An Agent-based approach to Grid Service Monitoring

Keith Rochford, Brian Coghlan, John Walsh
Department of Computer Science
Trinity College Dublin
Ireland
Email:{keith.rochford,coghlan,john.walsh}@cs.tcd.ie

Abstract— **The centralised management of distributed computing infrastructures presents a number of considerable challenges, not least of which is the effective monitoring of physical resources and middleware components to provide an accurate operational picture for use by administrative or management staff. The detection and presentation of real-time information pertaining to the performance and availability of computing resources is a difficult yet critical activity.**

This architecture is intended to enhance the service monitoring experience of a Grid operations team. We have designed and implemented an extensible agent-based architecture capable of detecting and aggregating status information using low-level sensors, functionality tests and existing information systems. To date it has been successfully deployed across eighteen Grid-Ireland sites.

I. Introduction

Effective monitoring of complex distributed infrastructures, such as electricity supply and traffic management systems, is essential if expected levels of service are to be achieved. Operations Centres are frequently established to satisfy this requirement, providing a central authoritative point of operational information. They are, however, only as good as the underlying information systems they depend on.

Distributed computing infrastructures such as Grids have similar monitoring requirements. For those responsible for the management and administration of the individual resources or the infrastructure as a whole it is desirable that some form of monitoring system is put in place to highlight problems and abnormalities. Centrally managed infrastructures, such as Grid-Ireland[1] or EGEE[2], present additional challenges. Large quantities of status information from disparate resources must be detected, aggregated and stored.

We describe the principles of and justifications for the monitoring system currently deployed on the Grid-Ireland infrastructure and provide a technical overview of the sensors, communication and presentation systems which make the aggregate real-time status information available to the members of the operations team. The monitoring information is presented in such a manner as to facilitate the management of the infrastructure in response to status and security events, and provide a tool to maximise the impact of public relations activities and demonstrations.

II. Grid Monitoring

Grid monitoring may be described as the activity of measuring significant grid related resource parameters in order to analyse usage, availability, behaviour and performance[3]. Its objectives include:

- Locating performance problems
- Tuning for better performance
- Fault detection and recovery
- Input to prediction services

The size and nature of grid systems means that monitoring systems must be capable of measuring a wide range of metrics relating to a substantial number of entities distributed across a large geographical area. Monitoring activities present those responsible for the maintenance of the infrastructures with considerable challenges and has become an active area of research in it's own right[4]. Traditional network and host monitoring tools often lack the scalability, dynamicity, or extensibility required for effective deployment within computational or data grids.

III. Centralised vs. Agent-based Monitoring

Remote entities are typically monitored using one of two approaches; a centralised approach where one or more processes executing at a single location are responsible for the monitoring of all entities or a federated approach involving multiple distributed processes and sensors. In a centralised system, all monitoring processes are executed from a single location and attempt to determine the status of monitored entities either through the establishment of some form of connection with the entity or by performing some operation involving it. Distributed monitoring architectures employ monitoring processes on or closer to the monitored entity. These processes, often referred to as agents, perform the required tests and report the outcomes, through push or pull mechanisms, to the interested parties. The area of software agents is one of active research and many types of agents have been identified or proposed. These range from simple processes to complex intelligent autonomous mobile elements of software. It is argued that the popularity of the term has led to its misuse and it might therefore be appropriate to present an explanation of what should be inferred by the term in the context of this work. We will use the term 'agent' to refer to

Proceedings of The Fifth International Symposium
on Parallel and Distributed Computing (ISPDC'06)
0-7695-2638-1/06 $20.00 © 2006 IEEE

an independent software component that performs operations on behalf of a user or other piece of software and exhibits some degree of autonomic behaviour.

