International Journal of Internet, Broadcasting and Communication Vol.7 No.1 42-48 (2015)
http://dx.doi.org/10.7236/IJIBC.2015.7.1.42

IJIBC 15-1-9

A Design of P2P Cloud System Using The Super P2P

Yean-Woo Jung[1], Seongsoo Cho[2], Jong-Yong Lee[3], KyeDong Jeong[3*]

[1]*Department of Information System, Kwangwoon University Graduate School of Information Contents, Korea*
blueshark2011@gmail.com

[2]*Institute of Information Science and Engineering Research, Mokpo National University, Korea*
css66@mokpo.ac.kr

[3,3*]*Department of Culture, Kwangwoon University, 20 Kwangwoon-ro, Seoul, Korea*

{jyonglee, gdchung}@kw.ac.kr

Abstract

Generally, the current commercial cloud system is hosted by the centralization large datacenter. However, the other clouding forms existed and one of them is the P2P cloud. The P2P Cloud is a distributed systems, is freedom to join and leave and is possible to provide the IaaS. Commonly, P2P Cloud System is based on Pure-P2P System. Since a number of connection paths exist, it has a high reliability and easily scalability of the network. However, there are disadvantages of the slow rate of route because of looking up the all peers in order to search for the peer. In this paper, it proposes the P2P cloud system based on super-peer P2P. Super-peer P2P system has fast routing time, because of searching for cluster unit and it also can has high reliability and high scalability. We designs the super Peer cloud service and proposes the system model and Resource Allocation Algorithm for IaaS in Super peer P2P environment.

Keywords: *P2P, Super Peer, Cloud, mobile agent,P2P cloud*

1. Introduction

A universal cloud system is the centralized cloud system which is based on central server in a large. In addition it is existed in other forms of cloud computing, it is P2P cloud. Generally the P2P cloud is fully distributed systems. The P2P cloud is not suited on a system that requires short communication delay as web service. But P2P cloud can provide such as the IaaS(Infrastructure as a Service), PaaS(Platform as a Service),

Manuscript Received: Jan. 11, 2015 / Revised: Jan. 15, 2015 / Accepted: Feb. 10, 2015
Corresponding Author: jyonglee@kw.ac.kr
Tel: +82-2-940-5288 , Fax: +82-2-940-5677
Department of Culture, Kwangwoon University, Korea

SaaS(Software as a Service) and etc and it can guarantee QoS(Quality of Service). Especially, P2P cloud can possible to cooperation with other device using the mobile agent system.[1] The mobile agent is the core technology of P2P system. The mobile agent is executed using the system resource of the other devices. Because it can collaborate with other devices using the mobile agent in the P2P system, it is advantageous to the distributed processing.[1, 4] In this paper, we designed the super peer P2P cloud system based on super peer P2P system. The super peer P2P is a fusion system of the pure p2p and hybrid p2p system. The super peer P2P builds a cluster with a number of super peer and clients and the multiple clusters is acts like peer of pure p2p. If one of super peer is disconnected, the other super peer replaces the roles. And the client may have a small load because the super peer is responsible for handling all query within a cluster.

The searching method of a super peer P2P is divided to internal and external search. The internal search is searching based on the peer management information that it is possessed by the super peer and the external search is that it is requested to the super peer of the other cluster and get it's result. Because the internal and external search, a super peer is able to search a cluster unit and it can be faster searching than the typical P2P search.[3, 5, 6]

In this paper, we propose for the resource allocation algorithm for providing IaaS, the required resources search algorithm for retrieving the required resources in policy and the search algorithms running internal search. This paper is organized as follows. The section 2 describes the resources for the management of the system model and a peer. It is proposed the resource allocation algorithm for providing service IaaS in chapter 3. The section 4 expresses the conclusions and future work.

2. System Model

In this paper, the super peer P2P cloud system(SPCS) designs on the basis of the super peer p2p system. The SPCS provides IaaS by allocating system resources of the other peers to the user. The structure of the SPCS is shown in Fig. 1.

