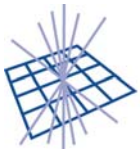

GridPP Project Management Board

Policy on Sites Stopping Stalled Jobs

Document identifier :	**PMB-113-Inefficient_Jobs**
Date:	**29/06/2007**
Version:	**0.5**
Document status:	**Draft**
Author	**PMB, Graeme Stewart**

Introduction

The Tier-2 review highlighted a number of issues raised by sites: one current problem is the number of inefficient jobs. Following discussion between the Deployment Team and the PMB, this document proposes a policy to enable UK sites to stop stalled jobs.

The Problem

Significant T2 computing resources are currently under-utilised due to a small, but significant number of jobs that are unable to complete, but occupy job slots for extended periods of time. This problem was considered as one of the generic problems to be resolved via a UK-wide policy to kill inefficient or stalled jobs. As an example of the problem, the efficiency (=CPU/wall time) recorded for all jobs at the UKI-SCOTGRID-GLASGOW Tier-2 site for the period from October 2006 – June 2007 (06Q4-07Q2) are shown in figure 1.

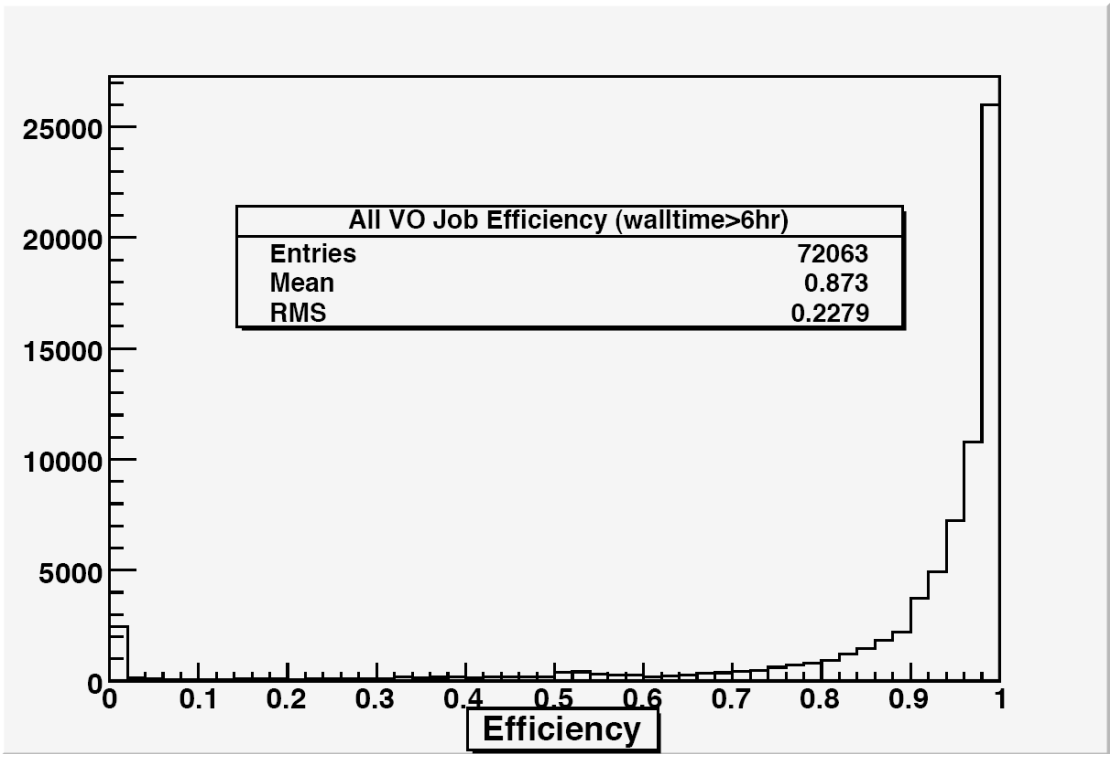

Figure 1: job efficiency for all VOs at a T2 site for jobs with wall time > 6 hours

The choice of a 6 hour period is relevant later w.r.t. the proposed policy. A small spike at zero, corresponding to < 2% efficiency, is noted in figure 1. This suggests that the scale of the problem is relatively small. However, for the largest VO (ATLAS), the problem is relatively much larger, with a spike corresponding to 6% of the total, see figure 2. This reflects a large complex computing environment and a large number of users within the VO who may not know that particular files are no longer available and have not placed sufficient checks in their code to allow for loss of required datasets.

