Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing

An Approach to Build Accessible Grid Service

YinJie[1], Wang Yue-Xuan[2], Wu Cheng[1]
1. Department of Automation, Tsinghua University, Beijing, 100084
2. Institute for Theoretical Computer Science, Tsinghua University, Beijing, 100084
E-mail: yinjie05@mails.tsinghua.edu.cn

Abstract

Grid service is stateful and transient web service, which can be invoked by clients, and is considered to be the mainstream of future internet. An approach to build grid services is provided as follows, a directed graph model of grid services is developed by first create several service providers and directed edges among every two of them, then increase the number of service providers and random connections among them, when size of service network grows, new providers are no longer randomly connected to previous providers but choose those with high incoming degree. This approach differs from previous methods of building small world complex networks in its imitation of real world situations. This approach is analyzed through simulation and has features of complex networks such as small world and scale free. Accessibility of grid services building through this approach is analyzed when service providers dynamically exit grid system and stop to provide their services.

1. Introduction

Service Oriented Architecture (SOA) has attracted many research interests recently. It's a trend to solve many difficult real world problems. The definition of Web Service and related information can refer to [1]. The concept of SOA and its related applications are introduced in [2] and [3], as well as some limitations of Web Service. For example, Web Service can not provide transitory and stateful services, which are confirmed to be important in some situations. Grid service is developed on Web Service and can provide transit and stateful services. The reason why stateful service is necessary is that Virtual Organizations in grid are organized and disbanded dynamically and resources in them generally provide transient services.

Grid is both an idea and a technology. It aims at connecting resources together for share and collaboration in form of grid services. The idea of grid originated from the 1960s, which suggested connections of mainframes so as to provide more powerful computing ability. This idea got developed and was well known to the world when [4], which introduced basic conceptions and key ideas of grid was published. Grid is expected to be an infrastructure like power supply system. The concept of grid and lots of grid based engineering projects are introduced in [5] and [6]. The sandglass architecture of grid, which is a protocol based architecture and consists of five levels, is proposed in [7]. The tool (Globus Toolkit 2) supporting sandglass architecture was published and it was used to develop grid projects. But in practice, sandglass architecture is not compatible with current industry network protocols and needs further improvement. Another version of architecture for grid, Open Grid Service Architecture (OGSA), is provided in [8]. The services based on OGSA are called grid services. Grid service is a kind of transient stateful Web Service and supports reliable and secure service invocation, lifetime management, notification, policy management, credential management and virtualization. Current version of toolkit to develop grid service, Globus Toolkit 4, was published in 2005.

Grid has achieved great success and many successful projects are on working, like programs in Science Grid [9], EUROGRID [10] and Data Grid [11], using grid technologies. But there is little literature focus on theoretical analysis of accessibility to grid service, although Universal Description and Integration (UDDI) provides some basic functions like publishing and locating information about Web Services. How to provide easy access to grid services is of vital importance, for there are so many services in grid and efficient service finding algorithm will facilitate easy access to grid services. This paper focuses on providing an algorithm, which creates relationship among grid services and demonstrating effect of this algorithm through simulation.

0-7695-2909-7/07 $25.00 © 2007 IEEE
DOI 10.1109/SNPD.2007.316

This paper is organized as follows. In section 2, we discuss grid service and key issues about it. In section 3, we present an algorithm to build up grid services and fulfil our simulation using this algorithm in section 4. We analyse simulation results and identifying issues that need further investigation in section 5.

2. Backgrounds

Service invoking process in grid is as follows. A client sends a service request to a scheduler. In Globus Toolkit 4, GRIS (Grid Resource Information Service) and GIIS (Grid Index Information Service) act as scheduler. The scheduler goes to find service provider that can provide the service needed. After negotiation, the URI (Universal Resource Identifier) of that service is returned. The client then invokes the service with input data and other necessary information.

While there are so many service providers in grid system, a suitable way to describe such system is using directed graph, which is under active research in computer science, physics and social science. In a directed graph, vertices represent service providers who can provide some kind of services. Here all services are modeled as non distinctive abstract service and all service providers can also be clients of other services. The directed edges represent accessing to some services, with arrow fly away from client and point to service provider.