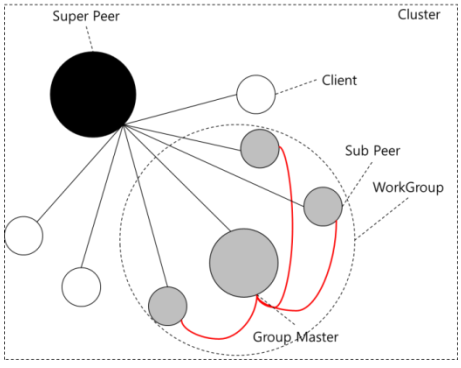

Figure 1. System model

The Super peer is the administrator that manages the group and peer inside the cluster. It process the query for all the sub-peer requests. The client is a user who participated in the SPSS and is the general peer. The Group Master is the client that is created a working group for allocating to sub peer from super peer. The Sub Peer is Client that is allocated in the group master. Through it provides the idle resources of sub peer to the

group master, the SPCS provides a service IaaS for the user. To provide IaaS form the SPCS, the SPCS allocates Sub Peer to create workgroups around the peer service requestor.

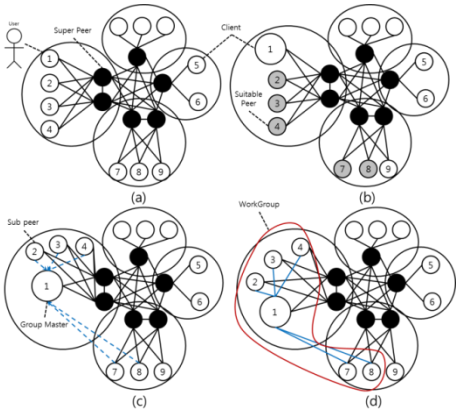

Figure 2. Process of Workgroup Generating

The workgroup provides IaaS to the user. It is generate by connection of a number of the sub peer and the group master.In the workgroup, Group Master send mobile agent to sub peer, the sub peer execute mobile agent and return back result, and group master with sub peer can cooperate. The sub peer of workgroup is connected with group master, but at the same time it still connected with super peer of self. Therefore allocation of processing is group master's work, but super peer is still doing connection management of sub peer.It shows in Fig. 2 to create the workgroup process.

(a) The user is requested allocation resource for using IaaS.

(b) The super peer allocates the client that used the user's service.

(c) The allocated clients is connecting to user being group master.

(d) Onceall connectionsarecompleted, the workgroup generating is over.

The resource allocation algorithm for generating workgroup is described in the Section 3.

2.1 Profile and List

To manage the peer, the super peer periodically checks over the lower-peer's connection and performance and creates the profile and list. Also, some of the generated profile and list is share with lower peer. The type of profiles and lists are Client Profile, Client State List, Group List, Group Member Profile, Group Member State List, Found Peer List and Selected Peer List.

```
- <CP-Profile>
  - <Host-Info>
      <IPAD>192.168.2.97</IPAD>
      <PortNum>8080</PortNum>
    </Host-Info>
  - <Net-Info>
      <ID>08</ID>
      <TimeStamp>20140603212432</TimeStamp>
    </Net-Info>
  - <System-Info>
    - <Resource>
        <CPU>2.5</CPU>
        <TotalR>4096</TotalR>
        <AvgR>2047</AvgR>
      </Resource>
    </System-Info>
</CP-Profile>
```

Figure 3. Client Profile

```
- <CP-List>
  - <CP>
      <ID>00</ID>
      <status>0</status>
      <PRank>1</PRank>
    </CP>
  - <CP>
      <ID>03</ID>
      <status>1</status>
      <PRank>2</PRank>
    </CP>
  - <CP>
      <ID>1F</ID>
      <status>0</status>
      <PRank>4</PRank>
    </CP>
  - <CP>
      <ID>07</ID>
      <status>0</status>
      <PRank>5</PRank>
    </CP>
</CP-List>
```

Figure 4. Client State List

The client profile has the IP address, port number, ID, timestamp and system information. The system

information indicates idle resource.

