Improved ATLAS HammerCloud Monitoring for Local Site Administration

M Böhler[1], J Elmsheuser[2], F Hönig[2], F Legger[2], V Mancinelli[3], and G Sciacca[4] on behalf of the ATLAS collaboration

[1] Albert-Ludwigs Universität Freiburg, Germany
[2] Ludwig-Maximilians-Universität München, Germany
[3] CERN, Switzerland
[4] University of Bern, Switzerland

E-mail: michael.boehler@cern.ch

Abstract. Every day hundreds of tests are run on the Worldwide LHC Computing Grid for the ATLAS, and CMS experiments in order to evaluate the performance and reliability of the different computing sites. All this activity is steered, controlled, and monitored by the HammerCloud testing infrastructure.

Sites with failing functionality tests are auto-excluded from the ATLAS computing grid, therefore it is essential to provide a detailed and well organized web interface for the local site administrators such that they can easily spot and promptly solve site issues.

Additional functionality has been developed to extract and visualize the most relevant information. The site administrators can now be pointed easily to major site issues which lead to site blacklisting as well as possible minor issues that are usually not conspicuous enough to warrant the blacklisting of a specific site, but can still cause undesired effects such as a non-negligible job failure rate.

This paper summarizes the different developments and optimizations of the HammerCloud web interface and gives an overview of typical use cases.

ATL-SOFT-PROC-2015-009
10 May 2015

1. Introduction

To cope with the huge amount of LHC data, the ATLAS computing model is based on a world wide grid infrastructure [1] with a three tiered structure. The Tier-0 is based at CERN, next to the ATLAS experiment [2], and is directly linked to ten Tier-1 sites all over the world, which are typically based at regional research institutes. Every Tier-1 center is then connected with a number of Tier-2/3 computer centers, mostly situated at universities and research laboratories. The Tier-1/2/3 structures are usually clustered geographically and are referred to as clouds. Not only the data management, but also the monitoring and coordination activities are organized cloudwise.

Various computing sites in all timezones, which use different hardware and software configurations, need to work reliably for central Monte-Carlo(MC) production, reconstruction, and for user analyses. In order to guarantee the full functionality of the different hardware and software components on every single computing site, the HammerCloud framework is used intensively to send tailor-made test jobs for the different use cases, to monitor the results of these tests, and to automatically blacklist sites (set a site offline), if the validation tests fail and whitelist sites (set a site online) if tests succeed again.

The test creation and submission is based on the distributed analysis framework GANGA [3]. The infrastructure of the HammerCloud framework and the GangaRobot, a tool designed to perform the regular testing, are described in detail in Refs. [4, 5]. This paper focuses on the HammerCloud web based monitoring and in particular on the new features added in the last year to provide local site administrators with a reliable tool to discover and tackle site issues, e.g. after their site has been auto-excluded by HammerCloud.

The HammerCloud framework is based on Django [6], a Python-based framework for web interfaces with database access. The HammerCloud web interface is used for both publishing and browsing the test results and adminstrative operations, e.g. creating and maintaining tests. The model-template-view structure of Django allows the addition of web pages in a modular way. The different web pages presented in the following are designed with Java Apps provided by jQuery 1.11.1 [7] and Highcharts 2.3 [8].

The second section discusses how the typical workload of the HammerCloud job submission can be monitored through the new workload monitoring web page. The site blacklisting mechanism with the dedicated monitoring is described in detail in the third section. Since the ATLAS grid is organised in different clouds, an extra cloud monitoring web page has been provided to summarise the site efficiencies within the different clouds. This page is discussed in the fourth section followed by a description of the site monitoring, which provides full information of all test jobs per computing site. Before the conclusion, a dedicated monitoring page for nightly software tests is briefly presented in section 6, and the new prototype of a benchmark monitoring web page is introduced in section 7.

