

# Accomplishment: Summary of Work

August 3, 2011

GSDC  
Supercomputing Center

Seo-Young Noh  
rsyoung@kisti.re.kr



# Contents

- 1. Goals of Visiting Research**
- 2. Virtual Machine Launching at Amazon EC2**
- 3. Building Virtual Cluster: FermiCloud + Amazon EC2**
- 4. Building OSG Site at FermiCloud**
- 5. vcluster: Automatic Scalable Virtual Cluster System**
- 6. Conclusions**

# 1. Goals of Visiting Research

## ■ CDF Support

- ➔ CDF On-Site Support (Every Wednesday Offline Computing Meeting)
- ➔ OSG reconfiguration according to CDF's request (May 16 ~ May 20)
- ➔ MOU Extension (July 21)

## ■ OSG Summer School Attending

- ➔ Held at University of Wisconsin Madison (June 26 ~ June 30)
- ➔ Helpful to understand the concept of OSG from lectures and exercises

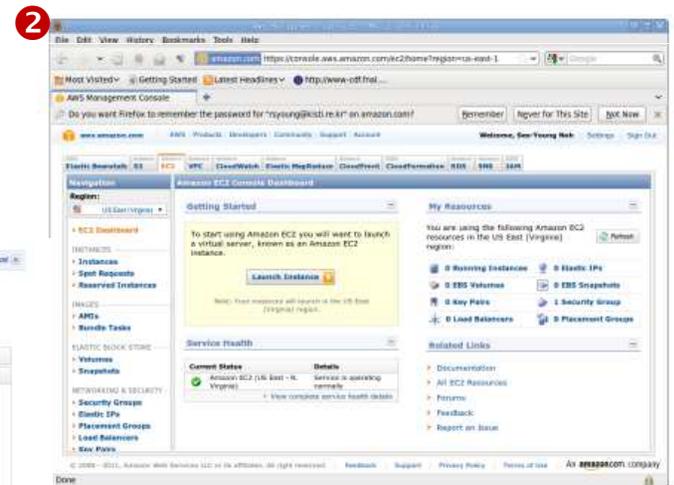
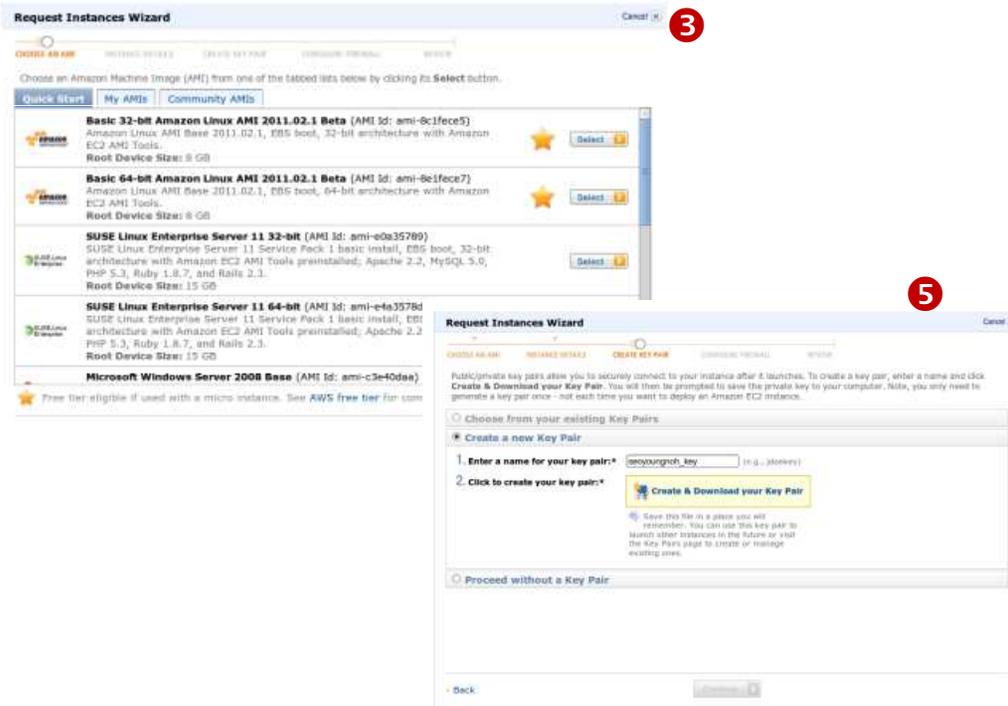
## ■ Understanding FermiCloud System + alpha (Contribution)

- ➔ Current Status and Planning
- ➔ GSDC has a plan to build a very similar system soon

# 2. Virtual Machine Launching at Amazon EC2

## Steps of Launching VM

1. Create an account & Connect to Web Console: <http://aws.amazon.com/console/>
2. Launch Instance
3. Create or Select Virtual Machine
4. Create KeyPair



