

Towards Topology-and-Trust-Aware P2P Grid

Yingjie Xia

Hangzhou Institute of Service Engineering, Hangzhou Normal University, Hangzhou, P.R. China
xtommy@163.com

Mingzhe Zhu and Yang Li

Hangzhou Institute of Service Engineering, Hangzhou Normal University, Hangzhou, P.R. China
zhumingzhe.cn@gmail.com, liyangtom@163.com

Abstract—Grid and P2P are both popular distributed network models nowadays. The grid is a platform to compactly coordinate the distributed resources, while the P2P model assigns an equal position to each peer and allows all the peers to move freely. Because of their different architectures, these two models have their own advantages and limitations, and therefore are used in different applications. In this paper, we propose a novel distributed network model called P2P Grid which hierarchically combines the grid and P2P. The P2P Grid is constructed based on the Globus Toolkit middleware and SW-R2P network topology, in order to absorb their respective advantages, especially in the trust based access control. We also describe the resource location algorithm and peer operation protocols for this SW-R2P based P2P Grid. Finally the simulated experiments show that our P2P Grid model has a great performance not only in efficiency and security, but also in scalability and robustness.

Index Terms—Grid, P2P, distributed network, Globus Toolkit, SW-R2P

I. INTRODUCTION

The grid is a distributed network model which integrates the distributed heterogeneous resources as a super computer. It can be used for various applications, such as scientific computation, data storage, equipment sharing, etc [1]. The grid is also regarded as a resource management system with centralized organization and cooperation. According to the traditional grid architecture and its functions, it has the core features of stability and security, while losing the performance on flexibility and scalability due to its centralized management. Recently, a new paradigm called cloud is proposed for the applications of large scale storage and elastic computing, such as Amazon S3 and EC2. Cloud can achieve the flexibility and scalability by redundant distributed resources, which needs a large amount of additional cost and is completely different from the traditional grid architecture.

Another prevalent distributed network model is Peer-to-Peer (P2P). In P2P networks, each peer is not only a resource consumer, but also a resource provider [2].

Through resource location algorithms, the consumer selects the most appropriate providers from the candidates to download the required resources. The P2P model has the advantages of robustness and flexibility, whereas the anonymous feature of each peer and the loose management of peers address great challenges on the network stability and security.

For both grid and P2P, they try to reach a tradeoff on the features of flexibility and stability. In this paper, we propose a novel distributed network model which hierarchically combines the grid and P2P. This model, which is called P2P Grid, extends the resource management of grid, and employs P2P to quickly locate the required resources. In the implementation, we modify the Globus Toolkit [3] as the grid component, and use SW-R2P [4] system as the P2P component. SW-R2P is a trusted small world overlay P2P network with Bayesian trust model and small world topology. Therefore, besides seamlessly combining the advantages of both P2P and grid to make up their respective disadvantages, our P2P Grid also has the features of node trust and network topology inherited from SW-R2P. The trust features help to locate the most appropriate nodes which meet the requirements specified in the submitted tasks. The small world topology improves the efficiency of node location algorithm and its relevant protocols.

The remaining parts of this paper are organized as follows. Section 2 reviews the related work about P2P, grid and P2P Grid. Section 3 describes the implementation of P2P Grid model and the related protocols in detail. The simulated experiments and corresponding analysis are specified in Section 4. Finally, we draw a conclusion with a summary of the contributions and propose some future research plans in Section 5.

II. RELATED WORK

There existed some research work on P2P Grid. Talia and Trunfio [5] compared the features of grid and P2P, and suggested a synergy between them. It only gave a rough description about using P2P network to integrate small grids, however without any specific design and implementation details, such as the node location algorithm, node operation protocols, etc. The design of this architecture also has some shortcomings, such as

Manuscript received December 1, 2009; revised December 21, 2009; accepted March 24, 2010.

