EFFICIENT NETWORK MONITORING FOR LARGE DATA ACQUISITION SYSTEMS

D.O. Savu, B. Martin, CERN, Geneva, Switzerland
A. Al-Shabibi, Heidelberg University, Heidelberg, Germany
R. Sjoen, University of Oslo, Norway
S.M. Batraneanu, S.N. Stancu, UCI, Irvine, California, USA

Abstract

Though constantly evolving and improving, the available network monitoring solutions have limitations when applied to the infrastructure of a high speed real-time data acquisition (DAQ) system. DAQ networks are particular computer networks where experts have to pay attention to both individual subsections as well as system wide traffic flows while monitoring the network. The ATLAS Network at the Large Hadron Collider (LHC) has more than 200 switches interconnecting 3500 hosts and totaling 8500 high speed links. The use of heterogeneous tools for monitoring various infrastructure parameters, in order to assure optimal DAQ system performance, proved to be a tedious and time consuming task for experts. To alleviate this problem we used our networking and DAQ expertise to build a flexible and scalable monitoring system providing an intuitive user interface with the same look and feel irrespective of the data provider that is used. Our system uses custom developed components for critical performance monitoring and seamlessly integrates complementary data from auxiliary tools, such as NAGIOS, information services or custom databases. A number of techniques (e.g. normalization, aggregation and data caching) were used in order to improve the user interface response time. The end result is a unified monitoring interface, for fast and uniform access to system statistics, which significantly reduced the time spent by experts for ad-hoc and post-mortem analysis.

INTRODUCTION

At the core of the ATLAS DAQ infrastructure there are 3 distinct computer networks responsible for the data transfers between the system's subcomponents. More than 200 switches and routers interconnect around 3500 hosts to build a real-time filtering system for particle collision events.

The ATLAS Networking Team's operational goals are to prevent network downtime and to be able to track down ad-hoc or post-mortem network issues as fast as possible. A complex software solution has been developed to help networking experts accomplish their goal while providing relevant and up-to-the-minute system information for other related DAQ teams.

The network monitoring software is designed as a modular solution around a central database (N-CDB). The N-CDB acts as a shared data structure for the various modules and is implemented as a core relational database with external binary file extensions. The most important software modules (Figure 1) are:

- **Topology Consistency:** responsible for keeping an up-to-date database representation of network devices, computers, connections and geographical location;
- **Statistics Collection:** gathering any network traffic related statistics and making the data available to other modules through the N-CDB;
- **The ADAM API:** a programmatic way of accessing the database data by external programs or scripts;
- **The User Interface:** a set of web-based applications to make all the information available in a structured and easy to navigate way;
- **Self-check module:** a warning mechanism sending detailed mail messages to experts about any module not functioning as expected.

TOPOLOGY CONSISTENCY

An accurate representation of installed devices and network connections in the N-CBD is assumed by every module and is critical for the system's operation. To keep the topology representation up to date, a full network discovery process is run periodically and is complemented by additional data. The discovery is then deployed in the production system through a semi-automatic procedure involving expert approval for topology changes.

The network discovery is primarily MAC address based, being able to identify network device uplinks and end node connections by inspecting the MAC address tables of network devices. Since the MAC address tables are dynamically populated by the traffic traversing network devices, it is essential to run the discovery when all the devices have generated traffic. Thus it is most efficient to run a discovery either during data taking, when almost all devices have network activity, or immediately after an induced ARP* broadcast event.

A network discovery based on MAC address tables or LLDP[#] does not reveal the nodes' function or geographical location. To extend the discovery knowledge with such complementary data, information from external databases is used and cached by the N-CDB. The purpose of the N-CDB cache is to avoid any run-time dependency and overloading of external databases.

*Address Resolution Protocol
[#] Link Layer Discovery Protocol

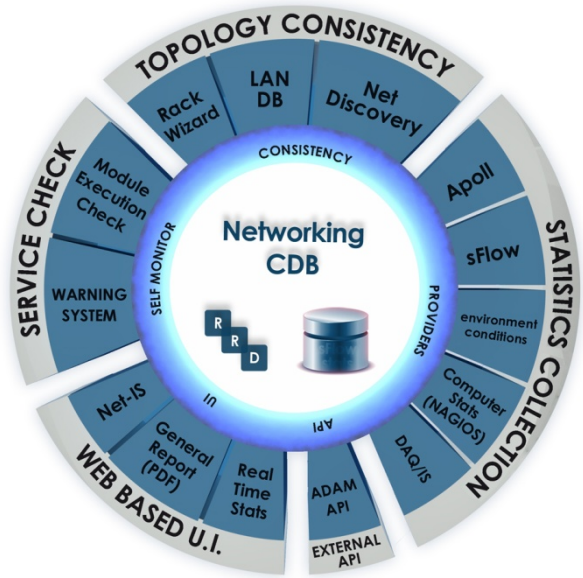

Figure 1: Diagram of the network monitoring software modules.

Such external databases include Atlas RackWizard, for geographical location, Sysadmin ConfDB, for functional host name mappings and IT LanDB, for auxiliary host mapping.

