International Journal of Future Generation Communication and Networking
Vol. 9, No. 9 (2016), pp. 215-230
http://dx.doi.org/10.14257/ijfgcn.2016.9.9.19

# Computational Nodes Location using Spatial Points Clustering in P2P Network System

Zhi Zeng[*]

*Department of Computer Science, Huizhou University, Huizhou 516007, China*
*zengzhi@hzu.edu.cn*

## *Abstract*

*In order to implement big data access and process with high efficiency, an algorithm of nodes location was proposed according to the state of computable resources. In this paper, we first describe and map the computational resource with javascript object notation(JSON) in P2P network system. Regarding the computational nodes as spatial points, then we present a generalized euclid distance(GED) model using the method of spatial points clustering. Through this model, the computational nodes can partition into multiple sub-group upon the characteristic attributes. After that, we calculate the spatial distance and attribute distance by spatial geometric model of global network positioning(GNP), ultimately implement the computable nodes location with efficiency, to provide the basis of load balance, especially in cloud computing. Experimental results show that our method not only can significantly improve system performance, also in accuracy of nodes location.*

***Keywords**: Computational Nodes Location; JSON; GED; Spatial Points Clustering; P2P*

## 1. Introduction

With the progress of internet technology, the data volume of massive information shows increasing rapidly. For instance, some big data image attains the level of Gigabyte. Therefore, in order to solve the contradiction between data increasing and rapid processing, it is obviously urgent to research and develop a feasible solution against efficient processing on massive information for big data so as to provide better service for users. In this paper we reference the characteristics of grid technology which can collective the heterogeneous resources from different areas. Due to the dynamic increasing resources in grid can provide the parallel ability by using single compute unit to complete super large scale computing. So the key issue is how to search and match the pertinent computable resource in current distributed network to implement the problem of nodes location. In general, the resource which mentioned herein mainly contains data resources, network resources and services resources. Typically, the efficient operation of the system can not separate from the resources provisioning, while the destination of resource provisioning is to make benefit maximization under the condition of using minimal infrastructures. That is to say, to guarantee resource utilization maximum. Traditional centralized computing has many limitations in the environment which is the different interests and the presence of private information, so it is of great value and significance to study the coordinated problem of distributed resources across administrative domains. With the continuous expansion of the network size, and make full use of resources on the computational node to enhance the performance of the system is a significative question which must face a research on selection of computable resource with high effectively, especially in distributed computing environment. In our study the

[*] Corresponding author.

ISSN: 2233-7857 IJFGCN

destination of selection on computable resource is to save the time of data migration according to network distance between computational nodes using GED model, so a method of spatial points clustering was presented. For speeding the data transmission in the distributed computing environment, a self-organizing P2P system was introduced. The whole process contains the following 4 steps.

1. description of computable resource with JSON;
2. mapping the computational nodes as a spatial points;
3. points clustering by dynamic querying attribute;
4. computational nodes selection by GED model.

The rest of this paper is structured in the following manner. Section 2 introduces a mechanism of unified description for computable resources. Section 3 presents an intelligent querying and matching algorithm to dynamic computable resources in P2P system. Section 4 presents an algorithm on selection of computational nodes. Section 5 shows some experiments to verify the efficiency of querying algorithm and selection algorithm base on the generalized distance model using spatial points clustering. Section 6 discusses related work, and finally, makes a conclusion about the whole paper.

## 2. Description for Computable Resources by JSON

Traditionally the idea to achieve interoperability in cloud computing is that encapsulate all resources as services under SOA (Service-Oriented Architecture)[1]. SOA has the advantage of management on computable resource. The computable resource commonly in this paper specifies data resources and hardware resources and software resources on runtime platform, so exploring the problem of the description for computable resource is of important. Considering the dynamic characteristic of resources, in the following we begin with the description in distributed network, analyze the method using JSON for dynamic resource indexing and routing, ultimately for dynamic resource message transmission and change providing the basis on P2P network system.

There are a lot of publicities about the XML advantages, such as cross language and platform. However, unless in particular Web applications, developers often broken mind by parsing XML. No matter what in the server or client side, it often leads to complex code and low efficiency. In fact, for most Web applications, they don't need complex XML to transmit data, scalability of XML has little advantage. Compared with the return XML and parse it, return to the HTML fragment greatly reduces the complexity of the system, but also the lack of flexibility. Therefore, many AJAX applications even directly return the HTML fragments to build dynamic Web pages. While as a lightweight data-interchange format, JSON is very suitable for interaction between server and JavaScript. In general, JSON is built on two structures. One constructer is a collection of name/value pairs, the other is an ordered list of values [2]. Same as XML, JSON is also string data structure base on pure text. So we can transmit a simple String, Number, Boolean data, also an array or a complex Object in internet. Regardless XML has been used widely in industry, and it has the feature of self-aware and self-discovery in the distributed system [3]. Beside that, we need take JSON play the advantages which is fast parsing speed and simple data expression. Especially in field of Ajax, future development must be let XML giving away and replaced by JSON. When the Ajax should become a Ajaj(Asynchronous JavaScript and JSON). Figure1 describe a XML and JSON format of computational resource about one node. From the figure we can see the XML describes the stateful and characteristics two parts. State part records dynamic attributes which web services existed in node, such as available percentage of CPU, number of memory usage, *etc*. Characteristic part descripts its static attributes of node, such as the core number of CPU, frequency of CPU and total number of memory.

```
<definition xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
 <resource-info node-name="node_Id">
        <node-state element-identifier="node-state-root">
         <free-disk="" free-memory="" os-name=""
          os-version="" os-virtualisation="" processes-count="" processes-running=""
           cpu-usage-percent="" memory-free-percent="" element-identifier="stateElementId" />
                ...Other Node State Elements...
        </node-state>

       <node-characteristics >
         <core-count="" core-speed="" core-speed-unit="" hardware-architecture=""
        total-disk="" total-disk-unit="" total-memory="" total-memory-unit=""/>
     ...Other Node Characteristic Elements...
       </node-characteristics>
   </resource-info>
```

