A Monitoring Architecture for High-Speed Networks in Large Scale Distributed Collaborations

Alexandru Costan[1], Ciprian Dobre[1], Valentin Cristea[1], Ramiro Voicu[1,2]
[1]*University "Politehnica" of Bucharest, Romania*
[2]*California Institute of Technology, USA*
E-mails: {alexc, cipsm, valentin}@cs.pub.ro, ramiro.voicu@cern.ch

Abstract

In this paper we present the architecture of a distributed framework that allows real-time accurate monitoring of large scale high-speed networks. An important component of a large-scale distributed collaboration is the complex network infrastructure on which it relies. For monitoring and controlling the networking resources an adequate instrument should offer the possibility to collect and store the relevant monitoring information, presenting significant perspectives and synthetic views of how the large distributed system performs. We therefore developed within the MonALISA monitoring framework a system able to collect, store, process and interpret the large volume of status information related to the US LHCNet research network. The system uses flexible mechanisms for data representation, providing access optimization and decision support, being able to present real-time and long-time history information through global or specific views and to take further automated control actions based on them.

Keywords: *Grid Computing, Research Networks, Distributed Systems, Grid Monitoring*

1. Introduction

Network monitoring and measurement is commonly viewed as an essential function for evaluating, managing and improving the performance and security of high speed networks in large scale distributed systems. Current monitoring techniques and architectures often force administrators or Virtual Organization responsible to trade-off functionality for interoperability. Traditional passive network monitoring approaches are not adequate for fine-grained performance measurements nor for security applications in data intensive distributed environments like Grids, where requirements for fast data transfers are increasing.

Hence the manifest need for a system able to distribute the network monitoring function among each of the nodes of a multiple network large scale system, such that monitor software resident in each node is responsible for providing status information about that node and its communications links. Nevertheless, several emerging applications may use the monitoring data gathered at multiple locations spread across the network. For instance, Grid management applications (furthermore, the economic model based) could use traffic characteristics, network resources accounting and performance evaluation that can be computed only by combining monitoring data from multiple nodes of a distributed network. However, a distributed monitoring infrastructure can be extended outside the border of a single organization and span multiple administrative domains across the Internet, as it is often the case of large scale collaborations. The installation of several geographically distributed network monitoring sensors provides a broader view of the network in which large-scale events could become apparent. Finally, recent research efforts have demonstrated that a large-scale monitoring infrastructure of distributed cooperative monitors can be used for building security applications able to detect distributed DoS attacks, worms, malicious traffic and security breaches.

Furthermore, the wide dissemination of a cooperative passive monitoring infrastructure across many geographically distributed and heterogeneous sensors necessitates a uniform access platform, which provides a common interface for applications to interact with the monitoring services.

Therefore we designed and implemented a distributed monitoring architecture within the MonALISA [1] framework for the USLHCNet infrastructure used in HEP data intensive collaborations. The main contributions of our work aim at developing a scalable set of loosely coupled self-describing dynamic services able to collect and represent monitoring information, perform global optimization tasks using mobile agents and facilitate the programming, access and coordination of several distributed monitoring sensors in a flexible and efficient way.

The remainder of this paper is organized as follows. Section 2 describes the USLHCNet high speed network infrastructure. In Section 3 we present MonALISA, the framework within which we developed our monitoring system. The architectural components, their interactions and the communication infrastructure are further detailed in Section 4. Section 5 outlines some important results obtained using our system. Section 6 summarizes related work in this field while Section 7 presents future research and concludes the paper.

2. USLHCNet

The USLHCNet is a high performance transatlantic network infrastructure developed to meet the HEP community's needs. This facility is developed as needed to address HEP's rapidly advancing requirements, while taking advantage of the equally-rapid evolution of (and occasional revolutions in) network technologies, in order to provide the most cost-effective solutions with adequate performance, year-by-year.

