

Cluster Optimisation using Cgroups at a Tier-2

G Qin, G Roy, D Crooks, S C Skipsey, G P Stewart and D Britton

School of Physics and Astronomy, Kelvin Building, University of Glasgow, GLASGOW, G12 8QQ, United Kingdom

E-mail: gang.qin@glasgow.ac.uk, gareth.roy@glasgow.ac.uk

Abstract. The Linux kernel feature Control Groups (cgroups) has been used to gather metrics on the resource usage of single and eight-core ATLAS workloads. It has been used to study the effects on performance of a reduction in the amount of physical memory. The results were used to optimise cluster performance, and consequently increase cluster throughput by up to 10%.

1. Introduction

Beginning in 2015, the ATLAS experiment started to deliver eight-core production workloads to the sites of the WLCG. The purpose of moving to eight-core production was to significantly reduce the overall memory usage of production payloads, and to remain within the ‘1-CPU to 2 GB of physical RAM’ regime. After the introduction of eight-core payloads, UKI-SCOTGRID-GLASGOW saw an overall reduction in site utilisation, caused by CPU cores being unused when the ideal mix of single and eight-core jobs was not available. In an attempt to maximise site throughput, a study was undertaken comparing the memory allocation requested by ATLAS eight-core payloads to the amount of memory actually used by such jobs.

Accurately measuring resource usage required a technology that would not double-count shared memory, which had been previously identified as an issue by ATLAS[1]. Control Groups (cgroups) [2] is a Linux kernel feature that supports resource allocation among user-defined groups or processes. Importantly, cgroups also accurately reports the memory usage of running processes in a variety of different ways, which allows accurate usage numbers to be extracted.

To gain a deeper understanding of cgroups and to better understand the performance of ATLAS eight-core simulation payloads, a study was carried out which capped the available physical memory of jobs. This allowed the performance and behaviour of such jobs in memory-starved environments to be examined.

Subsequently, cgroups were enabled on the entire production HTCondor (hereafter referred to as ‘Condor’) cluster in late 2014. Using the existing cgroups hierarchy put in place by Condor, a system was created to collect and aggregate metrics extracted from all running batch jobs. From data collected between January and December 2015, memory and CPU usage footprints were extracted and the type of ATLAS[3] workload was identified. This data has been used to optimise the amount of memory allocated to ATLAS workloads (initially ATLAS single-core production workloads) in order to increase cluster utilisation by up to 10%.

2. Performance in a Memory-constrained Environment

A series of tests were carried out to study how ATLAS eight-core simulation workloads could be affected by constraining the amount of available physical memory. Tests were carried out on a dual Intel Xeon X5650 system with 12 physical cores (or 24 Hyper-Threaded cores), 24 GB of DDR3 SDRAM and 24 GB of available swap space on disk. The test platform was configured as a Condor worker node with a cgroup hard limit which was varied between 2600 MB and 1000 MB in steps of 200 MB. Initial results were obtained by running a single ATLAS AthenaMP eight-core simulation workload which would have requested 8 CPU cores and 16 GB of physical memory. For each workload which was run, the resident set size (RSS) memory usage, swap space usage (SWAP), total memory usage (MEMORY, equal to RSS + SWAP) and CPU usage (CPU) was extracted for the lifetime of the job. In this case, the RSS is the actual physical memory usage and similar in concept to Proportional Set Size (PSS)[4] used in batch systems like SLURM.

