Tara Administrator Documentation

Release 0.1

Putt Sakdhnagool

Feb 12, 2019

Contents

CHAPTER 1

Contents

1.1 Tara Cluster

1.1.1 Timeline

Important Date	Plan
30 Jan 2019	Initialize document

1.1.2 System Overview

General Information

slurm Version: 18.08.5 (Jan 30, 2019)
munge Version: 0.5.13 (Sep 27, 2017)
PMIx Version : 3.0.2 (Sep 19, 2018)
nhc Version : 1.4.2 (Nov 12, 2015)
UCX Version : 1.4.0 (Oct 30, 2018)
lmod Version : 6.6.3
EasyBuild Version : 3.8

slurm user: `slurm`
slurm UID: `2001`
slurm group: `slurm`
slurm GID: `2001`

munge user: `munge`

munge UID: `2000`
munge group: `munge`
munge GID: `2000`

module user: `modules`
module UID: `2002`
module group: `modules`
module GID: `2002`

Initial Setup

- *Clean* installation of CentOS 7. (If possible, the initial state of Tara cluster machines.)
- An NFS storage mounted to all machines. (Can we create a virtual HDD and mount it to all machine?)

Machine Configuration

Node Class	NodeName	IP (InfiniBand)	IP (1GbE)	IP (Ext)	Notes
frontend	tara-frontend-1	172.20.1.2	172.21.1.2	10.226.33.1	

Machine Layout

1.1.3 System Installation

All RPM packages are built in the *tara-c-060* node built with `tara-build-centos7` image.

munge Version: 0.5.13 (Sep 27, 2017)
PMIx Version : 3.0.2 (Sep 19, 2018)
UCX Version : 1.4.0 (Oct 30, 2018)
Slurm Version: 18.08.5 (Jan 30, 2019)

Build Order

1. munge
2. OpenUCX
3. PMIx
4. Slurm
5. Lmod/EasuBuild

Common Tools

We will use `rpm-build` for building RPM packages from source code and `wget` for downloading the code.

```
$ yum install rpm-build wget
```

MUNGE

Building MUNGE RPMs

Download the latest version of MUNGE

```
$ wget https://github.com/dun/munge/releases/download/munge-0.5.13/munge-0.5.13.tar.xz
```

Note: EPEL repository does not contain the latest version of `munge` package.

Install MUNGE dependencies

```
$ yum install gcc bzip2-devel openssl-devel zlib-devel
```

Build RPM package from MUNGE source.

```
$ rpmbuild -tb --clean munge-0.5.13.tar.xz
```

Create MUNGE directory in parallel file system and move RPM files.

```
$ mkdir -p /utils/munge
$ mv rpmbuild/ /utils/munge/
```

Install and start MUNGE

Generate `munge.key`. **Need to do only once**

```
$ dd if=/dev/urandom bs=1 count=1024 > /utils/munge/munge.key
```

Create *munge* user and group.

```
$ groupadd munge -g 2000
$ useradd --system munge -u 2000 -g munge -s /bin/nologin --no-create-home
```

Install MUNGE from RPM.

```
$ rpm -ivh /utils/munge/rpmbuild/RPMS/x86_64/munge-0.5.13-1.el7.x86_64.rpm \
  /utils/munge/rpmbuild/RPMS/x86_64/munge-libs-0.5.13-1.el7.x86_64.rpm \
  /utils/munge/rpmbuild/RPMS/x86_64/munge-devel-0.5.13-1.el7.x86_64.rpm
```

Create MUNGE local directory and copy `munge.key`.

```
$ mkdir -p /etc/munge/
$ chown -R 2000:2000 /etc/munge/
$ chmod 500 /etc/munge/
$ cp /utils/munge/munge.key /etc/munge
$ chmod 400 /etc/munge/munge.key
```

Start MUNGE service

```
$ systemctl enable munge
$ systemctl start munge
$ systemctl status munge
```

Testing MUNGE installation

```
$ munge -n
$ munge -n | unmunge
$ munge -n | ssh <host> unmunge
$ remunge
```

Note: By default the Munge daemon runs with two threads, but a higher thread count can improve its throughput. For high throughput support, the Munge daemon should start with ten threads

OpenUCX

https://github.com/openucx/ucx/releases

```
yum install numactl numactl-libs numactl-devel

export LD_LIBRARY_PATH=/usr/local/cuda-10.0/lib64\
                     ${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}

./contrib/configure-release  --prefix=$PWD/install --with-cuda=/usr/local/cuda/

rpmbuild -bb --define "configure_options --enable-optimizations --with-cuda=/usr/
↪local/cuda" ucx-1.4.0/ucx.spec
```

PMIx

Build PMIx RPM Package

Install PMIx dependencies

```
$ yum install libtool libevent-devel
```

Download the latest stable version of PMIx

```
$ wget https://github.com/pmix/pmix/releases/download/v3.0.2/pmix-3.0.2.tar.bz2
```

Build PMIx package from PMIx source.

```
$ ./configure --with-munge=/usr --with-munge-libdir=/usr
$ rpmbuild -tb --clean --define "configure_options --with-munge=/usr" pmix-3.0.2.tar.
↪bz2
```

Note: PMIx script seems to support C11 features but will require gcc 4.9+

Create PMIx directory in parallel file system and move RPM files.

```
$ mkdir -p /utils/pmix
$ mv rpmbuild/ /utils/pmix/
```

Install PMIx from RPM.

```
$ rpm -ivh /utils/pmix/rpmbuild/RPMS/x86_64/pmix-3.0.2-1.el7.x86_64.rpm
```

Checking PMIx installation

```
$ grep PMIX_VERSION /usr/include/pmix_version.h

#define PMIX_VERSION_MAJOR 3L
#define PMIX_VERSION_MINOR 0L
#define PMIX_VERSION_RELEASE 2L
```

Slurm

Build SLURM RPM Package

Install SLURM and its plugins dependencies (See. slurm-plugins)

```
$ yum install readline-devel perl-ExtUtils-MakeMaker pam-devel hwloc-devel freeipmi-
↪devel lua-devel mysql-devel libssh2-devel
```

Download the latest stable version of SLURM

```
$ wget https://download.schedmd.com/slurm/slurm-18.08.3.tar.bz2
```

Build SLURM package from SLURM source with PMIx.

```
$ rpmbuild -tb --clean slurm-18.08.3.tar.bz2
$ rpmbuild -bb --clean --define "configure_options --with-ucx" slurm.spec
```

Create SLURM directory in parallel file system and move RPM files.

```
$ mkdir -p /utils/slurm
$ mv rpmbuild/ /utils/slurm/
```

Install Slurm

Create *slurm* user and group.

```
$ groupadd slurm -g 2001
$ useradd --system slurm -u 2001 -g slurm -s /bin/nologin --no-create-home
```

Install SLURM and its plugins dependencies (See. slurm-plugins)

```
$ yum install readline-devel perl-ExtUtils-MakeMaker pam-devel hwloc-devel freeipmi-
↪devel lua-devel mysql-devel libssh2-devel
```

Frontend

Install slurm from RPM packages.

```
$ rpm -ivh /utils/slurm/rpmbuild/RPMS/x86_64/slurm-18.08.3-1.el7.x86_64.rpm \
  /utils/slurm/rpmbuild/RPMS/x86_64/slurm-perlapi-18.08.3-1.el7.x86_64.rpm
```

Setup firewall

```
$ firewall-cmd --add-port 60001-63000/tcp --permanent
$ firewall-cmd --reload
$ iptables -nL
```

Slurmctld

Install slurmctld from RPM packages.

```
$ rpm -ivh /utils/slurm/rpmbuild/RPMS/x86_64/slurm-18.08.3-1.el7.x86_64.rpm \
  /utils/slurm/rpmbuild/RPMS/x86_64/slurm-slurmctld-18.08.3-1.el7.x86_64.rpm \
  /utils/slurm/rpmbuild/RPMS/x86_64/slurm-perlapi-18.08.3-1.el7.x86_64.rpm
```

Create required directory

```
$ mkdir -p /var/log/slurm/ /var/run/slurm/ /var/spool/slurm/
$ chown slurm:slurm /var/log/slurm/
$ chown slurm:slurm /var/run/slurm/
$ chown slurm:slurm /var/spool/slurm/
```

Setup firewall

```
$ firewall-cmd --add-port 6817/tcp --permanent
$ firewall-cmd --add-port 60001-63000/tcp --permanent
$ firewall-cmd --reload
$ iptables -nL
```

Edit PIDFile configuration in `/usr/lib/systemd/system/slurmctld.service` to the same localtion in `slurm.conf` (Current setting: `/var/run/slurm/slurmctld.pid`).