USLHCNet today consists of a set of 10 Gbps links interconnecting CERN, MANLAN in New York (the MANLAN exchange point is designed to facilitate peering among US and international research and education networks in New-York), Starlight in Chicago (StarLight is an international peering point for research and education networks in Chicago) and SARA in Amsterdam - a strategic place where interconnections to other transatlantic networks (IRNC, Gloriad, Surfnet, etc.) are simple and relatively inexpensive. SARA is also the location of the Dutch Tier1 and this provides the option for fast Tier1-to-Tier1 connectivity, in compliance with the LHC Optical Private Network (OPN) policies. The network has been architected to ensure efficient and reliable use of the 10 Gbps bandwidth of each link, up to relatively high occupancy levels, to cover a wide variety of network tasks, including: large file transfers, grid applications, data analysis sessions involving client-server software as well as simple remote login, network and grid R&D-related traffic, videoconferencing – VRVS/EVO and general Internet connectivity.

The USLHCNet backbone today includes three distinct OC-192 transatlantic circuits on three separate transatlantic cables (Geneva-New York, Geneva-Chicago and Amsterdam-Chicago) and another three continental OC-192 circuits (two New York-Chicago circuits and one Amsterdam-Geneva circuit). The network design follows the hierarchical model for LHC Computing developed at Caltech and provides connectivity and backup between CERN (the data generating point or Tier0 site) and the two US Tier1s - BNL for Atlas and FNAL for CMS.

The current topology of the network presented in Figure 1 shows both the US LHCNet Points of Presence and the close relation between US LHCNet and the NSF-funded network R&D project UltraLight - a research project led by Caltech whose purpose is to provide the network advances required to enable petabyte-scale analysis of globally distributed data, and to overcome the current vision of the network as a passive resource. One major advantage of this connectivity is that it provides the Tier2 centers and universities participating in UltraLight fast access to the Tier1 centers in US LHCNet and even to CERN if this should be required. Arrangements to provide connections between the US Tier2 centers and the European Tier1s over US LHCNet and GEANT2, which is required in the CMS Computing Model for example, are under discussion.

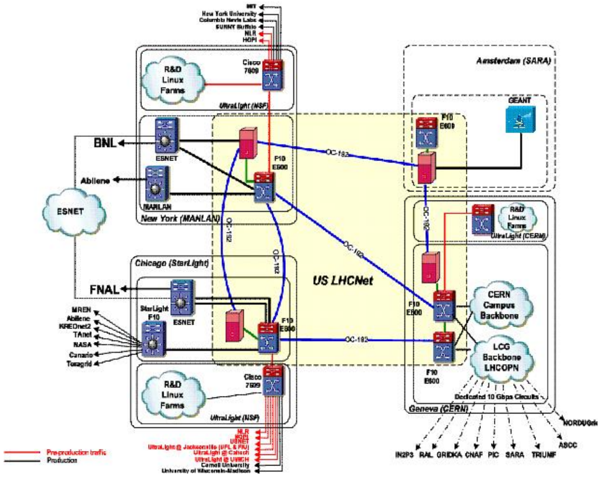

Figure 1. The USLHCNet network topology.

The US LHCNet backbone is architected and operated to guarantee 24x7x365 network availability and full performance, supporting both large-scale data transfers and real-time traffic therefore one of the main concern was monitoring. The USLHCNet network operational center relies on our architecture for extensive monitoring of the links, the equipment and the end hosts. For that purpose we developed a system within the MonALISA framework able to meet the high and strict requirements of this infrastructure and to cope with vast amounts of transferred data (ex: 500 TB of data sent from CERN to FNAL in two months) and high speeds.

3. MonALISA framework

MonALISA is a distributed monitoring system, relying on JINI and WebServices technologies, able to provide complete monitoring, control and global optimization services for complex systems. An agent-based architecture provides the ability to invest the system with increasing degrees of intelligence, to reduce complexity and make global systems manageable in real time. The scalability of the system derives from the use of a multi threaded engine to host a variety of loosely coupled self-describing dynamic services, the ability of each service to register itself and then to be discovered and used by any other services, or clients that require such information.

