An Information Retrieving Protocol for Resource Monitoring in Grid Computing

Wu-Chun Chung[1], Ming-Huang Guo[2], and Ruay-Shiung Chang[1]
[1]*Department of Computer Science and Information Engineering, National Dong Hwa University*
[1]*rschang@mail.ndhu.edu.tw*
[2]*Department of Information Management, Shih Hsin University*
[2]*mhguo@cc.shu.edu.tw*

ABSTRACT

To manage the large scale of resources with dynamically access in Grid computing, resource management is a key component. In this paper, a Grid Resource Information Monitoring (GRIM) prototype is introduced to accomplish the task. To support the constantly changing of resource states in the GRIM prototype, the push-based data delivery protocol named Grid Resource Information Retrieving (GRIR) is provided. The offset-sensitive mechanism, time-sensitive mechanism, and hybrid mechanism in GRIR are used to achieve a high degree of data accuracy with the less useless update messages. Experimental result shows that the proposed mechanism not only alleviates the update transmission but also achieves the less loss of data accuracy than the prior work.

1: INTRODUCTIONS

A Grid network topology has many Virtual Organizations (VOs) [4]. A VO is named that members are composed of disparate groups of organizations and/or individuals. Grid resources are inherently scattered over the wide-area network. It is obviously that a resource management to find and keep track of information about Grid resources is necessary. In this paper, we focus on the resource monitoring and provide a novel information retrieving protocol to support the precise data about resource status. To offer the latest information and balance the maintenance loading in Grids, a highly efficient Grid Resource Information Monitoring (GRIM) prototype is introduced. To efficiently collect the information with the less maintenance burden, another Grid Resource Information Retrieving (GRIR) protocol is proposed.

The remainder of this paper is organized as follows. Section 2 introduces the basic concepts, and some related investigations are discussed in Section 3. Section 4 presents the proposed data delivery protocol. The experimental analysis and simulation results are explained in Section 5. Section 6 concludes the paper and indicates future directions.

2: BACKGROUND

Grid Resource Information Monitoring (GRIM) prototype is similar to the monitoring system introduced in [12]. Figure 1 illustrates a prototype of GRIM. There are three basic layers: query layer, mediation layer, and information provider layer. Guests in the query layer inquire interested information about some resources in the Grids. The mediation layer composes Mediators to mediate the queries from guests and the information updating from hosts. Each Mediator records the registration of interested resource information and caches the distinct monitoring conditions of every monitored host for any guest. Information provider layer consists of monitored hosts. Every resource provides Grid services regard as a host in the Grid systems. In the GRIM prototype, if the resource state is updated frequently, the information will be more accurate, but the communication costs will increase. If the frequency of updating is sparse, the transmission overhead is lower, but the information consulted in caching is relatively stale. To reduce the unnecessary cost and achieve the highly accurate information is a challenge. The communication between Mediator and resources can be classified into three models. They are the pull-based model, the push-based model, and the comprehensive pull- and push-based model. In this paper, a Grid Resource Information Retrieving (GRIR) protocol based on the push data delivery model is proposed. The purpose of GRIR is to reflect the status of Grid resource to the Mediator, and hold on resource information closed to its precise status. To overcome the shortcoming of push-based, the time interval for updating is exploited. To decrease the useless transmission, the threshold is used to determine whether updating current status or not. By the threshold and the time interval approaches, the useless updated overhead is expected to alleviate and the bandwidth consumption would be reduced.

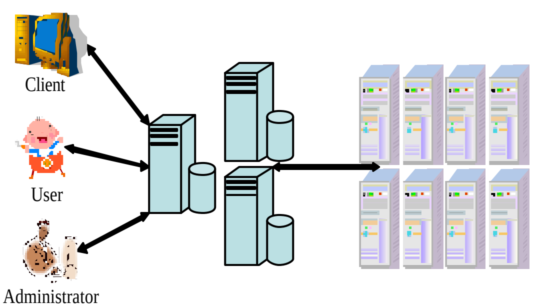

Figure 1. An illustration of GRIM prototype

3: RELATED WORKS

The Open Grid Services Architecture (OGSA) [3] is the principle for developing Grid systems. A new architecture for OGSA-based Grid information service [10] indicated the Monitoring and Discovery System with version 2 (MDS2) is not compatible with the OGSA-based Grid system. Therefore, FOrest Structured Information Service (FOSIS) are proposed.