Both methods of monitoring have advantages and the choice of which is best suited to a given situation is dependent on a number of factors including the size and nature of the systems, the required information and the degree of control that is required. It is often claimed that one approach is superior to the other, however the merits of each must be evaluated with respect to the individual systems and monitoring requirements. While it is true that the deployment of distributed monitoring processes can incur additional effort it is not necessarily true that the configuration need be any more complex. Unfortunately, centralised monitoring typically implies a more limited depth of data gathering than is possible with remote agents. In addition the potential for the management and control of remote entities through interaction with the distributed agents makes them an attractive choice for complex infrastructures. Also the scale of distribution can vary. The requirement for the installation of a software agent on each monitored host is dependent on the motivation of the monitoring activity and the amount of information pertaining to each host that is required. If the motivation for the adoption of an agent-based approach is merely to overcome the limitations imposed by network access policies, it may be sufficient to deploy a single agent to each network segment, domain, or site.

A combination of both approaches might be considered the best solution in terms of grid monitoring. Remote agents provide a level of access and control that would otherwise be unfeasible, yet there are a number of advantages offered by a centralised monitoring approach. It is of limited value to know the internal status information pertaining to a particular resource if we cannot determine whether or not the resource is accessible and usable through the normal operational mechanisms. Furthermore, in the interest of consistency and conceptual simplicity a central monitoring process might be implemented as a monitoring agent deployed at the Operations Centre.

IV. Motivations

Early grid service monitoring systems, such as MapCentre[5] and Ganglia[6], offered valuable but limited functionality, required complex configuration, and were employed with limited degrees of success on our particular infrastructure[1]. Monitoring traffic is often denied access into the remote networks hosting the monitored grid resources, greatly reducing the usefulness of these tools. The unsuitability of these systems prompted the investigation and assessment of alternatives.

Several existing host and network monitoring tools, including some supporting distributed monitoring, were assessed but did not offer a suitable solution. Many were intended for deployment within a single administrative domain. While changes to network settings, such as firewall rules, can be quickly authorised and actioned from within such environments, this is not true from outside the domain. From the viewpoint of an operations team the majority of the monitored resources reside within remote administrative domains which are beyond their control. In cases where the resources are remotely managed the remote network still remains beyond control; this makes efficient centralised monitoring of the resources impossible. This is the case for Grid-Ireland, which has a notably integrated remotely managed dedicated grid infrastructure, and in consequence a decision was taken to undertake research into this rather specialist monitoring field.

The monitoring solution described below employs a combination of distributed and centralised monitoring sensors along with the aggregation of information from existing monitoring tools and is specifically designed to speed the response to status and security events.

The immediate aims were to develop and deploy an extensible monitoring framework which would overcome the problems traditionally associated with the monitoring of distributed resources spanning multiple administrative domains to satisfy the information requirements of the operations team. The framework would take advantage of the considerable existing body of work carried out in the area of grid monitoring by aggregating existing systems and tools where provided, and allowing for the development of new sensors where none exist. A key concern was the administrative overhead involved in the deployment and management of the monitoring solution. While there were initially only six sites to monitor, the introduction of an additional twelve sites highlighted the need for an efficient solution.

V. Architecture

The architecture obeys the general principle of ensuring scalability by devolving work to the monitored nodes where possible. This is achieved through the combined use of monitoring agents local to each site, plus a central monitoring process. Traditional systems like Nagios[7] are configured at the site for the site, but this becomes an excessive burden. Here a more efficient integrated approach is taken.

A. Monitoring Agents

A single Monitoring Agent, implemented in Java, is deployed to each of the sites with the responsibility of executing service checks for hosts and services at that site. In addition to the distributed agents a central agent running at the Operations Centre determines metrics, such as site connectivity, and aggregates information from additional sources such as Site Functionality Tests[8], Grid Intrusion Detection Systems[9] and existing grid information systems, so that it may be included in the analysis and presentation process.

Configuration information specific to each site, host, or service is stored in a central database and made available to the agents via a web service, thereby minimising the amount of local site-specific configuration required. Upon start-up a remote agent queries the central configuration server for details of the hosts and services that it should monitor at its site. It then creates local monitoring objects that determine the availability of the specific service through the use of existing or

Proceedings of The Fifth International Symposium
on Parallel and Distributed Computing (ISPDC'06)
0-7695-2638-1/06 $20.00 © 2006 IEEE

