GridPP - Preparing for LHC Run 2 and the Wider Context

2015 J. Phys.: Conf. Ser. 664 052006

(http://iopscience.iop.org/1742-6596/664/5/052006)

21st International Conference on Computing in High Energy and Nuclear Physics (CHEP2015) IOP Publishing
Journal of Physics: Conference Series **664** (2015) 052006 doi:10.1088/1742-6596/664/5/052006

GridPP - Preparing for LHC Run 2 and the Wider Context

Dr Jeremy Coles
High Energy Physics Group,
Cavendish Laboratory, JJ Thomson Avenue, Cambridge, CB3 0HE, United Kingdom

E-mail: jeremy.coles@cern.ch

Abstract. This paper elaborates upon the operational status and directions within the UK Computing for Particle Physics (GridPP) project as it approaches LHC Run 2. It details the pressures that have been gradually reshaping the deployed hardware and middleware environments at GridPP sites – from the increasing adoption of larger multicore nodes to the move towards alternative batch systems and cloud alternatives - as well as changes being driven by funding considerations. The paper highlights work being done with non-LHC communities and describes some of the early outcomes of adopting a generic DIRAC based job submission and management framework. The paper presents results from an analysis of how GridPP effort is distributed across various deployment and operations tasks and how this may be used to target further improvements in efficiency.

1. Introduction

The UK Computing for Particle Physics (GridPP) project began in the year 2000 with a commitment to join the EU DataGrid. From this the first international grid infrastructure was created in 2001 and in subsequent years the project and resources evolved through various stages to be ready for LHC collisions in November 2009 [1]. LHC Run 1 continued until 2013. From an external perspective the resources available grew steadily, operated stably and played a role in the Higgs discovery announced in 2012. There is no doubt that the collaborative computing project formed to support LHC physics (the Worldwide LHC Computing Grid (WLCG) – of which GridPP is approximately a 12% contribution) has been a success. However, behind the stable external view, the computing hardware, supporting middleware, LHC experiment services and operating methods have been constantly evolving. This paper is intended to share a snapshot of this evolution as seen from within the GridPP project just before the LHC begins Run 2.

2. Resources

Throughout Run 1 and the LHC Long Shutdown 1 (LS1 – covering the period of early 2013 to early 2015) GridPP has increased available compute hardware to stay ahead of pledges it has made to WLCG and the LHC experiments. Figure 1 shows the aggregate increase in CPU (as measured in units of the HEP adopted HEPspec 2006 benchmark (HS06)) and figure 2 that for storage. A straightforward capacity plot hides several important changes that have taken place since the start of Run 1. Firstly the early worker nodes (WNs) typically had available 1 or 2 GB of RAM per job slot and 2 to 8 cores, and these were generally connected at 100Mbps. The machines were run in 32-bit mode under Scientific Linux 4. In discussing preparations for Run 2 it is instructive to mention the 2009 computing landscape because so many underlying things have changed and required new approaches over the years that the step change many might expect ahead of Run 2 does not appear – it has been smeared out over the proceeding few years by gradual change. As we have moved to 64-core+ nodes running 64-bit SL6 with gigabit connectivity (and site connectivity has moved to tens of gigabit per second capability), and as disk server capacities have increased, the experiment computing models have inevitably changed. Two aspects of this that will be briefly discussed later in this paper are the development of multi-core job scheduling and the increasing capabilities of virtual machine environments.

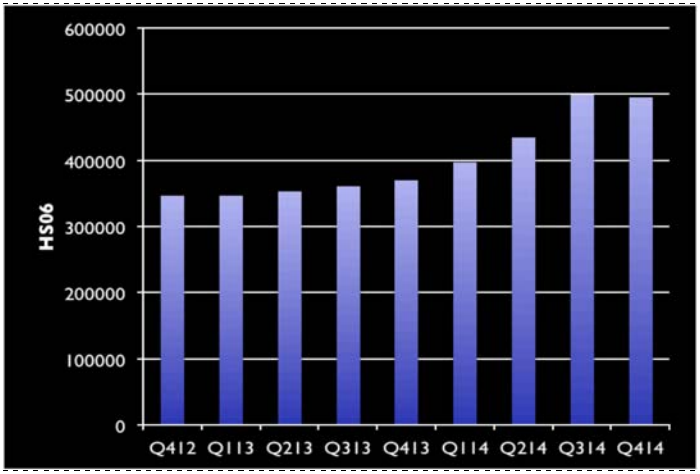

Figure 1. The deployed CPU (as measured in HS06) across the GridPP infrastructure in successive quarters. At the end of 2014 the GridPP pledge to WLCG was at the 200,000 mark. Additional resource above pledged levels was leveraged from local institutes in various ways and remained well utilised. Over pledge does not mean over provision!