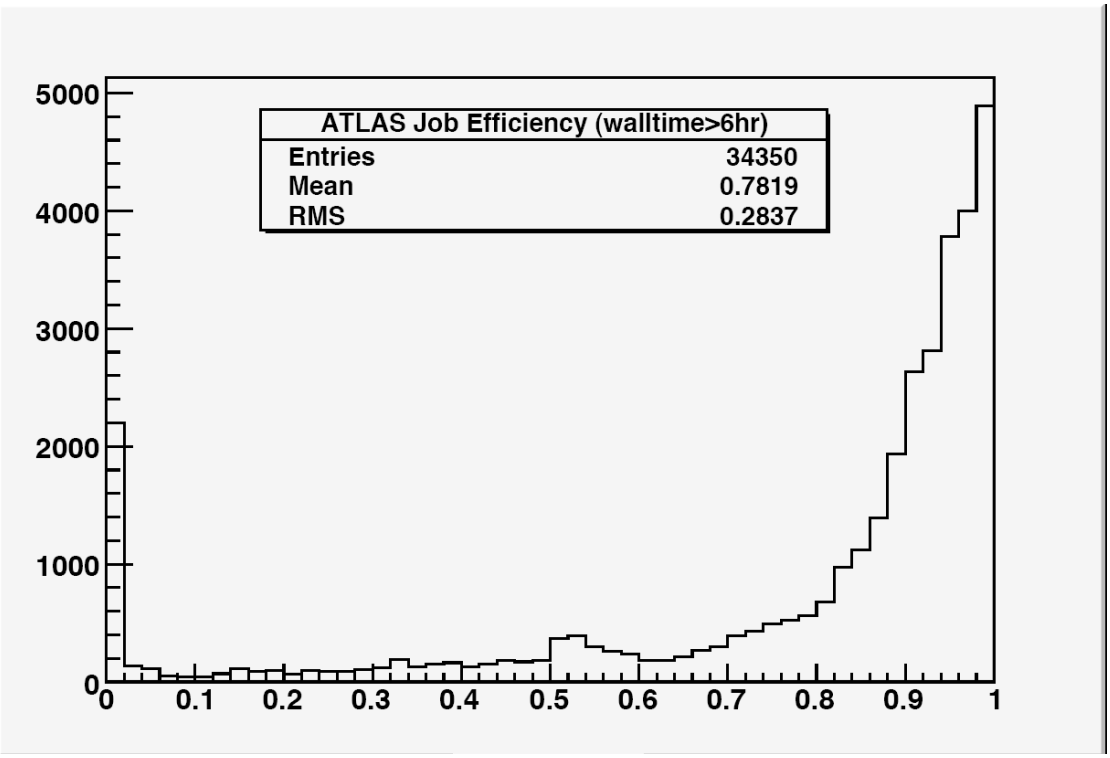

Figure 2: ATLAS job efficiency at a T2 site for jobs with wall time > 6 hours

The choice of a short 30 minute period includes more relatively inefficient jobs as seen in the increased spike at zero in figure 3 compared to figure 1.