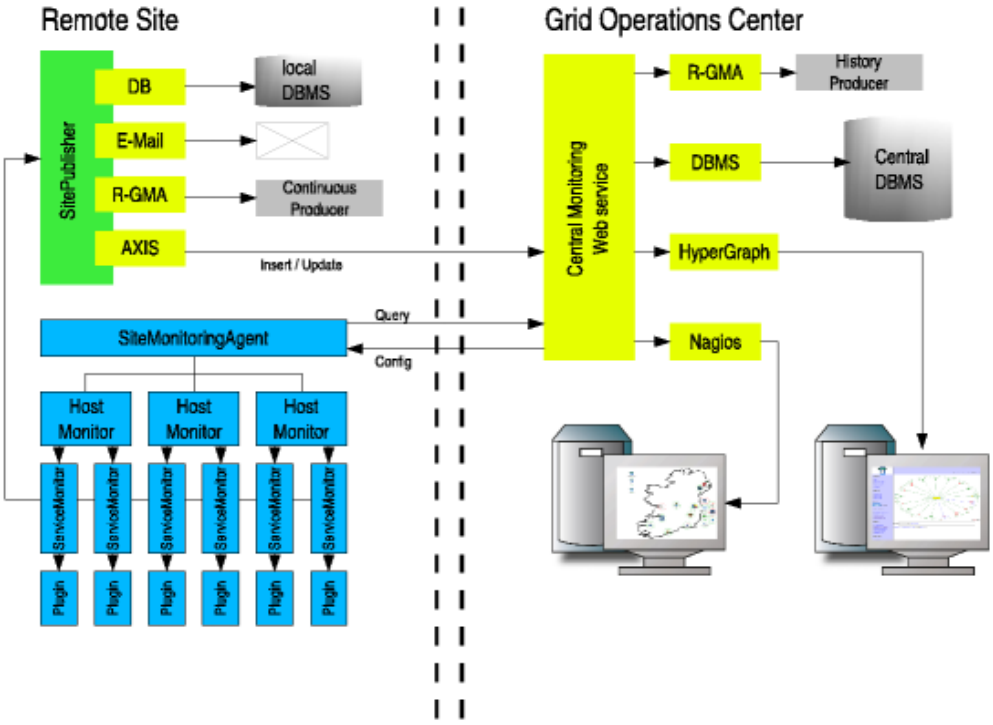

Fig. 1. Agent-based monitoring architecture

custom service checking objects. At intervals specific to each service, the monitoring objects report their status to interested subscribers via a local publisher. One such subscriber is the central monitoring service, where the results are archived and, if necessary, brought to the attention of an operator. The use of remote sensors communicating over established ports (HTTPS) not only distributes the overhead associated with the monitoring operations but also eliminates many of the connectivity problems commonly experienced with centralised network monitoring.

In order to satisfy the requirement for extensibility, the status of each monitored entity is determined by the agent using plugin modules implemented as Java objects or as Nagios-compliant plugins written in either Perl, C, or as a shell script. This allows the agent to take advantage of a wide range of existing service checks developed for the Nagios project[7]. Examples of such custom plugins include log parsing mechanisms, information system consumers and mechanisms to query a Ganglia[6] monitoring daemon for system load information.

The collected status information includes but is by no means limited to:

- The tcp connectivity to grid service ports such as GRAM, GRIS etc.
- Network reachability of the machines comprising the grid gateway
- Queue / Job activity on the Compute Elements
- The output of the Site Functionality Tests
- Grid Portal availability
- SSH reachability of the managed hosts
- Load information obtained via the Ganglia gmon daemon

B. Configuration Database

The configuration information defining what hosts and services should be monitored at each of the sites can be represented in text files or database tables, etc. In this case, advantage is taken of the database schema for version 1 of the LCG GOCDB[10] maintained at Rutherford Appleton Laboratory, UK, with several extensions. Hence the configuration information is stored in a mySQL database at the Operations Centre.

