Designing a New Infrastructure for ATLAS Online Web Services

Sergio Ballestrero[1], *Franco* Brasolin[2], *Arturo* Sanchez Pineda[3], *Diana Alessandra* Scannicchio[4,*], and *Matthew Shaun* Twomey[5]

[1]University of Johannesburg, South Africa
[2]Istituto Nazionale di Fisica Nucleare Sezione di Bologna, Italy
[3]INFN Gruppo Collegato di Udine and Università di Udine, Italy
[4]University of California Irvine, United States of America
[5]University of Washington, United States of America

Abstract. Within the ATLAS detector, the Trigger and Data Acquisition system is responsible for the online processing of data streamed from the detector during collisions at the Large Hadron Collider (LHC) at CERN. The online farm is composed of ~4000 servers processing the data read out from ~100 million detector channels through multiple trigger levels. The capability to monitor the ongoing data taking and all the involved applications is essential to debug and intervene promptly to ensure efficient data taking. The base of the current web service architecture was designed a few years ago, at the beginning of the ATLAS operation (Run 1). It was intended to serve primarily static content from a Network-attached Storage, and privileging strict security, using separate web servers for internal (ATLAS Technical and Control Network - ATCN) and external (CERN General Purpose Network and public internet) access. During these years, it has become necessary to add to the static content an increasing number of dynamic web-based User Interfaces, as they provided new functionalities and replaced legacy desktop UIs. These are typically served by applications on VMs inside ATCN and made accessible externally via chained reverse HTTP proxies. As the trend towards Web UIs continues, the current design has shown its limits, and its increasing complexity became an issue for maintenance and growth. It is, therefore, necessary to review the overall web services architecture for ATLAS, taking into account the current and future needs of the upcoming LHC Run 3. In this paper, we present our investigation and roadmap to re-design the web services system to better operate and monitor the ATLAS detector, while maintaining the security of critical services, such as Detector Control System, and maintaining the separation of remote monitoring and on-site control according to ATLAS policies.

1 Introduction

The online farm of the ATLAS [1] experiment at the Large Hadron Collider (LHC) consists of nearly 4000 servers with various characteristics processing the data readout from 100 million detector channels through multiple trigger levels. The ATLAS servers are connected to a

*e-mail: atlas-tdaq-sysadmins@cern.ch

dedicated network, the ATLAS Technical and Control Network (ATCN), which is separated from the CERN General Purpose Network (GPN). The access from GPN to ATCN and vice-versa is allowed via the ATLAS Gateway servers which are connected to both ATCN and GPN.

In case of a disconnection from GPN, ATLAS is meant to be able to continue to take data, up to a couple of days as limited by the local storage capacity. To address this requirement, all core services, such as Active Directory, DHCP, DNS, NTP, repositories, are duplicated inside ATCN.

Various web services are used to operate and monitor the ATLAS detector, both locally, from the ATLAS Control Room, and remotely, from CERN and outside. The capability to monitor the detector status and the ongoing data taking is essential to ensure the safety of the detector, and efficient data taking. It is also necessary to ensure the security of critical services, such as Detector Control System (DCS), and to guarantee the separation of remote monitoring and on-site control, as required by ATLAS policies.

The architecture for the web services had been designed at the beginning of the ATLAS operation (Run 1 - 2009). As the trend towards adoption of Web User Interfaces (UI) continues, this base architecture has been expanded, but its design has shown its limits, and its increasing complexity has become an issue for maintenance, growth and security.

A review of the overall web services architecture, taking into account the current and future needs of the upcoming LHC Run 3 in 2021, has become necessary.

An investigation has therefore been started to re-design the web services system.

2 Current web service architecture

The initial design was intended to primarily serve static content from a Network Attached Storage (NAS) in ATCN, served by an Apache [2] HTTPd running on dedicated Linux servers.

Over time, an increasing number of dynamic Web applications and User UIs have been added, to provide new functionalities and to replace legacy Linux native desktop UIs. These web UIs are typically provided by a back-end application hosted on a separate VM, and reverse-proxied by the main web servers, in one or multiple steps.

Unfortunately not every Web application is fully compatible with the Apache reverse proxying, e.g. Icinga Web 2 [3], [4]. For dealing these exceptions, the ATLAS Gateways provide a caching and forwarding HTTP proxy service, Squid [5], which can be used by clients in GPN.

