The Research of the Parallel Computing Development from the Angle of Cloud Computing

Zhensheng Peng*, Qingge Gong, Yanyu Duan and Yun Wang
Department of Information Engineering, Engineering University of PAP, Xi’an China
Email: *pzsyjsgldd@163.com

Abstract. Cloud computing is the development of parallel computing, distributed computing and grid computing. The development of cloud computing makes parallel computing come into people's lives. Firstly, this paper expounds the concept of cloud computing and introduces two several traditional parallel programming model. Secondly, it analyzes and studies the principles, advantages and disadvantages of OpenMP, MPI and Map Reduce respectively. Finally, it takes MPI, OpenMP models compared to Map Reduce from the angle of cloud computing. The results of this paper are intended to provide a reference for the development of parallel computing.

1. Introduction
Cloud computing is the development of distributed computing, parallel computing and grid computing. These concepts of computer sciences are the implementation of commercial [1]. Therefore, the cloud computing is not just a computing mode while it covers the service, market, technology and many other concepts. The service core of cloud computing is divided into three levels: takes infrastructure as a service—IaaS; takes platform as a service—PaaS and takes software as a service—SaaS (See Figure 1 below).

![Figure 1 Cloud services](image)

This article mainly analyzes technical from the aspect of cloud computing. To begin with, from technical perspective, after the division of physical host abilities through virtualization technology, the computing power of the host is no longer confined to a single physical host, but a virtual host based on segmentation which makes full use of the computing power of the single host. The second is a
distributed computing technology. It makes a single task into many subtasks, and then assigns to each host of the cluster which makes them parallel running coordinately with each other.

We can see that distributed computing is a technique which can aggregate the resources. It is the core concept of cloud computing essence from a macro level. The distributed parallel computing does not mean that there is no non-distributed parallel computing which is the focus of the discussion below.

2. The Overview of Parallel Computing

Parallel computation was appeared in the early 1960s. At that period, the transistor and core memory showed up. The processing unit became smaller and the memory became more compact and cheaper. The development of these technologies gave rise to the occurrence of parallel computer [2]. During this period, the parallel computers are mostly shared memory multiprocessor systems in small scale which called the mainframe.

For a long time, parallel computing has been developing rapidly in the field of high performance computing and the parallel computing architecture has also been in constant changing. The platform relied by parallel computing is called parallel computers which is composed of multiple nodes. The task is decomposed to various nodes and runs in parallel on each node. The early parallel node is not the completely independent host between each other. It more likes various modules of one host. In the 1990s, accompanying with the development of network equipment, MPI/PVM and the release of parallel programming standards, cluster architecture of parallel computer has been emerged. At this time, the node of hardware platform of parallel computing is the independent host. Modern architecture of parallel machine has Symmetrical Multi-Processing (SMP), Distributed Shared Memory (DSM) parallel machine, Massively Parallel Processor (MPP) and Parallel Vector Processor (PVP). The architecture of parallel machine are mainly used for high performance computing in specific areas. This is the foundation of cloud computing—the birth background of PC cluster of workstation [3]. Parallel computing has been brought into people's lives by PC cluster due to the rapid development of the microprocessor performance and network bandwidth.

3. The Implementation of Parallel Computing

The implementation of parallel computing has two levels [4]. First, it is the multi-core parallel computing and the multiple CPU in a single node. Although the parallel computing in single node is not the mainstream of implementation of cloud computing, multi-core is a very important aspect for a single node to improve the performance currently. Second, For cloud computing, the parallel between cluster nodes is emphasized more. At present, the nodes among cluster are usually connected by IP network. On the premise of enough bandwidth, each node is not restricted by geographical and space. So the parallel computing in cloud computing is called distributed parallel computing for many times. However, multi-CPU and multi-core are the trend of host development. So within a cluster, the parallel requirement of general level is existed such as the parallel between cluster nodes, multiprocessor within node internal and multi-core parallel. The parallel computing between nodes is equivalent to distributed parallel computing as we often said. Different from mature virtualization products, parallel computing has no mature products while only has the relatively mature tools. The realization of parallel computing relies on the familiar with business by developers and users as well as the correct and skilled use of parallel tool. There are three steps for parallel application software development and use.