(a) XML Structure

```
[
 {
   free-disk:" ", free-memory:" ",
  os-name:" ",  os-version:" ",os-virtualisation:" ",
  processes-count:" ", processes-running:" ",
  cpu-usage-percent:" ", memory-free-percent:" ",
  element-identifier:" "
  },
 {
core-count:" ", core-speed:" ",
  core-speed-unit:" ", hardware-architecture:" ",
  total-disk:" ", total-disk-unit:" ", total-memory:" ",
  total-memory-unit:" "
   }
]
```

(b) JSON Structure

**Figure 1. Two Different Structures of Computational Resource on a Node**

After the dynamic resource described by JSON in the distributed system, how to guarantee occurrence of dynamic resource for searching the pertinent resource to complete job execution is a base prerequisite. So it is needed research a method to supervise and update the state of resource and service in period, we reference the mechanic using connector to inspect the state in real time [4], make the info of dynamic resource in current node update and publish the stateful attributes. Therefore, we construct an inner self-organizing un-constructed P2P system to speed the data transmission and the dynamic attributes updating and publishing in our computing environment.

## 3. Querying Algorithm for Computable Resource in P2P System

### 3.1. Bitmap index & Spanning-Tree Generation

There are many type of resources in the distributed network system, also each type of resource contains many attributes. In general, traditional Distributed Hash Table(DHT) is efficiency for single dimension query. However, resources in distributed system always be multi-dimensional, including service type, processor speed, available memory, bandwidth and operation type, *etc*. Given an example for dynamic query as follows:

Instance OSType = "WinSrv2012" && Host CPU Utilization < "80%" && Host Processor Cores = "4" && Host Processors Speed = "2.3 GHz" && Host Location = "Asia"

The above multi-dimensional query can match JSON expression of each node resource through mapping mechanism. Bitmap is an index mechanism which describes attribute and corresponding info of resource. Commonly attribute can divide quantification and qualification two parts. Dynamic attribute which recorded by JSON is quantification, the static attribute belongs to qualification. In the latter sections, no matter what it is quantification or qualification, we calculate them by means of mapping method from multi-dimensional query to one dimensional range [5]. So we take a range query in d dimension expression as following:

$$[A_{1\min,}A_{1\max}],[A_{2\min,}A_{2\max}],\cdots,[A_{d\min,}A_{d\max}].$$

Towards the above example, one dimensional attribute Instance_OSType={WinSrv2003, Linux, MacOS, UNIX, XP} can implement bitmap index expression as the mapping method like Figure 2.

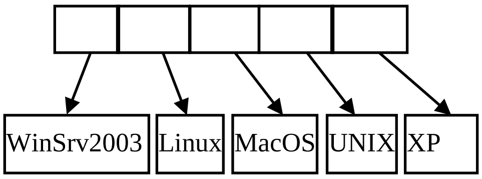


**Figure 2. Mapping Method of bit Index upon 1-D Attribute**

In P2P system, each peer node $P$ maintenances one local or multi-neighbor node index info of resources. The details on index mechanism are introduced in literal [6].

Figure 3 shows a P2P network with a single attribute $A1$, the value of $A1$ is encoded as a vector index of 5. The binary string of dash area is value of local bitmap index, another outside of binary string in white frame is the value of $LinkBitIdx(P \to P', A1)$. For example, node D has a local item $D$ and the value upon attribute $A1$ is $NodeIdx(D.A_1) = 10000$, while we have $LinkBitIdx(D \to E, A_1) = 00101$. In fact, we also can see the value of $LinkBitIdx(A \to C, A_1)$ is the value of bit vector by logical operator OR with node B, C, D, E and F.

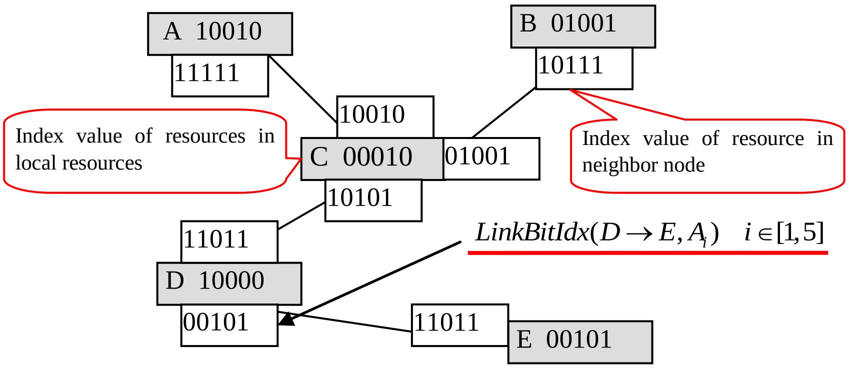

A 10010
11111
B 01001
10111
10010
C 00010
01001
10101
Index value of resources in local resources
Index value of resource in neighbor node
$LinkBitIdx(D \to E, A_i) \quad i \in [1,5]$
11011
D 10000
00101
11011
E 00101

**Figure 3. Sample of bit Index with Node Resource upon Overlap Network of Spanning Tree**

### 3.2. Querying Algorithm for Computable Resource

Considering the situation like Figure 4, for simplification, we only calculate the connected nodes. First case, node $P_1, P_2$ and $P_3$ have one resource matching query $Q$ at least. Now suppose node $P_1$ received a query $Q$, after $P_1$ inspects a local match, later forwards the query to its neighbors. Here node $P_2$ forwards the query to $P_3$, $P_3$ sends the query back to $P_1$, repeat it again, it will cause an infinite loop. In order to avoid loop we need improve the route policy, which is to attach an additional info such as node ID list to record all traversal nodes linked the local node [7].