A schematic representation of the resulting current architecture is shown in Figure 1.

2.1 Original requirements and design

The original design from 2009, and its later extensions, have addressed the main needs, such as a good service availability, the controlled access to web services from ATCN and, for selected ones, from GPN and outside CERN domain, and the capability of surviving a disconnection of ATCN from GPN.

The original design adopted two physically separate web servers, both connected to the NAS. Originally, the content on the NAS was mostly the view of the detector status, which is generated by DCS applications as static HTML pages and images, stored on the NAS, and published by the web servers.

The internal web server provides the service to ATCN, in particular to the shift crew in ATLAS Control Room; it is independent of any GPN services and so it can continue to work in case of disconnection.

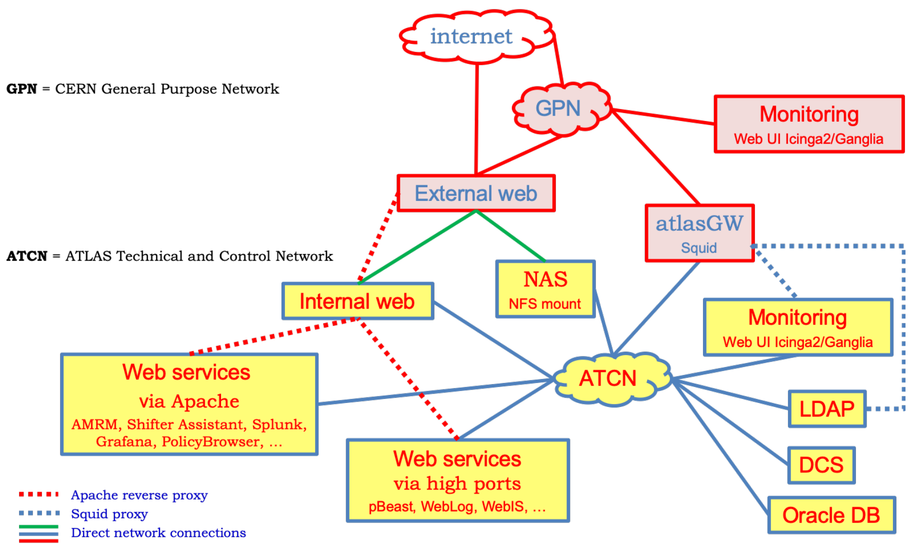

Figure 1. ATLAS Current Web Service architecture

The external web server provides service to GPN and public networks, allowing on-call experts to monitor the state of ATLAS from anywhere and at all times. Not every service that is available on the internal web server is made available on the external, and some are available with restricted functionality; as an example, the Operations wiki cannot be edited from outside, as the NAS storage is read-only for the external server.

The external web services are quite restrictive and control user access using the Single Sign-On (SSO) [6] service of CERN. As technical constraints prevent the SSO service from being available in ATCN, the internal one uses instead LDAP and Kerberos for controlling access to specific resources.

For service availability, a basic redundancy is provided by two additional "cold spare" servers which have been installed and configured similarly to the production internal and external web servers. In case of a failure on the active servers, they can be switched to take over in about 30 minutes, requiring a manual intervention to change the DNS aliases and consequently adjust the Apache configuration.

In the original design, only limited dynamic content was hosted on the web servers, namely a wiki system and some PHP pages. A few applications had already been reverse proxied in the first years. For example, the ATLAS electronic logbook is installed on the internal web server, so to be always available inside ATCN, but it is also reverse proxied to the external one so to be available from GPN and from the public internet. The log files and information related to the data taking are made available via reverse proxy from a dedicated application running on an online server to the internal web server, and then to the external one from where it is available outside. Similarly the Access Manager Role Management application [7], which is used to manage the roles and therefore the rights the users have inside the experiment, is reverse proxied from a dedicated VM to the public internet.

2.2 Security perspective

Security is one of the utmost important aspects to be ensured. On these servers the web areas with the static content from the NAS are mounted in read-only mode and only the administrators are allowed to login. A dedicated VM in ATCN has been setup with limited user access to allow non administrators users to modify the static web content.