Corresponding author, Yingjie Xia, Email: xtommy@163.com.

using P2P network to combine small grids. Since P2P network is an unstable structure, one P2P element lost just equals to a group of grid nodes lost, and a group of malicious nodes can invade network through gathering themselves as a grid and joining the P2P network. This architecture is too volatile for P2P grid. As for the research in topology, Ye et al. [6] proposed a P2P Grid model through simulating a bank structure. Peers were organized into P2P groups, just like the bank branches. All the P2P groups were centrally administrated by some super peers, therefore just the same as grid structure and losing the flexibility of P2P. All the peers in each P2P group are organized as a hierarchical tree whose node operation protocols are completely different from the protocols of P2P network.

As for the research in implementation, Pallickara and Fox [7] designed NaradaBrokering which is a distributed middleware framework integrating the computational grid, P2P network and other message-oriented middlewares. Heine et al. [8] implemented a semantic resource discovery by using ontology in his P2P Grid. This paradigm disseminates centralized resource description and matching to nodes as P2P structure. Both the models of NaradaBrokering and ontology-oriented focused on locating the resource providers, without considering any trust strategy to select the most appropriate provider from the candidates. Therefore, the node location algorithms and node operation protocols of both work have no implementation features of trust, and are quite different from our implementation.

Recently service oriented architecture (SOA) has been adopted in the research fields of distributed systems. Fox et al. [9] implemented an architecture with grids controlling central services while “services at the edge” were grouped into less organized “middleware P2P groups”. The hierarchy-structured grid service deepened the whole network and weakened its capability to locate services. Through inserting a connectivity layer in the standard grid architecture, Uppuluri et al. [10] designed a service-oriented P2P Grid framework which could provide better flexibility without compromising on efficiency. However, they use the unstructured P2P model, Gnutella, to implement the connectivity layer which uses the overloaded flooding algorithm for the node location.

Furthermore, all the related work introduced above focuses on the topology of network, but neglects the significant trust evaluations of peers which affects the quality of services provided.

III. IMPLEMENTATION OF P2P GRID

A. Architecture

The architecture of P2P Grid is shown in Fig. 1. All the nodes are categorized into two types, grid nodes and P2P nodes. The grid nodes are harnessed by the Globus Toolkit (GT) which is a middleware to orchestrate distributed grid resources such as computing cycles, data, storage, etc. The grid component of P2P Grid can provide the basic grid services, such as task management,

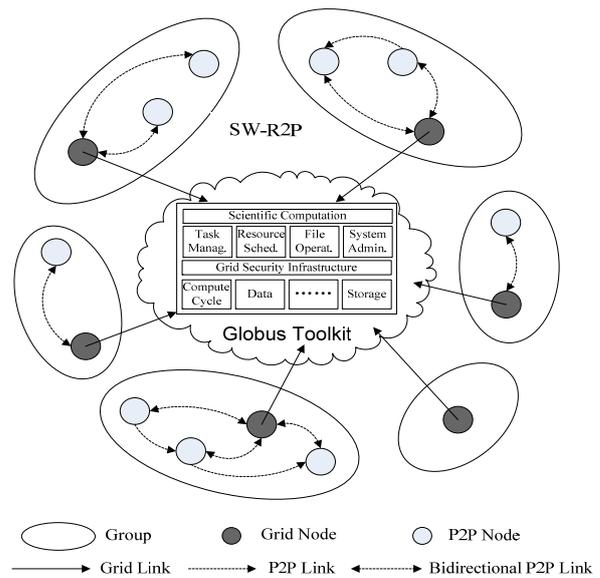


Figure 1. The P2P Grid architecture based on GT and SW-R2P.

resources scheduling, file operation and system administration. The P2P nodes are not involved in this grid component. According to the SW-R2P topology, all the nodes are divided into several peer groups which are orchestrated together through the grid component. There are also three types of links, including grid link, P2P link and bidirectional P2P link, which are categorized by their different roles in the network.

