Analysis of empty ATLAS pilot jobs

P A Love[1], M Alef[2], S Dal Pra[3], A Di Girolamo[4], A Forti[5], J Templon[6] and E Vamvakopoulos[7], on behalf of the ATLAS Collaboration

[1] Department of Physics, Lancaster University, United Kingdom.
[2] Karlsruhe Institute of Technology, Steinbuch Centre for Computing, Karlsruhe, Germany.
[3] INFN-CNAF, viale Berti-Pichat 6/2, 40127 Bologna, Italy.
[4] CERN, European Organization for Nuclear Research, Switzerland
[5] School of Physics and Astronomy, University of Manchester, Oxford Road, Manchester, United Kingdom
[6] National Institute for Subatomic Physics, Science Park 105, 1098 XG Amsterdam, Netherlands.
[7] Centre de Calcul de l'IN2P3, 43 Bd du 11 Novembre 1918, F-69622 Cedex, France.

E-mail: p.love@lancaster.ac.uk

ATL-SOFT-PROC-2017-036
02 February 2017

Abstract. In this analysis we quantify the wallclock time used by short empty pilot jobs on a number of WLCG compute resources. Pilot factory logs and site batch logs are used to provide independent accounts of the usage. Results show a wide variation of wallclock time used by short jobs depending on the site and queue, and changing with time. For a reference dataset of all jobs in August 2016, the fraction of wallclock time used by empty jobs per studied site ranged from 0.1% to 0.8%. Aside from the wall time used by empty pilots, we also looked at how many pilots were empty as a fraction of all pilots sent. Binning the August dataset into days, empty fractions between 2% and 90% were observed. The higher fractions correlate well with periods of few actual payloads being sent to the site.

1. Introduction

Central to ATLAS [1] distributed computing is the provision of resources on the Worldwide LHC Computing Grid (WLCG) [2]. A pilot model is used to provision resources where a simple job wrapper script is submitted to WLCG Computing Elements which then retrieve a job payload from the ATLAS Workflow Management System (PanDA) [3]. This late-binding pull-model enables PanDA to retain control of execution priority and improves reliability by protecting PanDA from ill-configured compute resources.

The application responsible for submitting these pilot jobs is AutoPyFactory (APF) [4] and ATLAS deploys a number of instances to provide the required scale and redundancy for reliable operation of the pilot system. APF is capable of submitting excessive numbers of pilot jobs which results in a job wrapper running on the resource without downloading a job payload. Such jobs are called 'empty pilots' and present unnecessary load on the site infrastructure. Job starts and stops are resource intensive for several core infrastructure services at sites, such as shared file systems, Compute Elements (CEs), and Local Resource Management System (LRMS) head nodes. Large numbers of very short jobs correspond to a high rate of job starts and stops, and can strain such site services beyond their normal capacities.

In this paper we quantify the number of empty pilots and summarise the amount of wallclock time used by empty pilots.

2. Pilot submission in ATLAS

ATLAS uses the AutoPyFactory application to submit and manage pilot jobs on the WLCG. About 600 resource endpoints are used at 160 sites. These are managed by 12 instances of APF to provide the required redundancy and scalability. The global site resources have a diverse number of job slots ranging from 100-10,000. APF provides a rich plugin system to moderate the number of pilots submitted to each resource and balancing the need to keep sites occupied without over-subscribing the smaller resources.

The plugins form a chain of logic taking input from the previous plugin and moderating the number of pilots submitted based of the APF configuration. The plugins currently used are listed in Table 1.

Table 1. List of plugins used by pilot factories

Ready	Checks the number of jobs ready to be run in the Workload Management Service (WMS), the number of previously submitted pilot still in idle state, and calculates the difference.
Scale	Multiplies by a factor the decision made by the previous plugin in the chain.
MaxPerCycle	Limit the maximum number of pilots to be submitted each cycle.
MinPerCycle	Limit the minimum number of pilots to be submitted each cycle.
StatusTest	Set number of pilots to submit when the WMS queue is in internal status test.
StatusOffline	Set number of pilots to submit when the WMS queue is in internal status offline.
MaxPending	Limit the number of pilots pending in the resource queue.

The pilot factories actually submit a wrapper script (a shell script) to the compute resources and this wrapper downloads the actual pilot code. The pilot code (written in Python) then contacts PanDA and requests a payload to be run. This payload is a job running one of the many workflows found in the ATLAS software framework.