The above directed graph model is a high level and abstract model for grid services. The focus is on connectivity of the graph, which means accessibility of certain service in real world. This model takes no account of some detailed processes like how to discover a suitable service provider, security issues, cost of invoking services and related provenance.

With the number of services increased, the directed graph becomes complex networks. Current research work about complex networks are focused on topology, shortest path between any two vertices, graph traversal and statistical characteristics of complex system like clustering and degree distribution.

The complex networks have been under research for a long time by different groups of researchers range from physicists, computer scientists to social scholars. Many real world phenomena can be classified into complex networks problems. A social instance is that a group of people as vertices with some kind of interaction between them as edges. These edges can represent friendship between individuals, business relationship among families. Social networks, information networks, technology networks and biological networks can also be considered to be complex network.

Early model to describe complex networks was random graphs. In random graphs, given N vertices and P, which is the probability of any two vertices that have an edge between them, random graphs theory can provide method to acquire some statistic information including clustering coefficient and degree distribution. The degree distribution in random graph is a Poisson like distribution. Another famous model of complex networks is small world model proposed in [12], which shows a random edge between two vertices with long distance will shorten whole average path length in an undirected graph. The small world model explained the phenomenon of six separations. Six separations mean any two people can find that they are linked by almost six other people and this phenomenon exists in a large variety of fields. It was found that degree distribution in some real networks do not similar to Poisson distribution as in random networks but power law distribution in [13]. Power law distribution is explained in [13] as result of two factors in an undirected graph, size increase and purposely connections with high degree vertices instead of randomly. The directed graph model of internet was introduced in [14] and through experiments on internet found that directed model of real internet had a longer average path length than undirected graph. Some reviews on developments and achievements in complex networks were introduced in [15], [16] and [17].

Compared with former works in complex networks, grid services are more dynamic and its major concerns lie in accessibility of services. Thus in directed graph model, connectivity of the graph is major concerns of our research. The algorithm introduced below will ensure connectivity for grid services. The relationship about graph size, dynamic probability of grid services and probabilities of service accessibility are discussed.

3. Ensuring algorithm

The following algorithm consists of two steps. The first step is free competition phase, in which vertex newly joining into system connects with old vertices randomly. The second step is preferential phase, in which new vertex connects part of its outgoing edges more preferably to those with high incoming degree, and part of its edges to the rest vertices randomly. In this algorithm we suppose that every service provider will provide services to others and at the same time is client of other services. Reflected in directed graph, this means that each vertex has to point to other vertices and be pointed from others. This two steps algorithm is used to build model of grid services as well as many real world applications. In the first step, each vertex is equal and competes freely. While in the second step, the winner with more incoming edges will attract more connections.

***Algorithm*:**

1) Given m_0 vertices surrounding as a circle, shown in figure 1. Each points to and pointed by its two nearest neighbors. There is $2m_0$ edges altogether.

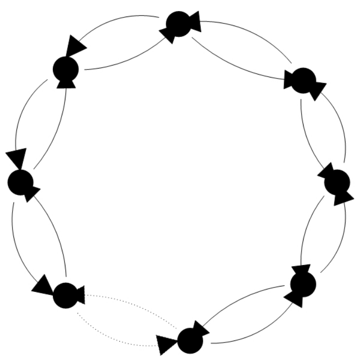

Figure 1 Initial topology for this algorithm

2) Create M vertices in turn. Every new vertex has a probability p_1 points to every existing vertex and a probability p_2 be pointed by every existing vertex. To ensure connectivity of the graph when graph size is not big enough, an additional incoming edge from a randomly selected vertex and an outgoing edge to a randomly selected vertex should be added to new vertex. Suppose every vertex is numbered according to its order into the system, we can get the following equations.

$$E(x_i^{in}) = \begin{cases} 2+\sum_{k=0}^{M}\frac{1+p_1(m_0+k)}{m_0+k} & (i\in[0,m_0]) \\ 1+(i-1)p_2+\sum_{k=i-m_0}^{M}\frac{1+p_1(m_0+k)}{m_0+k} & (i\in(m_0,m_0+M]) \end{cases} \quad (1)$$

$$E(x_i^{out}) = \begin{cases} 2+\sum_{k=0}^{M}\frac{1+p_2(m_0+k)}{m_0+k} & (i\in[0,m_0]) \\ 1+(i-1)p_1+\sum_{k=i-m_0}^{M}\frac{1+p_2(m_0+k)}{m_0+k} & (i\in(m_0,m_0+M]) \end{cases} \quad (2)$$

In equation (1), $E(x_i^{in})$ means expected value of number of edges pointed to the ith vertex x_i, and $E(x_i^{out})$ in equation (2) means expected value of number of edges pointed from the ith vertex. This step is known as phase of free competition.

