A P2P Approach to Resource Discovery in On-Line Monitoring of Grid Workflows

Bartłomiej Łabno[1], Marian Bubak[1,2], and Bartosz Baliś[2]

[1] Academic Computer Centre - CYFRONET, Poland
[2] Institute of Computer Science, AGH Poland

Abstract. On-line monitoring of Grid workflows is challenging since workflows are loosely coupled and highly dynamic. An efficient mechanism of automatic resource discovery is needed in order to discover new producers of workflow monitoring data fast. However, currently used Grid information systems are not suitable for this due to insufficient performance characteristics. We propose to associate the monitoring infrastructure with a P2P DHT infrastructure in order to achieve the automatic resource discovery wherein consumers of monitoring data can be notified fast about new producers. In our solution, the consumer of monitoring data can subscribe to any monitoring endpoint and the automatic resource discovery is handled transparently. We evaluate performance of the presented solution, and demonstrated a case study scenario of monitoring of a traffic management workflow.

Keywords: grid computing, monitoring, workflows, resource discovery, peer to peer, distributed hash tables.

1 Introduction

Monitoring of scientific workflows is important for many purposes including status tracking, recording provenance, or performance improvement. Resource discovery is an indispensable phase in Grid monitoring scenarios. A directory service is usually a part of the monitoring infrastructure to serve as an information service [13]. Monitoring producers advertise themselves in the directory service providing information about the monitored resources and endpoint where monitoring data can be acquired. Consumers need to discover the monitoring service endpoint using the name or attributes of the resource to be monitored.

In some cases, on-line monitoring is desirable in order to quickly respond to problems or react to environment changes, as in dynamic rescheduling, for example, wherein computing resources are dynamically re-allocated in response to performance degradations. However, the distributed and dynamic nature of workflows combined with the fully decentralized architecture of the monitoring infrastructure, required in the Grid, makes this task difficult to achieve. As workflow's activities dynamically emerge at unpredictable times and locations, a mechanism of *fast and automatic resource discovery* wherein new producers of workflow monitoring data are automatically discovered and transparently receive

E. Luque, T. Margalef, and D. Benítez (Eds.): Euro-Par 2008, LNCS 5168, pp. 37–46, 2008.

subscription requests on behalf of active subscribers, seems to be the key issue to enable on-line monitoring of Grid workflows.

Unfortunately, existing information services used in Grid production infrastructures are not suitable for frequently changing resources. Two factors are responsible for this. First, the high 'put latency' of directory services; second, the lack of efficient notification-based discovery mechanism. The 'put latency' denotes the delay from the registration of a new resource (or information update thereof) in the resource discovery system to the time it can actually be discovered (the information propagation delay). The 'get latency', on the other hand, is the delay needed to retrieve information from a directory service (the response time of a discovery request). While in most approaches the 'get latency' is of greatest concern (to ensure low response times regardless of a growing size of resource database), the low 'put latency' is essential for a fast discovery of changing resources required for on-line monitoring.

Peer-to-peer technologies are increasingly important for distributed systems, such as distributed storage systems [5] or in Grid technologies improving rich media content delivery [8]. The two technologies – the Grid and peer to peer are viewed as complimentary and likely to converge [9]. *Distributed Hash Table* is a special form of a peer-to-peer network and it acts as a hash table which is distributed among all the nodes in the network [14]. Each node keeps a piece of information which is usually a range of keys and associated values. An interface is provided for registering key – value pairs and for retrieving thereof. A DHT network is often structured, i.e. though it is not known in which node of the network a specific key is stored, one is guaranteed to reach this node by routing in no more than $log(n)$ hops [14] (n being the total number of nodes in the P2P network). The same complexity is guaranteed for putting a new key – value pair into the network. As a result, DHT networks can provide excellent performance, high scalability and availability. For example, for the Amazon's Dynamo, a highly available and scalable key-value store, the reported latencies are around $15ms$ for reads and $30ms$ for writes, while the $99.9th$ percentile latencies are around $200ms$ for reads and $300ms$ for writes [7].

