Towards Monitoring Security Policies in Grid Computing: a Survey

Abdulghani Suwan, Francois Siewe and Nasser Abwnawar
Software Technology Research Laboratory (STRL)
Faculty of Computing Science and Engineering
De Montfort University
Leicester, United Kingdom
{p02318242, FSiewe, p05110127}@dmu.ac.uk

***Abstract*—Grid computing systems are complex and dynamic systems and therefore require appropriate automated management, which would enable stable and reliable operation of the whole grid environment. The research community has addressed this requirement with a number of monitoring frameworks, which serve to collect data at various levels to support decision taking and management activities within grids. However, these existing solutions seem to implement little support for collecting security-related data and enforcing appropriate security policies and constraints in this respect. With an increasing role of network connections and users remotely accessing computational resources from various locations, grid systems are no longer seen as localised and isolated ecosystems, but are coming to be more open and distributed. In this light, it is becoming more and more important to enable monitoring framework with capabilities to collect security-related data and check whether these observations comply with certain security constraints. Accordingly, this paper presents a survey of existing grid monitoring systems with a goal to identify an existing gap of insufficient support for handling the security dimension in grids. The survey suggests that available grid monitoring frameworks are incapable of collecting security-related data metrics and evaluating them against a set of security policies. As a first step towards addressing this issue, the paper outlines several groups of security policies, which the authors expect to be further incorporated in their own research work, and by the wider community.**

***Keywords*—Grid; Grid computing; Monitoring System; Security policy; Policy enforcement; Survey.**

I. INTRODUCTION

Grid systems are traditionally seen as computational clusters, which serve to provide universal and steady access to the pool of resources, including computational power, data storage space, software support for deploying and running intensive computations, data analysis, etc. [1]. To a great extent, user interaction with a grid system is automated, and grid resources are allocated and provisioned automatically. It means that users are enabled to remotely access grid systems, run their computations or store their data themselves, avoiding interaction with the grid administrator. Moreover, grid users are also typically exempted from the underlying routines of handling task fragmentation and distribution, scheduling, data integrity checking, etc. All these background jobs are expected to be handled by the grid in a completely automated manner.

This flexible functionality, however, requires constant automated control and supervision to be executed by the grid administrators so as to support stable operation of the managed grid ecosystem. This in turn requires employing appropriate data collection and monitoring mechanisms, which would provide sufficient information for the interested party to take necessary actions. Broadly speaking, monitoring can be defined as a process of systematic collection of information about the current and past status of resources that are relevant in a particular scenario [2]. In the context of grid systems, which are characterised by their dynamic and complex nature, monitoring has become a particularly important task, which serves as a basis for providing reliable and cohesive services to users [3]. Since the emergence of grid computing, enabling automated monitoring capabilities has been identified as one of the key challenges and attracted attention and efforts both from the academia and the industry.

As a result, to date, grid computing in general and grid monitoring systems in particular have reached a considerable level of maturity. Moreover, grid computing is frequently seen as a precursor to cloud computing, which also results in employing already existing grid monitoring solutions to the emerging domain of cloud computing. For example, the two fundamental characteristics of cloud computing [4] – elasticity and load balancing – are supported by prompt and timely monitoring of the underlying infrastructure resources. These monitoring mechanisms were not developed from scratch, but rather relied on already existing, highly optimised and reliable solutions originating from the grid computing research.

Another recent extension to grid computing – namely, mobile grid computing [5] – can also potentially benefit from employing already existing techniques. Mobile grid computing is an emerging computing paradigm, which lies at the intersection of two research areas – namely, grid computing and mobile computing [5]. Its main concept is to extend the traditional capabilities of grids – that is, provisioning of a large pool of aggregated computational and storage resources in order to address computationally intensive tasks [6] – with computational capabilities of mobile devices over the network. From this perspective, mobile grid computing can be seen as an evolution of the grid concept from traditional, on-premises deployments to a distributed computing architecture, consisting of both computational clusters residing in a data centre and multiple mobile devices, such as smartphones, tablets and laptops, connected to the main cluster via a wireless network.