The Globus Alliance [21] provides a series of open source software on Grid applications. The Globus toolkits are developed by the Globus Alliance based on the Open Grid Services Infrastructure (OGSI) [15]. Monitoring and Discovery System (MDS) [19] is well-known monitoring software of Globus toolkits. The information sources are registered with index servers in MDS [2]. The users can query directory servers to obtain detailed descriptions about resources. MDS consists of three basic services: they are Information Provider, Grid Resource Information Service (GRIS), and Grid Index Information Service (GIIS). Information provider senses the status on a resource. Grid resource information service gathers information from information providers. Grid index information service, like a directory service, provides the registration service for registering interested events.

Grid Monitoring Architecture (GMA) [14] developed by the Grid Global Forum (GGF) [18] is another popular technique for resource monitoring. GMA is comprised of three mainly components, Director Service, Producer Service, and Consumer Service. Director service locates producers to store available resource and consumers to provide information registration. Producer service is the event source to offer data available of Grid resources. Consumer service is the event sink to demand monitoring of certain resources. Producers register events of monitored information, and consumers register events of interested resource to the director service.

Another approach [6] integrates with various monitoring systems to provide complete solutions and functionalities. The grid monitoring architecture is based on Croatian Grid (CRO-GRID) infrastructure network [17]. To gather information and monitor performance on hosts, the Ganglia monitoring system [7] is exploited. To predict and enable remote sensor management, the Network Weather Service (NWS) [16] is used. To test the grid middleware services of integrated software, the INCA Test Harness and Reporting Framework [9] is utilized. It also exploits GRIS and GIIS services of MDS to integrate information hierarchically based on Globus GSI security with numerous monitoring systems.

Resource management services [1] provide users friendly control over the entire process. The users could attain more information about the Grid through an interactive set of services. The Steering service allows users to interact with their job submissions. The Job monitoring service provides real-time information and status about submitted job. The Estimator service is to provide estimates of the resource required by using history-based information. The scheduler will select a site that has the least estimated run time and minimum queue time for submitted task.

Among these various architectures of resource monitoring systems, they do not have a detailed data delivery protocol to achieve the up-to-date information. In [11], the authors proposed a three-tier architecture for monitoring, clients, ad-hoc network warehouses, and data sources. The data warehouse maintains client-specified views and keeps the views updated as the source data changes over time. The proposal brings up the coherence protocols based on the pull model to serve the up-to-date information from data source in addition, but some of the updates at a source will be missed when under-polling. In order to minimize the number of missed updates, consistent strategies are developed according to the polling rate, such as Regular Polling strategy, Adaptive Polling strategy, and Slacker Polling strategy [12]. In regular polling strategy, data warehouses query data sources with regular intervals. The adaptive polling strategy is a dynamic polling approach according to the significantly different of polling results. The next time interval is dynamically adjusted by a constant damping factor d. However, they do not indicate how to define the significant difference in the value that is currently maintained by the data warehouse. In the slacker polling strategy, the data warehouse estimates the likely time of the next update according to the time series information returned from the data source. The moving average estimator estimates the next poll with a window size N. Another estimator named exponential weighted moving average (EWMA) gives the weighted parameter a to time series. The dynamic EWMA will choose the adaptive value of a that results in the less error. To estimates the best time to the next poll, the moving average estimator exploits window size to cache time series and the EWMA uses weights on time series. However, the immediate status will not update at the risk of the prediction miss in the pull-based model. To instantaneously reflect the chopped and changed status of Grid resources, the push-based data delivery model is preferred. We develop data delivery protocol in the push-based model to achieve the higher data accuracy with the lower update overhead and bandwidth consumption.

4: GRID RESOURCE INFORMATION RETRIEVING PROTOCOL

In this section, we introduce the consistency protocols for the resource information retrieving named Grid Resource Information Retrieving (GRIR) protocol in Grid system. In order to reflect the dynamically changing status of resources in time, the proposed GRIR is a push-based consistent protocol. To renew information, Offset-sensitive mechanism (OSM), Time-sensitive mechanism (TSM), and Hybrid mechanism are used.