The goal of this work is to investigate the peer to peer distributed hash table technologies for supporting the automatic resource discovery in on-line monitoring of Grid workflows. In our approach, essentially, one can subscribe to, e.g., 'all workflow $Wf1$ events', in *any* monitoring service endpoint. The actual automatic resource discovery is performed by means of the P2P DHT infrastructure associated with the monitoring infrastructure. Last but not least, it must be noted that the automatic resource discovery for Grid workflows requires a simple name-based lookup, as opposed to complex, *ad-hoc* content-based lookup featured by full-blown discovery services and required for more complex resource discovery scenarios based on attribute values (e.g. resource matching). Consequently, the DHT infrastructure introduced in the presented solution is not meant to replace the global discovery service but rather support it in some special scenarios where performance is critical.

The remainder of this paper is organized as follows. Section 2 overviews the related works. Section 3 presents the proposed solution for P2P-DHT-based automatic resource discovery. In Section 4, performance evaluation of the prototype monitoring infrastructure with P2P-DHT deployment is presented. Section 5 presents a case-study monitoring of a Coordinated Traffic Management workflow. Finally, Section 6 summarizes the presented work and overviews possible paths for future investigation.

2 Related Work

The goal of this section is to overview existing resource discovery services focusing on those found in large-scale production Grid deployments, and the analysis of their suitability for on-line monitoring of Grid workflows.

Globus *Monitoring and Discovery System* (MDS) [11,12] started as a centralized LDAP server (MDS-1), but the limitations of this solution led to a distributed LDAP architecture in MDS-2. Its successors, MDS-3 and MDS-4, are conceptually similar to MDS-2, but they follow the *Service Oriented Architecture* patterns. MDS-4 allows to acquire the monitoring data from the *Index Services*. Every Index Service aggregates Service Data Elements which describe resources registered to them. A hierarchy can be formed and an upper-level Index Service can aggregate information from the lower-level ones.

Berkeley Database Information Index (BDII) is an information system deployed at over 250 sites in the EGEE project. BDII was made as a replacement for Globus MDS2. It has a hierarchical architecture in which low-level Grid Resource Information Services (GRIS) provide information to site BDII which is in turn combined and exposed by top-level BDIIs. The information is refreshed in a top-down manner, i.e. a top-level BDII scans site BDIIs which obtain information from GRISes. The refresh is done by reloading the entire database and rebuilding indices every 2-3 minutes [2].

iGrid is a grid information service developed within the European GridLab [1]. The project started with Globus MDS and LDAP as a core, however, it has moved to a relational database storage, due to the disadvantages LDAP which is designed for frequent reads but not updates. Also, the deficiencies of LDAP-based queries and overall poor performance caused this. Still, a migration to a DHT in iGrid is planned to improve performance.

R-GMA is a relational implementation of the Grid Monitoring Architecture ([13]), developed within the European DataGrid. It is based on the relational model and it supports SQL as the query interface, though it does not feature a distributed RDBMS [6]. R-GMA has been reported to have performance problems and a relatively low throughput [15].

In general, the described systems have certain common characteristics. First, they are distributed, usually featuring a hierarchical architecture. Second, they are oriented towards high query performance, not update performance. Third, high scalability is achieved usually by using caching of information. Systems not using caching have been shown to display very low throughput of around

1-5 query requests per second [15]. A performance study of a web-service-based information system of the latest generation, MDS4 (Monitoring and Discovery System), has shown that it can sustain a throughput of 10 requests per second (dual 2.4 Xeon with 4GB RAM), regardless of the number of concurrent clients, but for the index size as small as 500 entries [12]. While the larger index sizes were not investigated we can note how throughput degrades with the index size by observing that for the index of size 500 it is twice as big as for the index of size 100 [12].

In summary, the analysis of existing Grid information systems operating on large-scale deployments of production Grids (such as BDII in the EGEE testbed) shows that the realistic update rate is of order of minutes. This is insufficient for the on-line monitoring scenario.