The emergence of mobile grid computing poses new challenges as to how these distributed systems should be properly monitored and managed in terms of security and data privacy. As opposed to the traditional grid architectures, where networking is not regarded as one of the primary issues to take into account, in mobile grids the networking dimension starts playing a dominant role. The wireless nature of network connections introduces new threats and makes resulting mobile grid systems vulnerable to a wide range of malicious activities such as eavesdropping, data tampering and data tracing [7]. As it will be explained below, existing grid monitoring systems currently seem to be unable to support the security dimension. Accordingly, novel appropriate monitoring and detection mechanisms are required in order to address these security issues and enable (mobile) grid systems with sufficient self-protective capabilities to maintain required quality of service (QoS). Primarily, such novel mechanisms are expected to be equipped with appropriate security policy checking and enforcement mechanisms – that is, with capabilities to check if collected security-related values are within the allowed constraints.

Given these considerations, this paper provides a survey of existing grid monitoring systems with respect to an extent to which they support monitoring of security-related aspects. The survey results suggest that the security dimension is currently beyond the capabilities of the existing grid monitoring solutions. To address this identified gap, the paper outlines several important groups of security constraints, which would complement existing grid monitoring systems with the required support for monitoring security-related aspects and enforcing security policies. Moreover, these features are also expected to act as a reference for designing and developing the authors' proposed grid monitoring system, which is enabled to support the security dimension in grids.

Accordingly, the rest of the paper is organised as follows. Section II provides an overview of security aspects and the related vocabulary of terms in the context of grid computing, and also discusses the motivation behind the presented research in more details. Section III is dedicated to the actual survey of the existing grid monitoring systems. This section surveys 13 different monitoring frameworks, paying special attention to their support for the security dimension. Section IV proceeds with a critical analysis of supported features in the examined grid monitoring systems, and identifies an existing gap related to insufficient support for enforcing security policies. Section V represents the authors' initial results to introduce the security dimension in grid monitoring systems by outlining security policies, which can be potentially monitored and enforced in grid ecosystems. Section VI concludes the paper.

II. Motivation: Enabling Security in Grids

As defined by Foster et al. [8], a security policy is a set of rules that define *security subjects* (e.g., users), *security objects* (e.g., resources) and relations between them. In the context of grid computing, security policies are typically defined using the following terms, which serve to describe grid nodes, hosts, resources and communication channels [8]:

- *Subject* is any entity, participating in a security operation. In grid computing environment, a subject is usually a user, a process operating on behalf of a user, a resource (e.g., file, computer, etc.), or a process working on behalf of a resource.
- *Credential* is information which is used to verify the identity of a subject. Most common examples of credentials include passwords and certificates.
- *Authentication* is the process by which a subject verifies its identity to a requestor, typically by providing its credentials. Authentication, in which both parties (i.e., the requestor and the authentication authority) authenticate themselves to each other at the same time is referred to as mutual authentication.
- *Object* is a resource, which is consistently protected by the relevant security policy.
- *Authorisation* is the method, which verifies whether a subject is permitted to access or use an object.
- *Trust domain* is a logical, administrative structure, within which a particular security policy is applicable and holds. In other words, a trust domain is a set of subjects and objects ruled by a single administration body and one security policy.

Using this vocabulary of terms related to security in grid environments, Foster et al. presented their Security Policy of Grid Computing and a reference architecture for implementing security-related aspects in grid systems [8]. This work established theoretical and practical foundations for designing and implementing grid environments in a secure, robust and reliable manner. The vision of implementing secure grids also depends on designing and implementing appropriate mechanisms and tools for executing continuous monitoring and control activities. Among other things, such activities may include authentication, authorisation and access control, intrusion detection, etc.