In the OSM, the updates are triggered according to the variation of resource states compared with last announced value to the Mediator. The diligent Announce-On-Changing (AOC) scheme and the sluggish Announce-Marginal-Changed (AMC) scheme are applied to accomplish the precise information coherency. The AOC performs the most up-to-date status about resources, and the AMC is based on the differences between the current monitored status and the last updated status. In AMC, a de/incremental threshold (DIth) is applied to trigger the update notification transmitted from the host to the Mediator. The static threshold (sDIth) and dynamic threshold (dDIth) are discussed in Dith separately. The sDIth is assigned with a constant threshold value, and the dDIth is a dynamic threshold value. The initial value of dDIth is set to 0 and dynamic calculated as $dDIth = \frac{1}{NC} \times \sum_{i=1}^{NC} MC_i$, where NC is the number of changes, and MC_i is the ith fluctuation of marginal changed between last announced value and current monitored value. If MC_i is larger or equal to the last average dDIth, the current status of monitored resource will be announced.

In the TSM, a host announces the changed status to the Mediator depends on the time interval between two updating messages. To achieve accurately information collection between the hosts and the Mediator, we propose two data delivery schemes based on the time interval: Announce-Regular-Interval (ARI) and Announce-Dynamic-Interval (ADI). A host announces the current status to the Mediator on a regular time schedule and the data maintained in the Mediator will be periodically updated if the information coherency adopts the ARI scheme. The Regular Time Interval (RTI) is assigned with a constant value in initial. If the RTI is expired, the current monitored resource status will be announced and reset the RTI to the initial value.

The dynamical adjustment of the time interval for a host sending the announced message to the Mediator is based on average time interval between two information changes. To calculate the average Dynamic Time Interval (DTI) from start to present according to the time interval if the status is changed as $DTI = \left\lfloor \frac{1}{NC} \times \sum_{i=1}^{NC} CI_i \right\rfloor$, where CI_i represents the ith different value of time interval between two successive changes as data changed. It is calculated from last changed timestamp to current changed timestamp. To set the initial value of DTI is calculated according to the status is changed in the first time. Then DTI will be dynamically recomputed as data changed. The different from the moving average estimator proposed in [12] is that we do not limit the window size for taking down the changed timestamps.

In the Hybrid mechanism, a comprehensive protocol Announce Marginal Changed with Dynamic Interval (AMCDI) is proposed to mix the notion of OSM and TSM and improve the useful announcing information. The announced message is transmitted relaying on dynamically revising time interval according to the marginal changed value of monitored resource status. To dynamically adjust the time interval for the next announcement., the Adaptive Fluctuant Factor (AFF) is defined and calculated as

$$AFF = \begin{cases} \frac{MC_i}{last\_dDIth}, & \text{if } MC_i < last\_dDIth \\ \frac{last\_dDIth}{MC_i}, & \text{otherwise} \end{cases}.$$

Relaying on the AFF, the next time interval for announcing information is computed as

$$T_{next} = \begin{cases} \left\lfloor T_{last} \times (1 + (1 - AFF)) \right\rfloor, & \text{if } MC_i < last\_dDIth \\ \left\lfloor T_{last} \times (1 - (1 - AFF)) \right\rfloor, & \text{otherwise} \end{cases}.$$

The proposed GRIR achieves the following objectives. First, the useless announced message transferred from hosts to the Mediator will be minimized. Second, the information coherency between the cached temporal data in the Mediator and the realistic state of the resources will be maximized.

5: EXPERIMENTAL ANALYSIS

We simulate the data delivery protocols on a Grid environment at the National Dong-Hwa University in order to estimate the effectiveness on performance metrics. For the experiments, we construct a basic experimental scenario with 6 nodes to analyze the results. One Mediator is used to monitor the hosts and interconnected with other hosts on a 100 Mbps LAN. For the experimental testbed, all hosts are running on the Linux kernel 2.4. One host equipped with an AMD Athlon 1009 MHz with 192 MB of memory, one host equipped with a Celeron 434 MHz with 256 MB of memory, and the others are equipped with an Intel Pentium IV and 256 MB of memory. A sensor is implemented with the data delivery protocols written in JAVA and in charge of continuously monitoring the memory utilization on each host. For evaluating the performance metrics of various data delivery protocols in the same circumstance, the data patterns about available memory were gathered by sensors from each host with a day period for each second.