C. Central Monitoring WebService

The central monitoring process, for entirely pragmatic reasons, needs to perform the following functions:

- provision of configuration information to each remote agent
- aggregation of information from remote agents and existing monitoring tools
- archiving and publishing of monitoring/status information

It is implemented as an AXIS[11] web service hosted at the Grid Operations Centre. Configuration information is made available to the monitoring agents via a web service employing the Jakarta Database Connection Pooling mechanisms to query

the MySQL database. Using publication mechanisms similar to those embedded within the monitoring agents, status information reported back to the central server is made available to consumer APIs, via re-publisher mechanisms, or passed on to registered listeners. Currently implemented listeners include the Nagios NSCA client, JDBC, XML-RPC, and R-GMA[12] mechanisms. In addition to the central publisher mechanism, the status reports are cached in memory providing rapid access to the latest metrics for all monitored entities. Further optimizations are being explored.

VI. Presentation and Alerting

Just as in many traditional Operations Centres, monitoring information is brought to the attention of the operations team in real-time by means of wall-mounted TFT displays, web-based tools and email/messaging alerting systems. Currently four 18 inch LCD display panels are used at the Operations Centre, but soon a number of 40 inch displays will replace these. In the near future, monitoring information and alerts will also be made available to the members of the operations team remotely through the use of mobile devices running web browsers and thin clients.

A. Reports/Displays/Alerts

The Nagios host and network monitoring system is employed at the operations centre for presentation and alerting, solely in order to take advantage of its web based display and reporting functionality in addition to its comprehensive alerting mechanisms. The configuration is purely passive in that Nagios is not responsible for any service monitoring directly but rather the information gathered from the remote sensors is fed into Nagios by means of the Nagios Service Check Acceptor server daemon, allowing the status of defined hosts and services to be updated.

B. Navigation of Information

For many existing host and network monitoring tools, navigation of the information is rather neglected. Not only is it desirable that this be easy, it should also be fast, so that an experienced operator can exhibit a ”musicians touch”, particularly in emergency situations. The most promising approach thus far assumes tree structures.

HyperGraph[13] is an open source project which provides Java code to work with hyperbolic geometry and in particular hyperbolic trees. Its extensive API facilitates the visualisation of graphs and hyperbolic trees which are extremely useful when dealing with large volumes of data in a hierarchical structure.

The HyperGraph API is used as an example of an alternative methodology for the display of status information, allowing the identification of problems ’at a glance’ and manipulating the graph for an improved view of the necessary information. Navigation of the infrastructure tree is very fast, which is useful when attempting to find the location of an urgent problem.

In this case (see Figure 3), the colour of the edges between the Operations Centre node and the individual site nodes is determined by the network reachability of the gateway machine at that site. The status of the site nodes themselves is determined by the latest results of the Site Functionality Tests for that site. The colour of a host node is determined by the maximum alert value of the services monitored on that host. Nodes representing services are coloured based on the output of the pulgin for that service. The graph is constructed in such a way that hovering the pointer over a node causes further information pertaining to that node to be displayed. Table I defines the colour code used in the graph.

Although very useful, HyperGraph is not the panacea. It’s assumption of tree structures is quite restrictive. Fast hierarchical navigation, particularly relational, is also very desirable and even necessary given our reliance on R-GMA. This is under investigation.

C. Alert Analysers

Of course, alerts are themselves an interesting data set, amenable to extraction of useful information regarding stability, behaviour, etc. More in-depth analysis and intelligent alerting is made possible through the publication and aggregation of the monitoring information into the R-GMA system[14] and the use of R-GMA consumers in the form of custom alert analysers[15]. These analysers can trigger event handlers and notification mechanisms based on queries made on the R-GMA producers.

It is important to recognise that the power of the SQL queries within individual analysers may be extended by building trees of or a series of dependent analysers, allowing complex analysis of the monitoring information. Analysers may contain advanced processing logic and include historical analysis, providing alert escalation mechanisms or the investigation of correlations among service outages and higher level alerts or security events etc.

Examples of event handlers and notification mechanisms currently under development are outlined in table II

VII. Deployment

The system described has been deployed across 18 Grid-Ireland sites with the task of monitoring almost 200 services on over 70 hosts. Up to 15,000 service/host check results are reported to the central monitoring service each day. It has achieved its objectives and proves to be a valuable tool for the operations team. Its extensible nature is tested regularly as additional checks are requested.

VIII. Future Work

While the current system is production-ready and has a stable history, a number of future avenues of research are planned. These include further extension to the agent architecture, improved agent and resource management, on-demand monitoring, and the aggregation of additional information sources, particularly with respect to security information[9].