In our P2P Grid, each peer group is regarded as one virtual organization (VO) of grid. The nodes in one group hold the certificates signed from the same Certificate Authority (CA). All the nodes in one group are linked with each other by P2P links or bidirectional P2P links. The links between grid node and P2P node are bidirectional, while the links between P2P nodes are unidirectional. The grid nodes of all the groups are gathered by grid links to implement the grid component. Different links have different functionalities which will be elaborated in the following resource location algorithm and nodes operation protocols.

Our P2P Grid is based on the SW-R2P network, which uses zero knowledge interactive proof (ZKIP) [11] and Bayesian trust model [12] to implement group identity verification and peer multi-faceted trust evaluation respectively. ZKIP supports peers to identify each other in a non-forgable manner. It serves as group identity authentication for constructing the small world topology of our P2P Grid. Bayesian trust model is adopted in our implementation which improves the traditional unilateral trust models and comprehensively evaluates different combinations of peer trust metrics. We use a tree structure of trust metrics as the Bayesian trust model, which consists of various hardware resource conditions and transfer speed. The hardware resources include CPU cycles, memory, disk space, etc. which are important requirements specified in task to run the computational programs. The transfer speed is also a key trust metric to evaluate the network between two nodes. These metrics cover all the necessary requirements for the computation

environment in the tasks. In our P2P Grid architecture, whether or not to build a grid link between two grid nodes is determined by the linking probabilities calculated from Bayesian trust instances held by source grid node toward destination grid node.

In our P2P Grid, the task submitted by the P2P node will be finally submitted to the network via the grid node of its group. We do not need to specify the locations of the computing nodes in the task. In each task description, there are only some requirements for target nodes, such as CPU cycles, memory, etc. P2P Grid supports to find the most appropriate nodes meeting the requirements through the resource location algorithm. Furthermore, the joining and leaving of nodes are flexible in our P2P Grid, which is different from the standard grid paradigm that costs a large amount of efforts to maintain the whole structure. All the protocols for the corresponding operations above are described in the next subsection.

The grid middleware, Globus Toolkit, is modified to combine grid and P2P compactly. Firstly, we modify the resource status collector, monitoring and discovery service (MDS), to collect the node trust values, such as the hardware resources of some other nodes, transfer speed, etc. After each task finishes, the node which submits the task makes a new instance of Bayesian trust model for each node which executes the task. According to the requirements specified in the submitted task, the grid resource allocation management (GRAM) chooses the most appropriate nodes based on the trust values which are maintained by MDS. The list of chosen nodes and their corresponding certificate information are transferred to the grid security infrastructure (GSI). GSI adds the certificate subjects of all chosen nodes to the temporary MAPFILE for access control during the task submission. GRAM also transfers the node list to mpich-G2 which can construct a parallel environment for the task execution.

B. Resource Location Algorithm

In the traditional grid, although it always makes an advertisement to provide a transparent computing environment without much care from users to execute a task, it still can only be implemented to specify the exact node for computation in a task. This should not be the ideal grid we expect. Our goal is to design a model which can automatically and accurately locate the required resources. As a key component of P2P Grid, the resource location algorithm is improved to adapt to the structure. Since there are two types of tasks, program involved and not involved, we need to analyze their corresponding algorithms respectively.

(a) Computational Task with Program

If there is no required computational program in the whole network, the task must include the computational program, and requests the nodes with qualified capabilities for computations. In P2P Grid, different nodes can have different performances on running the same task. Thus, for the computational task with program, the most important effort is to locate the nodes which fulfill the requirements most.

Assuming node *i* submits a task with the computational program, it has the requirements for the target node with its CPU at “VeryLow” level and transfer speed (TS) greater than “Fast” level. All the levels are defined in the similar strategy as SW-R2P. Node *i* first checks its held trust instances for the directly linked nodes, to find the nodes whose conditional probability $p(TS=“Fast”||“VeryFast”, CPU=“VeryLow”)$ is greater than the threshold value P_t . The probability value for Node *i* to Node *j* can be calculated through dividing the number of Bayesian instances, which Node *i* keeps for Node *j* and meets the requirements, by their total number of Bayesian instances. The nodes which have higher probabilities to meet the requirements will be selected as the candidates. Otherwise, the search scope is extended to other groups through the grid node of *i*. Based on the conditional probabilities of candidates, we use the Bayesian Decision Algorithm to determine the final one. If the search hops exceed the depth limit d_t , it will be regarded as there is no fulfilling resources in the network.

