BE-SST Installation and Build Guide

BE-SST is a combination of Behavioral Emulation (BE) modeling approach and the parallel discrete-event simulation framework from Sandia National Labs called Structural Simulation Toolkit (SST).

|  |
| --- |
| Refer to this paper to know more about BE-SST simulator: <https://dl.acm.org/citation.cfm?id=3225124&dl=ACM&coll=DL> |
| Ajay Ramaswamy, Nalini Kumar, Aravind Neelakantan, Herman Lam, and Greg Stitt. 2018. Scalable Behavioral Emulation of Extreme-Scale Systems Using Structural Simulation Toolkit. In Proceedings of the 47th International Conference on Parallel Processing (ICPP 2018). ACM, New York, NY, USA, Article 17, 11 pages. DOI: https://doi.org/10.1145/3225058.3225124 |

Majority of the installation instruction has been taken from the SST documentation and it has been modified to cater the setup (installation and build) of BE-SST. Refer to this link foe SST documentation: <http://sst-simulator.org/SSTPages/SSTBuildAndInstall9dot0dot0SeriesDetailedBuildInstructions/>

**Example Build and Install Directories**

These instructions will use the following conventions (the user can adjust these as they see fit):

* Download directory $HOME/scratch
* This directory will contain downloaded source code packages for SST and its dependencies.
* The following directories should be created on the users machine
  + $HOME/scratch
  + $HOME/scratch/src
* Download SST-core and BE element from the given GitHub link inside the $HOME/scratch/src directory:
* Installation directory $HOME/local
* This directory will be the installation directory for SST and its dependencies.
* The following directories should be created on the users machine
  + $HOME/local
  + $HOME/local/packages

### External Dependent Components

#### OpenMPI 2.1.3 (Strongly Recommended)

SST core is regularly tested with OpenMPI 2.1.3, and this version is known to work with SST. Installation of an MPI package from source code is typically unnecessary, since many Linux distributions provide OpenMPI as an optional installable package.

Refer to the SST documentation link shown above to install OpenMPI 2.1.3.

#### Boost 1.56 (Required for BE)

BE requires Boost library to be installed and build. Boost 1.56 can be obtained online at <http://sourceforge.net/projects/boost/files/boost/1.56.0/>

Refer to the SST documentation link shown above to install boost 1.56.

### SST CORE Build and Installation

After the prerequisite Components (OpenMPI and boost) have been successfully installed, the SST-Core can then be built and installed.

1. Navigate to SST-Core directory in $HOME/scratch/src.

$ cd $HOME/scratch/src/sst-core-devel

2. Set the home directory environment variable to the SST-Core installation (i.e. where you want the SST-Core files installed).

$ export SST\_CORE\_HOME=$HOME/local/sst-core-devel

3. Run the autogen.sh script to setup the configure for SST-core

$ ./autogen.sh

4. Configure SST-Core, being sure to make configure reference the location of SST-Core’s local prerequisite packages.

$ ./configure --prefix=$SST\_CORE\_HOME [other configure settings as needed]

5. Build and install SST-CORE

$ make all

$ make install

6. Update your PATH environment variable so that it contains the bin directory of the SST-CORE installation location.

$ export PATH=$SST\_CORE\_HOME/bin:$PATH

7. Test your sst-core install

$ which sst

$ sst --version

$ sst-info

8. To make the changes to the environment variables a permanent change, it may require editing of your login shell’s initialization file.

### SST Elements – BE Build and Installation

After [**SST-Core**](http://sst-simulator.org/SSTPages/SSTBuildAndInstall9dot0dot0SeriesDetailedBuildInstructions/#SSTCore) has been successfully installed, Behavioral Emulation (BE) can then be built and installed.

1. Navigate to the SST-element-BE directory in $HOME/scratch/src.

$ cd $HOME/scratch/src/sst

2. Set the home directory environment variable to the SST-Elements installation (i.e. where you want the SST-Elements files installed).

$ export SST\_ELEMENTS\_HOME=$HOME/local/sstelements-devel

3. Run the autogen.sh script to setup the configure for BE

$ ./autogen.sh

4. Configure SST-Elements, being sure to make configure reference the location of SST-Element-BE’s local prerequisite packages.

$ ./configure --prefix=$SST\_ELEMENTS\_HOME --with-sst-core=$SST\_CORE\_HOME –with-boost=$BOOST\_HOME [other configure settings as needed]

5. Build and install SST-ELEMENTS (This will register SST-Elements with the SST-Core)

$ make all

$ make install

6. Update your PATH environment variable so that it contains the bin directory of the SST-ELEMENTS installation location.

$ export PATH=$SST\_ELEMENTS\_HOME/bin:$PATH

7. To make the changes to the environment variables a permanent change, it may require editing of your login shell’s initialization file.

## Testing BE-SST Functionality

1. Run a very simple sanity test

$ sst --version

2. Run a very simple simulation

$ cd $HOME/source/src/sst-elements-BE/src/sst/elements/behavioralEmulation/tests

$ ./BE\_RUN.sh -c vulcan-proper.py