Analysis on Greedy-Search based Service Location in P2P Service Grid

Cheng Zhu, Zhong Liu, Weiming Zhang, Weidong Xiao, Dongsheng Yang
Department of Management Science and Engineering, National University of Defense Technology, Changsha, Hunan, P.R.C
zhucheng@nudt.edu.cn

Abstract

Service location based on greedy search is studied in this paper. A model is built to analyze the influence of the network topologies on this location method, and different network topologies are constructed to validate the model by ways of simulation. The results show that, though the model assumes an uniform degree distribution, it agrees with the simulation result well if the average degree of the network is reasonably big. Hops and relative QoS index of the node found in a service location process are used to evaluate the effectiveness of the location method as well as the probability of locating the first 5% nodes with higher QoS level. Both model and simulation result show that, the performance of greedy search based service location improves significantly with the increase of the average degree of the network. If changes of both topology and the QoS level of nodes can be ignored during a location process, it can be guaranteed that the location has high probability of finding the nodes with relatively high QoS level in small number of hops in a big service community. Model extension under arbitrary network degree distribution is further studied. A "Two-phase" topology construction and service location based on greedy-search is also put forward.

1. Introduction

Further development of Grid is putting more and more requirements on scalability and fault-tolerant, where P2P ideas can certainly be applied. Ian Foster gave a thorough comparisons and analysis on Grid and P2P, and pointed out the trends and potential profits of the merge of the 2 technologies[1]. On the other hand, service is becoming a basic application pattern of the Grid, which is partially reflected in the latest OGSA architecture. We believe that, service grid in form of peer-to-peer overlay network will be the future direction of the development of Grid, P2P and Web Service. In this overlay network, each node would be logically equal in providing and consuming services, joining or leaving dynamically without system-wide reconfiguration; Services provided by peers would be accessed via standard interfaces, and virtual organizations can be organized among peers to deal with tasks by way of coordination instead of under centralized control. We call such overlay systems P2P service grid.

An important problem in P2P service grid is the scalable and efficient service location. The main requirements include: 1) low performance overhead; 2) fast respond to service query; 3) ability to locate the service provider with good quality of service (QoS) among qualified providers; 4) self-organization capability to deal with dynamic nodes join and leave without centralized control. Above all, the key point here is to locate node(s) capable of providing certain type of service with relatively good service quality in a dynamic and decentralized overlay network environment. It's obvious that Napster-like centralized index server does not address scalability and fault-tolerance needs; Gnutella-like systems can not provide QoS guarantees by themselves; and distributed hash table (DHT) systems [5] treat each node equally and also lack QoS considerations.

To address this problem, let's first consider a wide-spread phenomenon in human society: when people are asked to recommend another person for a task, they will recommend someone who they know and is believed to be capable of performing such task. In addition, this "forwarding" process is also transitive in most cases. Researches in self-organized network show that, it's possible to locate resources efficiently in large-scale networks only with local knowledge. In fact, resource discovery and service location in Grid can borrow ideas from human society, because virtual organizations in Grid and human organizations are in fact similar systems[2]. In P2P service grid, if nodes providing same type of service form a service community overlay on the underlining overlay network, similar "greedy" request forwarding strategies can be adopted once an entry point of the community is obtained: always forward the location request to the neighbors with the best QoS until no neighbor has better service quality.

The greedy-search based service location (GSBSL) is QoS-aware, decentralized and simple to implement without too much network traffic and processing overhead. Similar resource location method has already been used in some computing grid and P2P systems[3]. However, the effectiveness of the method is still unclear. It's obvious that the network topology will have great influence on the performance of GSBSL. For example, on a fully connected network, greedy forward will always

Proceedings of the Third International Conference on Peer-to-Peer Computing (P2P'03)
0-7695-2023-5/03 $17.00 © 2003 **IEEE**

find the best service provider. Nevertheless, how would the performance vary with different network topologies? Can the service location method still achieve satisfying result in large networks with much fewer connections between nodes? Is there any theoretical guarantee on the location performance?

We build a theoretical model in this paper to analyze the influence of the network topologies on GSBSL. Our analysis shows that under reasonable condition, this service location method can achieve good result even in a practical large service community. Both model and simulation result show that, to some extent, the performance of GSBSL can be guaranteed.

2. Related work

Resource location method in P2P networks has been widely studied. In unstructured P2P systems, random walk, broadcast and its variants are compared and analyzed[4]. However, the analysis is solely based on simulation, and focused on finding a certain content without distinguish QoS of different searching result. In structured DHT P2P systems like Chord, lookup for the successor of a certain ID can be achieved in O(logN) hops, and load balance is guaranteed with high probability among all nodes[5]. However, DHT P2P systems treat each node equally, and have no inherent support for QoS-aware resource location.