The client state list has the Client's ID, Status and performance ranking. The ID is connecting with Client profile's ID.

```
- <Group-List>
  - <Group>
    - <Group-Info>
        <Group-ID>03</Group-ID>
        <GroupMaster-ID>1F</GroupMaster-ID>
      </Group-Info>
    - <Group-State>
        <Member-Count>7</Member-Count>
        <Group-Status>1</Group-Status>
      </Group-State>
    </Group>
</Group-List>
```

Figure 5. Group Lis

```
- <GM-Profile>
  - <Host-Info>
      <IPAD>192.168.7.93</IPAD>
      <PortNum>8080</PortNum>
    </Host-Info>
  - <Net-Info>
      <ID>02</ID>
    </Net-Info>
  - <Group-Info>
      <ID>00</ID>
      <Group-ID>03</Group-ID>
    </Group-Info>
  - <Super-Peer-Info>
      <IPAD>192.168.6.102</IPAD>
      <PortNum>8080</PortNum>
    </Super-Peer-Info>
</GM-Profile>
```

Figure 6. Group Member Profile

```
- <GroupMemberStateList>
  - <GM>
      <ID>00</ID>
      <Group-ID>03</Group-ID>
      <Connect-Status>1</Connect-Status>
      <Task-Status>1</Task-Status>
    </GM>
  - <GM>
      <ID>01</ID>
      <Group-ID>03</Group-ID>
      <Connect-Status>1</Connect-Status>
      <Task-Status>0</Task-Status>
    </GM>
  - <GM>
      <ID>02</ID>
      <Group-ID>03</Group-ID>
      <Connect-Status>0</Connect-Status>
      <Task-Status>0</Task-Status>
    </GM>
</GroupMemberStateList>
```

Figure 7. Group Member State List

The group list is a list to manage for group in super peer. It has a group ID, group master's ID, a number of members and status of the group.

The group member profile is profile about the member of group. It is stored by Super peer and Group Master and it has the IP address, port number, ID on the network, ID within the group, group ID, Super peer's IP address and port number.

The group member state list is stored by Super peer and group master. It involve about member's ID, group ID, connection status and working status.

```
- <FoundPeer-List>
    <ID>1F</ID>
  - <Peer>
    - <Host-Info>
        <IPAD>192.168.3.5</IPAD>
        <PortNum>8080</PortNum>
      </Host-Info>
    - <Routing-Info>
        <Passed-SuperPeer>3</Passed-SuperPeer>
        <SuperPeer-IPAD>192.168.3.1</SuperPeer-IPAD>
        <SupePeer-PortNum>8080</SupePeer-PortNum>
      </Routing-Info>
    - <System-Info>
        <CPU>3.5</CPU>
        <TotalR>4000</TotalR>
        <AvgR>2300</AvgR>
      </System-Info>
    </Peer>
</FoundPeer-List>
```

Figure 8. Found Peer List

```
- <SelectedPeer-List>
    <ID>1F</ID>
  - <Peer>
    - <Host-Info>
        <IPAD>192.168.3.5</IPAD>
        <PortNum>8080</PortNum>
      </Host-Info>
    - <Routing-Info>
        <Passed-SuperPeer>3</Passed-SuperPeer>
        <SuperPeer-IPAD>192.168.3.1</SuperPeer-IPAD>
        <SupePeer-PortNum>8080</SupePeer-PortNum>
      </Routing-Info>
    - <System-Info>
        <CPU>3.5</CPU>
        <TotalR>4000</TotalR>
        <AvgR>2300</AvgR>
      </System-Info>
    </Peer>
</SelectedPeer-List>
```

Figure 9. Selected Peer List

The found peer list is created by the super peer. For Searching over, It is recorded peers that meet the search criteria. It has IP address, port number, number of passing super peer, IP address of super peer, port number of super peer and resource information. The super peer sends the found peer list to the client.

The selected peer list is created by the client. It records the available peers and has the same information of the found peer list.

