

DS-Bench/Test-Env(D-Cloud) Setup Manual

Version E1.00

2013/09/01

Edited by DEOS R&D Center



DEOS Project

JST-CREST

Research Area

“Dependable Operating Systems for Embedded Systems Aiming at Practical Applications”



Japan Science and Technology Agency

Contents

1 Instructions4

2 Environmental configuration5

2.1 Configuration which does not use a virtual environment5

2.2 Configuration which uses a virtual environment6

3 D-Cloud8

3.1 Compatibility with Ubuntu12.048

3.2 The installation procedure of OpenStack.....8

3.3 The installation procedure of a D-Cloud controller8

3.3.1 When OpenStack is installed8

3.3.1.1 When installing from a debian package8

3.3.2 When OpenStack is not installed9

3.3.2.1 When installing from a debian package9

4 OpenStack.....10

4.1 The construction procedure of an OpenStack controller + computing node.....10

4.1.1 Network setting.....10

4.1.2 Adding a host name11

4.1.3 Adding a Repository11

4.1.4 Installation of OpenStack.....12

4.1.5 Project creation for D-Cloud.....17

4.2 The construction procedure of an OpenStack computing node19

4.2.1 Adding a host name19

4.2.2 Adding a Repository19

4.2.3 Installation of OpenStack (computing node)19

5 DS-Bench.....21

5.1 Compatibility with Ubuntu12.0421

5.2 Controller installation procedure21

5.2.1 Setup of CPAN shell.....21

5.2.2 Installation of DS-Bench controller21

5.2.2.1 Installing from a debian package21

5.2.3 Setup of Apache.....22

5.2.4 Settings of a Sedna DB server (optional changes).....23

5.2.5 Settings of a D-Cloud controller(optional changes).....23

5.3 Installation of a SednaDB server24

5.3.1	Installation from debian package	24
5.3.2	Modification of Sedna.....	24
5.4	Installation of a DS-Bench target.....	25
5.4.1	Installation from debian package	25
5.4.2	Modification of the repository file	25
6	Virtual machine.....	26
6.1	Setup of the network for virtual target use.....	26
6.2	Procedure of virtual target image creation.....	27
6.2.1	Registration of initial image.....	27
6.2.2	Starting the initial image	28
6.2.3	Create the virtual target image	29
6.2.4	Registering a virtual target machine	33

1 Instructions

In this Setup Manual (hereafter referred to as “this document”), installation and setup steps of DS-Bench and Test-Env (D-Cloud), and the creation procedure of a virtual machine are explained.

DS-Bench is a dependability measurement device and Test-Env is a tool which manages resources and does a system test quickly.

Please install OpenStack in a server which the Test-Env (D-Cloud) controller has been started, and perform the various setup procedures. When using only a physical machine, it is not necessary to carry out this procedure or other virtual machine creation procedures. Please refer to the separate document DS-Bench/Test-Env(D-Cloud) Execution Manual for directions for use.

Note: In subsequent sections "Test-Env (D-Cloud)" is simply referred to as "D-Cloud".

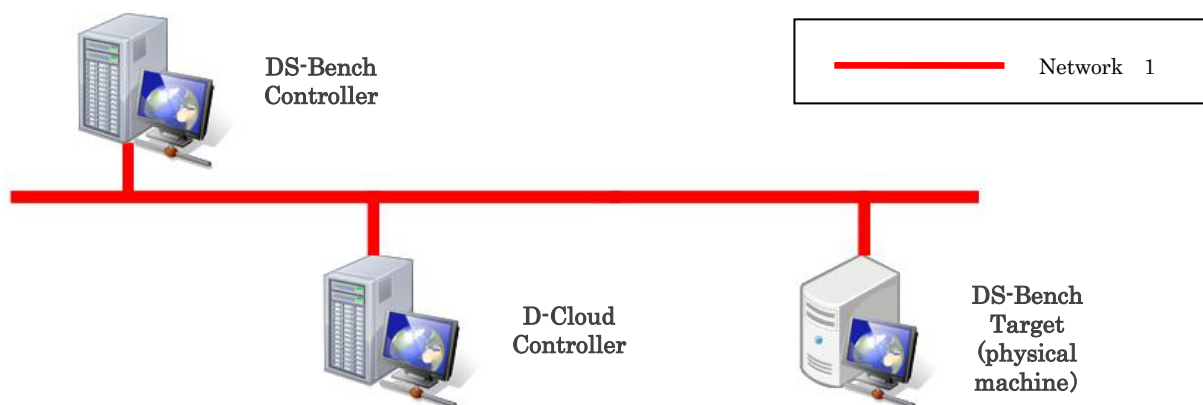
Each system name, product name, and service name in this document is the trade-mark or registered trade-mark of the company that created the respective system, product, or service.

2 Environmental configuration

This document explains how the following configurations are installed and set up.

