COMP Superscalar

Installation Manual

Version: 1.3

August 19, 2015

This manual only provides information about how to install and configure COMPSs. Specifically, it details the installation process for Debian based distributions and for RedHat based distributions, and the steps to configure COMPSs properly.

If you are not wondering to install COMPSs please consider using our already prepared *Virtual Machine* available at our webpage: `http://compss.bsc.es` .

For further information about the application execution please refer to the *COMPSs User Manual: Application execution guide* available at `http://compss.bsc.es` .

For further information about the application development please refer to the *COMPSs User Manual: Application development guide* available at `http://compss.bsc.es/` .

For full COMPSs application examples (codes, execution commands, results, logs, etc.) please refer to the *COMPSs Sample Applications* available at `http://compss.bsc.es/` .

Contents

1 COMP Superscalar (COMPSs)

COMP Superscalar (COMPSs) is a programming model which aims to ease the development of applications for distributed infrastructures, such as Clusters, Grids and Clouds. COMP Superscalar also features a runtime system that exploits the inherent parallelism of applications at execution time.

For the sake of programming productivity, the COMPSs model has four key characteristics:

- **Sequential programming:** COMPSs programmers do not need to deal with the typical duties of parallelization and distribution, such as thread creation and synchronization, data distribution, messaging or fault tolerance. Instead, the model is based on sequential programming, which makes it appealing to users that either lack parallel programming expertise or are looking for better programmability.

- **Infrastructure unaware:** COMPSs offers a model that abstracts the application from the underlying distributed infrastructure. Hence, COMPSs programs do not include any detail that could tie them to a particular platform, like deployment or resource management. This makes applications portable between infrastructures with diverse characteristics.

- **Standard programming languages:** COMPSs is based on the popular programming language Java, but also offers language bindings for Python and C/C++ applications. This facilitates the learning of the model, since programmers can reuse most of their previous knowledge.

- **No APIs:** In the case of COMPSs applications in Java, the model does not require to use any special API call, pragma or construct in the application; everything is pure standard Java syntax and libraries. With regard the Python and C/C++ bindings, a small set of API calls should be used on the COMPSs applications.

2 Debian-based distributions

2.1 Prerequisites

The commands described on the following sections require root privileges and Internet connection.

Once the installation process is finished, please log out and back in again to complete the installation.

2.2 Package Repository

To add the package repository you can easily download our predefined lists by executing the following command:

```
x86_64 :
   wget http://compss.bsc.es/releases/repofiles/repo_deb_x86-64.list -O /etc/apt/sources.
    list.d/compss-framework_x86-64.list

noarch :
   wget http://compss.bsc.es/releases/repofiles/repo_deb_noarch.list -O /etc/apt/sources.
    list.d/compss-framework_noarch.list
```

Next you need to add the repository key by executing:

```
wget -qO - http://compss.bsc.es/repo/debs/deb-gpg-bsc-grid.pub.key | apt-key add -
```

And finally, refresh the apt-get repositories:

```
apt-get update
```

2.3 Installation

Despite the fact that we recomend users to install the complete COMPSs Framework, we have built different packages to allow users customize as maximum as possible their installation. Next we describe the available packages and how to install them.

If you are willing to have a full COMPSs installation just follow the COMPSs Framework instructions and skip to next section.

- **COMPSs Framework**
  Contains the all COMPSs functionalities including the Runtime, all the bindings, all the tools and the cloud connectors.
  To install this package please run:

```
apt-get install compss-framework
```

- **COMPSs Runtime**
  Contains the COMPSs runtime to support the native functionalities. Install this package if you only need to support Java applications.
  To install this package please run:

```
apt-get install compss-runtime
```

  This package is composed of two sub-packages:

  - **COMPSs Engine**
    Contains the COMPSs Engine, essential to run COMPSs applications as master.
    To install this package please run:

```
apt-get install compss-engine
```

  - **COMPSs Worker**
    Contains the minimum installation to allow any machine run as a COMPSs worker.
    To install this package please run:

```
apt-get install compss-worker
```

- **COMPSs Bindings**
  Contains all the bindings to support C/C++ and Python applications.
  To install this package please run:

```
apt-get install compss-bindings
```

  This package is composed of three sub-packages:

  - **COMPSs Bindings Common**
    Contains the API to allow any binding communicate with the COMPSs Runtime. It is necessary for any binding installation.
    To install this package please run:

```
apt-get install compss-bindings-common
```

  - **COMPSs C/C++ Binding**
    Contains the C/C++ Binding
    To install this package please run:

```
apt-get install compss-c-binding
```