3. Resource Allocation Algorithm

In the SPCS, it forms a working group by assigning a peer to the user to provide the IaaS. For allocating the peer for user, after the requesting user must search for a resource, the peer to be assigned searchs on the basis of the retrieved resource. Figure 10 is the resource allocation algorithm. It is used as a parameter such as the Rcode and ID of the requester client resource allocation. The Rcode is represented the type of required resource in received message.

```
1.  RequiredResource RR = getRequiredResource(Rcode);
2.  Boolean AllocationState=1;
3.  int T;
4.  While(AllocationState)
5.  {
6.          if(SearchPeer(Client_ID, RR.RPC, RR.RCP, RR.RAR))
7.          {
8.                  String Group_ID;
9.                  T = getTime();
10.                 While(true)
11.                 {
12.                         if (CalcTimeM(T) >= 5)SPLTimeOver(Client_ID); return;}
13.                         if(CheckRecevieSPL(Client_ID))
14.                         {
15.                                 Group_ID = CreateGroupInfo(receive_select_result(Client_ID));
16.                                 end loop;
17.                         }
18.                 }
19.                 T = getTime();
20.                 While(true)
21.                 {
22.                         if (CalcTimeM(T) >= 2){ Allocate_Failed(Client_ID, Group_ID);}
23.                         if(CheckRecveieAWS(Client_ID))
24.                         {
25.                                 end loop;
26.                         }
27.                 }
28.                 SendGroupAccessCommand(Groud_ID);
29.                 int Count = getGroupCount(Group_ID);
30.                 T = getTime();
31.                 While(true)
32.                 {
33.                         if (CalcTimeM(T) >= 2){ Allocate_Failed(Client_ID, Group_ID);}
34.                         if(ReceiveSignal(Clientt_ID))
35.                         {
36.                                 Count--;
37.                         }
38.                         if(Count==0){end loop;}
39.                 }
40.                 UpdateGroupList(Group_ID);
41.                 AllocationState = 0;
42.         }
43.         else
44.         {
45.                 PeerSearchFailed(Cliend_ID);
46.                 end loop;
47.         }
48. }
49. return;
```

Figure 10. Resource Allocation Algorithm

From 1 to 3, it is to define the object for the resource allocation part. The RR is object for saveing the required resource, it is described in section 3.1. AllocationState is Bool variable for checking the resource allocation and T is variable for timer. From 4 to 47 is loop, it is repeated until AllocationState is be 0. The line 6 is to represent the searching peer using the RR and Clint_IDparameter, it is described in section 3.2. From 10 to 20, the client is waited to receive the Selected Peer List (SPL) and receives the SPL, the group information created and updated on the basis of SPL, it sends a profile and group member status group members list to Client. From 20 to 27, it is the waiting parts to receive the Peer Connection Waiting Signal from the client. The line 28 is send access command to group member. It checks to have completed all of the connected group members from 31 until 39. From 40 to 41, it is update the group list and completes work.

3.1 Required Resource Search Algorithm

For peer searching, it is searched the required resource in policy by type of Resource. Figure 11 is Required Resource Search Algorithm by using the Rcode as a parameter.

```
1.  RequiredResource RR = new RequiredResource();
2.  int i, n;
3.  PeerAllocatePolicy PAP = ReadPeerAllocatePolicy();
4.  n = PAP.GetColumnCount();
5.  For(i=0 to n)
6.  {
7.              if(PAP.getCode(i) == Rcode)
8.              {
9.                              RR.RPC = PAP.getPC(i);
10.                             RR.RCP = PAP.getCP(i);
11.                             RR.RAR = PAP.getAR(i);
12.                             end roof;
13.             }
14.             else{ i++; }
15. }
16. Return RR;
```

Figure 11. Required Resource Search Algorithm

From 1 to 4, it is to define the object and record the data for search policy. The RR object is to output the requested resource. The PAP is object for save Peer Allocate Policy, n stores the PAP's item count. From 5 to 15, RR stores the object which it finds a match Rcode in the policy. In 16 RR objects recovered.