The MonALISA framework is currently being used to monitor many grid sites in the U.S., Europe, Asia and Australia. It provides complete information, in near real-time, about resource utilization and job execution. Currently there are more than 370 sites running MonALISA. Overall, the system is collecting approximately 1.000.000 parameters in near real-time with a rate of 32.000 updated parameters per minute, collecting data related to approximately 70.000 nodes and thousands of jobs running concurrently in Grid systems. Some major Grid communities rely on MonALISA, among which are OSG, CMS, ALICE, D0, STAR, VRVS, LCG Russia, SE Europe GRID, APAC Grid, UNAM Grid, ABILENE, ULTRALIGHT, GLORIAD, LHC Net, and RoEduNet. We extended this framework to provide information about network traffic in major networks and connectivity between different grid sites.

4. USLHCNet Monitoring Architecture

In order to achieve complex monitoring of high-speed network, large-scale network events and characteristics, cooperation of many, and possibly heterogeneous, monitoring data collectors distributed over a wide-area network must be accomplished. In such an environment, the processing and correlation of the data gathered at each collector gives a broader perspective of the state of the monitored network, in which related events become easier to identify.

Therefore, within the MonALISA framework, we developed a distributed monitoring architecture for the USLHCNet network, able to collect relevant information and to further present global views from the dynamic set of services running in the distributed environment to higher level services. Our architectural model consists of Sensors – MonALISA Services executing monitoring tasks, which are able to interact autonomously with other services through dynamic Proxies or via Agents that use self-describing protocols; these Services are able to register themselves in some replicated and distributed Lookup Services, along with describing attributes; Clients can dynamically discover these Services by their attributes, interact with them and use the monitored data for further processing, representation or automated decision control. Figure 2 presents these main components of the architectural model and their communication infrastructure.

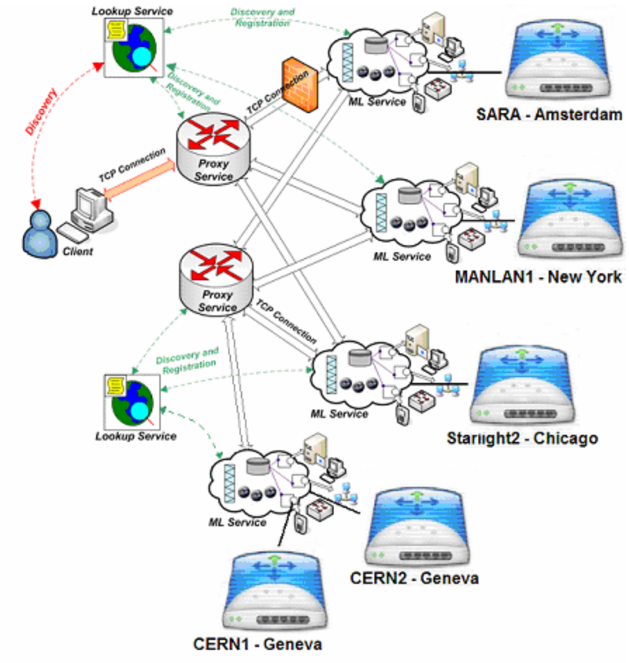

Figure 2. Communication infrastructure of monitoring services for the USLHCNet network.

We outline the functionalities and the architectural details of these components in the following sub-sections.

4.1 Monitoring Services

Following the existing topology configuration exhibited in Figure 1, we deployed MonALISA Service Farms to monitor routers and end hosts at each major interconnecting site:

- CERN in Geneva, the Tier0 center where the HEP data is first produced. Our Service is monitoring the end host, the CIENA CD/CI and two routers installed for security / redundancy reasons collecting information about the connections to US and Amsterdam and also the peers to NORDUGrid, Gridka, Triumf, etc.
- MANLAN1 in New-York providing connectivity and backup with US Tier1 – BNL for Atlas. We also monitor the end host, the CIENA CD/CI and peers to MIT, NYU, Columbia Nevis, NLR, Sunny Buffalo.
- Starlight2 in Chicago, US Tier1 – FNAL for CMS. Peering with NLR, HOPI, USNET, UltraLight, Cornell University, University of Wisconsin are monitored along with the end host and the installed CIENA CD/CI.
- SARA in Amsterdam, using GEANT2 infrastructure and providing fast Tier1-to-Tier1 connectivity to

Dutch Tier1; the Services tracks the links to New-York and CERN and the CIENA CD/CI.

