

ResourceDog: A Trusted Resource Discovery and Automatic Invocation P2P Framework

Bowei Yang, Guanghua Song, Yao Zheng

College of Computer Science, and Center for Engineering and Scientific Computation,
Zhejiang University, Hangzhou, 310027, P. R. China
boweiy@zju.edu.cn, ghsong@cs.zju.edu.cn, yao.zheng@zju.edu.cn

Abstract. In this paper we propose a trusted resource discovery and automatic invocation P2P framework. It provides a series of amazing features to overcome the drawbacks of existing P2P frameworks, such as totally decentralized structured and semantic searchable topology based on dual level DHT network hierarchy, distributed metadata storage, unified resource abstraction and trusted recommendation system based on statistic model of reputation. The ResourceDog framework provides a mechanism to discovery resources by tags, prepare runtime environment for resources automatically, gather and model out rank of rating for resources. The results demonstrate that the ResourceDog framework provides convenient, powerful and trusted resources for general users in P2P environment.

Keywords: P2P, dual level DHT, resource abstraction, reputation system

1 Introduction

1.1 Motivation

In the recent years, Peer-to-Peer (P2P) computing has gained a lot of popularity in many distributed applications over the Internet. These include distributed file sharing [5, 6], high performance computing [19] and so on.

In this paper we discuss a framework, called ResourceDog, for unifying the most important features of grid computing and P2P computing into a common computing framework.

Currently, grid computing is mainly providing services for scientific computing, and grid frameworks usually focus on providing highly reliable services on small scale but stable infrastructures [20]. On the contrary, P2P frameworks manage larger scale but not so devoted nodes. Not only scientific computing but also general computing needs a relatively unified, stable, flexible, reliable and high-powered framework. However most of existing popular P2P systems only cover file sharing and all of them are obsessed with single point of failure, non-supported to various resources or unable to identify malicious nodes. Original Bittorrent [5] or Emule [6]

needs a central server for seeds storing, Bittorrent protocol with DHT [1, 2, 3, 4] technology can't support searching for resources, none of them implement service oriented architecture. Since all of the control information of the P2P network is stored in the central server, all of the services will be down if the central server is down. The ResourceDog framework enables the automatically computing services invocation based on P2P architecture. Furthermore, statistical recommendation feature is included to resist malicious behaviors and to encourage contribution behaviors.

The presented P2P framework includes resource discovery algorithm, resource utilize architecture and recommendation model based on reputation and ranking. The three main parts will be covered in this paper respectively.

1.2 Solution Overview

Axis [7] is introduced as the main engine of ResourceDog to take advantage of standard web services, thus all language compatible with standard WSDL [8] will be accepted as the compatible service provider language. Dual level DHT mapping combined with word splitting method is adopted to provide the classical DHT topology with semantic search feature. Semantic Web service [9] and ontology is pulled in for service automatic invocation, and an innovational positive feedback statistic model is drawn out to support recommendation which tries to simulate the scenario in our social network.

2 Design

First we describe our ResourceDog P2P framework in its most basic form, in the below subsections we present the details.

In a P2P environment, node A may interact with any nodes in the same network, we represent the transaction from node A to B as $T_{a,b}(TN)$, where node A is the initiator and node B is acceptor, TN refers the name for the resource on node B. Two types of transaction are included in the P2P environment, the first is control transaction, and the second is functional transaction. Control transaction exists for maintaining P2P structure, notifying newly added nodes, reputation rating storage and retrieval and so on; Functional transaction is customized by resource providers of ResourceDog, which includes but not limited to file sharing, scientific computing, image processing and instant chat. In addition, after every transaction $T_{a,b}(TN)$, node A should provides a rating for this transaction denoted by $R_{a,b}(TN)$. This rating is maintained by several nodes in the dual level DHT network. Before every transaction starts, the recommendation rating worked out by modeling all the ratings for a resource TN on target node X $R_x(TN)$ will be presented to the initiator to assist in making decision.