## 2. Virtual Machine Launching at Amazon EC2

### ■ Private Key Mode Change

```
rsyoung@descartes:~/amz-ec2$ ls -al
total 12
drwxr-xr-x  2 rsyoung rsyoung 4096 2011-05-23 22:28 .
drwxr-xr-x  73 rsyoung rsyoung 4096 2011-05-23 22:28 ..
-rw-r--r--  1 rsyoung rsyoung 1692 2011-05-23 22:19 seoyoungnoh_key.pem
rsyoung@descartes:~/amz-ec2$ chmod 400 seoyoungnoh_key.pem
rsyoung@descartes:~/amz-ec2$
rsyoung@descartes:~/amz-ec2$ ls -al
total 12
drwxr-xr-x  2 rsyoung rsyoung 4096 2011-05-23 22:28 .
drwxr-xr-x  73 rsyoung rsyoung 4096 2011-05-23 22:28 ..
-r-----  1 rsyoung rsyoung 1692 2011-05-23 22:19 seoyoungnoh_key.pem
rsyoung@descartes:~/amz-ec2$
```

Change the mode of  
**<key>.pem** to **400**

### ■ ssh logon to VM

```
rsyoung@descartes:~/amz-ec2$ ssh -i seoyoungnoh_key.pem \
    ec2-user@ec2-184-73-87-66.compute-1.amazonaws.com

The authenticity of host 'ec2-184-73-87-66.compute-1.amazonaws.com
(184.73.87.66)' can't be established.
RSA key fingerprint is 78:c6:10:a9:94:ea:40:fb:5e:3c:5d:d6:9a:8c:72:d8.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ec2-184-73-87-66.compute-1.amazonaws.com,
184.73.87.66' (RSA) to the list of known hosts.

  ____|  __|_  )   Amazon Linux AMI
 _| (      /      Beta
  __|\____|____|

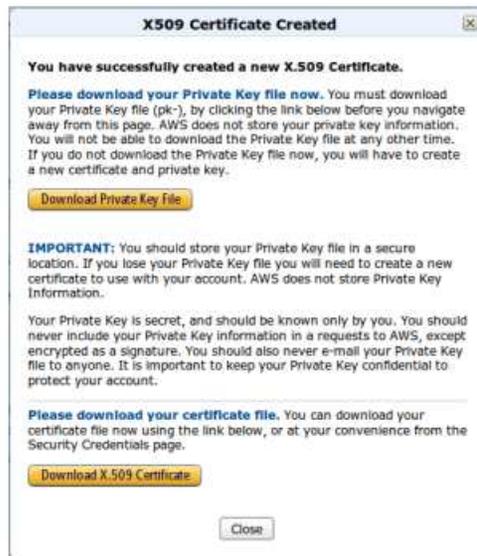
See /usr/share/doc/system-release-2011.02 for latest release notes. :-)
[ec2-user@domU-12-31-39-06-86-C5 ~]$
```

User ID should be always  
**ec2-user**

When opening an account, you  
should provide your credit card  
information

## 2. Virtual Machine Launching at Amazon EC2

- Management of VMs using EC2 API Tool
  - ➔ Amazon EC2 API tools serve as the client interface to the Amazon EC2 web service.
  - ➔ Able to register and launch instances, manipulate security groups, and more.
  - ➔ <http://aws.amazon.com/developertools/351>
- Download X.509 certificate & private key
  - ➔ Logon to <http://aws.amazon.com/security/>
  - ➔ Click AWS Security Credentials
  - ➔ Click X.509 Certificates Tab and click Create a new Certificate



Downloaded X.509 cert,  
private key & EC2 API Tool

```
rsyoung@descartes:~/ec2$ ls
cert-U6TIKUYTPVBEZKXMMIN6OZ6LJWKLM7WB.pem
ec2-api-tools.zip
pk-U6TIKUYTPVBEZKXMMIN6OZ6LJWKLM7WB.pem
rsyoung@descartes:~/ec2$ mv pk-U6TIKUYTPVBEZKXMMIN6OZ6LJWKLM7WB.pem \
ec2-private-key.pem

rsyoung@descartes:~/ec2$ mv cert-U6TIKUYTPVBEZKXMMIN6OZ6LJWKLM7WB.pem \
ec2-cert.pem

rsyoung@descartes:~/ec2$ ls
ec2-api-tools.zip ec2-cert.pem ec2-private-key.pem
rsyoung@descartes:~/ec2$
```