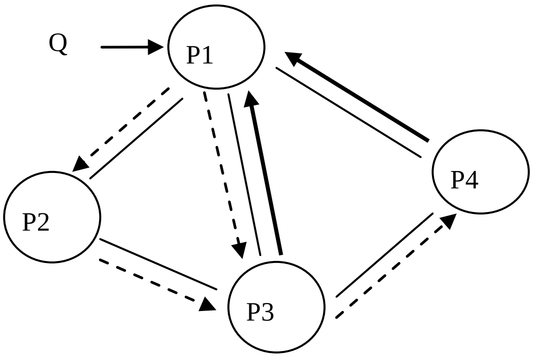


**Figure 4. Sample of Multi-times Execution on Message Forward with Cycle**

This method eliminates the loop, but a new problem appears. Notice that node $P_3$ received two queries $P_1, P_2$ which derived from different groups, so the list method can't prevent repeated multi-query. Based on this reason, we can give each node $P$ linked a query cache, that cache contains the latest query ID of number $m$. The detail route query algorithm on node $P$ as follows.

```
Algorithm1 Lookup(Q)
loop{
    Node $P$ waiting for query $Q$ from neighbor node $P_{in}$;
    If query $Q$ not in query cache of node $P$
        Add query $Q$ to query cache of node $P$;
        Resultset $R$=null;
        For all neighbor node $P_{out}$ except $P_{in}$
        {
         If $P_{out}$ not belongs the group of current query $Q$
             Add group ID to group ID belongs to query $Q$;
          Else
             Forward query $Q$ to $P_{out}$ for $Match(Q, P \to P_{out})$;
               Save the match result to $R$;
            }
      If exists resource of local match query $Q$
         $R = R + \{P\}$;
      }
   Return $R$ to $P_{in}$;
```

Algorithm1 receives a query from neighbor node $P_{in}$, then forwards to the rest neighbor nodes which have latency match, even to original node, final return the value to node $P_{in}$.

```
Algorithm2 $\mathrm{Match}(Q, P_i \to P_j)$
 If query $Q$ meets AND operator of branch query $Q_1, Q_2 \cdots\cdots$ then
    Return recursive calculation of branch query
        $Match(Q_1, P_i \to P_j) \wedge Match(Q_2, P_i \to P_j) \wedge \cdots$;
 If query $Q$ meets OR operator of branch query $Q_1, Q_2 \cdots\cdots$ then
    Return recursive calculation of branch query
        $Match(Q_1, P_i \to P_j) \vee Match(Q_2, P_i \to P_j) \vee \cdots$;
 else execute $Q := v_1 \le A \le v_2$ range query on attribute A
    {
     Take $a_0, a_1, \cdots, a_k$ as sub-division of attribute A;
    For all $i = 0 \cdots k-1$
        $b_i = \begin{cases} 1 & if[a_i, a_{i+1}) \cap [v_1, v_2] \neq \phi \\ 0 & otherwise \end{cases}$;
     $B := (b_0, b_1, \cdots, b_{k-1})$;
     }
  return $(LinkBitIdx(P_i \to P_j, A) \wedge B \neq 0)$ or $(ExtBitIdx(P_i \to P_j, A) \wedge B \neq 0)$.
```

Algorith2 is a match function $Match(Q, P_i \to P_j)$ for testing latency match of query $Q$ in sub-tree $T(P_i \to P_j)$. Suppose the query $Q$ can be divided in more sub-query, for each range query $Q := v_1 \le A \le v_2$, function will compare bit index expression $LinkBitIdx(P_i \to P_j, A)$ in the range $[v_1, v_2]$. If not exists intersection, then it did not exist match in sub-tree $T(P_i \to P_j)$, otherwise maybe a match.

## 4. Algorithm of Spatial Points Clustering on Computational node Location

### 4.1. Generalized Euclid Distance

Nowadays, study on overlay has become a hot field upon P2P [8]. It has emerged many overlay network-based applications such as Napster, Gnutella. Studies have shown that, the topology-aware technology which applied to P2P-based overlay network can significantly improve system performance. However, the purpose of topology-aware lies in reducing the delay of the overlay network, the network distance mentioned herein is a representation of the network latency, noted as RTT (round-trip time), its value can get by message ping using ICMP protocol. In general, considering the characteristics of geographic distribution on computable resource, thus we will inevitably mention the migration time of data and service, especially for data-intensive computing, due to the uncertainty of processing time and the time of data migration. Therefore, we must adopt a trade-off strategy to decide whether task migration or data migration, make budget minimal on network delay time. In this section we will explore a distance model using spatial points clustering. Commonly spatial clustering is decided by the characteristic of attributes. So the nodes which meet the condition are the nodes of sub-group clustering. The condition specifies the certain dynamic querying attribute (see example of section 3.1). Base on the attribute clustering, next we must select the nodes among sub-groups according to their minimal distance. In general, the computational nodes not only have multi-dimensional attributes also have spatial attributes. So in special study we need consider the 2 characteristics on the above. Here the spatial distance for multi-dimensional characteristic clustering is a generalized distance, which is metric of similarity and correlativity in multi-characteristic of attributes. Following is a model definition of GED (Generalized Euclid Distance).

$$D_{ij} = \omega_p d_s^{ij} + \omega_q \sqrt{\sum_{k=1}^{m} \omega_k (a_{ik} - a_{jk})^2} \, . \tag{1}$$

$$\omega_p + \omega_q = 1, \quad \sum_{k=1}^{m} \omega_k = 1. \tag{2}$$

Where $D_{ij}$ is the generalized distance between node $i$ and node $j$ , $d_s^{ij}$ denotes the spatial distance between node $i$ and $j$ in a certain geometry space $S$ . $a_{ki}, a_{kj}$ is the $k$ th attribute value of node $i$ and $j$ respectively. $m$ is the number of computational nodes. $\omega_p, \omega_q$ is the weight of attribute distance and spatial distance respectively. $\omega_k$ is the weight of each attribute. As we know here $D_{ij}$ is the spatial distance when $\omega_p = 1, \omega_q = 0$ . Similarly, $D_{ij}$ is the attribute distance when $\omega_p = 0, \omega_q = 1$ , in that time the clustering belongs to attribute clustering.

