An Alarms Service for Monitoring Multi-domain Grid Networks

Charaka Palansuriya, Jeremy Nowell, Florian Scharinger, Kostas Kavoussanakis, and Arthur S. Trew

EPCC, The University of Edinburgh,
Mayfield Road, Edinburgh, EH9 3JZ, UK
{charaka,jeremy,florian,kavousan,arthur}@epcc.ed.ac.uk

Abstract. Effective monitoring of multi-domain Grid networks is essential to support large operational Grid infrastructures. Timely detection of network problems is an essential part of this monitoring. In order to detect the problems, access to network monitoring data that exists in multiple organisations is necessary. This paper presents an Alarms Service that supports monitoring of such multi-domain Grid networks. The service allows timely detection of networking problems based on pre-defined, user-configurable conditions. The requirements gathered from real users for monitoring the networks are discussed. The paper shows how multi-organisation data access is resolved with the use of a standards-based access mechanism. The architecture of the Alarms Service is discussed, providing the reasons behind the design decisions where appropriate. A description of the current implementation of the Alarms Service and its deployment is provided.

Keywords: Alarms, multi-domain networks, Grid, federated networks, network monitoring, Network Monitoring Working Group (NM-WG).

1 Introduction

Grid infrastructures rely on networks as a fundamental component. Large Grid infrastructures are typically composed of many different federated network domains, covering local campus networks, National Research and Education Networks (NRENs), and continental scale research networks such as GÉANT2. Effective monitoring of these federated networks is fundamental in maintaining efficient operational Grids. However, different network domains are typically heterogeneous in both their administration and operation, and may be monitored using a variety of different software tools. Access to the network monitoring data gathered within each domain by these different tools is essential to provide a Grid-wide view of network health.

Our previous work [1] focused on providing access to heterogeneous federated network monitoring data, making use of standards developed by the OGF Network Measurements Working Group (NM-WG) [2]. During the course of this work it became clear that there was a strong requirement for alarms to be raised based on the network status. Grid and network operators need access to network monitoring data to diagnose problems in the network, but just as importantly they need alarms to notify

T. Doulamis et al.: (Eds.): GridNets 2009, LNICST 25, pp. 69–78, 2010.

them of such problems as soon as they occur. Therefore, we have developed a software framework that accesses network monitoring data, analyses the data collected, and raises alarms based on pre-defined, user-configurable conditions. We call this software framework the "Alarms Service". The service allows the timely detection and trouble-shooting of networking problems, reducing the chances of users of the networks encountering the problems (or symptoms of the problems) before the operators.

The requirements for the Alarms Service described in this paper have been largely gathered from members of the Large Hadron Collider Optical Private Network (LHCOPN) [3]. The architecture developed uses sources of data such as those provided by perfSONAR [4], and uses NM-WG standards to access the data. An implementation of the Alarms Service has been developed based on this architecture to monitor the LHCOPN and to obtain feedback from network operators and other potential users.

This paper describes various technical aspect of the Alarms Service. The remainder of the paper is organised as follows. Section 2 describes the motivation behind the work; section 3 highlights the requirements gathered from the users and network operators; section 4 provides a detailed description of the architecture; section 5 describes the current implementation; section 6 mentions the present deployment; and finally section 7 provides conclusions and future work.

2 Motivation

The motivation for the Alarms Service work came from speaking to various network operators during our previous work. The discussions revealed that it is both unrealistic and too demanding for operators of Grid and Network Operating Centres (GOCs and NOCs respectively) to continuously monitor their networks using historical data. They require an alarm system that is capable of alerting them to networking problems as they arise. These problems can then be investigated in detail using the other tools available to them.

Since the different parts of a federated network belong to different organisations, the first issue that operators face is how to access network monitoring data from other organisations, for example SNMP access to remote routers is not allowed. Due to such data access issues, it is difficult to use off the shelf alarm systems. The data access mechanism used by the Alarms Service is therefore a fundamental part of its architecture, providing a solution to the problem of cross-domain data sharing.