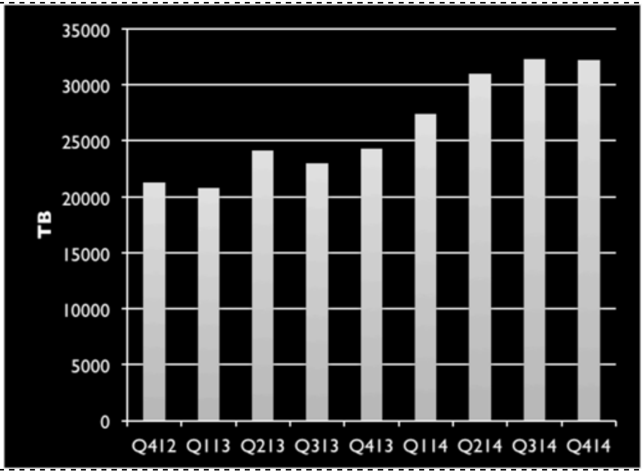

Figure 2. The deployed disk storage (as measured in TB) across the GridPP infrastructure in successive quarters. At the end of 2014 the GridPP pledge to WLCG was at the 15,000 level. As with CPU, provision above pledge was possible for various reasons and enabled more science to be undertaken.

In short, hardware provision itself for Run 2 has been business as usual, but before leaving the topic of resource provision it is worth noting that recent hardware purchases have seen CPU costs rising above WLCG long-term projections. Whilst this does not immediately affect Run 2 computing, overly optimistic cost projections are adding another pressure in the need to find significant performance gains for Run 3 and particularly Run 4.

3. Batch and Computing Element changes

From the start of Run 1 and up until 2013, the majority of GridPP sites (of which there are about twenty) ran a gateway set of EGI CREAM Computing Elements (CEs), which interacted with a Torque/Maui batch system. A couple of sites ran a GridEngine batch system. The choice was largely determined by local expertise and cost. However, as Run 1 progressed and the deployed resources at sites increased (to the level of individual sites having over 10,000 job slots) there was increasing concern about the non-existent support for Torque/Maui, and issues were seen to be increasing - particularly with Maui, which many sites complained became increasingly unresponsive, suffered memory leaks and failed to pickup correctly on new jobs and keep job farms full. At this time whole-node queues were being introduced to meet experiment needs and Torque/Maui struggled to schedule jobs effectively to these and did not cope well with emerging mixed farms based on SL5 and SL6. Although some WLCG sites worked to patch Torque/Maui it was agreed within GridPP that longer term a better-supported product needed to be deployed. Thus in preparation for Run 2 a selection of

GridPP sites explored options. In particular the Rutherford Appleton Laboratory (RAL) WLCG Tier-1 put significant effort into evaluations starting in August 2012 and explored options with LSF, HTCondor (HTC), Grid Engine and SLURM. The criteria included robustness, scalability and integration with the local and wider WLCG community environments, as well as the usual support levels, costs and functionality. On the basis of cost many commercial solutions were quickly rejected.

Ease of setup and overall stability and responsiveness quickly led to the RAL Tier-1 selecting HTC as a new batch system and simultaneously several GridPP Tier-2s also reached this conclusion. One potential barrier was HTC not being officially supported with the then in use EMI-3 CREAM CE – in particular dynamic publishing and accounting plug-ins were not available. Nevertheless some locally developed scripts bridged the gaps and deployment moved forward. The initial problems with CREAM CE integration raised the question of alternative middleware options and focus was quickly put on compatibility with ARC CEs (developed within Nordugrid) which were starting to gain favour in GridPP sites due to their simplicity (unlike the CREAM implementation they do not require MySQL and Tomcat to be installed, have easier configuration and have a straightforward route to publishing accounting records for central consumption). Since by this time the majority of GridPP user communities could submit jobs to ARC CEs (ATLAS and CMS using HTCondor-G to submit pilot jobs, LHCb via a submission mechanism in DIRAC, and non-LHC Virtual Organisations (VOs) via the WMS) the move had few barriers. The outcome of this is that as Run 2 approaches, about half of GridPP's sites run HTC as a batch system with ARC as their CE of choice, about five run a variant of Gridengine based on wider local considerations, two have SLURM, and the reminder (the smaller sites) are preparing to switch mainly to HTC.