2.1 Routing and Searching in the ResourceDog Framework

A ResourceDog node maintains a coordinate routing table that holds the IP address, last visit time (LVT), and the unique identifier (UID) for every node that has contacted with itself or obtained from its neighbors. All of the routing data is stored in the Derby [10] embedded database, with index key UID for fast accessing and retrieving. Compared with the improvement to speed up resource searching, lots of rare occupied coefficient routing data is meritorious, so the routing data purging process could be started with a long period. In our approach, the expression of UID for node A listened on IP AA.BB.CC.DD is:

$$UID(N_a) = (AA \ll 24) | (BB \ll 16) | (CC \ll 8) | DD \quad (1)$$

In ResourceDog, every resource such as files or services will be described with a short phrase stored in the metadata of this resource. During the resource publish process, ResourceDog will split words of the phrase, filter them and publish all of the meaningful words to remote nodes, the same as what the general search engine does.

Our example framework is implemented with Java, however, the method `java.lang.String.hashCode()` is too simple to generate a random and well-proportioned distributed hash code, Algorithm 1 presents a proved suitable enough method to achieve this goal in our environment.

Algo. 1. The hashcode generate procedure.

```

A = SUM((Integer) (every four bytes of input word))
B = 16807 * (A % 127773) - 2836 * (A / 127773)
IF B > 0 THEN:
    Return B
ELSE
    Return B + 2147483647
ENDIF

```

In addition, the distance D_{ar} between one node N_a with $UID(N_a)$ and one resource R_r with hash code $HashCode(R_r)$ is:

$$D_{ar} = ABS(UID(N_a) - HashCode(R_r)) \quad (2)$$

In the resource publish process, ResourceDog will filter all the index words, and publish them to the M most nearest nodes as the 1st DHT indexes for searching to every word respectively. And N copies of metadata for the resource should be published to network as the 2nd DHT indexes for reputation integrity checking, too. Here, M and N are the parameters to balance the robustness and overhead of ResourceDog P2P framework.