## 2. Virtual Machine Launching at Amazon EC2

### ■ Installation

```
[rsyoung@fermicloud048 ~]$
download ec2-api-tools.zip OpenNebula
[rsyoung@fermicloud048 ~]$ unzip ec2-api-tools.zip
Archive:  ec2-api-tools.zip
  creating:  ec2-api-tools-1.4.3.0/
  inflating:  ec2-api-tools-1.4.3.0/THIRDPARTYLICENSE.TXT
  creating:  ec2-api-tools-1.4.3.0/bin/
  inflating:  ec2-api-tools-1.4.3.0/bin/ec2-activate-license

----- 종 락 -----

  inflating:  ec2-api-tools-1.4.3.0/lib/xml-apis.jar
  inflating:  ec2-api-tools-1.4.3.0/lib/xmlsec.jar
  inflating:  ec2-api-tools-1.4.3.0/license.txt
  inflating:  ec2-api-tools-1.4.3.0/notice.txt
[rsyoung@fermicloud048 ~]$ ls
download ec2-api-tools-1.4.3.0 ec2-api-tools.zip OpenNebula
[rsyoung@fermicloud048 ~]$ mv ec2-api-tools.zip ./download/
[rsyoung@fermicloud048 ~]$ mv ec2-api-tools-1.4.3.0/ ec2
[rsyoung@fermicloud048 ~]$ ls
download ec2 OpenNebula
[rsyoung@fermicloud048 ~]$
```

I installed EC2 API Tool at fermicloud048

### ■ Setup Environment for EC2 API Tool

```
[rsyoung@fermicloud048 ~]$ cat ec2.bash
export PATH=$PATH:~/ec2/bin
export EC2_HOME=~/ec2
export EC2_PRIVATE_KEY=~/ec2/ec2-private-key.pem
export EC2_CERT=~/ec2/ec2-cert.pem
export JAVA_HOME=/usr/java/default/
[rsyoung@fermicloud048 ~]$
[rsyoung@fermicloud048 ~]$ source ec2.bash
```

X.509 cert and private key should be located at fermicloud048

Java must be installed because EC2 API Tool is a Java program.

## 2. Virtual Machine Launching at Amazon EC2

### ■ Check Version

```
[rsyoung@fermicloud048 ~]$  
[rsyoung@fermicloud048 ~]$ ec2-version  
1.4.3.0 2011-05-15  
[rsyoung@fermicloud048 ~]$
```

### ■ Create KeyPair

```
[rsyoung@fermicloud048 ~]$ ec2-add-keypair rsyoung-keypair > rsyoung-keypair  
[rsyoung@fermicloud048 ~]$ cat rsyoung-keypair  
-----BEGIN RSA PRIVATE KEY-----  
MIIEpAIBAAKCAQEAhQfNYnj0vb6NruJbHJp0NZ7hQp2rauzzQLAjAiBGeUzuf5miQvc+mVs7d8v  
95i2Bp55hicXRrrpnFGd2sELmi69IE3ndv8wBCBOGNff6p2ygNqQ4QS0C2ENKqSh3DoG+UNt3x7C  
  
----- 중 락 -----  
  
JoiMNYkt0uBwVY/mNuCyh4X4c88wBUhpu268zU12DKQpJKnDVBHozv/0xIh2R1f1J+7nIIGPXzhT  
S+7Efwtu1H5RzK07Cp0UcHLsV2aQ5Coh0JNeTKU9vd9FbVTJW6jqR/Lg0o6Yj2B10J+Xjg==  
-----END RSA PRIVATE KEY-----  
[rsyoung@fermicloud048 ~]$
```

### ■ Describe Image

```
[rsyoung@fermicloud048 ~]$ ec2-describe-images -o amazon | grep ami-8c1fece5  
IMAGE ami-8c1fece5 amazon/amzn-ami-2011.02.1.i386-ebs  
amazon available public i386 machine aki-407d9529  
ebs paravirtual xen  
[rsyoung@fermicloud048 ~]$
```

To create a VM from Amazon Machine Image, KeyPair is required.

We can use generated KeyPair previously, but generate a new one using EC2 API Tool

KeyPair should be located in EC2\_HOME directory as defined in ec2.bash

Search for ami-8c1fece5 which is Amazon owned

## 2. Virtual Machine Launching at Amazon EC2

### ■ Run Instance

```
[rsyoung@fermicloud048 ~]$ ec2-run-instances ami-8c1fece5 -k rsyoung-keypair
RESERVATION      r-ad8ae8c1      570421675223    default
INSTANCE         i-03ad1f6d      ami-8c1fece5    pending
rsyoung-keypair  0
2011-06-01T22:16:48+0000    ml.small
us-east-1a      aki-407d9529
monitoring-disabled    ebs    paravirtual    xen
sg-8e0079e7      default
[rsyoung@fermicloud048 ~]$
```