(1) On the demand analysis stage, according to business characteristics, the task is divided into multiple tasks which can be executed in parallel as much as possible. This is the basis of the realization of parallel computing and it always be the most effective phase.

(2) During the designing and coding phase, parallel tools are used for program design. The parallel technology used in this stage is the most key technology which is not only difficulty but also not mature. Moreover, most of the technical personnel has not handle the technology applied at this stage at the moment.

(3) Parallel deployment architecture and tools deployment applications are used in deployment stage. Through the deployment, this stage can achieve load balancing of occupied equipment. Load balancing in essence is also belonging to parallel computing. The method used in common is the load
balancing device such as F5 switches, the ESB and etc. The technology involved in this stage is relatively mature which has been widely used in the field.

4. The Overview of Programming model of Parallel Computing
The parallel computing introduced in the second stage is narrow sense of parallel computing. This paper mainly analyzes the OpenMP and MPI in this stage [5], which is the parallel computing technology and Map Reduce which is the typical representative of load balancing technology. It will be introduced in the third phase respectively.

4.1 OpenMP(Open Multi-Processing)
Now from x86 to minicomputer, multi-core is the mainstream of CPU. For single-threaded program, multi-core processors are difficult to improve the process efficiency [6]. For multithreaded program, it can achieve the purpose of improving performance through simultaneous calculation by different cores. Obviously, the execution performance of multi-threaded program depends on the task decomposition. The compilation and maintenance of multi-threaded programs have more complicated than single thread. In general, the execution performance of multithreaded program is still much higher than single thread when the host is multiprocessor, multi-core or has the ability to execute multiple threads at the same time by adopting multithreaded programming technology.

The compilation of traditional multithreaded programming has two ways in general. The first is by calling the operating system function, development tools, multi-threaded control statements or control function. The second is to use P0SIX thread control library functions based on P0SIX’s standards and methods. These methods are all through the main thread to generate more sub-threads; then it separate work from main thread to each sub-thread for operation; in the end, it recycle and integrate the results from the main thread.

Since the use of this method, there have been two problems. First of all, the complexity of program development is increased a lot than single-threaded program. Second, the portability of first method also has problems. Third, if you want to have good extensibility, multithreading should not only create a fixed number of threads, but also should increase threads by the growth of CPU cores. Although the above methods can achieve good scalability, the control is rather complicated. The developers need to achieve load balance scheduling of Multithreading on multi-core processors by their own. It is very complex, but using OpenMP could solve the problem.

![OpenMP programming model of parallel computing](image)

**Figure 2** OpenMP programming model of parallel computing

OpenMP is a multi-threaded programming method. Its parallel granularity is in threading level (See Figure 2 below). The range of application is parallel computing within light engine, which can use multi CPU or multi-core better. OpenMP is a set of API and runtime library which can simplify...
multithreaded development within light engine in order to let the designers and developers focus on the main energy of understanding and breaking up the task. Therefore, the specific details can be accomplished by OpenMP. By using OpenMP, the programming efficiency is much higher than multithreaded programming of general function library or own function library of operating system. It is superior to POSIX standards and methods. OpenMP is available as shared storage standard. It is an application programming interface (API) designed for compiling parallel programs for multiprocessing machine including a compilation statements and function library used to support it.

OpenMP can multithread the program simply by pseudoinstruction. The simplest case, it can parallel processing the statement within the loop, only through adds one line of compiler directives which used to guide the compile program. OpenMP is originated in C/C++, but now it has Java development kit of OpenMP, which can be used in Java programming.

4.2 MPI (Message Passing Interface)
Although OpenMP is very good, it is only a parallel computing technology in light engine. No matter the parallel computing or cloud computing, the mainstream of the parallel computing is a parallel between the host rather than a single machine. Therefore, MPI is really the representation of parallel computing technology.

**Figure 3** MPICH programming model of parallel computing
MPI standard version 1.0 was born in May 1994. This standard puts forward a kind of function interface description based on message passing. At present, MPI has grown to version 2.0, and it has become an accepted standard of high-performance computing. MPI itself is not a specific implementation but a standard description. The famous concrete implementation and wide use of MPI is the MPICH which is completed by development team of American Argoone National Laboratories. MPICH is free software. It provides binding support for Fortran, C language and so on, which offers to developers in the form of function library (See Figure 3 below). In the program design, using MPI has a larger degree of freedom even it can be used to achieve Map Reduce function of Hadoop.