Proceedings of The Fifth International Symposium
on Parallel and Distributed Computing (ISPDC'06)
0-7695-2638-1/06 $20.00 © 2006 IEEE

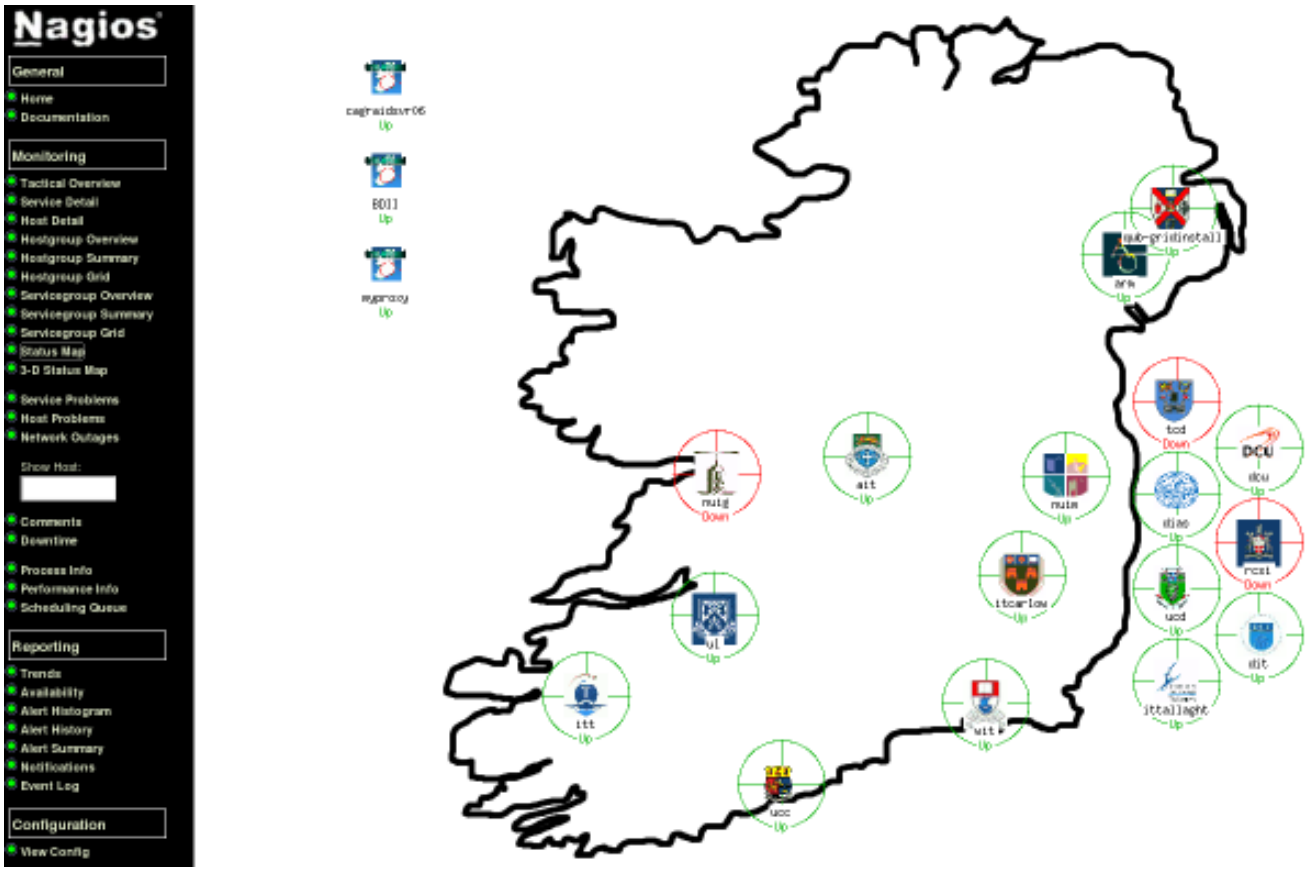

Fig. 2. Example Nagios Display

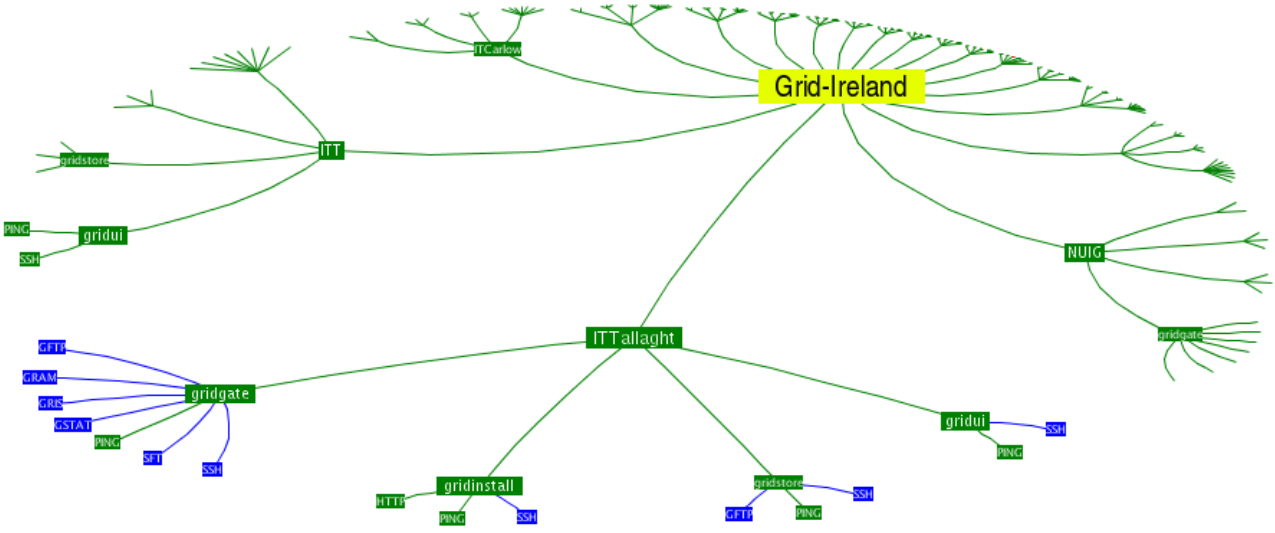

Fig. 3. A snapshot of the Hypergraph Display of Infrastructure Status in mid-navigation

Proceedings of The Fifth International Symposium
on Parallel and Distributed Computing (ISPDC'06)
0-7695-2638-1/06 $20.00 © 2006 IEEE

Colour	Connotation
Green	No problems or warnings
Amber	Warnings exist for this host or service
Red	Errors or critical warnings exist
Blue	The information for this host is considered stale
Grey	The status could not be determined

TABLE I

HYPERGRAPH INFRASTRUCTURE MONITORING KEY

Handler Name	Description
Console Handler	Prints the alert to an open console
JDBC Handler	invokes an insert statement on a JDBC resource
RGMA Handler	invokes an insert statement on an RGMA producer
MAILTO Handler	Sends an email to a user/operator detailing alert
SMS Handler	Sends a Short Message Service (text) to a user/operator
XML-RPC Handler	Invokes a method on an XML-RPC server

TABLE II

EXAMPLE ANALYSER EVENT HANDLERS

The area of real-time and historical status information analysis, with relevant warning and alerting systems, is where the majority of our future research lies. Customised presentation systems employing 3D visualisation and virtual instrumentation will also be investigated.

IX. CONCLUSIONS

This paper has outlined some of the difficulties associated with the effective monitoring of distributed computing infrastructures and has presented our efforts in the development of a new approach to the execution and mangement of grid service monitoring. Following a brief discussion of distributed versus centralised monitoring, we described some of the difficulties encountered in the deployment of monitoring tools within our own infrastructure and introduced our motivations for an agent-based solution. It has demonstrated the combined use of remote and centralised monitoring mechanisms along with the aggregation of existing information systems in order to satisfy the information requirements of a Grid Operations Centre. A number of presentation and alerting systems currently in use were also described.