3 The Automatic Resource Discovery Scenario

The common Grid monitoring scenario consists of the following phases: producer advertisement, resource discovery, subscription to monitoring data, and actual transfer of monitoring data. However, it is not sufficient for on-line monitoring of Grid workflows. For example, for a simple monitoring request 'subscribe to all monitoring events for workflow Wf1', the initial resource discovery will reveal only those data sources which monitor workflow activities existing at the time. However, the occurrence of further workflow activities will not be discovered. A notification mechanism is required to notify data consumers about new data sources upon their appearance. This scenario may be called *automatic resource discovery*.

Our goal is to realize the scenario wherein a consumer could subscribe to a workflow monitoring data only once to an arbitrary monitoring service endpoint, and the underlying resource discovery and data transfer will be handled automatically by the monitoring infrastructure itself. The proposed solution architecture is presented in Fig. 1. We assume a fully distributed monitoring system architecture consisting of *Sensors* – producers of monitoring data local to monitored resources, and *Monitors* managing Sensors and exposing interfaces to consumers of monitoring data. In addition, an external P2P Distributed Hash Table infrastructure is associated with the Monitor network to provide global information about Sensors.

The proposed scenario for automatic resource discovery in monitoring of Grid workflows is presented in Fig. 2. Workflow $wf1$ is monitored and its activities appear in various times and locations. The first workflow activity is monitored by Sensor $s1$. The sensor registers to its respective Monitor $m1$. This monitor performs a lookup in the DHT network in order to check if the workflow has already been registered in the monitoring system. If it is not the case (as in the example), $m1$ puts the first entry in the DHT, where the key is the unique workflow ID, and the value is the Monitor $m1$ endpoint. A consumer (tool), in order to request monitoring data for workflow $w1$, needs first to discover *any* Monitor endpoint (from a traditional information service). In this case it

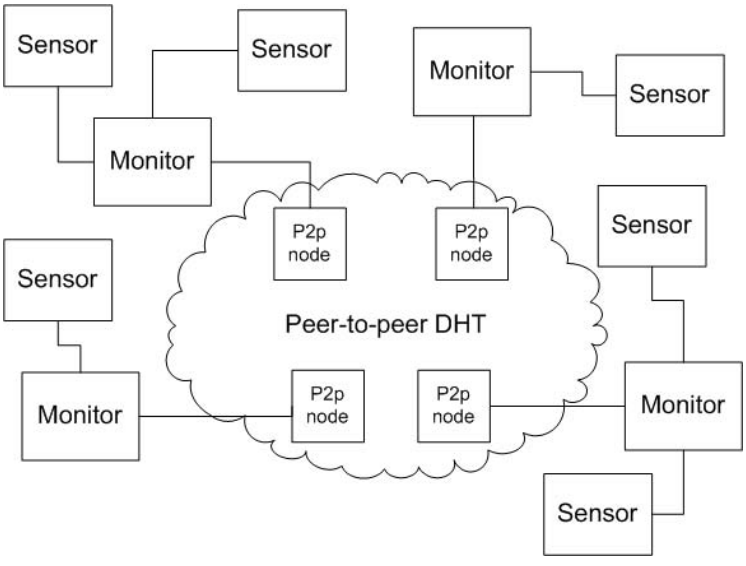

Fig. 1. Architecture of the monitoring system

obtains Monitor's $m0$ endpoint and subscribes in it. From now on, Monitor $m0$ manages the tool's subscription. It looks up the DHT for all producers for workflow $w1$ and forwards them subscription requests on behalf of the tool, and the monitoring data are pushed directly from from producers to the tool (consumer). When a new producer for workflow $w1$ appears at a later time ($m2$), $m0$ automatically detects it and sends it the subscription request. The discovery of new producers could be achieved either by using an external event notification service, or periodically polling the DHT for new entries. The latter solution is shown. Though it is less efficient that the notification-based one, the DHT still offers a sufficient latency to make this solution feasible.