The MonALISA Service monitors and tracks site computing farms and network links, routers and switches using SNMP, and it dynamically loads modules that make it capable of interfacing existing monitoring applications and tools (e.g. Ganglia, MRTG, LSF, PBS, Hawkeye.). The core of the monitoring service is based on a multithreaded system used to perform the many data collection tasks in parallel, independently.

The MonALISA services deployed in USLHCNet create a local database for short or long term history for the monitored information. The monitoring is done by a multi threaded engine for parallel and independent data collection tasks execution as pictured in Figure 3. Failing monitoring modules are automatically removed from execution queue so they don't affect the rest of the system. Clients can send two types of requests (predicates) to the services: history requests, that are served from the local database, and subscriptions for new data events. A special type of client, used by repositories or other services, can be used to store in a central place, selected monitoring information from many sites. It can be used to store long term history with high resolution. Both the services and the repositories share the same flexible mechanism to use database systems. From the configuration files the administrator can choose the time interval for which the data is stored and the data representation that is best for the site thus allowing both coarse-grained and fine-grained accurate network monitoring. This structure allows the storage engine to automatically select the best data source in terms of resolution and database interrogation speed.

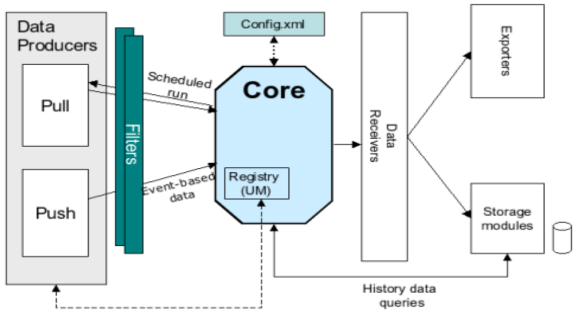

Figure 3. Data storage and handling in the monitoring Service.

The monitoring system has the ability to transport and store different types of monitoring data. This data must be user-definable so that any Service user (ex: network administrators) can implement its own data type; thus users can monitor proprietary parameters; self defined metrics, which are specific to their domain network, cluster or Virtual Organization connected through USLHCNet. This data also has to be self-describing so that the database engine can store it in transparent manner. The data producing modules can attach extra information to the parameters they produce by using the Registry object to map parameters to Unit instances (allowing specific new units, comments, correlations, events associated to monitored data) and hence enable efficient management and control of the network.

The modules used for collecting different sets of information, or interfacing with other monitoring tools, are dynamically loaded and executed in independent threads. A Monitoring Module is a dynamic loadable unit which executes a procedure (or runs a script / program or performs SNMP request) to collect a set of parameters (monitored values) by properly parsing the output of the procedure. In general a monitoring module is a simple class, which is using a certain procedure to obtain a set of parameters and report them in a simple, standard format. Monitoring Modules can be used for pulling data and in this case it is necessary to execute them with a predefined frequency (i.e. a pull module which queries a web-service) or to "install" (has to run only once) pushing scripts (programs) which are sending the monitoring results (via SNMP, UDP or TCP/IP) periodically back to the Monitoring Service. Allowing to dynamically load these modules from a (few) centralized sites when they are needed makes much easier to keep large monitoring systems updated and to provide new functionalities dynamically; users can also implement easily any new dedicated modules and use it in the MonALISA framework.

