Contribution ID: **517** Type: **Poster**

Sysematic analysis of job failures at a Tier-2, and mitgation of the causes.

Tuesday 22 May 2012 13:30 (4h 45m)

Failure is endemic in the Grid world - as with any large, distributed computer system, at some point things will go wrong. Wether it is down to a problem with hardware, network or software, the shear size of a production Grid requires operation under the assumption that some of the jobs will fail. Some of those are anavoidable (e.g. network loss during data staging), some are preventable but only within engineering tradeoffs (e.g. uninteruptable power supplies on worker nodes), and some are fully preventable (e.g. software problems).

It is not clear that the current level of job failures is at the minimum level. We have been logging all failed jobs, and classifying then according to how and why they failed. Some work have been invested into automated systems to collated job failure information from various sources, and these will be presented.

This work reports on that data, and supplies an analysis, as quantitative as possible, on what would need to have been different to have prevented those jobs from failing.

Some subtlty lies in the definition of a 'failed job'. From the perspective of an end user, any job that does not do what the user wants can be considered failed, but that is not the most useful definition for an infrastructure provider. However, it is useful to track such cases in order to provide a comparison point to the infrastrucutre caused job failures. Not all problems in the infrastructure result in user visibale job failures; for example a problem in a batch system schedular can result in no jobs being started at for some point, which is only visible as reduced throughput. These were tracked, but can't be quantified in the same scale as user visible job failure modes.

Clearly, there are some cases of job failure that it is not within the capability of a site administrator to resolve - If user code divides by zero, no aspect of site administration can resolve that. However, there are many other sources of problems other than that. Of particular interest are jobs that report a failure the first time, but succeded on a re-submission. Jobs falling into that description include (but are not limited to) all the jobs where something transient went wrong at a site. This is the class of failures which it is within the capability of a site manager to reduce to zero, which is the long term goal of this work.

Deep analysis of these probalemtic cases is required, in order to determine the underlying causes, and further work is needed in order to prevent the problems from re-occuring. One early observation was that, in general, the slower the rate of failures detected, the more likely a root fix was to be found. A detailed analysis of this will be reported, but if this observation holds then it suggests that there might be net super-linear improvements in reliability from this sort of work.

In cases where it appeared that the root cause was located in some component, although no precise reason could be found, an alternative for that component was sourced, and compared with the original. Where possible, such as with a computing element, these were run in parralell. One such case, were no alternative could be found, we wrote an alternative implementation of the BLAH parser for CREAM, and compared it's stability to the supplied one.

Many problems that have been identified come down to either a hardware problem, or some interaction between multiple components. A survey of these will be presented. For hardware problems, an estimated cost to prevent the problem from being user visible will be given, and for hte problems with interacting components a series of reccomendations can be given.

Overall, this work is of importance in ensuring the imporved user experience on the Grid, and also a reduction in the manpower required to operate a site. Although the analysis might not be feasable at smaller sites, the

lessons learned from this work will be directly applicable to many sites, and should contribute to the smooth running of the Grid in the future.

Summary

Some degree of job failure is inevitable, but this work assumes, and demonstrates, that the level of failure is not at the minimum level. Sources of job failures are found, analysed and mitigations and solutions given. In some cases this is as simple as configuration, but includes cases where new software components have been written, in addition to monitoring and analysis tools.

Some discussion is included over the concept of a failed job, and to what extent these can be eliminated - it clearly being infeasble to prevent any jobs fromfailing.

Overall, this work is of importance in ensuring the imporved user experience on the Grid, and also a reduction in the manpower required to operate a site. Although the analysis might not be feasable at smaller sites, the lessons learned from this work will be directly applicable to many sites, and should contribute to the smooth running of the Grid in the future.

Author: PURDIE, Stuart (University of Glasgow)

Co-authors: CROOKS, David (University of Glasgow); MITCHELL, Mark (University of Glasgow); SKIPSEY, Sam (University of Glasgow / GridPP)

Presenter: PURDIE, Stuart (University of Glasgow)

Session Classification: Poster Session

Track Classification: Computer Facilities, Production Grids and Networking (track 4)