4. The multicore problem

During 2013 it became apparent that for some (and an increasing number of) workflows the LHC experiments benefitted from running jobs across multiple cores in the same worker node. It was clear that the trend on the Run 2 timescale was for an increasing number of jobs to be run this way. The challenge for sites has been to maximise CPU usage while catering for mixed workloads.

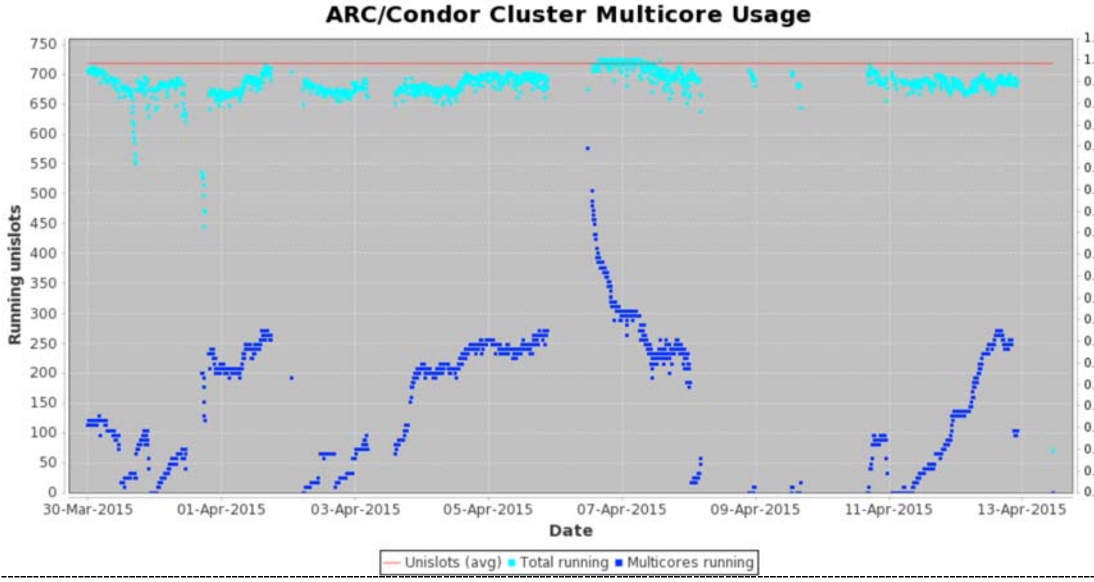

Figure 3. A plot showing the early promising results of an HTC/ARC drain rate controller as implemented by Stephen Jones at the Liverpool GridPP Tier-2 site. The target of filling 250 unislots (seen around 1st, 5th, 7th and 12th April) is made difficult by shortages of jobs.

GridPP was one of the leaders in exploring the options for integrating the scheduling of multi and single core jobs by first examining the experiment usage models (ATLAS runs separate multi and single core jobs with one payload per pilot job, whilst CMS combines both job types in the same pilot), exploring batch system capabilities (for example options for dynamic partitioning) and finally looking for ways to optimise scheduler tuning.

To run multicore jobs on non-dedicated resource requires CPU 'draining' – freeing up CPUs so they are all available for a multicore job to start. If a high-priority single core job arrives the draining can be stopped and the unused CPU cycles wasted for no benefit. Scheduling low priority jobs that will finish before a reservation start time provides a more efficient filling of job slots, but this 'back-filling' option is hard to achieve in practice because it not only requires a steady job supply but also predictable run times (but these vary with i/o needs, event complexity and so on). Back-filling is more successful where job parameters can be passed between the batch systems and jobs (parameters such as memory available, core count etc.) but investigations have shown different batchsystem/CE implementations make this tricky (for example virtual memory is used and reported differently – with multicore the shared memory is allocated multiple times and this has required recipes to be developed for managing the over-subscription). Multicore jobs have also required adaptation of accounting clients, the database schema and cpucount publishing.

