Architecture of a Network Monitoring Element

Augusto Ciuffoletti
CNAF-INFN, Bologna, Italy
augusto@di.unipi.it

Michalis Polychronakis
FORTH-ICS, Heraklio, Greece
mikepo@ics.forth.gr

Abstract

The scalability of a network monitoring system, a vital component of a Grid, is challenging. We propose a solution based on demand driven monitoring sessions that use passive network monitoring techniques on a domain oriented overlay network. Related aspects of security and group membership maintenance are also considered.

1 Introduction

Monitoring the network infrastructure of a Grid plays a vital role in the management and the utilization of the Grid itself. According to the Global Grid Forum schema [2], the management of network infrastructure observations is divided into three distinct activities: their *production*, using network monitoring tools, their *publication*, by way of powerful databases, and their *utilization* by administration and workflow analysis tools. In this paper, we focus on the production and publication activities.

We explore the usage of passive network monitoring techniques for evaluating the quality of the Grid infrastructure. One of our main concerns with publication is scalability, given the induced non-linear trend of computational complexity when the system increases in size. We use a domain-oriented overlay network which reduces the complexity of the task, thus improving the scalability of our architecture.

2 Issues of scalability and security

We observe that, to optimize distributed applications, we need to collect end-to-end network performance observations, which cannot be derived from observations of single links (as discussed in [1]). Since each pair of Grid Services should be individually monitored and results should be reported, this makes an $O(n^2)$ computational complexity for many aspects of network monitoring, from the size of the database containing the monitoring results, to the number of pings that probe the system. We combine a number of concepts in order to control the complexity of our solution.

Hierarchical overlay network. The *topology* of a Grid is always split into groups of resources (the *domains*) reachable through a set of links. The accessibility of resources within a group greatly depends on the connectivity of such access links. It is therefore reasonable to monitor groups of resources that have access links in common, instead of single Grid resources. From the experience with GlueDomains [1] we take the definition of an agent that handles the monitoring activity for a domain: its role corresponds to a new resource in the Grid architecture, that we call a *Network Monitoring Element* (NME). The network monitoring activity is organized into *sessions* associated to a NME.

Passive network monitoring. Such technique, which consists of deriving network characteristics by observing traffic generated by applications, fits our scalability requirements since it is *non-intrusive* by nature. Its implementation can be cleanly modularized: the critical component is a daemon which captures and processes network packets. Plugin-like components extract required measurements from data streams produced by the packet analyzer.

Application driven sessions. The above concepts greatly reduce monitoring overhead, yet cannot provide the exhaustive picture of network connectivity needed to optimize distributed applications. To this purpose we introduce *on demand* sessions, initiated and controlled by applications.

All the above concepts introduce security concerns, and require Network Elements to share pieces of knowledge. In order to give a comprehensive solution for network monitoring, such subproblems are addressed with scalable solutions.

3 The Network Monitoring Element

In Fig. 1 we illustrate the structure of a *Network Monitoring Element*. The upper layer is in charge of implementing the interfaces to the outside through a *Network Monitoring Service* (NMS). The NMS offers distinct interfaces for user applications (including resource brokers), the *Grid*

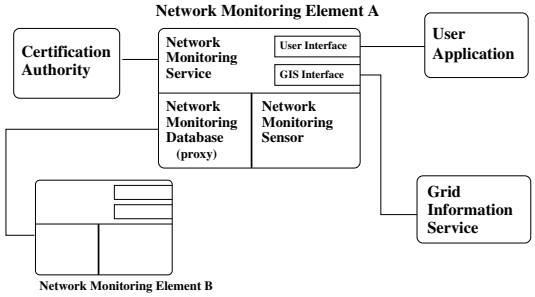

Figure 1. Interfaces between the NM Element and other Grid components.

Information Service (GIS), and for interaction with the Certification Authority.

The lower layer is composed of two distinct modules that do not interact with each other. The *Network Monitoring Sensor* supports monitoring sessions. We distinguish between *preconfigured* and *on demand* sessions: the former are configured directly by the NMS while the latter are configured by an outside user application through the NMS. Using different passive monitoring modules, built on top of MAPI [4], the NM Sensor can derive several network metrics, including per-application traffic throughput, packet loss ratio, and round-trip time.

The *Network Monitoring Database* (NMDB) describes the domain partition: for each element, corresponding to a Grid resource, it records the associated domain and other relevant attributes. Security issues impose the existence of a centralized certification authority: to make scalable the distribution of the certificates to the Network Elements, such certificates are stored in the NMDB.

4 Outline of the NMDB architecture

The implementation of this component is vital to the scalability of the monitoring architecture: from one point of view, it should work *as if* it where centralized (e.g., the certificate of a generic element should be downloaded only once), while from another, it should work *as if* it were local (e.g., the NMS should find locally the Domain containing a given Storage).

Since queries have poor locality, we cannot afford to keep only the *relevant* part of the database available. Hence, we propose that each NM Element holds an (almost) complete replica of the whole database.

Therefore we opt for the replication of the Grid database on each NM Element.

Since we consider that the frequency of updates should be quite low, corresponding to events creation or removal of Grid Elements, we traded off the number of update operations processed per time unit, in favor of a light footprint on traffic, resilience to failures, and predictability.

According with this conclusion, broadcasting of updates is performed using a number of circulating tokens, each containing a stack of recently issued updates. The number of tokens circulating in the system is tuned automatically, based on a feedback mechanism that enables each NMDB Proxy to inject or remove a token when needed.

The P2P protocol used for token circulation, which is secured using the same certificates that are stored in the database, is resilient to network and host failures, since lost tokens are regenerated automatically.

Although mostly based on randomized decisions, the protocol exhibits an excellent stability and predictability [3]: the load is evenly distributed in time and space, while the update latency remains constant.

5 Conclusions

We have outlined the architecture of a network monitoring service, also addressing related security and scalability issues. The basic building block of our architecture is the *Network Monitoring Element*, which monitors the network infrastructure between groups of resources sharing similar connectivity with the rest of the system. The monitoring activity is performed using passive monitoring, virtually without network overhead.

An exhaustive description of the architecture is in [3].

References

[1] S. Andreozzi, A. Ciuffoletti, A. Ghiselli, and C. Vistoli. Monitoring the connectivity of a grid. In *2nd Workshop on Middleware for Grid Computing*, pages 47–51, Toronto, Canada 2004.

[2] R. Aydt, D. Gunter, W. Smith, M. Swany, V. Taylor, B. Tierney, and R. Wolski. A grid monitoring architecture. Recommendation GWD-I (Rev. 16, jan. 2002), Global Grid Forum, 2000.

[3] A. Ciuffoletti and M. Polychronakis. Architecture of a network monitoring element. Technical Report TR-0033, CoreGRID Project, February 2006.

[4] P. Trimintzios, M. Polychronakis, A. Papadogiannakis, M. Foukarakis, E. P. Markatos, and A. Øslebø. DiMAPI: An application programming interface for distributed network monitoring. In *Proceedings of the 10th IEEE/IFIP Network Operations and Management Symposium (NOMS)*, April 2006.