2.1 Configuration which does not use a virtual environment

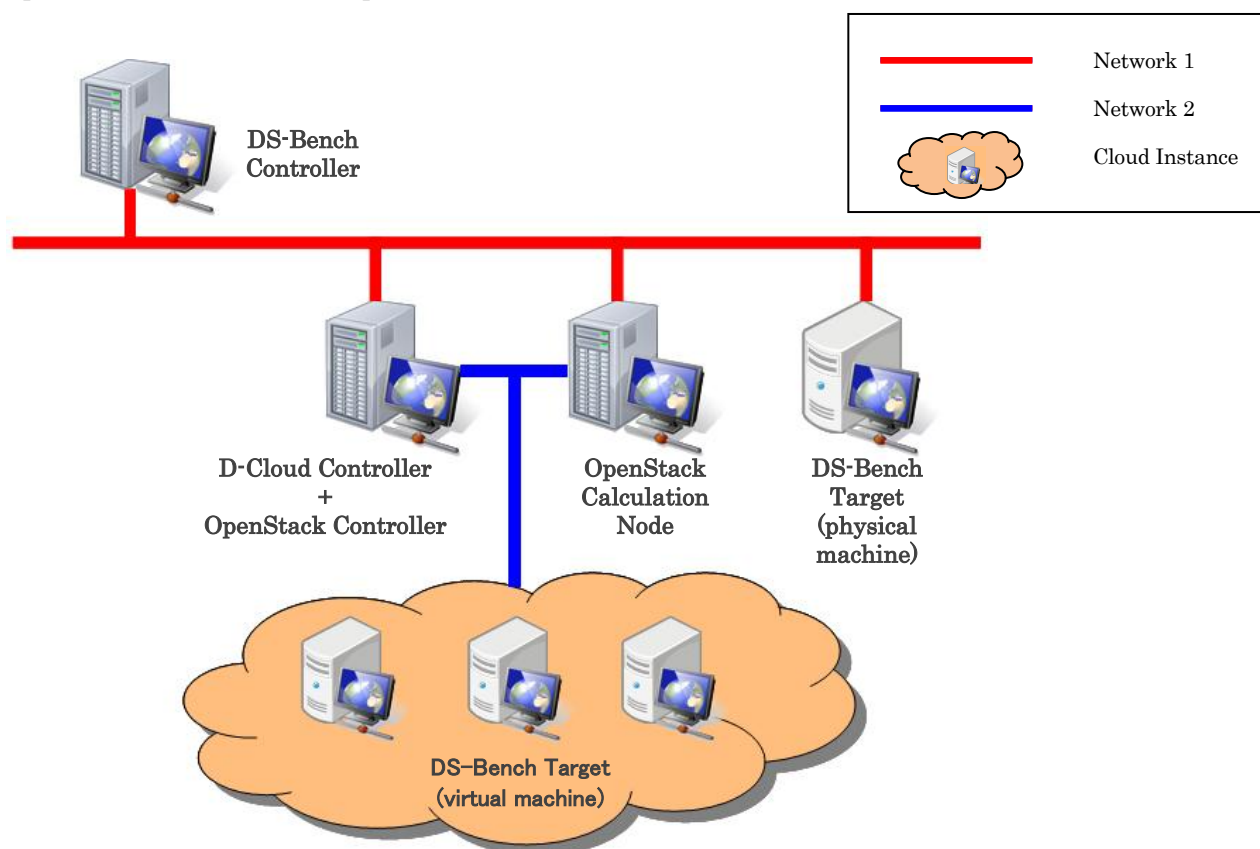
- DS-Bench Controller
Ubuntu Server Edition 64-bit 10.04.3 or Ubuntu Server Edition 64-bit 12.04.1
Sedna XML DB
- DS-Bench Target (physical machine)
Ubuntu Server Edition 64-bit 10.04.3 or Ubuntu Server Edition 64-bit 12.04.1
- D-Cloud Controller
Ubuntu Server Edition 64-bit 10.04.3 or Ubuntu Server Edition 64-bit 12.04.1



Node	Outline	Details
D-Cloud Controller	In order to manage information consistently, acquisition of information regarding a computational resource, and reservation, start-up, etc. of resources are performed via D-Cloud. Moreover, management and execution of Fault Injection are performed. Since a virtual environment is not used, withoutOpenStack.deb of D-Cloud is installed.	D-Cloud Controller 3.1 3.3
DS-Bench Controller	It can handle a benchmark tool and scenario data, can send a scenario execution command to a target, and can acquire and save the result. Moreover, these functions can be easily operated by WebUI. Sedna is installed as the DS-Bench controller and database.	DS-Bench Controller 5.1 5.2 Sedna DB 5.3
DS-Bench Target (physical machine)	Receives a demand from a DS-Bench controller, executes a test, generates an anomaly, and formats and returns the results. The package for DS-Bench targets is installed.	DS-Bench Target 5.4

2.2 Configuration which uses a virtual environment

- DS-Bench Controller
Ubuntu Server Edition 64-bit 10.04.3 or Ubuntu Server Edition 64-bit 12.04.1
Sedna XML DB
- DS-Bench Target(physical machine)
Ubuntu Server Edition 64-bit 10.04.3 or Ubuntu Server Edition 64-bit 12.04.1
- DS-Bench Target(virtual machine)
ubuntu-10.04-server-cloudimg-amd64 or ubuntu-12.04-server-cloudimg-amd64
- D-Cloud Controller + OpenStack Controller
Ubuntu Server Edition 64-bit 10.04.3 or Ubuntu Server Edition 64-bit 12.04.1
OpenStack 2011.3(Diablo) or OpenStack 2012.1.3(Essex)
- Calculation Node
Ubuntu Server Edition 64-bit 10.04.3 or Ubuntu Server Edition 64-bit 12.04.1
OpenStack 2011.3(Diablo) or OpenStack 2012.1.3(Essex)



Node	Outline	Details
D-Cloud Controller + OpenStack Controller	In order to manage information in a consistent manner, acquisition of information on a computational resource, and reservation, starting, etc. of resources are performed via D-Cloud. Moreover, management and execution of Fault Injection are performed. In order to build a virtual environment, installation and a setup of withOpenStack.deb of D-Cloud and OpenStack are performed.	D-Cloud Controller 3.1 3.3 OpenStack 4.1
DS-Bench Controller	It can handle a benchmark tool and scenario data, send a scenario execution instruction to a target, and acquire and save the result. Moreover, these functions can be easily operated by WebUI. Sedna is installed as the DS-Bench controller and a database.	DS-Bench Controller 5.1 5.2 Sedna DB 5.3
DS-Bench Target (physical machine)	Receives a demand from a DS-Bench controller, executes a test, generates an anomaly, and formats and returns the results. The package for DS-Bench targets is installed.	DS-Bench Target 5.4
OpenStack Calculation Node	The OpenStack which D-Cloud uses usually operates only on a D-Cloud controller. When there is a computational resource shortage in D-Cloud etc., please install and set up computing nodes as necessary. If a computing node is added, a virtual machine can be newly created using the computational resource of the computing node.	OpenStack 4.2
DS-Bench Target (virtual machine)	A DS-Bench virtual target is set up by the D-Cloud controller in which OpenStack is installed, or by an OpenStack computing node. Receives a demand from a DS-Bench controller, executes a test, generates an anomaly, and formats and returns the results.	DS-Bench Target 5.4 Virtual machine 6