Some GridPP sites are currently exploring more active approaches to maximising CPU efficiency. Figure 3 shows early results at Liverpool based on an ARC CE and HTC implementation called DrainBoss [2]. The python script encapsulates a proportional integral (PI) controller with conditional logic (to reduce negative corrections) that works to maintain a desired ratio between single and multicore jobs (it can be used in place of the HTCondor DEFRAG daemon which is an embedded solution that can periodically drain down nodes to allow multicore jobs to start). DrainBoss uses parameters such as the 'maximum number of machines to drain concurrently' or 'within a given time slot', and the 'total number already drained' to give feedback to the system. The results at the time of writing are encouraging and suggest further tweaks for better performance, but the testing is often interrupted. In the plot the dark blue dots show the multicore jobs running and the light blue the total jobs running. The drop around 2nd April was due to a shortage in both single and multicore jobs, the rapid rise in multicore slots on 6th April was caused by a shortage of single core jobs, and the lack of results around 9th April was caused by an unrelated problem with the site BDII. Given these early results further work is being done to include error handling, introduce a ramp up function, integrate it better with the Condor internal data structures and conduct more systematic testing and tuning.

5. Diversification of resource provision

Back in 2009 at the start of Run 1, there was a lot of momentum behind the provision of resources via what is still commonly seen as the Grid Middleware stack. The experiments generally submitted jobs via a Workload Management Server (WMS) to a CE, and the Worker Nodes (WNs) had to have local access to tagged experiment software and run a compatible operating system, which was usually Scientific Linux. The use of Virtual Machines (VMs) to quickly deploy and run services has matured somewhat in the last couple of years and now offers a much more flexible approach to resource provision as well as running services more flexibly. There have been other drivers to further explore alternative deployment and integration routes including expectations (outside communities including funding bodies have bought in to 'cloud' technologies more than 'core' grid technologies); support needs (there is a desire to adopt more mainstream supported solutions to reduce infrastructure risks); flexibility (many non-HEP communities are building their computing strategies around cloud offerings) and simplicity (finding middleware that is simpler to install, configure and maintain).

During LS1 a number of GridPP sites (and others), have worked with the LHC experiments to demonstrate the capabilities of, and gain experience with, a number of standard VM providers including: OpenStack; CloudStack; OpenVZ; OpenNebula and HTCondor. Each of these is now running within GridPP's infrastructure. Most are also now able to exploit the widespread adoption of the CernVM File System (CVMFS) which utilises nearby Squid caches to cascade experiment software in a fast, scalable, and reliable way [3]. For VM lifecycle management (that is to create the types of VM specified by the experiment, check VM health, respond to workloads according to some sharing policy etc.) the main standard tools used have been CloudScheduler and Condor Vaccum. The VM approach to WNs has given the experiments a very welcome ability to easily customise their environment and the work is being built upon to explore potential use of commercial offerings.

One interesting finding of the VM work is that the most widely adopted Cloud solutions, like OpenStack, are just as complicated to setup and configure (if not more so) than the classic Grid Middleware and come with additional multiple layering and queue complexities. Within GridPP this has led Andrew McNab at Manchester to come up with an alternative (Infrastructure-as-a-Client) approach called Vac [4]. This development acknowledges that the experiments already have a pilot framework to allocate jobs and so the implementation can therefore be stripped down to a state where the physical host creates the VMs for itself. A Vac VM factory daemon runs to create and supply contextualisation data to transient VMs that appear spontaneously "out of the vacuum" at sites. All that is needed is a boot image (uCernVM 3 images are only 20MB and can be pulled in via CVMFS) and a site-wide accessible user data file containing the contextualisation information. The only library dependency for the installation is libvirtc. The approach completely removes the need for site gatekeeper services, batch system head nodes, an information index and accounting services. As GridPP looks to the future and the need to run sites with decreasing manpower, it is exploring options for a "Vac-in-a-box" (ViaB) solution whereby a Vac installation can be installed in minutes from a USB memory stick – the underlying machines only requiring DNS from the site. Vac, Squid, dhcpd and tftpd are all wrapped up inside each Vac machine. Alongside VAC, Vcycle [5] was developed to manage the VMs across IaaS Cloud services like OpenStack.