3. Identifying empty pilots

We defined ‘empty pilots’ as those jobs submitted to WLCG resources which fail to retrieve a payload from PanDA and are short. The definition of short is on the order of minutes but its precise value is to be determined by examining the data. The ATLAS pilot does not persistently store information about whether it receives a job payload or not so we need to combine logs from various sources, as shown in figure 1.

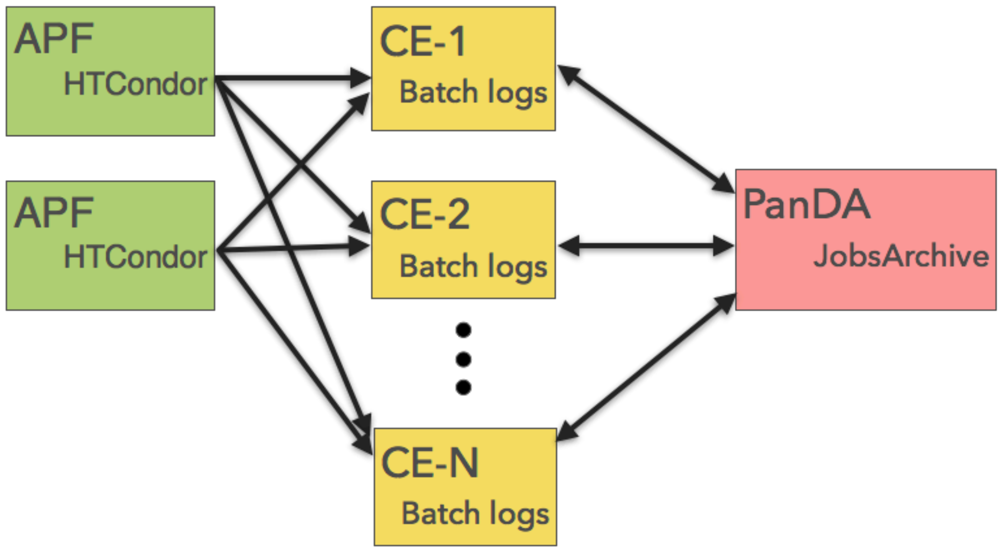

Figure 1 Source of log records from APF, Batch, and PanDA

The following four methods may be used to identify empty pilots:

1. Join APF job records with PanDA JobsArchive records
2. Join site batch records with PanDA JobsArchive records
3. Filter batch records using a CPU time and Wallclock time thresholds
4. Filter APF job records using a Wallclock time threshold

The first method is capable of tagging jobs for all sites without information from the site itself. The second method requires collaboration from the site in order to obtain batch records and also collaboration from ATLAS to provide the JobsArchive records. The third method may be used by the site without collaboration from ATLAS. The fourth method requires no information from the site.

A single calendar month (August 2016) was used to compare methods and evaluate the results across a number of sites.

4. Results

4.1. Classification of pilot behaviour

Exploring the combined data records we can identify various classes of pilot behaviour listed here:

1. jobs having short cputime, short walltime, no payload (empty pilot)
2. jobs having short cputime, short walltime, with payload (short healthy job)
3. jobs having short cputime, long walltime, no payload (bad pilot)
4. jobs having long cputime, long walltime, with payload (long healthy job)

In this study we are only concerned with the first class of job where the wrapper script passes through the compute infrastructure without processing a useful payload.

4.2. Site comparison with PanDA records for IN2P3-CC

The batch system records at IN2P3-CC were combined with PanDA job records and each job was tagged 'empty' if the batchid was not found in the PanDA database. Plotting the distribution of job duration with <1 hour wallclock time shows a clear distinction between empty and non-empty jobs (See figure 2.). The vast majority (99.5%) of empty pilots have a duration less than 60 seconds. The jobs with payload have cputime greater than 20 seconds for duration between 180-300 seconds. Therefore, we do not expect to have payload pilots with duration <180 seconds AND cputime <20 seconds. This supports the use of Method 2 as a way to tag empty pilots.