3 D-Cloud

3.1 Compatibility with Ubuntu12.04

When installing the package of DS-Bench/D-Cloud in Ubuntu12.04, it is necessary to install a **libdigest-sha1-perl** package. Please install according to the following procedure.

```
$ wget http://launchpadlibrarian.net/85191944/libdigest-sha1-perl_2.13-  
2bui.lud2.amd64.deb  
$ sudo dpkg -i libdigest-sha1-perl_2.13-2bui.lud2.amd64.deb
```

The operations below are the same whether using Ubuntu12.04 or Ubuntu10.04.

3.2 The installation procedure of OpenStack

Please install OpenStack when you use a virtual machine. Please refer to "[4 OpenStack](#)" for the installation procedure. When using only a physical machine, it is not necessary to carry out this procedure.

3.3 The installation procedure of a D-Cloud controller

3.3.1 When OpenStack is installed

3.3.1.1 When installing from a debian package

Please move to the folder with the deb package of a DS-Bench target and a D-Cloud controller, and execute the following commands. When installing a D-Cloud package, it is necessary to install a DS-Bench target package first, to have the proper dependency relation.

```
$ sudo dpkg -i -force-depends target.deb  
$ sudo apt-get install -f  
$ sudo dpkg -i d-cloud-withOpenStack.deb
```

During installation of a DS-Bench target package, when you are asked for certain input, please press the Enter key, without inputting anything.

While installing D-Cloud, you will be asked the following questions. Please make settings according to the environment in use. Here, the port numbers of the D-Cloud resources daemon and D-Cloud controller daemon are the default values.

```
Input Port Number for D-Cloud Controller Daemon [ 9903 ]:  
Input Port Number for D-Cloud Resource Daemon [ 9904 ]:  
Input Port Number for D-Cloud Map Server Daemon [ 7000 ]: 9905  
Input Path for novarc of OpenStack [ /var/dcloud/openstack/creds/novarc ]: /var/nova/novarc
```

Note: For the path of novarc, please specify the path created during the OpenStack installation. By default, the path is /home/"username"/novarc

3.3.2 When OpenStack is not installed

3.3.2.1 When installing from a debian package

Please move to the folder with the deb package of a DS-Bench target and a D-Cloud controller, and execute the following commands.

```
$ sudo dpkg -i --force-depends target.deb  
$ sudo apt-get install -f  
$ sudo dpkg -i d-cloud-withoutOpenStack.deb
```

While installing D-Cloud, you will be asked the following questions. Please make settings according to the environment in use.

```
Input Port Number for D-Cloud Controller Daemon [ 9903 ]:  
Input Port Number for D-Cloud Resource Daemon [ 9904 ]:  
Input Port Number for D-Cloud Map Server Daemon [ 7000 ]: 9905
```

Here, the port numbers of D-Cloud resources daemon and D-Cloud controller daemon are the default values.

This completes the installation and configuration of D-Cloud.

4 OpenStack

4.1 The construction procedure of an OpenStack controller + computing node

When you want to use a virtual machine, please install OpenStack in the server that the D-Cloud controller has started or is to install. Usually, the OpenStack which D-Cloud uses operates only on a D-Cloud controller.

4.1.1 Network setting

A bridge is set up by the following procedure. Here, the IP address of D-Cloud is set to "192.168.1.2", and a bridge name is set to "br0." When network edit finishes, be sure to reboot the network.

```
$ sudo apt-get install -y bridge-utils

$ sudo vi /etc/network/interfaces
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet manual
up ifconfig eth0 0.0.0.0 up

auto br0
iface br0 inet static
address 192.168.1.2
network 192.168.1.0
netmask 255.255.255.0
gateway 192.168.1.1
dns-nameservers 192.168.1.1
bridge_ports eth0
bridge_stp off
bridge_maxwait 1

$ sudo /etc/init.d/networking restart
```

Note: Although "**dns-nameservers**" is unnecessary with Ubuntu10.04, please be sure to enter this if using Ubuntu12.04.

4.1.2 Adding a host name

The host names of an OpenStack controller and of a computing node are registered into "/etc/hosts." Here as an example, "192.168.1.2" is added as the IP address of the controller and "controller" is added as the host name.

Note: The computing node IP address is set to "192.168.1.3", and the host name is set to be "compute".

```
$ sudo vi /etc/hosts
127.0.0.1      localhost
127.0.1.1      controller

192.168.1.2    controller
192.168.1.3    compute
```

4.1.3 Adding a Repository

In Ubuntu12.04, OpenStack is offered as a package, so you can install it as is, but if you install in Ubuntu10.04, please add a repository and install OpenStack from there. A repository is added by the following procedure.

```
$ gpg -keyserver keyserver.ubuntu.com -recv 7D21C2EC3D1B4472
$ gpg -export -armor 7D21C2EC3D1B4472 | sudo apt-key add -

$ sudo vi /etc/apt/sources.list
deb http://apt.cloudscaling.com/openstack-release/2011.3/ubuntu lucid main
```

The newest package list is acquired as follows.

```
$ sudo apt-get update
.
Reading package lists... 0%
Reading package lists... 100%
Reading package lists... Done
```