In experiments, we evaluate various data delivery protocols to trace the data patterns and report the experimental results on performance metrics. The update ratio (number of updates/duration of monitoring), miss update ratio (number of miss updates/duration of monitoring), and redundant update ratio (number of redundant updates/number of updates) are the primary concerned metrics in our study. First of all, the mean change interval (duration of monitoring/number of changes) for each host to exhibit the rate for a changed value of available memory is estimated. Figure 2 shows the mean change interval of available memory on each monitored host. The H1 presents the most frequently change occurred so that the average time interval for the value changed is the shortest with the mean change interval is measured as 1.6. The mean change interval of H2 and H4 are measured similar to the value of 6. The mean change interval of H3 and H5 are measured with the similar value about 11 which presents the less frequently change occurred.

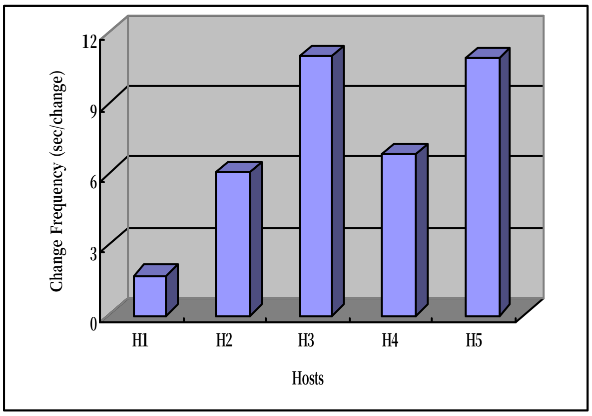

Figure 2. Mean change interval for each host

In the next, we examine the performance compared between our proposed push-based GRIR protocols and the pull-based protocol proposed in [12]. We compare the performance among our proposed time-sensitive mechanism with the pull-based moving average estimator and the dynamic exponential weighted moving average estimator (dynamic EWMA). The window size of the moving average estimator is set to 10. The parameter a of dynamic EWMA is dynamically chosen the value from 0.1 to 0.9 that results in the better performance.

Figure 3 and Figure 4 show the experimental results of proposed AOC, AMCd, and AMCDI protocols. The results clearly show that the more frequently update the data will result in the lower miss update ratio. In the simulation, the AOC provides the best effectiveness of miss update ratio, but costs the highest bandwidth consumption. In Figure 4, the AMCd and AMCDI perform the inefficient performance than AOC on miss update ratio. The reason is the changed value updated only when the differential value is larger or equal to the threshold value. In another aspect, the AMCDI performs better than AMCd because AMCDI updates the changed data when the time interval is expired even if the changed value is not exceeding the dDIth value. Comparing Figure 3 with Figure 4, AMCDI has the lower update ratio than the AOC with a little bit miss update ratio. It means AMCDI can alleviate the bandwidth consumption and achieve the high degree of data accuracy.

In Figure 4, adaptive polling strategy performs the inefficient performance, because the suddenly changes with large value that results in increasing the higher threshold value. Therefore, the great majority of slightly changed values will successively increase the next time interval to update. Contrasting with the static damping factor of adaptive polling strategy, AMCDI could dynamically adjust the AFF according to the presently differential fluctuated value and the last updated value. The next time interval could be shortened once the changed value exceeds the dDIth threshold. Besides, AMCDI could update the changed value with large margin and recompute the time interval for the next update even if time interval is not expired.