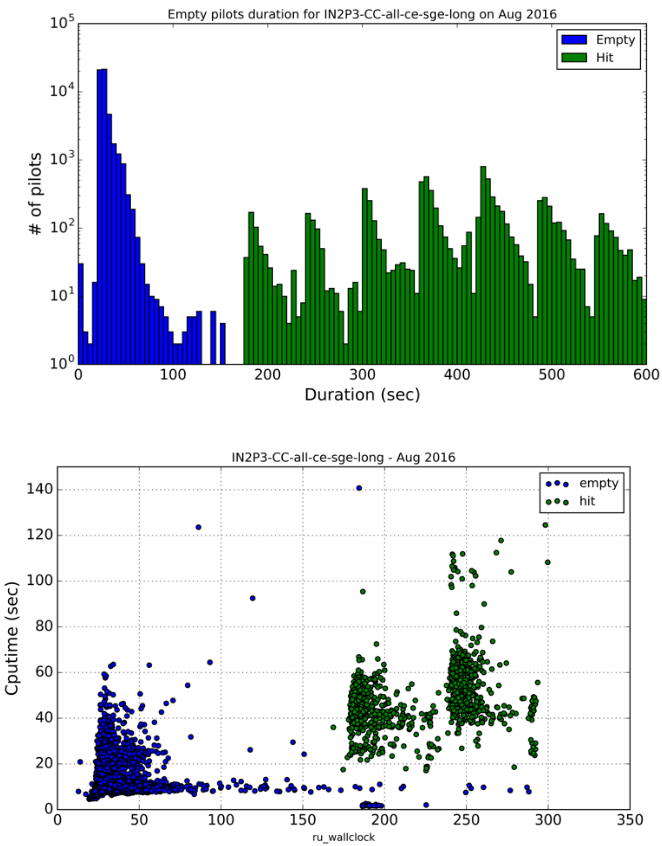

Figure 2 Distribution of wallclock and cputime for jobs run at IN2P3-CC. Empty jobs are tagged blue and jobs with payload are tagged green.

4.3. Site comparison with PanDA records for Nikhef

In a similar way to IN2P3-CC the batch system records from Nikhef were combined with PanDA records in order to identify jobs without payloads. In this case the CPU time and Wallclock time were correlated and clearly show the distinction between jobs with and without a payload. This result allows the CPU time threshold to be refined and supports the use of Method 3 as a way to tag empty pilots, but suggests that a cut on cpu time is not necessary (see figure 3).

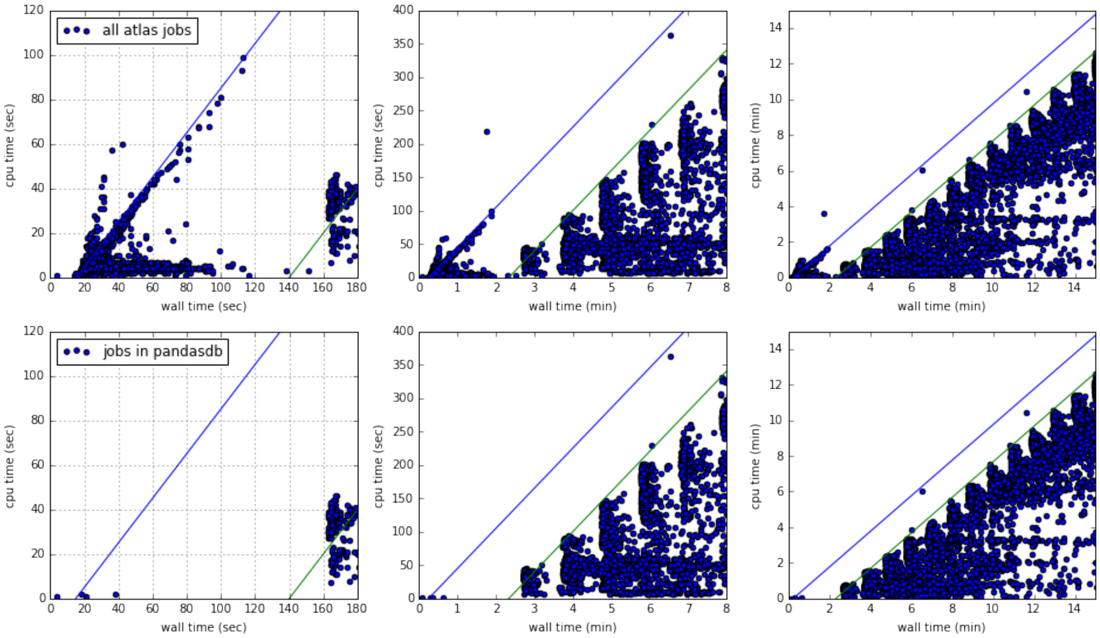

Figure 3 Correlations of cpu vs wall times for the August 2016 ATLAS dataset at Nikhef. The top panels from left to right show the dataset of all ATLAS jobs recorded in the Nikhef batch system at various time scales. The bottom panels show the same correlation, but only for those jobs found in the records of both the Nikhef batch system and PanDA. The blue and green lines follow prominent trends in the upper pane, and are replotted identically in the lower pane to aid comparison. The Nikhef+Panda (lower pane) plots are essentially empty for wall time < 160 seconds and essentially identical for wall time > 160 seconds, pointing towards "walltime < 160 sec" as a clean "empty pilot" tagging method.