2. HammerCloud Workload Monitoring

The HammerCloud framework submits many different test types in order to cope with the different needs: the Analysis Functional Tests (AFT) are used for site validation and blacklisting of the ATLAS analysis queues, the Production Functional Tests (PFT) are evaluated for the production queues, and further functional tests are submitted to test additional functionalities which are not critical for usual operation, but need to be monitored as well. The stress tests are submitted infrequently and are used to run large scale tests to measure the behaviour of the site under heavy load. The "nightly tests" are submitted to a couple of sites to test software development releases with realistic conditions.

Every day HammerCloud submits up to 60,000 test jobs to all ATLAS computing sites around the world, as shown in Fig.1 for a one-month time period. In average 17,000-25,000 AFT jobs are sent to the user analysis queues and 10,000-15,000 PFT jobs are sent to the ATLAS production queues, used for centrally organized Monte-Carlo production. The AFT and PFT results are then evaluated for site blacklisting. Further functional tests, around 7,000 jobs per day, up to 13,000 stress test, and 100 nightly tests, are performed by HammerCloud every day, as shown in Fig.1.

3. Site Blacklisting

The blacklisting functionality, being the tool that takes the decision of setting a site on- or offline, is the most critical operation of the HammerCloud framework. It has, hence, to be monitored carefully to guarantee both a reliable blacklisting method and an easy/transparent message system for site administrators, so that they can get immediate feedback for any occurring site issue. Figure 2 shows three elements from the auto-exclusion web page. The first one is the number of excluded queues. In total there are over 250 ATLAS queues world wide. If some central services fail, it might be possible that many or even all sites are excluded simultaneously. In such a situation ATLAS central grid operations is notified automatically by e-mail. A prominent spike in the *Number of excluded sites* chart indicates such an event. The site administrator can immediately see that it is a central problem and not a site issue. The

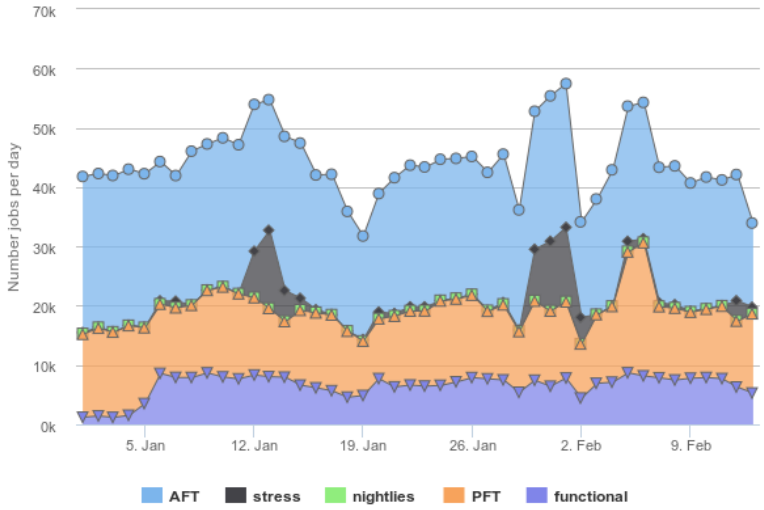


Figure 1. Total number of running test jobs on all ATLAS computing sites. The test jobs are split in Analysis Functional Tests (AFT) and Production Functional Tests (PFT) used for site blacklisting, stress tests used for heavy load on-demand site testing, additional functional tests, and tests of new software releases (nightlies).

second element (Fig. 2 middle) shows the number of blacklistings per day and cloud. This chart helps to spot cloud specific problems. The third element (Fig 2 bottom left) is the table with the top excluded sites for the last 30 days, which shows the *number of site exclusions per site* (if a site provides several queues, it is per queue). If a site is failing the critical AFT or PFT jobs the corresponding queue is turned off for analysis or production, but the test jobs continue to be sent there. When the test jobs succeed again, the queue is set online automatically. Since the blacklisting is evaluated every 30 minutes, a large number of site exclusions for one single site during a few days are a good hint that there are some problematic worker nodes. In this case, the test jobs fail when they are processed on problematical nodes, but the site is set online again when later tests are processed on properly working nodes.