In summary, as Run 2 approaches, GridPP is ready to provision resources under alternative routes as and when policy and strategy require it. In the meantime, the collaboration will continue developing experience with VM and container (for HEP workloads GridPP has found containers provide near native execution whereas VMs suffer a 15-20% reduction in performance) [6] deployments as well as undertake a wider deployment of Vac and Vcycle.

6. Working with non-LHC VOs

The Grid Middleware approach to resource provision has served WLCG and GridPP well over the years, but it has presented a number of barriers to non-LHC VOs (and increasingly so as the LHC VOs have developed in-house solutions to meet their job and data management needs and consequently reduced their grid middleware support). A very clear benefit coming from the work to look beyond the Grid Middleware approach, and especially in running jobs within VMs and the ability to fetch machine images remotely, has been the ease with which these approaches have been generalised to simplify the access to GridPP resources for other communities. Central to the approach now being widely used within GridPP is the 'interware' provided by the Distributed Infrastructure with Remote Agent Control (DIRAC) project [7]. The DIRAC Workload Management system enables task scheduling through the use of Generic Pilot Jobs and provides portal based monitoring, control and configuration. By providing a GridPP VM (which together with VO software can be pulled in via CVMFS), GridPP has been able to offer non-LHC VOs like SNO+ and T2K (along with many non-HEP VOs) the simplicity and power afforded by a centrally run pilot job framework. In the process GridPP has contributed various code patches and introduced new agents needed for administering a DIRAC server running for multiple VOs.

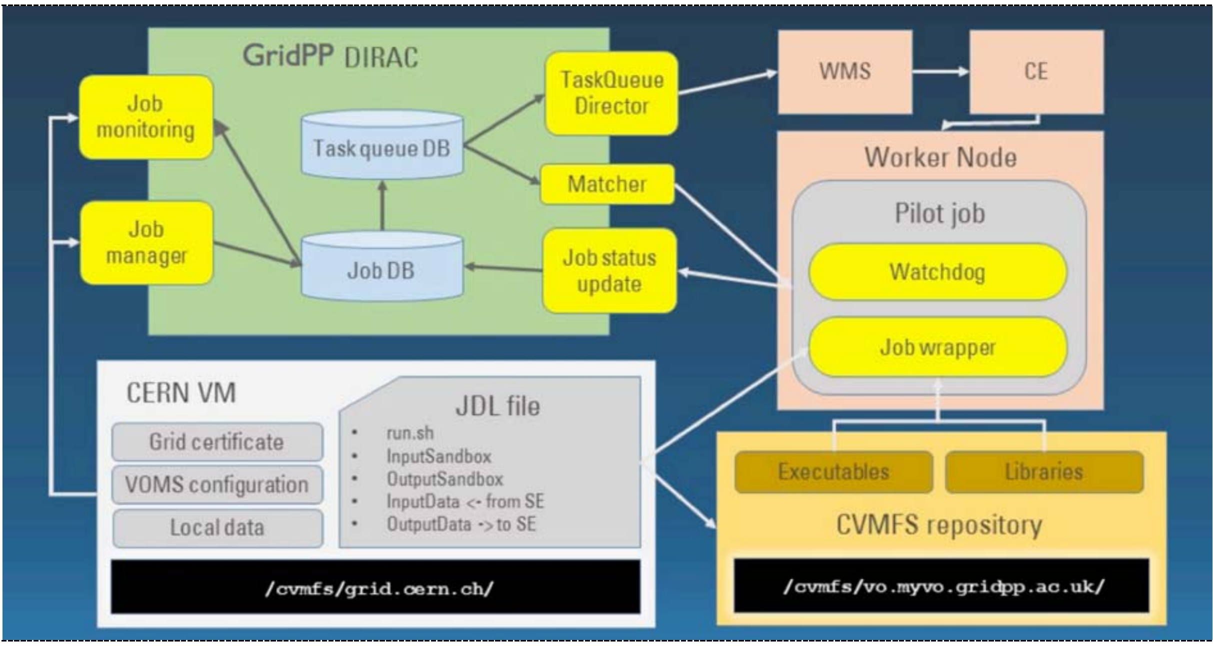

Figure 4. A high-level view of the components and their interactions as implemented by GridPP for its non-LHC user communities. The diagram and approach were derived in conjunction with Tom Whyntie at QMUL who applied them to the CERN@School use case [9].