  - **COMPSs Python Binding**
    Contains the Python Binding
    To install this package please run:

```
apt-get install compss-python-binding
```

- **COMPSs Tools**
  Contains all the COMPSs Tools.
  To install this package please run:

```
apt-get install compss-tools
```

  This package is composed of three sub-packages:

  - **COMPSs Extrae**
    Contains the COMPSs Extrae tool needed to generate and process application traces.
    To install this package please run:

```
apt-get install compss-extrae
```

  - **COMPSs Monitor**
    Contains the COMPSs Monitor tool needed to monitor the application execution.
    To install this package please run:

```
apt-get install compss-monitor
```

- **COMPSs Cloud**
  Contains all the COMPSs Connectors needed to interact with the Cloud.
  To install this package please run:

```
apt-get install compss-cloud
```

2.4 Post installation

Once your COMPSs package has been installed remember to log out and back in again to end the installation process.

If you need to setup your machine for the first time please take a look at Section 5 for a detailed description of the additional configuration.

3 RedHat-based distributions (zypper)

3.1 Prerequisites

The commands described on the following sections require root privileges and Internet connection.

Once the installation process is finished, please log out and back in again to complete the installation.

3.2 Package Repository

To add the package repository you can easily download our predefined lists by executing the following command:

```
x86_64      :  zypper addrepo -f http://compss.bsc.es/repo/rpms/stable/suse/x86_64 compss
noarch      :  zypper addrepo -f http://compss.bsc.es/repo/rpms/stable/suse/noarch compss
```

And finally, refresh the apt-get repositories:

```
zypper refresh
```

3.3 Installation

Despite the fact that we recomend users to install the complete COMPSs Framework, we have built different packages to allow users customize as maximum as possible their installation. Next we describe the available packages and how to install them.

If you are willing to have a full COMPSs installation just follow the COMPSs Framework instructions and skip to next section.

- **COMPSs Framework**
  Contains the all COMPSs functionalities including the Runtime, all the bindings, all the tools and the cloud connectors.
  To install this package please run:

  ```
  zypper install compss-framework
  ```

- **COMPSs Runtime**
  Contains the COMPSs runtime to support the native functionalities. Install this

package if you only need to support Java applications.
To install this package please run:

```
zypper install compss-runtime
```

This package is composed of two sub-packages:

- **COMPSs Engine**
  Contains the COMPSs Engine, essential to run COMPSs applications as master.
  To install this package please run:

  ```
  zypper install compss-engine
  ```

- **COMPSs Worker**
  Contains the minimum installation to allow any machine run as a COMPSs worker.
  To install this package please run:

  ```
  zypper install compss-worker
  ```

- **COMPSs Bindings**
  Contains all the bindings to support C/C++ and Python applications.
  To install this package please run:

  ```
  zypper install compss-bindings
  ```

  This package is composed of three sub-packages:

  - **COMPSs Bindings Common**
    Contains the API to allow any binding communicate with the COMPSs Runtime. It is necessary for any binding installation.
    To install this package please run:

    ```
    zypper install compss-bindings-common
    ```

  - **COMPSs C/C++ Binding**
    Contains the C/C++ Binding
    To install this package please run:

```
zypper install compss-c-binding
```

  - **COMPSs Python Binding**
    Contains the Python Binding
    To install this package please run:

```
zypper install compss-python-binding
```

- **COMPSs Tools**
  Contains all the COMPSs Tools.
  To install this package please run:

```
zypper install compss-tools
```

  This package is composed of three sub-packages:

  - **COMPSs Extrae**
    Contains the COMPSs Extrae tool needed to generate and process application traces.
    To install this package please run:

```
zypper install compss-extrae
```

  - **COMPSs Monitor**
    Contains the COMPSs Monitor tool needed to monitor the application execution.
    To install this package please run:

```
zypper install compss-monitor
```

- **COMPSs Cloud**
  Contains all the COMPSs Connectors needed to interact with the Cloud.
  To install this package please run:

```
zypper install compss-cloud
```

3.4 Post installation

Once your COMPSs package has been installed remember to log out and back in again to end the installation process.

If you need to setup your machine for the first time please take a look at Section 5 for a detailed description of the additional configuration.

4 RedHat-based distributions (yum)

4.1 Prerequisites

The commands described on the following sections require root privileges and Internet connection.

Once the installation process is finished, please log out and back in again to complete the installation.