**Instance ID = i-03ad1f6d**

### ■ Describe Instance

```
[rsyoung@fermicloud048 ~]$ ec2-describe-instances i-03ad1f6d
RESERVATION      r-ad8ae8c1      570421675223    default
INSTANCE         i-03ad1f6d      ami-8c1fece5
ec2-50-16-52-237.compute-1.amazonaws.com ip-10-196-175-16.ec2.internal
running rsyoung-keypair 0    ml.small      2011-06-01T22:16:48+0000
us-east-1a      aki-407d9529    monitoring-disabled    50.16.52.237
10.196.175.16    ebs    paravirtual    xensg-8e0079e7    default
```

**Full Host = ec2-50-16-52-237.compute-1.amazonaws.com**

**Status = Running**

## 2. Virtual Machine Launching at Amazon EC2

### ■ Logon to the VM

---

```
[rsyoung@fermicloud048 ~]$ ssh -i ./ec2/rsyoung-keypair \
    ec2-user@ec2-50-16-52-237.compute-1.amazonaws.com

The authenticity of host 'ec2-50-16-52-237.compute-1.amazonaws.com
(50.16.52.237)' can't be established.
RSA key fingerprint is 40:ff:13:8a:18:48:d4:20:71:74:47:03:ab:12:97:0a.
Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added 'ec2-50-16-52-237.compute-1.amazonaws.com,
50.16.52.237' (RSA) to the list of known hosts.

  ____|  __|_  )  Amazon Linux AMI
 _| (    /    Beta
___|\___|___|

See /usr/share/doc/system-release-2011.02 for latest release notes. :-)
[ec2-user@ip-10-196-175-16 ~]$
```

---

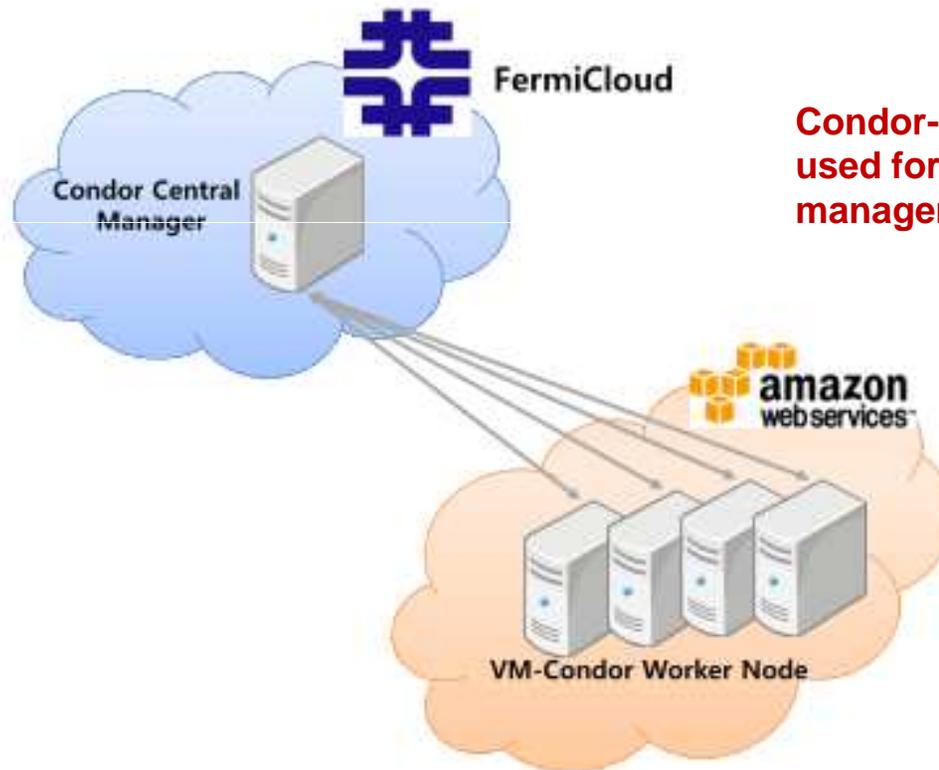
**Not covered today, but there is another tool named “EC2 AMI Tool”, which helps to create Amazon Machine Image.**

[http://aws.amazon.com/developertools/368?\\_encoding=UTF8&jiveRedirect=1](http://aws.amazon.com/developertools/368?_encoding=UTF8&jiveRedirect=1)

# 3. Building Virtual Cluster: FermiCloud + Amazon EC2

## ■ Concept

- ➔ Build a cluster system using FermiCloud + Amazon EC2 computing resource
- ➔ Install condor central manager at FermiCloud
- ➔ Create virtual worker nodes at Amazon EC2



Condor-7.7.0 prerelease version was used for both condor central manager and condor worker nodes