From the above model we can see that by tuning the ratio of weight between geometry and attribute distance, our model can implement attribute clustering, spatial clustering and multi-dimension spatial clustering respectively. In this process, there are two steps to select the computational nodes. Firstly, to form characteristic sub-group by attribute clustering, then implement the selection of computable node by unique spatial clustering. In fact, for multi-dimension spatial clustering we not only consider the similarity of attribute characteristic but also need to consider the neighbourhood of spatial location, due to only thus can more disclose the clustering character of computational nodes. In order to better calculating the spatial distance, there are some difficulties to measure distance fully for all nodes in the network, mainly caused by factors as the number of network nodes, node dynamically and network bandwidth consumption. However, most of the researches on spatial distance computing still stay in the phase of theoretical research. Our study adopts the space geometric model to predict the three-dimensional coordinates of the spatial distance as Figure 5.

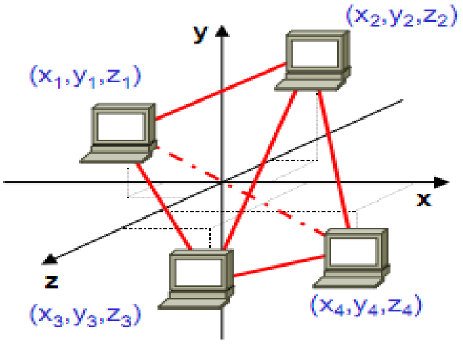


**Figure 5. Spatial Geometric Model base on 3-Dimension Coordinates [9]**

Study shows that the spatial geometric model of coordinate-based network is more accurate than a virtual topology mapping model. Furthermore, GNP (Global Network Positioning) was seen to be the most accurate and robust one [10]. The formalization described as follows:

Suppose network architectural model is a certain geometric space $S$ , using $c_S^h$ denotes the coordinate mainframe $h$ in space $S$ , distance function between coordinates expresses as $f^S(\cdot)$ . Thus the distance between mainframe $i$ and $j$ in space $S$ is follows.

$$d_S^{ij} = f^S(c_S^i, c_S^j) \tag{3}$$

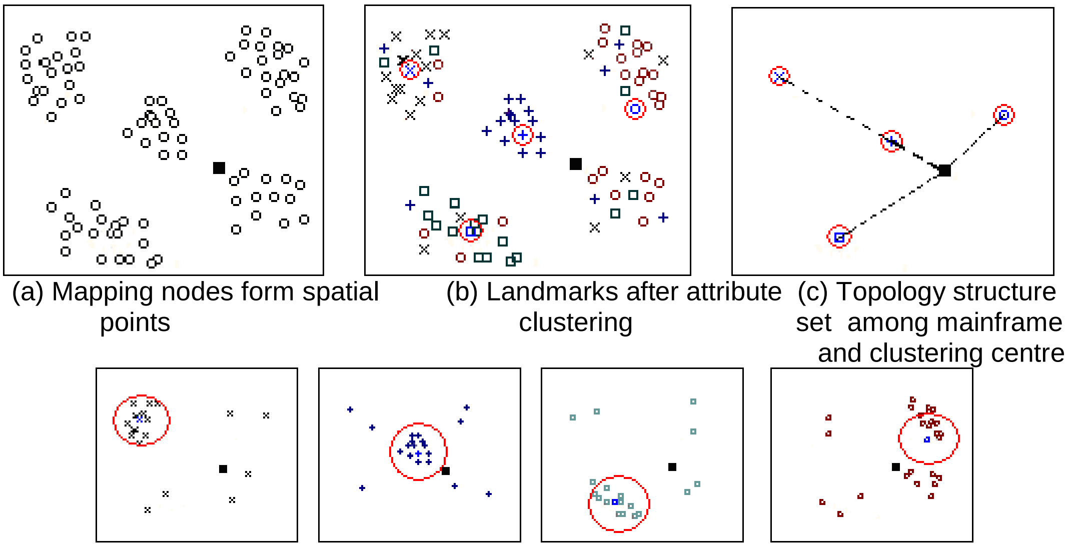
Calculation of spatial distance between mainframe $i$ and $j$ in GNP divides two steps. First, we set landmark nodes $L_i$ , $L_j$ in network, then calculate the distances between mainframe $i$ , $j$ and the nearest landmark node. So we have,

$$D_S^{ij} = d_S^{iL_i} + d_S^{L_iL_j} + d_S^{L_jj} \quad i, j < N \, . \tag{4}$$

In order to lessen the distance error $\varepsilon$ between solid measure and calculations, we use the following objective function, in which each $\varepsilon$ function can calculate by square difference.

$$f_{obj}(c_S^{L_1}, \cdots, c_S^{L_N}, c_S^H) = \sum_{L_i, L_j \in \{L_1, \cdots, L_N\}} \varepsilon(d_{L_iL_j}, d_S^{L_iL_j}) + \sum_{L_i \in \{L_1, \cdots, L_N\}} \varepsilon(d_{HL_i}, d_S^{HL_i}) \tag{5}$$

Where $\varepsilon(d_{ij}, d_S^{ij}) = (d_{ij} - d_S^{ij})^2$ , and $d_{ij}$ denotes solid value, $d_S^{ij}$ denotes calculation value in space $S$ . Here all distance error decided by the sum of each distance error among computational nodes. If only if $f_{obj}(\cdot)$ meets a given threshold value, our system would decide whether data migration or task migration to shorten the network latency. For Instance, Figure 6 is a sample of process on node selection, where the black square denotes the location of services. Firstly, we map all nodes as a spatial points set according multi-dimension query condition, then form sub-group using spatial points clustering to complete the node selection. Figure 6 (a) is a mapping points set of computable nodes. Figure 6 (b) is the result of attribute clustering, the blue identity marked by red circle is the landmark of each category. Figure 6 (c) is a topology structure among mainframe and clustering centre. Figure 6 (d) is the result of nodes selection.