The pseudocode of the resource location algorithm for computational task with program is specified as follows.

```

Resource Location Algorithm for task with Program (RLAP)
INPUT: i =Node submitting computational task
           c =Conditions set in computational task
OUTPUT: n =Selected resource node

PROCEDURE:
1 if (nList = Group_Lookup(i, c,  $p_t$ ))
// Node i searches its group under the condition c with
// the probability threshold  $P_t$ 
2   n = BayesianDecision(nList)
// Use Bayesian Decision Algorithm to select node
3 else // There is no node meeting the requirement
4   if (Identity(i) == "Grid") // Node i is Grid node
5     if (nList = Grid_Lookup(i, c,  $p_t$ ))
// Some nodes linked with i meeting the requirement
6       n = BayesianDecision(nList)
7     else
// No node linked with i meeting the requirement
8       forall k : all node linked with i
9         n = RLAP(k, c)
10      endfor
11  else // Node i is P2P node
12    j = GridPeer(i)
// Node j is Grid node of i's group
13    n = RLAP(j, c)
// Recursively search nodes from node j
14 Output n
    
```

In the pseudocode, the function BayesianDecision is the implementation of Bayesian Decision Algorithm. Assuming that $\{c_1, c_2, \dots, c_n\}$ is the set of candidate nodes, $\{\partial_1, \partial_2, \dots, \partial_n\}$ stands for the selection of corresponding node. If we use prior probability $P(c)$ to make decision, the error probability is calculated as $P(error) = \min\{1 - P(c_1), 1 - P(c_2), \dots, 1 - P(c_n)\}$ which does not meet the requirements. We use Bayesian Decision Algorithm to improve it. Suppose x is a light weight random variable, $P(x | c_j)$ is the class-conditional probability density representing the probability of x given the candidate c_j . In addition, $\lambda(\partial_i | c_j)$ is defined as the error probability of ∂_i when c_j is predicted. Based on these definitions, the posterior probability can be calculated through Bayesian theory, $P(c_j | x) = P(x | c_j) * P(c_j) / P(x)$, where

$$P(x) = \sum_{j=1}^n P(x | c_j) * P(c_j). \text{ The conditional risk for the}$$

candidate c_j is $R(\partial_i | x) = \sum_{j=1}^n \lambda(\partial_i | c_j) * P(c_j | x)$, which will be compared to select the minimum as the final one.

(b) Computational Task without Program

If there exists the required computational program in the computation nodes of P2P Grid, the submitted task does not need to include the program, but must specify the name of the program and its parameters. Besides, the task also sets the requirements for the nodes, like CPU cycles, memory, etc. The resource location algorithm for this kind of task is mainly to locate the nodes which not only hold the specified executable program but also meet the requirements described in the task.

Since the underlying SW-R2P network of P2P Grid is structured and each node maintains a Distributed Hash Table (DHT) for programs, the algorithm first checks the DHT of the task submitter, and tries to find the qualified nodes from the neighbour nodes directly linking with submitter. If there is no satisfying node, the algorithm will extend the search scope in the similar method as the task with program described above.

C. Node Operation Protocols

The node operation protocols of P2P Grid are implemented based on the underlying SW-R2P network, under the events of node joining, leaving, moving and failing.

(a) Node Joining Protocol (NJP)

When one node asks for joining the P2P Grid, it should specify one peer group as its target. The role of new node in the group is determined based on the evaluation of node reputation. If the specified group does not exist, a new group will be created and the node will act as its grid node. The grid links, P2P links and bidirectional P2P links in the group are adjusted according to the role of new node.

