Analysis Environment Details

Accounts and Resources

A description of the LCLS computing resources can be found at the PCDS Computing page.

Setting Up the Software Environment

For the analysis software to work correctly it has to run in a correct environment. LCLS Analysis environment consists of few environment variables that either point to a location of specific directories or define some configuration options such as type of the machine, compiler versions, etc. Users do not need to set environment variables manually, instead users need to execute just one simple command which defines the complete environment. The command that needs to be run from the "bash" shell is:

```
source /reg/g/psdm/etc/ana_env.sh
```

Other shells (e.g. tcsh) are not supported by the world-standard conda package management system we use.

The above command needs to be executed every time when a user logs in. For convenience it should be added to the user's shell login script (e.g. .bash_profile file for bash)

Environment variables

This paragraph is informational and you can skip it if you are not interested in the internals of this setup. Various procedures, scripts, etc. depend on a small number of the environment variables which define location of various parts of the software and the system. Here are the variables that have to be present in the user's shell environment:

- *SIT_ENV* – defines the type of the SIT environment, can be *default* for regular setup or *ana* for analysis environment
- *SIT_ROOT* – defines the location of the top directory for the PSDM software, usually `/reg/g/psdm`.
- *SIT_RELDIR* – directory with regular PSDM release
- *SIT_RELEASE* – the name of the software release to use, something like "current" or "ana-1.1.5"
- *SIT_ARCH* – the name of the hardware/software architecture to use like `x86_64-rhel5-gcc41-opt`
- *SIT_DATA* – list of directories where the *data files* reside. These are not regular XTC data files, but rather some additional data that software needs. Usually it includes "<test-release>/data" and `$SIT_RELDIR/$SIT_RELEASE/data` (that is "data" directory in the releases) but can also be extended to include other locations.
- *SIT_REPOS* – the locations of all "base" releases. Currently we only support single base release so this variable would look like `$SIT_RELDIR/$SIT_RELEASE`
- *LSF_PACKAGE_LIB* – the location of the lsf package lib directory. Typical users will not set this. When at LCLS, it will be set to /afs/slac.stanford.edu/package/lsf/curr/lib/. Offsite users that have their own lsf installation they want to try with an ana release can set this appropriately.
- *SIT_NO_MODIFY_OMPI_ENV* – typical users will not set this. Only offsite users that set OMPI environment variables may want to set this. see below.

Derived variables:

- *PATH* – should be extended to include `<test-release>/arch/$SIT_ARCH/bin` and `$SIT_RELDIR/$SIT_RELEASE/arch/$SIT_ARCH/bin`
- *LD_LIBRARY_PATH* – should be extended to include `<test-release>/arch/$SIT_ARCH/lib` and `$SIT_RELDIR/$SIT_RELEASE/arch/$SIT_ARCH/lib, and $LSF_PACKAGE_LIB`
- *PYTHONPATH* – should be extended to include `<test-release>/arch/$SIT_ARCH/python` and `$SIT_RELDIR/$SIT_RELEASE/arch/$SIT_ARCH/python`

The <*test-release*> place holder is the absolute path to test release which is a release that is based on a base release (in $SIT_RELEASE). All these variables could be set manually, but there is a single convenience script which can be sourced from the user's shell which does all the tedious work.

MPI and LSF

The starting with ana 0.18.0, openmpi will try to load lsf components. sit_setup will set LSF_PACKAGE_LIB to the appropriate directory at SLAC. For offsite users, it will instead set three OMPI environment variables to instruct openmpi to not try to load the lsf components. These variables are

OMPI_MCA_ess = ^lsf
OMPI_MCA_ras = ^lsf
OMPI_MCA_plm = ^lsf

Advanced users may want to set these variables themselves. If so, set

SIT_NO_MODIFY_OMPI_ENV=1

before setting up the ana environment.

Obtaining Kerberos ticket

If you plan to work with code repository and use commands like `svn`, `addpkg`, etc. then for authorization you will need to obtain a Kerberos ticket. Ticket can be obtained at any time after you logged in by running command `kinit` which will prompt your for your regular password. Kerberos ticket is valid for 25 hours after which time one needs to re-run the command. Without ticket svn will print an error message "authorization failed" when it tries to access repository (some svn commands do not require repository access).