CSCanada
Management Science and Engineering
Vol. 6, No. 1, 2012, pp. 18-20
DOI:10.3968/j.mse.1913035X20120601.2333

ISSN 1913-0341 [Print]
ISSN 1913-035X [Online]
www.cscanada.net
www.cscanada.org

Discussion on the Parallelism and Classification of Cloud Computing Tasks

ZHENG Ying[1,*]

[1] Editorial Department of Journal, Inner Mongolia University for Nationalities, Tongliao, 028043, China.
[*]Corresponding author.

Received 12 January 2012; accepted 7 March 2012.

Abstract
Cloud computing basic platform can propose different requirements for storage and computation in face of different users during its operation. Whether multiple nodes are adapted to store and process one problem at the same time has to take the characteristics of parallelism of the task into consideration. The parallelism of serial program has not been found a universal model to resolute automatically up to date. For one computational process, loop structure is not only the one that appears most frequent in program design, but one structure that occupies system computation resource mainly, therefore, doing research on the problem of parallelism of loop program is one most import aspect. The article studies the correlation between computational data and data partitioning method when executing computation. From the perspective whether the task can be processed in parallel, cloud computing tasks are classified.

Key words: Cloud Computing; Parallelism; Dependency Analysis; Loop Structure

ZHENG Ying (2012). Discussion on the Parallelism and Classification of Cloud Computing Tasks. *Management Science and Engineering*, *6*(1), 18-20. Available from: URL: http://www.cscanada.net/index.php/mse/article/view/j.mse.1913035X20120601.2333 DOI: http://dx.doi.org/10.3968/j.mse.1913035X20120601.2333

INTRODUCTION
Cloud computing basic platform can propose different requirements for storage and computation in face of different users during its operation, including pure information storage requirement, mass data processing requirement, high-performance computing requirement, users' requirement for daily information processing and graphics, graphic image processing requirement. Whether multiple nodes are adapted to store and process one problem at the same time need us to take the characteristics of parallelism of the task into consideration [1]. How to make a serial program parallelization has not found a universal method up to date[2] [3]. Different problems often need different analysis, which is why software technology of parallel computing always lags behind the speed of the development of hardware. Meanwhile, because there is no universal method of parallelization, the problems that parallel program can solve are few. Only a few industries such as the fields of meteorology, military, petroleum and so on have specialized computing software. This causes that parallel computing technology has not been popularized in common users, and there is also the problem for cloud computing technology. Whether cloud computing will be successful finally depends on whether it meet the needs of common users. The diversity of common users' requirements make us have to find out other solutions in the situations that we have not found general rules for the parallelization of serial program.

Because the requirements of most common users for computing are very few relatively, for example, word processing, everyday work, Web application, which occupy very little storage resource and computation resource. When users request this kind of services, it is unnecessary for cloud computing system to start parallel computation to process them. According to our tests, adopting parallel computation when the amount of computation is not large does not reduce but increase computing time because of the time used for network communication. For the applications that need a great deal of computation and large storage capacity, we must consider the problems of distributed storage and

parallel computing, which also shows the core part that cloud computing provides computing and storage according to the demand. In order to better analyze this kind of problems, let us understand the technology of parallelization of serial program firstly.

1. THE PARALLELIZATION OF SERIAL PROGRAM

We have not found a universal model to resolute the parallelism of serial program automatically up to date [4]. For one computational process, loop structure is not only the one that appears most frequent in program design, but one structure that occupies system computation resource mainly. Frequently, during the running of a program, most of the time is spent on executing loop program. The increase of the layer number of loop and nest makes the time for computing in exponential rise, which becomes the main computing bottleneck during the running of a program. Therefore, doing research on the problem of parallelism of loop program is one most import aspect for us to do the job about the parallelism of program.

Loop Structure 1:

```
for (p=0;p<1000;p++)
 {
…;
}
```

Loop Structure 2:

```
for (p=0; p<1000;p++)
 for (q=0; q<1000; q++)
   for (r=0;r<1000; r++)
{
…;
}
```

Because the two loops above differ by two layers of nests, their execution time may differ from each other about one million times. Therefore, when doing tasks partition, to process loop structure is an important link in the process of parallelism. The correlation between data has important influence on parallelism.

2. ANALYSIS ON THE DEPENDENCY BETWEEN DATA

2.1 Dependency *and Parallelism Between Data*

When we do research on the parallelism of loop program we must analyze the dependency between computational data[5] [6] and thus discuss the method of data partition when executing computation. As long as there is dependency between data, we have to pay more attention in the process of the parallelism of program. Analyzing the dependency between statements in a computing program is called dependency analysis. The dependency between statements can be divided into three ones: Flow-dependency, Anti-dependency and Output-dependency. For example, if the statement C_2 is implemented after the statement C_1, in which x, y, z, p, q are variables, then three dependencies can be described as follows:

Data flow-dependency

If certain computational problem has the following two basic statements:

C_1: y=x-3*z

C_2: p=y*(2*q+1)

The left variable y of the statement C_1 also appears on the right variable set of the statement C_2, and C_2 has to take its computational value from C_1, then the statement C_2 is flow-dependent on C_1. Here only C_1 is implemented before C_2 and after the value y is calculated out, the computing task of the statement C_2 is finished.