All the servers providing Web Services are configured to use iptables and all new updates from the CERN Linux repositories are applied immediately. Rootkit Hunter [8], a Unix based tool that scans for rootkits, backdoors and possible local exploits, is also installed and regularly run. In addition, on the GPN servers which are exposed to outside the CERN domain, the log files are collected and sent to CERN IT to be analysed and to perform security scans.

Finally Samhain [9], an integrity checker and intrusion detection system, and ClamAV [10], an open source antivirus engine, scan the NAS web areas exposed by the web servers.

2.3 Current difficulties and future needs

In the last years a number of applications have been added to be reverse proxied from dedicated VM, showing an increase in the usage of Web User Interfaces as replacement for classic desktop GUIs. Some examples are an UI to browse the Access Manager roles (which rights are granted by each role, to who and on which server); Shifter Assistant (guidance and suggestions for the shifters); Grafana (visualizations and dashboards for metrics used to monitor the data taking and the network).

The first consequence of this is that the 30 minutes of possible downtime in case of server failure are becoming less acceptable to users, accustomed to the high availability of modern cloud based services.

The other and more severe consequence is that we can expect that an increasing number of Web applications and UI will need to be integrated in the architecture. As the back-end servers are hosted each separately on one or more Virtual Machines (VMs) inside ATCN, making them available to ATCN and outside requires a chain of up to three Apache reverse HTTP proxies [11] :

- from the external web server (GPN) to the internal one (ATCN)
- from the internal web server to the VMs providing the application back-end
- on the VM itself, from an exposed standard HTTP/S port to the localhost port used by the back-end application; in this case the Apache reverse proxy is meant to protect direct access and optionally to provide a TLS encryption layer

This complex setup of Apache reverse proxies on multiple hosts is considerably time consuming and error-prone; adjustments and maintenance in general become increasingly difficult. So a single solution to address all the needs would be desirable.

It is also desirable to clean up and simplify the Squid configuration, which is used on the ATLAS Gateway servers to grant access from and to specific sites.

Finally, from a security perspective it would be good to be able to remove the two direct network links connecting ATCN and GPN systems, as these two networks should be fully separated. The first direct link is between the external web server and the internal one; it is used to reverse proxy the applications running in ATCN to the outside. The second direct link is between the external web server and the NAS; it allows the external web server to access, in read-only, the static web pages and the wiki content which are hosted on the NAS.

3 Future web service design

The difficulties that we are faced at the present and the increasing request of Web User Interfaces, as described in Section 2.3, made it necessary to study a new architecture that should be simpler and scalable for integrating more applications in the future.

Our investigation led us to look at HAProxy [12]. It is an Open Source, very fast and reliable solution offering reverse proxying and load balancing for TCP and HTTP-based applications that spreads requests across multiple servers. The operation mode of HAProxy makes its integration into existing architectures easy and risk-free, while still offering the possibility not to expose fragile web servers to the network. As of today, HAProxy is the world's fastest and most widely used software load balancer.

In our case, the proxy functionalities that have been configured and maintained in Apache on the two web servers will instead be handled by HAProxy, which, contrary to Apache, is specifically designed for reverse proxies and load balancing.

In addition we can to improve the redundancy of the system by implementing a High Availability system providing fast, automatic failover. The idea here is to adopt "Keepalived" [13], which is an Open Source routing software for Linux. It implements the VRRP [14] protocol for fast failovers, and a set of checkers to dynamically and adaptively maintain and manage load-balanced server pool according their health.

3.1 Proposed architecture

The resulting future architecture is shown in Figure 2. The HAProxy server takes care of reverse-proxying the web applications directly from all the back-end servers. Being connected to both GPN and ATCN, it will serve clients on both sides. The external web server is dropped, as its function is completely replaced by HAProxy. The internal web server instead will become a back-end of HAProxy, and will only serve the static web pages stored on the NAS.

As of Autumn 2019, the first HAProxy server has been installed and configured and tests are ongoing. A second one, shown in Figure 2, will be added in future to achieve full redundancy. Similarly to the ATLAS Gateways, it has been connected to both GPN and ATCN to make it available from both of them and Virtual IPs configured for both to be used by Keepalived.

3.2 Advantages and disadvantages

The newly designed architecture will allow the removal of the existing direct connection between ATCN and GPN and to cleanup Squid, improving security aspects. For example, the Icinga2 monitoring system will be accessed directly via the HAProxy and will no longer need a Squid rule.