Following script could be use for editing.

```
$ sed -i -e 's@PIDFile=/var/run/slurmctld.pid@PIDFile=/var/run/slurm/slurmctld.pid@g'␣
↪/usr/lib/systemd/system/slurmctld.service
```

Create `slurmctld.conf` in `/usr/lib/tmpfiles.d/`. The content of `slurmctld.conf` is as follows

```
d /var/run/slurm 0755 slurm slurm -
```

Start `slurmdbd` service

```
$ systemctl enable slurmdbd
$ systemctl start slurmdbd
$ systemctl status slurmdbd
```

Note: `slurmctld` receives SIGTERM after the first setup. The problem was solved by editing the **PIDFile** configuration in the `.service` file and run command `systemctl daemon-reload`.

SlurmDBD

Install slurmdbd from RPM packages.

```
$ rpm -ivh /utils/slurm/rpmbuild/RPMS/x86_64/slurm-18.08.3-1.el7.x86_64.rpm \
  /utils/slurm/rpmbuild/RPMS/x86_64/slurm-slurmdbd-18.08.3-1.el7.x86_64.rpm
```

Create required directory

```
$ mkdir -p /var/log/slurm/ /var/run/slurm/
$ chown slurm:slurm /var/log/slurm/
$ chown slurm:slurm /var/run/slurm/
```

Create `slurmdbd.conf` in `/usr/lib/tmpfiles.d/`. The content of `slurmdbd.conf` is as follows

```
d /var/run/slurm 0755 slurm slurm -
```

Configure MySQL

The following SQL code creates a database `slurm_acct_db` and user `slurmdbd` and grants administrator privilege on the database to `slurmdbd` user.

```
CREATE DATABASE slurm_acct_db;
create user 'slurmdbd'@'<slurmdbd_IP>' identified by '<password>';
grant all on slurm_acct_db.* TO 'slurmdbd'@'<slurmdbd_IP>';
```

Edit PIDFile configuration in `/usr/lib/systemd/system/slurmdbd.service` to the same localtion in `slurmdbd.conf` (Current setting: `/var/run/slurm/slurmdbd.pid`).

Following script could be use for editing.

```
$ sed -i -e 's@PIDFile=/var/run/slurmdbd.pid@PIDFile=/var/run/slurm/slurmdbd.pid@g' /
↪usr/lib/systemd/system/slurmdbd.service
```

Setup firewall

```
$ firewall-cmd --add-port 6819/tcp --permanent
$ firewall-cmd --reload
```

Start `slurmdbd` service

```
$ systemctl enable slurmdbd
$ systemctl start slurmdbd
$ systemctl status slurmdbd
```

Note: `slurmdbd` receives SIGTERM after the first setup. The problem was solved by editing the **PIDFile** configuration in the `.service` file and run command `systemctl daemon-reload`.

Slurmd

Install `slurmd` from RPM packages.

```
$ rpm -ivh /utils/slurm/rpmbuild/RPMS/x86_64/slurm-18.08.3-1.el7.x86_64.rpm \
  /utils/slurm/rpmbuild/RPMS/x86_64/slurm-slurmd-18.08.3-1.el7.x86_64.rpm \
  /utils/slurm/rpmbuild/RPMS/x86_64/slurm-perlapi-18.08.3-1.el7.x86_64.rpm \
  /utils/slurm/rpmbuild/RPMS/x86_64/slurm-pam_slurm-18.08.3-1.el7.x86_64.rpm
```

Setup firewall

```
$ firewall-cmd --add-port 6818/tcp --permanent
$ firewall-cmd --add-port 60001-63000/tcp --permanent
$ firewall-cmd --reload
```

Create required directory

```
$ mkdir -p /var/log/slurm/ /var/run/slurm/ /var/spool/slurm/
$ chown slurm:slurm /var/log/slurm/
$ chown slurm:slurm /var/run/slurm/
$ chown slurm:slurm /var/spool/slurm/
```

Create `slurmd.conf` in `/usr/lib/tmpfiles.d/`. The content of `slurmd.conf` is as follows

```
d /var/run/slurm 0755 slurm slurm -
```

Edit **PIDFile** configuration in `/usr/lib/systemd/system/slurmd.service` to the same localtion in `slurm.conf` (Current setting: `/var/run/slurm/slurmd.pid`).

Following script could be use for editing.

```
$ sed -i -e 's@PIDFile=/var/run/slurmd.pid@PIDFile=/var/run/slurm/slurmd.pid@g' /usr/
↪lib/systemd/system/slurmd.service
```

Start `slurmdbd` service

```
$ systemctl enable slurmd
$ systemctl start slurmd
$ systemctl status slurmd
```

Note: `slurmd` receives SIGTERM after the first setup. The problem was solved by editing the **PIDFile** configuration in the `.service` file and run command `systemctl daemon-reload`.

Bringing node to idle state using `scontrol`. For example,

```
$ scontrol update NodeName=tara-c-00[1-6] State=DOWN Reason="undraining"
$ scontrol update NodeName=tara-c-00[1-6] State=RESUME
```

Installing nhc —— ————-

Download RPM package

```
$ wget https://github.com/mej/nhc/releases/download/1.4.2/lbnl-nhc-1.4.2-1.el7.noarch.
↪rpm
```

Install nhc package

```
$ rpm -ivh /utils/nhc/lbnl-nhc-1.4.2-1.el7.noarch.rpm
```

PAM Setup

`/etc/pam.d/sshd`

After `password include password-auth` line, adds

```
account    sufficient   pam_slurm_adopt.so
account    required     pam_access.so
```

In *pam_access* configuration file (`/etc/security/access.conf`), add

```
+:root:ALL
-:ALL:ALL
```

To guarantee that slurm services start after NFS, update `/usr/lib/systemd/system/slurmd.service` from

```
After=munge.service network.target remote-fs.target
```

to

```
After=munge.service network.target remote-fs.target etc-slurm.mount
```

Lmod and EasyBuild

Lmod

Install Lmod from EPEL repository.

```
$ yum install lmod
```

EasyBuild

Create *modules* group and user with a home-directory on a shared filesystem

```
$ groupadd modules -g 2002
$ useradd -m -c "Modules user" -d /utils/modules -u 2002 -g modules -s /bin/bash␣
↪modules
```

Configures environment variables for bootstrapping EasyBuild

```
$ export EASYBUILD_PREFIX=/utils/modules
```

Download EasyBuild bootstrap script

```
$ wget https://raw.githubusercontent.com/easybuilders/easybuild-framework/develop/
↪easybuild/scripts/bootstrap_eb.py
```

Execute `boostrap_eb.py`

```
$ python bootstrap_eb.py $EASYBUILD_PREFIX
```

Update `$MODULEPATH`

```
export MODULEPATH="/utils/modules/modules/all:$MODULEPATH"
```

Test EasyBuild

```
$ module load EasyBuild
$ eb --version

# OPTIONAL Unittest
$ export TEST_EASYBUILD_MODULES_TOOL=Lmod
$ python -m test.framework.suite
```

Enable access to all users.