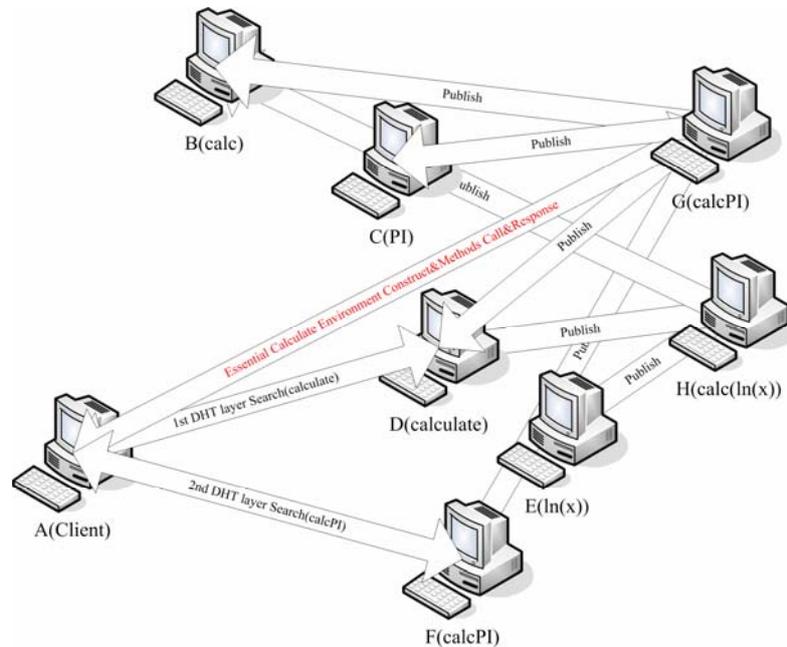


Fig. 1. An example of resource discovery

Figure 1 shows an example scenario. Node G is the provider of resource “calcPI”, and “calc(ln(x))” is provided by node H, G published 3 copies of 1st DHT layer indexes, “calc”, “calculate” and “PI” to node B, node D and node C respectively according to the most nearest publishing algorithm, 2 copy of 2nd DHT layer index, “calcPI” to node F and itself, also node H published 3 copies of 1st DHT layer indexes, “calc”, “calculate” and “ln(x)” to node B, node D and node E. As a client, node A provides “calculate” as the search key word, it checks its own nodes cache, and select M most nearest nodes to HashCode(“calculate”), node D is one of them. Node D will transfer to node A a list of resources which provide “calculate” as the 1st DHT layer search key word. In the demo shown in Figure 1, both of “calcPI” and “calc(ln(x))” provide “calculate” as the 1st DHT layer search key word, so “calcPI” and “calc(ln(x))” will be returned to node A, node A may select node F or node G for the details of “calcPI”. Then node A will compare the rating provided by node F with that provided by node G to check whether node G has been increased the rating by itself. If the rating on most of 2nd DHT layer nodes (says node F) linked with “calcPI” is different with that provided by the servant (says node G), the servant will be marked as cheating. At last node A get the host information (node G) of “calcPI”, and requests details from node G.

As shown in Figure 1, the reason that Node F holds “calcPI” as the 2nd DHT layer index is not only accelerating the search process, but also checking the integrity and authenticity of metadata for resources, for all of the recommendation data can be accessed by clients.

After every transaction, the two nodes will exchange their own nodes cache, and every node in the network will check their own cached resource indexes periodically,

if one of the resources is invalid or the entry is idle for some times, the index entry will be deleted. After every changing to the cache of a resource provider, it will calculate the remapping of the resource, if modification is required, it will republish the index of resource to network.

To a totally new participator, a list of bootstrap nodes is essential to join the P2P network. The functioning of ResourceDog framework doesn't depend on the details how this is done, but we adopt JXTA [11] as our bootstrap mechanism in the demonstration. The detailed demonstration will be shown in section 3.

2.2 Automatic Invocation

The ResourceDog framework provides a mechanism to access resources without any manual intervention. A resource will be defined as a mixture of file transfers and several remote method calls, simplified ontology is adopted to describe the structure of calling stubs. ResourceDog framework provides a session-like status machine routine to check whether the necessary precondition is satisfied. Lots of security mechanisms are implemented to enhance the framework, such as RSA asymmetric encryption and three-way handshake just like the same in TCP/IP protocol. All of the public keys will be treated as normal resources, they're stored and accessed by the same routine introduced in section 2.1.

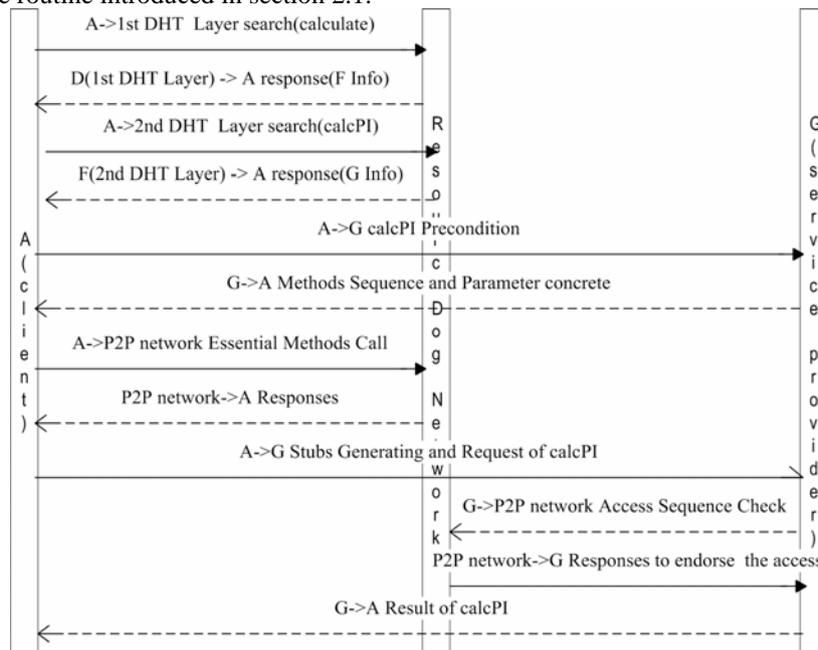


Fig. 2. An example resource automaton

A simplified resource call sequence demonstrated in Figure 1 is shown in Figure 2. All of the background transactions will be transparent to the end user.