(a) Mapping nodes form spatial points

(b) Landmarks after attribute clustering

(c) Topology structure set among mainframe and clustering centre

(d) Nodes selection using GED model for each category

**Figure 6. Spatial Points Clustering and Nodes Selection by using GED Model**

### 4.2. Algorithm of Computable Nodes Location

The problem of resource selection with high effectively is to study how to coordinate data, services, network and other computational resource to ensure enhancing system performance. As mentioned previously, the performance parameters mainly specified response time and resource utilization, *etc*. Herein we propose a selection algorithm on data, service, nodes and network distance.

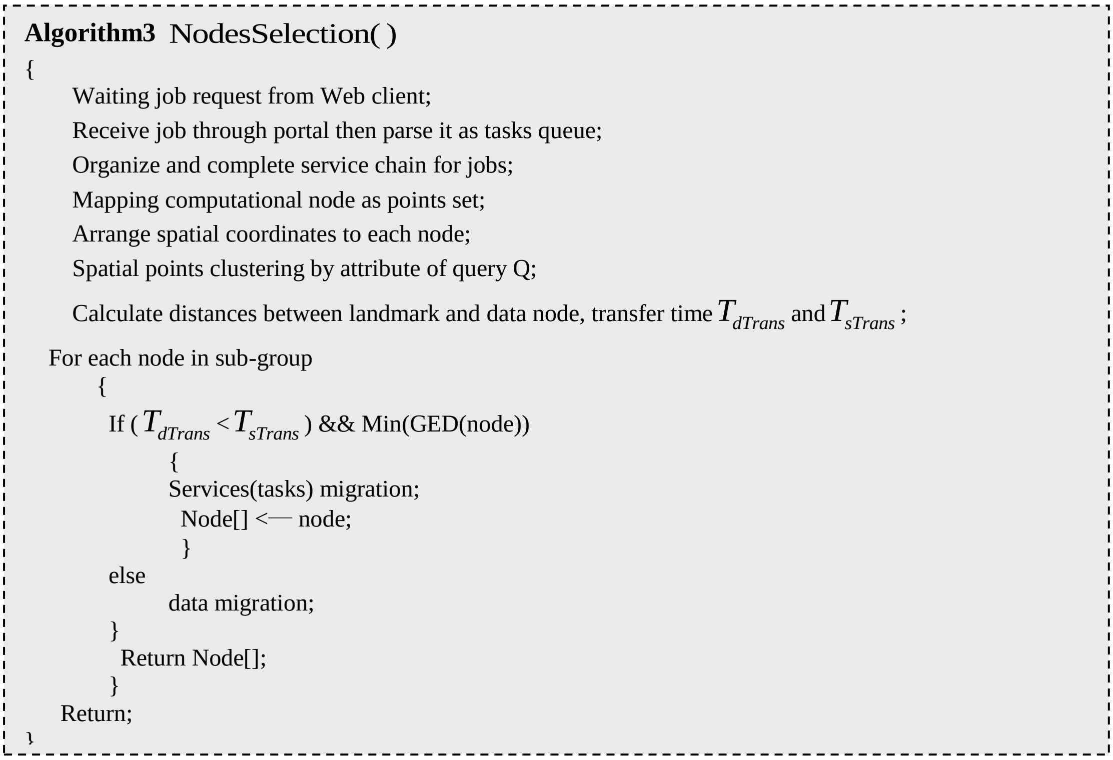
For presentation conveniently, assume that all nodes can meet single computing task in network space $S$ denotes $N_c$, the node which provide service denotes $N_s$, data node is $N_d$, the distance between $N_c$ and $N_d$ is $D_S^{N_c N_d}$, the distance between $N_c$ and $N_s$ is $D_S^{N_c N_s}$, then to complete a job based on $n$ tasks can be quantified as the problem of time consuming as follows.

$$T_C = \min\{\sum_{m=1\cdots n}(T_{N_c c_m} + T_{dTrans} + T_{sTrans})\} \tag{6}$$

Where $c_m$ denotes cost which task $m$ requiring data and service in computational node $N_c$. $T_{dTrans}$ and $T_{sTrans}$ denote migration time of data and tasks, respectively. Its value determined by the calculation value of network distance in section 4.1 and network speed. That is,

$$T_{dTrans} = f(D_S^{N_c N_d}, data), T_{sTrans} = f(D_S^{N_c N_s})\,. \tag{7}$$

Algorithm3 starts from a submitted job, then organizes task queue and search computational resources. Simultaneously, in order to improve system performance, we first need calculate the attribute and spatial distance according to GED, get the nodes which can lessen the transfer time by means of data and tasks migration, and finally execute and return the results to the client.



```
Algorithm3 NodesSelection( )
{
    Waiting job request from Web client;
    Receive job through portal then parse it as tasks queue;
    Organize and complete service chain for jobs;
    Mapping computational node as points set;
    Arrange spatial coordinates to each node;
    Spatial points clustering by attribute of query Q;
    Calculate distances between landmark and data node, transfer time $T_{dTrans}$ and $T_{sTrans}$;
  For each node in sub-group
      {
       If ( $T_{dTrans} < T_{sTrans}$ ) && Min(GED(node))
            {
            Services(tasks) migration;
              Node[] <— node;
              }
       else
            data migration;
       }
         Return Node[];
       }
   Return;
}
```