STATISTICS COLLECTION

Gathering statistics about traffic flows and network state is mandatory for monitoring the system's performance. The network overview statistics are the primary source of information for investigating most network problems. These statistics are collected by APoll, an internally developed software which polls SNMP counters from switches and routers every 30 seconds. Once a network issue is isolated, an in-depth analysis is performed by looking for relevant traffic samples collected via the sFlow protocol.

APoll, the high speed SNMP poller, has been developed by the networking team to address the limitations of commercial software used in production. It is implemented as a multi-threaded C++ application that dynamically adapts its number of threads to the number and responsiveness of the network devices. The run mode can be configured either as standalone, dumping basic traffic log files, or integrated mode when last minute statistics are synchronized with the N-CDB.

The SNMP* counter statistics, due to their time-series format, are stored in a structured set of Round Robin Database (RRD) binary files. The files are referenced by the metadata in N-CDB and accessed directly via the RRD library. A copy of the RRD files for the last 72 hours is also stored on a ramfs partition to reduce the I/O to non-volatile storage and improve application response. Additionally, the last SNMP counter values are stored in a N-CDB in-memory table. The table is used as a shared resource between all the applications that need or provide real-time data.

The sFlow sample data, on the other hand, does not resemble a standard time-series format. To efficiently store the sFlow gathered data, a tool is used to aggregate it over 1 minute time intervals and store it in a MySQL relational database. Then, various database tuning techniques are used to improve the response time.

Sometimes external factors, such as environmental conditions or faulty systems, affect the network's normal behaviour. Making information from directly related systems available to network monitoring modules gives the networking team an advantage in understanding and limiting the impact of an external event. Currently, full or partial information from NAGIOS (computer monitoring), PVSS (environmental conditions database) and DAQ/IS (data-taking information service) is accessible through the same user interface used for network monitoring. Examples of such external factors include rack power failure causing device crash events or DAQ data-taking process causing network discarded packets.

THE ADAM API

The information stored in the N-CDB is also of interest to shifters and system analysts. To facilitate programmatic access to information, a generic interface for data exchange has been implemented. The creation of a generic interface has been a joint effort between several ATLAS teams and lead to the definition of the ADAM (ATLAS Data API Mechanism) data exchange interface.

The ADAM API is fully implemented in the network

* Simple Network Management Protocol

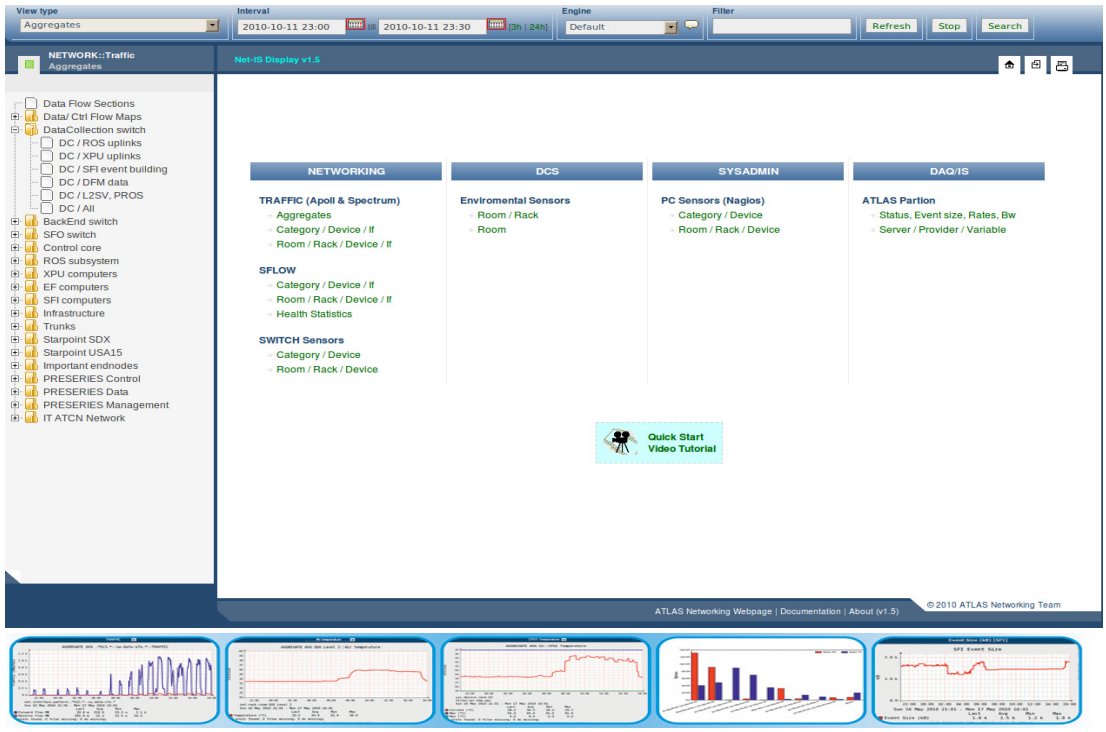

Figure 2: Main page of the Net-IS user interface providing access to historical statistics.

monitoring solution and provides network traffic, computer statistics and environmental conditions to external applications on demand. Through this API the network monitoring software can be a data provider for other external data analysis applications, and is currently used in the ADAM User Interface project.

