Dynamic Resource Allocation with the arcControlTower

Andrej Filipčič[1], **David Cameron**[2], **Jon Kerr Nilsen**[2], **for the ATLAS Collaboration**

[1] Jozef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
[2] University of Oslo, P.b. 1048 Blindern, N-0316 Oslo, Norway

E-mail: `andrej.filipcic@ijs.si`

Abstract.
Distributed computing resources available for high-energy physics research are becoming less dedicated to one type of workflow and researchers workloads are increasingly exploiting modern computing technologies such as parallelism. The current pilot job management model used by many experiments relies on static dedicated resources and cannot easily adapt to these changes. The model used for ATLAS in Nordic countries and some other places enables a flexible job management system based on dynamic resources allocation. Rather than a fixed set of resources managed centrally, the model allows resources to be requested on the fly. The ARC Computing Element (ARC-CE) and ARC Control Tower (aCT) are the key components of the model. The aCT requests jobs from the ATLAS job management system (PanDA) and submits a fully-formed job description to ARC-CEs. ARC-CE can then dynamically request the required resources from the underlying batch system. In this paper we describe the architecture of the model and the experience of running many millions of ATLAS jobs on it.

1. Payload Submission Practice

The job submission and execution instabilities experienced within the grid environment ten years ago led to the rejection of the direct payload submission practice in favor of the pilot mode submission. Although the classic batch system approach with job resource requirements known at the time of the submission has been successful elsewhere and continues to be successful in the high performance computing (HPC) world today, the pilot mode in the grid world has made many issues related to infrastructure or services instabilities irrelevant by design. Universal grid jobs called pilots are submitted to the computing elements and subsequently to the underlying batch systems. When they start execution on the worker nodes, they contact the central scheduling system to receive the job description, or in other words, they pull the jobs from the virtual organization scheduler. As a consequence, the complex middleware service infrastructure was simplified since a workload management system was not necessary any more and the overall reliability of the grid infrastructure has been greatly improved.

However, the pilot mode of submissions has a drawback which is becoming more evident today, especially for the ATLAS experiment [1], where the payloads have evolved in complexity from jobs with uniform requirements to a plethora of workloads requesting diverse resources, such as memory consumption, job duration and number of execution cores. A naive pilot model is not sufficient any more, and certainly not suitable for optimal usage of the computing resources.

In an ideal distributed world, the computing resources would be fully managed by a common universal scheduling and resource allocation system, resembling and extending the concept of the classic batch scheduler. The worker nodes would be fully allocated to the scheduler, while the permanent pilots would act as the batch system daemons and ask the central scheduler for the payload till the node resources are consumed. The central scheduler would manage the job execution order through priorities and fair-share of virtual organizations or user groups. This was never considered to be an option due to the diversity and complexity of the computing sites, nor was suitable due to administrative or political restrictions.

In distributed reality, the grid middle layer sits on top of the conventional batch systems, thus multi-level scheduling must be taken into account. The central scheduling and the site scheduling systems need to adapt to each other.

2. ATLAS Job Submission Modes

ATLAS has partially overcome the problem of diverse workloads by introducing custom queues per computing site, each serving a pilot stream of selected resource requirements. The problem with this approach is that it is manageable while the number of different payloads remains low. It certainly cannot provide a viable solution if in addition the job duration is considered as a resource requirement.

ATLAS introduced the queues tuned to specific memory, cputime, corecount consumption in the middle of LHC Run-1 to accomodate specific activities requesting higher resources than the conventional Monte-Carlo production and data processing. The ATLAS workaround was to define custom PanDA [2] queues, for example, the following queues are used at the UK Tier-1 site:

- RAL-LCG2_SL6, the default production queue
- RAL-LCG2_MCORE, the queue for 8-core jobs
- RAL-LCG2_HIMEM, the queue for jobs using 4GB of memory
- RAL-LCG2_VHIMEM, the queue for jobs using 8GB of memory
- ANALY_RAL_SL6, the queue for analysis jobs

Each ATLAS WLCG site must define at least three different queues. As a consequence, the complexity of the central scheduling system approaches a level that is impossible to maintain in the long term. The deployment of multicore queues is still not fully completed after one year of WLCG task force activity. In addition, new activities, such as detector upgrade studies, will likely demand even higher resources, requiring deployment of additional PanDA queues in the future.

3. arcControlTower

A different generic approach for ATLAS was introduced recently based on the arcControlTower (aCT) [3].