4. Cloud Monitoring

Since the ATLAS computing grid is organized in clouds, both for the technical and for the organisational point of view, the HammerCloud monitoring also provides cloud internal comparisons. Only the results of the AFT and PFT jobs used for blacklisting are summarised in the cloud overview of Fig. 3. Two tables, one for AFT and one for PFT jobs, show the total number of tests for a given time interval (can be selected between 1 -180 days) and the fraction of succeeded and failed jobs. The table can be sorted and exported into different file formats. The total number of test jobs per day and queue is also visualized in a chart (see Fig. 3 bottom) which is useful for monitoring whether the test jobs are processed continuously.

5. Site Monitoring

The site monitoring is the most powerful tool for site administrators to follow the progress of different tests: it shows if test jobs are running with an acceptable frequency and possible failure reasons. The user of the monitoring page needs to specify the time range, the site, and the test type (AFT and PFT jobs used for blacklisting, other functional test, stress tests, all types of tests), then all job results are evaluated accordingly and displayed in a table similar to the one of Fig. 4. For every test type, an extra table organized in tabs is created. The table shows time lines for 24 hours binned in 30 minute time intervals. The total efficiency per day is listed in the column on the right. The bins show all jobs completed in this interval sucessfully (c), all jobs failed (f), multiple final states (m), and no job finished (0). By selecting any time bin with a finished job, a link to the BigPanDA web interface [9, 10] is activated to allow the retrieval of the full details of the job. An expandable error report list indicates the time of failure, the