Another study [6] analyzes how to select downloading nodes with good QoS among search results according to their history records and current status. By this way, the responsibility of ensuring good QoS is entirely transferred to the user, which may become a problem when returning result is uncontrolled.

An ant based self-organized service location method is studied in [7]: ant crawls from one node to another according to the node's pheromone table, locating a suitable service in the service list on current node, and modifying the pheromone record of the node. Unfortunately, there is no quantitive result on this location method, and we believe that, just like adaptive algorithms in artificial neural networks (ANN) and genetic algorithms (GA), the efficiency of the algorithm would vary widely with different parameters, which has no definite rule to follow.

QoS-aware discovery of wide-area distributed services is studied in [8], which uses DNS like QoS feedback servers. QoS feedback information is recorded and propagated among servers to facilitate QoS-aware service discovery. However, specialized QoS feedback servers will definitely increase system overhead on management and configuration.

There are also studies on analysis models on resource discovery and service location. Performance models of some hybrid P2P systems are studied in [9], which mainly focuses on centralized systems. The purpose is to compare scalabilities of different P2P topologies. Another study [10] analyzes the expected hops and total nodes visited under probabilistic search forwarding in k-ary tree topology. An ideal analysis model is built in [11] for lookups in Freenet-like network.

To sum up, current studies on resource discovery and service location of P2P systems lack QoS considerations. A few related studies either use specialized servers or transfer responsibility of ensuring QoS to users. As far as we know, few study addresses self-organized service location in large-scale P2P environment with quantitive model analysis in addition to simulation. In addition, current analysis models mainly address on the hops and probability of successful content location under a certain topology. They are not suitable for analyzing the "goodness" of QoS-aware service location method in a real world application in which the environment is dynamic and unstructured.

On the contrary, GSBSL is a self-organized, QoS-aware simple service location method applicable in dynamic environment. Under reasonable assumptions, we build a theoretic model to analyze its performance in dynamic and unstructured network environment.

3. Analysis model on greedy-search based service location

In order to build the analysis model, we make the following assumptions:

Assumption 1: Nodes providing same type of service form an overlay network on the underlining overlay, which is called service community;

Assumption 2: Topology snapshot of a certain service community at a given time can be modeled as a connected unidirectional graph $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$. There is no multiple edges between any two connected nodes in $\boldsymbol{G}$;

Assumption 3: The change of topology of community and QoS of nodes over time is random and unpredictable;

Assumption 4: Node knows status of its neighbors in time;

Assumption 5: During a certain service location process, change of topology of service community and QoS of nodes can be ignored;

Assumption 6: For a certain service location activity, there exists a corresponding QoS evaluation method, so that all nodes in the community can be sorted according to their QoS value in total order at that moment.

We only focus on the location phase after we gain an entry point into a service community. The entry point can be obtained by out-of-band method or using current search techniques in P2P systems. We will explain on other assumptions in following sections. Under all these assumptions, the greedy-search based service location can be converted as the following process:

In a randomly generated graph $\boldsymbol{G}$, the nodes is randomly indexed form 1 to n, which reflects the node's “"inverse goodness” according to its QoS value in a certain service location activity. The rules for forwarding a service location request with a given probability initiated by node i, where i is the index number, is defined as the following: for a location request started from node i, all neighbors of i are checked, among which the node with the smallest index is supposed to be node j. if $j < i$, then the request is forwarded to j and continues, otherwise stops at i.

The whole process can also be viewed as a random sampling process: At each step, indices of some randomly selected nodes from a node set V are compared. The number of selected nodes is determined by the number of neighbors of the starting node. Among the nodes selected at each step, there may be some nodes selected before, and the ratio is closely related to the topology of $\boldsymbol{G}$.

An important performance parameter of the above location process is the length of search or the hop numbers. Since the index of the node stands for the node's QoS level in the whole community, the index of the final node located reflects the effectiveness of the location method. Taking the size of the community into account, we introduce another performance parameter besides hops: relative index of node found in the location. Relative index of node i is defined as $(i / n) * 100\%$, in which n is the total number of nodes in community. The third parameter we'd like to consider in the model is the hit probability of the first 5% node with the smallest relative index in a location process. As a result, to analyze the above greedy location process, we are looking forward to determining: 1) expectation of hops; 2) expectation of the relative index of the node found; 3) relation between the probability of being found and the node's index.

We make following assumptions on the topology of $\boldsymbol{G}$:

Assumption 7: The average degree of $\boldsymbol{G}$ is d, and cluster coefficient[14] is C;

Assumption 8: The degree and local cluster coefficient of each node in $\boldsymbol{G}$ is equal everywhere.

