

# Combo Manual

Kickstarter to Master

Things You May Want to Know

Copyright © 2014 Computational Biophysics and Materials Science Group

PUBLISHED

OURPHYSICS.ORG

This manual basically list the problems you may encounter when you maintain the combo, or reinstall the combo.

*V1.1.0, March 2014*



# Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
<b>1.1</b>	<b>Terminology</b>	<b>5</b>
<b>1.2</b>	<b>Name</b>	<b>5</b>
1.2.1	Hardware	5
1.2.2	Non-Hardware	5
<b>1.3</b>	<b>Notation</b>	<b>6</b>
1.3.1	Code	6
<b>1.4</b>	<b>Hardware</b>	<b>6</b>
1.4.1	Processor	6
1.4.2	Communication	6
<b>1.5</b>	<b>Structure</b>	<b>6</b>
<b>1.6</b>	<b>Software</b>	<b>6</b>
1.6.1	Operation System	7
1.6.2	Compiler	7
1.6.3	Development Tools	7
<b>2</b>	<b>Installation</b>	<b>9</b>
<b>2.1</b>	<b>Installing Operating System</b>	<b>9</b>
2.1.1	Download .ISO file and Boot from DVD	9
2.1.2	Selecting Rolls	9
2.1.3	Network Settings	10
2.1.4	Disk Partition	10
<b>2.2</b>	<b>Before Connecting Computer Nodes*</b>	<b>10</b>
2.2.1	TORQUE	10
<b>2.3</b>	<b>Connecting Computer Nodes</b>	<b>11</b>

<b>2.4</b>	<b>Installing Applications</b>	<b>12</b>
2.4.1	NVIDIA CUDA Toolkit . . . . .	12
2.4.2	Intel Parallel Studio . . . . .	12
2.4.3	OpenMPI . . . . .	13
2.4.4	NAMD . . . . .	14
2.4.5	FFTW and GROMACS . . . . .	15
<b>2.5</b>	<b>Post Configuration</b>	<b>15</b>
2.5.1	Locale Setting . . . . .	15

## Terminology

### Name

- Hardware
- Non-Hardware

### Notation

- Code

### Hardware

- Processor
- Communication

### Structure

### Software

- Operation System
- Compiler
- Development Tools

# 1 — Introduction

## 1.1 Terminology

## 1.2 Name

### 1.2.1 Hardware

**Frontend** a computer, also called "server node" to connect and control the computer nodes.

**Computer Node** a computer, connected and controlled by frontend, and can do process jobs.

### 1.2.2 Non-Hardware

**Combo** Combo is a nick name of our cluster.

**Computer cluster** A computer cluster consists of a set of loosely connected or tightly connected computers that work together so that in many respects they can be viewed as a single system.<sup>1</sup>

**MPI** Message Passing Interface (MPI) is a standardized and portable message-passing system designed by a group of researchers from academia and industry to function on a wide variety of parallel computers. The standard defines the syntax and semantics of a core of library routines useful to a wide range of users writing portable message-passing programs in Fortran or the C programming language. There are several well-tested and efficient implementations of MPI, including some that are free or in the public domain. These fostered the development of a parallel software industry, and there encouraged development of portable and scalable large-scale parallel applications.<sup>2</sup>

**Parallel computing** Parallel computing is a form of computation in which many calculations are carried out simultaneously, operating on the principle that large problems can often be divided into smaller ones, which are then solved concurrently ("in parallel"). There are several different forms of parallel computing: bit-level, instruction level, data, and task parallelism. Parallelism has been employed for many years, mainly in high-performance computing, but interest in it has grown lately due to the physical constraints preventing frequency scaling. As power consumption (and consequently heat generation) by comput-

<sup>1</sup>From Wikipedia - Item:Computer cluster, [http://en.wikipedia.org/wiki/Cluster\\_\(computing\)](http://en.wikipedia.org/wiki/Cluster_(computing)), retrived on Mar 21, 2014

<sup>2</sup>From Wikipedia - Item:Message Passing Interface, [http://en.wikipedia.org/wiki/Message\\_Passing\\_Interface](http://en.wikipedia.org/wiki/Message_Passing_Interface), retrived on Mar 21, 2014

ers has become a concern in recent years, parallel computing has become the dominant paradigm in computer architecture, mainly in the form of multi-core processors.<sup>3</sup>

