Integration of Grid and Local Batch Resources at DESY

IOP Conf. Series: Journal of Physics: Conf. Series **898** (2017) 082020 doi:10.1088/1742-6596/898/8/082020

Integration of Grid and Local Batch Resources at DESY

Christoph Beyer*, Thomas Finnern*, Andreas Gellrich*, Thomas Hartmann*, Yves Kemp*, Birgit Lewendel*

*DESY, Notkestraße 85, D-22607 Hamburg, Germany

E-mail: thomas.hartmann@desy.de

Abstract. As one of the largest resource centres DESY has to support differing work flows of users from various scientific backgrounds. Users can be for one HEP experiments in WLCG or Belle II as well as local HEP users but also physicists from other fields as photon science or accelerator development. By abandoning specific worker node setups in favour of generic flat nodes with middleware resources provided via CVMFS, we gain flexibility to subsume different use cases in a homogeneous environment.
Grid jobs and the local batch system are managed in a HTCondor based setup, accepting pilot, user and containerized jobs. The unified setup allows dynamic re-assignment of resources between the different use cases. Monitoring is implemented on global batch system metrics as well as on a per job level utilizing corresponding cgroup information.

1. Introduction

DESY maintains and supports a manifold scientific program on site and contributes to global science efforts. This includes analyses of High Energy Physics experiment data as well as the development of future accelerators or detectors, photon science at one of the largest free electron lasers, astro-particle, and theoretical physics. For these various users, DESY serves as one of the prominent national scientific IT infrastructures for data management and analysis. Since the miscellaneous science fields tend to have also differing workflows and analysis demands, the infrastructure has to be flexible to adapt to the various requests.
To reach the necessary flexibility with the analysis infrastructure, we try to separate all experiment and user specific software and dependencies from the actual worker node for the most general and unspecific compute node as possible. All user specific components are supposed to be loaded dynamically onto a node on request. This allows the creation of an integrated single batch system for the previously separated Grid batch system and local batch system.
While the Grid batch system served the international high-energy physics community in a relatively homogeneous framework, the local batch system, called *Batch Interactive Resource at DESY* (BIRD), has been providing analysis computing resources for users in the national HEP community [1].

2. Motivation

Before initiating the consolidation into one batch system, DESY had been running a batch system based on PBS/MySched[2] for Grid analyses since 2004. A local batch system based

on Son of Grid Engine[3] was created in the *National Analysis Facility* (NAF) effort [4] in 2007. As of end 2015 before the consolidation, the Grid farm offered 150,000 kHS06 on ∼15,000 cores [1] and the NAF about one third at ∼ 50,000 kHS06. Job submission was authorized for the Grid farm on CREAM CEs [5] with X509 based Grid certificates while the use of NAF resources required a registration of each user at DESY for a lightweight local account. While the computing infrastructures were separated, both batch systems already shared access to local storage instances.
During the growth of both compute clusters, scaling issues appeared for both batch systems. Especially for the initial PBS/Torque setup off the Grid cluster, serious scaling issues appeared that lead to the creation of an own scheduler *MySched* to scale beyond a few thousand active jobs in parallel.
With both batch systems requiring similar levels of operation and maintenance, the merge of both applications into one infrastructure reduces the needed level of manpower. In addition to the technical merger of the batch systems, the operational organization is to be rationalised to separate the day-to-day operations from high-level maintenance or development.
As basis for the combined batch system, HTCondor [6, 7] was chosen due to the wide-spread usage and support in the HEP community as well as its proven stability, flexible configuration and active development.
The final expansion stage will subsume about 20,000 cores into one batch system.