The arcControlTower was developed initially for ATLAS to serve NDGF Tier-1 [4] and associated Tier-2 sites. The distributed nature of NDGF Tier-1 for both computing clusters and more notably the distributed dCache storage pools was incompatible with a standard pilot job execution workflow. The pilot jobs usually transfer the input files from close storage to a local disk and push the outputs to the same storage after the payload execution. Remote transfers in case of NDGF would be too expensive and unmanageable if the worker nodes would transfer from a remote storage pool. In addition, some of the NDGF clusters were part of the larger shared infrastructure, such as HPC supercomputers, where installation of the grid middleware on the computing nodes was not possible.

The ARC Computing Element (ARC-CE) [5] was used to transfer the input and output files remotely while the batch jobs only executed the payload and did not spend the precious time on the worker nodes on transfers. The ARC-CE provided an input file cache to minimize the number of remote transfers. To make this work, the pilot model (Figure 1) needed to be adapted so that a fully defined job was submitted to ARC-CE to prepare the input files in advance of the batch job submission, as illustrated in Figure 2.

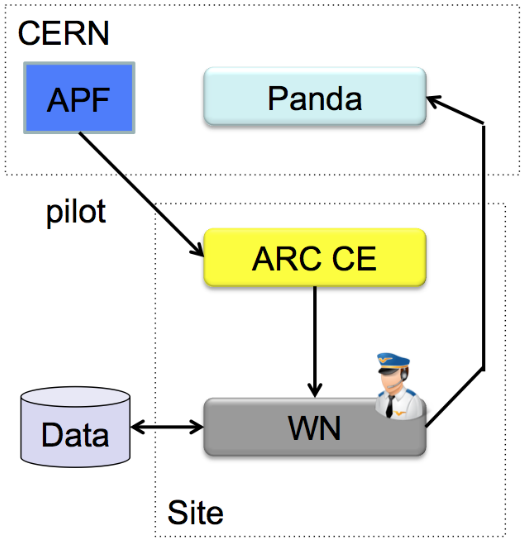

Figure 1. Worker node pulls a payload from PanDA

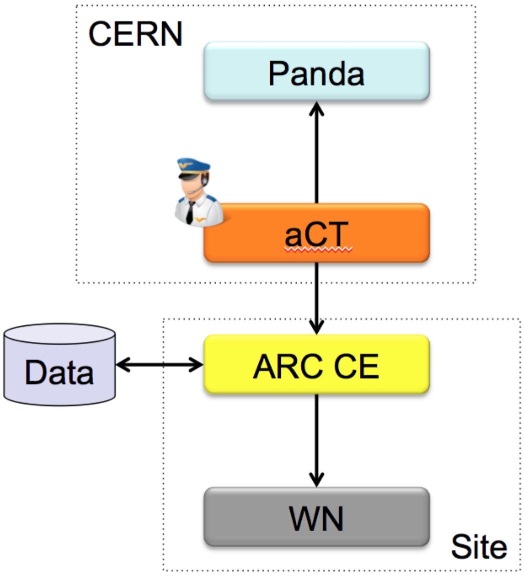

Figure 2. Payload is pushed to the node through intermediate service aCT

The arcControlTower can submit ATLAS jobs in two distinct ways. The first one, the ARC Native mode, is used to separate the job execution part from the file transfers and external communication to the PanDA service:

- aCT communicates with PanDA and submits predefined payload to ARC-CE
- ARC-CE transfers input and output files and submits to the batch system
- Pilot wrapper on worker nodes only executes the payload without accessing the external network, although outbound connectivity is still used for CVMFS [6] and Frontier [7] access
- ATLAS batch job does not use the grid middleware, it can execute on minimal operating system installations

The Native mode is optimal for sites with a capable shared filesystem which caches the input files. It also fits well the High Perfomance Computing sites with restricted connectivity, where the ATLAS software is installed locally on the shared filesystem instead if CVMFS cannot be configured due to site restrictions. This mode has been in production for ATLAS for 8 years serving the ATLAS sites associated to NDGF Tier-1.

The second mode of job submission, the aCT Truepilot mode, has the functionality very similar to the ATLAS Pilot Factory (APF):

- aCT fetches the payload and submits it to the ARC-CE, similar to the ARC Native mode
- ARC-CE submits the batch job with predefined payload
- the pilot on the worker node performs the same operations as on the conventional pilot sites, but skips pulling the payload from PanDA since already present

This mode of submission therefore sits somewhere in between the pull and the push mode, the payload is being pushed while the rest remains the same as in the pull mode.