Network operators do not want to be restricted in their choice of network monitoring framework. This presents another issue to be solved in order to gain access to the monitoring data that is necessary to analyse for alarms conditions. That is, how to access monitoring data collected by heterogeneous monitoring frameworks. Clearly, a standard interface such as the one developed by the NM-WG group is necessary.

Even with a standard interface, access to data sources at different locations and belonging to various organisations is necessary. This requires knowledge about the location of relevant data sources, access restrictions and so on.

A combination of different data sources is necessary to monitor certain alarm conditions. For example, router output packet drops may be stored in one data source

whilst utilisation is stored in another. A combined analysis of monitoring data for the two metrics is necessary, for example, to work out whether there is some network fault, rather than high utilisation, causing output drops. The necessary integration of these data sources is not trivial in a Grid or a federated network.

The above factors led to a ground-up development of an alarms framework. Prior to commencing the development work more specific user requirements were gathered.

3 Requirements

A list of prioritized alarm conditions were gathered from the operators of the LHCOPN. Indeed, the LHCOPN will be the first user of the Alarms Service. The most important conditions were determined to be as follows:

1. Routing Alarm: If the path, as determined by traceroute [5], changes and there are no network links down between the source and destination, then raise an alarm.
2. Routing Out of Network Alarm: If the path changes and one of the hops is outside the preferred (federated) network, then raise an alarm.
3. Interface Errors Alarm: If a router interface has input errors at a rate above a threshold then raise an alarm.
4. Interface Congestion Alarm: If a router interface drops packets at a rate above a threshold, whilst the link utilisation is below another threshold, then raise an alarm.

LHCOPN stated that the Routing Alarm is the most important since no suitable tool is available to monitor this condition. This alarm would indicate a possible re-route over another Tier1[1] site, thereby overburdening that site and affecting the network performance.

The Routing Out of Network Alarm was also indicated as essential, since this would signal that the network traffic is being routed outside the fast (10 Gbps) Optical Private Network (OPN) and there will be a significant performance degradation. There are also issues such as security, confidentiality and violation of Service Level Agreements (SLAs) to consider when such a routing problem occurs.

The Interface Errors Alarm was indicated as essential since it flags circuit errors that need to be investigated by the Network Operation Centres (NOCs). The LHCOPN needs to sustain very fast (several Gbps) data flows to support the transfer of the huge amount of data produced by the LHC at the Tier0 site at CERN to the Tier1 sites. If there are any interface errors then this will lead to packet drops, making it difficult to achieve the required performance.

The Interface Congestion Alarm was indicated as useful as this would indicate other network faults, apart from high network utilisation, causing output drops and thereby impacting data flow speed.

Further requirements for an Alarms Service were generated with the help of the LHCOPN as well as DANTE [6] and WiN-Labor [7]. In addition to the alarm conditions themselves, these include requirements that the status of the alarms MUST [8] be accessible via a web-based dashboard, and that an alarm SHOULD display what

[1] Note that data from CERN, i.e., Tier0, gets distributed to first level processing and storage centres called Tier1.

action is to be pursued to solve the problem. Users SHOULD be notified of alarms when they arise, for example via email. Another important requirement is that the alarm conditions as well as their threshold values MUST be configurable. Due to the distributed nature of the network monitoring data on which the alarms are raised, the alarm service MUST be able to access multiple data sources. The history of alarms SHOULD be available.

4 Architecture

4.1 Overview

The above motivating factors and requirements led to the development of the architecture shown in Fig. 1.