**Rocks** Rocks is an open-source Linux cluster distribution that enables end users to easily build computational clusters, grid endpoints and visualization tiled-display walls.<sup>4</sup> Rocks was initially based on the Red Hat Linux distribution, however modern versions of Rocks were based on CentOS, with a modified Anaconda installer that simplifies mass installation onto many computers.<sup>5</sup>

## 1.3 Notation

### 1.3.1 Code

In this manual, there are a lot of code example, to make it clear to distinguish code from others, some explanation are needed:

#### Command Line

1. Each command will start with "\$", the font is typewriter font, with various font sizes.
2. If a piece of command is too long to place in one line, a "\" is used to indicate ignoring a linebreak.
3. try not to directly copy from this manual and paste in terminal, for sometimes "~" and "\_" may looks good in manual, but inappropriate in terminal, pay attention when you paste a block of codes.

## 1.4 Hardware

Introduction about hardware of combo.

### 1.4.1 Processor

- 32 CPUs for frontend
- $16 \times 16$  CPUs for nodes
- $16 \times 1$  GPUs for nodes

### 1.4.2 Communication

- $1 \times HP2910 - 48GSwitch$
- $1 \times Vo..Switch$

## 1.5 Structure

To quickly understand how the combo works, we can image there are a team of army consisting of 16 solders as well as a captain.

## 1.6 Software

Here list the softwares installed on Combo.

<sup>3</sup>From WikiPedia - Item:Parallel computing, [http://en.wikipedia.org/wiki/Parallel\\_computing](http://en.wikipedia.org/wiki/Parallel_computing), retrived on Mar 21, 2014

<sup>4</sup>Rocks Official Website, [http://www.rocksclusters.org/wordpress/?page\\_id=57](http://www.rocksclusters.org/wordpress/?page_id=57), retrived on Mar 21, 2014

<sup>5</sup>From WikiPedia - Item:Rocks Cluster Distribution, [http://en.wikipedia.org/wiki/Rocks\\_Cluster\\_Distribution](http://en.wikipedia.org/wiki/Rocks_Cluster_Distribution), retrived on Mar 21, 2014

### 1.6.1 Operation System

The operating system is Rocks<sup>6</sup>, Current version is ver.6.2.

### 1.6.2 Compiler

- GNU Compiler (gcc, g++, fortran, JAVA...)

### 1.6.3 Development Tools

- NVIDIA CUDA Toolkit<sup>7</sup>

---

<sup>6</sup><http://rocksclusters.org>

<sup>7</sup><https://developer.nvidia.com/cuda-toolkit>





### Installing Operating System

- Download .ISO file and Boot from DVD
- Selecting Rolls
- Network Settings
- Disk Partition

### Before Connecting Computer Nodes\*

- TORQUE

### Connecting Computer Nodes

### Installing Applications

- NVIDIA CUDA Toolkit
- Intel Parallel Studio
- OpenMPI
- NAMD
- FTW and GROMACS

### Post Configuration

- Locale Setting

## 2 — Installation

To install the combo, basically, the following steps need to follow:

1. Install the Rocks in the frontend
2. Connect the computer nodes to frontend
3. Install the applications

Then, the details are as followed:

### 2.1 Installing Operating System

Installing operating system on frontend is as easy as you install Windows on your PC, once you know how to config right.

And what I am exactly doing here, is to let you know, how to do it right:

1. Download the .ISO and burn into the DVD
2. Put DVD into optical driver and follow the prompt to do one step after another.
3. done

#### 2.1.1 Download .ISO file and Boot from DVD

The file can be download from [http://www.rocksclusters.org/wordpress/?page\\_id=80](http://www.rocksclusters.org/wordpress/?page_id=80), the architecture is x86\_64 and the jumpo DVD is preferred.

**R** The installation guide is available on [http://www.rocksclusters.org/wordpress/?page\\_id=4](http://www.rocksclusters.org/wordpress/?page_id=4), to avoid iterance, only those with special attention are listed.