4.1.4 Installation of OpenStack

OpenStack is installed as follows.

```
$ sudo apt-get install -y rabbitmq-server
Reading package lists... 0%
.
.
Idconfig deferred processing now taking place
$
```

```
$ sudo apt-get install -y python-greenlet python-mysqldb
Reading package lists... 0%
.
.
Processing triggers for python-support ...
$
```

```
$ sudo apt-get install -y nova-network nova-compute nova-scheduler
Reading package lists... 0%
.
.
Idconfig deferred processing now taking place
$
```

```
$ sudo apt-get install -y libvirt-bin
Reading package lists... 0%
.
.
libvirt-bin is already the newest version.
libvirt-bin set to manually installed.
$
```

```
$ sudo apt-get install -y euca2ools unzip
Reading package lists... 0%
.
.
Setting up cloud-utils (0.11-0ubuntu1) ...
$
```

```
$ sudo apt-get install -y nova-volume nova-vncproxy nova-api nova-ajax-console-proxy
Reading package lists... 0%
.
.
Processing triggers for python-support ...
$
```

```
$ sudo apt-get install -y glance
Reading package lists... 0%
.
.
$
```

If you are making settings through Ubuntu12.04, please also install the following package.

```
$ sudo apt-get install -y nova-cert
Reading package lists... 0%
.
.
$
```

The setting script of mysql is edited as follows. If required, please set the root user's password after "MYSQL_PASS" and set the nova user's password after "NOVA_PASS". Here, both are set to be "nova". Please refer to the settings of Ubuntu10.04 or Ubuntu12.04 suited to the environment in use.

In the case of Ubuntu10.04

```
$ sudo vi ~/nova-mysql.sh
#!/bin/bash

MYSQL_PASS=nova
NOVA_PASS=nova

cat << MYSQL_PRESEED | sudo debconf-set-selections
mysql-server-5.1 mysql-server/root_password password $MYSQL_PASS
mysql-server-5.1 mysql-server/root_password_again password $MYSQL_PASS
mysql-server-5.1 mysql-server/start_on_boot boolean true
MYSQL_PRESEED

sudo apt-get install -y mysql-server
sudo sed -i 's/127.0.0.1/0.0.0.0/g' /etc/mysql/my.cnf
sudo service mysql restart
sudo mysql -u root -p$MYSQL_PASS -e 'CREATE DATABASE nova;'
sudo mysql -u root -p$MYSQL_PASS -e "GRANT ALL PRIVILEGES ON *.* TO 'nova'@'%' WITH GRANT OPTION;"
sudo mysql -u root -p$MYSQL_PASS -e "SET PASSWORD FOR 'nova'@'%' = PASSWORD('$NOVA_PASS');"
```

In the case of Ubuntu12.04

```
$ sudo vi ~/nova-mysql.sh
#!/bin/bash

MYSQL_PASS=nova
NOVA_PASS=nova

cat << MYSQL_PRESEED | sudo debconf-set-selections
mysql-server-5.5 mysql-server/root_password password $MYSQL_PASS
mysql-server-5.5 mysql-server/root_password_again password $MYSQL_PASS
mysql-server-5.5 mysql-server/start_on_boot boolean true
MYSQL_PRESEED

sudo apt-get install -y mysql-server
sudo sed -i 's/127.0.0.1/0.0.0.0/g' /etc/mysql/my.cnf
sudo service mysql restart
sudo mysql -u root -p$MYSQL_PASS -e 'CREATE DATABASE nova;'
sudo mysql -u root -p$MYSQL_PASS -e "GRANT ALL PRIVILEGES ON *.* TO 'nova'@'%' WITH GRANT OPTION;"
sudo mysql -u root -p$MYSQL_PASS -e "SET PASSWORD FOR 'nova'@'%' = PASSWORD('$NOVA_PASS');"
sudo mysql -u root -p$MYSQL_PASS -e "GRANT ALL PRIVILEGES ON nova.* TO nova@localhost IDENTIFIED BY '$NOVA_PASS';"
```

The edited script is executed as follows.

```
$ sudo bash ~/nova-mysql.sh
Reading package lists... 0%
.
.
mysql start/running, process 11521
$
```

OpenStack-related authority is exercised as follows. The following red character portion is also displayed in Ubuntu12.04.

```
$ sudo chown -R nova:root /etc/nova
$
$ sudo ls -ltrR /etc/nova
/etc/nova:
合計 12
-rw-r--r-- 1 nova root 19 2011-09-23 03:25 nova-compute.conf
-rw-r--r-- 1 nova root 3080 2011-09-23 02:53 api-paste.ini
-rw----- 1 nova root 254 2011-09-23 02:53 nova.conf
-rw----- 1 nova root 254 2012-08-25 02:53 policy.json
$
```

The configuration file of OpenStack is edited as follows. Please set the red character portions as you like. Please refer to the settings of Ubuntu10.04 or Ubuntu12.04 suited to the environment in use.

In the case of Ubuntu10.04:

```
$ sudo vi /etc/nova/nova.conf

--dhcpbridge_flagfile=/etc/nova/nova.conf
--dhcpbridge=/usr/bin/nova-dhcpbridge
--logdir=/var/log/nova
--state_path=/var/lib/nova
--lock_path=/var/lock/nova
--flagfile=/etc/nova/nova-compute.conf
--force_dhcp_release=True
--use_deprecated_auth
--verbose
--sql_connection=mysql://nova:nova@controller/nova
--network_manager=nova.network.manager.FlatDHCPManager
--public_interface=eth0
--flat_interface=eth0
--flat_injected=false
--bridge=eth0
--bridge_interface=br100
--network_host=192.168.1.2
--routing_source_ip=192.168.1.2
--fixed_range=10.0.0.0/8
--network_size=1024
--cc_host= 192.168.1.2
--ec2_url=http://192.168.1.2:8773/services/Cloud
--s3_host=192.168.1.2
--s3_dmz=192.168.1.2
--glance_api_servers=192.168.1.2:9292
--image_server=nova.image.glance.GlanceImageService
--rabbit_host=192.168.1.2
--vnc_enabled=true
--vncproxy_url=http://192.168.1.2:6080
--vnc_console_proxy_url=http://192.168.1.2:6080
```

Note: The setting of `nova:nova@controller` specifies `nova:[the password of the nova user of mysql]@[the computer name]`.

