

Research on the design of distributed computing platform

Wei Guanghui

ChongQing College of Electronic Engineering, ChongQing,401331,China

qdbiq@163.com

Keywords: Distributed Computing Hierarchical Resource Classification

Abstract: the article puts forward the design scheme of a distributed computing platform, firstly, analysis and comparison of the common distributed computing model, is proposed based on three layer logic architecture of distributed computing system, puts forward the design scheme of establishing a computing node model of client server based on the three layer structure of the distributed computing platform framework then, the detailed design of the computing platform, client and server nodes. Finally, the distributed computing platform is implemented, and the work done by the platform is verified, which shows the feasibility and correctness of the computing platform.

1. Introduction

In engineering practice, a large number of complex calculations are often needed to achieve the goal. At the same time, the rapid development of Internet and the scale of data such as data sharing and information aggregation are also speeding up. Most of the processes involve massive data processing, and the computation complexity is quite high. These are very high requirements for the performance of the computer, but in fact, it is unrealistic to buy high performance supercomputers with high prices. So, put forward a scheme that can integrate the computer network resources, complex tasks or massive data of moderate decomposition, decomposition of these smaller calculated fragment allocation to multiple computer resources for distributed computing, and finally calculate the final result[1]. Such a scheme can fully tap the potential of idle processing in the network, and then efficiently and rationally utilize these resources to complete complex computing tasks, speed up the computation execution efficiency, and reflect the high parallelism of computation, which is an effective, practical and feasible work plan[2].

In today's business process and scientific research, we often need to deal with large-scale data, which is limited by the processing power of a single computer, which often leads to too long computation time and even can not produce computing results. This paper mainly discusses the calculation of large amount of computation, task decomposable and combined results, and introduces a distributed computing solution to achieve the goal.

2. Related Research

In foreign countries, distributed computing and storage platform mature infrastructure such as Google including GFS, Map/Reduce[3], Big, Table and Hadoop system, the core part is the data center, the formation of a new distributed computing platform application virtualization, SOA, WEB2.0 and a series of technology. In order to avoid a large amount of high price hardware investment, enterprises and individuals can also get higher computing power from the distributed computing platform through the Internet. ETI@home is a distributed computing project that uses

the Internet to deal with astronomical data with personal computers. The SETI@home program is usually running on the user's personal computer, usually in screen protection mode or after the platform mode. It takes advantage of the redundant processor resources and does not affect the user's normal use of the computer[4]. In China, the development of distributed cloud computing is also very fast. Since 2008, enterprises like IBM and Alibaba have set up cloud computing center and e-commerce cloud computing center before and after China, and the investment has reached hundreds of millions of RMB. Chinese universities established Cloud Computing Laboratory, from theory into practical application, especially in the construction of the data center application to a large number of theories of cloud computing, virtualization has experienced physical data center, data center, private cloud, hybrid cloud four stages of development. The government has actively promoted the development of cloud computing and the Internet of things. Since 2010 to 2011, a large number of cloud computing development plans have been released, providing strong support for the development of new industries such as cloud computing[5].

3. Design of distributed computing platform

According to the actual demand of the distributed computing platform, the following is the design direction of the platform, and the detailed design of the distributed computing platform is introduced in detail. It is described in detail from four aspects: the overall framework, the server side, the client side and the computing node.

3.1 design direction. In a distributed computing platform under the network condition, the need to achieve the following objectives: one is to design a universal distributed computing platform, and the platform of the single interface and easy to use, can use the cyber source more convenient; two is a distributed computing platform in the enterprise network, you can use the resources a lot of computing network is three; each computing node can participate in the use of computing resources in the network computing environment are widely distributed in the network; four is the data and computing policy files are independent of each other, and the distributed computing platform has a good adaptability for future expansion provides better space for further development and integration; five is the distributed computing platform is also calculated service platform performance and load balancing.