(b) Node Leaving Protocol (NLP)

When one node leaves from its group, the only thing it should do is to find a substitutor. The substitutor should be the node in the same group, and has the closest reputation to the leaving node. The reputations of all the nodes in the group are calculated, and compared with the reputation of leaving node to determine the substitutor. If the leaving node is the only node in the group, the group will be removed from the network.

(c) Node Moving Protocol (NMP)

The node moving protocol is used for the node moving from one group to another group. It is simply implemented through the combination of NJP and NLP, as leaving its original group and joining a new group.

(d) Node Failing Protocol (NFP)

The difference between the node failing and node leaving is that the former event happens accidentally without any notification in advance. Therefore, the node failing protocol has to try the best to get back the history data. Just like the SW-R2P, the P2P Grid periodically backups the interaction records in a trust center. If one node fails, the group can use its stored records to determine the substitutor and restore important links.

IV. SIMULATION EXPERIMENTS AND ANALYSIS

A. Simulation Setup

The whole simulation environment is set up by 200 personal computers (PC). We choose 20 PCs as the grid nodes, and install the modified Globus Toolkit in them. In the other PCs, through using different network ports to install multiple SW-R2P middleware on one PC, each PC can simulate multiple P2P nodes. The resource location algorithm and node operation protocols of P2P Grid are implemented by some minor modifications on the corresponding implementations of SW-R2P middleware. The whole simulation network is divided into 20 groups, and each group is managed by one grid node.

By looking up the required resources and running the computational programs, the experiments test the efficiency of resource location algorithm, robustness and scalability of P2P Grid. The purpose of the simulation experiments is to prove that P2P Grid can combine the advantages of both grid and P2P structures, achieving a safe, efficient and scalable infrastructure for scientific computations.

B. Resource Lookup Experiment

This experiment measures the average number of lookup hops between two peers. We simulate the network with the size from 1000 to 10000. Each peer holds a distinct resource object, and looks up a random object for 10 times to calculate the average hops. The P2P Grid is compared with Chord and SW-R2P. Chord is implemented by using the Chord Simulator from University of Iowa. We modify it in the network implementation in order to adapt to our test environment. In SW-R2P, the maximum number of long links for each exterior peer is 10. Fig. 2 illustrates the average hops of random resources lookup in different sizes of networks.

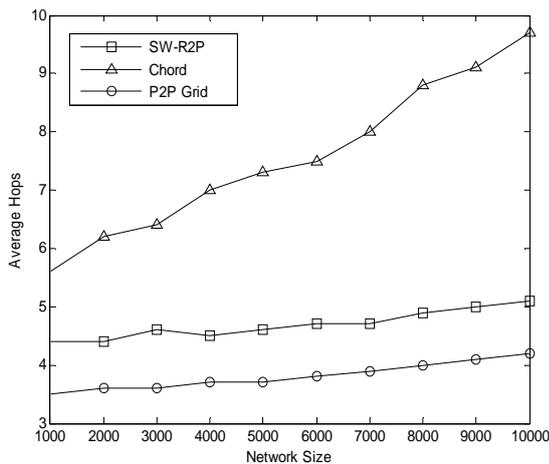


Figure 2. Average hops of random resource lookup in experiment setup 1

We can find that from 1000 to 10000 nodes the average hops of P2P Grid and SW-R2P keep stable around 4 and 5 respectively, while the average hops of Chord rises rapidly from 6 to 10 as the network size increases. The results show that P2P Grid and SW-R2P have much better performance than Chord in resources lookup because of their small world topology. In this simulation experiment, P2P Grid performs better than SW-R2P because there are 19 grid links for each grid node in P2P Grid which are much more than the maximum number of long links for each exterior peer in SW-R2P. The more grid links make each group have more neighbors, therefore shortening the average distance between nodes in different groups. This experiment indicates that P2P Grid inherits the advantages of SW-R2P in resource lookup.

We also adjust the experiment setup by randomly selecting each ten peers of simulated network to hold same resource object. The other setups keep the same. The experimental results are shown in Fig. 3.