7. Site and infrastructure provider tasks

GridPP has recently had to submit a funding proposal for the continuation of the project into a 5th phase (GridPP5). Central to this proposal has been the need to explain and justify work that is undertaken within the context of the European Grid Infrastructure (EGI), WLCG and on a regional basis. It was decided that the best way to do this would be to carry out a bottom up analysis that accounts for the just over 50 Full Time Equivalents (FTEs) of effort. The first step in the analysis was to capture all work undertaken under a clearly defined task area – this was easiest for EGI where core tasks were already identified and committed effort accounted. The second and subsequent steps required a full review of activities carried out by each staff member and an estimate of the percentage time they spent on various contributions. By further estimating effort contributed by WLCG partners it was possible to show that the UK contributes a proportionate share of the effort required to perform the international tasks required to run WLCG. The task results are shown in Figure 5.

The results suggest that about half of the effort within GridPP goes into the Tier-1 and Tier-2 functional services (figures 6 and 7 look at these in more detail), and the rest go into a large number of activities that support the infrastructure in operation and evolution. This is significant because it indicates that there is a clear need for effort beyond running site hardware and a clear case needs to be made for this effort. For example an estimated 5% of effort is expended in keeping up with and catering for technological changes (such as evolving the batch systems, investigating alternatives like CEPH and preparing for IPv6) while another 5% is required to maintain a secure and well-monitored set of services. Another clear finding of the study was that GridPP and WLCG benefit significantly through the leveraging of additional effort and physical resources possible through maintaining many 'small' sites. However, there is a balance to be made in terms of monitoring and support overheads for the experiments. Several potential ways towards savings for GridPP have been identified ranging from increased partnering with other communities to share costs in common areas, to reducing Tier-2 effort by implementing more lightweight approaches such as Vac. The GridPP average Tier-2 effort is