The way of message passing is a model widely used in parallel computers especially in distributed memory parallel computers. More than a decade, this model has made substantial progress in important calculated applications. In designing MPI, the goal is not to choose someone from existing message passing system but make full use of the most abstract characteristics among these systems.

The main advantage of establishing message passing standard is the portability and easy to use. The standardized benefit is particularly evident in higher level based on low-level messaging program and distributing storage communication environment constructed by abstract program. The definition of message passing standards can offer clearly defined library to manufacturers. Therefore, they can effectively realize these libraries or provide hardware support for library application in some cases. The expansibility is strengthened.

The powerful advantage of MPI is the portability in wide range. The MPI achieved in the upper deck of communication protocol between standard Unix processor will provide the portability for workstation PC system and different kinds of workstation network. This standard includes a point-to-point communication, set operations, process group, communication context, topology structure, binding with Fortran 77 and C language, environmental management and query and interface description.

4.3 Map Reduce

Map Reduce is a programming architecture based on cloud computing system proposed by Google and is a kind of parallel programming model which can deal with huge amounts of data for parallel computing of large-scale data set. The concept and main idea of "Map" and "Reduce" are both originated from functional programming language and vector programming language [7]. Due to that Map Reduce has common characteristics of functional and vector programming language, this programming model is particularly suited for searching, mining, analyzing and machine intelligence learning of unstructured and structured mass data. It can greatly simplify the difficulty of parallel programming.

![Figure 4 Map Reduce programming model of parallel computing](image-url)
Originally, some computational problems are very simple but we have to deal with complicated details, fault-tolerant mechanism as well as the data distribution policy, load balance and so many details which led to the complicated and difficult program code. Therefore, all these problems in computing applications can be wrapped in a library and be responsible by a system as a whole. Otherwise, the computing task of data is abstracted as Map and Reduce. The source of this abstraction is based on the Lisp and Map and Reduce concept of many kinds of functional language. Many computing tasks of data are related to Map. The Map is responsible for processing each logical Record among input data and produces a set of intermediate Key/Value pairs, and then performs Reduce operation to all intermediate results which have the same key. The intermediate results from Map should be merged and then generated the final results.

For large-scale data set operation, Map Reduce is managed and controlled by a main node [8]. The concrete computing tasks is accomplished by sending to various sub nodes from the master. In this way, Map Reduce achieves reliable execution and fault-tolerant mechanism of parallel tasks. In each time period, the master node will mark the working condition of various sub nodes. Once marked for death, all the tasks of this node will be assigned to other sub nodes for re-performance.

Compared with traditional distributed programming mode, Map Reduce has packaged the parallel processing, fault tolerance, localized calculation, load balance and other details and also provided a simple and powerful interface. By this interface, large data computing tasks can be concurrent and distributed automatically. During operation, system can solve the distribution problem of input data, which can across the program execution and scheduling of cluster nodes and deal with the failure of processing nodes and communications between the management nodes. Therefore, the compilation of parallel programs becomes easier. The programmers can use the resources of large distributed systems without the concurrent processing or distributed programming experience. Meanwhile, it has great economic efficiency by high performance cluster composed by general PC to achieve super high performance. And Map Reduce also has good generality. Many different types of computing applications can be solved simply by Map Reduce programming model.

5. The Comparison of Parallel Computing Programming Model

OpenMp is shared storage in threads level, which is only suitable for shared bus and memory, and SMP machines and DSM with single operating system image [9]. It has poor extensibility but high requirement on the machine. General dual-core machines are suitable for OpenMP because it can improve certain of operating speed. OpenMP has been widely used in scientific computing.

| Parallel computing programming model | OpenMP | MPI | Map Reduce |
|-------------------------------------|--------|-----|------------|
| Parallel granularity               | Threadin g level | Progressing level | Progressing level |
| Way of data distribution           | Implicit formulation | Explicit formulation | Explicit formulation |
| Expansibility                      | Bad | Good | Good |
| Failure of processing node         | No | No | Yes |
| Data storage and data processing position | Separate | Separate | Transfer from calculation to storage |
| DFS support                        | No | No | Yes |
| Real-time                          | Good | Good | Bad |
| Single demand processing           | Suit | Suit | Unsuited |
| Immediately read                   | Yes | Yes | No |
| The difficulty of learning         | Easy | More difficult | Easy |