4.2 Package Repository

To add the package repository you can easily download our predefined lists by executing the following command:

```
x86_64 :
   wget http://compss.bsc.es/releases/repofiles/repo_rpm_centos_x86-64.repo -O /etc/yum.
    repos.d/compss-framework_x86-64.repo

noarch :
   wget http://compss.bsc.es/releases/repofile/repo_rpm_centos_noarch.repo -O /etc/yum.
    repos.d/compss-framework_noarch.repo
```

4.3 Installation

Despite the fact that we recomend users to install the complete COMPSs Framework, we have built different packages to allow users customize as maximum as possible their installation. Next we describe the available packages and how to install them.

If you are willing to have a full COMPSs installation just follow the COMPSs Framework instructions and skip to next section.

- **COMPSs Framework**
  Contains the all COMPSs functionalities including the Runtime, all the bindings, all the tools and the cloud connectors.
  To install this package please run:

```
   yum install compss-framework
```

- **COMPSs Runtime**
  Contains the COMPSs runtime to support the native functionalities. Install this package if you only need to support Java applications.
  To install this package please run:

```
yum install compss-runtime
```

This package is composed of two sub-packages:

- **COMPSs Engine**
  Contains the COMPSs Engine, essential to run COMPSs applications as master.
  To install this package please run:

```
yum install compss-engine
```

- **COMPSs Worker**
  Contains the minimum installation to allow any machine run as a COMPSs worker.
  To install this package please run:

```
yum install compss-worker
```

- **COMPSs Bindings**
  Contains all the bindings to support C/C++ and Python applications.
  To install this package please run:

```
yum install compss-bindings
```

This package is composed of three sub-packages:

- **COMPSs Bindings Common**
  Contains the API to allow any binding communicate with the COMPSs Runtime. It is necessary for any binding installation.
  To install this package please run:

```
yum install compss-bindings-common
```

- **COMPSs C/C++ Binding**
  Contains the C/C++ Binding
  To install this package please run:

```
yum install compss-c-binding
```

  - **COMPSs Python Binding**
    Contains the Python Binding
    To install this package please run:

```
yum install compss-python-binding
```

- **COMPSs Tools**
  Contains all the COMPSs Tools.
  To install this package please run:

```
yum install compss-tools
```

  This package is composed of three sub-packages:

  - **COMPSs Extrae**
    Contains the COMPSs Extrae tool needed to generate and process application traces.
    To install this package please run:

```
yum install compss-extrae
```

  - **COMPSs Monitor**
    Contains the COMPSs Monitor tool needed to monitor the application execution.
    To install this package please run:

```
yum install compss-monitor
```

- **COMPSs Cloud**
  Contains all the COMPSs Connectors needed to interact with the Cloud.
  To install this package please run:

```
yum install compss-cloud
```

4.4 Post installation

Once your COMPSs package has been installed remember to log out and back in again to end the installation process.

If you need to setup your machine for the first time please take a look at Section 5 for a detailed description of the additional configuration.

5 Additional Configuration

5.1 Configure SSH passwordless

By default, COMPSs uses SSH libraries for communication between nodes. Consequently, after COMPSs is installed on a set of machines, the SSH keys must be configured on those machines so that COMPSs can establish passwordless connections between them. This requires to install the OpenSSH package (if not present already) and follow these steps **in each machine**:

1. Generate an SSH key pair

```
$ ssh-keygen -t dsa
```

2. Distribute the public key to all the other machines and configure it as authorized

```
        For every other available machine (MACHINE):
$ scp ~/.ssh/id_dsa.pub MACHINE:./myDSA.pub
$ ssh MACHINE "cat ./myDSA.pub >> ~/.ssh/authorized_keys; rm ./myDSA.pub"
```

3. Check that passwordless SSH connections are working fine

```
        For every other available machine (MACHINE):
$ ssh MACHINE
```