Assumption 8 is a strong assumption in order to simplify problem, so that we can evaluate effectiveness of GSBSL with only two parameters of the community topology, e.g. average degree d and cluster coefficient C. We will discuss situation under arbitrary degree distributions in next section.

We define the following symbols:

$i(k)$: beginning node of the k^{th} step search when location request is initiated from node i ($k \geq 0$), $i(0) = i$;

$\Gamma_{i(k)}$: set of nodes having connection with $i(k)$, e.g. neighborhood of $i(k)$.

$\boldsymbol{V_i}(k)$: set of nodes searched after $(k-1)$ step searches, when location request is initiated from node i, ($k \geq 1$), $\boldsymbol{V_i}(0)=\{i(0)\}$;

$N(k)$: expectation of the number of nodes searched after $(k-1)$ step searches, ($k \geq 1$), $N(0)=1$;

$\boldsymbol{V}_{i(k)}$: set of new node searched in the k^{th} step search, when location request is initiated from node i ($k \geq 0$), $i(k+1) \in \boldsymbol{V}_{i(k)}$;

$D(k)$: expectation of the number of new nodes searched in the k^{th} step search ($k \geq 0$), $D(0) = d$;

w_i: probability of location request initiated from node i, $\sum w_i = 1$;

k_{max}: maximum hop lengths, e.g. time to live (TTL) of a location request;

$p_{i,j}(k)$: probability for j to become the next beginning node when node i is the beginning node of the k^{th} step search ($k \geq 0, 1 \leq j < i$);

L: expectation of hops when searching $\boldsymbol{G}$ with GSBSL;

I: expectation of the index of the node found when searching $\boldsymbol{G}$ with GSBSL;

$P(j)$: probability of node j being found in a GSBSL process ($1 \leq j \leq n$);

$SearchLength(i,k)$: expectation of remaining hops, when node i is the beginning node of the k^{th} step search, $k \geq 0$;

$SearchIndex(i,k)$: expectation of the index of the node found, when node i is the beginning node of the k^{th} step search, $k \geq 0$;

$Find(i,j,k)$: probability of node j being found when node i is the beginning node of the k^{th} step search ($j < i$, $k \geq 0$).

According to the above analysis, our aim is to compute L, I and $P(j)$ ($1 \leq j \leq n$). With the definitions, we have:

$$L = \sum_{1 \leq i \leq n} (w_i \times SearchLength(i,0)) \tag{1}$$

$$I = \sum_{1 \leq i \leq n} (w_i \times SearchIndex(i,0)) \tag{2}$$

$$P(j) = \sum_{j \leq i \leq n} w_i \times Find(i,j,0) \tag{3}$$

$$SearchLength(i,k) = \begin{cases} 0 & i = 1 \\ \sum_{1 \leq j < i} (p_{i,j}(k) \times (1 + SearchLength(j,k+1))) & i > 1 \end{cases} \tag{4}$$

$$SearchIndex(i,k) = \begin{cases} 1 & i = 1 \\ \sum_{1 \leq j < i} (p_{i,j}(k) \times SearchIndex(j,k+1)) + & \\ (1 - \sum_{1 \leq j < i} p_{i,j}(k)) \times i & i > 1 \end{cases} \tag{5}$$

$$Find(i,j,k)=\begin{cases}0 & i=1, j>1\\ 1 & i=1, j=1\\ \sum_{i<m\le j} p_{i,m}(0)\times Find(m,j,k+1) & 1<i\le j\end{cases} \tag{6}$$

When node i becomes the beginning node of the k^{th} step search, it's possible to either jump to node $i-1$, $i-2$, … , 1as the next beginning node to continue the $(k+1)^{th}$ step search, or to end at node i. The probability value of $p_{i,j}(k)$ is determined by $|\boldsymbol{V}_{i(k)}|$ and $|\boldsymbol{V}_i(k)|$.

According to the definition of the cluster coefficient, it's easy to see that, there would be up to $C(d\text{-}1)$ nodes among the d nodes checked in the step 0 becoming neighbors of the beginning node of step 1, and the initiating node must be neighbor of the beginning node of step 1. As a result, the number of new node searched in step 1 would be $(1\text{-}C)(d\text{-}1)$. Suppose the k^{th} step search is going on, and consider the possible connections between $i(k)$ and other nodes in $\boldsymbol{V}_i(k)$, nodes in $\boldsymbol{V}_i(k)-\{i(k)\}$ can be divided into following 4 categories:

1) $\boldsymbol{V}_{i(k)}(1)$: nodes must have connection with $i(k)$, $\boldsymbol{V}_{i(k)}(1) = \{i(k\text{-}1)\}$, $|\boldsymbol{V}_{i(k)}(1)| = 1$;
2) $\boldsymbol{V}_{i(k)}(2)$: nodes must not have connections with $i(k)$, $\boldsymbol{V}_{i(k)}(2) = \{i(k\text{-}2)\}\cup\ldots\cup\{i(0)\}$, $|\boldsymbol{V}_{i(k)}(2)| = k\text{-}1$;
3) $\boldsymbol{V}_{i(k)}(3)$: nodes may have connection with $i(k)$, which has connection with $i(k\text{-}1)$. $\boldsymbol{V}_{i(k)}(3) = \Gamma_{i(k-1)}$ - $\{i(k\text{-}2)\}$ - $\{i(k)\}$, $|\boldsymbol{V}_{i(k)}(3)| = (d - 2)$. Because of greedy search strategy, $\forall v \in \boldsymbol{V}_{i(k\text{-}1)}$, there is no connection between v and $i(k\text{-}2)$. So the expected number of all possible connection between nodes in $\Gamma_{i(k-1)}$ is $d(d\text{-}1)/2 - D(k\text{-}1)$. From assumptions 7~8, we obtain the expectation of connections between $i(k)$ and nodes in $\boldsymbol{V}_{i(k)}(3)$:
$$l(V_{i(k)}(3)) = \frac{Cd(d-1)/2}{d(d-1)/2 - D(k-1)}(d-2) \tag{7}$$
4) $\boldsymbol{V}_{i(k)}(4)$: nodes may have connection with $i(k)$, which has no connection with $i(k\text{-}1)$. $\boldsymbol{V}_{i(k)}(4) = \boldsymbol{V}_i(k)-\{i(k)\}-(\boldsymbol{V}_{i(k)}(1)\cup\boldsymbol{V}_{i(k)}(2)\cup\boldsymbol{V}_{i(k)}(3))$, $|\boldsymbol{V}_{i(k)}(4)| = N(k) - (d + k - 1)$. Let $\boldsymbol{V}_{i(k)}(5) = \boldsymbol{V} - \{i(k)\} - (\boldsymbol{V}_{i(k)}(1) \cup \boldsymbol{V}_{i(k)}(2) \cup \boldsymbol{V}_{i(k)}(3))$, then $\boldsymbol{V}_{i(k)}(4) \in \boldsymbol{V}_{i(k)}(5)$. $|\boldsymbol{V}_{i(k)}(5)| = (n - (d + k - 1))$. The expectation of connections between $i(k)$ and node in $\boldsymbol{V}_{i(k)}(5)$ is $(d\text{-}1) - l(\boldsymbol{V}_{i(k)}(3))$, so the expectation of connections between $i(k)$ and node in $\boldsymbol{V}_{i(k)}(4)$ is:
$$l(V_{i(k)}(4)) = \frac{N(k)-(d+k-1)}{n-(d+k-1)}\times((d-1)-l(V_{i(k)}(3)))$$
(8)

Nodes searched in GSBSL are shown in figure 1. With (7)~(8), we have:

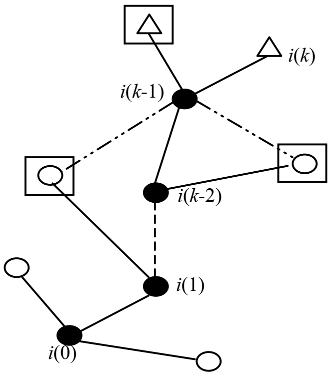


Fig. 1 Nodes searched in greedy search process. Triangle marks node in $\boldsymbol{V}_{i(k\text{-}1)}$, and square marks node in $\boldsymbol{V}_{i(k)}(3)$.

$$D(k)=\begin{cases} d & k=0\\ (1-C)(d-1) & k=1\\ d-(1+l(V_{i(k)}(3))+l(V_{i(k)}(4)) & k>1\end{cases} \tag{9}$$

$$N(k)=\begin{cases}1 & k=0\\ 1+\sum_{0\le m<k} D(m) & k>0\end{cases} \tag{10}$$

Because $D(k)$ and $N(k)$ are respectively expectations of $|\boldsymbol{V}_{i(k)}|$ and $|\boldsymbol{V}_i(k)|$, we use them to estimate $p_{i,j}(k)$ $(1\le j<i)$:

$$p_{i,j}(k)=\begin{cases}0 & if \quad \begin{array}{l}(n-[N(k)]-j<[D(k)]-1)\\ or(k>k_{\max})\end{array}\\ C_{n-[N(k)]-j}^{[D(k)]-1}/C_{n-[N(k)]}^{[D(k)]} & else\end{cases} \tag{11}$$

$[D(k)]$ and $[N(k)]$ are round value of $D(k)$ and $N(k)$. Since k will be no more than n, it's easy to see from (1)~(11) that, time complexity to compute L or I is $\boldsymbol{O}(n^3)$, and space complexity is $\boldsymbol{O}(n^2)$; and those for $P(j)$ $(1\le j\le n)$ are $\boldsymbol{O}(n^4)$ and $\boldsymbol{O}(n^3)$ respectively.