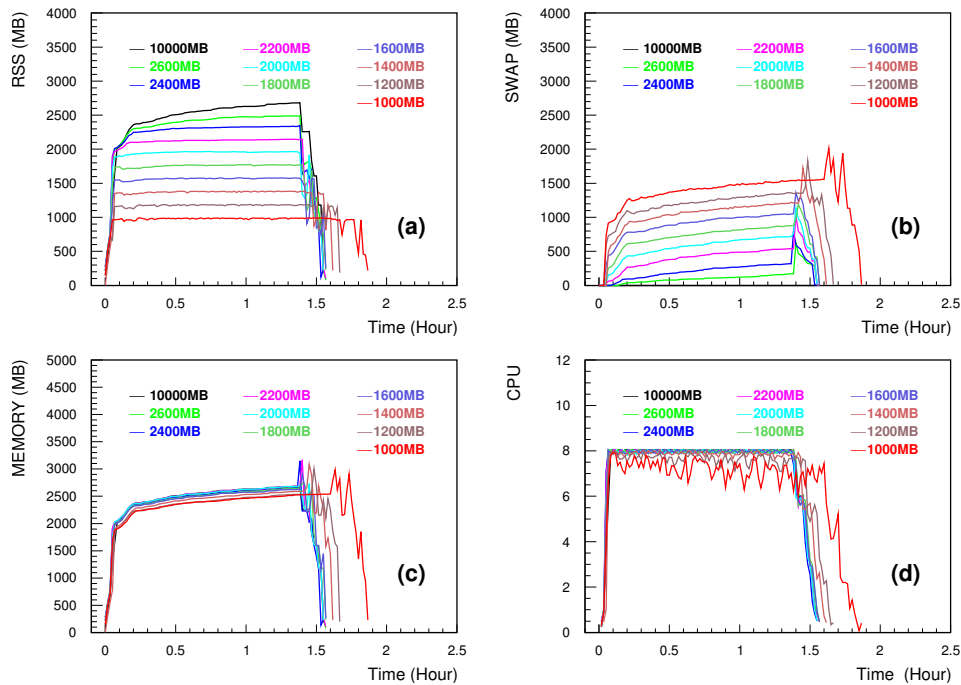


Figure 1. (a) RSS footprints of a single 8-core ATLAS simulation workload; (b) SWAP footprints of these workload; (c) MEMORY footprints of these workload; (d) CPU footprints of these workload.

Figure 1 illustrates the results obtained from this study. From Figure 1(c) it can be observed that the peak memory usage was approximately 3500 MB even though the requested memory allocation would have been 16 GB, which leads to a 4.5 times under-utilisation of memory. From Figure 1(a) it can be seen that the overall running time of the workload is not greatly impacted until the memory limit is reduced below 1400MB, and Figure 1(d) shows that below this the CPU utilisation is affected for the whole of the job's lifetime due to the need to constantly stage data from the swap partition. These results demonstrate that, for this particular payload, a 16 GB requirement for memory usage under-utilises system resources, and that moderate amount of swapping does not necessarily affect the running time of the simulation.

3. Resource Usage of ATLAS Jobs

Having examined the performance of typical ATLAS workloads on an isolated node, the next step involved investigating how adjustments to the cgroups configuration would affect the performance of our cluster as a whole. This entailed the development of tools to extract various pieces of useful information from both the local cluster and from remote ATLAS servers. Parameters for approximately 2.5 million jobs such as MaximumMemory, MaximumCpuTime, AverageMemory, and Runtime were collected during 2015.

3.1. ATLAS Eight-core Simulation

During 2015, details of about 55,000 successfully-completed ATLAS eight-core simulation jobs were collected. Figure 2(a) shows the distribution of MaximumMemory with a maximum value of 11 GB and a mean value of 5.66 GB, along with the distribution of AverageMemory which has a maximum value of 10 GB and a mean of 4.81 GB. The overall AverageCpu usage is illustrated in Figure 2(b), with the corresponding normalised efficiency being 94%. Figure 2(c) shows that the mean Runtime of these jobs is 4.3 hours. Runtimes vary up to 48 hours, but only 1.65% run for longer than 24 hours.