Data Anti-dependency

If certain computational problem has the following two basic statements:

C_1: p=x*(4*y-2)

C_2: x=q-(5*z+1)

The execution orders of the two statements above are contrary to the situation of data flow-dependency. The right variable x of the statement C_1 also appears on the left variable set of the statement C_2, which is called the statement C_2 is anti-dependent on C_1. In the situation, before the statement C_1 does not take the value x, the statement C_2 cannot be executed, otherwise the value x will be changed.

Output-dependency

If certain computational problem has the following two basic statements:

C_1: y=x-y

C_2: y=x/p ($p \neq 0$)

In the situation, the left variables of C_1 and C_2 are same, so it is called the statement C_2 is output-dependent on C_1.

Considering three dependencies above, some principles about the program parallelism are obtained.

(1) The computation of the data without dependency can be executed in parallel or in serial. For example the statement C_1 and C_2 below, there is no any dependency between data, and we can execute the two computing statements at the same time.

C_1:y=4*x-8*z

C_2:p=2*q*(3*r-1)

(2) The computation of the data with flow-dependency or out-dependency cannot be executed in parallel.

(3) For the computation of the data with anti-dependency, as long as it is ensured that the value x in C_1 is read firstly and the value x obtained after computation in C_2 is written, it is allowed to be executed in parallel.

2.2 Dependent Distance and Dependent Direction Vector

Given that A and B are *n* integral subscript vectors in *n* layers of loops. Supposed that there is data dependency

between A and B, dependent distance vector D is defined as B-A; and dependent direction vector d is defined as d_i : if $a_i<b_i$,then d_i =”<”; if $a_i=b_i$,then d_i =”=”; if $a_i>b_i$,then d_i =”>”.

For example, there are three layers of loop nests below:

for $p=x_1$ *to* y_1 *do*
  for $q=x_2$ *to* y_2 *do*
    $r=x_3$ *to* y_3 *do*
    $E(p+2, q\ r-2) = E(p, q, r) + K$
endfor
endfor

A=*(p,q,r),* B=*(p+2,q,r-2)*, then the dependent distance vector between three-dimensional iterations of the array E is *D=(p+2-p,q-q,r-2-r)*=(2,0,-2) and dependent direction vector *d*=(<,=,>).

Dependent direction vector is very useful for computing the dependency between loop bodies. Their dependency is delivered through dependent direction vector but not through the loop outside of the mark “=”. Dependent distance vector indicts the practical distance of cycling alternation between visiting twice in the same storage unit. They pled directive roles in developing parallelism.

According to the analysis above, from the perspective whether the tasks can be processed in parallel, we classify some tasks in cloud computing into several types below:

Everyday work application: the amount of data and computation for this kind of tasks is little. Parallel computing method is not used and the data is not distributed stored.

SaaS application: the amount of data is large but its computation amount is little. Parallel computing method is not used and the data is distributed stored.

High-performance computation: the amount of data and computation for this kind of tasks is large. Parallel computing method is used and the data is distributed stored.

Information service: the amount of data is large but its computation amount is little. Parallel computing method is not used and the data is distributed stored.

Search engine: the amount of data and computation for this kind of tasks is large. Parallel computing method is used and the data is distributed stored.

CONCLUSION

In fact, not all data and tasks need distributed storage and computation in a cloud computing system. For everyday work application, the amount of data and computation for this kind of tasks is little. Parallel computing method is not used and the data is not distributed stored. For high-performance computation and search engine service, the amount of data and computation for this kind of tasks is large. Parallel computing method must be adopted and the data is also distributed stored to finish the task effectively. So each application resident in cloud computing system has to be classified by us when we introduce. Different kinds of application systems adapt to different processing methods during execution. Not all applications adopt distributed and parallel computing to process, which reduces the difficulty in system design greatly.

REFERENCES

[1] YANG Zhi-hao, ZHAO Tai-yin, YAO Xing-miao, LI Lei (2011). Private Cloud Computing System Realization Method Adaptable to Data and Computing Intensive Tasks (in Chinese). *Application Research of Computers, 2*, 621-624.

[2] CHEN Guoliang, AN Hong & CHEN Ling (2004). *Parallel Algorithms in Practice*. Beijing: Higher Education Press.

[3] LU Kai (2005). *High-Performance Computer Parallel File System*. Changsha: National Defense University Press.

[4] XIAO Yonghao, HUANG Qingnan (2011). Parallel Computation of Impact Problems Based on Block Data Structure (in Chinese). *Computer Aided Engineerin, 1*, 33-36.

[5] Calvin Lin & Lawrence Snyder (2009). *Parallel Program Design Principles*. Beijing: Mechanical Industry Press.

[6] ZHU Yongzhi, LI Bingfeng, SUN Tingting & LI Pei (2011). Research on Scalability of Parallel Computing System (in Chinese). *Computer Engineering and Applications, 21*, 47-49.