



The Compact Muon Solenoid Experiment

Conference Report

Mailing address: CMS CERN, CH-1211 GENEVA 23, Switzerland



01 February 2020 (v4, 06 August 2021)

WLCG Web Proxy Auto Discovery for Dynamically Created Web Proxies

Dave Dykstra for the CMS Collaboration

Abstract

The WLCG Web Proxy Auto Discovery (WPAD) service provides a convenient mechanism for jobs running anywhere on the WLCG to dynamically discover web proxy cache servers that are nearby. The web proxy caches are general purpose for a number of different http applications, but different applications have different usage characteristics and not all proxy caches are engineered to work with the heaviest loads. For this reason, the initial sources of information for WLCG WPAD were the static configurations that ATLAS and CMS maintain for the Conditions data that they read through the Frontier Distributed Database system, which is the most demanding popular WLCG application for web proxy caches. That works well for use at traditional statically defined WLCG sites, but now that usage of commercial clouds is increasing, there is also a need for web proxy caches to dynamically register themselves as they are created. A package called Shoal had already been created to manage dynamically created web proxy caches. This paper describes the integration of the Shoal package into the WLCG WPAD system, such that both statically and dynamically created web proxy caches can be located from a single source. It also describes other improvements to the WLCG WPAD system since the last CHEP publication.

Presented at *CHEP2019 24th International Conference on Computing in High Energy and Nuclear Physics*,
Published in *EPJ Web Conf.*, 245 (2020) 07026, doi: <https://doi.org/10.1051/epjconf/202024507026>

Web Proxy Auto Discovery for Dynamically Created Web Proxies

Dave Dykstra^{1,*} 

¹Scientific Computing Division, Fermilab, Batavia, IL, USA

Abstract. The WLCG Web Proxy Auto Discovery (WPAD) service provides a convenient mechanism for jobs running anywhere on the WLCG to dynamically discover web proxy cache servers that are nearby. The web proxy caches are general purpose for a number of different http applications, but different applications have different usage characteristics and not all proxy caches are engineered to work with the heaviest loads. For this reason, the initial sources of information for WLCG WPAD were the static configurations that ATLAS and CMS maintain for the Conditions data that they read through the Frontier Distributed Database system, which is the most demanding popular WLCG application for web proxy caches. That works well for use at traditional statically defined WLCG sites, but now that usage of commercial clouds is increasing, there is also a need for web proxy caches to dynamically register themselves as they are created. A package called Shoal had already been created to manage dynamically created web proxy caches. This paper describes the integration of the Shoal package into the WLCG WPAD system, such that both statically and dynamically created web proxy caches can be located from a single source. It also describes other improvements to the WLCG WPAD system since the last CHEP publication.

1 Introduction

WLCG Web Proxy Auto Discovery [1] (WLCG WPAD) is a service that locates web proxies that grid jobs can use. The primary applications that use WLCG web proxies are the Frontier Distributed Database Caching system [2] (referred to as just Frontier below) and the CernVM File System [3] (CVMFS), but smaller use cases take advantage of the web proxies as well. ATLAS and CMS are the two primary users of Frontier, and they maintain their own lists of web proxies (that is, squids [4]) for use by Frontier at their traditional grid sites. They maintain their lists in very different ways, which makes it difficult for other applications to locate the squids and makes it difficult for ATLAS and CMS to run jobs opportunistically at each other's sites. The WLCG WPAD provides a standard mechanism for jobs to locate the squids.

As of this writing, the primary use cases of the WLCG WPAD system are LHC@Home [5], CMS jobs running opportunistically at non-CMS sites in the United States, and the default configuration of the CERN Virtual Machine (CernVM) [6]. Many LHC@Home jobs are also run at WLCG sites, and the WLCG WPAD directs those jobs to use the local squids at those sites. These use cases also utilize the openhtc.io Content Delivery Network (CDN, hosted on Cloudflare) for when jobs are run outside of WLCG sites [7].

Formerly, the WLCG WPAD service only included squids from the ATLAS & CMS lists of statically configured squids, plus a small number of statically configured exceptions. Now that the LHC experiments are branching out into dynamically created environments such as

* Co responding author: dwd@fnal.gov

commercial clouds, a more dynamic registration of squids was also needed. This has now been done by integrating a Shoal [8] server into the WLCG WPAD system and the Shoal agent into the Squid package distributed by the Frontier project. This integration is the primary focus of this paper. A secondary focus is some other recent developments in the WLCG WPAD service.