3. Generic Worker Node Setup

In our batch worker node setup we aim to keep a node as generic as possible and keep specific user dependencies separated. Thus, a worker node is installed and managed using a Puppet/Foreman setup with only a Scientific Linux 6 [8] or CentOS 7 [9] operating system with additional DESY-specific configurations. The HTCondor worker node client is added and its configuration managed via Puppet from a centrally managed git repository. Figure 1 shows a schematic illustration of a generic worker node with external file systems and caches for user dependencies. To make data accessible for processing, users can either use HTCondor's own staging mechanism or use shared network file systems. Due to their history, Grid jobs do not utilize the staging feature.
A fast shared scratch space, *DUST*, is available on each worker node via `NFS4`. The DUST system uses the IBM Spectrum Scale technology (formerly known as GPFS) [10] and is optimized for high parallel file I/O for job inputs and outputs.
Similarly, long term storage, based on dCache [11], is imported via `NFS4.1` onto each node for read access. For space and load management, major user groups have their own dCache instances with up to ∼200 storage pools, serving local jobs as well as grid-wide data flows in parallel.
For both, project and long term storage, access control is based on user and group IDs, where the group identity is set according to each group's VOMS mapping or login server (for details see section 4). Additionally, native `NFS4` Kerberos access control is evaluated, but currently not used.
Kerberos token based authentication is already possible in the cluster as it is a requirement for allowing access to the local AFS file system [12]. Here, the Kerberos token is used to request a corresponding AFS token. The full authentication chain will be added as soon as the support is available in HTCondor for Kerberos/AFS token submission (currently testing with the HTCondor team). Due to the architecture of HTCondor, the users' job submission has to be kerberized as well. To prepare long-running campaigns, Kerberos tokens with lifetimes up to two weeks can already be created. A transparent renewal or replacement of tokens as well as the

[1] We operate worker nodes with hyper-threading enabled on supported CPUs.

handling of tokens in the HTCondor job wrapper's context are also currently in development.
Providing AFS on the worker nodes is still necessary, since traditionally user home and group directories at DESY have been exported via AFS and legacy user workflows still rely on it. Due to the limitations of AFS, users are encouraged not to use the AFS file system for massive parallel or high I/O file access and are encouraged to migrate their workflows to use the other, more powerful and future-proof, storage options.
Grid related static client software, e.g., binaries, libraries or configurations, are dynamically staged on request using CVMFS [13]. To load-balance requests for files in CVMFS, DESY operates several squid web proxies for the WLCG and other HEP experiments. Additionally, DESY hosts own CVMFS stratum-0 and stratum-1 servers for Belle II, ILC and one instance for dedicated local usage. The introduction of CVMFS has greatly reduced the maintenance work for Grid user with the user groups centrally administering their respective software frameworks. Thanks to the separation of client software from the actual worker node setup, scaling proved to be simple as additional nodes can be transparently integrated into the HTCondor cluster. Currently, we operate the HTCondor cluster of ∼12,000 slots on 283 nodes with one active and one fall-back HTCondor central manager. The HTCondor resource brokers are fed by two Grid ARC-CEs [14] with each a HTCondor scheduler and for the local batch system one remote submit scheduler, which serves the group workgroup servers. Without apparent load limits, we are confident that the batch system's final expansion stage can easily be handled with the current setup. Nonetheless, we are investigating if an extended setup may be beneficial for a resilient operation with additional servers for load-balancing and fall-back.
A schematic overview of the consolidated batch system is shown in Figure 3 with Grid and local batch system identity management nodes (see section 4) brokering jobs to the worker node farm with attached network file systems and caches.