Change permissions of `/utils/modules/`

```
chmod a+rx /utils/modules
```

Add `z01_EasyBuild.sh` to `/etc/profile.d/`. The content of the file is as follows

```
if [ -z "$__Init_Default_Modules" ]; then
  export __Init_Default_Modules=1
  export EASYBUILD_MODULES_TOOL=Lmod
  export EASYBUILD_PREFIX=/utils/modules
  module use $EASYBUILD_PREFIX/modules/all
else
  module refresh
fi
```

EasyBuild robot path.

```
/utils/modules/software/EasyBuild/3.7.1/lib/python2.7/site-packages/easybuild_
↪easyconfigs-3.7.1-py2.7.egg/easybuild/easyconfigs
```

Setup Lmod on other nodes

Install Lmod

```
$ yum install lmod
```

Add `z01_EasyBuild.sh` to `/etc/profile.d/`. The content of the file is as follows

```
if [ -z "$__Init_Default_Modules" ]; then
  export __Init_Default_Modules=1
  export EASYBUILD_MODULES_TOOL=Lmod
  export EASYBUILD_PREFIX=/utils/modules
  module use $EASYBUILD_PREFIX/modules/all
else
  module refresh
fi
```

Intel License Manager

https://software.intel.com/en-us/articles/intel-software-license-manager-users-guide

1.1.4 Slurm Configuration

Services

SLURM package to be installed

Node Class	Services
Controller (VM)	`slurm`, `slurm-perlapi`, `slurm-slurmctld`
Compute	`slurm`, `slurm-perlapi`, `slurm-slurmd`, `slurm-pam`
Frontend	`slurm`, `slurm-perlapi`
SlurmDBD (VM)	`slurm`, `slurm-dbd`

Plugins Dependencies

List of plugins and their dependencies to be installed when building SLURM RPM packages.
Need to check that the package contains these plugins after installing

Plugins	Dependencies
MUNGE	`munge-devel`
PAM Support	`pam-devel`
cgroup Task Affinity	`hwloc-devel`
IPMI Engergy Consumption	`freeipmi-devel`
Lua Support	`lua-devel`
My SQL Support	`mysql-devel`
X11	`libssh2-devel`

- **[TBD]**
  - **InfiniBand Accounting**: `libibmad-devel`, `libibumad-devel`
  - **cgroup NUMA Affinity**: ???

Configuration

Configuration in `/etc/slurm.conf`

Config	Value	Detail
SlurmctldHost	*slurmctld*	Might need to set as *slurmctld slurmctld.hpc.nstda.or.th*
AuthType	*auth/munge*	
CryptoType	*crypto/munge*	
GresTypes		Removed *gpu*, See. gres
JobRequeue	*1*	Automatically requeue batch jobs after node fail or preemption.
LaunchType	*launch/slurm*	
MailProg	`/bin/mail`	
MpiDefault	*pmix*	
PrivateData	*jobs,usage,users*	Prevents users from viewing, jobs, usage of any other user, and information of any user other than themselves.
ProctrackType	*proctrack/cgroup*	The `slurmd` daemon uses this mechanism to identify all processes which are children of processes it spawns for a user job step. **The slurmd daemon must be restarted for a change in ProctrackType to take effect**
SlurmctldPidFile	`/var/run/slurm/slurmctld.pid`	Local file
SlurmctldPort	*6817*	
SlurmdPidFile	`/var/run/slurm/slurmd.pid`	Local file
SlurmdPort	*6818*	
SlurmdSpoolDir	`/var/spool/slurm/slurmd`	Should be local file system
SlurmUser	*slurm*	
SlurmdUser	*root*	
StateSaveLocation	`/var/spool/slurm/slurm.state`	Should be local file system
SwitchType	*switch/none*	
TaskPlugin	*task/affinity,task/cgroup*	See. cgroups
TaskPluginParam	*Sched*	
TopologyPlugin	*topology/tree*	
RoutePlugin	*route/topology*	**[TBD]**
TmpFS	`/tmp`	A node's TmpDisk space
CpuFreqGovernors	*OnDemand, Performance, PowerSave, UserSpace*	See. cpu-governers
CpuFreqDef	*Performance*	Default: Run the CPU at the maximum frequency.

Note: The topology.conf file for an Infiniband switch can be automatically generated using the slurmibtopology tool found here: https://ftp.fysik.dtu.dk/Slurm/slurmibtopology.sh

Job Scheduling

Config	Value	Detail
FastSchedule	*1*	
SchedulerType	*sched/backfill*	
SchedulerParameters		
SelectType	*select/cons_res*	See. Consumable Resources in Slurm
SelectTypeParameters	*CR_Socket_Memory*	Sockets and memory are consumable resources.
KillWait	*30*	The interval given to a job's processes between the SIGTERM and SIGKILL signals upon reaching its time limit.
OverTimeLimit	*5*	Number of *minutes* by which a job can exceed its time limit before being canceled.
PreemptMode	*REQUEUE*	Preempts jobs by requeuing them (if possible) or canceling them.
PreemptType	*preempt/qos*	Job preemption rules are specified by Quality Of Service (QOS).

Job Priority

Config	Value	Detail
PriorityType	*priority/multifactor*	See. Multifactor plugin
PriorityDecayHalfLife	*7-0*	The impact of historical usage (for fare share) is decayed every 7 days.
PriorityCalcPeriod	*5*	Halflife decay wii be re-calculated every 5 minutes
PriorityFavorSmall	*NO*	Larger job will have higher priority. Allocating whole machine will result in the 1.0 job size factor.
PriorityFlags	TBD	
PriorityMaxAge	*7-0*	Job will get maximum age factor (1.0) when it reside in the queue for more than 7 days.
PriorityUsageResetPeriod	*NONE*	Never clear historic usage
PriorityWeightAge	*1000*	
PriorityWeightFairshare	*10000*	
PriorityWeightJobSize	*1000*	
PriorityWeightPartition	*1000*	
PriorityWeightQOS	*1000*	
PriorityWeightTRES		

- If **PriorityFavorSmall** is set to *YES*, the single node job will receive the 1.0 job size factor
- **[TBD]** Some interesting values for **PriorityFlags**
  - *ACCRUE_ALWAYS*: Priority age factor will be increased despite job dependencies or holds.

    This could be beneficial for BioBank job where jobs have dependencies, so the dependent jobs could run as soon as the prior job is finished due to high age factor. However, users could abuse this system by adding a lot of job and hold them to increase age factor.

- *SMALL_RELATIVE_TO_TIME*: The job's size component will be based upon the the job size divided by the time limit.

  In layman's terms, a job with *large allocation and short walltime* will be more preferrable. This could promote a better user behavior, since users who have better estimation of their need will get a better priority and will eventually encourage users to parallelize their programs. However, serial programs, e.g. MATLAB if limited by the license, with a long running time will face a problem when trying to run on the system. Such problem could be solved by having a specialized partition, with high enough priority to compensate for the job size, for serial jobs.

