TELKOMNIKA, Vol. 11, No. 2, February 2013, pp. 935~940
ISSN: 2302-4046

Building Structured Peer-to-Peer Resource Sharing Platform Using Object Encapsulation Approach

Shufang ZHANG[1], Jun HAN [2]
[1,2]School of Mechanical and Electronic Engineering, Suzhou University, China
[2]School of Information Science and Technology, University of Science and Technology of China, Hefei, China
*Corresponding author, e-mail: zhshf_sztc@126.com, hanjun_sztc@126.com

Abstract

This paper designed and built a distributed hash table based computing resource sharing platform named OE-P2RSP. By employing Peer-to-Peer services, OE-P2RSP allowed users to submit jobs to be run in the system and to run jobs submitted by other users on any resources available over the Internet, essentially allowing a group of users to form an Ad hoc set of shared resources. OE-P2RSP is based on structured peer on network share, improves the communication mode of system by encapsulated object, object group, and physical proximity principle, and the use of physical proximity principle, and gathers computing resources better in the same research institution or enterprise within local area network. The experimental results obtained via simulations show that the system can reliably execute scientific applications on a widely distributed set of resources with good load balancing and low matchmaking cost and that OE-P2RSP has good efficiency, load balancing and scalability.

Keywords: *structured peer-to-peer, resource sharing, scalability, volunteer computing, Cycle stealing*

1. Introduction

Among peer-to-peer resource-sharing platforms, centralized unstructured peer-to-peer network Napster can not resist the attack from malicious nodes, so it is lake of strong robustness; distributed unstructured peer-to-peer network Gnutella or kaZaA have solved the single point of failure problem, but the expansion of the system is rather poor because of message flooding problem. Subsequent structured distributed Harsh table DHC solved the aforesaid problem [1-3].

In this paper we propose an Object Encapsulation Peer-to-Peer Resource Sharing Platform (OE-P2RSP), which is a system established on the application of DHC. OE-P2RSP is based on Pastry, and change from the node ID mapping to the object encapsulation based mapping; the difference between OE-P2RSP and other related projects is that OE-P2RSP is mainly used in the field of parallel and distributed computing, instead of data/file sharing, live video/on-demand field. Current peer-to-peer computing resources sharing projects always require the independence between main task and subtasks, so they mainly support the primary masterslave style of parallel computing [4, 5]. OE-P2RSP is a distributed object programming model, so it allows the direct or indirect communication between entities by remote method invocation, which makes OE-P2RSP also support distributed computing with data-dependent problems, and makes the scope of its application becomes sufficiently widespread. In this paper, we will use the principle of object moving and physical location adjacent to make OE-P2RSP gathering and sharing idle computing resources on the Internet more efficiently.

Then we test the performances of OE-P2RSP by simulations, and the results show that OE-P2RSP has good efficiency, load balancing and scalability.

2. The Architecture of OE-P2RSP

2.1. Platform Overview

OE-P2RSP is a non-centralized peer-to-peer computing resource sharing platform. It consists of a large-scale voluntary free node on the Internet, it also provides a distributed object

Received September 14, 2012; Revised December 31, 2012; Accepted January 15, 2013

encapsulation based programming model which can perform a parallel distributed computing by the inner nodes communication mechanism.

The difference between OE-P2RSP other existing computing resources sharing platform is that it can support communication between parallel sub-task operations, and the architecture is different from the voluntary computing platform using C/S model, has the characteristics of self-organization, non-concentration, scalability and large-scale. By using the direct communication between the peer entities, OE-P2RSP can solve more parallel computing problems, such as the distributed data-dependent traveling salesman problem and the distributed genetic algorithm. So we can fully excavate and make full use of the free resources on Internet.

2.2. Pastry Network

OE-P2RSP platform is based on distributed hash table Pastry network, and the Pastry network consists of k-bit network nodes which has the only NodeID voluntarily. NodeID distribution is of non-centralized manner and need to across the address space to achieve load balancing.