# 3. Building Virtual Cluster: FermiCloud + Amazon EC2

## ■ Install Condor Central Manager

URL: <http://www.cs.wisc.edu/condor/downloads-v2/download.pl>

No RPM provided. Install script was used

```
[root@fermicloud005 condor-7.7.0]# ls
bin          DOC          include  libexec      README
condor_configure  etc          INSTALL  LICENSE-2.0.txt  sbin
condor_install  examples    lib       man           src
[root@fermicloud005 condor-7.7.0]# ./condor_install --install=. \
--prefix=/opt/condor_7_7_0 --type=manager,submit,execute \
--owner=condor

Installing Condor from /home/rsyoung/download/condor-7.7.0 to /opt/condor_7_7_0

Condor has been installed into:
/opt/condor_7_7_0
```

```
Configured condor using these configuration files:
global: /opt/condor_7_7_0/etc/condor_config
local: /opt/condor_7_7_0/local.fermicloud005/condor_config.local
```

In order for Condor to work properly you must set your CONDOR\_CONFIG environment variable to point to your Condor configuration file: /opt/condor\_7\_7\_0/etc/condor\_config before running Condor commands/daemons. Created scripts which can be sourced by users to setup their

```
Condor environment variables. These are:
sh: /opt/condor_7_7_0/condor.sh
csh: /opt/condor_7_7_0/condor.csh
```

```
[root@fermicloud005 condor-7.7.0]#
```

Current Condor will be

Condor Central Manager  
Submitter  
Executor

Local & Global  
Environment variables  
need to change

Need to source ether  
condor.sh or condor.csh

# 3. Building Virtual Cluster: FermiCloud + Amazon EC2

## ■ Configuration

```
[root@fermicloud005 etc]# diff condor_config condor_config.old
55c55
< RELEASE_DIR          = /opt/condor_7_7_0
---
> RELEASE_DIR          = /usr/local/condor
84c84
< CONDOR_ADMIN         = rsyoung@kisti.re.kr
---
> CONDOR_ADMIN         = root>${FULL_HOSTNAME}
223,224c223,224
< #ALLOW_WRITE = ${FULL_HOSTNAME}, ${IP_ADDRESS}
< ALLOW_WRITE = *.fnal.gov, *.compute-1.amazonaws.com
---
> ALLOW_WRITE = ${FULL_HOSTNAME}, ${IP_ADDRESS}
> #ALLOW_WRITE = *.your.domain, your-friend's-machine.other.domain
1041,1042c1041
< UPDATE_COLLECTOR_WITH_TCP = True
< COLLECTOR_SOCKET_CACHE_SIZE = 1000
---
> #UPDATE_COLLECTOR_WITH_TCP = True
1056,1057c1055,1056
< HIGHPORT = 40040
< LOWPORT = 40000
---
> #HIGHPORT = 9700
> #LOWPORT = 9600
1073c1072
< DAEMON_LIST = MASTER, STARTD, SCHEDD, COLLECTOR, NEGOTIATOR
---
> DAEMON_LIST = MASTER, STARTD, SCHEDD
[root@fermicloud005 etc]#
```

**Not covered all, but most important ones.**

**Need to add the domain name of EC2's virtual machines to join the condor central pool**

**Set HIGHPORT & LOWPORT with Amazon's firewall port range**

# 3. Building Virtual Cluster: FermiCloud + Amazon EC2

## Start Condor Central Manager

```
[root@fermicloud005 /]# source /opt/condor_7_7_0/condor.sh
[root@fermicloud005 /]#
[root@fermicloud005 /]# /opt/condor_7_7_0/sbin/condor_master
[root@fermicloud005 /]#
[root@fermicloud005 /]# ps -ef | grep condor_ | grep -v grep
condor 21872 1 0 23:12 ? 00:00:00 condor_master
condor 21873 21872 0 23:12 ? 00:00:00 condor_collector -f
condor 21874 21872 0 23:12 ? 00:00:00 condor_negotiator -f
condor 21875 21872 0 23:12 ? 00:00:00 condor_schedd -f
condor 21876 21872 0 23:12 ? 00:00:00 condor_startd -f
root 21877 21875 0 23:12 ? 00:00:00 condor_procd -A
/tmp/condor-lock.fermicloud0050.477419000891135/procd_pipe.SCHEDD
-R 10000000 -S 60 -C 4716
[root@fermicloud005 /]#
```

Source & execute  
condor\_master

Condor\_master  
Collector  
Match Maker  
Submitter  
Executor

## Check Condor Pool

```
[root@fermicloud005 /]# condor_status
```

Name	OpSys	Arch	State	Activity	LoadAv	Mem	ActvtyTime
slot1@fermicloud00	LINUX	X86_64	Owner	Idle	0.080	1974	0+00:00:04
slot2@fermicloud00	LINUX	X86_64	Owner	Idle	0.000	1974	0+00:00:05
Total Owner Claimed Unclaimed Matched Preempting Backfill							
	X86_64/LINUX	2	2	0	0	0	0
	Total	2	2	0	0	0	0

```
[root@fermicloud005 /]#
```

Two slots are available  
because fermicloud005  
has two cores and we set  
it as an executor