3.2 overall design. In order to ensure the rational use of resources and the flexibility of distributed computing platform, we first analyze the network structure and select the suitable network hierarchy before we build the overall architecture. According to the number of tasks and the number of computing resources, a single network hierarchy and a complex network hierarchy can be set up. The single network hierarchy is simple, and the complex network hierarchy is discussed here only.

The complex hierarchical network structure consists of the upstream server, the server side, the computing node and the client. The complex hierarchical network structure is shown in Figure 1-1. The functions of each part are as follows:

The client is the user that submits the task, and can submit a task processing application to the server, waiting for the server or the computing node to return the results.

The main function of the server is to accept task requests and assign tasks. The server acts as a connection point between the client and the computing node, and assumes a dual task. The user request is received, and the computation task is finally returned to the computation result. For the request of computing node, the attribute of computing resource is acquired and stored, the task scheduling algorithm is invoked, the sub tasks are allocated to each computing node, and the results returned by sub tasks are verified.

The main function of the computing node is to complete the subtask calculation request and feed back the results of the subtask to the server.

The overall performance of the platform is closely related to the independence of the task and the granularity of the task. That is, the more independent the task and the larger the granularity of the task, the better the overall performance of the platform. At the same time, in this case, a very obvious advantage is that the task is easy to divide, adapt to the multi process parallel computing, and facilitate the verification of the results. It can be concluded that the tasks adapted to the distributed computing platform are as follows:

- (1) the similarity of the task is high.
- (2) the task is easy to decompose and the granularity is larger.
- (3) the subtasks after decomposition are independent of each other, and the connection between tasks (Communication) is less.

The task flow of the composite network hierarchy is as follows:

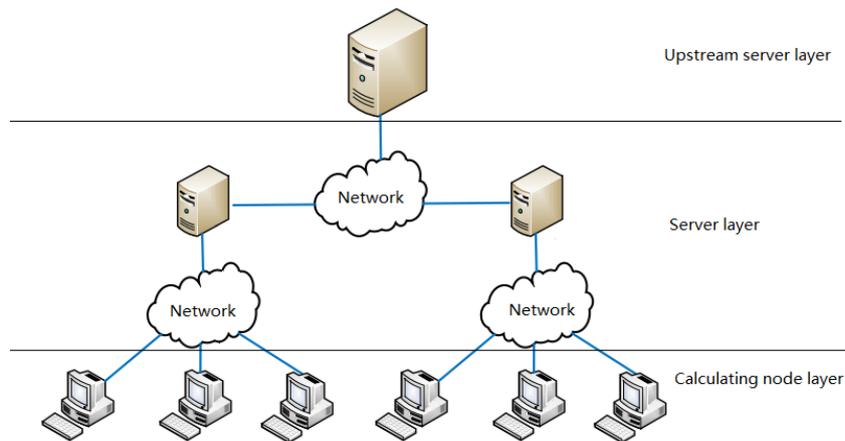


Figure 1 Composite Network Hierarchy

The functional design of 3.3 layers

3.3.1 client

The task decomposer is designed on the client side, and tasks are decomposed according to the attributes of task data. This way is easy to divide the computing tasks, and can specify a certain scope to decompose tasks. For example, for solving 1000 data processing problems, we can divide the time point attributes into 40 tasks, which are distributed at different computing nodes for parallel computing.

The parallel level of computing task is mainly considered from two aspects: hardware level, signal transmission and instruction parallelism. The purpose is to improve the efficiency of computer processors, and software level is mainly reflected in program level. According to the above description, the detailed granularity description is shown in Table 1.1.

Table.1 Granularity partition table

| Granularity classification | project | Parallel implementation |
|----------------------------|--------------------------|-------------------------|
| Small | Instruction transmission | Compiler |
| Medium | Page program | Programmer |
| crude | Whole program (process) | Programmer |

The distributed computing platform is mainly for coarse-grained tasks, computing task is a process of independent implementation of the solving process, which includes data module and algorithm module.