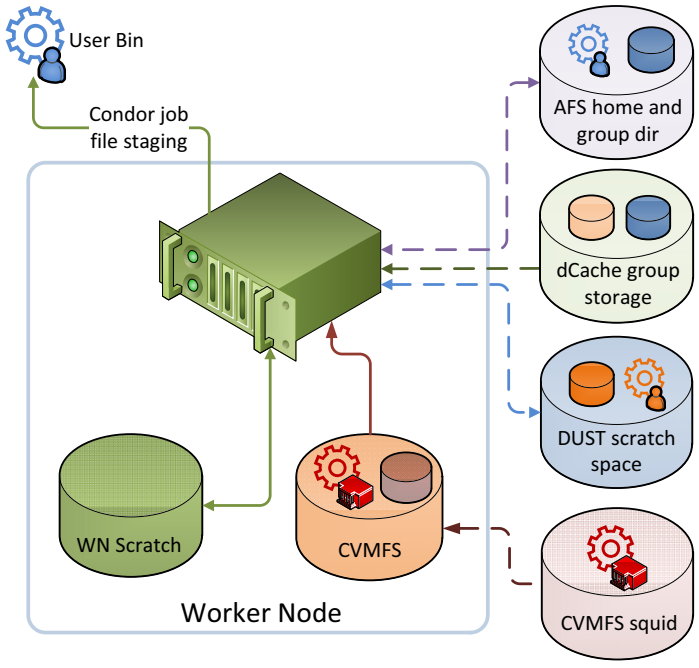

Figure 1. Generic batch worker node with basic operating system and HTCondor client installation and user requirements realized as external resources. Fast scratch "DUST", dCache long-term storage, and user and group home directories are mounted via NFS4 or AFS network file systems. Client software is staged and cached via CVMFS.

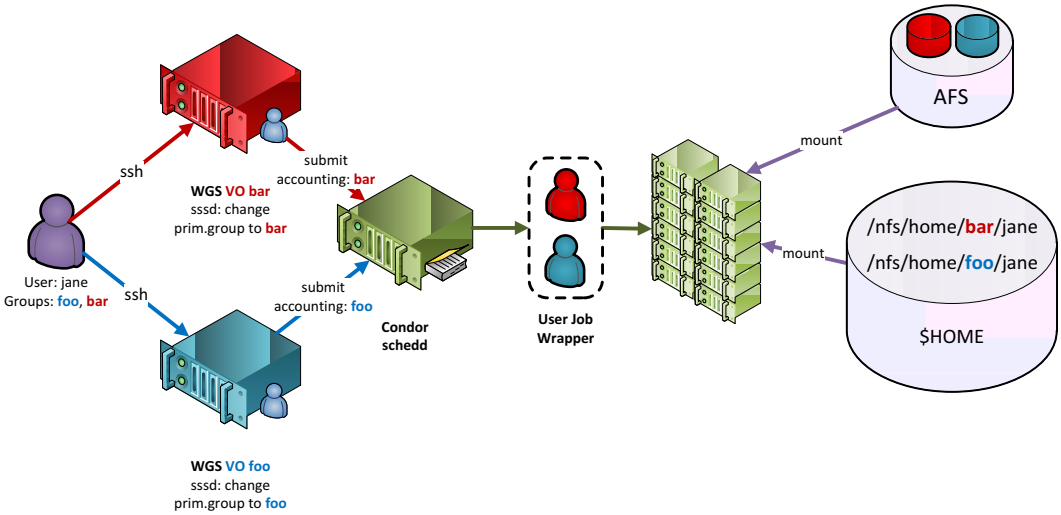

Figure 2. Workflow for submitting user jobs in the local batch system view. For a specific group identity, the user has to login to the corresponding group's workgroup server. From the workgroup server batch jobs are remote submitted to a dedicated scheduler host, which keeps the job information and submits the jobs to the cluster. The `$HOME` setup is currently discussed as an alternative to `$HOME` on AFS.

4. User Authorization, Authentication and Mapping

Due to both user communities historically being separate, the authentication and authorization is handled twofold for the Grid and the local bath system users.

4.1. Grid

Grid jobs can be submitted to two ARC Compute Elements directly or an upstream DNS load-balancer. With valid Grid proxies, the VOMS roles and groups are mapped to local users and groups, e.g., atlasusr007:atlasusr ... cmsplt001:cmsplt ... belleprd002:belleprd. Compared to several hundred identities before, the number of identities to be mapped has been greatly reduced since the advent of Grid pilot jobs. Due the limited number of static identities, the mapping to local user IDs can easily be handled manually as a table.