4 Performance Evaluation

In order to study the feasibility of the scenario presented in the previous section, we have made the performance evaluation of our solution. To this end, we have deployed our GEMINI monitoring framework [3] and the Bamboo DHT network[1]. We have used a Core Duo 2 GHz Pentium processor machine with 2 GB RAM running Open SuSe 10.1. In addition, we have used the OpenDHT project infrastructure[2] which is a deployment of Bamboo DHT on the PlanetLab infrastructure[3] distributed all over the world.

[1] Bamboo project homepage: `http://bamboo-dht.org/`
[2] OpenDHT project homepage: `http://opendht.org/`
[3] PlanetLab project homepage: `http://www.planet-lab.org/`

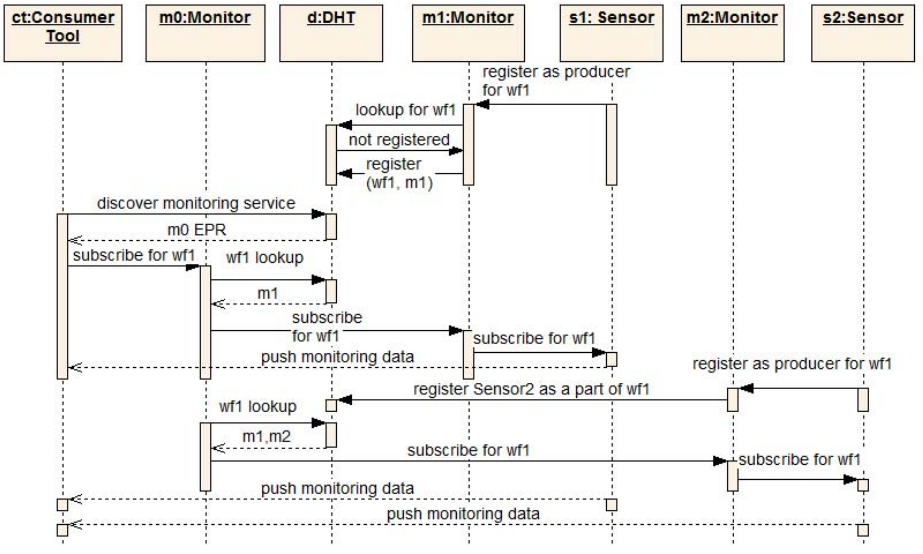

Fig. 2. Scenario of automatic resource discovery and subscription for workflow monitoring data

Test results are presented in Tab. 1 as average operation latencies. The tests performed on one node show that the fastest operation is get; remove is about five times slower while the most time consuming one is put which is about seventeen times longer than the remove operation. While those times are considerably larger on a 247-nodes deployment, they are still satisfactory, around one second each.

Table 1. Average time of operations on the Bamboo DHT network with one node and OpenDHT network with 247 nodes

Network type	Operation	Average time, s
Bamboo DHT with one node	put	0.166
	get	0.002
	remove	0.010
OpenDHT - Bamboo on 247 nodes	put	1.106
	get	0.902
	remove	0.998

The obtained performance characteristics of the tested systems allowed us to perform a model-based performance analysis of the presented solution which revealed that it can accommodate very high workloads. Up to 10 job arrivals per second have been tested which is much more than in the current large-scale production Grids. The **automatic discovery delay** (the time from the arrival

Table 2. Average latency of get operation for different DHT deployments

Operation	System type	Average time, s
get	Bamboo DHT with one node	0.002
	OpenDHT – Bamboo on 247 nodes	0.902

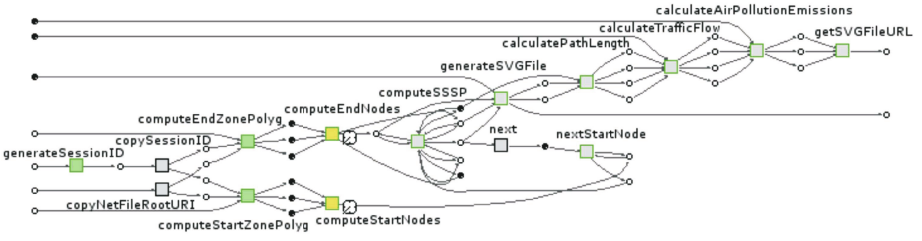

Fig. 3. Coordinated Traffic Management workflow

of a new workflow job until the beginning of the monitoring data transfer) was stable at around 1500ms, which is perfectly sufficient for on-line monitoring.

