

HPS @ OSG

Container

Singularity is required for running jobs on OSG. Fat or slim container (i.e. containing all HPS software or not)? Slim is easier to maintain and allows faster changes. If slim, preferably centos7 as used at JLab, assuming JLab is where the jobs will be submitted from. OSG has facilities to take your DockerHub repository, trigger on its changes, and build and deploy corresponding singularity container on CVMFS. In general, these images are installed by OSG in an official, sanctioned location available on CVMFS. Need to follow up on how exactly to register all that. And before going to production mode, we'd ideally want our own container, although testing can happen with an existing one.

CVMFS/XROOTD

We have write-access to both, supported by JLab. CVMFS is appropriate for smaller things you have to read in full that don't change frequently, e.g. software builds, configuration files, databases. XROOTD is appropriate for streaming larger data, e.g. files from random triggers for background overlay. If slim container, full software stack (and sqlite snapshots) goes on CVMFS, which ideally should be final and well organized, e.g. naming scheme with OS/compiler version in it. Note, CVMFS is not good for things that change regularly, e.g. propagation time means you don't easily know what you got.

Workflow

Can it be organized to include the full chain in a single job, and the jobs be 100% independent of each other? e.g. starts with event generator, all the way to reconstruction, bringing back only the necessary LCIO file (which should be optimized to drop unnecessary info). If the jobs need input data, there's two ways, the less efficient transferring it in the submit payload, or XROOTD, and either way we need numbers on the size per job. Note, need to care for random number seeding, although maybe that's already done, e.g. relying on unique 1-second clock isn't good enough.

Network

Ideally the jobs should need no network access. Using sqlite instead of the conditions database server is a must. Jeremy mentioned something else in hps-java that uses network but can be disabled.

Numbers

We need total #EVENTS/CPU/RAM/IO per job. Short is good (few hours at most), to avoid preemption. 1 GB memory is fine. Total data in/out per job.

Submission

Currently submission from JLab is manual, only from a dedicated submit node, limited to 1 person per experiment, and the monitoring is also mostly manual via command-line tools. The person will need to be available for contact from scicomp and myself.

JLab is working on changing that model, e.g. hooks in SWIF to switch between different farms to let anyone with scicomp credentials submit OSG jobs just as they do now for JLab's farm, and also JLab is becoming a full OSG site eventually. But when we discussed it earlier this summer, it seemed that HPS would want to use OSG well before that would be available and advice was to not wait but pursue the current method.

Preliminary Tests

I did basic tests, submitting jobs under the HPS OSG project, using a slim centos7 container from Gluex, running some CLAS12 (java-based) software from CVMFS, and getting some dumb data back. For further testing, we'd need a script that runs the full simulation chain, and assuming the slim-container option, HPS software installed on CVMFS. But I suspect that's not so useful yet until we get all the ducks above in a row first. Also, we may discuss if hps-mc should be generating the submit files, or instead use a single templated one and just run a script from hps-mc.