## 5. Simulation and Experiments

In order to verify the efficiency of our algorithm on resource selection under P2P system, we use PeerSim platform to generate P2P system which contains N nodes' topology structure, and proceeds large-scale simulate experiments for network topologies. Let some of nodes to construct a distributed network, set attribute value for all nodes. We simulate the behavior of each node, make attribute value as distributed randomly with uniform in range [0, 1]. In order to obtain the confidence interval less than 5%, the simulation process repeats to calculate 100 times. Now we select a node as a start node with randomly, choose a probability density from uniform distribution, bit index is make up of $k$ attributes.

In P2P system, query routing and maintenance cost are the two standards of the evaluation on the system performance. Average query routing length can effectively reflect the efficiency of resource discovery, and the differences of search efficiency in different network scale within group and different style of resources organization. Figure 7 depicts the comparison chart of the query routing cost for 3 groups uniform distribution with different algorithms. From the figure we can see with the network size increases, a corresponding increase in query routing cost. In P2P system our algorithm is superior to the other two algorithms on dynamic resource query routing cost.

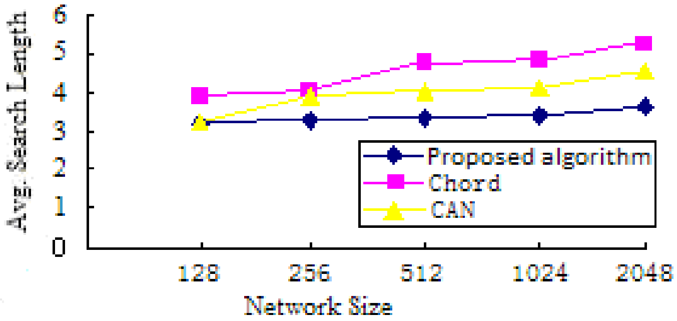


**Figure 7. Comparison of Cost on Search & Route**

The cost of range query based on P2P refers the number of nodes required to access for each multi-dimensional range query results. The queries range is the ratio between query interval in each dimension and the data distribution range. Figure 8 is an impact of searching cost on dimension with different network scale. Figure 9 is an impact of searching cost on range. We can see as the query dimension and the query range increases, the average jump points have been raised accordingly, and the larger the network size, the faster upgrading.

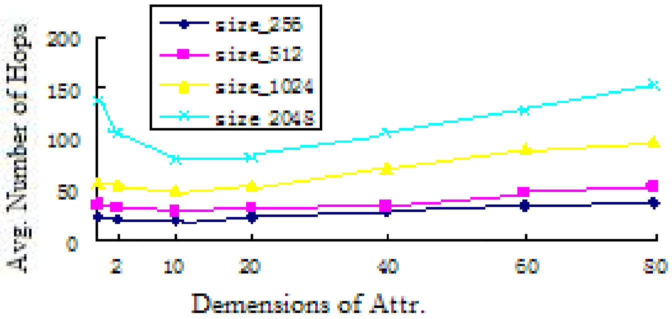


**Figure 8. Impact of Searching Cost on Dimension**

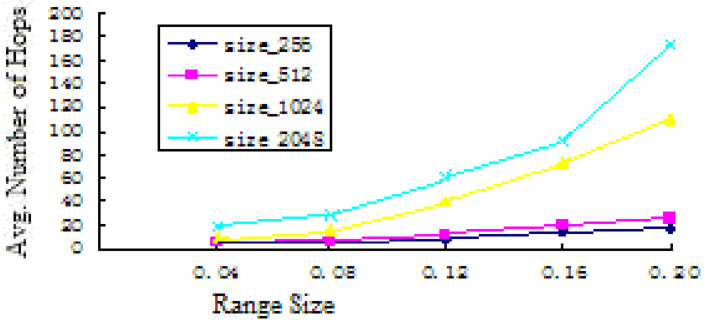
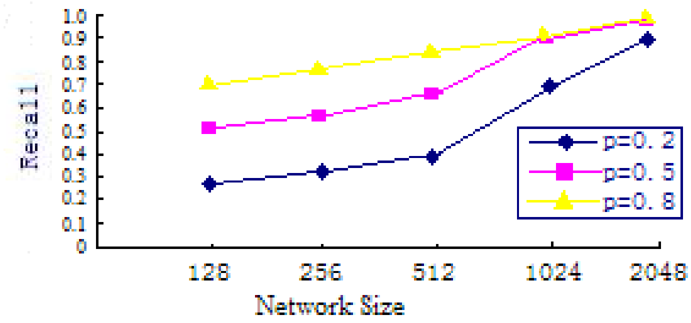
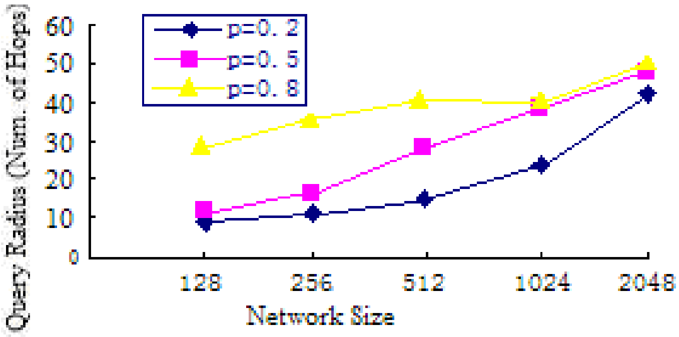