5 Case Study: Monitoring of Coordinated Traffic Management Workflow

To demonstrate workflow monitoring, we have chosen the Coordinated Traffic Management (CTM) workflow constructed from application services provided by Softeco Sismat within the K-Wf Grid Project [4]. This application targets the computation of the emission of traffic air pollutants in an urban area and has been developed in tight collaboration with the Urban Mobility Department of the Municipality of Genoa, which provided a monolithic implementation of the model for the pollutant emission calculations, the urban topology network and real urban traffic data. The CTM application workflow has been divided into several different steps in order to allow the semi-automatic composition of services and the definition of a set of ontologies which describe the CTM domain and feed the system with the information needed for the proper selection and execution of services [10].

The main CTM application functionalities are best route, traffic flow and air pollutant emissions calculations. Data graphical representation in SVG format is also supported. For monitoring, a complex use case was used. It consisted of several executable transitions and three control transitions (Fig. 3).

The first activity in the workflow is the generation of a session ID. Next, there are two activities done in parallel – the computation of start and end zone district polygons. Subsequently, node coordinates for start and end are computed, also in parallel. After that, a set of calculations is done for nodes computed in earlier activities. Finally, computations responsible for calculating path length, traffic flow and air pollutants emission follow. Results are also written to the SVG format file, which is done in one of the activities.