Through comparing Fig. 3 with Fig. 2, the more copies of resources exist in the network, the fewer average hops are taken by all the test cases. The average hop difference of Chord between two experiment setups is around 1, while the other two small world based models do not make such big progress. The reason is that the resource copies can increase the probability to locate the required resource. However, in small world network topology, if some of the copies exist in the same group, they can not increase the probability of resource lookup. When the experiment achieves extreme that all the nodes hold the same resource, the three test cases have the same average hop. From these two experiments and the extreme, we can infer that P2P Grid always performs better than Chord in the required resource lookup.

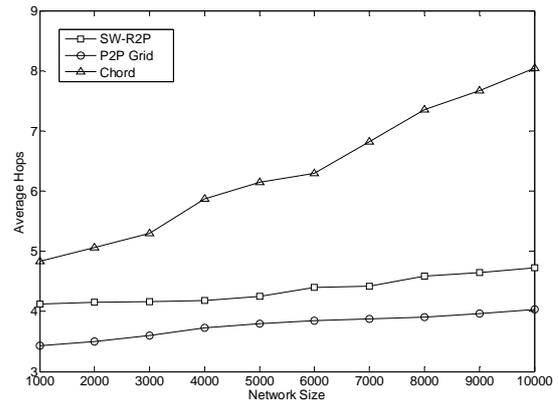


Figure 3. Average hops of random resource lookup in experiment setup 2

C. Robustness Experiment

The users mainly use P2P Grid for scientific computations, especially for parallel computational tasks. The robustness of P2P Grid infrastructure influences not only the efficiency of computations, but also the correctness of results. In this experiment, the test cases are three MPI based programs, PI, Poisson equation and crank simulation. We run them with and without the interference of malicious nodes in P2P Grid, and compare the computation time and resulting correctness. The behaviors of malicious nodes include faking identity to join network and deliberately failing to disturb executions. The percentage of malicious nodes in all the nodes is 10%. Since we need to run MPI programs as test cases, the environment is set up by real computers instead of simulated nodes. We deploy PI, Poisson and crank simulation on 10, 20 and 150 nodes respectively, and repeat to run them 200 times to get the average. The computation time and resulting correctness are shown in Fig. 4 and Fig. 5.

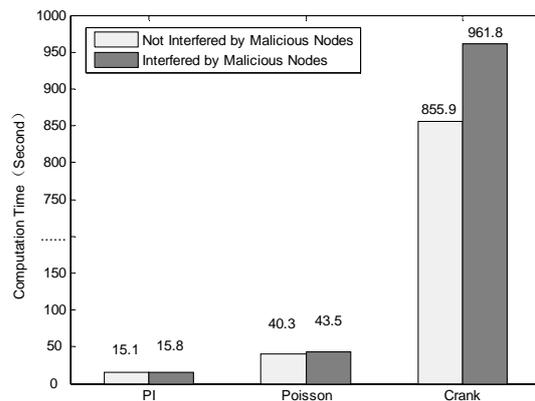


Figure 4. Computation time of test cases in P2P Grid

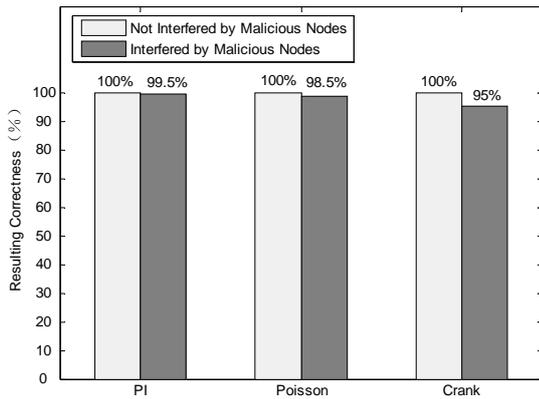


Figure 5. Resulting correctness of test cases in P2P Grid

From the experiment, we can find it takes a little more computation time to run these three test cases in the network with malicious nodes than without malicious nodes. As more nodes used for computations, the time difference between two kinds of network becomes greater. This is for the reason that more nodes involved imply more probability to interact with malicious nodes. Since P2P Grid uses X509 certificate as user identity, it can eliminate the attacks by faking identities. The resulting correctness of test cases are all more than 95% because the Bayesian trust model of underlying SW-R2P facilitates to interact with more reliable nodes and leave malicious nodes far away. This experiment shows the robustness performance of P2P Grid for scientific computations.