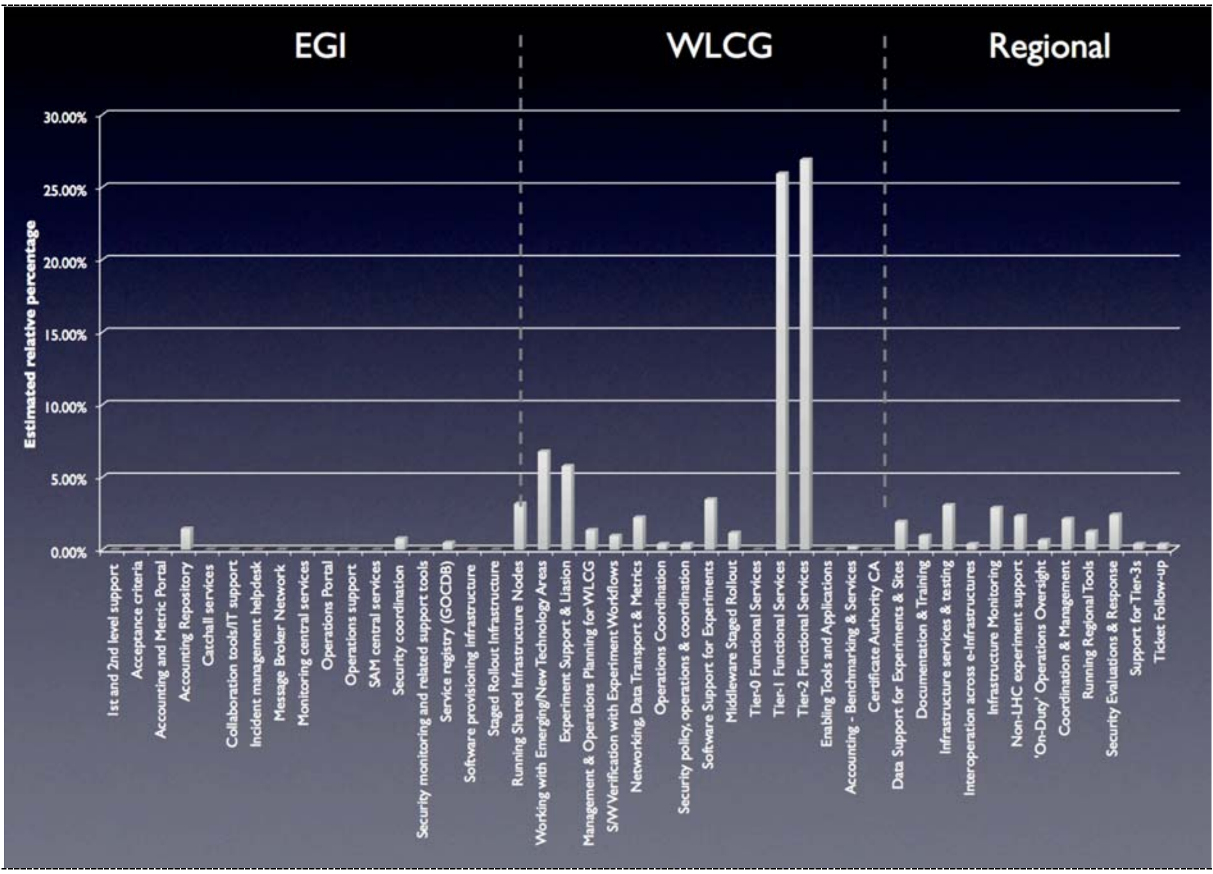

Figure 5. Infrastructure tasks as captured by GridPP in February 2015 and the *estimated* percentage of GridPP effort that goes into each. All the tasks are needed to run the current infrastructure.

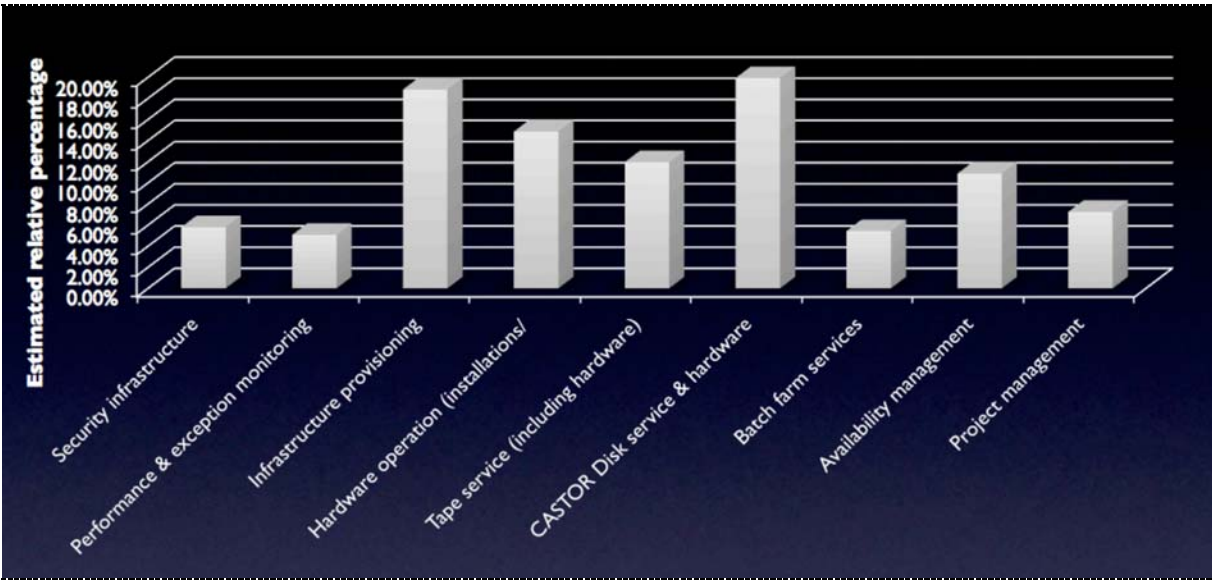

Figure 6. An indicative breakdown of GridPP Tier-1 tasks and the relative effort that goes into providing each task. Increased economies of scale to dilute the infrastructure provisioning costs falling only on GridPP and improved disk deployment and maintenance are possible areas for savings.