For example, with an increasing number of users accessing the (mobile) grid resources, it is becoming a challenging task to monitor and ensure that users are authorised to access grid resources and run computations. From the grid provider's perspective it is important to perform be aware of i) users' access rights and resources to be accessed, ii) users' current geo-location and connection type (i.e., a secure local connection or a far less secure and unreliable remote wireless connection), iii) the exact time at which users are accessing the grid (e.g., grid resources may be provisioned only during some periods based on the organisational policies), etc. These and other similar security-related concerns have been taken as key criteria to survey existing grid monitoring systems and examine to what extent they support the security dimension. The next section proceeds with the results of the conducted study.

III. Assessment of Available Monitoring Systems

This section aims at identifying and surveying the most prominent and widely used monitoring frameworks specifically designed and developed to monitor grid computing systems. The surveyed monitoring frameworks are widely used for grid monitoring purposes all around the globe (with a particularly wide adoption in Europe). It is worth noting that the survey aims at providing a grid-focused view on monitoring tools, and explicitly omits (numerous) cloud-oriented approaches, which are driven by different requirements (e.g., SLA satisfaction, balancing physical and virtualised resources, application performance, business goals, etc.) and tend to provide a more high-level view on monitoring the internals of the system.

Accordingly, in this survey, the main goal was to address the most relevant and important aspects of the existing grid monitoring frameworks. It has to be noted that due to space constraints the section only provides an overview, and refers the interested reader to a comprehensive study in [9], in which authors compare existing monitoring systems and classify them into application-, resource-, performance- and job status-oriented approaches. Accordingly, based on this existing study 13 grid monitoring systems have been identified, which are now considered in more details one by one in alphabetical order. The survey results are summarised in Table 1, whereas actual critical analysis and discussion of the survey results are provided in the following section of this paper.

A. Ganglia

Ganglia[1] is widely used in high-performance computing environments in order to primarily monitor computational resources [10]. Its main focus is on monitoring clusters, grids, and cloud infrastructures. Ganglia is based on carefully designed and engineered data structures and algorithms in order to achieve efficient monitoring of grid resources [9]. As claimed by its description, this system is highly optimised and advanced to be capable of monitoring clusters with more than 50,000 running hosts. However, the application scope of Ganglia is limited – it is strictly targeted at monitoring resources, and typically neglects other important areas, including security.

B. GridICE

GridICE[2] was created at Istituto Nazionale di Fisica Nucleare (INFN) in the frame of the European DataTAG project [9]. It is a monitoring system, which facilitates the process of monitoring of scattered resources in grid architectures, and can be described as multi-dimensional monitoring framework as it is capable of capturing a wide range of monitored metrics. It is equipped with data collection capabilities to gather, aggregate and display the monitored data to the user. GridICE can be configured to aggregate collected data based on user requirements and specifications – for example, to monitor certain aspects of the grid virtual organisation or the grid operation centre. GridICE is enabled with detection and notification services, and can also capture network-related statistics.

C. GridMon (UK Grid Network Monitoring)

GridMon[3] is a grid network monitoring system which monitors network-related information, aggregates the collected data and displays it to the user [11]. The system is a collection of tools which can measure such metrics as connectivity, network performance, network jitter, packet loss rate, round trip time, and TCP and UDP throughput. GridMon was developed in the context of creating a connected grid infrastructure across the UK, and is not publicly available for download and usage [9].

D. GridRM

This is another monitoring system for networks which implements the Grid Monitoring Architecture (GMA). A GridRM [12] gateway is deployed on each grid site to access information regarding local grid resources. Equipped with a relational database, it is capable of collection data from other monitoring services (e.g. MDS) over the Simple Network Management Protocol (SNMP) and presenting this data to the users via standardised views. It also provides a Web-based user interface to access monitored data remotely and run custom queries to retrieve required aggregated information.

E. G-PM/OCM-G

The OCM-G system [13] is a grid application monitoring framework, which offers online monitoring tools, configurable by the central manager, which orchestrates the monitoring process and passes monitoring requests to local monitoring agents. G-PM is a graphical extension to this system for visual performance analysis (in the form of charts, diagrams, etc.). It offers standard performance metrics and also supports creating user-defined custom metrics.