The workflow of the Truepilot mode is shown in Figure 3 where the differences to the APF pilot mode are marked green.

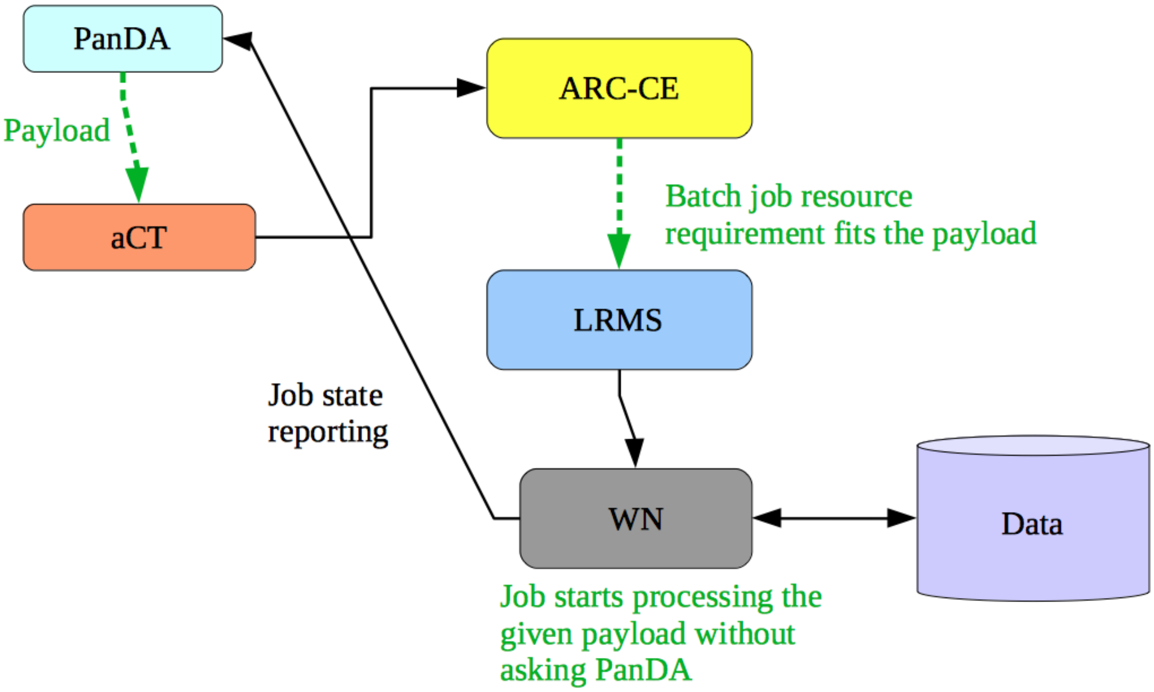

Figure 3. The arcControlTower Truepilot mode of ATLAS job submission. The differences to the pilot submission mode are marked with the dashed lines.

Comparing the APF and aCT Truepilot submission, the latency of execution is minimal for the first. When the pilot asks for the payload, the highest priority job starts execution immediately, but all the batch jobs have uniform resource requirements. The Truepilot mode however knows the job requirements for the given payload in advance, so the corresponding batch job resources, such as memory, cputime and core count, are reserved dynamically on per-payload level. This provides several simplifications and benefits,

- the same PanDA queue can serve payloads with different resource requirements
- the batch system can place a mixture of memory-heavy and memory-light payloads to best fit resources of a given node
- jobs with short walltimes can backfill the nodes draining for a multicore or a foreign big parallel job execution
- short analysis jobs could gain more computing slots on opportunistic resources of a given site since they can be drained quickly when claimed by the resource owner

The latter is essential for efficient preemptive usage of idle computing nodes on supercomputing sites which can provide extensive resources to ATLAS for short production multicore jobs.

There are disadvantages of the aCT Truepilot mode as well. The late binding of the ATLAS payload to a pilot in execution is partially lost, the payload needs to wait for some time in the batch queue, although the waiting time can be reduced to a bare minimum by keeping the waiting queue as short as possible, typically on the level of 15% of the number of running ATLAS jobs. The user job priority has been recently introduced in ARC release 15.03, which reorders the execution of the batch jobs of the specific user according to the given priorities, thus the execution order can be preserved even for out-of-order payload submission. The highest priority ATLAS payloads can thus be executed with the batch system latency of the order of minutes.