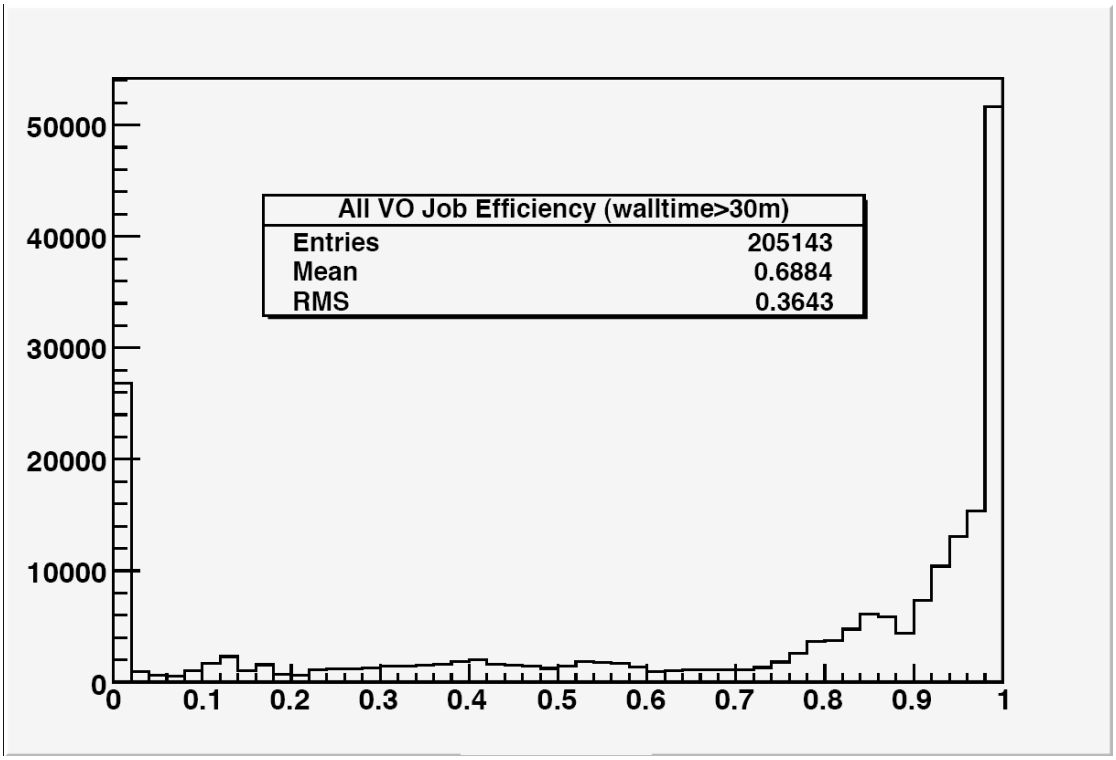

Figure 3: job efficiency for all VOs at a T2 site for jobs with wall time > 30 minutes

The ATLAS jobs subset shown in figure 4 are again less efficient. These predominantly short jobs, which exit themselves before occupying 6 hours of wall clock time are, however, not taking up significant CPU time.