Both Grid ARC CEs each run a HTCondor `schedd` schedule daemon. Whether an externalisation of the submission to a separate scheduler with remote job submission is feasible, has not been evaluated but may be interesting for the future as well as investigating the HTCondor own CE implementation as alternative.

Due to the small number of Grid groups and small number of users per group, resource allocation in HTCondor can also be managed as a small group table.

Accounting information is aggregated on the ARC CEs both for the WLCG/EGI resource management as well as for internal evaluation from the HTCondor schedd daemons on both CEs.

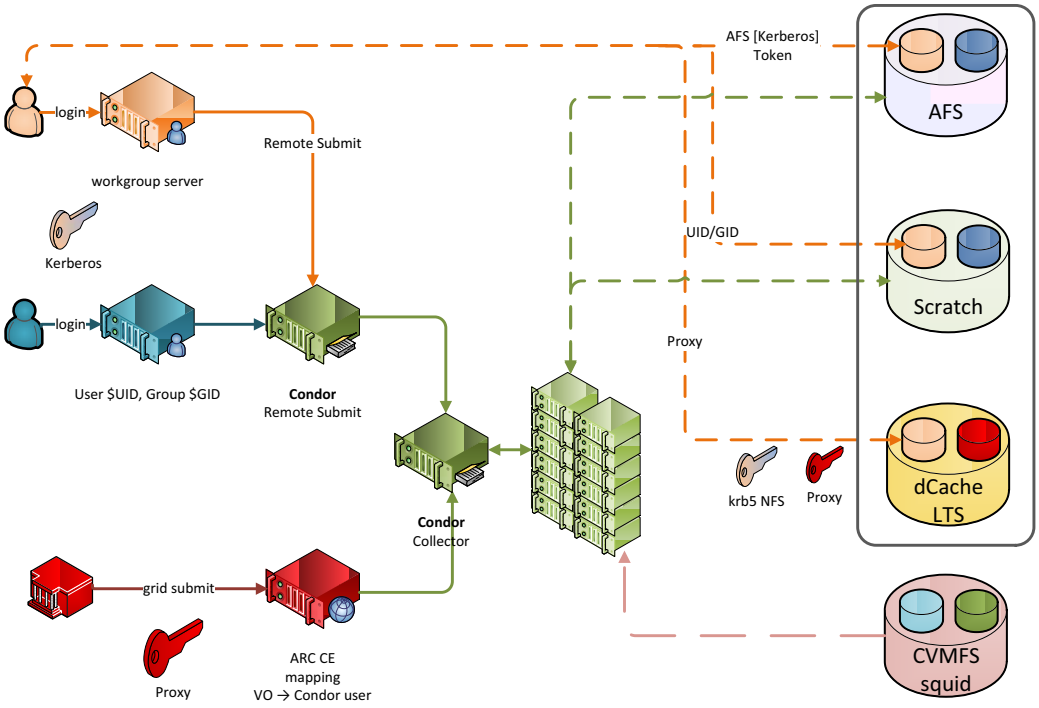

Figure 3. Schematic overview of the HTCondor batch system with job submissions via ARC-CEs for Grid jobs and group-specific workgroup servers for NAF users with storage back-ends for fast scratch usage, long-term storage, home directory access and CVMFS cache.

4.2. Local Batch System

As a national resource, the local batch system can be used by a large number of users. A user is either eligible as ordinary user, i.e., because he/she is either working on the DESY campus, or can obtain a lightweight user account for identity management only if working at another institute. For job submission, the user has to login to his or her group's workgroup server. Workgroup servers do not run scheduler daemons as the Grid CEs but do a *remote submit* of jobs to a dedicated scheduler server, which hosts the job's shadow information mirroring the worker nodes' job states. Thus, workgroup servers can easily be scaled or removed without loosing jobs.