The aCT Native mode has been successfully used for many years in Nordugrid ATLAS sites and in the last year on several HPC sites as well, where the pull mode is forbidden by the site policies. For the HPC sites, even installing a custom service on site is difficult, so the ARC-CE was enhanced with an ssh-enabled backend which can transparently submit and monitor the batch jobs over an ssh connection and use the HPC shared filesystem either through sshfs or directly through libssh [8]. The HPC sites in Europe (SuperMUC, Hydra and CSCS Piz Daint) and a site in China (Shanghai PI) are fully integrated in the ATLAS production system through aCT Native mode [9, 10].

Past experience with ATLAS job execution and measurements of their resource usage already provides precise job requirements information for all the ATLAS payloads. In addition, a small subset of jobs of a given task, the scout jobs, probes for the memory and cputime consumption, so the bulk of the task payload can be submitted with matching resource requirements. Both Native and Truepilot aCT modes of submission can use the available computing resources much more efficiently, especially in case of payloads with diverse requirements.

The Truepilot mode has been used at the LRZ-LMU Munich Tier-2 site for three months and is being tested with a smaller amount of jobs at the RAL Tier-1 site. The amount of PanDA queues serving LRZ-LMU has been reduced, all the custom high-memory queues have been removed as they have become obsolete. The new submission mode is best suited for sites where modern batch systems such as SLURM [11] or HTCondor [12] with advanced resource reservations and cgroups job limits are deployed.

4. Conclusions

The arcControlTower is a flexible service providing ATLAS job submission mechanisms to computing sites which would otherwise be unusable for ATLAS production due to the limited architecture of the common pilot submission model within the standard WLCG site infrastructure. The aCT Native mode enables ATLAS job execution on sites with non-standard infrastructure, such as HPC sites or clusters accessing remote storage, or platforms difficult to integrate in grid infrastructure such as the ATLAS@Home volunteer computing project using BOINC [13]. In addition, the aCT Truepilot mode can mimic the ATLAS Pilot Factory functionality to submit the payload with predefined resource requirements to sites with the ARC Computing Element. Both submission modes provide per-job dynamic resource reservations to optimally use the site computing resources.

References

[1] ATLAS Collaboration 2008 *JINST* **3** S08005

[2] Maeno T et al, on behalf of the ATLAS Collaboration 2011 *J. Phys.: Conf. Ser.* **331** 072024

[3] Nilsen J K 2015 ARC control tower: A flexible generic distributed job management framework. Proceedings of the 21st International Conference on Computing in High Energy and Nuclear Physics, J. Phys.: Conf. Ser.

[4] NDGF Tier-1 web site URL `http://neic.nordforsk.org/about/strategic-areas/tier-1`

[5] Ellert M, Grønager M, Konstantinov A *et al.* 2007 *Future Gener. Comput. Syst.* **23** 219–240 ISSN 0167-739X

[6] CernVM File System web site URL `http://cernvm.cern.ch/portal/filesystem`

[7] Frontier web site URL `http://frontier.cern.ch/`

[8] Sciacca F G et al, on behalf of the ATLAS Collaboration 2015 The ATLAS ARC ssh back-end to HPC. Proceedings of the 21st International Conference on Computing in High Energy and Nuclear Physics, J. Phys.: Conf. Ser.

[9] Hostettler M et al, on behalf of the ATLAS Collaboration 2015 ATLAS computing on the HPC piz daint. Proceedings of the 21st International Conference on Computing in High Energy and Nuclear Physics, J. Phys.: Conf. Ser.

[10] Mazzaferro L et al, on behalf of the ATLAS Collaboration 2015 Bringing ATLAS production to HPC resources - a use case with the hydra supercomputer of the max planck society. Proceedings of the 21st International Conference on Computing in High Energy and Nuclear Physics, J. Phys.: Conf. Ser.

[11] Jette M A, Yoo A B and Grondona M 2002 *In Lecture Notes in Computer Science: Proceedings of Job Scheduling Strategies for Parallel Processing (JSSPP) 2003* (Springer-Verlag) pp 44–60

[12] HTCondor web site URL `http://research.cs.wisc.edu/htcondor/`

[13] Cameron D et al, on behalf of the ATLAS Collaboration 2015 ATLAS@Home: Harnessing volunteer computing for HEP. Proceedings of the 21st International Conference on Computing in High Energy and Nuclear Physics, J. Phys.: Conf. Ser.