Data module: contains parameter files, organized by XML; contains specific data files, organized by a specific file structure.

Algorithm module: contains algorithm files, organized by DLL.

3.3.2 server side

Server nodes can be set up one or more. The server is divided into five modules: data management module, task scheduling module, data transmission module, monitoring module and business interface module.

Customer information contains four data tables of data management module in server, which are customer information table, calculate the project information table, task information table, user classification information table; contains three tables of data resource information, are nodes table, table, table load task solving.

The function interface module mainly implements various kinds of operations, that is, to complete various business logic tasks and to complete the management information service.

Task scheduling module is the main part of distributed computing platform service. Its main purpose is to allocate tasks to reasonable computing resources for task processing, and to return computing tasks completed by computing resources.

3.3.3 computing node

The main function of the computing node is to accept the assigned computing task and return the result of the calculation. This part will also involve task management and data transmission. At the same time, computing nodes need to feedback to server side, which can reflect attribute information of computing resource performance. Therefore, the computing nodes are designed into four parts: transmission part, task management part, computing task execution thread and computing resource attribute information feedback thread.

Transmission part: transfer the computing task and return the result of the calculation.

Task management part: management of the computing tasks assigned to the server, which can be used to suspend, restart, delete and other related operations for computing tasks.

Calculate the task execution thread: perform and complete the computing task assigned by the server.

Calculate the resource attribute information feedback thread: collect the properties that reflect the performance of the resource (main CPU and memory size), and feed this information back to the server. According to the attributes of each computing resource, the server classifying it in advance and assigning the scheduling task.

The design of the computing node is shown in Figure 2.

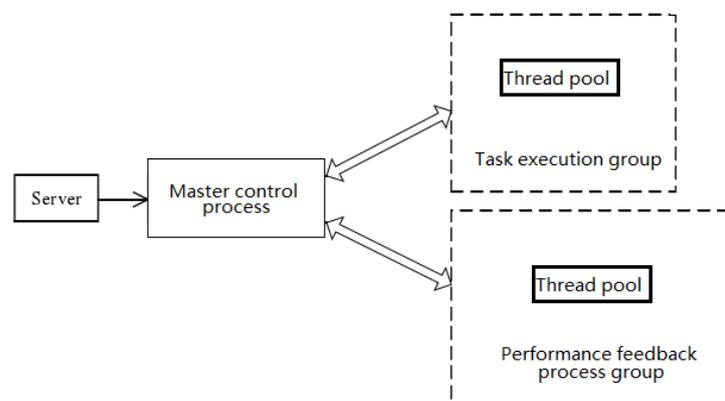


Figure 2 computing node design

4. Summary

According to the functional requirements of distributed computing platform, the design goal of distributed computing platform is put forward. Then the architecture design and task flow of distributed computing platform are mainly designed. Among them, the client, server and computing node are described and designed in detail.

Reference

- [1]Zhang Yu-Jun,CHEN Sheng-Yong,LING Hai-Feng,XU Xin-Li,Multi-Agent Based Distributed Computing Framework for Master-Slave Particle Swarms, *Journal of Software*, 2012, 23(11):3000—3008.
- [2] Czerwinski, Debski, Pietak, Distributed Agent-Based Platform for Large Scale Evolutionary Computations, 2011 International Conference on Complex ,Intelligent and Software Intensive Systems (CISIS), 2011: 462-466.
- [3]Wu Quan-Yuan, Network Computing Middleware, *Journal of Software*, 2013, 24(1):67-76.
- [4]Bawa, Sharma, Modified min-min heuristic for job scheduling based on QoS in Grid environment, *Information Management in the Knowledge Economy International Conference, India; IEEE*, 2013:166-171.
- [5] Bawa, Sharma, Modified min-min heuristic for job scheduling based on QoS in Grid environment,*Information Management in the Knowledge Economy International Conference, India; IEEE*, 2013: 166-171.