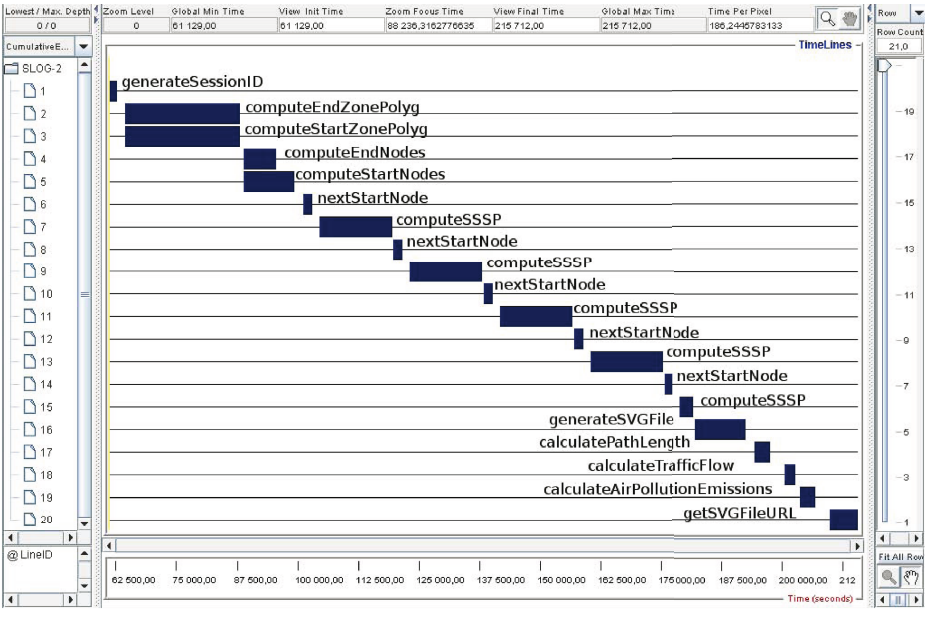

Fig. 4. Monitoring results – global view

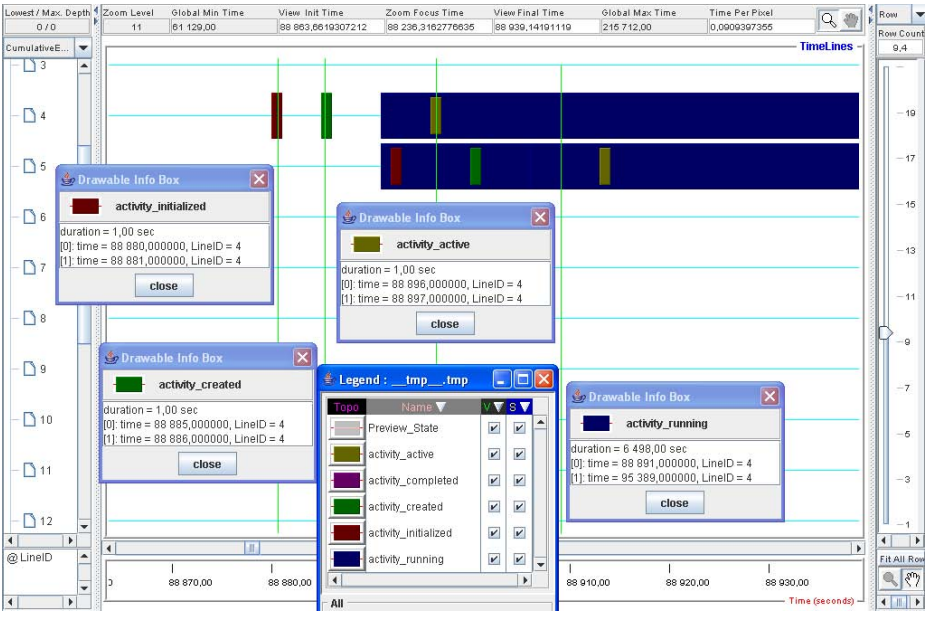

Fig. 5. Monitoring results – detailed local view

During the running of the above scenario monitoring data were acquired by instrumentation of the worklow enactment engine and workflow activities. For each activity, several events were produced: when it was initialized, created, went to the active state, at start and end of the actual running phase, and when it has completed. Each of these events has a precise time of occurrence. Activities such as initialization, creation, activation and completion should be treated as a point in time. The running state is treated as a period of time.

For visualization of the monitoring results, we have used the Jumpshot tool[4]. To this end, events collected from GEMINI were translated to Jumpshot's SLOG-2 format. We can observe how the workflow was executed, how much time each activity has taken, and when it was invoked. Fig. 4 presents a global view showing all activities. However, due to the time scale, only the running periods can be seen. Fig. 5 presents a zoom of a particular diagram section to show more details – individual events can be seen now. Additionally, windows describing individual bars (representing events and periods) are shown.

6 Conclusion

We have presented how P2P DHT systems can be used to achieve on-line monitoring of Grid workflows through automatic resource discovery. The DHT network was used not as a replacement of a global discovery service, but rather as a supporting infrastructure which can improve performance in certain scenarios. Thus, a traditional directory service can be used in our monitoring architecture only to discover any monitoring endpoint, while the subsequent discovery of workflow activities is handled by the monitoring infrastructure associated with the DHT network. In the future, we plan to integrate our solution for monitoring of resources in applications in the context of media and banking solutions in the Gredia project[5].

Acknowledgement. This work is partly funded by the European Commission under projects GREDIA IST-034363 and CoreGrid IST-2002-004265, and by the related Polish SPUB-M grant.