Network message can be addressed to the address of space of the NodeID, when passing messages the route model of Pastry is O(LogN) jump, the node has the location informations of both physical location and virtual location. Physical location indicates in physical network the minimum delay to reach the nearest location. If good physical location data is lacked, the system will assume the two nodes close to the IP address are the physical locations. Virtual locations are identified by telling the difference between NodeID. The nodes maintain the massage of routing information and each node stores a small group of nodes those are close to its physical location (such as neighbors) and its virtual location (such as its leaf set). The most important routing mechanism is that the touting table contains O(LogN) entities which allow each node to forward pass messages to a node and the forwarding node of at least 1 bit is relatively close to the final destination node.

2.3. Addressing Objects and Programming Model

OE-P2RSP can create objects which are located on Pastry network that consists of voluntary machines. The communication between these objects are achieved by remote method invocation RMI (remote method calls) and realized by the RMI technology of Java and .NET technology of Microsoft. Even so, when we use OE-P2RSP we need not to specify a specific server but to choose a machine from voluntary machines. In spite of the joining and leaving of nodes are very frequent, in OE-P2RSP the voluntary machine that can control given remote object can be changed over time.

When a object is created, it will be assigned an object ID which is similar with the nodes in Pastry, then a construction message will be sent to the object ID and inform the nearest node that accept the message, then the node will create a remote object instance. At this node the information need to be informed is just the object ID. If another node joins the network, it will be assigned a node ID (also relatively close to the object), and then the object will be transferred to the new node. OE-P2RSP can complete the transfer automatically; this is similar with the log and checkpoints which can improve the reliability of system.

3. Improvement of Communication Mode

OE-P2RSP has some improvement than in Pastry in communication mode. In addition to use object encapsulates, OE-P2RSP improve the system's communication mode by object group and the physical proximity principle.

3.1. Allocation of Object Identifier

Object ID does not require a centralized generation. A single customer or node need to activate the collection of relative remote objects frequently. And the remote object needs not random communication, so it's a good manner to use the adjacent neighbor nodes. By using this non-random way, we can make sure that the objects can be assigned in the same physical host even if the numbers of objects are far more than that of entities. Since there is frequent joining and exiting of nodes, so the mapping and location information of objects will not be affected. So this proximity principle of object ID is one-dimensional and the communication mode between objects is folded into a one-dimensional space by developers. The programming interface API which is to assign object ID inform system to launch a unified object distribution function. Start Layout method requires a parameter N (N is the number of objects in objects group), in this way system will cover all the address space correctly. After this call, the next N objects will be activated automatically, and the object ID will be created according to this mode. Code is as follows:

```
Island [] islands=new Island [numIslands];
OE-P2RSP Channel.Current.StartLayout(numIslands);
For (int i=0; i< numIslands; i++)
Islands[i] =new Island (i);
```

Then the system will use different random offset address in order to avoid the conflict between the original object ID and the current object ID. In section 4.1 the experimental result indicates the communication efficiency and load balance performance by using this method to assign object ID.

3.2. Object Groups

As previously mentioned by adding the object communication opportunities in the same machine we can improve the efficiency of the platform, but we can not make sure at the same machine all the time. In this section we use the method of object groups to make the objects in the same object group communicate each other frequently and ensure they are assigned to the same node. We can realize object groups by adding extra bits to the object ID and lengthen length of node ID, converting the object ID to a decimal, not a integer. The objects with the same integer part will be mapped to the same node.

If a group of objects were assigned to the same integer part, this group of objects will be mapped to a single machine, even if the network has a large number of other free nodes. The actual physical nodes can also change over time. But the objects are grouped together, improving the parallelism. Another optimization approach is to encapsulate these object groups to a single remote object container, the message can be transmitted forward through the object. The advantage of this method is that each object in the group maintains a private addressable remote client.

Distance proximity principle. By adding the communicate opportunities of two objects in the same machine we can improve the efficiency of the platform, but some objects may be positioned to different machines. If they can not locate on the same machine, the physical proximity principle in this section ensures that they are in physically similar positions. To accomplish this function OE-P2RSP continue to allocate object ID according to the method in section 3.1, but we change the node ID so that the nodes which match well will be very close physically. If two objects communicate frequently, they will be allocated well-matched object ID, and these two objects may be controlled at the same node or close nodes, meaning close physical locations.