2.3 Reputation and Recommendation

Currently, lots of P2P systems are receiving accusations that malicious nodes, free-ride [12] and bad behaviors are flooding over them, so security, trust, ranking and recommendation mechanism is urgently required by P2P systems. Various models [13, 14] are proposed to add rating or ranking feature to P2P systems, but few of them adopt statistical model to generate a recommendation rating for later comers to simulate the scenario in our social network.

Rating for transactions is widely used in modern auction websites, Ebay [15] will ask the clients to vote for the transaction they participated, 1 for “Positive”, 0 for “Neutral” and -1 for “Negative”, so that the later comers will know the ratios for all of “Positive”, “Neutral” and “Negative”.

ResourceDog will refine the result of the voting for resources. A statistic modeling will take all of the voting data as input and generate a recommendation ratio for later comers.

In our voting system, all of the voters will vote for a certain resource in their own ways. The vote rating will follow the Gaussian distribution as the voting goes on, since the voting behaviors are independent of each other. Figure 3 will show several probability and cumulative distributions for Gaussian distribution.

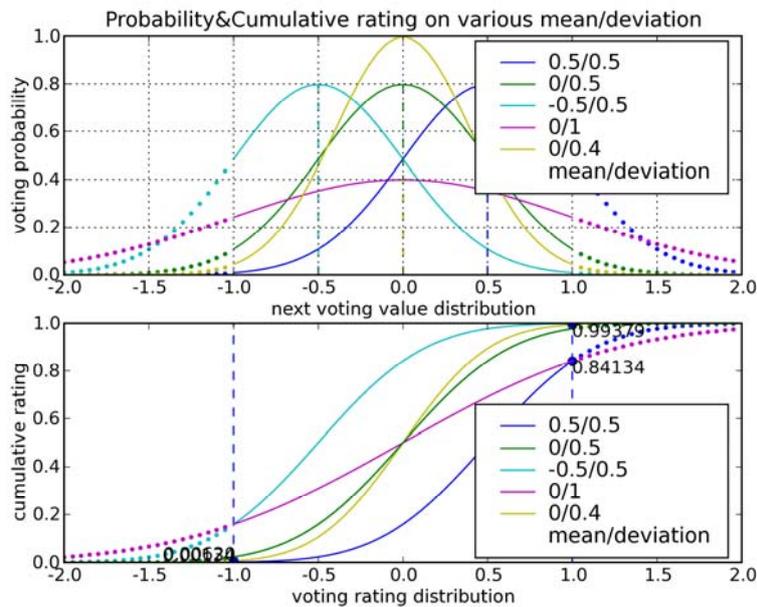


Fig. 3. Probability and cumulative distribution chart

The cumulative distribution function (*cdf*) can be formulated by an airy function called “Error Function” (*erf*).

Error function is used widely in probability, statistics, partial differential equations, and denoise-smoothing communication.

$$cdf(x) = \frac{1}{2} \left(1 + erf\left(\frac{x-\mu}{\sigma\sqrt{2}}\right) \right) \quad (3)$$

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \quad (4)$$

ResourceDog framework generates the recommendation rating for a certain resource using cumulative function denoted by error function.