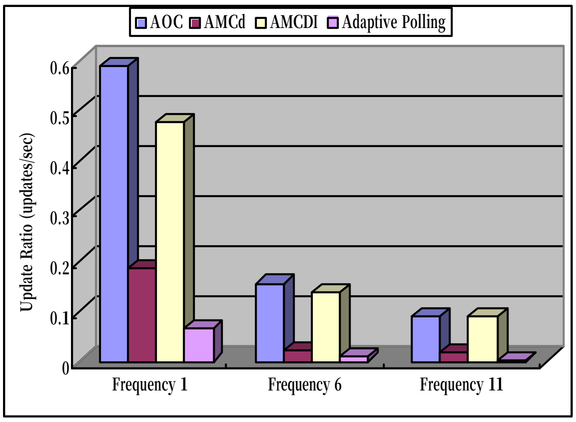

Figure 3. The effect of various offset-sensitive protocols on update ratio

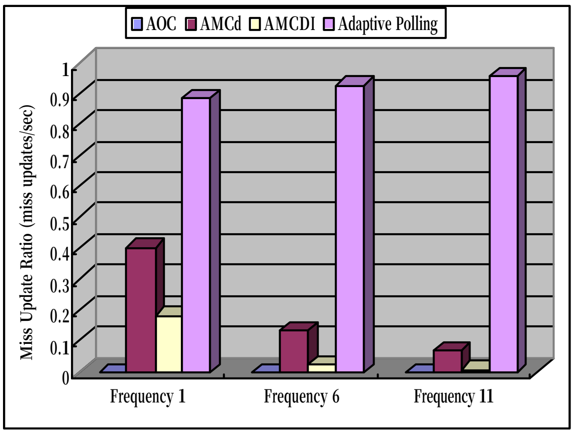

Figure 4. The effect of various offset-sensitive protocols on miss update ratio

Various performance metrics are measured on proposed AMCDI, ARI, ADI protocols with the pull-based moving average estimator, and the dynamic EWMA estimator. The experimental results are shown in Figure 5, Figure 6, and Figure 7. The periodically time interval to update of ARI is set according to the average intermission time measured by two successively updates occurred for each class. As expected that the static time interval of ARI is the less effectiveness than the dynamical time interval for the next update on miss update ratio. The moving average estimator has the more frequently updates to achieve the lower miss update ratio than the ADI and dynamic EWMA estimator. The uselessly redundant transmission of moving average estimator is higher than the others. Because the same monitored value with the last updated value will not cause update even if the time interval is expired, the AMCDI presents the best performance on redundant update ratio. For the miss update ratio, the AMCDI exhibits the best performance than the others. The reason is AMCDI updates the changed value when the dynamically adjusted time interval is expired or the monitored value has a larger margin than the threshold value during the intermission time. Besides, AMCDI will not transmit the redundant monitored value if no change is occurred.