By using IP address when we describe node ID we can ensure well-matched node ID are close physically. The shortcoming of this method is that sometimes node ID will not distribute to the entire address space. To achieve this technique, OE-P2RSP uses initialized node ID and current node ID. Initialized node ID is generated by the IP address. The current node ID is used by the node. When normal communication massage is routed by the ID of current node, the joining message will be routed to the node that is initialized nearest.

Though the routing table of each node is designed to the standard route format, it can not be used in initialized node ID. So we depend on the order of the current node ID, so do the initialized node ID. This means we can use the leaf set to route messages, at the worst condition we need N/2 network jumps. In fact joining message is completed by connecting the nearest physical node. Leaving node will be placed to a more isolated place, rather than a place where a large number of nodes gather. Pastry network is designed to recover automatically from a single node, so the platform can reduce the possibility of local error recover by change song aspects of Pastry network. This treatment must be a balance between reliability and efficiency.

 ISSN: 2302-4046

4. Experiments and Performance Analysis

4.1. Construct the Simulator

We construct a Pastry network with simulator and simulate a set of objects in one-dimensional nearest neighbor configuration space of periodic communications. OE-P2RSP can improve the efficiency and performance of Pastry network, and the improvement can be analyzed by the following two performance metrics. The first indicator is the communication efficiency, which indicates the ratio of sending messages between objects when network hops are using message. It indicates the number of hops every message needs to cross network. If the ratio is relatively large, it means some messages can be transferred without network. The simulator doesn't consider the physical communication price between different nodes, because Pastry network has considered the physical position price of nodes when they are selecting by network hops.

As mentioned earlier OE-P2RSP has been trying to reduce the number of hops which network require. So the second character is load balance rate, which shows the balance of performance of objects when getting through the network. It is calculated by the standard deviation statistics of the number of objects on each node. If the load balancing ratio is relatively small, it indicates that each node has about the same load.

4.2. The Results by Using Object ID

The first test shows the performance of Pastry network which is improved by OE-P2RSP platform using object package and object ID. Object ID can be a good way to improve the efficiency of network communications, especially in the condition when object number is more than node number, load balancing rate can be improved obviously and along with the increase of nodes, the number of network is increasing, the rate of improvement of load balancing effect is reducing.

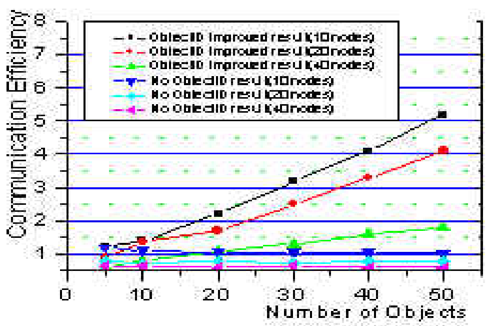


Figure 1. Communication Efficiency Improvement by Object ID Package

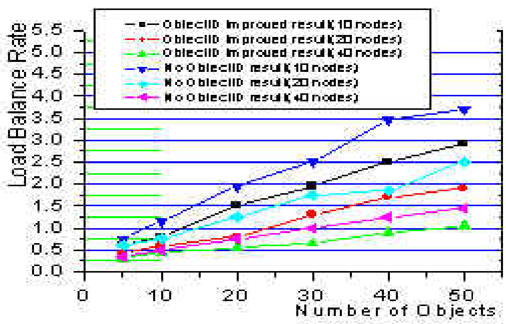
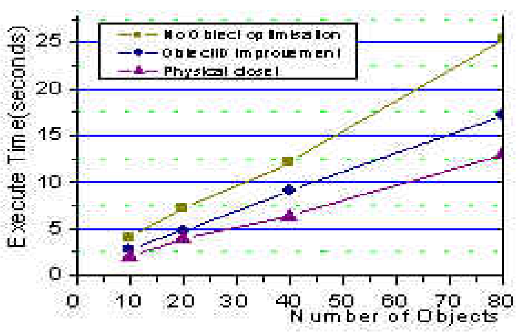
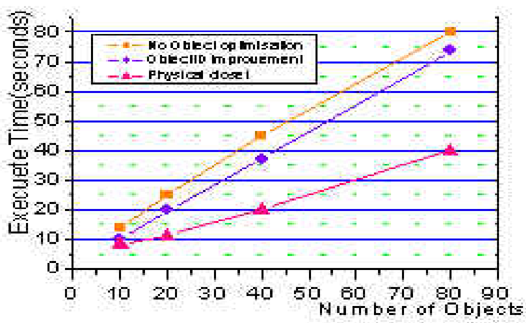