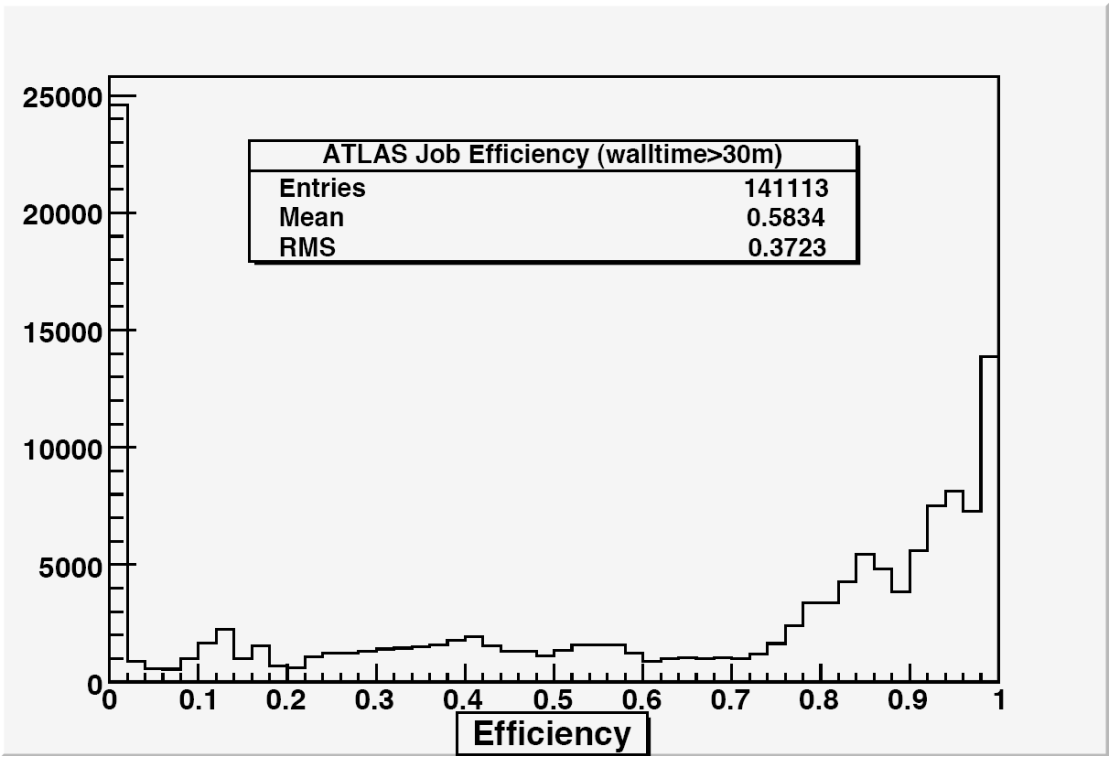

Figure 4: ATLAS job efficiency at a T2 site for jobs with wall time > 30 minutes

The more problematic jobs are the longer jobs with wall times > 24 hours. The relative scale of problematic jobs is seen in figure 5, indicating a significant fraction (~15%) of all long jobs are highly inefficient, usually due to stalled data management commands.

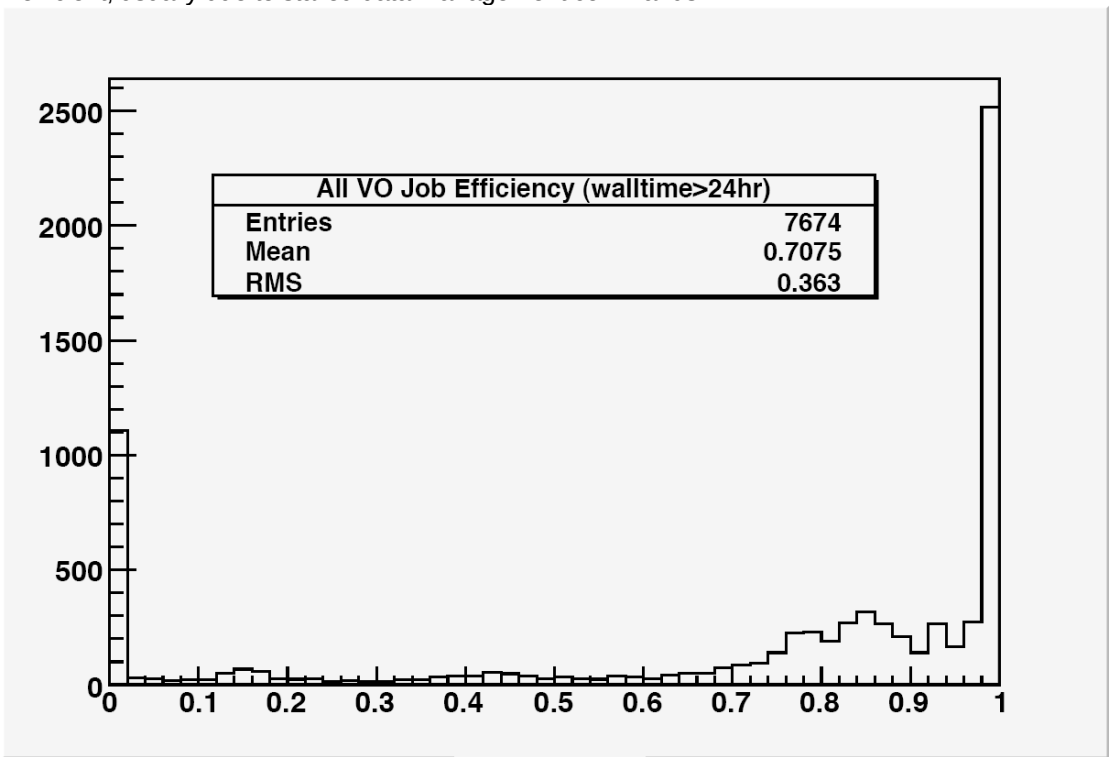

Figure 5: job efficiency for all VOs at a T2 site for jobs with wall time > 24 hours