3) When any new vertex joins into the system, randomly select a number, ξ_0, between 1 and n_0 and point these ξ_0 edges to M+m_0 vertices previously formed. The ξ_0 edges are pointed to M+m_0 vertices with probability proportional to their incoming degree, which is the number of incoming edges. This means a selected number of edges will be pointed from the new vertex to M+m_0 vertices according to equation (3).

$$x_j^{in} \Big/ \sum_{j=0}^{M+m_0} x_j^{in} \quad (3)$$

Then randomly select a number, ξ_1, between 1 and n_1, point them to the rest of existing vertices. Incoming edges of the new vertex are selected randomly from all existing vertices with a select a number, ξ_2, between 1 and n_2. The above n_0, n_1 and n_2 are constant numbers and can be adjusted by users. This step is the process of preferential phase. Vertices with more incoming edges have more chances to attract more incoming edges.

In this step, the number of incoming and outgoing edges is independent of size of system but depends only on n_0, n_1 and n_2. This is true in real life, because it is impossible for new vertex to know the size of system. When a new service provider joins in grid system, he shares his resources to those he trusts and invokes services well known (with more incoming edges in directed graph) or by some other means (randomly selected outgoing edges in directed graph) while taking no account of size of the system he is in.

After this step, we can calculate incoming degree ($x_i^{in\prime}$) and outgoing degree ($x_i^{out\prime}$) of each vertex in equation (4) and (5).

$$x_i^{in\prime} = \begin{cases} x_i^{in} + \frac{x_i^{in}}{\sum_{i=0}^{M+m_0} x_i^{in}}\xi_0 + \frac{1}{M+m_0+N}\xi_1 & i\in[0,M+m_0] \\ \frac{1}{M+m_0+N}\xi_1 & i\in(M+m_0,M+m_0+N] \end{cases} \quad (4)$$

$$x_i^{out\prime} = \begin{cases} x_i^{out} + \frac{1}{M+m_0+N}\xi_2 & i\in[0,M+m_0] \\ \frac{1}{M+m_0+N}\xi_2 & i\in(M+m_0,M+m_0+N] \end{cases} \quad (5)$$

In above two equations, N means the number of vertices be added into system in step 3. x_i^{in} and x_i^{out} represent incoming degree and outgoing degree from step 2 with expected values shown in equation (1) and (2).

Since in (4) and (5), x_i^{in} and ξ_0 are independent, $\sum_{i=0}^{M+m_0} x_i^{in}$ and $\sum_{i=0}^{M+m_0} x_i^{out}$ are constants determined by M, m_0, p_1 and p_2. So we have the following equation (6) and (7).

$$E\left(\sum_{i=0}^{M+m_0} x_i^{in}\right) = \sum_{i=0}^{M+m_0} E(x_i^{in}) = M+2m_0+2Mm_0p_1+\frac{M+2m_0-1}{2}p_2M + \frac{p_1}{4}(M+m_0)(M+2m_0+2)+m_0\sum_{k=0}^{M}\frac{1}{m_0+k}+\sum_{i=m_0+1}^{M+m_0}\sum_{k=i}^{M+m_0}\frac{1}{k} \quad (6)$$

$$E\left(\sum_{i=0}^{M+m_0} x_i^{out}\right) = \sum_{i=0}^{M+m_0} E(x_i^{out}) = M+2m_0+2Mm_0p_2+\frac{M+2m_0-1}{2}p_1M + \frac{p_2}{4}(M+m_0)(M+2m_0+2)+m_0\sum_{k=0}^{M}\frac{1}{m_0+k}+\sum_{i=m_0+1}^{M+m_0}\sum_{k=i}^{M+m_0}\frac{1}{k} \quad (7)$$

Because x_i^{in} and ξ_0 are independent between each other, expected values of (4) and (5) can be shown in (8) and (9).