THE USER INTERFACE

The user interface is designed to provide all the data an expert needs to investigate a problem and yet be simple to use for non experts. A set of web-based applications, integrated as a portal, offers access to real-time reports and historical statistics about network state and complementary systems. For offline visualization of the current topology a comprehensive PDF report is generated daily and made available for download.

The Net-IS web application (Figure 2) is the main networking interface used by experts and shifters. It provides direct access to any historical statistics through a single interface with the same look and feel regardless of the data source. The statistics can be grouped to match predefined or custom rules via an aggregation layer based on regular expressions.

The Net-RT web application provides real-time statistics about any network device or network traffic. Various table-based reports, as well as a 2D map displaying network uplinks load, are updated in real-time according to the N-CDB in-memory table.

A different option to access the N-CDB data is through a Command Line Interface (CLI). The CLI tool was developed to allow experts to access and change the N-CDB data manually without the risk of affecting its low level consistency.

The user interface has been described in more detail in the "Integrated System for Performance Monitoring of the ATLAS TDAQ Network" paper, published in the Proc. Computing in High Energy Physics 2010.

SELF-CHECK MECHANISM

The reliability of the system may be affected by a misbehaving module. The purpose of the self-check mechanism is to identify and inform experts in real-time about any faulty modules. To make the self-check mechanism more robust, the checking of module execution status and the mail reporting have been implemented as two distinct components.

The execution status check is implemented using a set of scripts that monitor each module's functionality and expected results, and then report the status to the N-CDB. If a check script fails to execute within a timeout period, it will flag this problem to the warning component. A module can also flag a warning and send a detailed message if it anticipates an imminent problem likely to require expert intervention.

The warning component then checks all the active module status information and sends error-triggered and daily service status mail reports.

CURRENT STATUS AND FUTURE PLANS

Started as an independent network monitoring software, the current ATLAS DAQ network monitoring solution has grown by integrating additional system information.

This extension improved the understanding of overall system behaviour by enabling the correlation between networking events and external factors. Two years after its deployment it is the main tool used by both experts and non experts to analyze network problems. It has proven to be more efficient compared to the alternate commercial solution when it comes to performance monitoring.

Future plans to improve the current solution include the extension of the N-CDB by storing network event logs from multiple sources and the addition of an event processing engine. Such a feature will bring together the network events signalling a state change, with the collected statistics that provide a more complete picture of the circumstances associated with the generated event.

REFERENCES

[1] DO. Savu, A. Al-Shabibi, B. Martin, R. Sjoen, SM. Batraneanu and S. Stancu, “Integrated System for Performance Monitoring of the ATLAS TDAQ Network”, in Proceedings of the CHEP 2010, Taipei, Taiwan, Oct. 2010.

[2] M. Ciobotaru, L. Leahu, B. Martin, C. Meirosu and S. Stancu, “Networks for ATLAS trigger and data acquisition”, in Proceedings of the CHEP 2006, Mumbai, India, Feb. 2006.

[3] S.M. Batraneanu, A. Al-Shabibi, M. Ciobotaru, M. Ivanovici, L. Leahu, B. Martin and S. Stancu, "Operational Model of the ATLAS TDAQ Network, Proc. IEEE Real Time 2007 Conference, Chicago, USA, May 2007.

[4] R. Sjoen, S. Stancu, M. Ciobotaru, S.M. Batraneanu, L. Leahu, B. Martin and A. Al-Shabibi, "Monitoring Individual Traffic Flows whithin the ATLAS TDAQ Network”, CHEP, Prague, Czech Republic, 21-27 Mar 2009.

[5] S. Kolos et al., “Online Monitoring software framework in the ATLAS experiment”, in Proceedings of the CHEP 2003, La Jolla, USA.

[6] W. Vandelli et al., “Strategies and Tools for ATLAS Online Monitoring”, in IEEE Transactions on Nuclear Science (TNS), June, 2007, volume 54, pp 609-615.

[7] SNMP, Simple Network Management Protocol [Online]. http://en.wikipedia.org/wiki/Simple_ Network_ Management_Protocol.

[8] WWW Technologies [Online]. Available: http://en.wikipedia.org/wiki/World_Wide_Web.

[9] Django, The Web framework for perfectionists with deadlines. [Online]. http://www.djangoproject.com

[10] RRDTool, OpenSource industry standard, high performance logging and graphing system. [Online]. http://oss.oetiker.ch/rrdtool/

[11] NAGIOS, The Industry Standard In IT Infrastructure Monitoring. [Online]. http://www.nagios.org

[12] LLDP, Link Layer Discovery Protocol. [Online]. http://en.wikipedia.org/wiki/Link_Layer_Discovery_ Protocol