In the case of Ubuntu12.04:

```
$ sudo vi /etc/nova/nova.conf

--dhcpbridge_flagfile=/etc/nova/nova.conf
--dhcpbridge=/usr/bin/nova-dhcpbridge
--logdir=/var/log/nova
--state_path=/var/lib/nova
--lock_path=/var/lock/nova
--flagfile=/etc/nova/nova-compute.conf
--force_dhcp_release=True
--use_deprecated_auth
--verbose
--sql_connection=mysql://nova:nova@controller/nova
--network_manager=nova.network.manager.FlatDHCPManager
--public_interface=eth0
--flat_interface=eth0
--flat_injected=false
--bridge=eth0
--bridge_interface=br100
--network_host=192.168.1.2
--routing_source_ip=192.168.1.2
--fixed_range=10.0.0.0/8
--network_size=1024
--cc_host= 192.168.1.2
--ec2_url=http://192.168.1.2:8773/services/Cloud
--s3_host=192.168.1.2
--s3_dmz=192.168.1.2
--glance_api_servers=192.168.1.2:9292
--image_server=nova.image.glance.GlanceImageService
--rabbit_host=192.168.1.2
--vnc_enabled=true
--vncproxy_url=http://192.168.1.2:6080
--vnc_console_proxy_url=http://192.168.1.2:6080
--iscsi_helper=tgtadm
--libvirt_use_virtio_for_bridges
--connection_type=libvirt
--root_helper=sudo nova-rootwrap
```

The process is started as follows.

```
$ sudo nova-manage db sync
$
```

Note: In the case of Ubuntu12.04, the message for DEBUG may be outputted at the time of the above-mentioned command execution, but please continue setup as is.


```
$ sudo restart nova-network
nova-network start/running, process 11642
$
```

```
$ sudo restart nova-api
nova-api start/running, process 11695
$
```

```
$ sudo restart nova-compute
nova-compute start/running, process 11710
$
```

```
$ sudo restart nova-scheduler
nova-scheduler start/running, process 11780
$
```

```
$ sudo restart glance-api
glance-api start/running, process 11876
$
```

```
$ sudo restart glance-registry
glance-registry start/running, process 11883
$
```

In the case of Ubuntu12.04, also perform the following.

```
$ sudo restart nova-cert
nova-scheduler start/running, process 11751
$
```

4. 1. 5 Project creation for D-Cloud

An administrator is created as follows. "nova-admin" is created here.

```
$ sudo nova-manage user admin nova-admin
.
.
$
```

A project is created as follows. "d-cloud" is created here.

```
$ sudo nova-manage project create d-cloud nova-admin
$
```

A network is set up as follows. Set the red character portions as appropriate.

```
$ sudo nova-manage network create public --bridge=br100 --bridge_interface=eth0
--fixed_range_v4=10.0.1.0/24 --num_network=1 --network_size=256
$
```

The network setup is checked as follows.

```
$ sudo nova-manage network list
id IPv4 IPv6 start address DNS1 DNS2 VlanID project uuid
1 10.0.1.0/24 None 10.0.1.2 8.8.4.4 None None None None
$
```

A configuration file is developed as follows. Please set the path which is deployed as appropriate.

```
$ sudo nova-manage project zipfile d-cloud nova-admin ./novacred.zip
$
$ unzip novacred.zip
Archive: novacred.zip
  extracting: novarc
  extracting: pk.pem
  extracting: cert.pem
  extracting: cacert.pem
$
$ source novarc
$
```

The port used with a virtual machine is opened as follows. Here, the DS-Bench target daemon's port number is entered as "9804", the default value. Make settings according to the environment in use.

```
$ euca-authorize -P icmp -t -1:-1 default
$
$ euca-authorize -P tcp -p 22 default
$
$ euca-authorize -P tcp -p 9804 default
$
$ euca-authorize -P tcp -p 5001 default
$
```

4.2 The construction procedure of an OpenStack computing node

When a computing node is required, install and make settings as necessary using the following procedures. If a computing node is added, a virtual machine can be newly created using the computational resource of the computing node.

4.2.1 Adding a host name

Please refer to "4.1.2 Adding a host name" for the procedure.

4.2.2 Adding a Repository

Please refer to "4.1.3 Adding a Repository" for the procedure.

4.2.3 Installation of OpenStack (computing node)

Only the package required for the computing node is installed.

```
$ sudo apt-get install -y nova-compute
Reading package lists... 0%
.
.
.
Processing triggers for python-support ...
$
```

OpenStack related authority is changed as follows.

```
$ sudo chown -R nova:root /etc/nova
$
$ ls -ltrR /etc/nova
/etc/nova:
total 12
-rw-r--r-- 1 nova root 19 2011-09-23 03:25 nova-compute.conf
-rw-r--r-- 1 nova root 3080 2011-09-23 02:53 api-paste.ini
-rw----- 1 nova root 254 2011-09-23 02:53 nova.conf
$
```

The configuration(/etc/nova/nova.conf) file of OpenStack is corrected so that the content is the same as the file corrected in 4.1.4 "Installation of OpenStack".

Note: You may copy the file.

The process is started as follows.

```
$ sudo restart nova-compute
nova-compute start/running, process 11710
$
```

Check the section below to verify whether the computing node has been added.

```
$ sudo nova-manage service list
Binary      Host          Zone    Status  State  Updated_At
nova-network controller    nova    enabled :-)    2012-03-06 05:26:37
nova-compute controller    nova    enabled :-)    2012-03-06 05:26:40
nova-scheduler controller    nova    enabled :-)    2012-03-06 05:26:40
nova-compute compute      nova    enabled :-)    None
$
```

Note: In the case of Ubuntu12.04, "nova- cert" also is displayed.

5 DS-Bench

5.1 Compatibility with Ubuntu12.04

Refer to "3.1 Compatibility with Ubuntu12.04" for the method of establishing compatibility.