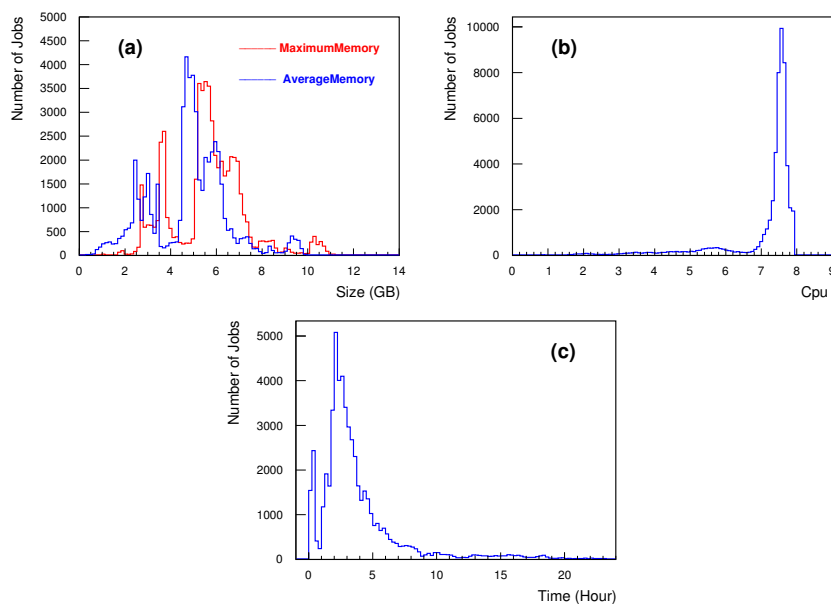


Figure 2. (a) Distribution of MaximumMemory & AverageMemory; (b) Distribution of AverageCpu; (c) Distribution of Runtime.

3.2. ATLAS Eight-core Reconstruction Job

In the same period, details of about 29,000 successfully-completed ATLAS eight-core reconstruction jobs were collected. Figure 3(a) shows that the mean MaximumMemory value is about 11 GB with a maximum of 19 GB, while AverageMemory shows a mean of 6 GB and a maximum of 14 GB. The overall AverageCpu usage is illustrated in Figure 3(b) with a calculated normalised CPU efficiency of 61%. Figure 3(c) shows the Runtime of these jobs, with a mean value of ~ 2.8 hours. In a very small proportion of jobs, a maximum Runtime of up to 48 hours was observed, but only 0.3% of jobs run for longer than 24 hours, with a peak Runtime centred at ~ 1.5 hours. At UKI-SCOTGRID-GLASGOW, all production queues have a 48-hour

maximum Runtime; given the results from Figure 2 and Figure 3, this could be easily reduced to 24 hours, or even much lower, to allow better scheduling of resources.

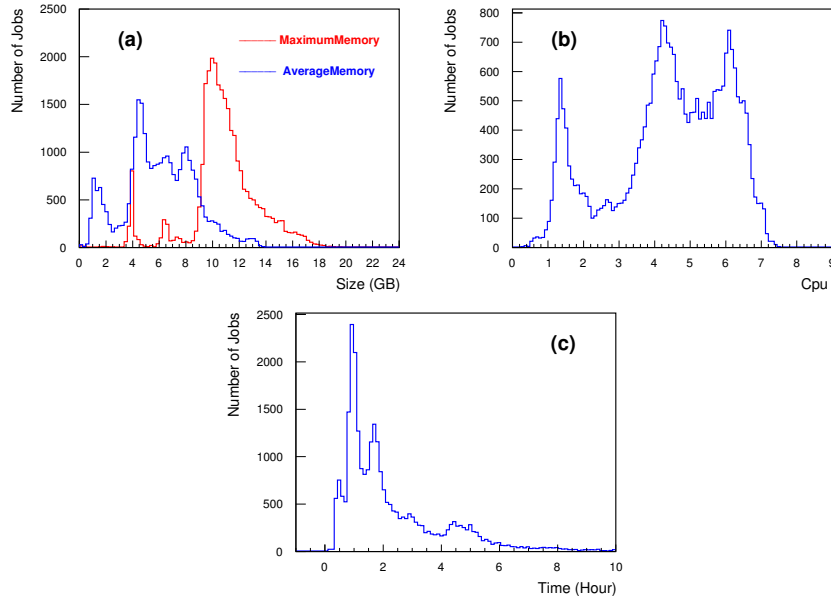


Figure 3. (a) Distribution of MaximumMemory & AverageMemory; (b) Distribution of AverageCpu; (c) Distribution of Runtime.

3.3. ATLAS Single-core Production

During 2015, details of about 294,000 successfully-completed ATLAS single-core production jobs were collected. Figure 4(a) shows that the mean MaximumMemory value is ~ 0.98 GB with a maximum of ~ 2.4 GB, while AverageMemory shows a mean of 0.83 GB and a maximum of ~ 2.3 GB. AverageCpu usage has a mean value of 0.96, as illustrated in Figure 4(b). Figure 4(c) shows the Runtime of these jobs, with a mean value of ~ 5.63 hours; about half finished within two hours, while 3.5% have a Runtime in excess of one day.