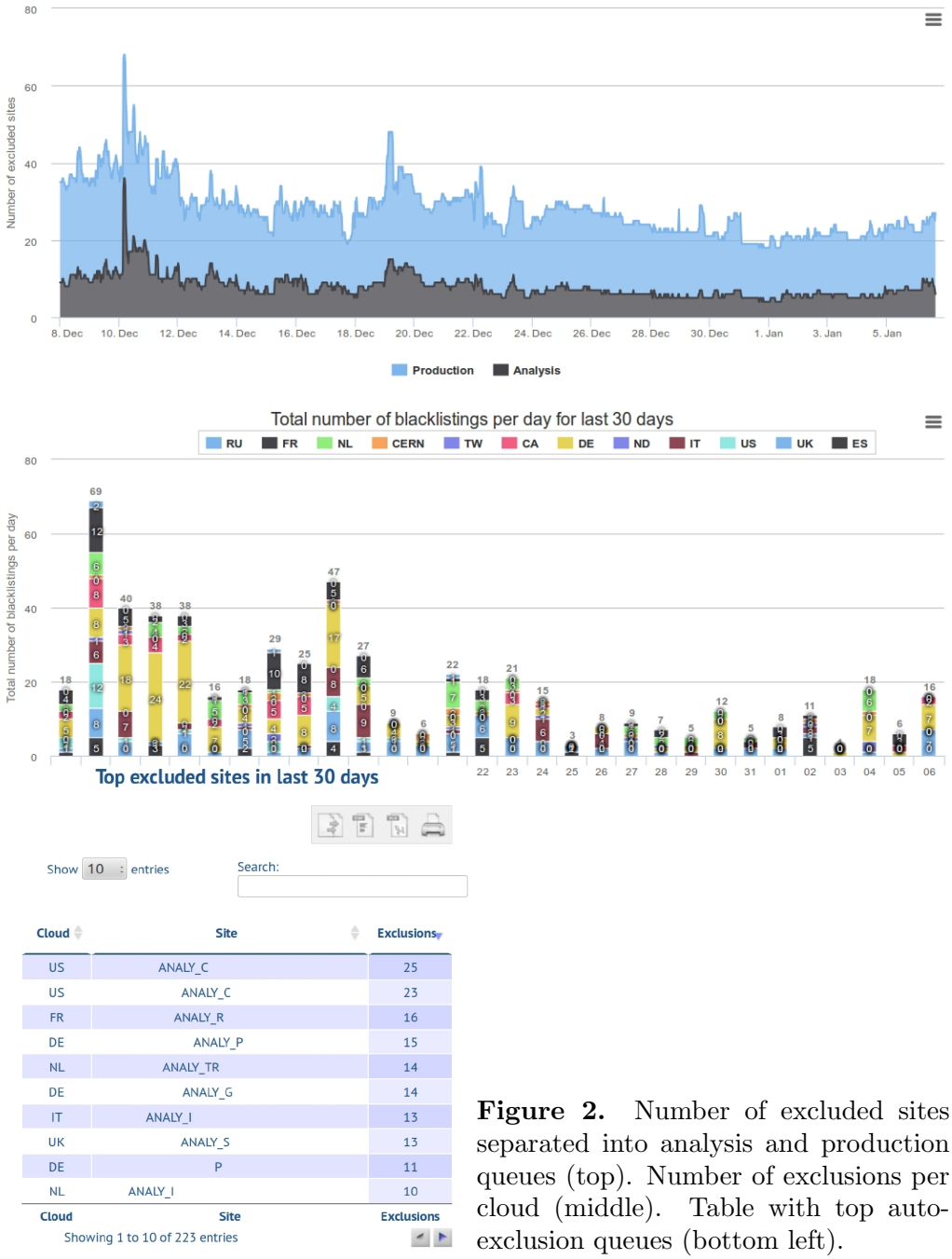
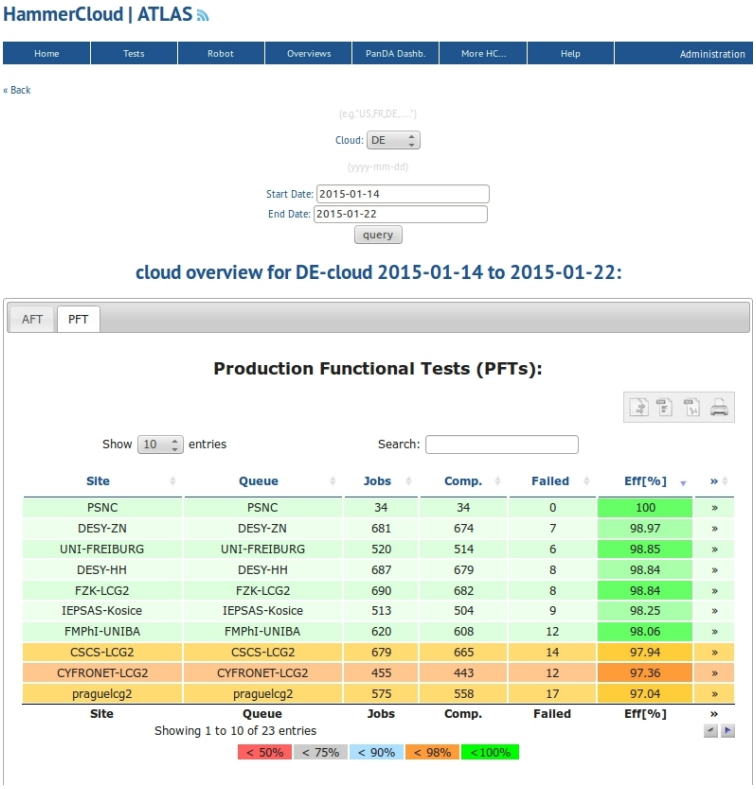


Figure 2. Number of excluded sites separated into analysis and production queues (top). Number of exclusions per cloud (middle). Table with top auto-exclusion queues (bottom left).

hostname of the node, where the job was processed, the PanDA job ID and the error message. Additional charts show the errors per worker node and the efficiency per CPU type.