$$E\left[x_i^{in\prime}\right]=\begin{cases} E\left[x_i^{in}\right]+\dfrac{E\left[x_i^{in}\right]}{\sum_{i=0}^{M+m_0} E\left[x_i^{in}\right]}E[\xi_0]+\dfrac{1}{M+m_0+N}E[\xi_1] \\ (i\in[0,M+m_0]) \\ \dfrac{1}{M+m_0+N}E[\xi_1] \\ (i\in(M+m_0,M+m_0+N]) \end{cases} \quad (8)$$

$$E\left[x_i^{out\prime}\right]=\begin{cases} E\left[x_i^{out}\right]+\dfrac{1}{M+m_0+N}E[\xi_2] \\ (i\in[0,M+m_0]) \\ \dfrac{1}{M+m_0+N}E[\xi_2] \\ (i\in(M+m_0,M+m_0+N]) \end{cases} \quad (9)$$

The expected values of ξ_0, ξ_1 and ξ_2 are $(1+n_0)/2$, $(1+n_1)/2$ and $(1+n_2)/2$, respectively. Combining equation (1) into (8) and (9), we have equations (10) and (11), which depict degree distribution of this algorithm.

$$E\left[x_i^{in\prime}\right]=\begin{cases} \left(2+\sum_{k=0}^{M}\dfrac{1+p_1(m_0+k)}{m_0+k}\right)\left(1+\dfrac{1+n_0}{2\sum_{i=0}^{M+m_0}E\left[x_i^{in}\right]}\right) \\ +\dfrac{1+n_1}{2(M+m_0+N)} \qquad (i\in[0,m_0]) \\ \left(1+(i-1)p_2+\sum_{k=i-m_0}^{M}\dfrac{1+p_1(m_0+k)}{m_0+k}\right)\left(1+\dfrac{1+n_0}{2\sum_{i=0}^{M+m_0}E\left[x_i^{in}\right]}\right) \\ +\dfrac{1+n_1}{2(M+m_0+N)} \qquad (i\in(m_0,M+m_0]) \\ \dfrac{1+n_1}{2(M+m_0+N)} \qquad (i\in(M+m_0,M+m_0+N]) \end{cases} \quad (10)$$

$$E\left[x_i^{out\prime}\right]=\begin{cases} 2+\sum_{k=0}^{M}\dfrac{1+p_1(m_0+k)}{m_0+k}+\dfrac{1+n_1}{2(M+m_0+N)} \\ (i\in[0,m_0]) \\ 1+(i-1)p_1+\sum_{k=i-m_0}^{M}\dfrac{1+p_2(m_0+k)}{m_0+k} \\ +\dfrac{1+n_2}{2(M+m_0+N)} \qquad (i\in(m_0,M+m_0]) \\ \dfrac{1+n_2}{2(M+m_0+N)} \\ (i\in(M+m_0,M+m_0+N]) \end{cases} \quad (11)$$

We can see from (10) and (11) that once parameters like p_1, p_2, n_0, n_1, n_2, m_0, M, N are given, the distribution of edges can be roughly determined.

From this algorithm, we can find that from any of the service provider, you can invoke any services available on grid system when there is no services fluctuation dynamically in grid system.

4. Simulation and results

According to algorithm proposed in Section 3, we simulate the system by assign related parameters in three groups below, as shown in table 1.

Table 1 Parameters in simulation

	m_0	M	N	p_1	p_2	n_0	n_1	n_2
G1	3	100	5000	0.1	0.1	10	40	50
G2	3	100	5000	0.02	0.02	5	15	20
G3	3	100	5000	0.01	0.01	1	2	3

Group 1 has more edges than Group 2 and Group 3. Group 3 has the least edges of these three groups.

Figure 2 is incoming degree distribution with parameters set by above three groups and figure 3 shows outgoing degree distribution.