3.4. ATLAS Single-core Analysis

During 2015, details of about 463,000 successfully-completed ATLAS analysis single-core jobs were collected. Figure 5(a) shows that the mean MaximumMemory value is ~ 0.41 GB with a maximum of ~ 1.5 GB, while AverageMemory shows a mean of 0.32 GB and a maximum of ~ 1.5 GB. The overall AverageCpu usage is 0.79, as illustrated in Figure 5(b). Figure 5(c) shows that approximately 90% of jobs finished within five hours, and only 1.7% of jobs ran for longer than one day.

4. Evaluation of Memory Utilisation

Table 1 summarises the results obtained from the above studies. From Table 1 it can be seen that the amount of memory requested for ATLAS 8-core reconstruction workloads correctly estimates their requirement. However, for all other ATLAS workloads, the requested memory over-estimates the requirement significantly, and in the case of single-core analysis does so by a factor of two. In the case of analysis, this over-estimation is a function of the payloads executed at UKI-SCOTGRID-GLASGOW and may not be indicative of all other sites.

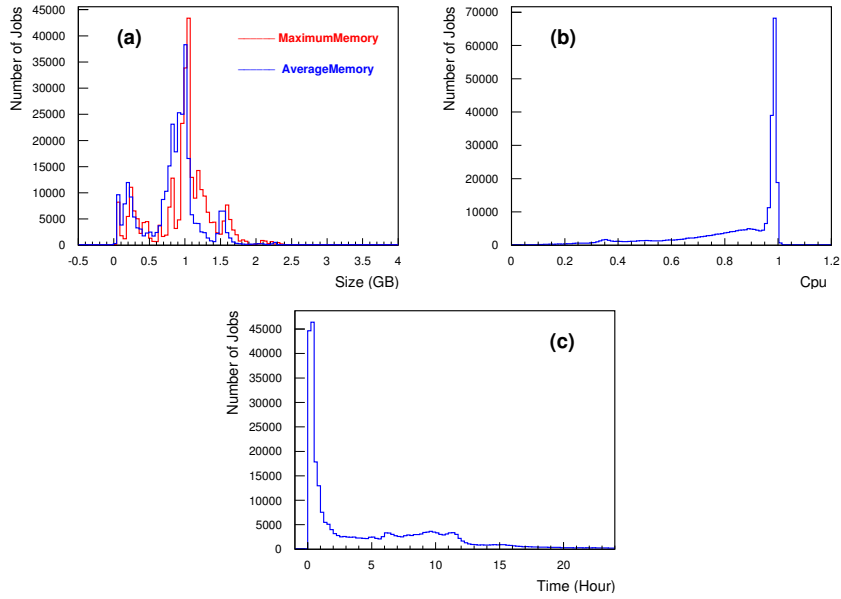


Figure 4. (a) Distribution of MaximumMemory & AverageMemory; (b) Distribution of AverageCpu; (c) Distribution of Runtime.