Our distributed monitoring solution boasts a number of advantages. The lightweight Java agents are easily deployed at remote sites requiring minimal local configuration. Configuration management is achieved through web interfaces to the configuration database. The use of standard communication protocols has resulted in a reliable system, capable of operating within the constraints of tightly managed network security due to its use of standard ports. Site firewall rules for these ports are not typically subject to change following security audits. Where appropriate, existing tools have been used in a flexible and extensible manner, improving the efficiency of the development effort and the overall usefulness of the system. The archiving and republishing of the monitoring information makes this a valuable component on which to base future work.

X. REFERENCES

REFERENCES

[1] B. Coghlan, J. Walsh, D. O'Callaghan, Grid-ireland deployment architecture, in: P. M. Sloot, A. G. Hoekstra, T. Priol, A. Reinefeld, M. Bubak (Eds.), Advances in Grid Computing - EGC 2005, LNCS3470, Springer, Amsterdam, The Netherlands, 2005.

[2] Enabling Grids for E-sciencE (EGEE) (2006). URL `http://www.eu-egee.org/`

[3] S. Andreozzi, N. De Bortoli, S. Fantinel, A. Ghiselli, G. Tortone, C. Vistoli, Gridice: a monitoring service for the grid, Proc. Cracow Grid Workshop, Poland, December, 2003.

[4] S. Zanikolas, R. Sakellariou, A taxonomy of grid monitoring systems, Future Gener. Comput. Syst. 21 (1) (2005) 163–188.

[5] P. P. Bonnassieux F., Harakaly R., Mapcenter: an open grid status visualization tool, in: ISCA 15th International Conference on parallel and distributed computing systems, Louisville, Kentucky, USA, September 19-21, 2002.

[6] Gagnlia project documentation. URL `http://ganglia.sourceforge.net/docs/`

[7] Nagios project documentation. URL `http://www.nagios.org/docs/`

[8] Site functionality tests. URL `http://goc.grid.sinica.edu.tw/gocwiki/Site_Functional_Tests`

[9] S. Kenny, B. Coghlan, Towards a grid-wide intrusion detection system, in: P. M. Sloot, A. G. Hoekstra, T. Priol, A. Reinefeld, M. Bubak (Eds.), Advances in Grid Computing - EGC 2005, LNCS3470, Springer, Amsterdam, The Netherlands, 2005.

[10] Grid operations centre database. URL `http://goc.grid-support.ac.uk/gridsite/gocdb/`

[11] Axis - apache webservices project. URL `http://ws.apache.org/axis/`

[12] R. Byrom, B. Coghlan, A. Cooke, R. Cordenonsi, L. Cornwall, A. Datta, A. Djaoui, L. Field, S. Fisher, S. Hicks, S. Kenny, J. Magowan, W. Nutt, D. O'Callaghan, M. Oever, N. Podhorszki, J. Ryan, M. Soni, P. Taylor, A. Wilson, X. Zhu, R-gma: A relational grid information and monitoring system, Proc. Cracow Grid Workshop, Poland, December, 2002.

[13] Hypergraph project documentation. URL `http://hypergraph.sourceforge.net/docs.html`

[14] A. Cooke, A. Gray, L. Ma, W. Nutt, J. Magowan, M. Oevers, P. Taylor, R. Byrom, L. Field, S. Hicks, J. Leake, M. Soni, A. Wilson, R. Cordenonsi, L. Cornwall, A. Djaoui, S. Fisher, N. Podhorszki, B. Coghlan, S. Kenny, D. O'Callaghan, R-gma: An information integration system for grid monitoring, in: Proc.Int.Conf. Cooperative Information Systems (CoopIS'03), Catania, Sicily, 2003.

Proceedings of The Fifth International Symposium on Parallel and Distributed Computing (ISPDC'06)
0-7695-2638-1/06 $20.00 © 2006 IEEE

[15] B. Coghlan, S. Kenny, Grid-wide intrusion detection: First results after deployment, Submitted to special issue of the Journal of Parallel and Distributed Computing on Security in Grid and Distributed Systems.

Proceedings of The Fifth International Symposium
on Parallel and Distributed Computing (ISPDC'06)
0-7695-2638-1/06 $20.00 © 2006 IEEE