D. Scalability Experiment

This experiment tests the scalability of P2P Grid by setting nodes to join and leave on purpose. The environment setup and test cases are similar to experiment 2, and the experiment conditions are 10% nodes leaving, normal and 10% new nodes joining. Under these three conditions, the resulting correctness of all test cases is 100%, and we compare the average computation time which are shown in Fig. 6.

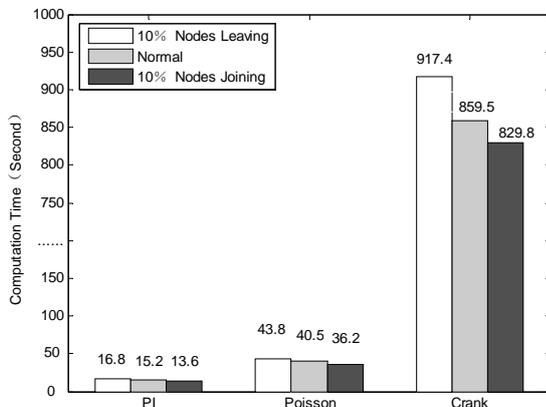


Figure 6. Computation time of test cases in P2P Grid under three conditions

Through adding or removing nodes, the average computation times of PI and Poisson decrease and increase proportionally. This is due to the few communications between nodes in these two test cases. However, because of a large amount of communications for the parallel computation, the corresponding decreasing time and increasing time of crank simulation are 7% and 4% respectively. The experiment results show that P2P Grid performs a good scalability to run the parallel programs with few communications between nodes.

V. CONCLUSION AND FUTURE WORK

In this paper, we propose a novel distributed network based on the hierarchical combination of grid and P2P. This network model, called P2P Grid, uses SW-R2P as the underlying P2P component and the Globus Toolkit as the grid component. Two resource location algorithms in P2P Grid are implemented according to two different kinds of tasks, computational task with program and without program. The node operation protocols are implemented based on the corresponding protocols in SW-R2P. The simulation experiments show that P2P Grid can achieve high efficiency, robustness and scalability for parallel scientific computations.

In the future work, we suggest exploring some cutting edge research work to strengthen the P2P Grid model. First we plan to add more aspects to the Bayesian trust model in SW-R2P, and adjust their corresponding weights in order to build a more realistic trust evaluation system. Then we will try to apply P2P Grid to some specialized areas, such as Computational Fluid Dynamics, Bioinformatics, etc. These are practical tests for P2P Grid. Finally, in order to make the whole model more concise and efficient, we will try to optimize the grid component of P2P Grid by reducing some unnecessary parts of GT.

ACKNOWLEDGMENT

The authors thank a project supported by Scientific Research Fund of Zhejiang Provincial Education Department under grant number Y200907427. We also thank the support from the Ministry of Commerce of the People’s Republic of China, Hangzhou Government and Hangzhou Normal University. The authors do appreciate the helpful discussions among the faculty of Hangzhou Institute of Service Engineering and Staff in Center for Service Engineering, and would like to thank them for their assistance in this project.

REFERENCES

- [1] I. Foster, C. Kesselman and S. Tuecke, “The Anatomy of the Grid: Enabling Scalable Virtual Organizations,” *International Journal of High Performance Computing Applications*, vol. 15, pp. 200-222, 2001.
- [2] A. Oram, *Peer-to-Peer: Harnessing the Power of Disruptive Technologies*. O’Reilly Media, 2001.
- [3] I. Foster and C. Kesselman, “Globus: A Metacomputing Infrastructure Toolkit,” *International Journal of Supercomputer Applications*, vol. 11, pp. 115-128, 1997.