Health Check

Config	Value	Detail
HealthCheckProgram	`/usr/sbin/nhc`	`nhc` can be installed from https://github.com/mej/nhc. For more information See. [1] and [2]
HealthCheckInterval	*3600*	
HealthCheckNodeState	*ANY*	Run on nodes in any state.

Should we set **HealthCheckNodeState** to *IDLE* to avoid performance impact?
Other possible values: *ALLOC*, *MIXED*

Warning: According to this documentation, there are some bugs in `nhc` version 1.4.2.

Logging and Accounting

Config	Value	Detail
AccountingStorageType	*accounting_storage/slurmdbd*	
AccountingStorageHost	*slurmdbd*	
AccountingStoragePort	*6819*	
AccountingStoreJobComment	*YES*	
AccountingStorageEnforce	*associations*	Enforce following job submission policies. • **associations**: No new job is allowed to run unless a corresponding association exists in the system.
ClusterName	*tara*	
JobCompType	*jobcomp/filetxt*	If using the accounting infrastructure this plugin may not be of interest since the information here is redundant.
JobAcctGatherFrequency	*30*	
JobAcctGatherType	*jobacct_gather/linux*	
SlurmctldLogFile	`/var/log/slurm/slurmctld.log`	
SlurmdLogFile	`/var/log/slurm/slurmd.log`	
SlurmSchedLogFile	`/var/log/slurm/slurmsched.log`	
SlurmSchedLogLevel	*1*	Enable scheduler logging
AccountingStorageTRES		**[TBD]** Default: Billing, CPU, Energy, Memory, Node, and FS/Disk. Possible addition: GRES and license.
AcctGatherEnergyType	*acct_gather_energy/ipmi*	**[TBD]** For energy consumption accounting. Only in case of exclusive job allocation the energy consumption measurements will reflect the jobs real consumption

Prolog and Epilog Scripts

Config	Value	Detail
Prolog		
Epilog		
PrologFlags	*contain*	
PrologSlurmctld		Executed once on the ControlMachine for each job
EpilogSlurmctld		Executed once on the ControlMachine for each job

- `pam_slurm_adopt`: `PrologFlags=contain` must be set in `slurm.conf`. This sets up the "extern" step into which ssh-launched processes will be adopted. For further discussion See. Issue 4098.

Node Configuration

Node Class	NodeName	Notes
freeipa	-	
slurmctld	slurmctld	
slurmdbd	slurmdbd	
mysql	-	
frontend	-	
compute	tara-c-[001-006]	
memory	tara-m-[001-002]	FAT nodes
dgx	tara-dgx1-[001-002]	dgx1 is reserved.

Warning: Changes in node configuration (e.g. adding nodes, changing their processor count, etc.) require restarting both the `slurmctld` daemon and the `slurmd` daemons.

NodeName: The name used by all Slurm tools when referring to the node
NodeAddr: The name or IP address Slurm uses to communicate with the node
NodeHostname: The name returned by the command `/bin/hostname -s`

TmpDisk: Total size of temporary disk storage in **TmpFS** in megabytes (e.g. "16384"). *TmpFS* (for "Temporary File System") identifies the location which jobs should use for temporary storage. Note this does not indicate the amount of free space available to the user on the node, only the total file system size. *The system administration should ensure this file system is purged as needed so that user jobs have access to most of this space.* The Prolog and/or Epilog programs (specified in the configuration file) might be used to ensure the file system is kept clean.

Note: `slurmd -C` command can be used to print hardware configuration of a compute node in `slurm.conf` compatible format

slurm.conf

```
# COMPUTE NODES
NodeName=tara-c-[001-006] CPUs=4 RealMemory=512 Sockets=2 CoresPerSocket=2␣
↪ThreadsPerCore=1 State=UNKNOWN TmpDisk=256
NodeName=tara-m-[001-002] CPUs=8 RealMemory=1024 Sockets=2 CoresPerSocket=4␣
↪ThreadsPerCore=1 State=UNKNOWN TmpDisk=512
NodeName=tara-dgx1-[001-002] CPUs=4 RealMemory=1024 Sockets=2 CoresPerSocket=2␣
↪ThreadsPerCore=1 State=UNKNOWN TmpDisk=512
# NodeName=tara-dgx1-[001-002] CPUs=4 RealMemory=1024 Sockets=2 CoresPerSocket=2␣
↪ThreadsPerCore=1 Gres=gpu:volta:8 State=UNKNOWN TmpDisk=512
```

Partitions

Partition	AllocNodes	MaxTime	State	Additional Parameters
debug (default)	tara-c-[001-002]	02:00:00	UP	DefaultTime=00:30:00
standby	tara-c-[001-006]	120:00:00	UP	
memory	tara-m-[001-002]	120:00:00	UP	
dgx	tara-dgx1-002	120:00:00	UP	OverSubscribe=EXCLUSIVE
biobank	tara-dgx1-001	UNLIMITED	UP	AllowGroups=biobank OverSubscribe=EXCLUSIVE

AllowAccounts: Comma separated list of accounts which may execute jobs in the partition. The default value is "ALL"

AllowGroups: Comma separated list of group names which may execute jobs in the partition. If at least one group associated with the user attempting to execute the job is in AllowGroups, he will be permitted to use this partition. Jobs executed as user root can use any partition without regard to the value of AllowGroups.

AllowQos: Comma separated list of Qos which may execute jobs in the partition. Jobs executed as user root can use any partition without regard to the value of AllowQos.

OverSubscribe: Controls the ability of the partition to execute more than one job at a time on each resource. Jobs that run in partitions with `OverSubscribe=EXCLUSIVE` will have exclusive access to all allocated nodes.

`slurm.conf`

```
# PARTITIONS
PartitionName=debug Nodes=tara-c-[001-002] Default=YES MaxTime=02:00:00␣
↪DefaultTime=00:30:00 State=UP
PartitionName=standby Nodes=tara-c-[001-006] MaxTime=120:00:00 State=UP
PartitionName=memory Nodes=tara-m-[001-002] MaxTime=120:00:00 State=UP
PartitionName=dgx Nodes=tara-dgx1-002 MaxTime=120:00:00 State=UP␣
↪OverSubscribe=EXCLUSIVE
PartitionName=biobank Nodes=tara-dgx1-001 MaxTime=120:00:00 State=UP␣
↪AllowGroups=biobank OverSubscribe=EXCLUSIVE
```

Accounting

With the SlurmDBD, accounting is maintained by username (not UID). A username should refer to the same person across all of the computers. Authentication relies upon UIDs, so UIDs must be uniform across all computers

> **Warning:** Only lowercase usernames are supported.

SlurmDBD Configuration

SlurmDBD configuration is stored in a configuration file `slurmdbd.conf`. This file should be only on the computer where SlurmDBD executes and should only be readable by the user which executes SlurmDBD.

Config	Value	Detail
AuthType	*auth/munge*	
DbdHost	*slurmdbd*	The name of the machine where the `slurmdbd` daemon is executed
DbdPort	*6819*	The port number that the `slurmdbd` listens to for work. This value must be equal to the **AccountingStoragePort** parameter in the `slurm.conf` file.
LogFile	*/var/log/slurm/slurmdbd.log*	
PidFile	*/var/run/slurm/slurmdbd.pid*	
SlurmUser	*slurm*	The name of the user that the `slurmctld` daemon executes as. The user must have the same UID as the hosts on which `slurmctld` execute.
StorageHost	*mysql*	
StorageLoc		The default database is `slurm_acct_db`
StoragePass		
StoragePort		
StorageType	*accounting_storage/mysql*	
StorageUser	*slurmdbd*	

Warning: `slurmdbd` must be responding when `slurmctld` is first started.

