

LSF Replacement

LSF may soon be replaced at SLAC, where the new batch allocation system may not support all Fermi OSes such as Windows and Mac. The committee's [restricted Confluence page is here](#). Two top contenders are SLURM and Univa Grid Engine: *SLURM does not support Windows and they further claim it would be a 'big deal' to make that happen. On the other hand, UGE (Univa Grid Engine) hints that OSX support might not be so difficult (and they do support Windows).*

It certainly not clear what the ultimate decision will be, and perhaps LSF will still be deemed the "best deal".

Our Options

Keep LSF

Amber has offered the possibility that we could retain some LSF licenses. Who would maintain LSF in that scenario?

Use Hudson/Jenkins as a "Batch Allocation System"

Tony's Idea:

My basic idea is that somewhere within the RM scripts you submit a job to LSF (via Navid's old batch interface?). Instead you could submit the same script to Hudson and it would deal with queuing the job and running it on the appropriate machine. On the mac and windows machines which belong to Fermi this would be simple, we just need to run the Jenkins build agent (Java) on those machines. For machines which do not belong to us (e.g unix LSF nodes) it may be harder -- I don't know how many types of unix machines we currently run the build on?

Replace SCons RM completely with Hudson/Jenkins

- - What features of the current RM would we need to recreate for Hudson?
 - And how is that going to be accomplished?

We actually briefly discussed this back in Sept. 2011. Below are some exchanges with Tony from last year:

Need to teach Hudson about LATEST, HEADs and Releases:

We normally run hudson whenever a change is made in cvs/subversion. I imagine it can be configured to run on specific tags, alternatively we could write a module which triggers the build externally as we did for the workbook.

Handling things like Doxygen generation, running unit tests, red dot summary etc

I didn't read the full list of tasks, but for EXO we certainly run make and the use root to generate documentation, so doing similar things for Fermi should be possible. This would certainly require some setup, maybe Kim Lo would be available to help with some of this? We should also be able to store extra information about the builds in our own database, and augment the hudson interface with extra web pages using information from the database.