#### 2.1.2 Selecting Rolls

At beginning, you are required to select rolls to install. Of course, tick them all is a safe choice, anyway, I will list the rolls recommended to be installed on frontend:

**R** **Roll** In Rocks, a roll can be regarded as a package, The "Rolls" extend the system by integrating seamlessly and automatically into the management and packaging mechanisms used by base software, greatly simplifying installation and configuration of large numbers of computers.<sup>1</sup>

---


<sup>1</sup>Breaking News - Operating Systems & Middleware: SDSC Enhances Rocks Cluster Management Toolkit,

1. boot,
2. base,
3. ganglia,
4. HPC,
5. python,
6. os,
7. kernel,


### 2.1.3 Network Settings

In this section, pay **attention** to network configuration. If you have never plugged the wires, the default interface is:

- eth2: Private Network: 192.168.1.18; Netmask: 255.255.0.0
- eth3: Public Network: 144.214.25.68; Netmask: 255.255.0.0

 Do not choose *eth0* or *eth1*, or you will suffer a lot afterwards.

- Gateway: 144.214.2.254; DNS Server: 144.214.5.20

 These settings are according to CityU CSC.

Anyway, make sure every textbox is filled appropriately except longitude and latitude.

### 2.1.4 Disk Partition

To better manage our cluster, manual disk partition is required. If it is a reinstallation, delete all the existing partition. Then simply press "add". `/dev/sdaN` (where N is a number starts from 1) will be created automatically. Maximum number for N is 5, after 5 it will create new devices automatically. However ext4 might have the problem of a maximum volume of 16T, therefore `/data` and `/data2` should be specially created using logical volumes.

#### Partition Setting

File System	Mounted on	Size (MB)	Format ?
	/	20000	yes
	/boot	500	yes
	/var	5000	yes
	swap	10000	yes
	/export	max	yes
<code>/dev/mapper/vg_combo_LogVol00</code>	<code>/data</code>	11T	no
<code>/dev/mapper/vg_combo_LogVol01</code>	<code>/data2</code>	9T	no

## 2.2 Before Connecting Computer Nodes\*

This part is very special, because the **torque** is preferred to be installed before connecting computer nodes. The reason is what we use is **Torque Roll**, it will automatically identify nodes only when nodes are added after torque installed.


### 2.2.1 TORQUE

Torque is not packaged in rocks DVD, we need to download torque<sup>2</sup> from `ftp://ftp.uit.no/pub/linux/rocks/torque-roll/`.

<http://web.archive.org/web/20070927031015/http://www.gridtoday.com/04/0216/102698.html>, retrived on Mar 19, 2014

<sup>2</sup>Torque+Maui job queueing system, packaged by HPC Group at University of Tromso, Norway

1. under `/export/rocks/install` (ensure you are there), run  
`$ rocks add roll /root/Desktop/torque-6.0.0-1.x86_64.disk1.iso` (differ if the ISO file differs),
2. `$ rocks enable roll torque` ,
3. `$ rocks create distro` (make sure you are really in `/export/rocks/install`, for this step is to create a Rocks distribution, which is used to install Rocks nodes with torque.)
4. `$ rocks run roll torque | sh` (run the torque roll),
5. `$ reboot` (after reboot, you can use `$ pbsnodes -a` to see whether it is running or not)

 I have to complain about torque here. The version Rocks Official Website<sup>3</sup> recommend torque package from *HPC Group at University of Tromso, Norway*, which is version 6.0.0. I cannot find any documentation or tutorial of this version. Oh that is terribly annoying. If you did not pay attention to the small-size introduction about Rocks, and Google "torque", you will find another torque...

We are about to connect computer nodes, if you do not want to create partitions one by one, here's a trick! You might create a XML file for nodes to read when creating their partitions:

- `$ cd /export/rocks/install/site-profiles/6.2/nodes/`
- `$ cp skeleton.xml replace-partition.xml`
- `$ vi replace-partition.xml`

Then insert the following lines in-between `<pre>` and `</pre>`:

```
<pre>
echo "clearpart --all --initlabel --drivers=sda
part / --size 100000 --ondisk sda
part swap --size 80000 --ondisk sda
part /mydata --size 1 --grow --ondisk sda" > /tmp/user_partition_info
</pre>
```

Remember to execute


- `$ rocks create distro`

in order to make it work.