Figure 2. Load Balance Rate Improvement by Object ID Package



(a) The Experimental Results Running Short Process Length



(b) The Experimental Results Running Long Process Length

Figure 3. The Experimental Results

Figure 1 and Figure 2 show the improvement by object ID. Dotted line represents the result which is not optimized; the solid line shows the optimal results. Each line expressed an independent test result when different numbers of Voluntary machines add in OE-P2RSP platform.

The results of this experiment using distance proximity. The second experiment tested the proximity principle of Pastry network performance improvements. As we have described, the principles can be a good method to improve load balancing. The improvement will get weak and the communication efficiency will slightly decreases with the increase of nodes. This improvement is directly related to the rate of load balancing, the network can use the smallest number of hops to route messages to the destination node.

Figure 3 show the experimental results, which dotted line represents the result without using the proximity principle, the solid line represents the result when use proximity principle. Each line is that a different number of voluntary drive to join OE-P2RSP platform independent test results. Each line expressed an independent test result when different numbers of voluntary machines add in OE-P2RSP platform.

Tests run a total of three times, respectively, using normal strategy, enhanced object ID strategies and physical proximity principle. Test using a different number of objects, the length of the different processing steps executed repeatedly. In the same situation tests are run several times and we use the average of the results. Test also set a different message size, message size base 1KB, although this setting does not affect the results much sense. We use two processing length, the first is about 0.12 seconds to complete each object, the second for each object is about 0.39 seconds to complete. Figure 3(a) shows the experimental results running short process length. Figure 3(b) shows the experimental results running long process length.

5. Related Works

At home and abroad there has been a lot of use of structured peer networks to build the project for sharing computing resources. Robust, decentralized, scalability, efficiency are important performance indicators of these systems. Resource management and scheduling, routing node are key technologies of these platforms. Most of the distributed hash table can relatively simply and efficiently discover network resources to meet operational needs, but they are mainly for file sharing of data, instead of sharing computing resources [6, 7].

Similar projects to this article include CompuP2P [15], Chord [2], CCOF [16], WaveGrid [17], Pastry [1], etc. CompuP2P Uses non-centralized network topologies to manage the resource-sharing system under peer network environment. where all nodes in Chord logical ring way and organized into a processor market, idle processor resources can trade on the network, But CompuP2P has some scalability issues, because it does not limit the number of nodes on the market or balance market transactions by way of balancing the load. CCOF project in this regard achieved great success, it is in a dynamic peer network environment, and you can submit via the internet to users efficiently schedule jobs to run on idle computing resources. However Wave Grid project is the follow-up study of CCOF project. In Wave Grid area a time-sensitive application layer overlay network is built, which is based on CAN network. Wave Grid also describes using idle computing resources to execute jobs submitted by the user process, but Wave Grid host is totally dependent on the availability of the resource requirements of the job, the effect of load balancing and the availability of this project system is relatively poor.

Article [18] uses Pastry way to obtain the idle computing resources with physical proximity, and uses resources availability network protocol to limit the flooding, but this also leads to poor search inefficiency on and idle resources in long distance. Article [14] also describes the mechanism to find free computing resource in sharing platform; it uses the property of their resources to form Chord identifier, and completes the search by allowing descriptor.

OE-P2RSP in this article belongs to an improved platform of distributed hash tables, mainly used to support and complete communication between objects directly or indirectly, it is for the sharing of computing resources, and thefts idle processor cycles on the network, and improves the performance of the original Pastry network by package objects, object groups, and physical proximity and other technologies.

6. Conclusions

This paper presents and describes OE-P2RSP. OE-P2RSP is based on structured peer on network share, improves the communication mode of system by encapsulated object, object group, and physical proximity principle, and the use of physical proximity principle, and gathers computing resources better in the same research institution or enterprise within local area network.