4.4. Comparison of sites using APF job records

Figure 4 shows a comparison of five WLCG sites (NIKHEF-ELPROD, INFN-T1, IN2P3-CC, FZK-LCG2, MAN-HEP_SL6) where the distribution of job duration is shown for empty and payload jobs. The blue bars are empty jobs and the green bars are jobs with a payload as identified by PanDA records. The distributions have different features which illustrate the diverse behaviour at each site. In absolute terms NIKHEF and INFN-T1 have many more records tagged as empty pilots. The variation in empty job distributions between sites is expected due to the difference in available slots at each site and also the difference in workload for each ATLAS queue. The existence of empty (blue) jobs with duration >300 sec indicate this method is not clean when tagging empty jobs.

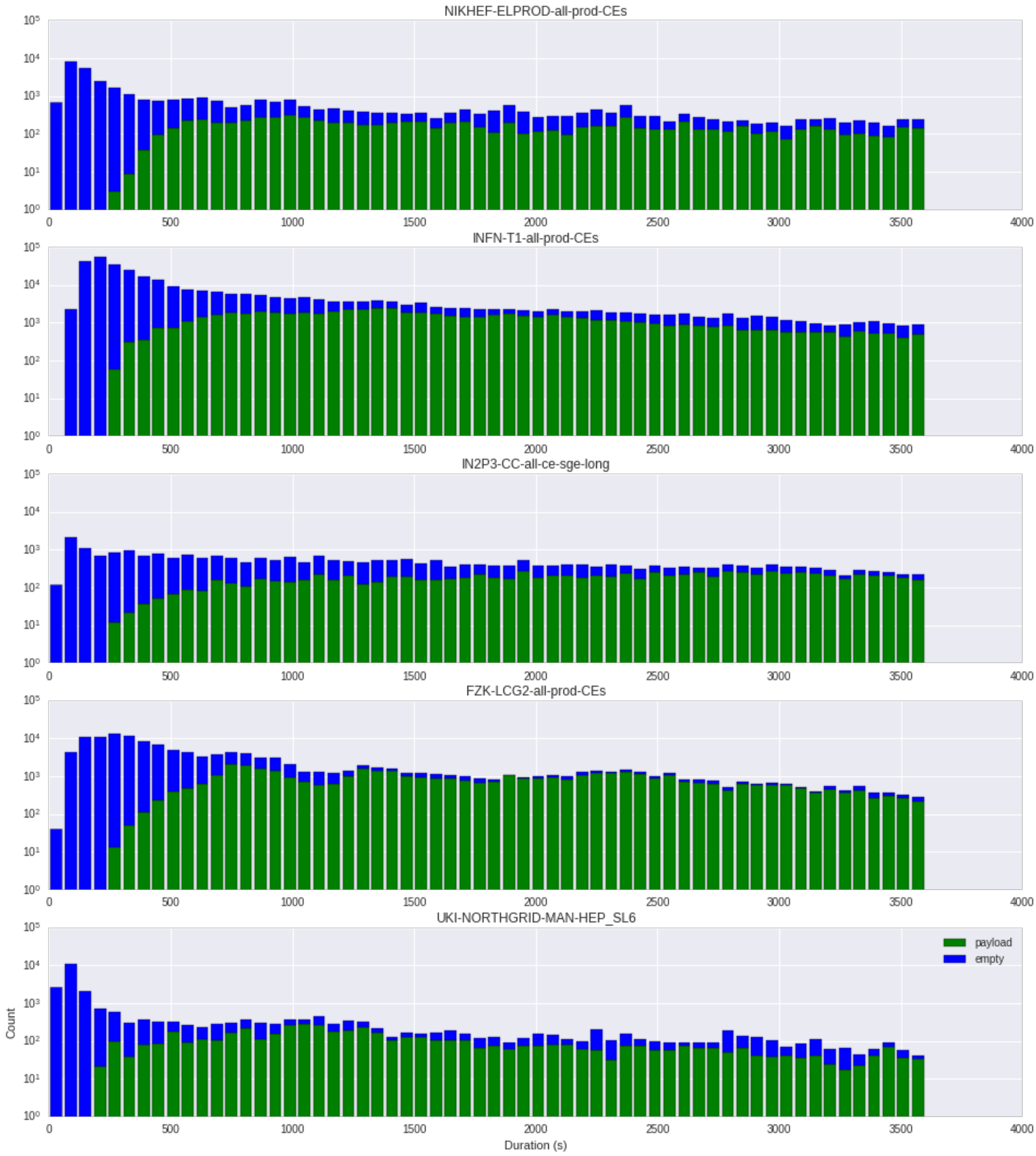

Figure 4 Job duration distribution for five WLCG sites with empty jobs tagged by Method 1 where jobs with payload are tagged if they are found in the PanDA records.