4. Model and Simulation Result

We construct different topologies to run simulations to validate the model result.

4.1. Simulation schemes

Under assumption 5, static simulation can be used. E.g. a certain topology is constructed at the beginning, and then the nodes are randomly indexed. A few nodes (10) are selected randomly to initiate the search process, and hops and final node index are recorded. To minimize deviation, for each type of topology, we construct and run simulations several times (5) and record the average numbers. We use 3 topology generating methods:

1) PLOD[12]: to generate topology with power-law degree distribution;

2) Neighbor introduction[3]: randomly choose a node as the entry point for the joining node, who accepts the new node as its neighbor as well as introducing the new node to some of its old neighbors. It can be used to construct topologies with large cluster coefficient.
3) Random: randomly choose some existing node as neighbors for the joining node, simulating Gnutella-like P2P topologies.

4.2. Results Analysis

Results of both model computation and simulation are compared in figure 2~3, among which index(%) stands for the average relative index of the node found (expectation for model results, and the following are the same); hops means average hop numbers in location; first 5(%) stands for the ratio of the first 5% nodes with smallest relative index found in all simulations (probability for model result). n is total number of node in topology, and s stands for the standard deviation of nodes degree:

$$s = \sqrt{\frac{1}{n-1}\sum_{1\le i\le n}(d_i - d)^2} \qquad (12)$$

d_i is degree of node i. In addition, we set $w_i = 1/n$ and k_{max} =10 in model computation. We can see from figure 2~3 that model result agrees well with simulation result when s is small, both of which indicate the following results:

1) Average degree d has crucial influence on the effectiveness of the location method. Increase d will significantly improve the performance. If d is reasonably big, about 10~20, it can be guaranteed under our assumptions to find a node with relative high QoS level in short hops even in a community with large size;
2) In a topology with large cluster coefficient, location process terminates faster than those with smaller ones, while the effectiveness goes down a little. This is because the search will focus more on the surroundings;
3) For the topologies with the same average degree and cluster coefficient, an increment in size of the community causes the search hops to increase and the effectiveness to decrease. However, this change is quite slow compared with the topology expansion.

To comprehensively understand the location performance, we initiate searches from last 5% nodes with big relative index (which means poor QoS), and find out that, except for 15~20% increase in hop numbers, the final location quality is about the same as the result in the figures..

To evaluate the effect of assumption 8 on model result, we construct topologies with skewed degree distribution using PLOD, and compare the model and simulation result. The figure 4 shows that, with increment of the average degree, model results gradually converge to the simulation result. Though there exist gaps between the two when average degree is small, they do not affect us to use the model to analyze the influence of the community topology on GSBSL's effectiveness. Since cluster coefficient has little influence compared with the average degree, we do not further consider its effect on model accuracy.

4.3. Discussion

The size and the dynamic feature of P2P service grid makes it very difficult to build models to analyze detailed performance of a given self-organized and decentralized service location method while keeping complexity at an acceptable level. In our model, we ignore the change of community topology and node's QoS level during a location process, which makes it possible to use relative index of the node as a measurement of the effectiveness of the location method. Topological changes during a few hops in the service location process are trivial compared with the evolution of the whole network, so we believe the topological changes can be safely ignored during the model analysis. However, if service location processes are concurrently carried out and followed by service delivery, it may be plausible that some nodes' QoS would change during the interval between the service location activity and the service delivery. Nevertheless, we care more about the topological influence on the GSBSL in the model. Moreover, if the interval between two service location activities are long enough or the relative QoS level of a node do not change much after accepting an incoming task (which is plausible when the capability of nodes is skewed or workload is low), assumption 5 still makes sense.

4.3.1. Model extensions under arbitrary degree distributions. Suppose the generating function of the degree distribution in the service community overlay topology is $G_0(x)$, consider the node's degree reached by following an arbitrary connection in the topology. Since the connection arrives at a node with a probability proportional to the degree of the node, the generating function of the degree distribution of the node arrived following a randomly chosen connection is[13]:

$$G_1(x) = \frac{\sum_k k p_k x^k}{\sum_k k p_k} = x\frac{G_0'(x)}{G_0'(1)} \qquad (13)$$

where k ranges from 1 to n, p_k is the probability for a node in network with degree of k. Therefore, the expectation of the degree of the node arrived following a randomly chosen connection is:

$$d' = G_1'(1) \quad (14)$$

The model can immediately be extended to fit arbitrarily degree distributions.