5.2 Controller installation procedure

5.2.1 Setup of CPAN shell

This operation is performed in order to install the CPAN module during the controller installation process. Thus, the module does not need to be installed here. **Moreover, if a Perl module has ever been installed using CPAN shell, the operation described in this section is unnecessary.**

Execute the following command when you use CPAN for the first time.

```
$ sudo perl -MCPAN -e shell
```

During execution, prompts will appear to input settings for using CPAN. Please input values according to the environment in use. After making the settings, the prompt returns. If the settings are satisfactory, please save them by inputting "o conf commit". Then, CPAN shell is ended by inputting "q."

```
cpan[1]> o conf commit  
cpan[2]> q
```

5.2.2 Installation of DS-Bench controller

5.2.2.1 Installing from a debian package

Please move to a folder with the deb package of a DS-Bench target and a D-Cloud controller, and execute the following commands.

```
$ sudo dpkg -i --force-depends ctrl.deb  
$ sudo apt-get install -f
```

In an environment that already uses Apache, please specify the DocumentRoot of Apache as the DocumentRoot of DS-Bench. When Apache is not installed, install it as a dependent package of DS-Bench controller. In that case, please specify "/var/www" as the DocumentRoot of DS-Bench.

Set the port number, IP address, etc. according to the environment in use. Here, the IP address of the D-Cloud controller is set to 192.168.1.2.

5.2.3 Setup of Apache

Please add the following setup to "/etc/apache2/sites-available/default". At that time, please set the DocumentRoot of Apache to be "\$ {DocumentRoot}". When there is the same "Directory" block already, we recommend that it be deleted or commented out.

```
<Directory ${DocumentRoot}>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>

<Directory "${DocumentRoot}/cgi-bin">
    DirectoryIndex main.cgi
    AllowOverride None
    AddHandler cgi-script .cgi .pl
    Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
    Order allow,deny
    Allow from all
</Directory>
```

After a setup finishes, **be sure to reboot Apache**. Moreover, although this system uses CGI, it does not need "ScriptAlias". Remove "ScriptAlias" from applicable directories.

5.2.4 Settings of a Sedna DB server (optional changes)

A setting of a Sedna DB server can be changed by editing the "/var/dsbench/config/sedna.conf" file as follows. Settings can also be changed in the DS-Bench screen later.

```
SEDNA_HOST      : 192.168.1.4  
SEDNA_DB_NAME   : dsbench  
SEDNA_DB_USERNAME : SYSTEM  
SEDNA_DB_PASSWORD : MANAGER  
SEDNA_COLLECTION_RESULT_BENCHMARK : result_benchmark  
SEDNA_COLLECTION_RESULT_BENCHMARK_SET : result_benchmark_set
```

5.2.5 Settings of a D-Cloud controller(optional changes)

The settings of a D-Cloud controller can be changed by editing the "/var/dsbench/config/cloud.conf" file as follows. Settings can also be changed in the DS-Bench screen later.

```
CLOUD_CONTROLLER : 192.168.1.2  
CLOUD_PORT       : 9903
```

5.3 Installation of a SednaDB server

5.3.1 Installation from debian package

Please move to the folder with the deb package of Sedna, and execute the following command.

```
$ sudo dpkg -i Sedna.deb
```

During the installation, prompts for entering a database, etc. will appear. Please set these according to your environment. Here, the default value is used.

```
Input DB Name for DSbench [ dsbench ]:  
Input Benchmark Collection Name for DSbench [ result_benchmark ]:
```

5.3.2 Modification of Sedna

Please download and thaw Sedna-0.002.tar.gz from the following URL.

<http://www.cpan.org/authors/id/D/DR/DRUOSO/>

Please change the 353rd line of the Sedna.xs file in the thawed folder as follows.

```
(before change) memcpy((char*)((int)result + curlen), buffer, ret);  
(after change) memcpy((result + curlen), buffer, ret);
```

When the change is completed, please save, compile, and install it.

```
$perl Makefile.PL  
$make  
$make test  
$sudo make install
```


5.4 Installation of a DS-Bench target

5.4.1 Installation from debian package

Please move to a folder with the deb package of Sedna, and execute the following commands.

```
$ sudo dpkg -i --force-depens target.deb
$ sudo apt-get install -f
```

During the installation, the following prompts for entering daemons will appear. Please make settings according to your environment. Here, the default values are used.

```
Input Port Number for DSbench-Controller Daemon [ 9803 ]:
Input Port Number for DSbench-Target Daemon [ 9804 ]:
```

5.4.2 Modification of the repository file

The repository file is modified as follows in order to install a benchmark program.

Since the repository is created when installing a DS-Bench controller, the DS-Bench controller is made the referent. An example of the modification is shown below.