The problem is relatively larger for the ATLAS VO, where more than 20% of the longest jobs are <2% efficient. The ATLAS long jobs constitute more than 50% of the long jobs at this particular Tier-2 site. Introducing cuts will enable more efficient jobs (for all VOs) to access the resources.

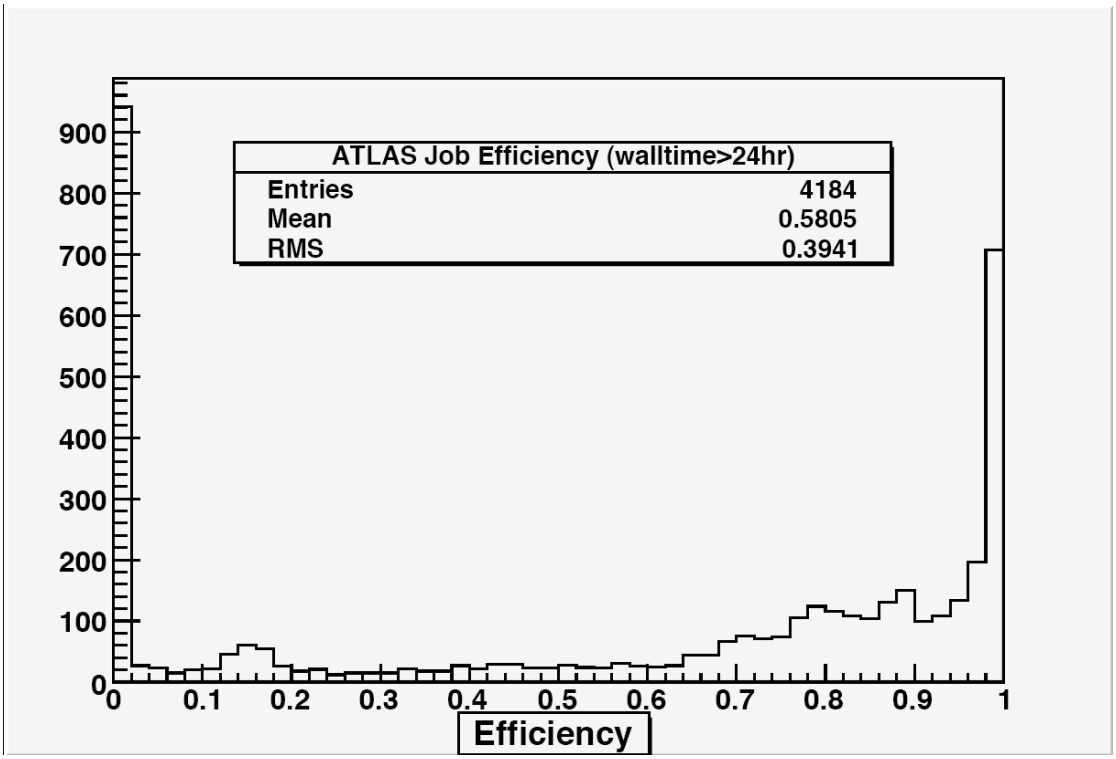

Figure 6: ATLAS job efficiency at a T2 site for jobs with wall time > 24 hours

Scale of Resource Loss

Table 1 shows the scale of resources being lost to inefficient (<0.02) jobs.