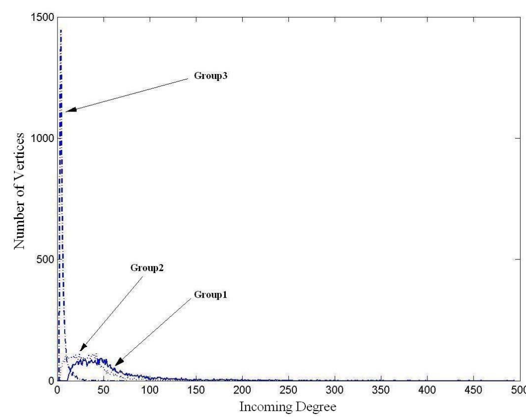

Figure 2 Incoming degree distribution

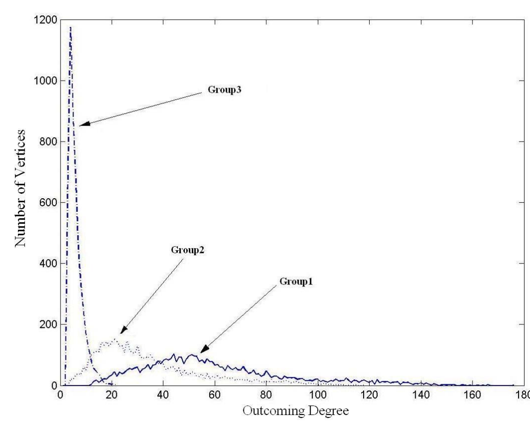

Figure 3 Outgoing degree distribution

From figure 2 and figure 3, we can see that incoming and outgoing degree distribution do not obey normal distribution, but power law distribution $k^{-\gamma}$. And γ of Group 3 in figure 3 is about 2.855, which is similar with result of [14]. In [14] γ of incoming edge degree of internet in 1997 is about 2.72.

The average path length of Group 1, Group 2 and Group 3 are 3.54, 5.99 and 6.57 respectively. It is clear and easy to understand that average path length decreases when edges increased in the system. But it is different from the average path length of 16 from [14]. The reason may lie in its relatively small size of simulation when compared with real world internet.

From above simulation, we can see that this algorithm is reasonable in modeling grid services. The following experiments are to simulate system when some service providers leave grid system and stop their services dynamically. In this case, related vertices and all edges pointed to them and from them will be removed in directed graph. We will show relationship about probability of successfully accessing to a service, probability that service providers leave grid system and size of grid system through simulation.

Figure 4 shows the relationship when dynamic occurs randomly among vertices in grid system. Figure 5 demonstrates this relationship when vertices with top highest incoming degree are removed from system.

The axes X in figure 4 and figure 5 represent number of service providers, Y represents probability that every service provider stops providing service and Z represents probability that a client can access all services in the system.

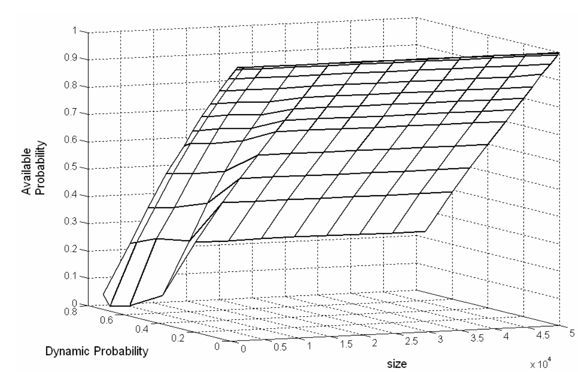

Figure 4 Relationship among size, dynamic probability and access probability under random dynamic condition

From figure 4, we can find that when vertices are randomly removed from graph, remaining vertices are still well connected and this property will not change when size of directed graph varied. In this situation, the only reason for a service not accessible is it leaved grid system for some reason. While in figure 5 when vertices with highest incoming degree are removed, the rest of vertices are not necessarily connected and some isolated sub graph thus formed. The reason for a service beyond access may lie in its provider leave the system or it is still in system but the path from clients to it was cut.

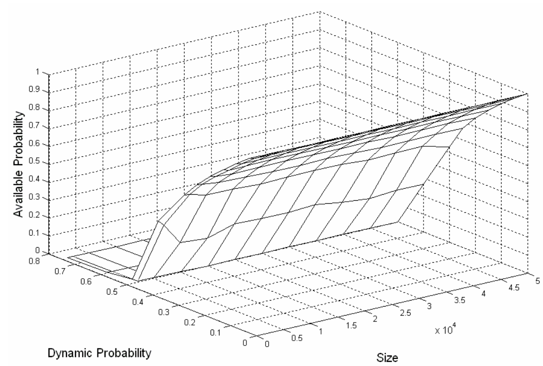

Figure 5 Relationship among size, dynamic probability and access probability under preferential condition

As can be seen from figure 5, services on the system can hardly be accessed when dynamic probability amounts to 50 percent, which results from the fact all vertices are almost isolated and edges among them has been removed away. In real world, most dynamics occur randomly except in case of malevolent intrusion.