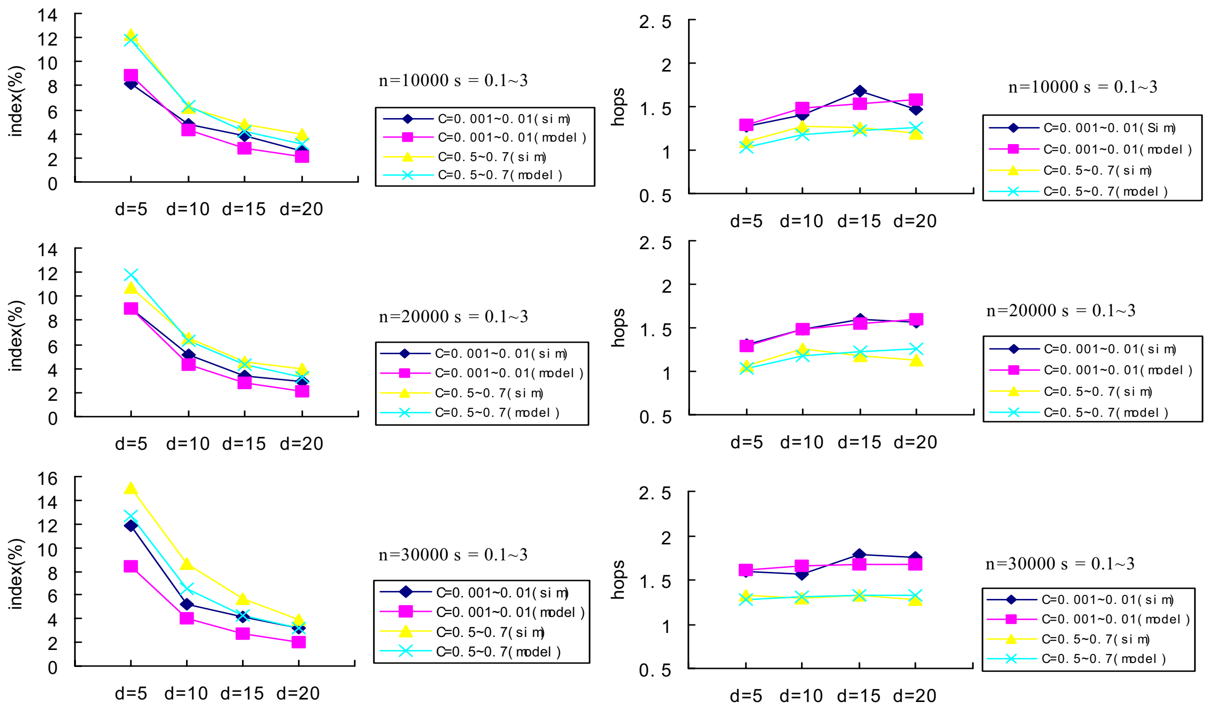


Fig. 2 Hops and relative index of node found in different topologies

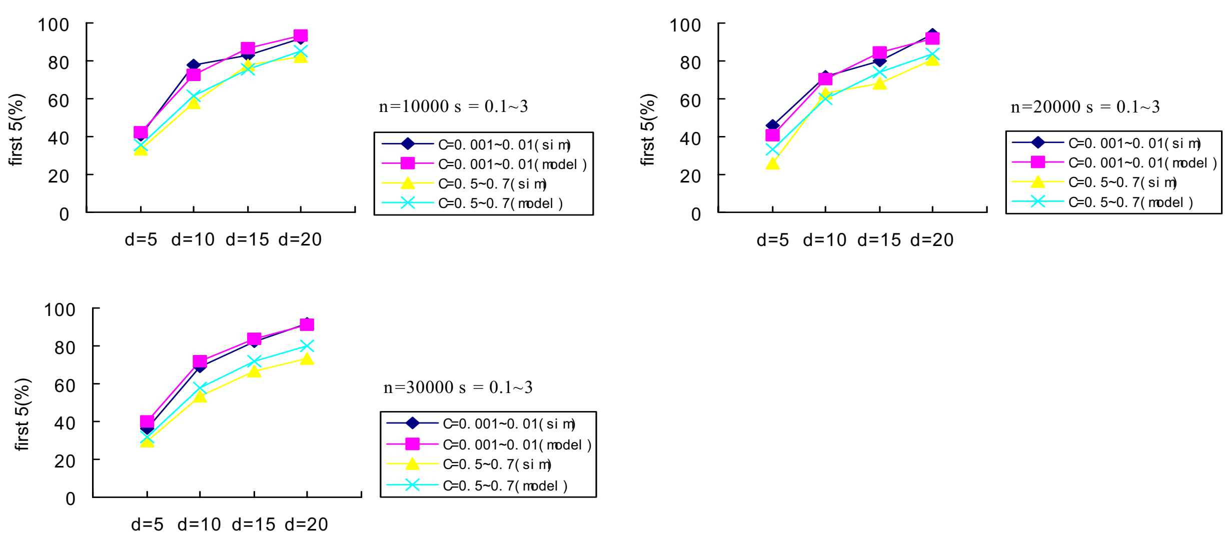


Fig. 3 Proportion of first 5% nodes with best QoS found in different topologies

If the node's QoS level has no relation with its degree, then we only need to substitute d in (7)~(8) with d', and replace (9) with

$$D(k) = \begin{cases} d & k = 0 \\ (d'-1) - C(d-1) & k = 1 \\ d' - (1 + l(V_{i(k)}(3)) + l(V_{i(k)}(4)) & k > 1 \end{cases} \quad (9')$$

4.3.2. Influence of average path length of community overlay on effectiveness of GSBSL. Average path length of a connected network is defined as the average over the shortest path length between any two nodes in the network, which is important topological parameters. In an application level overlay network as P2P service grid, the

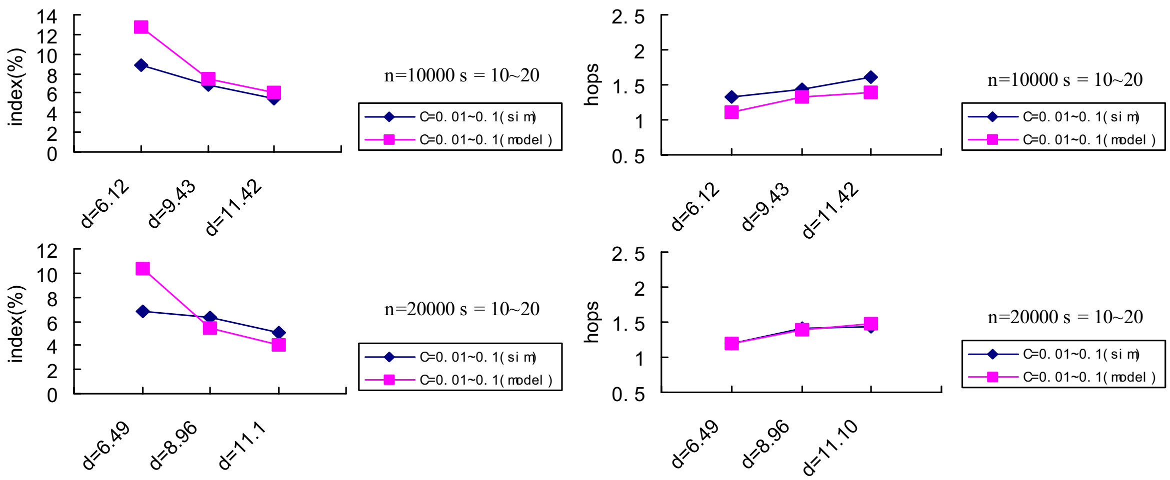