References

1. Aloisio, G., Cafaro, M., Epicoco, I., Fiore, S., Lezzi, D., Mirto, M., Mocavero, S.: Resource and Service Discovery in the iGrid Information Service. In: Gervasi, O., Gavrilova, M.L., Kumar, V., Laganá, A., Lee, H.P., Mun, Y., Taniar, D., Tan, C.J.K. (eds.) ICCSA 2005. LNCS, vol. 3482, pp. 1–9. Springer, Heidelberg (2005)
2. Astalos, J., Flis, L., Radecki, M., Ziajka, W.: Performance Improvements to BDII – Grid Information Service in EGEE. In: Proc. CGW 2007, Krakow, Poland. ACC CYFRONET AGH (2008)

[4] See `http://www-unix.mcs.anl.gov/perfvis/software/viewers/index.htm`

[5] Gredia project homepage: `http://www.gredia.eu`

3. Balis, B., Bubak, M., Łabno, B.: GEMINI: Generic Monitoring Infrastructure for Grid Resource and Applications. In: K-WfGrid – The Knowledge-based Workflow System for Grid Applications, Proc. CGW 2006, Krakow, vol. II, pp. 60–73 (2006)
4. Bubak, M., Fahringer, T., Hluchy, L., Hoheisel, A., Kitowski, J., Unger, S., Viano, G., Votis, K.: K-WfGrid Consortium: K-Wf Grid – Knowledge based Workflow system for Grid Applications. In: Proc. CGW 2004, Poland, p. 39. Academic Computer Centre CYFRONET AGH (2005) ISBN 83-915141-4-5
5. Cannataro, M., Talia, D., Tradigo, G., Trunfio, P., Veltri, P.: SIGMCC: a System for Sharing Meta Patient Records in a Peer-to-peer Environment. Future Generation Computer Systems 24(3), 222–234 (2008)
6. Cooke, A., et al.: The Relational Grid Monitoring Architecture: Mediating Information about the Grid. Journal of Grid Computing 2(4) (December 2004)
7. Decandia, G., et al.: Dynamo: Amazon's Highly Available Key-value Store. In: SOSP 2007: Proceedings of twenty-first ACM SIGOPS symposium, pp. 205–220. ACM Press, New York (2007)
8. Fortino, G., Russo, W.: Using P2P, GRID and Agent Technologies for the Development of Content Distribution Networks. Future Generation Computer Systems 24(3), 180–190 (2008)
9. Foster, I.T., Iamnitchi, A.: On Death, Taxes, and the Convergence of Peer-to-Peer and Grid Computing. In: Kaashoek, M.F., Stoica, I. (eds.) IPTPS 2003. LNCS, vol. 2735, pp. 118–128. Springer, Heidelberg (2003)
10. Gubala, T., Harezlak, D., Bubak, M., Malawski, M.: Semantic Composition of Scientific Workflows Based on the Petri Nets Formalism. In: Proc. 2nd IEEE International Conference on e-Science and Grid Computing (Avaialble only on CD-ROM). IEEE Computer Society Press, Los Alamitos (2006)
11. Schopf, J.M., Pearlman, L., Miller, N., Kesselman, C., Foster, I., D'Arcy, M., Chervenak, A.: Monitoring the grid with the Globus Toolkit MDS4. Journal of Physics: Conference Series 46, 521–525 (2006)
12. Schopf, J.M., Raicu, I., Pearlman, L., et al.: Monitoring and discovery in a web services framework: Functionality and performance of Globus Toolkit MDS4. Technical report, Mathematics and Computer Science Division, Argonne National Laboratory (2006)
13. Tierney, B., Aydt, R., Gunter, D., Smith, W., Taylor, V., Wolski, R., Swany, M.: A grid monitoring architecture. Technical Report GWD-PERF-16-2, Global Grid Forum (January 2002)
14. Trunfio, P., Talia, D., Papadakis, H., Fragopoulou, P., Mordacchini, M., Pennanen, M., Popov, K., Vlassov, V., Haridi, S.: Peer-to-Peer resource discovery in Grids: Models and systems. Future Generation Computer Systems 23(7), 864–878 (2007)
15. Zhang, X., Freschl, J.L., Schopf, J.M.: Scalability analysis of three monitoring and information systems: MDS2, R-GMA, and Hawkeye. J. Parallel Distrib. Comput. 67(8), 883–902 (2007)