## 2.3 Connecting Computer Nodes

Just mentioned before, frontend is "*Caption*" and we need to establish connections among "*Caption*" and "*Solders*".

1. Ensure you have installed Torque roll and rebooted,
2. Cut off Internet, Shutdown all the nodes by your hands,

 The reason for cutting off internet is that if someone knock on Combo during insert-ether, Combo will recognise it as a node which is obviously not preferable.

3. Open Terminal, run `$ insert-ethers`, choose **Compute** as type,
4. boot each node one by one, (take your time, maybe switch on one every 1 min), choose "Network Booting" when prompted,
5. partition nodes according to:

File System	Mounted on	Size	Format ?
/dev/sda2	/	100000	yes
/dev/sda1	swap	80000	yes
/dev/sdb	/mydata	whatever	yes

<sup>3</sup><http://www.rocksclusters.org/>

## 2.4 Installing Applications

The most challenging mission just starts. In this section, we need to install several applications. We will deal with them one by one.

### 2.4.1 NVIDIA CUDA Toolkit

Install CUDA is relatively easy, first install NVIDIA driver, then CUDA Toolkit. CUDA Toolkit could be download from <https://developer.nvidia.com/cuda-downloads>, the installation guide on official website([urlhttp://docs.nvidia.com/cuda/cuda-getting-started-guide-for-linux/index.html](http://docs.nvidia.com/cuda/cuda-getting-started-guide-for-linux/index.html)), here only list some interesting points: (Applied to NVIDIA CUDA Getting Started Guide for Linux, DU-05347-001\_v5.5, July 19, 2013)

1. \$ sh cuda-linux64-rel-5.5.22-16488124.run -extract=/share/apps/cuda-install
2. \$ sudo sh cuda-linux64-rel-5.5.22-16488124.run -prefix=/share/apps/cuda
3. add folowing code to ~/.bashrc:

```
# Setting for CUDA
export PATH=/share/apps/cuda/bin:$PATH
export LD_LIBRARY_PATH=/share/apps/cuda/lib64:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH=/share/apps/cuda/lib:$LD_LIBRARY_PATH
```



Here is a trick to quickly sync the .bashrc to all node, use the code in frontend:

```
for i in compute-0-0 compute-0-2 compute-0-3 compute-0-4 compute-0-5 compute
-0-6 compute-0-7 compute-0-8 compute-0-9 compute-0-10 compute-0-11
compute-0-12 compute-0-13 compute-0-14 compute-0-15 compute-0-16 ; do
  scp ~/.bashrc ${i}:/ ; done

for i in compute-0-0 compute-0-2 compute-0-3 compute-0-4 compute-0-5 compute
-0-6 compute-0-7 compute-0-8 compute-0-9 compute-0-10 compute-0-11
compute-0-12 compute-0-13 compute-0-14 compute-0-15 compute-0-16 ; do
  ssh ${i} source ~/.bashrc ; done
```

4. \$ source ~/.bashrc to update environment variables, and do this to each node.
5. Next, you need to install NVIDIA driver to each node. on each node, run  
\$ /share/apps/cuda/NVIDIA-Linux-x86\_64-319.37.run  
on each node, follow prompt.
- To test whether CUDA is set up, do the following test steps(I found the test method on the official guide is somehow incorrect, here is the appropriate method):
  1. run \$ cuda-install-samples-5.5.sh ~ in CUDA installation directory,
  2. \$ cd ~/NVIDIA\_CUDA-5.5\_Samples/1\_Uutilities/deviceQuery/,
  3. \$ make,
  4. \$ ./deviceQuery,
 You will see "Result = PASS" at the end of terminal window if CUDA is set up right.

### 2.4.2 Intel Parallel Studio

Intel Parallel Studio is not a free software, to use it legally and without cost, we need to register for non-commercial version(<http://software.intel.com/en-us/non-commercial-software-development>).