6. Nightly Test Monitoring

The nightly test monitoring shares several functionalities with the site monitoring. Since very few test jobs are submitted per day in order to validate new software releases (nightlies), the results are arranged differently. Usually up to 10 different nightly releases are tested per day on up to 5 different computing sites. Therefore the same tested software releases are shown



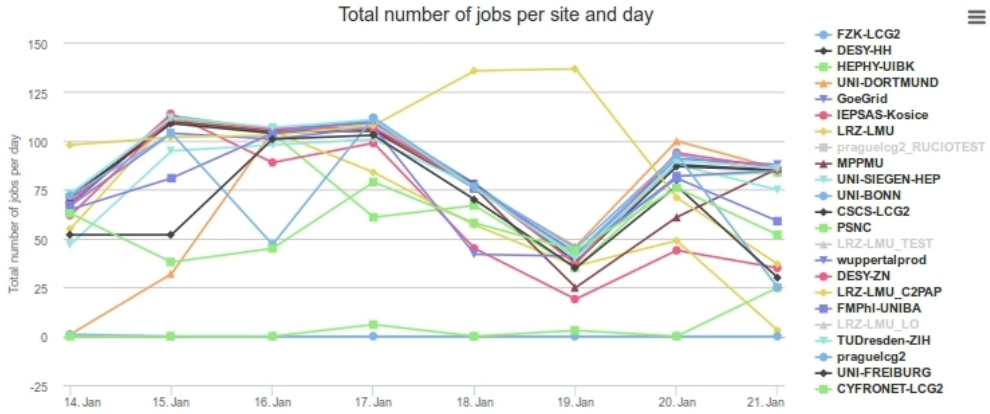


Figure 3. Cloud overview. This page considers explicitely the test jobs used for site blacklisting. The chart gives an overview of the total number of jobs processed per day on a particular queue. The table summarizes the test job efficiency for AFT and PFT jobs separately.

together in one block (see Fig. 5). Tabs separate the different sites on which these tests were performed. This allows to compare directly the results for one nightly release performed at different sites. In the first column the test job efficiency per test is summarised and highlighted in different colors (100%, 99% - 75%, 74% - 50% and smaller then 50%) for a fast look-up for release shifters. Furthermore, the lifetime of the test (usually 24 hours), the tested software release, and the job states are shown in the table. The job states are again directly linked to BigPanDA web interface for further details. The expandable error report allows a quick check of why jobs have failed without consulting other monitoring services.

7. HammerCloud Benchmark Monitoring

The wide variety of different hardware components installed at the different computing sites makes the direct comparison of the site performance almost impossible. Nevertheless, a dedicated