MPI is parallel granularity in process level with distributed storage and appeared data distribution with good scalability. It is suitable for all kinds of machines but the programming model is
complicated: need to analyze program computing tasks, and map it to distributed process collection for calculation. Since MPI is based on message, there is no restriction on task partition and data partition [10]. Powerful flexibility increases the complexity. Since MPI is the process level. It needs to solve the problem of communication delays and unbalanced load to ensure efficiency. MPI program has complicated debugging and bad reliability.

By dispersing the large-scale operation on data set to network nodes, Map Reduce can achieve reliability [11]; each node could periodically report back the finished work and updated status. If a node keeping silence over a predetermined time interval, the master node will record the state as death and then assign the task to other nodes. Therefore, Map Reduce creates parallel computing model based on data partitioning which has powerful fault tolerance and is also very easy to learn. It can be used for distributing, sorting, web log analysis, index construction, document clustering and statistical machine learning.

6. Conclusion
Early parallel computing is mainly used for parallel computers. With the emergence of cluster, cloud computing brings the parallel computing into people's lives. Cloud computing is a kind of distributed parallel computing. Map Reduce is parallel programming model of cloud computing. The main difference between traditional MPI and OpenMP is that Map Reduce is technical delegate of load balancing. OpenMP and MPI are mainly used for concurrent design and coding.

OpenMP is a kind of multithreaded programming method and its parallel granularity is thread level. The range of application is parallel computing in light engine which can make good use of multi CPU or multi-core. And MPI is a message passing standard library and its parallel granularity is process level, which is a parallel between the host rather than a single machine. Therefore, MPI is the representative technology of parallel computing. Map Reduce is parallel programming model for concurrent processing huge amounts of data. It is also a kind of task scheduling model with high efficient at the same time. The greatest advantages of Map Reduce are in shielding the underlying and implementing the details which effectively reduce the parallel programming difficulty and improve programming efficiency.

OpenMP and MPI, as the traditional parallel computation models, OpenMP + MPI which is the hybrid parallel computing model provides a good parallel strategy for computer cluster consisted by multiple processors. As the distributed parallel programming mode of cloud computing, Map Reduce is a hotspot of current researches.

7. References
[1] Hayes, Brian. "Cloud computing." Communications of the Acm 51.7(2008):9-11.
[2] Nogi, Tatsuo. "Parallel computer." US, US4514807. 1985.
[3] Agbaria, Adnan, and R. Friedman. "Starfish: Fault-Tolerant Dynamic MPI Programs on Clusters of Workstations." Cluster Computing 6.3(2003):227-236.
[4] Grama, Ananth, et al. "Introduction to Parallel Computing." 22.2(2003):N12.
[5] Dorta, I., et al. "MPI and OpenMP implementations of branch-and-bound skeletons." Advances in Parallel Computing 13.04(2004):185-192.
[6] Michailidis, Panagiotis D., and K. G. Margaritis. "Open Multi Processing (OpenMP) of Gauss-Jordan Method for Solving System of Linear Equations." IEEE, International Conference on Computer and Information Technology IEEE Xplore, 2011:314-319.
[7] Thusoo, Ashish, et al. "Hive: a warehousing solution over a map reduce framework." Proceedings of the Vldb Endowment 2.2(2009):1626-1629.
[8] Yang, Hung Chih, et al. "Map reduce-merge: simplified relational data processing on large clusters." ACM SIGMOD International Conference on Management of Data ACM, 2007:1029-1040.
[9] Song, Fengguang, S. Moore, and J. Dongarra. "Analytical modeling and optimization for affinity based thread scheduling on multicore systems." (2009):1-10.
[10] Quinn, Michael J. "Parallel Programming in C with MPI and OpenMP." IEEE Distributed Systems Online 5.1(2004):7.1-7.3.
[11] Bersani, Marcello M, et al. "Efficient large-scale trace checking using Map Reduce." International Conference on Software Engineering ACM, 2015:888-898.