Job Walltime Cut	Number of Inefficient Jobs	Walltime used by Inefficient Jobs (min)	Total Walltime Used (min)	Fraction of Walltime wasted by Inefficient Jobs
All Jobs	64976	1275192932	14584387568	0.087
> 30 minutes	26832	1226102816	14497309856	0.085
> 6 hours	2467	958032536	11227734855	0.085
> 12 hours	1838	905090087	9150630922	0.099
> 18 hours	1676	881031858	6907171129	0.128
> 24 hours	1098	756108670	4146076111	0.182

Table 1: Loss of cluster wallclock time due to inefficient jobs on UKI-SCOTGRID-GLASGOW, 2006-11 to 2007-06.

Distribution of Resource Loss

It can be seen from Figure 7 that the distribution of resource loss is extremely variable – most inefficient jobs arrive in batches, indicating a particular problem in an offsite component. In the 200 days of data accumulated inefficient jobs appear in about 15 “bunches” – periods of more then 20 per day. We believe, therefore, that this sets an appropriate threshold for sites to contact users that will not be too onerous for the site, but will provide direct feedback to users and VOs on particular problems and help improve efficiency across the grid.

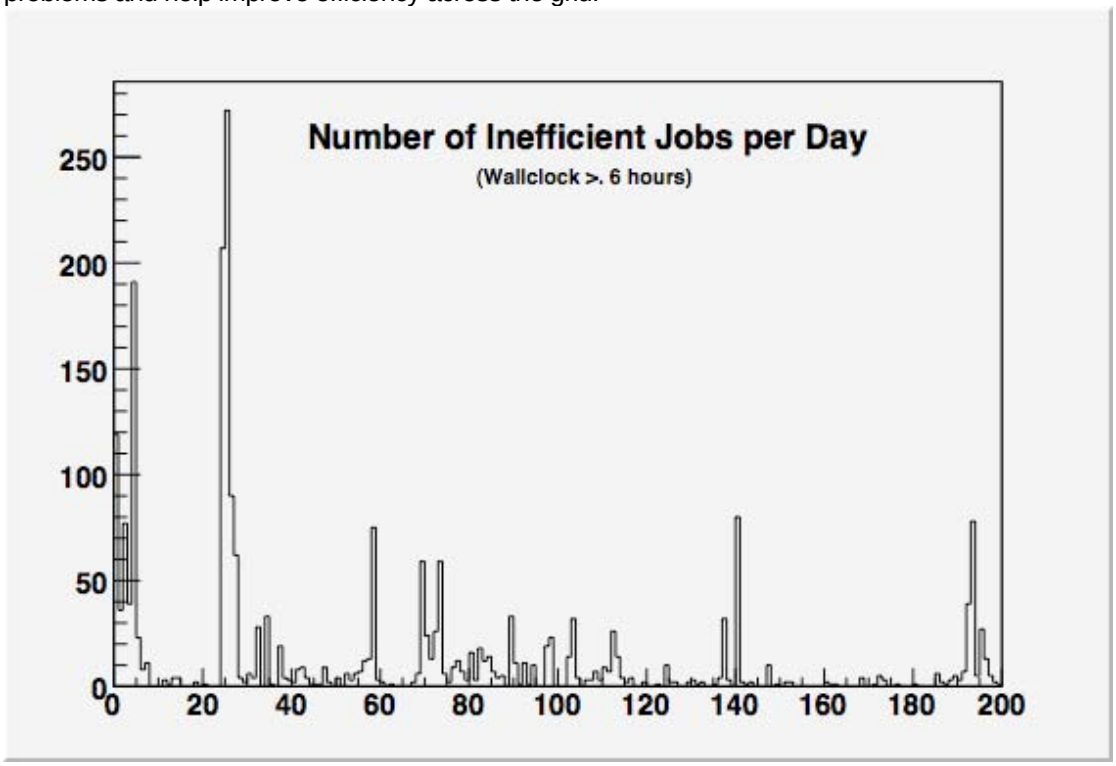

Figure 7: Number of inefficient jobs per day using wallclock > 6 hours.

Draft Policy

All UK sites are given flexibility to deal with stalled jobs (in order that their CPUs are occupied more fully overall) according to the following intervention scheme:

1) Any job consuming **<10** minutes CPU over a given **6** hour period (efficiency < 0.027) is considered stalled.