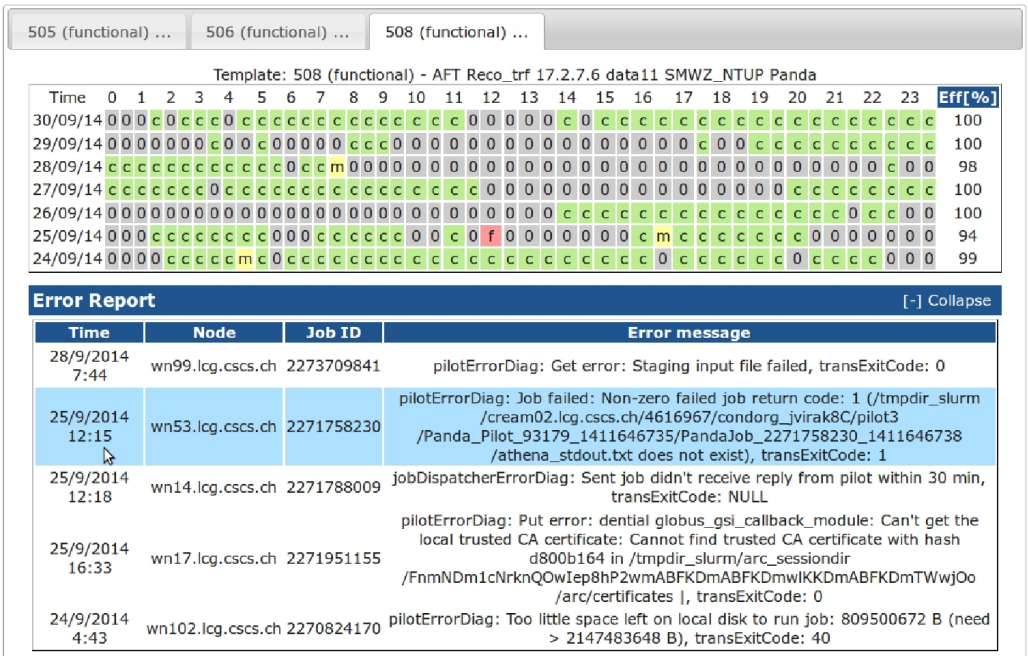


Figure 4. The site overview web page provides for every queue on a given site this tabular environment. For every test type an extra table is displayed and can be selected by a tab. A detailed error report can be expanded. The table shows a timeline per day binned in 30 minutes time intervals. 0 entries show that no test job finished in 30 minutes. The bins are linked to the BigPanDA web interface for further job details.

benchmark test is sent to ∼100 sites all over the world. It runs over 10 events per job, with the same random seed to avoid completely different event topologies with completely different simulation times. Because these benchmark tests are sent with a rather low frequency, less than 500 jobs per day in total at all sites, only an evaluation over a longer period, e.g. the last 30 days can give a hint of the site performance. These tests measure the average processed event rate in [1/s]. The comparison of the event rate is shown in Fig. 6 (upper plot). The histogram is sorted by the event rate: on the right side, one can find the "fastest" site, which, for this specific test, is *Tokyo-LCG* with an event rate of 0.0043 processed events per second. The error bars indicate the spread over the different test results per site. An interesting aspect of this comparison is that sites with a huge error bar, such as e.g. *UNI-FREIBURG*, have many different CPU types installed which have sensibly different performances, while sites which use an unique set of computer nodes types, e.g. Tokyo-LCG for the tests analysed in Fig. 6, have a much smaller error bar.

In order to consider the performances of the different compute node generations installed at the individual sites, an additional histogram, also published on the benchmark web page (see Fig. 6 lower plot) shows the event rate versus the number of the HEPSPEC06 [12] normalized to the number of logical CPUs. The y-axis shows the measured event rate and the x-axis considers the benchmark value in HEPSPEC06 measured by each site individually and published on the REBUS web interface [13], normalized to the total number of logical CPUs.

One would expect a rising linear relation in this two-dimensional histogram, since the larger the HEPSPEC06 value per CPU, the higher one would expect the event rate to be. Also shown is an error bar indicating the spread of the different test results per site. The same behaviour