The recommendation rating (*RRR*) for a certain resource *t* with total cumulative rating ($\sum_0^n R(TN_{x,t})$) will be:

$$RRR_x(\sum_0^n R(TN_{x,t})) = erf\left(\text{coefficient} * \frac{\sum_0^n R(TN_{x,t})}{n}\right) \quad (5)$$

After a voting of *Z* with voting rating for this transaction ($R(TN_{z,t})$), *RRR'* will be:

$$\begin{aligned} RRR_x' &= erf\left(\text{coefficient} * \sum_0^{n+1} R(TN_{x,t})\right) \quad (6) \\ &= \text{coefficient} * cdf'(x) \\ &= \text{coefficient} * \frac{1}{2} * \left(1 + erf\left(\frac{\sum_0^n R(TN_{x,t}) + R(TN_{z,t})}{\sigma'\sqrt{2}}\right)\right) \end{aligned}$$

ResourceDog will store every $R(TN_{x,t})$ of TN_t on the 2nd layer DHT network, since every 2nd layer DHT nodes is determinate by well-known Hashing algorithm, so that clients are able to check the integrity of recommendation of resources with different 2nd layer DHT nodes. Malicious nodes will be detected easily by the routine illuminated in figure 1.

3 Experiments

3.1 Routing and Searching

In order to demonstrate the usage of this framework and to evaluate the performance, we will give out several demos.

We built a local area network comprised of 30 PCs, all the PCs run different operating systems including Linux and Windows. They are divided into 4 100M

based fast ethernet network segments. All nodes are configured with a stable enough initializing seeding node.

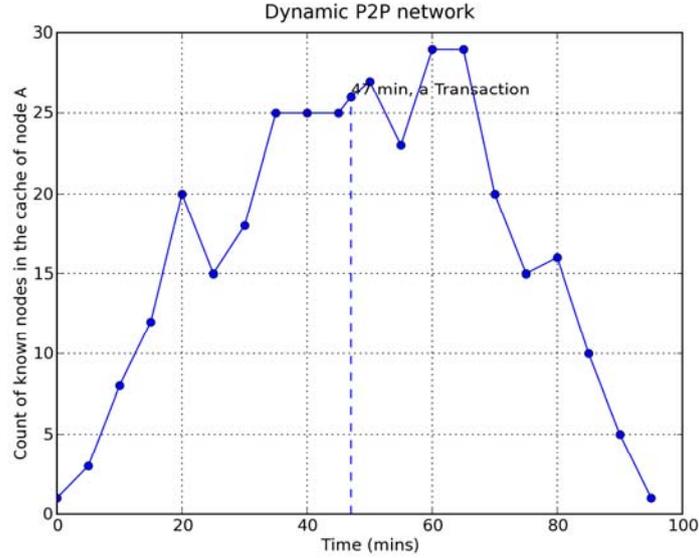


Fig. 4. Demonstrating for dynamic changing of our P2P network

A node in the network will collect neighbor nodes information every 5 minutes from all the known neighbors or exchange with buddies after every transaction. A node in figure 4 received a notification of neighbor changing at 47 minutes for a transaction occurred at that point.

The first demo we present is a simple numeric computing Message Passing Interface (MPI) [16] parallel routine, a general LAM-MPI [17] parallel process will be invocated by ResourceDog framework.

The numeration integration of Π can be presented by:

$$\Pi = \int_0^1 \frac{4}{1+x^2} dx \quad (7)$$

Π equals the area surround by the curve and x-axis on $[0, 1]$, so the area can be approximately calculated by lots of rectangles, in our demo, we divide $[0, 1]$ into 1,000,000,000 intervals, and accumulate all of their areas. The execution time comparison of pure LAM-MPI and LAM-MPI on the ResourceDog framework using variant number of nodes in a cluster is listed in Table 1.

Table 1. Execution time(s) comparison

Test platform	1 node	2 nodes	4 nodes	8 nodes	16 nodes
Pure LAM-MPI	50.921	25.461	12.748	6.388	3.204
Resource Dog	52.114	26.701	13.904	7.572	4.331

According to Table 1, Resource Dog P2P framework takes about one second to prepare the environment for MPI execution such as authentication and remote process calling in our demo.

3.2 Resource Automatic Invocation

Web service based on HTTP protocol is connectionless oriented, so real time resources can't be built on traditional web service platform. The second demo will introduce a new architecture for real time resources on ResourceDog. An Internet Relay Chat (IRC) [18] like chatting service has been implemented on our ResourceDog connectionless oriented framework.