# 3. Building Virtual Cluster: FermiCloud + Amazon EC2

## ■ Work Node VM Creation at Amazon EC2

```
rsyoung@descartes:~$ source ./ec2/ec2.bash
rsyoung@descartes:~$
rsyoung@descartes:~$ ec2-add-group condor -d "condor worker node"
GROUP      sg-f453159d      condor condor worker node
rsyoung@descartes:~$
rsyoung@descartes:~$ ec2-authorize condor -P tcp -p 40000-40050 \
-s 131.225.154.62/32

GROUP      condor
PERMISSION condor  ALLOWS  tcp      40000   40050
FROM      CIDR    131.225.154.62/32      ingress
rsyoung@descartes:~$
rsyoung@descartes:~$ ec2-authorize condor -P udp -p 40000-40050 \
-s 131.225.154.62/32

GROUP      condor
PERMISSION condor  ALLOWS  udp      40000   40050
FROM      CIDR    131.225.154.62/32      ingress
rsyoung@descartes:~$
rsyoung@descartes:~$ ec2-authorize condor -p 22
GROUP      condor
PERMISSION condor  ALLOWS  tcp      22      22
FROM      CIDR    0.0.0.0/0      ingress
rsyoung@descartes:~$
```

**Need to create a security group - condor**

**Set TCP port range**

**Set UDP port range**

**Open ssh port (22)**

# 3. Building Virtual Cluster: FermiCloud + Amazon EC2

## ■ Run Instance

```
rsyoung@descartes:~$ ec2-run-instances ami-8elfece7 -d fermicloud005.fnal.gov \  
-t m1.large -k rsyong-keypair -g condor  
  
RESERVATION    r-47fe972b      570421675223    condor  
INSTANCE       i-eld66d8f      ami-8elfece7    pending rsyong-keypair 0  
m1.large       2011-06-03T04:50:39+0000    us-east-1d      aki-427d952b  
monitoring-disabled ebs      paravirtual     xen      sg-f453159d      default  
rsyoung@descartes:~$  
rsyoung@descartes:~$ ec2-describe-instances i-eld66d8f
```

Launch a VM using  
EC2 API Tool

-d = user data  
-g = security group

## ■ Logon to the Instance

```
rsyoung@descartes:~$ ssh -i ./ec2/rsyong-keypair \  
ec2-user@ec2-50-16-165-161.compute-1.amazonaws.com  
  
The authenticity of host 'ec2-50-16-165-161.compute-1.amazonaws.com  
(50.16.165.161)' can't be established.  
RSA key fingerprint is 21:81:68:ee:22:44:62:37:a8:c8:f9:7a:59:70:aa:fd.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added 'ec2-50-16-165-161.compute-1.amazonaws.com,  
50.16.165.161' (RSA) to the list of known hosts.  
Last login: Fri Jun  3 22:17:18 2011 from descartes.dhcp.fnal.gov  
  
  __|  __|_ ) Amazon Linux AMI  
  _| ( / Beta  
  ___|\___|___|  
  
See /usr/share/doc/system-release-2011.02 for latest release notes. :-)  
[ec2-user@ip-10-111-45-200 ~]$
```

# 3. Building Virtual Cluster: FermiCloud + Amazon EC2

## ■ How to Check Meta Data of Instance

```
[ec2-user@ip-10-111-45-200 ~]$ wget -q \  
http://instance-data.ec2.internal/latest/meta-data
```

```
[ec2-user@ip-10-111-45-200 ~]$  
[ec2-user@ip-10-111-45-200 ~]$ cat index.html  
ami-id  
ami-launch-index  
ami-manifest-path  
block-device-mapping/  
hostname  
instance-action  
instance-id  
instance-type  
kernel-id  
local-hostname  
local-ipv4  
mac  
network/  
placement/  
profile  
public-hostname  
public-ipv4  
public-keys/  
reservation-id  
security-groups  
[ec2-user@ip-10-111-45-200 ~]$
```

Using wget, we can download meta data of running instance.

When installing condor worker node, some information is needed.

### To get the public hostname

```
[ec2-user@ip-10-111-45-200 ~]$ wget -q \  
http://instance-data.ec2.internal/latest/meta-data/public-hostname
```

```
[ec2-user@ip-10-111-45-200 ~]$  
[ec2-user@ip-10-111-45-200 ~]$ cat public-hostname  
ec2-50-16-165-161.compute-1.amazonaws.com  
[ec2-user@ip-10-111-45-200 ~]$
```

# 3. Building Virtual Cluster: FermiCloud + Amazon EC2

## ■ Install Condor Work Node

---

```
[root@ip-10-111-45-200 condor-7.7.0]# ./condor_install --install=. \
--prefix=/opt/condor_7_7_0 --type=execute \
--owner=condor
```