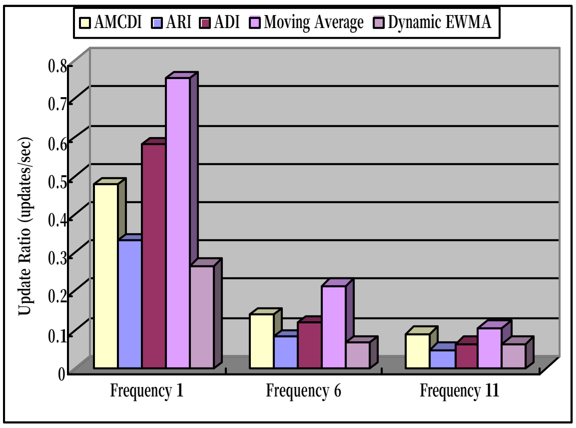

Figure 5. The effect of various time-sensitive protocols on update ratio

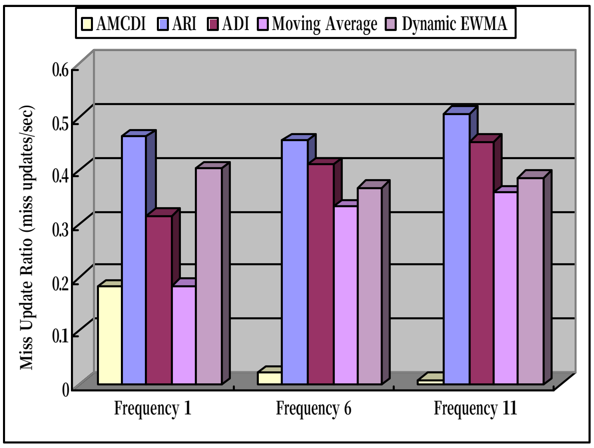

Figure 6. The effect of various time-sensitive protocols on miss update ratio

In general, the more the higher update ratio is calculated, the lower the miss update ratio will achieve. The AOC scheme, as expected, performs the less miss update ratio to accomplish the most freshness of data, but it results in the higher network bandwidth. In the offset-sensitive mechanism (OSM), the AMC scheme reduces the more bandwidth consumption, but it increases the higher out-of-date information about hosts (in Figure 4). Besides, the AMC scheme may occur the predicament if the changed data is rarely exceeded the DIth. In the AMCDI scheme, the changed data would be announced as the time interval expired even if the changed value is lower than the desired threshold value. Therefore, the AMCDI scheme could solve the predicament of OSM operation.

In the time-sensitive mechanism (TSM), the dynamic time interval of ADI scheme has higher accurate information than the regular time interval of ARI scheme, but the AMCDI scheme performs better than the ADI scheme, as shown in Figure 6. If the marginal changed value at present is larger or equal to the desired threshold value, the AMCDI scheme will announce the current monitored status even if the time interval is not expired. Besides, the AMCDI scheme will not announce the same monitored value even if the time interval is expired. Though, the AMCDI scheme could accomplish the higher degree of data accuracy (Figure 6) and reduce the useless redundant updates to alleviate the update overhead (Figure 7).

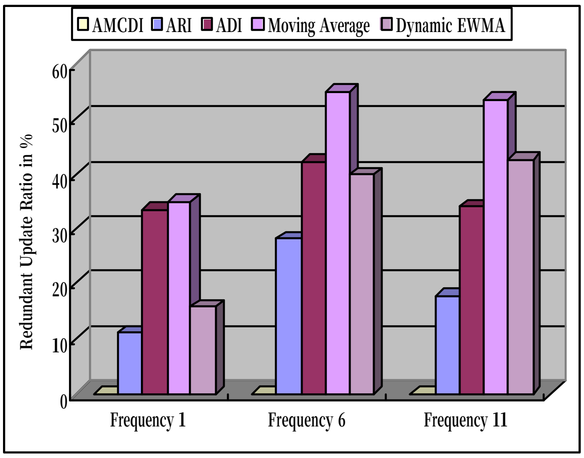

Figure 7. The effect of various time-sensitive protocols on redundant update ratio

To sum up the experimental results, the AMCDI is the preferred scheme in GRIR. Compared with the prior work, the AMCDI scheme performs the better performance than the others because it can more adaptive adjust the time interval according to the dynamically calculated AFF. In summary, the AMCDI scheme not only performs the better effectiveness on bandwidth consumption but also achieves a little bit loss of data fidelity than the AOC scheme.

6: CONCLUSION AND FUTURE WORKS

In this paper, we mainly focus on the data delivery protocol in the push-based model to support the data accuracy. The management of the mediation layer is another vital issue. The management of multiple Mediators could be centralized coordination, but it might arise the single point of failure. The distributed coordination framework is useful to balance the workload of monitoring, but the resource discovery latency is another problem [5, 13]. In the future, we would like to combine the centralized and distributed coordination framework with the peer-to-peer technology in the Grids such as the Super-Peer model [8]. On the other hand, the scheduling for the requests from the query layer, the communication of resource discovery between multiple mediators in the mediation layer, and the load balance of the management in the mediation layer will remain as a matter to be discussed further.

REFERENCES

[1] Arshad Ali, Ashiq Anjum, Tahir Azim, Julian Bunn, Atif Mehmood, Richard McClatchey, Harvey Newman, Waqas ur Rehman, Conrad Steenberg, Michael Thomas, Frank van Lingen, Ian Willers, and Muhammad Adeel Zafar, "Resource Management Services for a Grid Analysis Environment," Proceedings of International Conference on Parallel Processing Workshops, pp. 53-60, June 2005.

[2] Karl Czajkowskiy, Steven Fitzgeraldz, Ian Fosterx, and Carl Kesselman, "Grid Information Services for Distributed Resource Sharing," Proceedings of the 10th IEEE International Symposium on High-Performance Distributed Computing, pp. 181-194, August 2001.