2 Frontier Squid/Shoal Agent Integration

The Frontier project maintains a Squid distribution called `frontier-squid` [9] which is the standard recommended Squid package for the WLCG. To make the dynamic registration as easy as possible to do, the Shoal agent software has been included in the `frontier-squid` package. In order for a system administrator to enable the registration, they simply set a configuration option². Then whenever the `frontier-squid` service is started, it contacts a special URL on one of the WLCG WPAD servers to find out the URL of the Shoal server and its own public IP address (the latter is often difficult for a client to figure out). On the recommendation of the Shoal maintainers, the URL of the Shoal server is currently their primary server at the University of Victoria, but that can be changed at any time. A configuration for the Shoal agent is created, and a `shoal-agent` is started as a background process along with the squid processes. The Shoal agent registers the Squid and updates the registration periodically, so if the host machine goes down then the Shoal server will remove the registration. If the `frontier-squid` service is stopped, then `shoal-agent` is also stopped.

Integrating the Shoal agent into the `frontier-squid` package was a challenge especially because `shoal-agent` is written in Python. Python dependencies can be difficult to manage because of various compatibility issues between different versions, so adding dependencies to the rpm could cause support problems. This is especially the case because the CERN distribution of `frontier-squid` has been using a single rpm file across different Red Hat Enterprise Linux (RHEL) operating system major releases. Those dependency issues were solved by using a tool called `Pyinstaller` [10] which builds a standalone distribution including all needed python library functions into a single binary file or compact directory. That makes the dependencies only needed at rpm build time, not at installation time.

3 WLCG WPAD/Shoal Server Integration

The four WLCG WPAD servers (two at CERN, two at Fermilab) get all their information and configuration from the WLCG Squid monitor (a primary/backup pair of machines at CERN) every 5 minutes. The WLCG Squid monitor reads all the registered squids from the University of Victoria Shoal server also every 5 minutes and puts a subset of the information into a JSON file that gets redistributed to the WLCG WPAD servers.

The subset of Shoal server information that is selected is the public IP address of each registered squid, the private IP address if known, and the city and country code of the squid. Only the addresses are really needed; the other information is just for diagnostics. If a private IP address of a matched squid is different than the public IP address, WLCG WPAD will tell clients to use the private IP address, otherwise it will tell them to use the public IP address. That choice might not always be the best, but worker nodes are often configured to communicate with squids over a private network so it is most likely correct.

Although the Shoal system itself matches clients to squids using geographical longitude and latitude, that is not the way the WLCG WPAD matches. It matches based on a different (non-free) Maxmind GeoIP database [11] that maps public IP addresses to organization

² Set `SQUID_AUTO_DISCOVER=true` in `/etc/sysconfig/frontier-squid`

names; clients and squids in the same organization get matched together. That method is more likely to locate squids that are accessible, since squids are normally restricted to be used within organizations. If publicly accessible caching is needed, the openhtc.io CDN caching is most likely superior anyway, so there is not a need for publicly accessible squids.

The WLCG squid monitor gets its WLCG WPAD information from multiple sources, as shown in Figure 1. If there is information about squids in a single organization from both Shoal and the statically registered ATLAS and CMS sources, WLCG WPAD gives priority to the static sources. It will only tell clients to use squids registered in Shoal if there are no static sources for an organization.