F. HTCondor Hawkeye

Hawkeye[4] is a monitoring framework, which is capable of collecting and storing data, as well as incorporating information about other computer systems [14]. The Hawkeye system can be used to monitor individual servers, clusters of servers or whole data centres. The system is also equipped with a set of components to support data monitoring in an HTCondor computational group.

G. MapCenter

MapCenter [15] is a flexible monitoring system, enabled with user interface to present and visualise run-time information on services and applications running on the grid. It relies on R-GMA to automatically collect data, and MDS for remote access over the network. It also supports dynamic discovery, based on efficient and transparent monitoring techniques, which enables rapid deployment of the MapCenter system in multiple grid environments.

H. Monitoring and Discovery System (MDS)

MDS[5] is one of the most prominent monitoring frameworks widely used as a part of Globus Toolkit (GT) – a toolkit for building and managing grids – or independently.

[1] http://ganglia.sourceforge.net/

[2] http://sourceforge.net/projects/gridice/

[3] http://gridmon.dl.ac.uk/gridmon/graph.html

[4] http://research.cs.wisc.edu/htcondor/

[5] http://toolkit.globus.org/toolkit/mds/

Hierarchically structured, it enables management of static and dynamic information related to the current status of grid components. MDS provides an index service, which is used by managed grid systems to deliver collections of low-level data via a special registration protocol and caching mechanism so as to minimise the amount of non-stale data being transferred [16].

I. Mercury Grid Monitoring System

Mercury[6] was designed to meet the requirements of grid performance monitoring. It supports data monitoring and collection of metrics based on both pull and push models, and is targeted at controlling grid resources and applications in a scalable way. Mercury partially implements the GMA, and also follows a modular approach, which facilitates simplicity, proficiency, convenience and low insensitivity.

J. Nagios

Nagios[7] [17] enables resource and application monitoring based on an extensible architecture. It offers various monitoring services such as monitoring of host resources (e.g., CPU/memory utilisation, response times, etc.) and monitoring of network services and protocols (e.g., SMTP, POP3, HTTP, PING etc.).

K. R-GMA

R-GMA [18] is based on the GMA, which uses relational model for data storage. This allows the users the ability to run customised SQL-like queries to retrieve required information from the system. R-GMA also offers its users a global view on the grid system, including service availability and application monitoring [9].

L. Scalea-G

Scalea-G [19] is a generic performance analysis and monitoring system for grid systems. It implements the Open Grid Services Architecture[8] (OGSA) and provides a setup for performance analysis and monitoring of various parameters belonging to network resources, computational resources and applications. Both push and pull data collection models are supported to enable scalable and flexible monitoring solution. Scalea-G also supports dynamic source code instrumentation to enable tracing and profiling of grid applications.

M. visPerf

visPerf [20] is another grid monitoring system, which supports visualisation of grid resources. This system uses agents which can extract necessary information from log files and/or can access the grid middleware API. Developed in the frame of GridSolve project[9], it allows connecting to NetSolve servers for accessing system information for monitoring purposes.

IV. Monitoring Systems Comparison

The results of the survey on grid monitoring frameworks are summarised in Table 1. This presented classification relies on a taxonomy, which includes 5 different dimensions (i.e. orientations) describing the monitoring activity, which may be supported by a particular monitoring system. Accordingly, if a framework supports particular type of monitoring, it is marked with a "+" sign; otherwise, it is marked with a "-" sign.

The following list explains each of the dimensions in more details to help the reader understand the survey results:

- *Application-oriented monitoring* targets at the internal behaviour of a grid application and/or its components, e.g. its memory usage or the execution time of a specific routine.
- *Job status-oriented monitoring* is focussed on the execution status of a submitted job, rather than the actual application executing it. In its simplest form, such kind of monitoring is configured to check whether a job is still running, already completed or failed.
- *Resource-oriented monitoring* handles data belonging to the infrastructure level of a grid ecosystem – that is, underlying hardware resources. In the first instance monitoring systems of this kind target at the overall utilisation of computing, storage and networking resources within a grid system.
    - Accordingly, *computing resources* refer to grid hardware resources, which are often seen as the key characteristic of grid computing, which, in the first instance, is expected to offer computing power to the user. Typically, to access these computing resources, users are required to deploy and run their own arbitrary applications, or, alternatively, grid systems allow configuring pre-deployed software with custom user data for further execution.
    - The second important element in grid resources is *storage*, which refers to hardware resources offered as services to enable users to store and retrieve arbitrary amounts of their data.
    - The *networking resources* are typically not directly exposed to the user, but nevertheless, it is important to monitor these resources, as it is a vital component, enabling the underlying communication between computing resources, storage resources, and the users' remote computers.
- *Performance-oriented monitoring* is intended to collect required information to determine the performance of the whole grid system or an individual application. As opposed to the application-oriented monitoring, it targets at collecting quantitative data, which will provide precise numeric values. In this respect, three monitoring techniques, used to collect this required information, can be identified. It is worth noting that these techniques are not mutually exclusive and are often used together to achieve better results.

[6] http://www.lpds.sztaki.hu/mercury/

[7] https://www.nagios.org/

[8] http://toolkit.globus.org/ogsa/

[9] http://icl.cs.utk.edu/netsolve/

- *Tracing* is used to collect monitored events occurred during operation of the system and record them in a trace file.
- *Sampling* is used to collect information by taking samples of the required measurings within pre-defined time intervals.
- *Profiling* can be defined as the process of aggregating and filtering large amounts of (noisy) information, which serves to provide a more holistic and integrated view on the performance of the monitored system.

- *Security policy-oriented monitoring* is the last but not the least aspect, which was taken into consideration when surveying existing grid monitoring systems. This type of monitoring serves to provide a wide range of information related to grid security (e.g., user access rights and privileges, remote access to the grid, state of the network, etc.), and check if the observed values comply with corresponding security policies. Examples of such policies include putting various kinds of constraints on jobs, users, geo-location, resources, etc.

Table 1. Grid monitoring systems survey summary.

Monitoring framework	Type of monitoring								
	Application-oriented	Job status-oriented	Resource-oriented			Performance-oriented			**Security policy-oriented**
			Computing	Storage	Network	Sampling	Tracing	Profiling	
Ganglia	-	-	+	+	+	+	+	-	-
GridICE	-	-	+	+	-	+	+	-	-
GridMon	-	-	-	-	+	+	-	-	-
GridRM	-	+	+	+	+	+	-	-	-
G-PM/OCM-G	+	-	+	-	-	+	+	-	-
HTCondor Hawkeye	-	-	+	+	+	-	-	+	-
MapCenter	-	-	-	-	-	+	-	-	-
MDS	+	+	+	+	+	+	-	+	-
Mercury	+	-	+	+	+	+	+	+	-
Nagios	-	-	+	+	+	+	+	-	-
R-GMA	+	+	+	+	+	+	+	+	-
Scalea-G	+	-	+	-	+	+	+	+	-
VisPerf	-	+	+	+	+	-	-	+	-

As it is seen from the table, existing grid monitoring approaches seem to support the four traditional dimensions (i.e., application-, job status-, resource, and performance-oriented types of monitoring) to a lesser or greater extent. However, the situation changes when it comes to monitoring security-related aspects of a grid system and checking/enforcing associated security policies. The conducted survey suggests that existing systems were designed and implemented with little support for this important dimension, and tend to provide minimum capabilities to execute even simple security checks.

The lack of support for checking security policies can be potentially explained by a number of reasons. First, grid computing has been historically associated with relatively localised deployments within a single data centre [21] (as opposed, for example, to a highly-distributed and network-dependent cloud computing environments), where number of network connections was limited and stable, and, therefore, there was no real pressing demand for enforcing various security checks. Second, grid computing has never been seen as a commercial product to be offered to a wide range of customers [22]. They have been primarily serving scientific purposes, and therefore the number of users accessing grid resources is typically limited and easily controlled. This again did not require implementing sophisticated security monitoring mechanisms. The third reason is more pragmatic – since there was relatively little interest and support from the industry, which would drive and advance the security research in grids, research efforts focussed on more relevant issues. In this light, cloud computing can be seen as a representative example; it is a highly commercialised and industry-driven research area, where security and privacy have been admitted by the major market influencers to be the key preventing factor since the very emergence of clouds [23].