For example, considering the cluster shown in Figure 5.1, users will have to execute the following commands to grant free ssh access between any pair of machines:

```
me@localhost:~$ ssh-keygen -t id_dsa
# Granting access localhost -> m1.bsc.es
me@localhost:~$ scp ~/.ssh/id_dsa.pub user_m1@m1.bsc.es:./me_localhost.pub
me@localhost:~$ ssh user_m1@m1.bsc.es "cat ./me_localhost.pub >> ~/.ssh/authorized_keys;
    rm ./me_localhost.pub"
# Granting access localhost -> m2.bsc.es
me@localhost:~$ scp ~/.ssh/id_dsa.pub user_m2@m2.bsc.es:./me_localhost.pub
me@localhost:~$ ssh user_m2@m2.bsc.es "cat ./me_localhost.pub >> ~/.ssh/authorized_keys;
    rm ./me_localhost.pub"

me@localhost:~$ ssh user_m1@m1.bsc.es
user_m1@m1.bsc.es:~> ssh-keygen -t id_dsa
user_m1@m1.bsc.es:~> exit
# Granting access m1.bsc.es -> localhost
me@localhost:~$ scp user_m1@m1.bsc.es:~/.ssh/id_dsa.pub ~/userm1_m1.pub
me@localhost:~$ cat ~/userm1_m1.pub >> ~/.ssh/authorized_keys
# Granting access m1.bsc.es -> m2.bsc.es
me@localhost:~$ scp ~/userm1_m1.pub user_m2@m2.bsc.es:~/userm1_m1.pub
me@localhost:~$ ssh user_m2@m2.bsc.es "cat ./userm1_m1.pub >> ~/.ssh/authorized_keys; rm
    ./userm1_m1.pub"
```

```
me@localhost:~$ rm ~/userm1_m1.pub

me@localhost:~$ ssh user_m2@m2.bsc.es
user_m2@m2.bsc.es:~> ssh-keygen -t id_dsa
user_m2@m2.bsc.es:~> exit
# Granting access m2.bsc.es -> localhost
me@localhost:~$ scp user_m2@m1.bsc.es:~/.ssh/id_dsa.pub ~/userm2_m2.pub
me@localhost:~$ cat ~/userm2_m2.pub >> ~/.ssh/authorized_keys
# Granting access m2.bsc.es -> m1.bsc.es
me@localhost:~$ scp ~/userm2_m2.pub user_m1@m1.bsc.es:~/userm2_m2.pub
me@localhost:~$ ssh user_m1@m1.bsc.es "cat ./userm2_m2.pub >> ~/.ssh/authorized_keys; rm
    ./userm2_m2.pub"
me@localhost:~$ rm ~/userm2_m2.pub
```

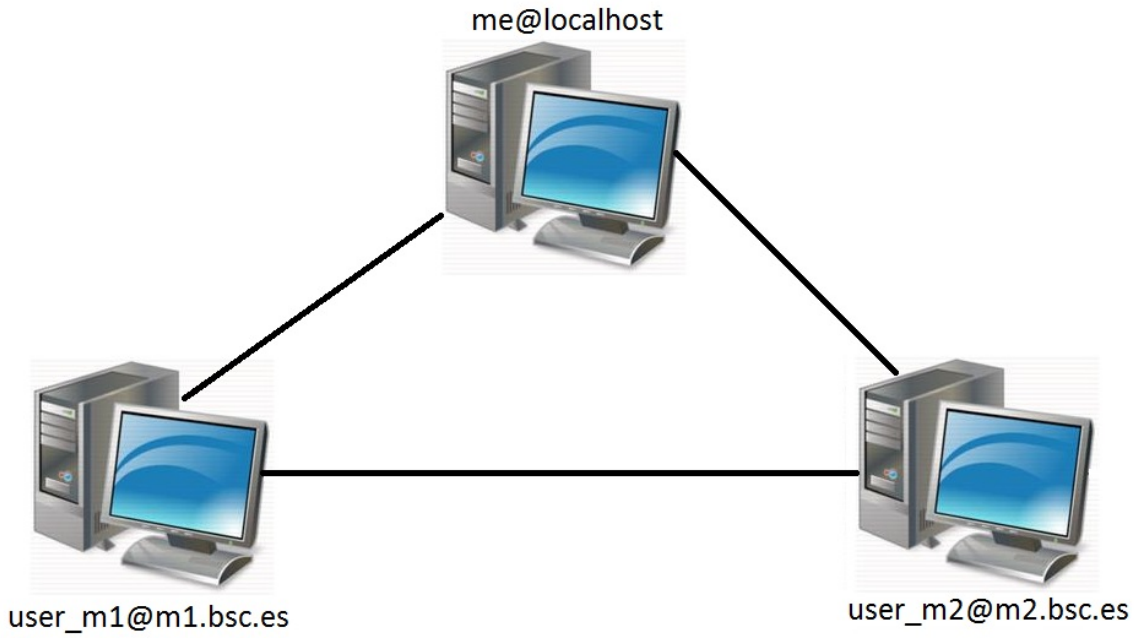

Figure 1: Cluster example

5.2 Configure the COMPSs Cloud Connectors

This section provides information about the additional configuration needed for some Cloud Connectors.

5.2.1 OCCI (Open Cloud Computing Interface) connector

In order to execute a COMPSs application using cloud resources, the rOCCI (Ruby OCCI) connector has to be configured properly. The connector uses the rOCCI CLI client (upper versions from 4.2.5) which has to be installed in the node where the COMPSs main application runs. The client can be installed following the instructions detailed at `http://appdb.egi.eu/store/software/rocci.cli`

Please find more details on the COMPSs framework at

http://compss.bsc.es