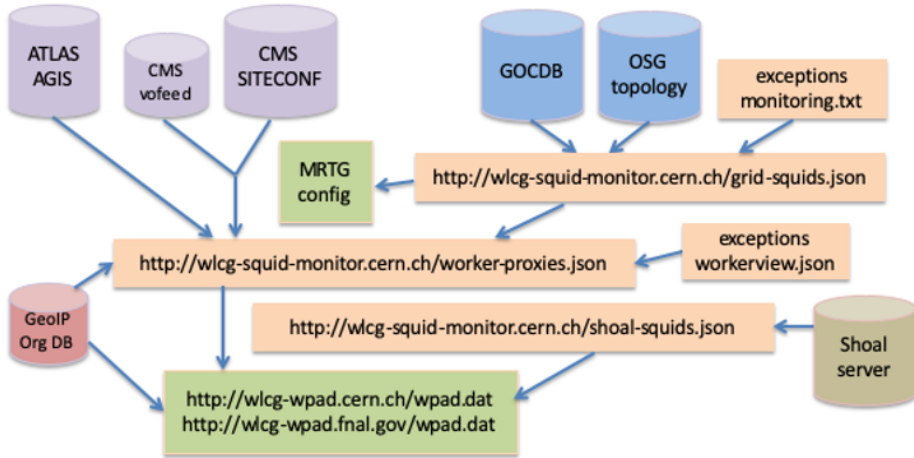


Figure 1. WLCG WPAD information flow

4 Limitation on WLCG WPAD/Shoal Integration

If there are multiple squids registered in Shoal for the same organization, WLCG WPAD will give a load-balanced list of all of the squids to all clients in that organization. This is not likely to be a good model especially for commercial clouds, where there might be more than one independent squid service, perhaps in different virtual Local Area Networks (LANs). One possible solution is to use the existing WLCG WPAD exceptions mechanism that supports distinguishing subsets of GeoIP organizations by manually registering IP address ranges. Alternatively, if all of the addresses in a cloud virtual LAN goes through a single Network Address Translation (NAT) address, WLCG WPAD could be easily extended to match based on that NAT address.

5 WLCG WPAD Variations

The WLCG WPAD service provides slightly different answers depending on which of several DNS aliases is used to access it in the cern.ch and fnal.gov domains. The behaviour variations are specified in a configuration file. The primary alias is **wlcg-wpad**. When a match is found to either statically or dynamically registered squids, the response includes backup proxies at CERN and Fermilab (which are on dedicated squid TCP ports on the same 4 machines) so that jobs at the grid sites will keep running in a degraded mode if their local squids are experiencing problems. The backup proxies are monitored, and the corresponding

squid owners are contacted by the WLCG Squid Operations team when the backup proxy usage is high from a given site. WLCG WPAD client requests are directed to 3 different TCP ports on the backup proxy machines, depending on the destination URLs: one for CMS frontier, one for ATLAS frontier, and one for all WLCG CVMFS. Those same backup proxy ports are used separately from the WLCG WPAD service as well, in static configurations around the WLCG. The `wlcg-wpad` alias returns an error if a match for a squid is not found.

The most heavily used alias is the LHC@Home alias, **lhhomeproxy**. That returns the same answers as `wlcg-wpad` when a squid match is found, but if no match is found it returns a response of only `DIRECT`, without any backup proxy machines. In that way the clients will use the `openhtc.io` Cloudflare CDN when no local squids are found.

The **cernvm-wpad** alias is used as a default configuration by the CernVM distribution. The intention for it is to make CernVM virtual machines work well by default with no custom configuration. The `cernvm-wpad` alias returns the same answers as `lhhomeproxy`, but additionally keeps track of the number of requests that return `DIRECT` from a given geoIP organization in a configurable amount of time. If the number of requests within the time period exceeds a configurable threshold, then all requests for that organization get redirected to a fourth squid TCP port on the backup proxy machines. In that way the WLCG Squid Operations team can notice cases where people need to be contacted to create their own local squid services. The current configured limits are set so that 125 requests within 15 minutes from the same organization will cause redirections to backup proxy machines for 12 hours.

The **cmsopsquid** alias is used by US CMS jobs run opportunistically at non-CMS sites, including some High Performance Computing (HPC) sites. Its configuration is almost the same as `cernvm-wpad`, except that too many `DIRECT` responses get redirected to the CMS frontier squid TCP port on the backup proxy machines instead of the `cernvm-wpad` port.