2) If the site identifies that the problem is due a well known problem, e.g., a hanging `lcg-cp` command, then the jobs may be deleted at once, with the user or VO being informed of the problem. (Note, the site should attempt to identify the SE, SURL or LFN involved to help debug the underlying data management issue).

3) In cases where the reasons for the stall are not obvious (e.g., a binary just hanging), sites should contact users or VO production teams informing them of the number of stalled jobs at a site, providing as much information as possible to help debug the problem. (An example of such an email is given in Appendix 1).

    1) The user or VO should respond within **6** hours, stating whether the site should simply delete the jobs or take further debugging steps.

    2) If no response is received within **6** hours the site may delete the jobs, informing the user of the action taken.

In order to contact users via e-mail the CIC portal lookup of a user's email from their DN will be used.

In order to contact VO production teams Operational Contact email address from the CIC portal will be used.

If no contact is possible with the user or VO because this information is unavailable then the site may delete the stalled jobs from the batch system immediately. If more that **20** jobs are deleted in any **24** hour period this way by the site then the site should raise a GGUS ticket against the user or VO for future reference.

All inefficient job parameters (highlighted in bold) will be reviewed at the outset and on an **annual** review basis by DTeam.

Notes

1. The proposed cut corresponds to 2.7% inefficiency is considered to be well out in the tails of any reasonable job.

2. Another cut (type 2) could be implemented to deal with jobs that run for 48 hours at 100% efficiency then stop when, e.g., uploading output files. This is left for a second phase.

3. Cut 1 is easily implementable as a cron script or a nagios alarm.

4. Cuts of type 2 are trickier to automate since this requires profiling of cpu/wallclock over the job's life using e.g. nagios. We think that as policy it's the right criterion to use and could be used as a next step.

5. On the lookup of a user's email, if the user/VO does not have this information available then the site will make no further attempts to contact the user. Killed sgm or prd jobs will be reported via the appropriate VO operational contacts (e.g. atlas-comp-oper@cern.ch).

6. Overarching WMS problem is that the default retry rate is 3, so these deleted jobs will be re-submitted and will hang about on other resources (typically elsewhere).

7. Problem jobs usually do come in batches, so e-mail will typically be required. However, the user may be confused when receiving an e-mail from a site stating that their job has been deleted as inefficient. The problem may not be with the job per se, but with the external dataset availability which then may work on re-submission (due to datasets becoming available again). It would be helpful if the site examined in detail the hanging command on an example job and passed this information back to the user (e.g., https://gus.fzk.de/pages/ticket_details.php?ticket=22717).

8. Another implementation problem and another reason for e-mail and site-user interaction so that users know what is going on.
All "qdel"ed jobs would get an exit status of 271. However, the owner of the will not see this in their output. They'll actually get an edg-job-status:

```
Current Status:     Aborted
Status Reason:      Job RetryCount (3) hit
```

Getting all of the logs (edg-job-get-logging-info -v 2) gives the failure in each case as the cryptic:

```
Event: Done
- exit_code               =    1
- host                    =    svr023.gla.scotgrid.ac.uk
- level                   =    SYSTEM
- priority                =    asynchronous
- reason                  =    Cannot read JobWrapper output, both from
Condor and from Maradona."
```

9. Can Tier-2 sites (with modest manpower) be spending any time contacting individual users via e-mail? What is an acceptable commitment from a Tier-2 site in terms of reporting back user problems?

10. The issue was raised informally with the ATLAS VO who were content that if a job hung around for even an hour it was most likely irreparably stuck.

11. VO support people may well end up with the tickets from the users whose jobs are cancelled (given that the user messages that return are unclear).

12. The policy should be floated to all the VOs when GridPP reach a consensus but before we implement.

13. We need a policy for VOs to define genuinely inefficient jobs – e.g. stripping or merging jobs which have high i/o and low efficiency. (The 2% efficiency cut is meant to be sufficiently low to address these scenarios).