- [4] Y. Xia, G. Song and Y. Zheng, "SW-R2P: A Trusted Small World Overlay P2P Network with Zero Knowledge Identification," *Journal of Computers*, vol. 3, pp. 3-11, 2008.
- [5] D. Talia and P. Trunfio, "Toward a Synergy between P2P and Grids," *IEEE Internet Computing*, vol. 7, pp. 94-96, 2003.
- [6] C. Ye, W. Jiang and S. Sun, "The Combine of P2P with Grid: Research of P2P-Grid Model," *Microcomputers and Its Applications*, vol. 24, pp. 4-6, 2005.
- [7] S. Pallickara and G. Fox, "NaradaBrokering: A Distributed Middleware Framework and Architecture for Enabling Durable Peer-to-Peer Grids," in *Proc. ACM/IFIP/USENIX International Middleware Conference*, pp. 41-61, 2003.
- [8] F. Heine, M. Hovestadt and O. Kao, "Towards Ontology-Driven P2P Grid Resource Discovery," in *Proc. IEEE/ACM International Workshop on Grid Computing*, pp. 76-83, 2004.
- [9] F. Berman, G. Fox, A. Hey, *Grid Computing: Making the Global Infrastructure a Reality*. Chichester, UK: Wiley, 2003.
- [10] P. Uppuluri, N. Jabisetti, U. Joshi and Y. Lee, "P2P Grid : Service Oriented Framework for Distributed Resource Management," in *Proc. IEEE International Conference on Services Computing*, pp. 347-350, 2005.
- [11] S. Goldwasser, and C. Rackoff, "Knowledge Complexity of Interactive Proof Systems," *SIAM Journal on Computing*, vol. 18, pp. 186-208, 1989.
- [12] Y. Wang and J. Vassileva, "Bayesian Network Based Trust Model," in *Proc. IEEE/WIC International Conference on Web Intelligent*, pp. 249-279, 2003.

Yingjie Xia was born in Fenghua, Zhejiang Province, P.R. China on November 7th, 1982. Now he is an instructor in Hangzhou Institute of Service Engineering, Hangzhou Normal University, Hangzhou, Zhejiang, P.R. China. He is also affiliated as assistant director of Center for Service Engineering in Hangzhou Normal University. He was sponsored by China Scholarship Council as a visiting scholar in National Center for Supercomputing Applications (NCSA), University of Illinois at

Urbana-Champaign (UIUC), USA. He received his Ph.D. Degree in the College of Computer Science, Zhejiang University in 2009. His major is computer science and technology.

His research interests cover distributed computing, grid computing and P2P network. He deployed the Zhejiang University Campus Grid in 2006 by hierarchically combining the Globus Toolkit and Sun Grid Engine. This work is summarized and published in the proceeding of APWeb, Harbin, 2006. And he also implemented a R2P system, P2P system with role and reputation based access control. The relative paper about this system is published in the proceeding of First International Workshop of Knowledge Discovery and Data Mining, Adelaide, Australia, 2008. His work focused on to improve the R2P system and proposed a SW-R2P model, a trusted small world P2P network in Zhejiang University. Now he tries to combine two different distributed network model, P2P and Grid, and implements a paradigm called P2P Grid which absorbs their both advantages.

Mr. Xia has published 13 papers as the first author. He got a public patent of his R2P system. He will apply for the patent of the SW-R2P system in China.

Mingzhe Zhu is a Research Assistant in Center for Service Engineering, Hangzhou Normal University. includes the biography here. He received the Bachelor degree from the Zhejiang University in 2008. He specializes in P2P, distributed computing and cloud computing.

Yang Li is an associate professor in Hangzhou Institute of Service Engineering and Center for Service Engineering, Hangzhou Normal University. He is also serving as the assistant director of the department of computer science in the Institute of Service Engineering. He received his Ph.D. degree from Zhejiang University. He has published more than 15 papers in network security, sensor network and distributed network.