3.2 Peer Search Algorithm

The super peer completes the Searching Required Resource and it runs the internal and external search using a parameter as the resource information needed by searching. The peer search algorithm that runs take over the resources necessary information is shown in Figure 12. The search operation used the parameters-RCP, the Client ID, RPC and RAR. It is about the functions and objects used in the algorithm are described below.

```
1.  int i n;
2.  ClistStateList CSL = ReadClientList();
3.  FoundPeerList FPL = new FoundPeerList();
4.  n = CSL.GetColumnCount();
5.  For(i=0 to n)
6.  {
7.              String ID = CSL.getID(i);
8.              int S = CSL.getStatus(i);
9.              if(S == 0)
10.             {
11.                             ClientProfile CP = ReadClientProfile(ID);
12.                             if (CP.CPU >= RCP And CP.AR >= RAR)
13.                             {
14.                                             FPL.AddClient(CP), RPC--, i++;
15.                                             if RPC ==0 { end roof; }
16.                             }
17.                             else { i++; }
18.             }
19.             else{ i++; }
20. }
21. if (RPC =! 0 )
22. {
23.             RequestSearchPeer(Client_ID, RPC, RCP, RAR);
24.             T = GetTime();
25.             while(true)
26.             {
27.                             if (getreturn())
28.                             {
29.                                             n = receive_search_result();
30.                                             RPF = RPC- n;
31.                                             if (CalcTimeM(T) >= 3) {end roof;}
32.                             }
33.             }
34.             if (PRC > 0) {PeerSearchFailed(); return;}
35. }
36. SendFPL(FPL); return;
```

Figure 12. Peer Search Algorithm

From 1 to 4, this part is to define the object for the internal search and record the data. The CSL is object for save Client State List. The FPL is the object for creating a list of peer which it is sent to the client and stores the number of items in the list of clients in the n. The loop for the internal search is, from 05 until 20. It is to search the profile of peer that is usable in the Client Status List. From 21 to 35, whether or not the search is completed, continuous testing and external search run. To Take place at around 3 minutes outside

of search, the search terminates after external peer to allocate all the failed transmission if not found. 36 transmits the information of the peer to the client is detected.

4. Conclusion

In this paper, we design the super peer P2P cloud system based on super peer P2P system. The designed system does routing by the unit of cluster. It is faster than peer unit routing of pure p2p cloud system. The super peer P2P is high reliability and high scalability. Because it executes network query, the lower load applied to the client. In this paper, it proposes the resource allocation algorithm. The SPCS creates the workgroup using the resource allocation algorithm and workgroup can provide the IaaS. The workgroup is an effective model for tasks that require a dispersion treatment, as a large-scale computation.

References

[1] OzalpBabaoglu, Moreno Marzolla, Michele Tamburini. Design and Implementation of a P2P Cloud System. SAC '12 Proceedings of the 27th Annual ACM Symposium on Applied Computing. Pages 412 417. ACM New York, NY, USA. 2012

[2] Beverly Yang, Hector Garcia Molina. Designing a Super Peer Network. Computer Science Department, Stanford University. 2003

[3] Min, Su-Hong, Joanne Holliday, and Dong-Sub Cho. "Optimal super-peer selection for large-scale p2p system." Hybrid Information Technology, 2006. ICHIT'06. International Conference on. Vol. 2. IEEE, 2006.

[4] Danny B. Lange, Mitsuru Oshima. Mobile agents with Java: The Aglet API. World Wide Web. 1998, Volume 1, Issue 3, pp 111 121

[5] Merz, Peter, Matthias Priebe, and Steffen Wolf. "A simulation framework for distributed super-peer topology construction using network coordinates." Parallel, Distributed and Network-Based Processing, 2008. PDP 2008. 16th Euromicro Conference on. IEEE, 2008.

[6] Liu, Meirong, ErkkiHarjula, and Mika Ylianttila. "An efficient selection algorithm for building a super-peer overlay." Journal of Internet Services and Applications 4.1 (2013): 1-12.