Access control to data on the attached network storages is based on user/group IDs for files in NFS4.1 attached storages and Kerberos/AFS tokens for files in AFS, respectively. Since users can potentially be members of more than one group, we use the workgroup servers for identity management as well. To access a specific group's shared data, a user has to login to the respective group's workgroup server, where his or her primary group is set via `sssd` to the corresponding primary group (see Figure 2). With the user's group switching, home directory paths etc. can be adapted in a job wrapper to reflect the selected group.

In the current development state, a user's home directory points to the user's home directory in AFS and the primary group ID changed according to the workgroup server. In the future combined with a migration from AFS, we are planning to make Grid and user directories fully determined by the login server with optionally group specific home directories, i.e., user home directories within groups' namespaces, e.g., `/nfs/home/$GROUP/$USER`. Additionally, resource

shares and accounting will be harmonized with the group identity selection.
Although the number of users in the local batch system is significantly larger than the number of Grid users, the group identity coupled to the workgroup servers reduces the maintenance work for resources and can be handled similar as in the Grid case. With respect to the batch system monitoring and management, using only one central scheduler for the local batch users reduces the need to aggregate accounting information from a number of scattered schedulers while still allowing to scale new schedulers if needed.

5. Operations and Monitoring

To suspend job starts on a worker node in case of local problems, we adapted a common script [15] to regularly check the worker node's operativeness. Complementarily, various checks are regularly run per node testing the general node status, e.g., partition usage or network performance, as well as specific tests, e.g., the states of HTCondor daemons, number of recent job starts, CPU utilization by batch jobs or the CVMFS health. On the more central servers, as the ARC CE and schedulers or the collector/negotiator server, tests are run correspondingly, e.g., the Grid BDII/LDAP status or the states of daemons. To test the whole process flow, we send regularly test jobs to each scheduler. All tests results are aggregated in a central Icinga instance with visualisations in a Grafana installation [16] (see Figure 4).
If an alarm is raised for the basic services, the DESY computing operations group is notified and handles commonplace problems. If serious problems arise, alarms are escalated to the system administration or the Grid computing group depending on the affected service.
For job monitoring we investigated collecting statistics in near real-time from each job's cgroup information (see Figure 5). However, this approach did not scaled well to several thousand jobs. Thus, we decided to move to locally aggregating a job's resource usage statistics over its lifetime. We plan to add a job's aggregated distribution to the job summary as well as feed it into the monitoring system.
To apply new kernels or other maintenance works requiring to reboot a worker node, we use HTCondor's ClassAd mechanism to coordinate a rebooter daemon.

6. Resource Migration

Gaining experience with the HTCondor batch systems, worker nodes were subsequently migrated from the PBS/MySched cluster into the new pool over the year 2016 and newly acquired computes being added directly to the new pool (see Figure 6). As of end 2016, the majority of existing Grid resources were migrated to the HTCondor batch system with new nodes added as well.
To keep support for smaller Grid user groups, the previous Grid batch system is being operated with legacy resources, while major HEP Grid users are evacuated and migrated to the consolidated system.
The migration of the local batch system is planed for 2017 with users as well as resources moved to the new system consecutively.

7. Outlook

Both user communities, Grid and local batch system users, have somewhat different requirements. As the main Grid users, the HEP experiments prefer a larger throughput and can accept latencies in their job start-up and response rate with production tasks up to several days. In contrast local batch system users tend to run day to day cycles with only a maximum of a few thousand jobs per task and prefer responsiveness over an overall large throughput. Thus, Grid jobs saturate their available resources constantly, while batch system users show more fluctuating use patterns.
How well both requirements and use patterns can coexist can only be evaluated when in addition

Figure 4. Operations view on the overall worker node health and utilization in Grafana.

Figure 5. Near real-time job monitoring and history from cgroup statistics, that is being replaced by aggregating job statistics locally due to scaling problems.