V. Next Steps: Towards Monitoring Security Policies in Grids

Given the results of the survey, several considerations and suggestions can be drawn to create grid monitoring systems capable of capturing various security-related metrics and checking them against a set of predefined security policies. As it was already mentioned, a policy checking mechanism may rely on putting and checking a number of constraints, which, when violated, are supposed to indicate a potential security breach. Accordingly, below is a (non-exhaustive) list of desired security-related constraints for a monitoring framework, which are expected to have the potential to increase the overall security of grid ecosystems.

- *User-based constraints* may serve to restrict access to the grid for specific users, limiting the number of users accessing grid resources simultaneously or the number of users from the same virtual organisation, etc.
- *Time-based constraints* are intended to govern the exact time frame, during which users are allowed to access grid resources and execute their jobs.
- *Location-based constraints* are required to restrict access to grid resources from specific remote locations. With the emergence of mobile grids, which are associated with a high number of temporary and potentially insecure connections, this feature is seen of paramount importance.
- *Resource-based constraints* deal with checking the amount of resources allocated to individual users. These can be coarse-grained (e.g., number of computational nodes

within a grid) or more fine-grained (e.g., number of parallel jobs to be executed, amount of CPU/memory resources within a single node, etc.).

Accordingly, violation of the above-mentioned constraints leads to executing certain reactive actions so as to maintain the security and, as a consequence, stability of the grid ecosystem. Respectively, such reactive actions might include (temporary or permanent) access restrictions for specific users from specific locations at specific times to specific grid resources. These high-level guidelines are expected to act as a reference for devising a set of grid security policies and a corresponding policy enforcement mechanism. These two key components constitute the architecture of the future Grid Security Policy Monitoring System (GridSPMS) – a system, which is intended to address the identified gap of insufficient support of the security dimension in grid monitoring systems.

VI. Conclusion

The central concern of this paper is the lack of appropriate mechanisms for monitoring the security dimension and enforcing respective security policies in grid environments. Despite the fact that considerable research efforts have been put into designing and implementing secure architectures for grids, corresponding grid monitoring frameworks seem to be incapable to incorporate the required security aspects. As suggested by the conducted survey of 13 existing grid monitoring frameworks, albeit they seem to successfully collect and analyse data at various levels, they are not designed to collect security-related data and, accordingly, check if it complies with certain security constraints. As a first step towards addressing this identified gap, a non-exhaustive list of potential security policies was outlined, which are expected to be incorporated by future grid monitoring frameworks. Moreover, these high-level guidelines serve to design and develop a framework, which will enable monitoring the security dimension in grids and enforcing corresponding security policies. Currently, the proposed GridSPMS is being implemented, and "work-in-progress" results are expected to be published in the near future.

References

[1] I. Foster and C. Kesselman, "What is the Grid," *Three Point Checkl.*, vol. 20, 2003.

[2] A. G. Ganek and T. A. Corbi, "The dawning of the autonomic computing era," *IBM Syst. J.*, vol. 42, no. 1, pp. 5–18, 2003.

[3] B. Tierney, R. Aydt, D. Gunter, W. Smith, M. Swany, V. Taylor, and R. Wolski, "A grid monitoring architecture," 2002.

[4] P. Mell and T. Grance, "The NIST definition of cloud computing," *Natl. Inst. Stand. Technol.*, vol. 53, no. 6, 2009.

[5] A. Litke, D. Skoutas, and T. Varvarigou, "Mobile grid computing: Changes and challenges of resource management in a mobile grid environment," in *5th International Conference on Practical Aspects of Knowledge Management*, 2004.

[6] I. Foster and C. Kesselman, Eds., *The Grid: Blueprint for a New Computing Infrastructure*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999.

[7] A. Bichhawat and R. C. Joshi, "A Survey on Issues in Mobile Grid Computing," *Int J Recent Trends Eng. Technol.*, vol. 4, no. 2, 2010.