- download the **parallel\_studio\_xe\_2013\_sp1\_update1.tgz**, place it somewhere, untar it using \$ tar -xzf parallel\_studio\_xe\_2013\_sp1\_update1.tgz.
- You can either choose to run install\_GUI.sh or install.sh program under uncompressed directory, follow prompt.
- When there is **Custom** or **Advanced** option, choose them.
- When prompted to define the installation location, use /share/apps/ as target directory.
- x86 option can be ticked can also be unticked. Ignore the warning at the end if you did not tick to install x86 compilers.

 Remember to put the following codes into `~/bashrc`:

```
#for Intel mkl
source /share/apps/intel/mkl/bin/mklvars.sh intel64
```

### 2.4.3 OpenMPI

I followed this blog<sup>4</sup> to install OpenMPI. Maybe someday this blog disappears, then you cannot find it, in case, I will lead you through it.

1. I found till now, we do not have `g++` compiler installed, use `$ yum install gcc-c++` to install `g++` compiler first.
2. copy the following code to `~/bashrc` if they are not there:

```
#for Intel
source /share/apps/intel/bin/compilervars.sh intel64
#GNU compiler
export PATH=/share/apps/intel/bin:$PATH
export INCLUDE=/share/apps/intel/include:$PATH
export LD_LIBRARY_PATH=/share/apps/intel:$PATH
export CC=gcc
export CFLAGS=-O3
export CXX=g++
export CXXFLAGS=-O3
export FC=gfortran
export F77=gfortran
export F90=gfortran
export FFLAGS=-O3
```

3. run `$ source ~/.bashrc` to make bashrc file work.
4. download OpenMPI from <http://www.open-mpi.org/>, and place into `/share/apps/`
5. then create 2 folders in `/share/apps/`:

```
cd /share/apps
mkdir openmpi-install
mkdir openmpi-1.6.5
```

6. run `$ tar -C openmpi-install -xf openmpi-1.6.5.tar.gz` to uncompress it into **openmpi-install** folder, then `cd` into the folder.
7. run `$ ./configure --prefix=/share/apps/openmpi-1.6.5` to do install configure,
8. then run `$ make all install` to install it.
9. add following code to `~/bashrc`:

```
# Settings for OpenMPI
export PATH=/share/apps/openmpi-1.6.5/bin:$PATH
export INCLUDE=/share/apps/openmpi-1.6.5/include:$INCLUDE
export LD_LIBRARY_PATH=/share/apps/openmpi-1.6.5/lib:$LD_LIBRARY_PATH
```

also, run `$ source ~/.bashrc` to make it work. and till now, the `~/bashrc` file should look like this:

```
#for Intel
source /share/apps/intel/bin/compilervars.sh intel64
#GNU compiler
export PATH=/share/apps/intel/bin:$PATH
export INCLUDE=/share/apps/intel/include:$PATH
export LD_LIBRARY_PATH=/share/apps/intel:$PATH
export CC=gcc
export CFLAGS=-O3
export CXX=g++
export CXXFLAGS=-O3
export FC=gfortran
export F77=gfortran
```

<sup>4</sup>Shane Tarleton, Install OpenMPI with GNU Compilers, <http://www.shanetarleton.com/install-openmpi-with-gnu-compilers>


```
export F90=gfortran
export FFLAGS=-O3
# Settings for OpenMPI
export PATH=/share/apps/openmpi-1.6.5/bin:$PATH
export INCLUDE=/share/apps/openmpi-1.6.5/include:$INCLUDE
export LD_LIBRARY_PATH=/share/apps/openmpi-1.6.5/lib:$LD_LIBRARY_PATH
```

#### 2.4.4 NAMD


(Added on 12 July 2015 when we are doing the reinstallation of Combo after an unknown attack causing the whole thing to crash)

Generally I followed the release note of NAMD (Here<sup>5</sup> for NAMD 2.10) and I do recommend all of you to do so. Do not bother to understand every word of it for the first glance, but do bear in mind that they are one of the criteria for your understanding of UIUC softwares. In the following section, I would try to combine what they tell us about a MPI (probably with InfiniBand) plus CUDA version of NAMD 2.10 so you do not have to scroll up and down the page.


- Firstly of cause, download the correct tar ball (2015/07/15: NAMD\_2.10\_Source.tar) and send it to Combo using your own method (e.g. FileZilla or scp).
- tar xvf NAMD\_2.10\_Source.tar.gz

 I am not sure why the release note suggests xzf options and also the file is end with gz extension while I can only untar the file using xf options showing that the file is NOT zipped. Anyway, it does not matter much.