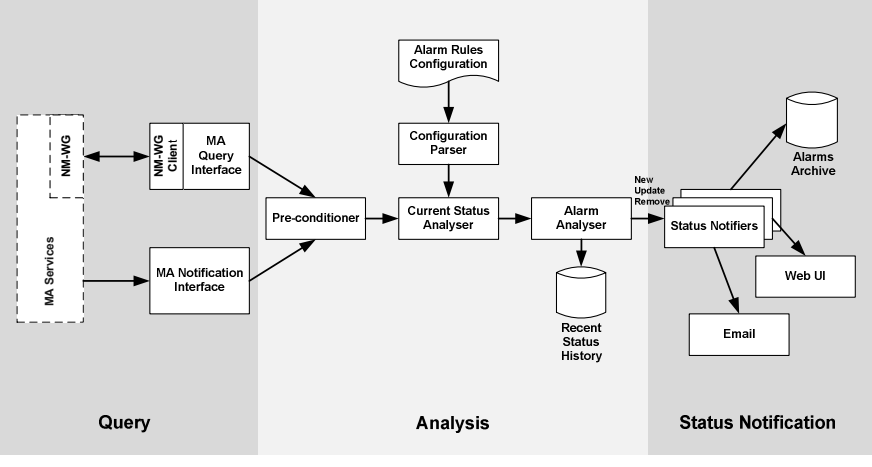

Fig. 1. The Alarms Service Architecture

The Alarms Service expects to utilise various external sources of network measurement data, known as Measurement Archives (MAs), shown in dotted lines in Fig. 1. The data may be obtained from an MA on request via the MA Query Interface; or alternatively by allowing a data source to send status updates via the MA Notification interface. This latter case is especially interesting if the data source already does some processing of the underlying measurement data, and is able to determine if there is a problem itself. The data received by the MA Query Interface and the MA Notification Interface is fed into the Pre-Conditioner component for cleaning. The Pre-Conditioner performs various tasks such as removal of any *outliers* (i.e., a single measurement can be significantly off the norm and does not qualify for an alarm to be raised) [9] and then presents data to the Current Status Analyser. The Current Status Analyser uses the cleaned up measurement data along with the configuration details (e.g., alarm conditions and how often to analyse) to check whether an alarm condition has been reached. When an alarm condition is reached, the Current Status Analyser

informs the Alarm Analyser. The Alarm Analyser checks the Recent Status History store to detect whether it is a new alarm, if it was already detected or if an existing alarm is not valid anymore. The Alarm Analyser can also be used to detect conditions such as *flapping* (i.e., when an alarm rapidly changes state between on and off). The Alarm Analyser informs registered notifiers about status changes (new, update or remove alarm). Multiple notifiers, implementing the Status Notifier interface, fulfil different purposes. Examples include the archival of the alarm history in a database, the email notification of appropriate people about new alarms, and the display of the current alarm status on a web page.

The architecture is both flexible and extensible. It allows the use of multiple data sources, new alarm conditions via an alarms configuration file, and different notification mechanisms (e.g., SNMP traps).

As shown in Fig. 1 (by the shaded regions), the architecture conceptually consists of three main functional blocks: Query, Analysis and Status Notification. These functional blocks are discussed below. In order to help the discussion an analysis of the metrics necessary to monitor the alarms conditions and where to get the data for these metrics is presented. The NM-WG schema used is also discussed.

4.2 Metrics and Measurement Archives

In order to detect the alarm conditions it is necessary to monitor many different metrics; for example, route-changes, network link status, router packet drops and network link utilisation. These metrics are monitored by different tools and then made available through Measurement Archives (MAs). The Alarms Service has to be able to access the different MAs to obtain the required measurement data.

For instance, to monitor the Routing Alarm condition, it is necessary to obtain data about route-changes and network link status. The route-change information can be obtained from the perfSONAR Hades system [10]. Hades returns data based on traceroute measurements. However, to obtain information on the network link status it is necessary to access another MA; in this case the perfSONAR E2E Link Management system [11]. The E2E Link Management System supplies information on the status of an end-to-end network link by aggregating information from each segment of the path.

Similarly, to monitor the Interface Errors Alarm the Alarm Service requires data for router interface errors, whilst to monitor the Interface Congestion Alarm, data for router packet-drops and link utilisation are needed. This data can be accessed via a perfSONAR RRD MA [4], available as part of the perfSONAR MDM bundle. It is quite possible that each network administrative domain provides its own MA for its own routers; therefore the Alarms Service needs to be capable of accessing several different MAs simultaneously.