For `slurmctld` accounting configuration See. slurmctld-logging-accounting

MPI

We will support only MPI libraries and versions that support `PMIx` APIs as follow

- OpenMPI
- MPICH (version 3) (Do we need MPICH2 ?)
- IntelMPI

Generic Resource (GRES) Scheduling

Since we require the DGX-1 node to be exclusively allocated, there is no need for GRES.

For more information, see. DGX Best Practice

Warning: `gres.conf` will always be located in the **same directory** as the `slurm.conf` file.

Topology

In the production system, this script will be used for generating `topology.conf` and we will manually edit the file as needed.

> **Warning:** `topology.conf` will always be located in the same directory as the `slurm.conf` file.

Cgroups

```
###
# cgroup.conf
# Slurm cgroup support configuration file
###
CgroupAutomount=yes
#
TaskAffinity=no
ConstrainCores=yes
ConstrainRAMSpace=yes
```

Note: Slurm documentation recommends stacking *task/affinity,task/cgroup* together when configuring **TaskPlugin**, and setting `TaskAffinity=no` and `ConstrainCores=yes` in `cgroup.conf`. This setup uses the *task/affinity* plugin for setting the affinity of the tasks and uses the *task/cgroup* plugin to fence tasks into the specified resources, thus combining the best of both pieces.

> **Warning:** `cgroup.conf` will always be located in the same directory as the `slurm.conf` file.

Job Preemption

Tara configuration set **PreemptType** to *preempt/qos*, which will use QOS to determine job preemption.

To add a QOS named `biobank-preempt`, use following `sacctmgr` command

```
sacctmgr add qos biobank-preempt PreemptMode=REQUEUE
```

`PreemptMode=REQUEUE` indicates that a job with this QOS will be requeued after preempt.

To add a QOS named `biobank`, which has **Priority** value of 100 and coule preempts a job with `biobank-preempt` QOS.

```
sacctmgr add qos biobank Priority=100 set Preempt=biobank-preempt
```

Notes

CPU Frequency Governer

From https://wiki.archlinux.org/index.php/CPU_frequency_scaling#Scaling_governors

Governor	Description
Performance	Run the CPU at the maximum frequency.
PowerSave	Run the CPU at the minimum frequency.
OnDemand	Scales the frequency dynamically according to current load. Jumps to the highest frequency and then possibly back off as the idle time increases.
UserSpace	Run the CPU at user specified frequencies.
Conservative (not used)	Scales the frequency dynamically according to current load. Scales the frequency more gradually than ondemand.

- Configure SLURM PAM module to limit access to allocated compute nodes.
  - On job termination, any processes initiated by the user outside of Slurm's control may be killed using an `Epilog` script configured in `slurm.conf`.

1.1.5 Appendices

List of Installed Packages

MUNGE installation

```
$ yum install rpm-build wget

Installing:
 rpm-build                  x86_64     4.11.3-32.el7              base        147 k
 wget                       x86_64     1.14-15.el7_4.1            base        547 k
Installing for dependencies:
 bzip2                      x86_64     1.0.6-13.el7               base         52 k
 dwz                        x86_64     0.11-3.el7                 base         99 k
 elfutils                   x86_64     0.170-4.el7                base        282 k
 gdb                        x86_64     7.6.1-110.el7              base        2.4 M
 patch                      x86_64     2.7.1-10.el7_5             updates     110 k
 perl                       x86_64     4:5.16.3-292.el7           base        8.0 M
 perl-Carp                  noarch     1.26-244.el7               base         19 k
 perl-Encode                x86_64     2.51-7.el7                 base        1.5 M
 perl-Exporter              noarch     5.68-3.el7                 base         28 k
 perl-File-Path             noarch     2.09-2.el7                 base         26 k
 perl-File-Temp             noarch     0.23.01-3.el7              base         56 k
 perl-Filter                x86_64     1.49-3.el7                 base         76 k
 perl-Getopt-Long           noarch     2.40-3.el7                 base         56 k
 perl-HTTP-Tiny             noarch     0.033-3.el7                base         38 k
 perl-PathTools             x86_64     3.40-5.el7                 base         82 k
 perl-Pod-Escapes           noarch     1:1.04-292.el7             base         51 k
 perl-Pod-Perldoc           noarch     3.20-4.el7                 base         87 k
 perl-Pod-Simple            noarch     1:3.28-4.el7               base        216 k
 perl-Pod-Usage             noarch     1.63-3.el7                 base         27 k
 perl-Scalar-List-Utils     x86_64     1.27-248.el7               base         36 k
 perl-Socket                x86_64     2.010-4.el7                base         49 k
 perl-Storable              x86_64     2.45-3.el7                 base         77 k
 perl-Text-ParseWords       noarch     3.29-4.el7                 base         14 k
 perl-Thread-Queue          noarch     3.02-2.el7                 base         17 k
 perl-Time-HiRes            x86_64     4:1.9725-3.el7             base         45 k
 perl-Time-Local            noarch     1.2300-2.el7               base         24 k
 perl-constant              noarch     1.27-2.el7                 base         19 k
 perl-libs                  x86_64     4:5.16.3-292.el7           base        688 k
 perl-macros                x86_64     4:5.16.3-292.el7           base         43 k
```

(continues on next page)

(continued from previous page)

```
 perl-parent                noarch     1:0.225-244.el7            base         12 k
 perl-podlators             noarch     2.5.1-3.el7                base        112 k
 perl-srpm-macros           noarch     1-8.el7                    base        4.6 k
 perl-threads               x86_64     1.87-4.el7                 base         49 k
 perl-threads-shared        x86_64     1.43-6.el7                 base         39 k
 redhat-rpm-config          noarch     9.1.0-80.el7.centos        base         79 k
 unzip                      x86_64     6.0-19.el7                 base        170 k
 zip                        x86_64     3.0-11.el7                 base        260 k
```

```
$ yum install gcc bzip2-devel openssl-devel zlib-devel

Installing:
 gcc                       x86_64       4.8.5-28.el7_5.1        updates        16 M
 gcc-c++                   x86_64       4.8.5-28.el7_5.1        updates       7.2 M
 bzip2-devel               x86_64       1.0.6-13.el7            base         218 k
 openssl-devel             x86_64       1:1.0.2k-12.el7         base         1.5 M
 zlib-devel                x86_64       1.2.7-17.el7            base          50 k
Installing for dependencies:
 cpp                       x86_64       4.8.5-28.el7_5.1        updates       5.9 M
 glibc-devel               x86_64       2.17-222.el7            base         1.1 M
 glibc-headers             x86_64       2.17-222.el7            base         678 k
 kernel-headers            x86_64       3.10.0-862.14.4.el7     updates       7.1 M
 libmpc                    x86_64       1.0.1-3.el7             base          51 k
 libstdc++-devel           x86_64       4.8.5-28.el7_5.1        updates       1.5 M
 mpfr                      x86_64       3.1.1-4.el7             base         203 k
 keyutils-libs-devel       x86_64       1.5.8-3.el7             base          37 k
 krb5-devel                x86_64       1.15.1-19.el7           updates       269 k
 libcom_err-devel          x86_64       1.42.9-12.el7_5         updates        31 k
 libselinux-devel          x86_64       2.5-12.el7              base         186 k
 libsepol-devel            x86_64       2.5-8.1.el7             base          77 k
 libverto-devel            x86_64       0.2.5-4.el7             base          12 k
 pcre-devel                x86_64       8.32-17.el7             base         480 k
```