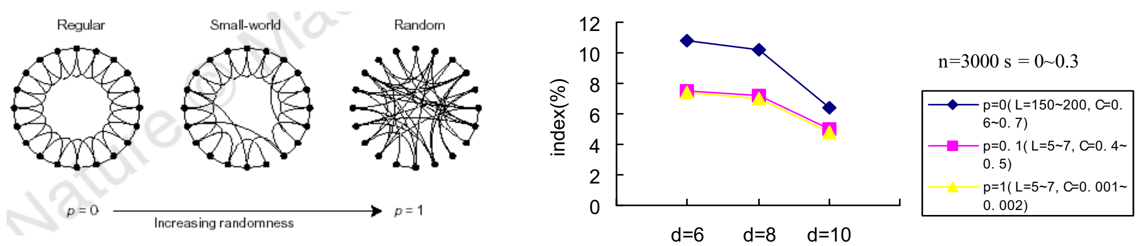


Fig. 4 Influence of s on model result



(1) (2)

Fig.5 effectiveness of GSBSL on regular, random and small-world networks

average path length reflects the expected shortest hops existing between any two randomly chosen nodes. Small-world phenomenon shows that, average path length of networks may differ a lot even under similar degree distribution and cluster coefficient. We use the same topology construction method with [14] to study the effect of the average path length on the GSBSL. Result in figure 5 shows that, while small average path length is beneficial to GSBSL, the benefits fade with the increase of the average degree. We believe that assumption 3 makes the effectiveness of GSBSL insensitive to the average path length. In a dynamic and unpredictable environment like unstructured P2P network, such assumption is acceptable.