[8] I. Foster, C. Kesselman, G. Tsudik, and S. Tuecke, "A security architecture for computational grids," in *Proceedings of the 5th ACM conference on Computer and communications security*, 1998.

[9] M. Gerndt, R. Wismüller, T. U. München, Z. Balaton, G. Gombás, P. Kacsuk, Z. Nemeth, N. Podhorszki, H. Truong, U. Wien, T. Fahringer, U. Innsbruck, E. Laure, M. Bubak, and T. Margalef, *Performance Tools for the Grid: State of the Art and Future*. 2004.

[10] M. Massie, B. Li, B. Nicholes, V. Vuksan, R. Alexander, J. Buchbinder, F. Costa, A. Dean, D. Josephsen, P. Phaal, and others, *Monitoring with Ganglia*. O'Reilly Media, Inc., 2012.

[11] M. Leese and R. Tasker, "Building the e-Science Grid in the UK: GridMon-Grid Network Performance Monitoring," 2003.

[12] M. Baker and G. Smith, "GridRM: an extensible resource monitoring system," in *2003 IEEE International Conference on Cluster Computing, 2003. Proceedings*, 2003.

[13] M. Bubak, W. Funika, and R. Wismüller, "The CrossGrid performance analysis tool for interactive Grid applications," in *Recent Advances in Parallel Virtual Machine and Message Passing Interface*, Springer, 2002.

[14] T. Tannenbaum, "HawkEye: A Monitoring and Management Tool for Distributed Systems," 2003. [Online]. Available: http://research.cs.wisc.edu/htcondor/CondorWeek2003/presentations/leroy_hawkeye.ppt.

[15] F. Bonnassieux, R. Harakaly, and P. Primet, "MapCenter: An Open Grid Status Visualization Tool," in *proceedings of ISCA 15th International Conference on parallel and distributed computing systems*, 2002.

[16] K. Czajkowski, S. Fitzgerald, I. Foster, and C. Kesselman, "Grid information services for distributed resource sharing," in *High Performance Distributed Computing, 2001. Proceedings. 10th IEEE International Symposium on*, 2001.

[17] E. Imamagic and D. Dobrenic, "Grid Infrastructure Monitoring System Based on Nagios," in *Proceedings of the 2007 Workshop on Grid Monitoring*, New York, NY, USA, 2007.

[18] A. Cooke, A. J. G. 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, and D. O'Callaghan, "R-GMA: An Information Integration System for Grid Monitoring," in *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*, R. Meersman, Z. Tari, and D. C. Schmidt, Eds. Springer Berlin Heidelberg, 2003.

[19] H.-L. Truong and T. Fahringer, "SCALEA-G: A Unified Monitoring and Performance Analysis System for the Grid," in *Grid Computing*, M. D. Dikaiakos, Ed. Springer Berlin Heidelberg, 2004.

[20] D. Lee, J. J. Dongarra, and R. S. Ramakrishna, "visPerf: Monitoring Tool for Grid Computing," in *Computational Science — ICCS 2003*, P. M. A. Sloot, D. Abramson, A. V. Bogdanov, Y. E. Gorbachev, J. J. Dongarra, and A. Y. Zomaya, Eds. Springer Berlin Heidelberg, 2003.

[21] N. Bessis, E. Asimakopoulou, T. French, P. Norrington, and F. Xhafa, "The Big Picture, from Grids and Clouds to Crowds: A Data Collective Computational Intelligence Case Proposal for Managing Disasters," in *2010 International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC)*, 2010.

[22] I. Foster, Y. Zhao, I. Raicu, and S. Lu, "Cloud computing and grid computing 360-degree compared," in *Grid Computing Environments Workshop, 2008. GCE'08*, 2008.

[23] R. Moreno-Vozmediano, R. S. Montero, and I. M. Llorente, "Key Challenges in Cloud Computing: Enabling the Future Internet of Services," *IEEE Internet Comp.*, vol. 17, no. 4, 2013.