14. We are wary of becoming a black hole for a VO's jobs and hence have set cuts well out in the tails. Although the RB may send the retries elsewhere, there is nothing to stop it sending us all the other jobs queued by that user to other UK sites (which will similarly ultimately abort the job).

15. We are aware that stopping stuck jobs is not a replacement for proper long- term debugging of site storage systems and experiment data access methods. However intervention from sites looks like a good place to initiate deployment discussion based on actual examples.

16. The simple "qdel" method could be (significantly) improved. We will raise this as a requirement on the SA1 top ROC issues list. See https://twiki.cern.ch/twiki/bin/view/EGEE/SA1_TCG
It is proposed to kill the hanging command on the WN, instead of “qdel”ing. This should allow some output to get back the the user and prevent re-submission by the RB.

17. Does anyone know what list of users is known to the CIC Portal? Is a google search for the user name better?

Pros versus Cons

Of killing jobs without informing users:

Pros	Cons
Users jobs are explicitly stopped from being (highly) inefficient.	Users may be unaware why their jobs failed.
The proposed cut would increase available resources by ~10% based on current data (at no extra cost).	Resources are not (yet) contentious.
Inefficient users within VOs would not use up resources.	WMS will resubmit jobs anyway.
Site resources are freed.	Sites could be blacklisted.
The CPU efficiency cut is simple and transparent.	The problems are really i/o related and should be addressed by more appropriate cuts.
Easy to implement on UK site basis.	Better to implement WLCG-wide.

of *post facto* e-mail notification after a deletion:

Pros	Cons
E-mail notification is useful feedback from the site to the user stating that job was not making efficient use of the CPU resources.	E-mail feedback requires non-automated methods (but could be automated).
User may wish to follow-up the problem. E-mail exchange encourages helpful site-user interaction. The inefficiency may e.g. be symptomatic of a lost dataset.	This is additional (non-automated) effort. Open to proliferation. Error reporting should anyway be handled via "the Grid".

of the e-mail notification before taking action:

Pros	Cons
Real steps to understand the problem can be taken by the site and the user, hopefully leading to code improvements and a rise in job efficiencies at all sites.	This is the most time consuming option for the site. Sites may decide to never intervene on stalled jobs if they feel they cannot spare the effort, thus the situation remains the same, with substantial resource being wasted and no feedback for users The additional 6 hour window will result in more resources being wasted.

No e-mail notification requires no additional site effort (or e-mail feedback could be automated). In this case the user is unaware why their particular job failed at this particular site but this is just a subset of the general case.

Appendix 1: Example of Email to VO on stalled Jobs

The following email is a sample of the information sites should give to VOs or users when their jobs have stalled:

From: g.stewart@physics.gla.ac.uk
Subject: Stalled LHCb jobs at Glasgow
Date: 29 June 2007 10:34:58 BDT
To: lhcb-production@cern.ch

Hi

We've got 23 LHCb jobs currently at Glasgow which have stalled.

The jobs started between 06/28/2007 12:04:58 and 06/28/2007 14:40:31 (UT+1) and thus far have consumed just over a minute of CPU.

I assume these are pilot jobs which have failed to start their payload properly?

An example of the stuck command is:

```
lhcbprd  22119  0.0  1.3 299320 111192 ?     S    Jun28   0:15
/tmp/WMS_node129_011662_https_3a_2f_2flcgrb02.gridpp.rl.ac.uk_3a9000_2fq58aFUmVeJc-
nkcbTla7eQ/lib/slc3_ia32_gcc323/ld-linux.so.2 --library-path
[…]
```

Which seems to be halted on:

```
node129:~# strace -p 22119
Process 22119 attached - interrupt to quit
futex(0x43385e0, FUTEX_WAIT, 2, NULL
```

Shall I kill these off, or do you want to perform any more diagnostics on them?

Cheers

Graeme

```
--
Dr Graeme Stewart - http://wiki.gridpp.ac.uk/wiki/User:Graeme_stewart
ScotGrid - http://www.scotgrid.ac.uk/ http://scotgrid.blogspot.com/
```