In the chat service, nodes is grouped by the topic servers, ResourceDog servers can setup a topic service and publish the metadata to the DHT network as the way JXTA [11] publishes pipeline advertisements, others interested with the topic can join the channel, two actions will be taken during the joining process. First, the client create a socket with the local ResourceDog server, second, the local ResourceDog server registers itself to the topic server. After every receiving of messages, the topic server will scan the registered local servers list, and send the message to them one by one, then the local server will send the message to its client in time via local connection oriented socket. No connections will be established between topic server and the local servers. Thus real-time chatting is achieved on the connectionless framework.

3.3 Reputation and Recommendation Simulation

In this section, we will show the experimental result by a simulated demonstration.

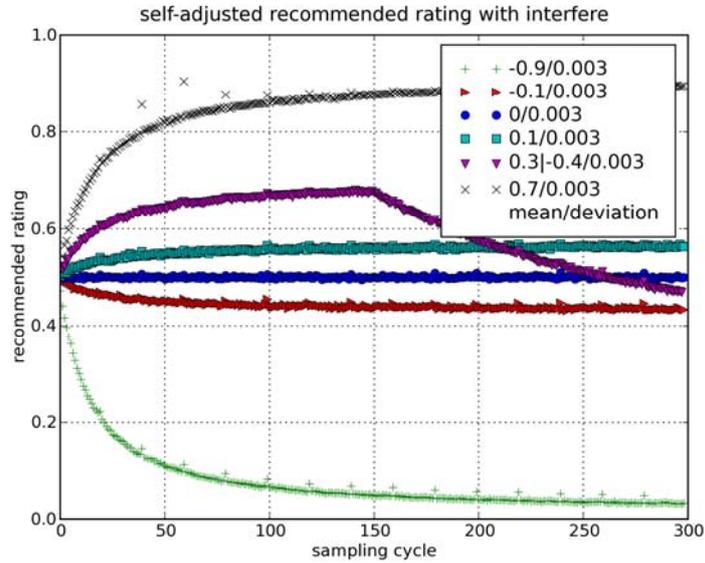


Fig. 5. Recommended rating after every voting

As shown in Figure 5, we simulate 300 users to rate for 6 resources respectively. Every user will vote for a resource at a normal distribution with certain mean value and standard deviation, also an interferential voting is included every 20 cycles. The simulated result shows a good-fit chart for true situation, since the recommendation rating accurately shows the cumulative effect of ratings and can be self-adjusted to anti-interference as the voting going on, moreover the resource will never be totally opponent or recommended.

4 Related Works

There have been some researches focusing on various aspects of P2P framework. Some interesting models and frameworks have been proposed. Here we describe a few related examples from two aspects, resource discovery and trust computing.

JXTA [11] community sponsored by Sun micro system mainly aims to build a wide area P2P network composed of all of smart devices. It provides a convenience and platform independent routine to organize all of nodes and access the demanded resource. However, as the core conception of JXTA framework, advertisements can't support semantic search, so JXTA framework does not overcome the search problems. Also JXTA framework never refers to reputation problems of nodes in the network.

In trust computing area, some researches just focus on the hierarchy of reputation system and reputation data storage strategy [21].

EigenTrust [22] only worked out the global reputation rating for nodes, but ignored ratings of resources, also it doesn't simulate the real society network to give out the recommendation rating based on statistical models.

In [23], Damiani et al. propose a distributed reputation sharing and assessing model, based on Gnutella. They developed a protocol for Gnutella to enhancement the security and reliability of nodes and resources in Gnutella by assigning, sharing, and combining reputations on servants and resources.

5 Conclusions and Future Work

In this paper, we discussed the architecture and algorithm of the ResourceDog P2P framework. ResourceDog provides a complete discovery, invocation and recommendation P2P platform that caters scientific computing, flexible computing and trusted computing. The experimental results show that ResourceDog is more stable, flexible and secure than most of existing P2P platforms. Applications on the Internet, such as file sharing, real-time chatting, and format transformation can be managed and rated properly under the framework. Therefore, we can utilize variant applications in one-stop framework and defend the malicious behaviors.