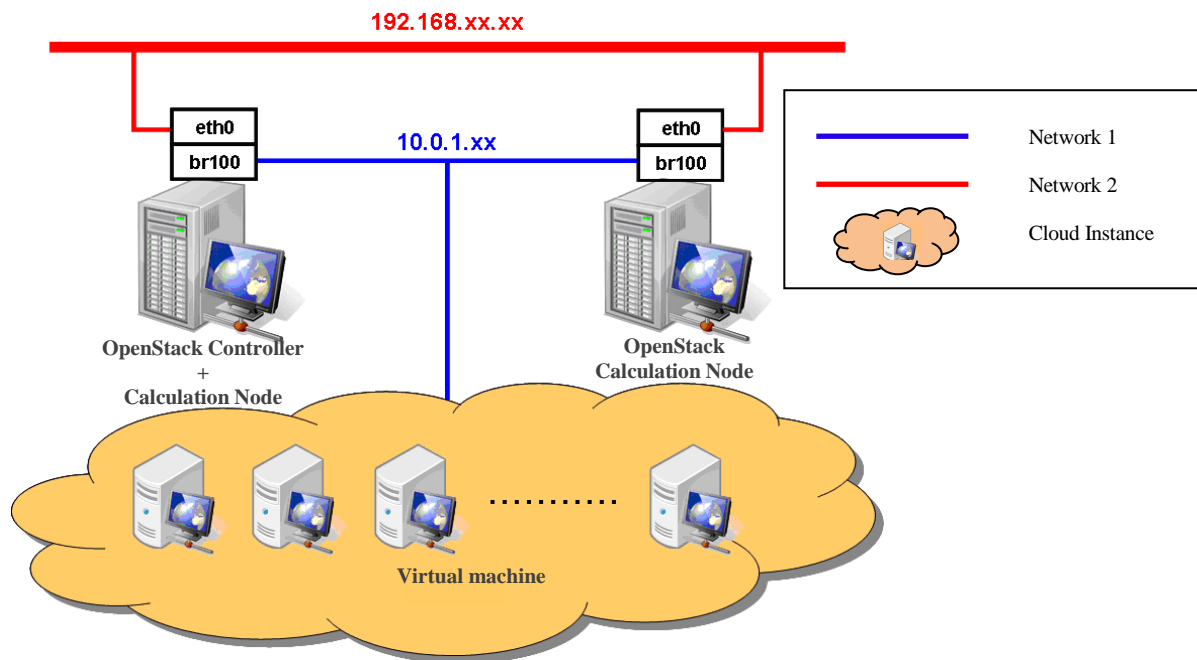
```
$ cat /etc/apt/sources.list.d/dsb.list
deb http://[ IP of a controller or host name]/deb/ ./
deb-src http://[ IP of a controller or host name]/deb/ ./
```

After the repository file modification, do not fail to update the package list

```
$ sudo apt-get update
```

6 Virtual machine

6.1 Setup of the network for virtual target use



To enable the communication between virtual machines (IP address 10.0.1.X) and the DS-Bench controller (IP address 192.168.X.X), you must configure the routing table of DS-Bench controller.

Note: For communication between the physical target of DS-Bench and a virtual target, also set up a routing table on a physical target. Route information is added to "/etc/network/interface" as follows.

```
$ sudo vi /etc/network/interface
up route add -net 10.0.1.0 netmask 255.255.255.0 gw D-CloudIP address
```

Reboot the network after the setup and verify that the route command reflects the settings. Moreover, verify that a ping command is transmitted.

6.2 Procedure of virtual target image creation

6.2.1 Registration of initial image

Upload to the controller server the image that you download from the formal homepage of Ubuntu

<http://uec-images.ubuntu.com/>

as follows. Navigate to the directory where the image file is stored.

```
$ cd /var/nova/images/test/
$ pwd
/var/nova/images/test/
```

Refer to the settings of Ubuntu12.04 or Ubuntu10.04 suited to your environment.

For Ubuntu10.04

- image file `/var/nova/images/test/lucid-server-cloudimg-amd64.img`
- kernel file `/var/nova/images/test/lucid-server-cloudimg-amd64-vmlinuz-virtual`

Register the virtual kernel to OpenStack. Here, the registration name "org-kernel" is specified.

```
$ sudo nova-manage image kernel_register lucid-server-cloudimg-amd64-vmlinuz-virtual
admin --name=org-kernel
Image registered to 1 (00000001).
```

An image is registered into OpenStack. Here, the registration name "org-img" is specified.

The kernel which was registered a little while ago is specified.

```
$ sudo nova-manage image image_register lucid-server-cloudimg-amd64.img admin
--name=org-img --kernel=00000001 --cont_format=ami --disk_format=ami
```

Verify that the image is registered.

```
$ glance index
```

ID	Name	Disk Format	Container	Format	Size
2	org-img	ami	ami	1476395008	
1	org-kernel	aki	aki	4108960	

For Ubuntu12.04

- Image file `/var/nova/images/test/precise-server-cloudimg-amd64-disk1.img`

Register the virtual image to OpenStack. Here, the registration name "ubuntu-12.04" is specified.

```
$ glance add name="ubuntu-12.04 " is_public=true container_format=ovf disk_format=qcow2
< precise-server-cloudimg-amd64-disk1.img
  Uploading image 'ubuntu-12.04'
  ===== [100%] 31.3M/s, ETA 0h 0m 0s
  Added new image with ID: 4ec748ac-8e05-40b6-8de6-2cef2be762c0
```

Verify that the virtual image is registered.

```
$ glance index
ID                Name                Disk Format  Container Format  Size
-----
4ec748ac-8e05... ubuntu-12.04       qcow2       ovf                233832448
```

6.2.2 Starting the initial image

To start the initial image:

Create a key in order to be able to log in with ssh. Create "`~/ssh`" directory as follows.

```
$ euca-add-keypair orgkey > ~/.ssh/orgkey
$
```

Change the permission of the created key.

```
$ chmod 600 ~/.ssh/orgkey
$
```

Check the ID of the initial image.

```
$source novarc
$ euca-describe-images
IMAGE  ami-00000001  None (ubuntu-12.04)  available  public  machine  instance-store
```

Start an instance of the initial image using the key that was created.

```
$ euca-run-instances -t m1.tiny -k orgkey ami-00000001
RESERVATION      r-i40e30j5    [host name] default
INSTANCE         i-00000001    ami-00000001    server-1 server-1 pending
                  orgkey ([host name], None)      0
$
```

Verify the start of the instance. After the completion of the start, an IP address is assigned. The status then becomes "running".

```
$ euca-describe-instances
RESERVATION      r-931xv678    d-cloud default
INSTANCE         i-00000001    ami-00000002    10.0.1.2  10.0.1.2  running orgkey ([host
name], [server name])  0  m1.tiny [date time]  nova  aki-00000001  ami-00000000
```

6.2.3 Create the virtual target image

Log in to an instance of an initial image via ssh. Use the key which was created in Section 3.2. Here, the user name is "ubuntu".

```
$ ssh -i ~/.ssh/orgkey ubuntu@10.0.1.2
The authenticity of host '10.0.1.2 (10.0.1.2)' can't be established.
RSA key fingerprint is 94:f5:22:ef:a0:15:ad:1a:d6:a0:43:b7:8d...
Are you sure you want to continue connecting (yes/no)? yes
.
.
.
ubuntu@server-1:~$
```

Install the DS-Bench target as follows. Using the scp command, copy a target.deb package to the instance that has been started, and install it referring to the procedure of Section 5.4.

```
$ sudo scp -i ~/.ssh/orgkey "DS-Bench target package" ubuntu@10.0.1.2:/home/ubuntu/
```

Since a repository file is initialized when a virtual machine starts, operations need to be carried out separately. The procedure is as follows.