Type = execute  
Owner = condor

```
Installing Condor from /home/ec2-user/download/condor-7.7.0 to /opt/condor_7_7_0
```

```
Condor has been installed into:
/opt/condor_7_7_0
```

User id, condor,  
should be present.

```
Configured condor using these configuration files:
global: /opt/condor_7_7_0/etc/condor_config
local: /opt/condor_7_7_0/local.ip-10-111-45-200/condor_config.local
```

```
In order for Condor to work properly you must set your CONDOR_CONFIG
environment variable to point to your Condor configuration file:
/opt/condor_7_7_0/etc/condor_config before running Condor commands/daemons.
Created scripts which can be sourced by users to setup their
Condor environment variables. These are:
sh: /opt/condor_7_7_0/condor.sh
csh: /opt/condor_7_7_0/condor.csh
```

Create user condor with nologin option

```
[root@ip-10-111-45-200 condor-7.7.0]#
```

---

```
[ec2-user@ip-10-111-45-200 condor-7.7.0]$ sudo su
[root@ip-10-111-45-200 condor-7.7.0]#
[root@ip-10-111-45-200 condor-7.7.0]# useradd -s /sbin/nologin condor
[root@ip-10-111-45-200 condor-7.7.0]#
[root@ip-10-111-45-200 condor-7.7.0]# grep condor /etc/passwd
condor:x:500:501::/home/condor:/sbin/nologin
[root@ip-10-111-45-200 condor-7.7.0]#
```

---

# 3. Building Virtual Cluster: FermiCloud + Amazon EC2

## ■ Setup Configuration

```
[root@ip-10-111-45-200 etc]# diff condor_config condor_config.old
50,53d49
< CONDOR_HOST = fermicloud005.fnal.gov
< PRIVATE_NETWORK_NAME = amazon-ec2-us-east-1d
< TCP_FORWARDING_HOST = 50.16.165.161
< PRIVATE_NETWORK_INTERFACE = 10.111.45.200
68c64
< LOCAL_CONFIG_FILE = /opt/condor_7_7_0/local/condor_config.local
---
> LOCAL_CONFIG_FILE = /opt/condor_7_7_0/local.domU-12-31-39-0C-09-52/
condor_config.local
1045c1041
< UPDATE_COLLECTOR_WITH_TCP = True
---
> #UPDATE_COLLECTOR_WITH_TCP = True
1059,1060c1055,1056
< HIGHPORT = 40050
< LOWPORT = 40000
---
> #HIGHPORT = 9700
> #LOWPORT = 9600
[root@ip-10-111-45-200 etc]#
```

**CONDOR\_HOST =  
fermicloud005.fnal.gov**

**The following environment  
variable should be set with the  
values retrieved using “wget”**

**PRIVATE\_NETWORK\_NAME  
TCP\_FORWARDING\_HOST  
PRIVATE\_NETWORK\_INTERFA  
CE**

**HIGHPORT are LOWPORT are  
set to Amazon’s firewall port  
range**

# 3. Building Virtual Cluster: FermiCloud + Amazon EC2

## ■ Run Condor Work Node

```
[root@ip-10-111-45-200 local]# cd /  
[root@ip-10-111-45-200 /]#  
[root@ip-10-111-45-200 /]# source /opt/condor_7_7_0/condor.sh  
[root@ip-10-111-45-200 /]#  
[root@ip-10-111-45-200 /]# /opt/condor_7_7_0/sbin/condor_master  
[root@ip-10-111-45-200 /]#  
[root@ip-10-111-45-200 /]# ps -ef | grep condor_ | grep -v grep  
condor      1107      1  0  04:31 ?        00:00:00 /opt/condor_7_7_0/sbin/condor_master  
condor      1108    1107  0  04:31 ?        00:00:00 condor_startd -f  
[root@ip-10-111-45-200 /]#
```

**Condor Master  
Condor Executor**

## ■ Run Condor Work Node

```
[root@ip-10-111-45-200 /]# condor_status
```

Name	OpSys	Arch	State	Activity	LoadAv	Mem	ActvtyTime
slot1@fermicloud00	LINUX	X86_64	Owner	Idle	0.000	1974	0+00:15:04
slot2@fermicloud00	LINUX	X86_64	Owner	Idle	0.000	1974	0+00:15:05
slot1@ip-10-111-45	LINUX	X86_64	Owner	Idle	0.080	3736	0+00:00:04
slot2@ip-10-111-45	LINUX	X86_64	Owner	Idle	0.000	3736	0+00:00:05

```
Total Owner Claimed Unclaimed Matched Preempting Backfill  
X86_64/LINUX      4      4      0      0      0      0      0  
Total             4      4      0      0      0      0  
[root@ip-10-111-45-200 /]#
```

**EC2 Work Node Joined to  
Condor Central Pool**

**Not covered topics:**

- 1. How to create an AMI image which contains condor**
- 2. Condor job submission**