At present, only some typical types of resources have been tested on ResourceDog, we will try to deploy more types of resources in the framework. Furthermore, smarter searching with Semantic Web services and optimized routing algorithms will be adopted to adapt for the complex real-network.

6 References

- [1] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable contentaddressable network", in Proceedings of the ACM SIGCOMM, San Diego, CA, Aug. 2001, pp. 161–172.
- [2] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan, "Chord: A scalable PeerToPeer lookup service for internet applications", in Proceedings of the ACM SIGCOMM, 2001, pp. 149–160.
- [3] P. Druschel and A. Rowstron, "Pastry: Scalable, distributed object location and routing for largescale peertopeer systems", in Proc. of the 18th IFIP/ACM International Conference on Distributed Systems Platforms(Middleware 2001), Nov. 2001.
- [4] B. Zhao, J. Kubiatowicz, and A. Joseph, "Tapestry: An infrastructure for faulttolerant widearea location and routing", University of California at Berkeley, Computer Science Department, Tech. Rep. UCB/CSD011141, 2001.
- [5] Bittorrent URL: <http://www.bittorrent.org>
- [6] Emule URL: <http://www.emule-project.net>
- [7] AXIS URL: <http://ws.apache.org/axis>
- [8] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, "Web services description language (wsdl) 1.1 Technical report", <http://www.w3.org/TR/wsdl/>, 2001.
- [9] Erhard Rahm and Philip A. BernsteinDOI, "A survey of approaches to automatic schema matching", in The International Journal on Very Large Data Bases, 2001, pp. 334-350.
- [10] Apache Derby URL: <http://db.apache.org/derby>

- [11] Li Gong, "JXTA: a network programming environment", Internet Computing, IEEE, 2001, pp. 88-95
- [12] Adar, E., and Huberman, B. A., "Free Riding on Gnutella", 2000, Technical report, Xerox PARC
- [13] Karl Aberer and Zoran Despotovic, "Managing trust in a peer-2-peer information system", in Proceedings of the international conference on Information and knowledge management, 2001, pp. 310-317
- [14] Fatih Emekci, Sahin, O.D., Divyakant Agrawal and Amr El Abbadi, "A peer-to-peer framework for Web service discovery with ranking", in Proceedings of IEEE International Conference Web Services, 2004, pp. 192-199
- [15] Ebay URL: <http://www.ebay.com>
- [16] Message Passing Interface Forum, "MPI: A Message-Passing Interface Standard", 1994
- [17] Greg Burns, Raja Daoud, and J. Vaigl. "Lam: An open cluster environment for mpi", in Proceedings of Supercomputing Symposium, 1994, pp. 379-386
- [18] J. Oikarinen and D. Reed "Internet Relay Chat Protocol", <http://tools.ietf.org/html/rfc1459>, 1993
- [19] Abramson, D., Sasic, R., Giddy, J. and Hall, B., "Nimrod: A Tool for Performing Parameterised Simulations Using Distributed Workstations", in Proceedings of 4th IEEE Symp. on High Performance Distributed Computing, 1995, pp. 112-121
- [20] I. Foster and A. Iamnichi, "On Death, Taxes, and Convergence of P2P and Grid Computing", in Proceedings of the 2nd Int'l Workshop on Peer-to-Peer Systems (IPTP3'03), 2003, pp. 118-128
- [21] D. Dutta, A. Goel, R. Govindan, and H. Zhang, "the Design of a Distributed Rating Scheme for Peer-to-Peer Systems," 1st Workshop on Economic Issues in P2P Systems, 2003.
- [22] S. Kamvar, M. Schlosser, and H. Garcia-Molina, "The Eigentrust Algorithm for Reputation Management in P2P Networks", in Proceedings of 12th ACM World Wide Web (ACM WWW), 2003, pp. 640-651
- [23] E Damiani, S Paraboschi, P Samarati and F Violante, "A reputation-based approach for choosing reliable resources in peer-to-peer networks", in Proceedings of the 9th ACM conference on Computer and communications security, 2002, pp. 207-216