Slurm Installation

```
$ yum install libtool libevent-devel

Installing:
 libtool                   x86_64         2.4.2-22.el7_3         base         588 k
 libevent-devel            x86_64         2.0.21-4.el7           base           85 k
Installing for dependencies:
 autoconf                  noarch         2.69-11.el7            base         701 k
 automake                  noarch         1.13.4-3.el7           base         679 k
 m4                        x86_64         1.4.16-10.el7          base         256 k
 perl-Data-Dumper          x86_64         2.145-3.el7            base          47 k
 perl-Test-Harness         noarch         3.28-3.el7             base         302 k
```

```
$ yum install readline-devel perl-ExtUtils-MakeMaker pam-devel hwloc-devel freeipmi-
↪devel lua-devel mysql-devel libssh2-devel

Installing:
 perl-ExtUtils-MakeMakernoarch        6.68-3.el7               base        275 k
 readline-devel         x86_64        6.2-10.el7               base        138 k
 hwloc-devel            x86_64        1.11.8-4.el7             base        208 k
 lua-devel              x86_64        5.1.4-15.el7             base         21 k
 mariadb-devel          x86_64        1:5.5.60-1.el7_5         updates     754 k
```

(continues on next page)

(continued from previous page)

```
 pam-devel              x86_64        1.1.8-22.el7             base        184 k
 libssh2-devel          x86_64        1.4.3-10.el7_2.1         base         54 k
 freeipmi-devel         x86_64        1.5.7-2.el7              base        260 k
Installing for dependencies:
 gdbm-devel             x86_64        1.10-8.el7               base         47 k
 libdb-devel            x86_64        5.3.21-24.el7            base         38 k
 ncurses-devel          x86_64        5.9-14.20130511.el7_4    base        712 k
 perl-ExtUtils-Install  noarch        1.58-292.el7             base         74 k
 perl-ExtUtils-Manifest noarch        1.61-244.el7             base         31 k
 perl-ExtUtils-ParseXS  noarch        1:3.18-3.el7             base         77 k
 perl-devel             x86_64        4:5.16.3-292.el7         base        453 k
 pyparsing              noarch        1.5.6-9.el7              base         94 k
 systemtap-sdt-devel    x86_64        3.2-8.el7_5              updates      73 k
 hwloc-libs             x86_64        1.11.8-4.el7             base        1.6 M
 ibacm                  x86_64        15-7.el7_5               updates      77 k
 libibcm                x86_64        15-7.el7_5               updates      15 k
 libibumad              x86_64        15-7.el7_5               updates      22 k
 libibverbs             x86_64        15-7.el7_5               updates     224 k
 librdmacm              x86_64        15-7.el7_5               updates      61 k
 libtool-ltdl           x86_64        2.4.2-22.el7_3           base         49 k
 rdma-core-devel        x86_64        15-7.el7_5               updates     203 k
 OpenIPMI-modalias      x86_64        2.0.23-2.el7             base         16 k
 freeipmi               x86_64        1.5.7-2.el7              base        2.0 M
```

Lmod Installation

```
$ yum install lmod

Installing:
 Lmod                    x86_64          6.6.3-1.el7              epel          192 k
Installing for dependencies:
 lua-bitop               x86_64          1.0.2-3.el7              epel          7.9 k
 lua-filesystem          x86_64          1.6.2-2.el7              epel           28 k
 lua-json                noarch          1.3.2-2.el7              epel           23 k
 lua-lpeg                x86_64          0.12-1.el7               epel           59 k
 lua-posix               x86_64          32-2.el7                 epel          116 k
 lua-term                x86_64          0.03-3.el7               epel           10 k
 tcl                     x86_64          1:8.5.13-8.el7           base          1.9 M
```

Git Installation

```
$ yum install git

Installing:
 git                     x86_64        1.8.3.1-14.el7_5        updates        4.4 M
Installing for dependencies:
 libgnome-keyring        x86_64        3.12.0-1.el7            base           109 k
 perl-Error              noarch        1:0.17020-2.el7         base            32 k
 perl-Git                noarch        1.8.3.1-14.el7_5        updates         54 k
 perl-TermReadKey        x86_64        2.30-20.el7             base            31 k
 rsync                   x86_64        3.1.2-4.el7             base           403 k
```

1.2 Manuals

1.2.1 Quick Reference

xCat

Show status of a `node` or nodes in `group`.

```
lsdef <node/group> | egrep 'Object|status=|currstate'
```

Postscript path

```
/install/postscripts/
```

Running postscripts

```
updatenode <node/group> <script>
```

freeIPA

Show all registered host

```
ipa host-find
```

Register a list of hosts

```
for n in `seq -w 001 060`
do
    echo $n
    ipa host-add --force --password=1q2w3e4r tara-c-$n-node-ib.tara.nstda.or.th
done
```

Delete a list of hosts

```
for n in `seq -w 001 060`
do
    echo $n
    ipa host-del tara-c-$n-node-ib.tara.nstda.or.th
done
```

Client join

```
ipa-client-install --mkhomedir --domain tara.nstda.or.th --server freeipa.tara.nstda.
↪or.th --ntp-server freeipa.tara.nstda.or.th --force-join --password '1q2w3e4r' --
↪unattended
```

munge

Munge service

```
systemctl enable munge
systemctl start munge
```

Copy `munge.key`

```
# Enable munge service
psh compute systemctl enable munge
psh fat systemctl enable munge
psh gpu systemctl enable munge

# Copy munge.key
psh tara-c-[001-010]-node scp tara-xcat:/etc/munge/munge.key /etc/munge/munge.key
psh tara-c-[011-020]-node scp tara-xcat:/etc/munge/munge.key /etc/munge/munge.key
psh tara-c-[021-030]-node scp tara-xcat:/etc/munge/munge.key /etc/munge/munge.key
psh tara-c-[031-040]-node scp tara-xcat:/etc/munge/munge.key /etc/munge/munge.key
psh tara-c-[041-050]-node scp tara-xcat:/etc/munge/munge.key /etc/munge/munge.key
psh tara-c-[051-060]-node scp tara-xcat:/etc/munge/munge.key /etc/munge/munge.key

psh tara-m-[001-010]-node scp tara-xcat:/etc/munge/munge.key /etc/munge/munge.key

psh tara-g-[001-002]-node scp tara-xcat:/etc/munge/munge.key /etc/munge/munge.key

# Check munge.key
psh all sha256sum /etc/munge/munge.key

# Start munge service
psh compute systemctl start munge
psh fat systemctl start munge
psh gpu systemctl start munge

# Test munge connection with frontend-1
psh tara-c-[001-010]-node "munge -n | ssh tara-frontend-1-node unmunge | grep 'ENCODE_
↪HOST'"
psh tara-c-[011-020]-node "munge -n | ssh tara-frontend-1-node unmunge | grep 'ENCODE_
↪HOST'"
psh tara-c-[021-030]-node "munge -n | ssh tara-frontend-1-node unmunge | grep 'ENCODE_
↪HOST'"
psh tara-c-[031-040]-node "munge -n | ssh tara-frontend-1-node unmunge | grep 'ENCODE_
↪HOST'"
psh tara-c-[041-050]-node "munge -n | ssh tara-frontend-1-node unmunge | grep 'ENCODE_
↪HOST'"
psh tara-c-[051-060]-node "munge -n | ssh tara-frontend-1-node unmunge | grep 'ENCODE_
↪HOST'"

psh tara-m-[001-010]-node "munge -n | ssh tara-frontend-1-node unmunge | grep 'ENCODE_
↪HOST'"

psh tara-g-[001-002]-node "munge -n | ssh tara-frontend-1-node unmunge | grep 'ENCODE_
↪HOST'"

# Test munge connection with frontend-2
psh tara-c-[001-010]-node "munge -n | ssh tara-frontend-2-node unmunge | grep 'ENCODE_
↪HOST'"
psh tara-c-[011-020]-node "munge -n | ssh tara-frontend-2-node unmunge | grep 'ENCODE_
↪HOST'"
psh tara-c-[021-030]-node "munge -n | ssh tara-frontend-2-node unmunge | grep 'ENCODE_
↪HOST'"
psh tara-c-[031-040]-node "munge -n | ssh tara-frontend-2-node unmunge | grep 'ENCODE_
↪HOST'"
psh tara-c-[041-050]-node "munge -n | ssh tara-frontend-2-node unmunge | grep 'ENCODE_
↪HOST'"
psh tara-c-[051-060]-node "munge -n | ssh tara-frontend-2-node unmunge | grep 'ENCODE_
↪HOST'"
```