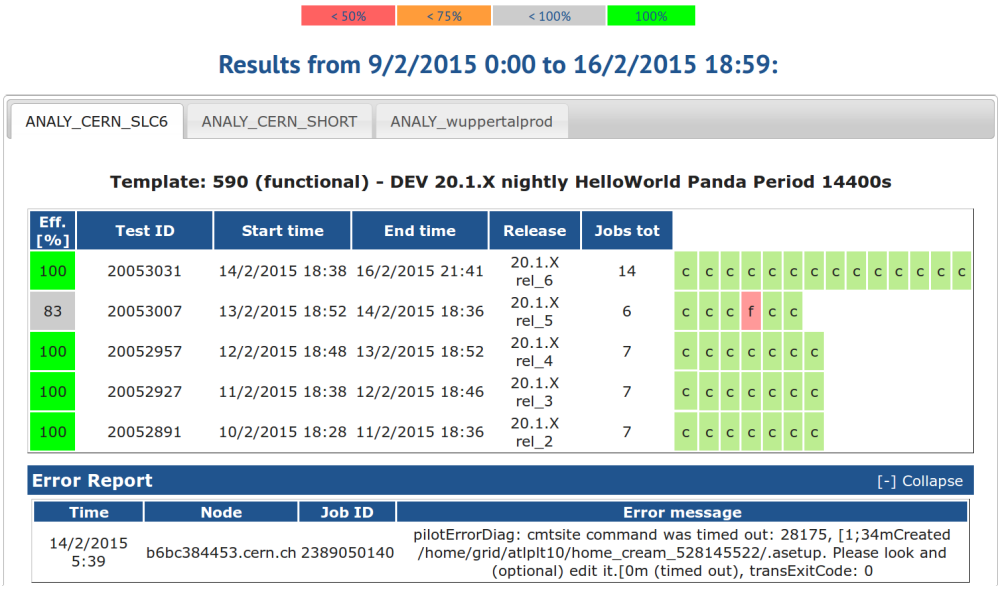


Figure 5. The nightly monitoring provides the same features as the site monitoring, with only small adjustments. The tables with the same test types are arranged next to each other, can be selected by tabs and there is no timeline, since each test job is displayed individually.

is seen as in the event rate histograms (see Fig. 6 upper plot) that multi-CPU sites like *UNI-FREIBURG* have large error bars while sites with unique or similar CPU types have a much smaller spread in event rates. Even though the distribution exhibits an increasing event rate for larger *HEPSPEC06/logical CPUs* values, most sites have large error bars, so the increase is rather inconspicuous. An additional comparison of event rates per CPU type (not shown here) provides another comparison of direct performance across sites.

8. Summary and Conclusion

The HammerCloud framework is heavily used for site exclusion and a clear, detailed, and intuitive monitoring has been provided. The new functionalities, developed to extract and visualize the most relevant information, have been described in detail. The site administrators can now easily detect major site issues, which may lead to site blacklisting, as well as possible minor issues, more difficult to spot but which, nonetheless, need to be fixed. The updated site blacklisting monitoring was improved in terms of speed and additional information was added. The site administrators can now easily judge if their site was blacklisted due to a central failure or due to site problems. The cloud monitoring was developed for a better overview within the different ATLAS computing clouds to allow for an easy comparison of issues within a cloud. The most important HammerCloud tool for site administrators is the newly implemented site monitoring, which allows to track down problems, and to understand if no HammerCloud test jobs could be sent to a given queue. It also provides links to the central BigPanDA web interface for more details, e.g. the log files of the test job. Including software nightly tests into the HammerCloud testing infrastructure required a dedicated monitoring tool. Based on the developments for the site monitoring a tailor-made nightly monitoring web page was created. Finally, the HammerCloud Benchmark monitoring allows for a comparison of site performances, e.g. the direct comparison of the event rate per CPU type of different sites allows to compare