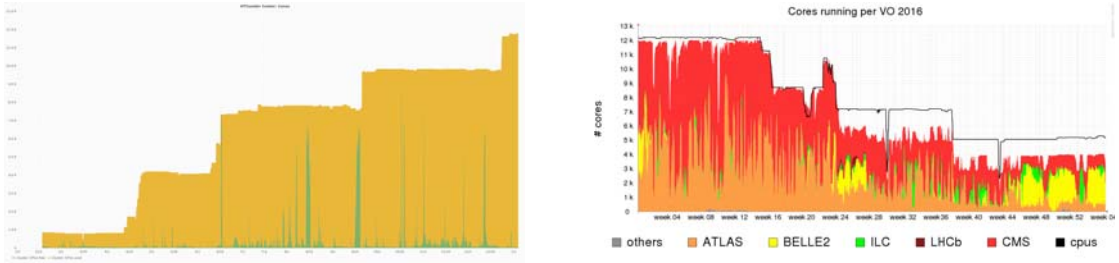

Figure 6. Development of the number of available compute slots in the consolidated HTCondor cluster and the dedicated Grid cluster over the year 2016.

to the Grid users a significant part of the local batch system resources as well as users has been migrated into the HTCondor pool. We expect that resource match making could benefit from the various jobs with the Grid jobs providing a continuous base workload and local batch system user jobs providing more statistics, allowing a faster response compared to the relatively uniform Grid tasks.

Depending on the overall system performance, we consider quota surplus flows from one into the other user community. Here, we have to avoid blocking resources by long-running and relatively static Grid workflows that could affect the responsiveness as preferred by the local batch system community. Conversely, we have to exclude harming the Grid section from individual users with limited computing experiences overstraining the batch system.

One option we will investigate is to allow to some degree sharing quota surplus between both communities where dedicated preemptable jobs are used to fill slots from unclaimed local batch system resources. If the nominal local batch system would suddenly be requested, preemptable jobs could be evacuated faster than long-running Grid jobs and thus combine a continuous utilization with a high turnover rate.

HTCondor has proven very flexible and powerful in its configuration possibilities. In general, we expect to better serve the contrasting communities with one large, united infrastructure.

References

[1] T. Finnern "Status of DESY Batch Infrastructures", HEPiX Fall 2015 at Brookhaven National Laboratory BNL (2015)
http://pubdb.desy.de/record/275834/files/DESY_Th. Finnern_BatchInfrastructures.pdf
[2] A. Gellrich: "Job Scheduling in Grid Farms" (paper, poster (14-18 Oct, 2013), poster, (CHEP 2013, Amsterdam, The Netherlands) (published as: A Gellrich 2014 J. Phys.: Conf. Ser. 513 032038)
[3] Son of Grid Engine https://arc.liv.ac.uk/trac/SGE/
[4] Andreas Haupt, Y. Kemp: "The NAF: National Analysis Facility at DESY", J.Phys.Conf.Ser. 219 (2010) 052007
[5] CREAM (Computing Resource Execution And Management) Service
https://wiki.italiangrid.it/CREAM
[6] HTCondor homepage https://research.cs.wisc.edu/htcondor/
[7] D. Thain, T. Tannenbaum and M. Livny: "Distributed computing in practice: the Condor experience, Concurrency" - Practice and Experience, volume 17, number 2-4, year 2005, pages 323-356
[8] Scientific Linux https://www.scientificlinux.org/
[9] CentOS 7 https://www.centos.org
[10] IBM Spectrum Scale technology (previously GPFS)
http://www-03.ibm.com/systems/storage/spectrum/scale/index.html
[11] dCache *https://www.dcache.org/*
[12] OpenAFS http://www.openafs.org/
[13] CVMFS https://cernvm.cern.ch/portal/filesystem
[14] ARC Compute Element http://www.nordugrid.org/arc/ce/
[15] GridPP: Workernode Health Check Script
https://www.gridpp.ac.uk/wiki/Example_Build_of_an_ARC/Condor_Cluster#Workernode_Health_Check_Script
[16] Grafana http://grafana.org/