[3] Ian Foster, Carl Kesselman, Jeffrey M. Nick, and Steven Tuecke, "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration," Technical Report, Open Grid Service Infrastructure Working Group, Global Grid Forum, June 2002. Available: http://www.globus.org/alliance/publications/papers/ogsa.pdf

[4] Ian Foster, Carl Kesselman, and Steven Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," International Journal of High Performance Computing Applications, Vol. 15, No. 3, pp. 200-222, Fall 2001.

[5] Adriana Iamnitchi and Ian Foster, "On Fully Decentralized Resource Discovery in Grid Environments," Proceedings of the Second International Workshop on Grid Computing, Vol. 2242, pp. 51-62, 2001.

[6] Emir Imamagic, Branimir Radic, and Dobrisa Dobrenic, "CRO-GRID Grid Monitoring Architecture," 27th International Conference on Information Technology Interfaces, pp. 65-72, June 2005.

[7] Matthew L. Massie, Brent N. Chun, and David E. Culler, "The Ganglia Distributed Monitoring System: Design, Implementation, and Experience," Parallel Computing, Vol. 30, No. 7, pp. 817 840, July 2004.

[8] Carlo Mastroianni, Domenico Talia, and Oreste Verta, "A Super-Peer Model for Resource Discovery Services in Large-Scale Grids," Future Generation Computer Systems, Vol. 21, No. 8, pp. 1235-1248, October 2005.

[9] Shava Smallen, Catherine Olschanowsky, Kate Ericson, Pete Beckman, and Jennifer Schopf, "The Inca Test Harness and Reporting Framework," Proceedings of ACM/IEEE Conference on Supercomputing, pp. 55-64, November 2004.

[10] Hailong Sun, Ying Zhang, Chunming Hu, and Jinpeng Huai, "A new architecture for OGSA-based grid information service," Proceedings of IEEE International Conference on Services Computing, pp. 575-578, September 2004.

[11] Radhakris hnan Sundaresanz, Tahsin Kurcy, Mario Lauriaz, Srinivasan Parthasarathyz, and Joel Saltz, "A Slacker Coherence Protocol for Pull-based Monitoring of On-line Data Sources," Proceedings of the 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid, pp. 250-254, May 2003.

[12] Radhakrishnan Sundaresanz, Tahsin Kurcy, Mario Lauriaz, Srinivasan Parthasarathyz, and Joel Saltz, "Adaptive Polling of Grid Resource Monitors Using a Slacker Coherence Model," Proceedings of the 12th IEEE International Symposium on High Performance Distributed Computing, pp. 260-269, June 2003.

[13] Sanya Tangpongprasit, Takahiro Katagiri, Kenji Kise, Hiroki Honda, and Toshitsugu Yuba, "A Time-To-Live Based Reservation Algorithm on Fully Decentralized Resource Discovery in Grid Computing," Parallel Computing, Vol. 31, pp. 529–543, June 2005.

[14] Brian Tierney, Ruth Aydt, Dan Gunter, Warren Smith, Martin Swany, Valerie Taylor, and Rich Wolski, "A Grid Monitoring Architecture," The Global Grid Forum Draft Recommendation (GWD-Perf-16-3), August 2002.

[15] Steven Tuecke, Karl Czajkowski, Ian Foster, Jeffrey Frey, Steve Graham, Carl Kesselman, Tom Maquire, Thomas Sandholm, David Snelling, and Peter Vanderbilt, "Open Grid Services Infrastructure (OGSI) Version 1.0," The Global Grid Forum Draft Recommendation (draft-ggf-ogsi-gridservice-33), June 2003.

[16] Rich Wolski, Neil Spring, and Jim Hayes, "The Network Weather Service: A Distributed Resource Performance Forecasting Service for Metacomputing," Journal of Future Generation Computing Systems, Vol. 15, No. 5-6, pp. 757-768, October 1999.

[17] CRO-GRID. Avaliable: http://www.cro-grid.hr/

[18] Global Grid Forum. Available: http://www.gridforum.org/

[19] Monitoring and Discovery System. Available: http://www.globus.org/toolkit/mds/

[20] The Globus Alliance. Available: http://www.globus.org/