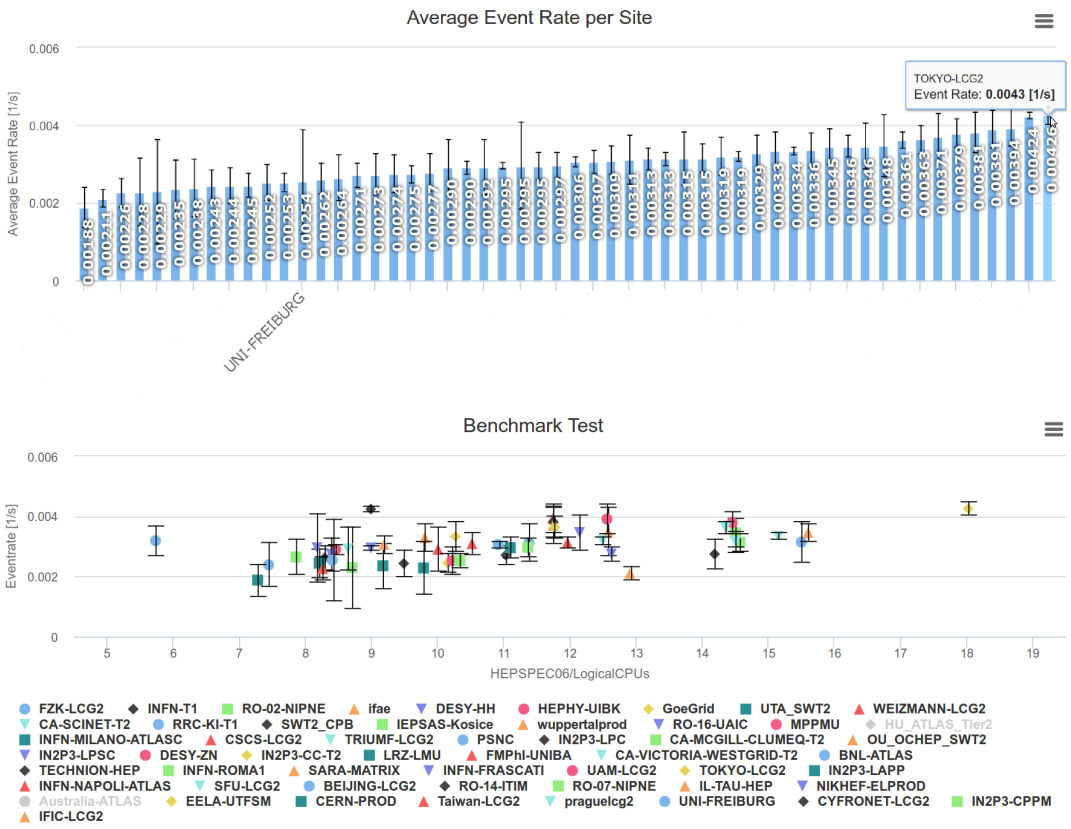


Figure 6. Dedicated benchmark tests are sent to ∼100 sites worldwide, which run over 10 events per job, in total up to 500 jobs per day. The Benchmark web page considers all tests of the last 30 days and compares: the averaged event rate per site (upper plot) the averaged event rate over HEPSPEC06/logical CPUs per site (lower plot).

and optimize individual setups.

References

[1] ATLAS Collaboration 2005 CERN-LHCC-2005-022
[2] ATLAS Collaboration 2008 *JINST* **3** S08003
[3] Moscicki J T et al. 2009 *Computer Physics Communications* Volume **180** Issue 11 Pages 2303-2316
[4] Legger F on behalf of the ATLAS Collaboration 2011 *J. Phys.: Conf. Ser.* **331** 072050
[5] Elmsheuser J et al 2014 *J. Phys.: Conf. Ser.* **513** 032030
[6] Django (Version 1.5) 2013 webpage https://djangoproject.com
[7] jQuery Foundation Documentation 2015 webpage: http://contribute.jquery.org/documentation/
[8] Highcharts 2015 webpage: http://www.highcharts.com/docs
[9] Nilsson P et al. 2008 PoS(ACAT08)027
[10] Maeno T et al. 2008 J. Phys. Conf. Ser. **119** 062036
[11] Kaushik D on behalf of the ATLAS Collaboration 2015 *The Future of PanDA in ATLAS Distributed Computing* Proceedings of the CHEP2015 conference *J. Phys.: Conf. Ser.*
[12] J L Henning 2006 *SIGARCH Comput. Archit. News* Volume **34** Issue 4 September 2006 Pages 1 - 17
[13] WLCG REsource Balance & USage 2015 webpage: http://wlcg-rebus.cern.ch/apps/topology