4.3 NM-WG Schema

As identified when discussing the motivation for this work, accessing different MAs is facilitated by the use of standard interfaces wherever possible, which in this case means interfaces following the schema defined by the OGF NM-WG [2]. These are XML schemas, which are defined for information such as the subject of a

measurement (e.g., a network path or router interface), the metric measured (e.g., link utilisation or round-trip time), the time of a measurement, and the measurement data itself. The schemas are used by the Alarms Service to send a Request for particular measurement data, and then receive the corresponding Response containing the data, sent by an MA.

4.4 Query

Referring back to Fig. 1, one of the three main functional blocks of the architecture is the Query block, which is responsible for accessing various data sources and obtaining measurement data for metrics of interest. The Query block consists of an MA Query Interface (to "pull" data) and an MA Notification Interface (to "push" data).

The MA Query Interface uses the NM-WG schema to request and retrieve measurement data. More specifically, MA data is accessed via an NM-WG Web service interface. The use of this standard interface enables uniform access to heterogeneous MAs such as the perfSONAR RRD MA and perfSONAR Hades MA. Even though the data required by the current Alarms Service can be accessed via this NM-WG interface (since all necessary MAs provide such an interface) the architecture is flexible enough to add an MA query interface for non-NM-WG compliant data sources.

The MA Notification Interface is designed to allow the measurement data sources to push their data, or potentially aggregated measurements, to the Alarms Service. The idea behind this is that the Alarms Service a) does not need to pull data at regular intervals and b) can be used to combine pre-processed measurement data or even alarms from other systems.

4.5 Analysis

The components in the Analysis functional block form the Alarms Engine. The Alarms Engine consists of Pre-conditioner, Configuration Parser for rules, the Current Status Analyser and the Alarm Analyser. The engine is based on a user-configurable rules-based system. Rules provide a natural way to define alarms and are used in many different alerting systems (e.g., for monitoring values of shares in a stock market). Within the Alarms Service, they give flexibility in defining the conditions when an alarm should be raised. The user can define these conditions without using other programming skills, allowing them to be easily fine-tuned as part of routine maintenance. An *ad hoc* mechanism could have been used, but that would have "reinvented the wheel" and could have led to usability issues. The alarm rules are defined using the following structure:

```
rule "<rule name>"
 when
  <parameterised alarm condition>
 then
  <consequence>
end
```

An example of a rule specifying an Interface Congestion Alarm is shown below:

```
rule "Interface Congestion"
 when
  A router has output drops > "0" but utilisation is < "80" percent
 then
  Log : "Router has output drops"
  Raise InterfaceCongestionAlarm
end
```

The Alarms Engine triggers the querying of all configured MAs at user-defined intervals. It first constructs the query for each measurement subject and metric to be sent to the associated MA. After having retrieved the monitoring data for these via the NM-WG Web service interface, it then packs the data into internal data containers. The data in these containers is compared against the user-configured alarm rules within the Current Status Analyser. If the measurement data satisfies the condition of a rule, this rule gets activated and the associated data (subject, metric, alarm condition) is compiled and forwarded to the Alarm Analyser. The Alarm Analyser is designed to compare the raised alarm against the previous history of the subject in question, and to decide if the alarm should be raised via the Status Notification (see next section) or not. The Alarms Analyser resets an alarm if its condition is no longer there.

The separation between analysing the current status and the status history of a subject allows the detection of certain patterns, e.g. flapping between two states.

4.6 Status Notification

A status notification mechanism is provided to plug-in different notification methods. For example, status notifiers can be implemented to send emails, implement SNMP traps, write to web interfaces or integrate with a network weather-map system.

Concrete implementations of this interface register with the Alarm Analyser (described above). The interface provides three different types of notifications to declare if an alarm for a subject is “new”, “updated” or “cleared”. It is then up to the concrete notification implementation to process or ignore an alarm, depending on the type of notification.