(continues on next page)

(continued from previous page)

```
psh tara-m-[001-010]-node "munge -n | ssh tara-frontend-2-node unmunge | grep 'ENCODE_
↪HOST'"

psh tara-g-[001-002]-node "munge -n | ssh tara-frontend-2-node unmunge | grep 'ENCODE_
↪HOST'"

# Test munge connection with tara-slurmctl
psh tara-c-[001-010]-node "munge -n | ssh tara-slurmctl unmunge | grep 'ENCODE_HOST'"
psh tara-c-[011-020]-node "munge -n | ssh tara-slurmctl unmunge | grep 'ENCODE_HOST'"
psh tara-c-[021-030]-node "munge -n | ssh tara-slurmctl unmunge | grep 'ENCODE_HOST'"
psh tara-c-[031-040]-node "munge -n | ssh tara-slurmctl unmunge | grep 'ENCODE_HOST'"
psh tara-c-[041-050]-node "munge -n | ssh tara-slurmctl unmunge | grep 'ENCODE_HOST'"
psh tara-c-[051-060]-node "munge -n | ssh tara-slurmctl unmunge | grep 'ENCODE_HOST'"

psh tara-m-[001-010]-node "munge -n | ssh tara-slurmctl unmunge | grep 'ENCODE_HOST'"

psh tara-g-[001-002]-node "munge -n | ssh tara-slurmctl unmunge | grep 'ENCODE_HOST'"
```

Machine	Method
tara-xcat	manual
tara-frontend-[1-2]-node	manual
tara-slurmctl	manual
tara-slurmdb	manual
tara-c-[001-060]-node	`psh`
tara-m-[001-010]-node	`psh`
tara-g-[001-002]-node	`psh`

Slurm

`slurm.conf`

Machine	Location	Method
tara-xcat	tarafs	`/etc/slurm/ -> /tarafs/utils/slurm/`
tara-frontend-[1-2]-node	tarafs	`/etc/slurm/ -> /tarafs/utils/slurm/`
tara-c-[001-060]-node	tarafs	`/etc/slurm/ -> /tarafs/utils/slurm/`
tara-m-[001-010]-node	tarafs	`/etc/slurm/ -> /tarafs/utils/slurm/`
tara-g-[001-002]-node	tarafs	`/etc/slurm/ -> /tarafs/utils/slurm/`
tara-slurmctl	tarafs	`/etc/slurm/ -> /tarafs/utils/slurm/`
tara-slurmdb	**local**	`/etc/slurm/`

`/install/postscripts/confSlurmd`

```
#!/bin/bash

mkdir -p /var/log/slurm/ /var/run/slurm/ /var/spool/slurm/
chown slurm:slurm /var/log/slurm/
chown slurm:slurm /var/run/slurm/
chown slurm:slurm /var/spool/slurm/

echo "d /var/run/slurm 0755 slurm slurm -" > /usr/lib/tmpfiles.d/slurmd.conf
```

(continues on next page)

(continued from previous page)

```
sed -i -e 's@PIDFile=/var/run/slurmd.pid@PIDFile=/var/run/slurm/slurmd.pid@g' /usr/
↪lib/systemd/system/slurmd.service

systemctl enable slurmd
systemctl start slurmd
systemctl status slurmd
```

Setup PAM access control

```
psh tara-c-[001-010]-node scp tara-xcat:sshd /etc/pam.d/sshd
psh tara-c-[011-020]-node scp tara-xcat:sshd /etc/pam.d/sshd
psh tara-c-[021-030]-node scp tara-xcat:sshd /etc/pam.d/sshd
psh tara-c-[031-040]-node scp tara-xcat:sshd /etc/pam.d/sshd
psh tara-c-[041-050]-node scp tara-xcat:sshd /etc/pam.d/sshd
psh tara-c-[051-059]-node scp tara-xcat:sshd /etc/pam.d/sshd
psh tara-m-[001-010]-node scp tara-xcat:sshd /etc/pam.d/sshd
psh tara-g-[001-002]-node scp tara-xcat:sshd /etc/pam.d/sshd

psh tara-c-[001-010]-node scp tara-xcat:access.conf /etc/security/access.conf
psh tara-c-[011-020]-node scp tara-xcat:access.conf /etc/security/access.conf
psh tara-c-[021-030]-node scp tara-xcat:access.conf /etc/security/access.conf
psh tara-c-[031-040]-node scp tara-xcat:access.conf /etc/security/access.conf
psh tara-c-[041-050]-node scp tara-xcat:access.conf /etc/security/access.conf
psh tara-c-[051-060]-node scp tara-xcat:access.conf /etc/security/access.conf
psh tara-m-[001-010]-node scp tara-xcat:access.conf /etc/security/access.conf
psh tara-g-[001-002]-node scp tara-xcat:access.conf /etc/security/access.conf
```

EasyBuild

```
export EASYBUILD_PREFIX=/tarafs/utils/modules

python bootstrap_eb.py $EASYBUILD_PREFIX

export MODULEPATH="/tarafs/utils/modules/modules/all:$MODULEPATH"

chmod a+rx /tarafs/utils/modules
```

`/etc/profile.d/z01_EasyBuild.sh`

```
if [ -z "$__Init_Default_Modules" ]; then
    export __Init_Default_Modules=1
    export EASYBUILD_MODULES_TOOL=Lmod
    export EASYBUILD_PREFIX=/tarafs/utils/modules
    module use $EASYBUILD_PREFIX/modules/all
else
    module refresh
fi
```

```
psh tara-c-[001-010]-node scp tara-xcat:z01_EasyBuild.sh /etc/profile.d/z01_EasyBuild.
↪sh
psh tara-c-[011-020]-node scp tara-xcat:z01_EasyBuild.sh /etc/profile.d/z01_EasyBuild.
↪sh
psh tara-c-[021-030]-node scp tara-xcat:z01_EasyBuild.sh /etc/profile.d/z01_EasyBuild.
↪sh
```

(continues on next page)

(continued from previous page)

```
psh tara-c-[031-040]-node scp tara-xcat:z01_EasyBuild.sh /etc/profile.d/z01_EasyBuild.
↪sh
psh tara-c-[041-050]-node scp tara-xcat:z01_EasyBuild.sh /etc/profile.d/z01_EasyBuild.
↪sh
psh tara-c-[051-060]-node scp tara-xcat:z01_EasyBuild.sh /etc/profile.d/z01_EasyBuild.
↪sh
psh tara-m-[001-010]-node scp tara-xcat:z01_EasyBuild.sh /etc/profile.d/z01_EasyBuild.
↪sh
psh tara-g-[001-002]-node scp tara-xcat:z01_EasyBuild.sh /etc/profile.d/z01_EasyBuild.
↪sh

psh compute "module load EasyBuild && eb --version"
psh fat "module load EasyBuild && eb --version"
psh gpu "module load EasyBuild && eb --version"
```