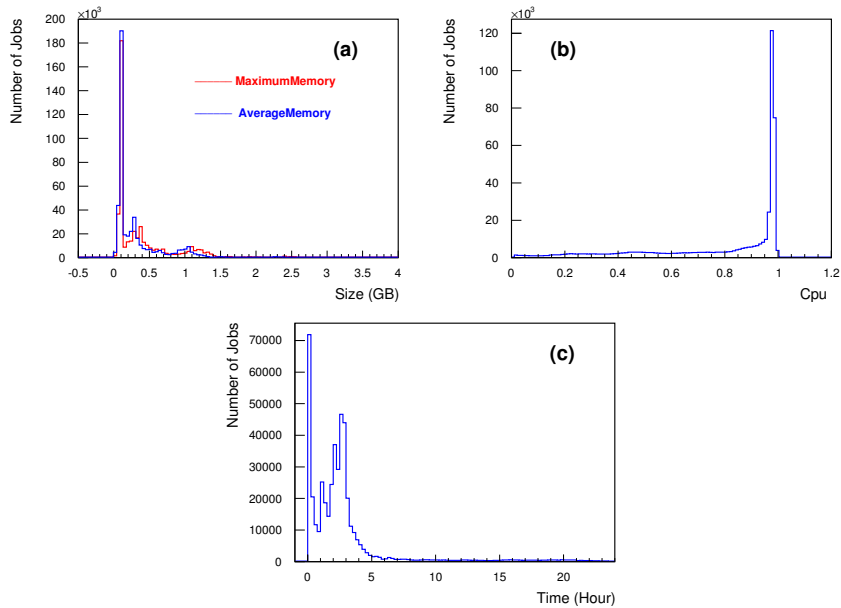


Figure 5. (a) Distribution of MaximumMemory and AverageMemory; (b) Distribution of AverageCpu; (c) Distribution of Runtime.

There are two changes that ATLAS could make to improve utilisation of system resources. Firstly, if they were to distinguish between eight-core simulation and eight-core reconstruction jobs at the queue or pilot level, an improvement in utilisation of up to 31% could be obtained. Secondly, if the memory requested for single-core jobs was reduced to ~ 2 GB, a significant increase in system utilisation could be achieved.

Table 1. Memory requested by various types of ATLAS job compared to memory actually used.

Job Description	Cores Requested	Memory Requested	Maximum Memory Used	Average Memory Used
1 8-core Simulation	8	16 GB	11.0 GB	5.66 GB
2 8-core Reconstruction	8	16 GB	19.0 GB	11.00 GB
3 1-core Production	1	3 GB	2.4 GB	0.98 GB
4 1-core Analysis	1	4 GB	1.5 GB	0.41 GB

5. Optimisation of Cluster Resources

As discussed in the introduction, with the inclusion of eight-core payloads an overall reduction in site utilisation occurred when the correct mix of single-core to eight-core payloads was unavailable. This was due to older generations of hardware not being easily quantised into units of eight cores. Assuming the data presented on memory utilisation was indicative of all existing ATLAS workloads, it was possible to better utilise local resources by modifying the amount of memory allocated to each job. At first, reducing the memory allocation for single-core jobs to 1 GB of physical memory (based on average usage) allowed all unused cores to be used, resulting in a significant increase in overall cluster utilisation. However, this reduction in memory resulted in jobs being killed due to worker nodes running out of memory and excessively swapping. In order to better utilise resources without causing failed jobs, a value of 1.8 GB per single-core job was selected (Table 2 illustrates suggestions for additional limits for other ATLAS workloads). By doing this, it was observed that the number of used cores increased to ~ 460 with only ~ 180 unused cores, representing an improvement in cluster utilisation of $\sim 10\%$. Cluster operation in this configuration was observed over a period of six months, and it was found that this setting provided the greatest improvement in utilisation without risking instability due to the overloading of machines.

Table 2. Suggested memory allocations for various types of ATLAS job, compared to requested memory.

Job Description	Cores Requested	Memory Requested	Memory Allocated	Saving
1 8-core Simulation	8	16 GB	12.0 GB	25%
2 8-core Reconstruction	8	16 GB	16.0 GB	0%
3 1-core Production	1	3 GB	1.8 GB	40%
4 1-core Analysis	1	4 GB	1.5 GB	63%

6. Conclusions

In this paper we have introduced a system used to extract memory and CPU footprints from the UKI-SCOTGRID-GLASGOW HTCondor cluster. We have examined the detailed resource usage of single and eight-core ATLAS workloads. Based on the results, the site has adjusted its scheduling policy with the result that an increase in cluster utilisation of approximately 10% has been obtained.

References

- [1] Status and Future Evolution of the ATLAS Offline Software, R. Seuster et al., J. Phys.: Conf. Ser., 664(7) 072044, 2015.
- [2] Red Hat Enterprise Linux 6 Resource Management Guide, P. Ondrejka et al., 2013.
- [3] ATLAS Computing Technical Design Report, 2005.
- [4] Formulating Optimized Storage and Memory Space Specifications for Linux Network Embedded Systems, K. Tsiligkos and A. Meliones, HPCC/CSS/ICISS, pp. 580–584., IEEE, 2014.