4.5. Summary of empty pilots at WLCG sites

A summary of empty pilots at five WLCG sites is shown in table 2. Site batch records were used and empty pilots were selected using Method 3 with a threshold of cputime<60s & wallclock<60s for the month of August 2016. There are wide variations between sites as expected by the different workloads assigned to the site queues. On the whole when looking at daily figures, wallclock time used by empty pilots is on the order of ~0.5%; however, the *fraction* of jobs can range from a few percent up to ~90% in number. This number is of concern to site operators because it places a load on the middleware components and in some cases can occupy job slots which would otherwise be allocated to real jobs.

This degradation in the cluster utilization is caused by the fact that it takes some time until the batch system will start the next job reusing an idle job slot. This will happen not before its next scheduling cycle (aka: scheduling run, negotiation cycle). This dead-time is not being accounted by the batch system. However, the average dead-time between a short job and the next job reusing the slot can be estimated as the difference between the average scheduling cycle time and the average wallclock time. For instance, FZK-LCG2 has estimated an average dead-time of around 1.5 minutes per short job, and a huge fraction of short jobs does indeed affect the cluster utilization.

A measurement of the average dead-time has also been performed at INFN-T1, where a self limiting mechanism has been implemented to reduce negative impact from short job bursts. The dead-time is estimated to be around 24±26 seconds [5].

Table 2 Summary statistics for daily empty-pilot fractions during August 2016

Site (August 2016 daily data)	Fraction of wallclock for short jobs (mean ± stddev)	Fraction of short jobs (mean ± stddev)
CC-IN2P3	(0.08 ± 0.11)%	(25 ± 14)%
FZK-LCG2	(0.22 ± 0.69)%	(40 ± 24)%
INFN-T1	(0.01 ± 0.01)%	(2 ± 3)%
MANC-HEP	(0.14 ± 0.13)%	(28 ± 17)%
NIKHEF-ELPROD	(0.84 ± 1.72)%	(41 ± 30)%

4.6. Reduction of empty pilots when workload is high

When there is no (or few) assigned job then the pilot factories will throttle job submission (via 'Ready' plugin), but for reasons not completely understood, a site can still experience a large number of empty pilots during such periods. An obvious candidate explanation is the difference between "pilot submission" and "pilot execution"; payload retrieval is only attempted when the pilot starts to execute. The delay between submission and execution can be many hours in some cases, so that jobs submitted during periods of high payload availability might be executed during a period of no available payload.

5. Conclusions

Results show a wide variation of wallclock time used by short jobs depending on the site and queue, and changing with time. The mean fraction of wallclock time used by short jobs over a single month can range from 0.1% to 0.8% depending on the site, plus the time of the idle gaps between every short job and the next one reusing the job slot, which are being measured by the site accounting. The variation in wallclock usage may be explained by different workloads for each resource with a greater fraction when the workload is low, but this requires further study. Aside from the wall time used by empty pilots, we also looked at how many pilots were empty as a fraction of all pilots sent. This fraction ranged from 2-40% and the large number is correlated to periods where few payloads have been assigned to the site.

References

[1] The ATLAS Collaboration 2008 The ATLAS Experiment at the CERN Large Hadron Collider JINST 3 S08003 (doi:10.1088/1748-0221/3/08/S08003)

[2] Bird, I et all 2014 Update of the Computing Models of the WLCG and the LHC Experiments Tech. Rep. CERN-LHCC-2014-014. LCG-TDR-002 CERN Geneva URL https://cds.cern.ch/record/1695401

[3] Maeno T et al. 2012 Evolution of the ATLAS PanDA production and distributed analysis system *J. Phys.: Conf. Ser.* **396** 032071

[4] Caballero J et al on behalf of the ATLAS Collaboration 2012 AutoPyFactory: A Scalable Flexible Pilot Factory Implementation J. Phys.: Conf. Ser. 396 032016 (doi:10.1088/1742-6596/396/3/032016)

[5] Stefano Dal Pra (2016) Adjusting the fairshare policy to prevent computing power loss. Proceedings of the CHEP2016 conference *J. Phys.: Conf. Ser.*