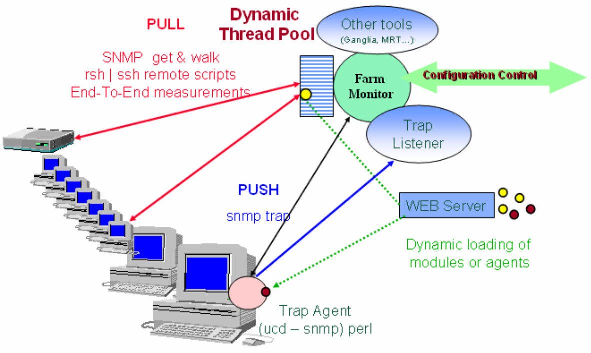

Figure 4. USLHCNet Monitoring Service information gathering model.

This architectural model of the Service, presented in Figure 4, makes it relatively easy to monitor a large number of heterogeneous nodes with different response times, and at the same time to handle monitored units which are down or not responding, without affecting the other measurements.

The USLHCNet Monitoring Services track information about WAN IN/OUT real-time and history traffic on each node's peering links, WAN status (in/out errors, admin errors, interfaces status, operational status, speed), capacity of the links, CIENA ETTP traffic and alarms, Optical Switches controlled, SFlow (throughput in/out, packet analysis of transferred data by applications) and also service node's status (Load, CPU usage, Memory, IO parameters etc.).

4.2. LookUp Services

The deployed MonALISA services are able to discover each other in the distributed environment and to be discovered by the interested clients. Each MonALISA service registers itself with a set of Lookup Services (LUSs) as part of one or more groups and it publishes some attributes that describe it. In this way any interested application can request MonALISA services based on a set of matching attributes. The registration uses a lease mechanism. If a service fails to renew its lease, it is removed from the LUSs and a notification is sent to all the services or other application that subscribed for such events. Remote event notification is used in this way to get a real overview of this dynamic system. Lookup services have replicated information. It is important for the monitoring service to be registered in two or more distributed lookup services, because if one fails responding, interested clients can find the MonALISA services registered in the other online lookup services. Thus, the single point of failure problem can be avoided and a more reliable network for registration of services can be achieved in the distributed environment. The JINI technology used allows dynamically adding and removing Lookup Services from the system. Clients can dynamically discover Services by their attributes using the LUSs.

4.3. Proxy Service

The interaction between clients and services is made available through transparent Proxy services. The Proxy services also publish themselves in lookup services. In this way mutual discovery is used to connect monitoring services with Proxies and clients with the nearest and less used Proxy service. It is also worth to emphasize that a monitoring service running behind a firewall or NAT is loaded in the distributed system because it initiates connections to all the available proxies found in the lookup services. At the same time the Proxy service does an "intelligent" multiplexing of subscribed data for multiple clients and can forward multicast messages sent by endpoints. For redundancy, scalability and dynamic load-balancing of clients, it is important to have two or more proxy services in the system.

4.4 Clients

The network monitoring information from USLHCNet can be accessed by various types of clients which we describe in the following.

A. Interactive Clients

The Interactive Clients deliver to users an intuitive graphical and global interface of the states of the distributed monitored network. After discovering all available monitoring services using the MonALISA Registration and Discovery mechanism the Interactive Client requests history and real-time data from the services the user is interested in (namely, relevant monitored parameters – ex: WAN traffic, traffic types, network topology, etc.), and dynamically updates its interface as the state of the system changes and real-time monitoring data come.

The graphical interface used for monitoring high speed networks like USLHCNet is composed of several dynamically loadable panels that statistically show the monitoring sites, routers, and links in the global system. The user can also influence the system, if it has rights, using the administration interfaces. We developed relevant dashboards for the monitored network:

- The 3D Map Panel locates the deployed MonALISA Services on a 3D view of the world geographical map showing the monitoring WAN links, real-time traffic on them, the capacity of the links, the optical switches controlled and other end host parameters like sites' Load, CPU usage, IO parameters, etc.
- The Groups Panel presents an USLHCNet community view with all its monitoring sites. The user can plot common parameters from the selected services in the monitored community. The plot shows comparatively history or real-time parameter values from selected sites.
- The OS GMap Panel shows the state of the monitored optical switches, the machines connected to them and the created optical paths. It uses different layouts for displaying: Random, Grid, Radial, Layered, Map, Elastic. If authorized with the right credentials, the user can administer the network of optical switches, from the panel administration interface, creating or releasing optical paths.
- The WAN Panel is dedicated to WAN traffic plotting in a single window all WAN traffic through each monitored WAN link by MonALISA services. Shows the inbound and outbound WAN traffic for each link. The Load Panel globally shows the load distribution on every monitored end host, the number of nodes

that are not loaded (have load lower than 0.5), the number of nodes that are medium loaded (have load between 0.5 and 1) and the number of nodes that are loaded (have load more that 1).

B. Repositories

The USLHCNet Repositories are special types of clients used for long periods storage and further processing of monitoring data. They subscribe to a set of parameters or filter agents to receive selected information from all the Services. This offers the possibility to present global views from the dynamic set of services running in the distributed network environment to higher level services. The received values are further stored locally into a relational database, optimized for space and time. The collected monitoring information is further used to present a synthetic view of how the USLHCNet high speed network performs. The system targets developing the required higher level services and components of the network management system that provide decision support, and eventually some degree of automated decisions and control.

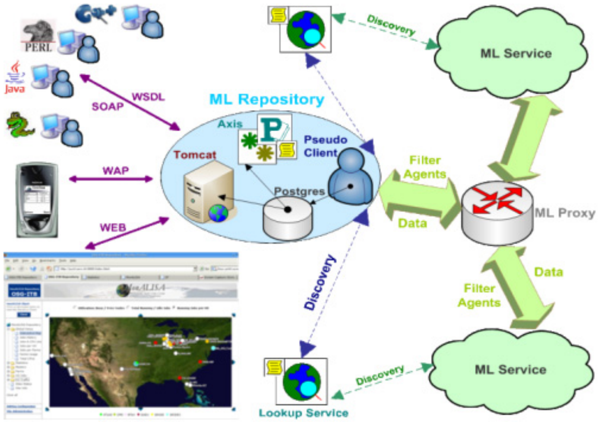

Figure 5. USLHCNet Repository Architecture and Communication Infrastructure

The USLHCNet Repository registers with a set of predicates and stores the received values in the local database. A predicate has the following pattern: Service / Monitoring_Target / Node / start_time / end_time / function_list, where Service is the deployed sensor on site (ex: CERN1, AMS), Monitoring_Target is the generic targeted monitored level (ex: WAN, WAN_Stats, sFlow, CIENA), Node is the data producer (ex: a router, stating its IP address), start/end_time indicate the time frame for the desired data, function_list is the actual list of needed monitoring parameters. The predicates may also be specified through regular expressions. These parameters (functions) are then dynamically plotted into a large variety of graphical charts, statistics tables, and interactive map views, following the configuration files describing the needed views, and thus offering customized global or specific perspectives. The same mechanism is used to offer access to this information from mobile phones using the Wireless Access Protocol (WAP). The WSDL/SOAP interface is also available so that clients can access information received from several monitoring Services.

The main components of the Repository system, presented in Figure 5, are the storage client, responsible for data collection and storage, and the servlet engine which ensures the translation of user customized requests from the interface into appropriate queries for the storage client, according to the predicate pattern presented above; furthermore, it plots the results in a flexible manner, according to properties set in configuration files.

We enhanced the USLHCNet Repository System with fault tolerance capabilities in order to achieve high availability. Hence, we used replication of the Repository Service aiming for a warm standby configuration: in case one repository fails, one or more replicas are ready to take over clients' queries in a transparent way. In this respect we deployed three instances of the repository, located at distinct locations (the main USLHCNet Repository [2] and replicas at CERN [3] and RoGrid at UPB) so that local network failures wouldn't affect the system. Deployed repository replicas are permanently aware of each other's current state and after recovery from failure an instance synchronizes its state with the other running replicas ensuring consistency of monitored data.