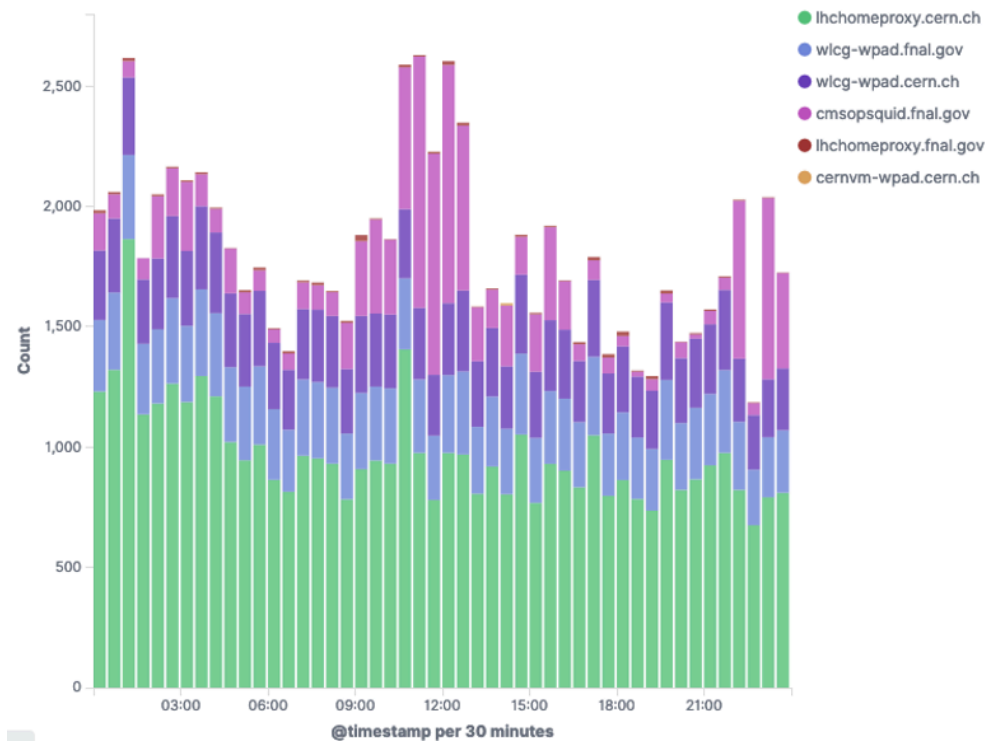


Figure 2. WLCG WPAD 24 hour activity (31 January 2020)

6 WLCG WPAD Monitoring

A new Kibana-based monitoring system for the WLCG WPAD service has also been created recently. It has plots for counts of the DNS aliases used, for the GeoIP organizations of the requests for each type of response, and for the types of responses from each DNS alias. Figure 2 is one of the plots for counts of the requests to the different DNS aliases. This monitor has proven to be helpful in quickly detecting and diagnosing operational problems.

7 Conclusions

The WLCG Web Proxy Auto Discovery service has been an effective in helping expanding WLCG computing to new resources. The addition of dynamic squid registration is expected to further aid that expansion to dynamically changing resources such as commercial clouds.

Acknowledgements

Thanks to Colson Driemel and Marcus Ebert of the University of Victoria for assistance with Shoal, and to Julio Lozano Bahilo from ATLAS and the Instituto de Física Corpuscular (CSIC-UV) for developing the Kibana-based WLCG WPAD monitor.

This document was prepared by CMS using the resources of the Fermi National Accelerator Laboratory (Fermilab), a U.S. Department of Energy, Office of Science, HEP User Facility. Fermilab is managed by Fermi Research Alliance, LLC (FRA), acting under Contract No. DE-AC02-07CH11359.

References

1. D. Dykstra, et. al., J. Phys. Conf. Ser. **898**, 052043 (2017)
2. D. Dykstra, J. Phys. Conf. Ser. **331**, 042008 (2011)
3. J. Blomer, et. al., J. Phys. Conf. Ser. **396**, 052013 (2012)
4. Squid: <http://www.squid-cache.org>
5. LHC@Home: <http://lhathome.web.cern.ch>
6. P. Buncic, et. al., J. Phys.: Conf. Ser. **219** 042003 (2010)
7. D. Dykstra, et. al., EPJ Web of Conferences **214**, 04023 (2019)
8. I. Gable, et. al., J. Phys. Conf. Ser. **513** 032035 (2014)
9. frontier-squid: <https://twiki.cern.ch/twiki/bin/view/Frontier/InstallSquid>
10. Pyinstaller: <https://www.pyinstaller.org>
11. Maxmind ISP database: <https://www.maxmind.com/en/geoip2-isp-database>