References

[1] Jabri MA. Dealing with Grid-Computing Authorization Using Identity-Based Certificateless Proxy Signature. Cluster, Cloud and Grid Computing (CCGrid). *IEEE/ACM International Symposium on, Newport Beach*. 2011; 544 - 553.

[2] Chandrasekhar Boyapati, Barbara Liskov, Liuba Shrira. *Ownership Types for Object Encapsulation*. POPL 2003: the 30th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages. New Orleans. Louisiana. USA. 2003; 38: 213-223.

[3] Zang shufang, Hanjun. An Object Encapsulation Approach for P2P Resource Sharing Platform. *Lecture Notes in Electrical Engineering*. 2012; 28: 265-273.

[4] Herry Z Kotta, Kalvein Rantelobo, Silvester Tena, Gregorius Klau. Wireless Sensor Network for Landslide Monitoring in Nusa Tenggara Timur. *TELKOMNIKA.* 2011; 1(9): 9-18.

[5] Angelino C, Canonic M, Guazzone M. *Peer-to-peer desktop grids in the real world: the share grid project*. Proc of the 8th IEEE International Symposium on Cluster Computing and the Grid. Lyon. 2008: 609-614.

[6] Fox G, Pallickara S, Rao Xi. Towards enabling peer-to-peer grids. *Journal of Concurrency and Computation: Practice and Experience.* 2005;17:1109-1131.

[7] Nazareno Andradea, Francisco Brasileiroa, Walfredo Cirnea, Miranda Mowbrayb. Automatic grid assembly by promoting collaboration in peer-to-peer grids. *Journal of Parallel and Distributed Computing*. 2007; 67(8): 957-966.

[8] Morrison JP, Coghlan B, Shearer A, Foley S, Power D, Perrott R. WEBCOM-G: A Candidate Middleware for Grid-Ireland. *International Journal of High Performance Computing Applications.* 2006; 20(3): 409-422.

[9] KA Hawick, HA James, Craig J. A Distributed High Performance Computing Environment. *Lecture Notes in Computer Science*. 1998; 1401: 598 - 606.

[10] Zhang Shufang, Hanjun. Using the Reputation Score Management for Constructing FPFS System. *Lecture Notes in Electrical Engineering*. 2012; 28: 739-746.

[11] Jim Madsen, Jikku Venkat. Is distributed computing commercially viable. *Internet World*. 2001; 7(7): 14.

[12] Zhou Jingtao, Mo Rong, Wang Mingwei. *SDDG: Semantic desktop data grid. Information Sciences and Interaction Sciences (ICIS)*, 2010 3rd International Conference on Chengdu. 2010: 240-245.

[13] Reich C, Bubendorfer K, Buyya R*. An Autonomic Peer-to-Peer Architecture for Hosting Stateful Web Services.* Proc of the 8th IEEE International Symposium on Cluster Computing and the Grid. Lyon. 2008: 250-257.

[14] Arna Fariza, Afrida Helen, Ardinur Mahyuzar. WAP Based An Alternative Solution for Traffic Transportation Problem in Sidoarjo Surrounding Area Using AHP. *TELKOMNIKA*. 2009; 7(2): 137-142.

[15] Di Sheng, Wang, Cho-Li. Conflict-*minimizing dynamic load balancing for P2P desktop Grid. Grid Computing (GRID)*, 2010 11th IEEE/ACM International Conference on. Brussels. 2010: 137–144.

[16] Korpela Eric J. SETI@home, BOINC, and Volunteer Distributed Computing. *Annual Review of Earth & Planetary Sciences*. 2012; 40(1): 69-87.

[17] Wei-Chang Yeh, Shang Chia Wei. Economic-based resource allocation for reliable Grid-computing service based on Grid Bank. *Future Generation Computer Systems.* 2012; 28: 989-1002.

[18] Domenico Talia, Paolo Trunfio, Jingdi Zeng. *Peer-to-Peer Models for Resource Discovery in Large-Scale Grids: A Scalable Architecture. High performance computing for computational science , VECPAR*. Rio de Janeiro, Brazil. 2006: 66-78.