C. Web Services

Web Services are other special types of clients integrated with the Service as well as with the Repository and provides an interface for publishing the monitoring data using a WSDL/SOAP technology. Hence, the collected raw or processed information is made available to any client written in C/C++, Java, Perl, Python, etc.

5. Monitoring USLHCNet using the distributed architecture

We have been monitoring USLHCNet using our architecture for more than one year, tracking 2000 parameters from 7 deployed MonALISA services on 150 nodes. The repositories has been serving ~1,000,000 requests at an average rate of 120 requests/hour with peaks of 1500 requests/hour. The average collection rate is ~2400 results/minute.

In order to meet the requirements of a non intrusive and reliable monitoring we adopted the strategy of keeping monitoring data locally for a reasonable time frame (~weeks) and for longer periods of time (~years) in the repository. To meet the fail over requirements we are monitoring the WAN links on both ends in different MonALISA services, located very close to the edge routers, or the other network equipments (e.g. Ciena CD/CI, Optical switches like Glimmerglass and Calient). The ability of the framework to keep the monitoring data for long periods of time will help the HEP community to understand the network related bottlenecks, if any. As depicted in Figure 6 (generated by the MonALISA repository), the aggregated monitored traffic on the USLHCNet network in the last year was about 7 Pbytes.

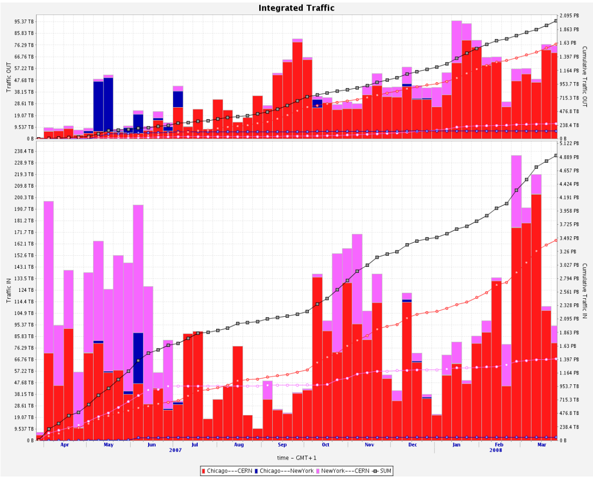

Figure 6. Integrated traffic IN/OUT for last year.

This traffic is expected to grow once the luminosity inside the LHC accelerator will grow. Using the last year's simulations and traffic tests, our monitoring architecture proved that capability of the USLHCNet to sustain the data traffic that will be generated by LHC physics analysis, once the experiments started.

6. Related work

The network monitoring in large scale distributed systems has been addressed by a number of authors: solutions are differentiated in the way they cope with scalability issues, features implemented or architectural models. We present in the following the most relevant to our research.

The Network Weather Service (NWS) [4] is a distributed system that periodically monitors and dynamically forecasts the performance various network and computational resources can deliver over a given time interval. The service operates a distributed set of performance sensors (network monitors, CPU monitors, etc.) from which it gathers readings of the instantaneous conditions. While the design of NWS implementations focuses on providing the functionality necessary to investigate the effectiveness of dynamic scheduling in local medium and widearea computational settings it only monitors few metrics (rtt, throughput and latency). These implementations do not scale well however and lack the robustness necessary to make the NWS a reliable system service.

TopoMon [5] can be regarded as an evolution of NWS which mainly measures bandwidth and latency of end-to-end network paths. This information is necessary but not sufficient for applications with communication patterns where multiple sites compete for the same links. TopoMon extends NWS with tools and support for managing link level topology, which is relevant (for instance, in view of a reservation service that cannot ignore the existence of shared links when allocating end to end communication resources), but doesn't address issues like flexible monitored parameters or scalability.