- cd NAMD\_2.10\_Source
- tar xvf charm-6.6.1.tar
- cd charm-6.6.1
- env MPICXX=mpicxx ./build charm++ mpi-linux-x86\_64 --with-production

 In fact you can simply type ./build to enter an interactive mode choosing compilation options one by one and I DO recommend you to try that.

- cd mpi-linux-x86\_64/tests/charm++/megatest
- make pgm
- mpirun -n 4 ./pgm (run as any other MPI program on your cluster)
- cd ../../../../ (go to the very original NAMD\_2.10\_Source directory)

 I do recommend you to copy the following lines into a bash script instead of copying them to the console one by one.

- wget http://www.ks.uiuc.edu/Research/namd/libraries/fftw-linux-x86\_64.tar.gz
- tar xzf fftw-linux-x86\_64.tar.gz mv linux-x86\_64 fftw
- wget http://www.ks.uiuc.edu/Research/namd/libraries/tcl8.5.9-linux-x86\_64.tar.gz
- wget http://www.ks.uiuc.edu/Research/namd/libraries/tcl8.5.9-linux-x86\_64-threaded.tar.gz
- tar xzf tcl8.5.9-linux-x86\_64.tar.gz
- tar xzf tcl8.5.9-linux-x86\_64-threaded.tar.gz
- mv tcl8.5.9-linux-x86\_64 tcl
- mv tcl8.5.9-linux-x86\_64-threaded tcl-threaded
- ./config Linux-x86\_64-g++ --charm-arch mpi-linux-x86\_64 --with-cuda --cuda-prefix /share/apps/cuda

 What follows the --charm-arch is the name of your charm build directory.

- cd Linux-x86\_64-g++


<sup>5</sup><http://www.ks.uiuc.edu/Research/namd/2.10/notes.html>



- `gmake -j32`
- `./namd2 src/alanin`
- `mpirun -n 4 ./namd2 src/alanin`

### 2.4.5 FFTW and GROMACS

Installing FFTW and GROMACS should be difficult if you have not ever experienced. Don't worry, our Kevin has worked out the right way to install it and you, just need to do copy and paste work. :-)


 I still want to make sure again you have got **g++** installed. if not, run `$ yum install gcc-c++`(if installed, you can still run it to make sure).

#### FFTW

1. Download FFTW from <http://www.fftw.org/download.html>, untar it and enter the directory.
2. run `$ ./configure MPICC=mpicc --enable-shared --enable-mpi --prefix=/share/apps/fftw3` to configure installation.
3. `$ make,`
4. `$ make install.`

#### GROMACS

1. You need **CMAKE** to finish this installation, download from <http://www.cmake.org/cmake/resources/software.html> and install it according to its online guide.
2. Download latest GROMACS from <http://www.gromacs.org/Downloads>, untar it and enter the directory.
3. run `$ mkdir build` to create a folder to store build files,
4. `$ cd build,`
5. `$ CMAKE_PREFIX_PATH=/share/apps/fftw3:/share/apps/openmpi-1.6.5/ cmake .. \`  
`-DGMX_DOUBLE=ON -DCMAKE_INSTALL_PREFIX=/share/apps/gromacs -DGMX_MPI=ON`

 The "\" at the end of the first line means ignore line break afterwards, this is one-line command; also, the `/share/apps/openmpi-1.6.5/` and `/share/apps/fftw3/` may not be accurate for the versions you install may differ from those I did.

6. `$ make,`
7. `$ make install.`

Don't forget to add the following code to `~/ .bashrc` file:

```
|| #for GROMACS
|| source /share/apps/gromacs/bin/GMXRC.bash
```

also, run `$ source ~/ .bashrc` and sync to each node to make it work.

## 2.5 Post Configuration

### 2.5.1 Locale Setting

Basically, every time you login into SSH, it will give you a warning:  
warning: setlocale: LC\_CTYPE: cannot change locale (UTF-8)  
fixxing it could be very easy, open `/etc/environment` and add this line to it:

```
|| LC_ALL=C
```