It will also simplify the overall Apache configuration as many multiple and chained reverse proxies will be removed from the configuration.

The only disadvantage identified yet is that a few aliases will need to be defined and used by HAProxy to match the Apache Virtual Hosts configuration on the various servers providing the web services So users will need to get used to new URLs.

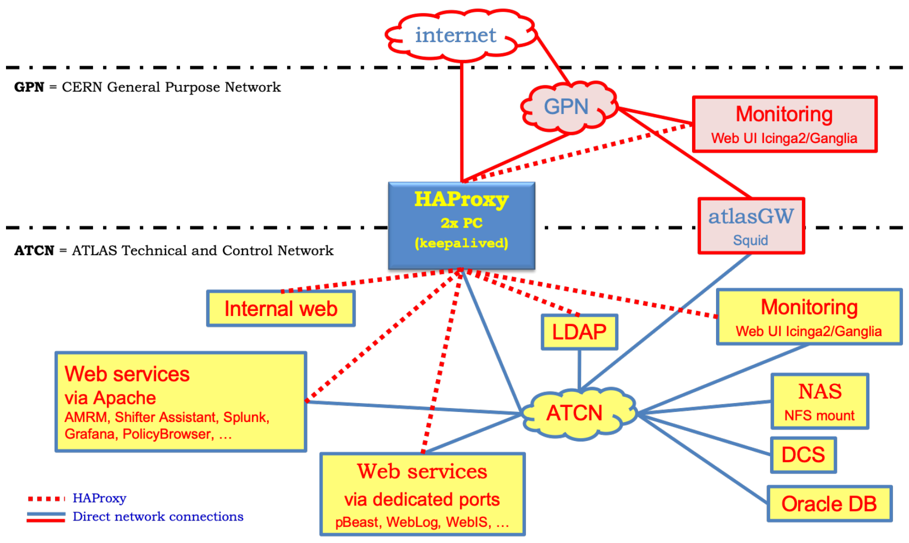

Figure 2. ATLAS new Web Service architecture based on HAProxy

4 Conclusions

The adoption of HAProxy has been proven to be a viable solution for the web services.

The initial tests showed that the monitoring URLs were accessible without the need of Squid, the basic static web pages are still available and some of the web applications (VMs based) are also accessible.

Thanks to its mode of operation, the HAProxy configuration can be performed in steps, allowing the current and the new configuration to coexist. This aspect is important as the current production environment cannot be disrupted.

Various features and requirements still need to be addressed and ironed out. The ATLAS DNS needs to be modified to answer with the internal (ATCN) IP to the requests sent to the external (GPN) one so to fulfil the requirement of surviving a disconnection from GPN.

Some applications are accessed via a specific network port which, in the current architecture, is used by the reverse proxy configuration.

Currently the ATLAS Operation wiki pages are read-only from GPN and outside CERN domain and read-write from ATCN, and the same functionality needs to be implemented into the new architecture. The Twiki pages are used to provide documentation for shifters and experts and should be modified only from inside ATCN.

The SSO implementation at CERN will soon change and the alternative has to be investigated to restrict the access to the HAProxy system.

The aim is to complete the transition to the new architecture before the start of Run 3 which as of today is scheduled for May 2021, in order to provide to the ATLAS community a modern, secure and reliable web service.

References

[1] ATLAS Collaboration, JINST **3**, S08003 (2008)
[2] *Apache*, https://httpd.apache.org/
[3] *Icinga2*, https://www.icinga.com/products/

[4] *Icinga Web 2*, https://icinga.com/docs/icingaweb2/
[5] *Squid*, http://www.squid-cache.org/
[6] *Single Sign-On*, https://en.wikipedia.org/wiki/Single_sign-on
[7] Valsan M L, et al., J.Phys.Conf.Ser. **331**, 022042 (2011)
[8] *Rootkit Hunter*, http://rkhunter.sourceforge.net/
[9] *Samhain*, https://www.la-samhna.de/samhain/
[10] *Clamav*, https://www.clamav.net/
[11] *Reverse proxy*, https://httpd.apache.org/docs/2.4/howto/reverse_proxy.html
[12] *HAProxy*, https://www.haproxy.com
[13] *Keepalived*, https://www.keepalived.org/
[14] *VRRP*, https://en.wikipedia.org/wiki/Virtual_Router_Redundancy_Protocol