OpenUCX

```
yum install numactl numactl-libs numactl-devel

export LD_LIBRARY_PATH=/usr/local/cuda-10.0/lib64\
                      ${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}

./contrib/configure-release  --prefix=$PWD/install --with-cuda=/usr/local/cuda/

rpmbuild -bb --define "configure_options --enable-optimizations --with-cuda=/usr/
↪local/cuda" ucx-1.4.0/ucx.spec
```

GPFS

1. deattach GPFS before deploy osimage @ionode

```
mmumount all -N tara-c-055-node-ib.tara.nstda.or.th,tara-c-056-node-ib.tara.nstda.or.
↪th,tara-c-057-node-ib.tara.nstda.or.th,tara-c-058-node-ib.tara.nstda.or.th,tara-c-
↪059-node-ib.tara.nstda.or.th,tara-c-060-node-ib.tara.nstda.or.th
mmshutdown -N tara-c-055-node-ib.tara.nstda.or.th,tara-c-056-node-ib.tara.nstda.or.th,
↪tara-c-057-node-ib.tara.nstda.or.th,tara-c-058-node-ib.tara.nstda.or.th,tara-c-059-
↪node-ib.tara.nstda.or.th,tara-c-060-node-ib.tara.nstda.or.th
mmdelnode -N tara-c-055-node-ib.tara.nstda.or.th,tara-c-056-node-ib.tara.nstda.or.th,
↪tara-c-057-node-ib.tara.nstda.or.th,tara-c-058-node-ib.tara.nstda.or.th,tara-c-059-
↪node-ib.tara.nstda.or.th,tara-c-060-node-ib.tara.nstda.or.th
mmlscluster
```

2. deploy osimage @xcat

```
nodeset tara-c-[001-054]-node osimage=tara-compute-centos7
rsetboot tara-c-[001-054]-node net -u
rpower tara-c-[001-054]-node reset

nodeset dev osimage=tara-dev-centos7
rsetboot dev net -u
rpower dev reset

nodeset tara-c-060-node osimage=tara-build-centos7
```

(continues on next page)

(continued from previous page)

```
rsetboot tara-c-060-node net -u
rpower tara-c-060-node reset


### for watching deploy status
watch -n 1 "lsdef tara-c-[001-054]-node  | egrep 'currstate|Object|status='"

watch -n 1 "lsdef dev | egrep 'currstate|Object|status='"


updatenode tara-c-[055-060]-node -P confGPFS
```

3. attach GPFS after deploy osimage @ionode

```
mmaddnode -N tara-c-055-node-ib.tara.nstda.or.th,tara-c-056-node-ib.tara.nstda.or.th,
↪tara-c-057-node-ib.tara.nstda.or.th,tara-c-058-node-ib.tara.nstda.or.th,tara-c-059-
↪node-ib.tara.nstda.or.th,tara-c-060-node-ib.tara.nstda.or.th
mmlscluster
mmlslicense
mmchlicense client --accept -N tara-c-055-node-ib.tara.nstda.or.th,tara-c-056-node-ib.
↪tara.nstda.or.th,tara-c-057-node-ib.tara.nstda.or.th,tara-c-058-node-ib.tara.nstda.
↪or.th,tara-c-059-node-ib.tara.nstda.or.th,tara-c-060-node-ib.tara.nstda.or.th
mmlslicense
mmstartup -N tara-c-055-node-ib.tara.nstda.or.th,tara-c-056-node-ib.tara.nstda.or.th,
↪tara-c-057-node-ib.tara.nstda.or.th,tara-c-058-node-ib.tara.nstda.or.th,tara-c-059-
↪node-ib.tara.nstda.or.th,tara-c-060-node-ib.tara.nstda.or.th
mmgetstate -N tara-c-055-node-ib.tara.nstda.or.th,tara-c-056-node-ib.tara.nstda.or.th,
↪tara-c-057-node-ib.tara.nstda.or.th,tara-c-058-node-ib.tara.nstda.or.th,tara-c-059-
↪node-ib.tara.nstda.or.th,tara-c-060-node-ib.tara.nstda.or.th
mmmount all -N tara-c-055-node-ib.tara.nstda.or.th,tara-c-056-node-ib.tara.nstda.or.
↪th,tara-c-057-node-ib.tara.nstda.or.th,tara-c-058-node-ib.tara.nstda.or.th,tara-c-
↪059-node-ib.tara.nstda.or.th,tara-c-060-node-ib.tara.nstda.or.th
```

TODO

- Setup `topology.conf`
- Cleanup `tara-frontend-1-node` and `tara-frontend-2-node`
- Missing `libevent-devel` when installing OpenMPI

1.2.2 EasyBuild

Default robot path

```
/utils/modules/software/EasyBuild/3.7.1/lib/python2.7/site-packages/easybuild_
↪easyconfigs-3.7.1-py2.7.egg/easybuild/easyconfigs
```

OpenMPI

The default OpenMPI easyconfig (`.eb` files) does not enable Slurm and PMIx support by default. Following flags must be added to the easyconfig file.

```
configopts += '--with-slurm --with-pmi=/usr/include --with-pmi-libdir=/usr/lib64'
```

Note that different version of OpenMPI could have different `configure` script. For example, `OpenMPI/2.1.2-GCC-6.4.0-2.28` would require

```
configopts += '--with-slurm --with-pmi=/usr --with-pmi-libdir=/usr'
```

OpenMPI installation can be verified by using `ompi_info` command. For example, to verify Slurm and PMIx installation use

```
$ ompi_info | egrep -i 'slurm|pmi'
```

MPICH

MPICH installation can be verified by using `mpichversion` or `mpiexec --version`.

Known Issues

- OpenMPI 2.1.2 is known to have compatibility issue with PMI. OpenMPI 2.1.3 should be used instead.

1.3 Contributing

First off, thank you for considering contributing to our documentation. Please read the following sections in order to know how to ask questions and how to work on something.

1.3.1 Contributing to User Guides and Tutorials

If you are using any specific software on any ThaiSC platform, you might already developed

- a set of scripts to efficiently run the software
- an effient workflow to work with the software
- troubleshooting steps for your software

Then your experiece will be valuable for the other users and we would appreciate your help to complete this document with new topics/entries.

To do that, you first need to *install* this repository to your local machine.

1.3.2 Installing Documentation Repository

Here is a step by step plan on how to install this repository and generate your local copy of this document.

First, obtain Python 3.6 and virtualenv if you do not already have them. Using a virtual environment will make the installation easier, and will help to avoid clutter in your system-wide libraries. You will also need Git in order to clone the repository.

First, you need to clone the repository using following command

```
git clone https://github.com/puttsk/thaisc.git
cd thaisc
```

Next, you will need to verify that your `pip` version is higher by using

```
pip --version
```

If the version is lesser than 18, you should upgrade `pip` before continuing.

```
pip install --upgrade pip
```

Once you have these, create a virtual environment inside the directory, then activate it:

```
virtualenv venv
source venv/bin/activate
```

Next, install the dependencies using `pip`

```
pip install -r requirements.txt
```

The source code of the document is in the `/docs` directory. To build your local document

```
cd docs
make html           # For building HTML document
make latexpdf       # For building PDF document
```

The HTML document will be in `./docs/_build/html/index.html` directory and the PDF document will be in `./docs/_build/latex/ThaiSCDocumentation.pdf`.