Another network monitoring architecture – GlueDomains [6] partitions the Grid into Domains as a way to limit network monitoring overhead. This is obtained by assuming that certain observations are representative of connectivity between subsystems, not merely between path endpoints. GlueDomains architecture centers around a number of specialized units hosting the agents in charge of monitoring the network. Such agents are able to autonomously (re)configure their activity based on a dynamic description of the network monitoring topology, available from a relational database. The monitoring activity based on domain partitioning of Grid resources has as main target the Network Element, which abstracts the network infrastructure in charge of interconnecting two domains. A Grid-wide deployment of GlueDomains was carried out as part of the Italian branch of the Large Hadron Collider Computing Grid Project (LCG). Apart from the statistics collected (usual packet loss and roundtrip time, together with an experimental one way jitter measurement tool, published through the GridICE Grid Information Service [7]), the most relevant results from the GlueDomains experiment concern the ease of deployment, as well as the resilience, and stability of the architecture. However, the existence of a centralized repository for configuration data, together with an extended use of active monitoring techniques, limits the scalability of the GlueDomains prototype to approximately 50 domains, which is reasonable only for a small-scale grid.

As seen from above, existing techniques either present a restricted view of network behavior and state, or do not efficiently scale to higher network speeds and heavier monitoring workloads. Our proposed monitored architecture allows more accurate parameter collection (through fine tuning of the deployed Service), proves to be more flexible (as it can be extended by users to meet their specific needs through the pluggable components and coarse grained monitoring, has no single point of failure, the and the agents system used ensures scalability thus making it suitable in large scale distributed environments.

7. Conclusions and Future Work

We have presented a distributed network monitoring architecture used for the USLHCNet infrastructure. Among the strengths of our system are reliability, flexibility, scalability and the ease of extension. Besides typical accounting mechanisms, Grid management and network security solutions rely on efficient and optimized monitoring which may be achieved through the use of this architecture.

We plan to use this system as a vehicle for our future research in performance monitoring. Reliability is one of our main concerns. The LHC network availability requirements are very strict: the target is 99.95% uptime, which corresponds to less than an hour of unscheduled downtime per year. Therefore we plan to further develop the high availability of our system by improving redundant monitoring at both ends of major links in order to cope with network failures which can occur at a node; this allows us to guarantee the continuity of service.

We intend to enhance the framework both by adopting new relevant panels for the interactive client and by extending the optical switches monitoring service. In addition we are exploring new flow accounting, aggregation and representation techniques appropriate for large scale distributed computing environments.

system – the loadable modules), offers both fine grained

8. References

[1] I. C. Legrand, H. Newman, R. Voicu, C. Cirstoiu, C. Grigoras, M. Toarta, C. Dobre, “MonALISA: An Agent based, Dynamic Service System to Monitor, Control and Optimize Grid based Applications”, *Proc. of Computing in High Energy and Nuclear Physics CHEP04*, Interlaken, Switzerland, 2004.

[2] USLHCNet Monitoring Repository, http://hermes5.uslhcnet.org:8080, accessed in March 2008

[3] USLHCNet CERN Replica Repository, http://monalisa4.cern.ch:8888, accessed in March 2008.

[4] R Wolski, N Spring, J Hayes, “The Network Weather Service - A Distributed Resource Performance Forecasting Service for Metacomputing”, *Journal of Future Generation Computing Systems*, 1999.

[5] M. den Burger, T. Kielmann, H. E. Bal, “TOPOMON: A monitoring tool for grid network topology”, *International Conference on Computational Science (2)*, pages 558–567, 2002.

[6] S. Andreozzi, A. Ciffoletti, A. Ghiselli, C. Vistoli, “GlueDomains: Organization and accessibility of network monitoring data in a grid”, *Technical Report TR-05-15, Universita di Pisa*, Italy, May 2005.

[7] C. Aiftimiei, S. Andreozzi, G. Cuscela, N. D. Bortoli, G. Donvito, S. Fantinel, E. Fattibene, G. Misurelli, A. Pierro, G. Rubini, G. Tortone, “GridICE: Requirements, architecture and experience of a monitoring tool for grid systems”, *Proc. of the International Conference on Computing in High Energy and Nuclear Physics CHEP06*, Mumbai, India, February 2006.