4.3.3. Influence of network connectivity on model application. Sometimes, the community overlay may become disconnected. However, researches in complex network show that, in many cases there exists a big connected component covering most nodes in the network. In such circumstance, if we replace the network with its biggest connected component, our model still applies.

5. Conclusion

Through building a theoretical analysis model as well as simulation, we analyze the effectiveness of greedy search based service location method (GSBSL) in P2P service grid in this paper. We find out that the average degree of the community overlay topology has great influence on the location quality. If the average degree is reasonably big (10~20) and the change of community topology and QoS of nodes can be ignored during a location process, the greedy search based service location method will find the node with relatively high QoS level in a quite large service community in short hops. The model presented in this paper is simple to apply, and agrees well with simulation result.

Because of the appealing features of GSBSL, we are considering to adopt the "two-phase" topology construction and service location strategy. E.g. after a new node joins the P2P service grid, it will continue to search and join its belonging community (communities), which forms a separate overlay on the underlining overlay network. When locating a service provider, we should first gain an entry point to the specified service community, and apply GSBSL to reach a provider with good QoS. An implicit precondition of the above scenario is the predefined service ontology, which is feasible in a service grid systems.

We will go on to study on the “two-phase” topology construction and service location based on greedy search, and address on the problem of how to gain an entry point to a certain service community efficiently and quickly. We will compare the whole system design and performance with other self-organized P2P grid systems.

6. Acknowledgements

We thank Wen Dou, Xiaojun Duan and Zhenyuan Wang of National University of Defense Technology for their help on the content of the paper, and we also thank Zao Yang of Ubicom for his revision of the English expressions in this paper.

7. References

[1] Ian F, Adriana I. On Death, Taxes, and the Convergence of Peer-to-Peer and Grid Computing. In: Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS'03). Heidelberg: Springer-Verlag, 2003. http://iptps03.cs.berkeley.edu/final-papers/death_taxes.pdf

[2] Shan JL. Grid Society – A System view of the Grid and P2P Environment. In: Proceedings of the 1st International Workshop on Grid and Cooperative Computing (GCC 2002). Beijing: Publishing House of Electronic Industry, 2003.

[3] Wen D, Yan J, Huaiming W, et al. A P2P Approach for Global Computing. to appear In: Proceedings of the IPDPS03, New York: IEEE Press, 2003.

[4] Qin L, Pei C, Edith C, et al. Search and Replication in unstructured peer-to-peer networks. In: Proceedings of the 16th ACM International Conference on Supercomputing (ICS'02), New York, 2002.

[5] Sylvia R, Scott S, Ion S. Routing Algorithms for DHTs: Some Open Questions. In: Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS '02), Heidelberg: Springer-Verlag, 2002. http://www.cs.rice.edu/Conferences/IPTPS02/174.pdf.

[6] Daniel S. Bernstein, Zhengzhu Feng, et al.. Adaptive Peer Selection. In: Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS '03), Heidelberg: Springer-Verlag, 2003.

[7] Artur A, Sven G, Vadim K, et al. Self-Organizing Control in Planetary-Scale Computing, In: Proceedings of the 2nd International Symposium on Cluster Computing and the Grid (CCGrid 2002), New York: IEEE Press, 2002.

[8] Dongyan X, Klara N, Duangdao W. QoS-Aware Discovery of Wide-Area Distributed Services, In: Proceedings of the 1st International Symposium on Cluster Computing and the Grid (CCGrid 2001), New York: IEEE Press, 2001.

[9] Yang, Beverly, Garcia M. Comparing Hybrid Peer-to-Peer Systems, In: Proceedings of the VLDB 2001, Roma: Morgan Kaufmann 2001. 561~570.

[10] Daniel A. Menascé, Lavanya K. Probability Scalable P2P Resource Location Services. ACM SIGMETRICS Performance Evaluation Review, 2002, 30(2): 48~58.

[11] Hui Z, Ashish G, Ramesh G. Using the Small-World Model to Improve Freenet Performance, ACM SIGCOMM Computer Communication Review, 2002, 32(1): 78~88.

[12] Christopher R. Palmer, J. Gregory Steffan, Generating Network Topologies that Obey Power Laws, In: Proceedings of the IEEE Globecom’00, San Francisco, 2000.

[13] Newman MEJ, Strogatz SH, Watts DJ. Random Graphs with Arbitrary Distributions and Their Applications. Physical Review, 2001, 64(2):

[14] Strogatz SH, Watts DJ. Characteristics of Small World Networks. Nature, 1998, 393(6):440~442.