**Figure 9. Impact of Searching Cost on Range**

In addition, in the group-based P2P overlay network system, the query in the communication process can be terminated, and the spread does not reach all matching nodes within the entire network. This will have certain impact in the analysis of recall and network scale. Figure 10 (a) is a case of the recall rate with the changes of network size for different group node and different matching probability. The figure shows that, in order to reach all nodes with potentially match, the query had to across more groups, the longer the queries distance between this group and another group in the network path is, the higher probability path does not match the group. According the previous section description, which means that the query spread along this path is likely to stop, because it’s impossible for routing to pass a none-matched group. So this group has a probability of match down with number of nodes and network size reduction. Figure 10 (b) is a line charts about query radius, indicating when the query does not reach all nodes, the query radius is lower, where the value of the probability p is relevant, because it determines the matching probability for each group. As previously mentioned, one that does not contain any matching node group may exceed this routing chain to reach the other nodes, the query radius with the system scale also increase.



(a) Relation of network size and recall



(b) Relation of network size and query radius

**Figure 10. Relation of Different Match Probability between Network Size and Recall/Query Radius**

Hit ratio reflects the precision of nodes for node location with different network size. For verifying our proposed node selection algorithm, in a distributed computing environment, We use an internal mechanism for P2P to simulate the node attributes query and update, monitoring and publishing the dynamic changes of attribute value to the master node in real-time, and implement the node selection algorithm in Eclipse 3.6.1 platform, completed from the client node generates a random calculation task to simulate node selection. Figure 11 describes the impact of hit ratio on certain query dimension $k$ in different network size. In the figure we can see the precision of hit ratio will increase slightly with the increase of network size under certain query dimension.

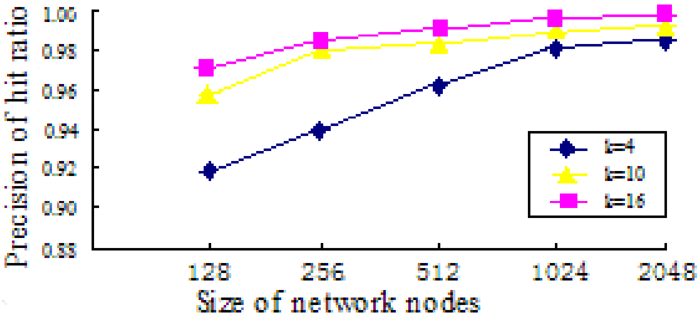


**Figure 11. Impact of Hit Ratio in different Network Size**

For the purpose of verifying the efficiency on querying and selection computable resources, we take a resolution with 14252*14312 high resolution spectra image and a multi-spectral image 7129*7151 make fusion comparative experiments, respectively, adopt a distributed C/S system, grid computing, cloud computing and a P2P system with point clustering algorithm which is proposed in this paper to calculate the task completion time. Experiments show that the increase of network size in different network platform, the task completion time will decrease. The proposed algorithm also has certain reduce in task completion time, and presents a relatively stable situation, and is superior to other three platform. Figure 12. indicates the total search time in our P2P platform is less than the ones in distributed computing, grid computing and cloud computing when searching the same number of resources, due to the algorithm of resource selection is used. What's more, the trends of system performance are similar with the same resources on search time.

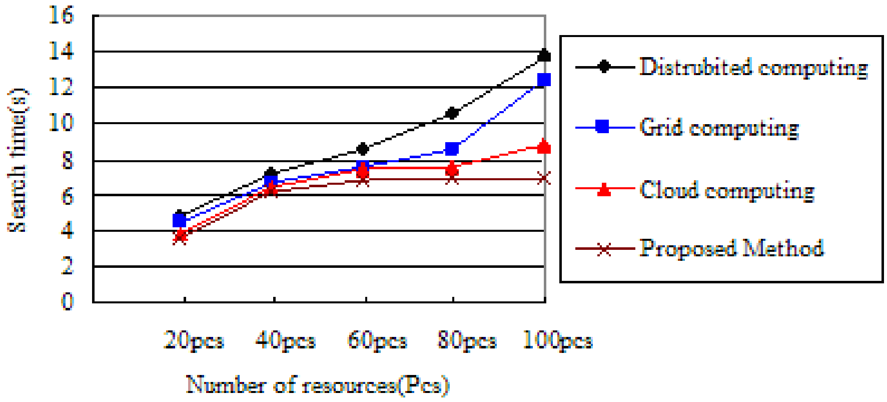


**Figure 12. Comparison of Search Time for different Platform**

## 6. Related Work

Nowadays, there are many researches on discovery of resources and services, including centralized and distributed two methods. But when the resource is too large and operations are frequently, centralized method may have problems about service bottleneck and single point of failure. While if we search the resources through flooding mode in distributed computing environment, there may be caused phenomenon of network delay. Therefore, how to express and discovery these resources to form a supercomputer is the need to address the core issues towards heterogeneous resources under distributed network. Faced the high variable load and high availability requirements in our system, usually used the P2P routing information and distributed structure to avoid single point of failure and problem of efficiency bottlenecks. Ranjan *et al*. first proposed a hierarchical P2P cloud deployment architecture [11], focusing on cloud deployment technology service discovery and load balancing, as well as a concern of DHT in the index and manage negative hybrid deployment information classic structured P2P network management model, However, this model failed to solve the problems of the unstructured P2P networks as well as multi-attribute dynamic queries. In order to obtain the optimal search cost towards unstructured P2P network for cluster computing, Tewari *et al*. proposed a model of flooding search for optimal performance upon the search network topology [12]. For the next generation heterogeneous or service-oriented network, studying a mechanism to discovery the node status information and improve the search performance upon P2P systems is very needed. Considering the dynamic of Grid resources, K.Wu and C.Wu proposed a search algorithm based on the state of a dynamic P2P environment [13]. Through shortening the search response time to improve system performance and to obtain better load balancing, but they still have no mention the issue of dynamic resource organization. In terms of enhancing system performance, Amedro *et al*. adopted a middleware such as SOA by coupling the resource s of Grid and cloud computing to handle high communication-intensive applications to get a better performance-price ratio [14]. Z. Stevan *et al*. take the infrastructure of cloud computing to Grid, then build arbitrary complex grid to meet given the highest quality of service requirements [15]. All this proposed methods and theories achieved a certain effect to some extent, but lack of a systematic approach to express and discovery a large number of computable resources and services. So in our study we discuss the united description of computational resources and services for each node and using the structure of spanning tree to manage the entire overlay network, then for these resources to implement bitmap index and routing search algorithm. Towards these resources, how to provisioning with high effectively on selection of resources and services is a key issue. In recent years, promoted by SLA(Service Level Agreement), it has appeared some researches on resource provisioning related measure in grid and cloud computing under different constraints, and resource provisioning on demand according to workflow and loader [16]. But users are more concerns about the system response time when they submitted jobs. So based on the previous research our study more concerns on the selection of computable resource. This is mainly because the nodes selection is the basis of provisioning on resource and service, and the latency time is also an important factor to impact the system performance.