5 An Implementation

The Alarms Service described above is implemented using the Java programming language. Java provides portability to different platforms and good performance.

The NM-WG requests and responses are implemented as described in [1]. The NM-WG client shown in the architecture uses Apache Axis2 to submit XML requests conforming to the NM-WG schema. Similarly, XML responses received are checked for compliance with the NM-WG schema prior to further processing.

The Alarms Engine uses Drools [12] to implement a rules-based system. The alarm rules specify which state of the measurement data containers resembles an alarm condition. However, this would lead to a low-level, programming-like configuration

syntax, which would not be particularly user-friendly. Hence, the "Domain Specific Language" feature of the Drools library is used to create a more natural and user-friendly configuration language. This language is designed to read like a normal English sentence (e.g., `A router has output drops > "0"`) and is automatically mapped to the actual internal rules code, lowering the learning curve for the configuration significantly.

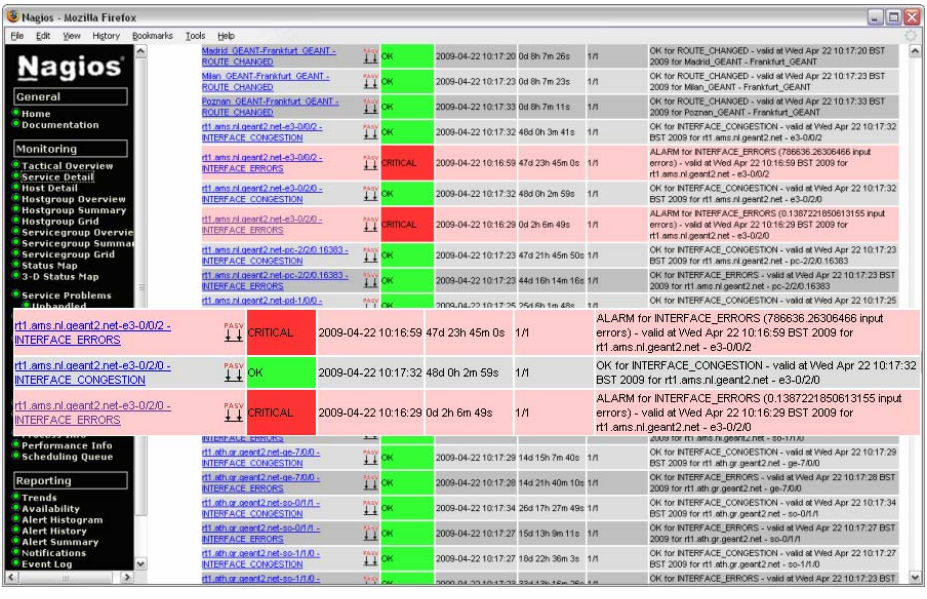

Fig. 2. Nagios web dashboard showing two alarms (replicated for clarity in the zoomed area) being raised (marked as "CRITICAL") for the GÉANT network

In order to help monitor the LHCOPN, an early version of the Alarms Service has been produced and released. This version is capable of monitoring the four alarm conditions described in this paper. It displays alarms on a Nagios-based [13] web dashboard (see Fig. 2). Email notifications can also be delivered via Nagios. Note that the Alarms service is an independent, Java application that is only loosely coupled with Nagios. This allows us to support in the future other web-based and notification mechanisms, as required.

Of the components in the architecture shown in Fig. 1, the pre-conditioner, the Alarms Archive, the Recent Status History and the MA Notification Interface have not been implemented yet. The pre-conditioner is delayed until there is a clear requirement for it. For example, whether "false positives" (i.e., an alarm being raised when there is no reason to do so, which may be due to a data anomaly) are noted in practice. See the deployment section for more on false positives. If Nagios is used as the web interface then the Alarms Archive is not required. This is because Nagios maintains its own alarms archive (Nagios refer to this as "Alert History"). The Recent Status History is also not implemented, as again Nagios maintains its own recent

status. The Measurement Archive Notification Interface has not been implemented yet as none of the MAs currently used to supply data are capable of sending notifications – they all require active queries.