The repository file to be used is created in the home directory.

```
ubuntu@server-1:~$ vi ~/sources.list
```

The contents of the repository file become as follows.

```
deb http://[ IP of a controller or host name]/deb/ ./
deb-src http:// [IP of a controller or host name]/deb/ ./
```

Confirm the repository file that you created.

```
$ cat ~/sources.list
deb http://[ IP of a controller or host name]/deb/ ./
deb-src http:// [IP of a controller or host name]/deb/ ./
```

In order to apply the repository file created when the machine was started, add the steps for copying over the original repository file to the startup script.

```
$ vi /etc/rc.local
#!/bin/sh -e
sleep 5
logger -t initsourceslist cp /home/ubuntu/sources.list /etc/apt/sources.list
cp /home/ubuntu/sources.list /etc/apt/sources.list
exit 0
```

Change the order of execution of the startup script.

```
$ sudo mv /etc/rc2.d/S99grub-common /etc/rc2.d/S97grub-common
$
$ sudo mv /etc/rc2.d/S99ondemand /etc/rc2.d/S97ondemand
$
$ sudo mv /etc/rc2.d/S99rc.local /etc/rc2.d/S98rc.local
$
$ sudo mv /etc/rc2.d/S20ds-target /etc/rc2.d/S99ds-target
```

Shut down the virtual machine.

```
ubuntu@server-1:~$ sudo shutdown -h now
Broadcast message from ubuntu@server-1
      (/dev/pts/0) at ...
.
.
$
```

Create an image of the shut-down instance. Verify the directory of the instance.

```
$ ls -l /var/lib/nova/instances/
total 8
drwxr-xr-x 2 nova nogroup 4096 yyyy-mm-dd hh:mi _base
drwxr-xr-x 2 nova nogroup 4096 yyyy-mm-dd hh:mi instance-00000001
$
```

Create an image. Here, the name of the created image is "**target-img**", and "**/var/nova/images/test**" is specified as the path of the directory to save.

```
$ qemu-img convert -f qcow2 /var/lib/nova/instances/instance-00000001/disk -o  
raw /var/nova/images/test/target-img  
$
```

Check the image that is created.

- Ubuntu10.04

```
$ ls -l /var/nova/images/test/  
total 1991356  
-rw-r--r-- 1 root root 1476395008 yyyy-mm-dd hh:mi lucid-server-cloudimg-amd64.img  
-rw-rw-rw- 1 root root 4110432 yyyy-mm-dd hh:mi lucid-server-cloudimg-amd64-vmlinuz-virtual  
-rw-r--r-- 1 root root 1476395008 yyyy-mm-dd hh:mi target-img  
$
```

- Ubuntu12.04

```
$ ls -l /var/nova/images/test/  
total 1991356  
-rw-r--r-- 1 root root 1476395008 yyyy-mm-dd hh:mi precise-server-cloudimg-amd64-desk1.img  
-rw-r--r-- 1 root root 1476395008 yyyy-mm-dd hh:mi target-img  
$
```


6.2.4 Registering a virtual target machine

Register to OpenStack the target image that you created as follows. Go to the directory where the image file is stored.

```
$ cd /var/nova/images/test/
$ pwd
/var/nova/images/test/
```

Register a virtual target image. The kernel used is the initial kernel.

- Ubuntu10.04

```
$ sudo nova-manage image image_register target-img admin --name=target-img --kernel=00000001
--cont_format=ami --disk_format=ami
Image registered to 3 (00000003).
IMAGE aki-00000001    local (testkernel)    available    public        x86_64    kernel
```

Verify that the instance is registered.

```
$ glance index
```

ID	Name	Disk Format	Container Format	Size
3	target-img	ami	ami	1476395008
2	org-img	ami	ami	1476395008
1	org-kernel	aki	aki	4108960

- Ubuntu12.04

```
$ glance add name="12.04-target" is_public=true container_format=ovf disk_format=qcow2 <
target-img
  Uploading image '12.04-target'
===== [100%] 31.3M/s, ETA 0h 0m 0s
Added new image with ID: 4ec748ac-8w35...    public        x86_64    kernel
```

Verify that the instance is registered.

```
$ glance index
```

ID	Name	Disk Format	Container Format	Size
4ec748ac-8w35-40b6-8de6---	12.04-target	qcow2	ovf	2147483648
4ec748ac-8e05-40b6-8de6---	ubuntu-12.04	qcow2	ovf	233832448

After registration, **reboot the daemon of D-Cloud without fail.**

```
$ sudo /etc/init.d/dc-mapserver restart
Stopping ... success
Starting ... success
$
$ sudo /etc/init.d/dc-resource-dsb restart
Stopping ... success
Starting ... success
$
$ sudo /etc/init.d/dclld restart
Stopping ... success
Starting ... success
$
```

From a DS-Bench screen, verify that Target (virtual machine) is registered.

Main screen> Configuration screen> Target list

Target list			
No	Machine name	Type	Status
1	trg-tutorial	physical	Active
2	target-img-3-m1.tiny.00	virtual	Power off
3	target-img-3-m1.tiny.01	virtual	Power off
4	target-img-3-m1.tiny.02	virtual	Power off

Since the value(<instances> <limit>) of the configuration file(/ var / dcloud / config / resource_dsb.xml) of D-Cloud is "3", number of images displayed has become "3".

The set value above is the maximum start-up number of virtual machines, so please be careful when you change this.



DEOS Project