5. Conclusion and Future work

From simulation, we can see that system built up using this algorithm, shows its robustness even under large randomly dynamics. The contribution of this algorithm to real world system is that it can provide a rule for new service providers in grid system. When a service provider wants to join a virtual organization in grid system, it should let some services know its existence by providing its URI into these services' accessible lists, which are databases recording their accessible services. At the same time it should record some services' URI to provide invocation when necessary. According to simulation, if a service is known by a little number of clients, it will be known by the entire system, taking no consideration of the security and authorization issues.

Future work can be developed from the following three aspects.

1. The costs to access a service. Above work shows probability of accessibility to a service but the cost is not considered. The costs includes how to find a service that is really on work, optimal path to reach a service and how to choose a more suitable one when there are many similar services. One possible way is to choose services with highest outgoing degree and assign them a more important role in the system, such as distributed service schedulers. Any new service provider should let at least one of schedulers know its existence.

2. When authorization and authentication is considered, the transitivity of connections among vertices will be challenged. How to incorporate security issues into the model deserves future work.

3. How to make full use of every vertices in the system, how to schedule jobs to the most suitable providers in order to balance throughput of the whole system and every vertex in the system need further consideration because a crowd service provider makes its clients impatient while a provider with no service invoking will lost its interest in providing service.

6. Acknowledgements

This work was supported in part by the National Natural Science Foundation of China Grant 60604033 and the Hi-Tech research & Development Program of China Grant 2006AA10Z216.

7. References

[1] Graham, S., Simeonov, S., Boubez, T. etc, "Building Web Service with Java: Making Sense of XML, SOAP, WSDL and UDDI", Indianapolis, IN: Sams Publishing, 2002

[2] Fountain, Thomas C., "Web Service Oriented Architecture: Smart Operations and IT Strategy", *Proceedings of the International Conference on Web Services*, 2003, pp481-484

[3] Shin, David, "Service-oriented architectures", *Scientific Computing and Instrumentation*, vol.21, no.6, 2004, pp26-28

[4] I. Foster and C. Kesselman, *The Grid: Blueprint for a New Computing Infrastructure* , Morgan-Kaufmann, 1998.

[5] I. Foster and C. Kesselman, *The Grid 2: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann, 2004.

[6] Fran Berman, Geoffrey Fox and Tony Hey, Grid Computing: Making the Global Infrastructure a Reality, Wiley, http://www.grid2002.org/ , 2003

[7] I. Foster, C. Kesselman, S. Tuecke. "The Anatomy of the Grid: Enabling Scalable Virtual Organizations". *International J. Supercomputer Applications*, 15(3), 2001, pp200-222

[8] I. Foster, C. Kesselman, J. Nick, S. Tuecke, "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration", http://www.Globus.org/research/papers/ogsa.pdf, January, 2002.

[9] http://www.doesciencegrid.org

[10] http://www.eurogrid.org

[11] http://eu-datagrid.web.cern.ch/eu-datagrid

[12] Watts, Duncan J., and Steven H. Strogatz, "Collective dynamics of `small-world' networks", *Nature* 393, 1998, pp. 440-442

[13] Albert-Laszlo Barabasi and Reka Albert, Emergence of Scaling in Random Networks, Science, October 1999, pp. 509-512

[14] Andrei Broder, Ravi Kumar, Farzin Maghoul etc, "Graph structure in the web", *Computer Networks* 33, 2000, pp.309-320

[15] SH Strogatz, "Exploring complex networks", *Nature*, 410, 2001, pp.268-276

[16] R. Albert, and A.-L. Barabasi, "Statistical mechanics of complex networks", *Review of Modern Physics*, 74, 2002, pp.47-91

[17] Mark EJ Newman, "The structure and function of complex networks", *SIMA Review*, 45(2), 2003, pp.167-256