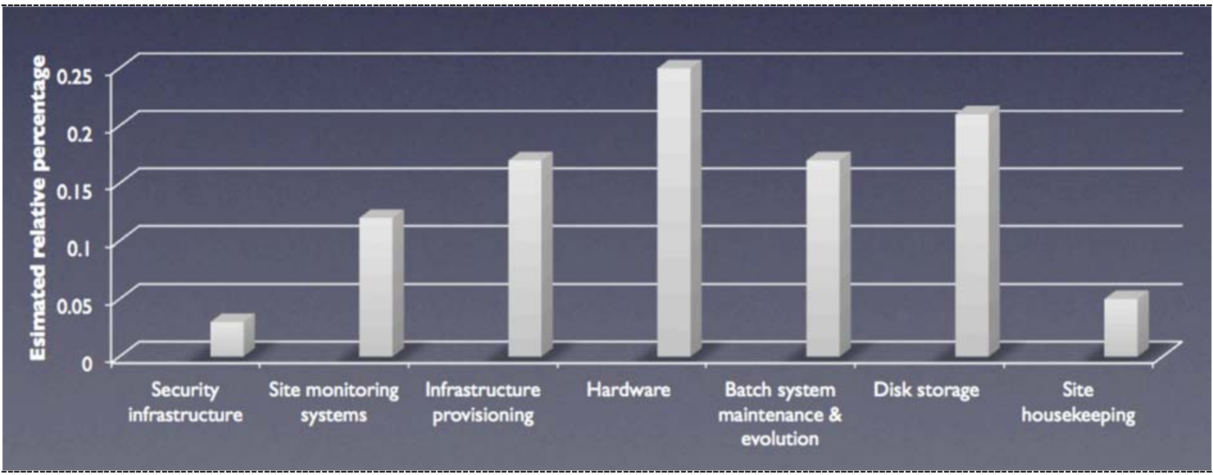

Figure 7. An indicative breakdown of GridPP Tier-2 site tasks and the relative effort that goes into providing each task. A possible route to reducing required Tier-2 effort is through lightweight batch and infrastructure provisioning approaches.

already well below the WLCG survey average of 2.8 FTE, but it is expected this may be reduced further in the coming years.

Acknowledgements

This paper draws upon the work of many individuals from across GridPP and is written on behalf of the collaboration. The author wishes to specifically acknowledge Stephen Jones (Liverpool) who wrote DrainBoss, Alessandra Forti (Manchester) who led the multicore work, Andrew Lahiff (RAL Tier-1) whose work drove forward the HTC evaluations, Tom Whyntie (QMUL) who pushed the wider VO work, Andrew McNab (Manchester) who developed the Vac and Vcycle implementations, and the team at Imperial College who implemented DIRAC.

8. References

[1] Faulkner P J W *et al*, The GridPP collaboration, GridPP: development of the UK computing Grid for particle physics, *Journal of Physics G: Nuclear and Particle Physics* **32** N1-N20, 2006; doi: 10.1088/0954-3899/32/1/N01.
[2] For more details see this presentation **http://tinyurl.com/nm8uefh.**
[3] http://cernvm.cern.ch/portal/startcvmfs
[4] "Running jobs in the vacuum". A McNab et al 2014 J. Phys.: Conf. Ser. 513 032065. For up-to-date information visit http://www.gridpp.ac.uk/vac/.
[5] http://www.gridpp.ac.uk/vcycle/.
[6] "Evaluation of containers as a virtualisation alternative for HEP workloads" by A Washbrook et. al. CHEP 2015 contribution 356: http://indico.cern.ch/event/304944/session/7/contribution/356.
[7] A. Tsaregorodtsev et al. DIRAC - Distributed infrastructure with Remote Agent Control. 2003.
[8] A Tsaregorodtsev, M Bargiotti, N Brook, A C Ramo, G Castellani, P Charpentier, C Cioffi, J Closier, R G Diaz, G Kuznetsov, Y Y Li, R Nandakumar, S Paterson, R Santinelli, A C Smith, M S Miguelez, and S G Jimenez. Dirac: a community grid solution. Journal of Physics: Conference Series, 119(6):062048, 2008.
[9] The step-by-step approach taken for the CERN@School example is explained in T Whyntie's 2014 talk **http://tinyurl.com/o3bgojx.**