiki/MapReduce

[9] The Message Passing Interface (MPI) standard. http://www.mcs.anl.gov/research/projects/mpi/

[10] The OpenMP API specification for parallel programming. http://openmp.org/wp/openmp-specifications/

[11] Michael Isard, Mihai Budiu, Yuan Yu, Andrew Birrell and Dennis Fetterly. Dryad: distributed data-parallel programs from sequential building blocks. Proceedings of the 2nd ACM SIGOPS/EuroSys European Conference on Computer Systems 2007

[12] Apache Hadoop. http://wiki.apache.org/hadoop/

[13] WebMap. http://developer.yahoo.com/blogs/hadoop/posts/2008/02/yahoo-worlds-largest-production-hadoop/

[14] Hive. http://hive.apache.org/

[15] TERRA satellite. http://en.wikipedia.org/wiki/Terra_(satellite)

[16] AQUA satellite. http://en.wikipedia.org/wiki/Aqua_(satellite)

[17] Gordon: Data intensive Supercomputing. http://www.sdsc.edu/supercomputing/gordon/