## 7. Conclusion and Future Work

In this paper, we introduced two algorithms for big data processing with high effectively in P2P system. By using a unified JSON description for computational resource of node and bitmap index, with the aid of spanning-tree network, we proposed a intelligent querying algorithm to get computational resources on nodes. After that, base on improving the system performance, an algorithm on node selection was proposed. Finally, we validate the efficiency of the proposed algorithm by spatial points clustering using GED model. Experiments shows that our proposed algorithms are available and feasible, so as to solve other big data process provide reference for computational resources location.

Future work includes optimizing user-transparent scheduling method and heuristics to handle faulty nodes in a bounded-broadcast tree.

## Acknowledgements

This work is supported in part by doctoral startup fund of Huizhou University No. $C_5$13.0201.

## References

[1] Z. Zeng, “Research on Resource and Service Provisioning with High Efficiently for Massive HRRS Image under Gloud Environment “, China: Zhejiang University, **(2012)**.

[2] Introducing JSON, http://www.json.org/, Accessed February 2009.

[3] A. Goscinski and M. Brock, “Toward dynamic and attribute based publication, discovery and selection for cloud computing”, Future Gener. Comput. Syst., vol. 26, no. 7, **(2010)**, pp. 947–970.

[4] M. A. G. Brock, “Publishing Dynamic State Changes of Resources Through State Aware WSDL”, Web Services, ICWS IEEE International Conference, **(2008)**.

[5] J. G. Z. Yu, “Web Service Discovery and Dynamic Invocation Based on UDDI/OWL-S”, Lecture Notes in Computer Science. C. Bussler and A. Haller, Springer Berlin / Heidelberg, **(2006)**, pp. 47–55.

[6] R. S Chang and M. S. Hu, “A resource discovery tree using bitmap for grids”, Future Generation Computer Systems, vol. 26, **(2010)**, pp. 29–37.

[7] M. Marzolla, M. Mordacchini and S. Orlando, “Tree vector indexes: Efficient range queries for dynamic content on peer-to-peer networks”, Proceedings of the 14th Euromicro Conference on Parallel, Distributed and Network-based Processing (PDP 2006), IEEE Computer Society, **(2006)**, pp. 457-464.

[8] Q. Z Xia, G. G Xie and Y. H Min, “ISP: Indexed-Based Structured P2P Networks”, Chinese Journal of Computers, no. 4, **(2006)**, pp. 4602-4610.

[9] B. Donnet, B. Gueye and M. A. Kaafar, “A Survey on Network Coordinates Systems, Design, and Security”, IEEE Communications Surveys & Tutorials, vol. 12, no. 4, **(2010)**, pp. 488-503.

[10] E. Ng, “Predicting Internet Network Distances with Coordinate-based Approaches”, IEEE INFOCOMM, **(2002)**.

[11] R. Ranjan and L. Zhao, “Peer-to-Peer Cloud Provisioning: Service Discovery and Load-Balancing”, Computer Communications and Networks. N. Antonopoulos and L. Gillam, Springer London, **(2010)**, pp. 195–217.

[12] S. Tewari and L. Kleinrock, “Optimal Search Performance in Unstructured Peer-to-Peer Networks with Clustered Demands”, IEEE Journal on Selected Areas in Communications, vol. 25, no. 1, **(2007)**.

[13] K. Wu and C. Wu, “State-based search strategy in unstructured P2P”, Future Generation Computer Systems, **(2011)**.

[14] B. Amedro and F. Baude, “Combining Grid and Cloud Resources by Use of Middleware for SPMD Applications”, CLOUDCOM '10, **2010**, Washington, DC, USA, IEEE Computer Society.

[15] S. Zonjic, B. Hariri and S. Shirmohammadi, "Multidimensional query based routing for virtual environments", Proceedings of the 2009 IEEE international conference on Virtual Environments, Human-Computer Interfaces and Measurement Systems, **(2009)**, pp. 107–112.

[16] C. Vázquez, E. Huedo and R. S. Montero, "On the use of clouds for grid resource provisioning", Future Generation Comp. Syst., **(2011)**, pp. 600-605.

## Author

**Zhi Zeng**, he is currently a teacher in the dept. of computer science at Huizhou University in Guangdong province, China. He received the B.E. degree in Computer Science from Jiangxi Normal University, and the M.E. degree in Computer Science from Zhejiang University, and Ph.D in GIS in Zhejiang University, China. His research interests contain grid computing and cloud computing, and new generation GIS including remote sensing image processing, with a focus on massive data processing in network computing. He has published the paper more than 10 in recent years.