6 Deployment

The early version released [14] is currently deployed to monitor the first few Tier1 sites of the LHCOPN that currently supply monitoring data. More sites, including the Tier0 (CERN), will be added to the deployment of this Alarms Service as and when data become available from them. The deployment is carried out and maintained by the DANTE Service Desk. This version is successfully monitoring and raising relevant alarms for the configured network paths and router interfaces. So far no false positives have been noted from this deployment.

The same version of the Alarms Service has successfully been deployed by the authors at EPCC to monitor some of the networks for which data is publically available, for instance the GÉANT2 and GARR networks.

7 Conclusion and Future Work

This paper introduced an Alarms Service for monitoring multi-domain Grid networks. The service uses a flexible architecture to address the requirements gathered from the users and operators of such networks. The service uses an NM-WG standard interface to reduce the complexities of accessing monitoring data from heterogeneous sources. The service also uses a rules-based user configurable alarms definition mechanism. It is envisaged that with the adoption of the Alarms Service by underlying network infrastructures – such as the LHCOPN – large Grid infrastructures will benefit from smoother operations with timely troubleshooting.

At present, the Alarms Service has been deployed to monitor the LHCOPN. Projects such as GÉANT3 and DEISA2 [15] have noted an interest in the evaluation of the Alarms Service as part of their multi-domain network monitoring solution. Based on the deployments and interest, some of the unimplemented architectural components and other capabilities may get added to the Alarms Service.

Acknowledgments

We would like to thank: DANTE [6] for helping us with gathering requirements; LHCOPN [3] for contributing requirements and providing useful feedback following the various demonstrations of the software; and WiN-Labor [7] for help specifying the behaviour of the Alarm Service, providing prompt access to Hades and helping with MA-related issues.

This work has been funded by the UK Joint Information Systems Committee (JISC) and by the European Union under DEISA2 (RI-222919).

References

1. Kavoussanakis, K., Phipps, A., Palansuriya, C., Trew, A., Simpson, A., Baxter, R.: Federated network performance monitoring for the grid. In: 3rd International Conference on Broadband Communications, Networks, and Systems, San Jose, California, USA (2006)
2. Network Measurements Working Group, https://forge.gridforum.org/projects/nm-wg
3. Large Hadron Collider Optical Private Network (LHCOPN) TWiki, https://twiki.cern.ch/twiki/bin/view/LHCOPN/WebHome
4. PerfSONAR Services, http://www.perfsonar.net/services.html
5. Gurun, S., Szymanski, B.: Automatic Internet Routing Behaviour Analysis Using Public WWW Traceroute Service. Technical report, Department of Computer Science, University of California, Santa Barbara (2008)
6. DANTE, http://www.dante.net/server/show/nav.13
7. WiN-Labor, http://www.win-labor.dfn.de/German/mainpage.html
8. RFC2119, http://www.ietf.org/rfc/rfc2119.txt
9. Holleczek, T.: Statistical Analysis of IP Performance Metrics in International Research and Educational Networks. Master's Thesis, Department of Computer Science, Friedrich-Alexander-University Erlangen-Nuremberg (2008)
10. Hades MA Service, https://wiki.man.poznan.pl/perfsonar-mdm/index.php/Hades_MA_Service
11. PerfSONAR E2E Link Monitoring: System Design and Documentation, https://wiki.man.poznan.pl/perfsonar-mdm/images/perfsonar-mdm/1/12/GN2-JRA4-06-010v240.pdf
12. Drools, http://www.jboss.org/drools/
13. Nagios, http://www.nagios.org/
14. NPM Alarms Service, http://www.npm-alarms.org/
15. Distributed European Infrastructure for Supercomputing Applications, http://www.deisa.eu/