## 4. Building OSG Site at FermiCloud

### ■ Why it is valuable

- ➔ Education purpose: once OSG site has been built on physical machines, it is dangerous to play with it. So, no chance to play with it for those who joined a team later
- ➔ Even it is built in the virtual environment, it can serve for production with high flexibility which cannot be achieved by physical machines

### ■ List of Servers

Server	Description	Host
gatekeeper	OSG Gatekeeper	cloudgate.fnal.gov
condor	Central Manager	
gums	Grid User Management System	fermicloud014.fnal.gov
osg-client	OSG Job Submission UI	fermicloud009.fnal.gov
worker node	Condor Worker Node	fermicloud001.fnal.gov

# 4. Building OSG Site at FermiCloud

## ■ Create Proxy

---

```
[rsyoung@fermicloud009 ~]$ voms-proxy-init -voms cdf
Enter GRID pass phrase:
Your identity: /C=KR/O=KISTI/O=GRID/O=KISTI/CN=15861537 Seoyoung Roh
Creating temporary proxy ..... Done
Contacting voms-01.pd.infn.it:15001
[/C=IT/O=INFN/OU=Host/L=Padova/CN=voms-01.pd.infn.it] "cdf" Done
Creating proxy ..... Done
Your proxy is valid until Thu Jul 7 09:15:04 2011
[rsyoung@fermicloud009 ~]$
```

---

Installation of each server  
is not covered

## ■ Authentication Test

---

```
[rsyoung@fermicloud009 ~]$ globusrun -a -r cloudgate.fnal.gov

GRAM Authentication test successful
[rsyoung@fermicloud009 ~]$
```

---

## ■ Managedfork Test

---

```
[rsyoung@fermicloud009 ~]$ globus-job-run \
    cloudgate.fnal.gov:2119/jobmanager-fork /bin/date

Wed Jul 6 21:58:33 CDT 2011
[rsyoung@fermicloud009 ~]$
```

---

Prevent users from forking too  
many processes at gatekeeper

# 4. Building OSG Site at FermiCloud

## ■ Job Submission

```
[rsyoung@fermicloud009 01.globus-jobs]$ ./batch-job-submit.sh
 1 : https://cloudgate.fnal.gov:5357/21976/1310008164/
 2 : https://cloudgate.fnal.gov:5358/22044/1310008176/
 3 : https://cloudgate.fnal.gov:5359/22111/1310008189/
 4 : https://cloudgate.fnal.gov:5360/22197/1310008204/
 5 : https://cloudgate.fnal.gov:5362/22270/1310008220/
 6 : https://cloudgate.fnal.gov:5363/22347/1310008232/
 7 : https://cloudgate.fnal.gov:5364/22425/1310008246/
 8 : https://cloudgate.fnal.gov:5365/22499/1310008256/
 9 : https://cloudgate.fnal.gov:5367/22574/1310008270/
10 : https://cloudgate.fnal.gov:5368/22674/1310008288/
[rsyoung@fermicloud009 01.globus-jobs]$
```

**10 jobs are submitted to cloudgate.fnal.gov from osg-client (fermicloud009)**

**When a job is submitted, an URL is returned and it is used to check the status of the job later.**

## ■ Check the Condor Queue

```
[root@cloudgate ~]# condor_q
-- Submitter: cloudgate.fnal.gov : <131.225.154.205:9637> : cloudgate.fnal.gov
ID      OWNER      SUBMITTED  RUN_TIME  ST  PRI  SIZE  CMD
74.0    cdf         7/6 22:09   0+00:02:10 R  0   0.0  data
75.0    cdf         7/6 22:09   0+00:00:00 I  0   0.0  data
76.0    cdf         7/6 22:09   0+00:00:00 I  0   0.0  data
77.0    cdf         7/6 22:10   0+00:00:00 I  0   0.0  data
78.0    cdf         7/6 22:10   0+00:00:00 I  0   0.0  data
79.0    cdf         7/6 22:10   0+00:00:00 I  0   0.0  data
80.0    cdf         7/6 22:10   0+00:00:00 I  0   0.0  data
81.0    cdf         7/6 22:11   0+00:00:00 I  0   0.0  data
82.0    cdf         7/6 22:11   0+00:00:00 I  0   0.0  data
83.0    cdf         7/6 22:11   0+00:00:00 I  0   0.0  data

10 jobs; 9 idle, 1 running, 0 held
[root@cloudgate ~]#
```

**10 Jobs are in the queue**

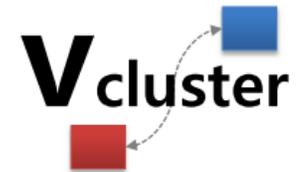
## 4. Building OSG Site at FermiCloud

### ■ Check Job Status

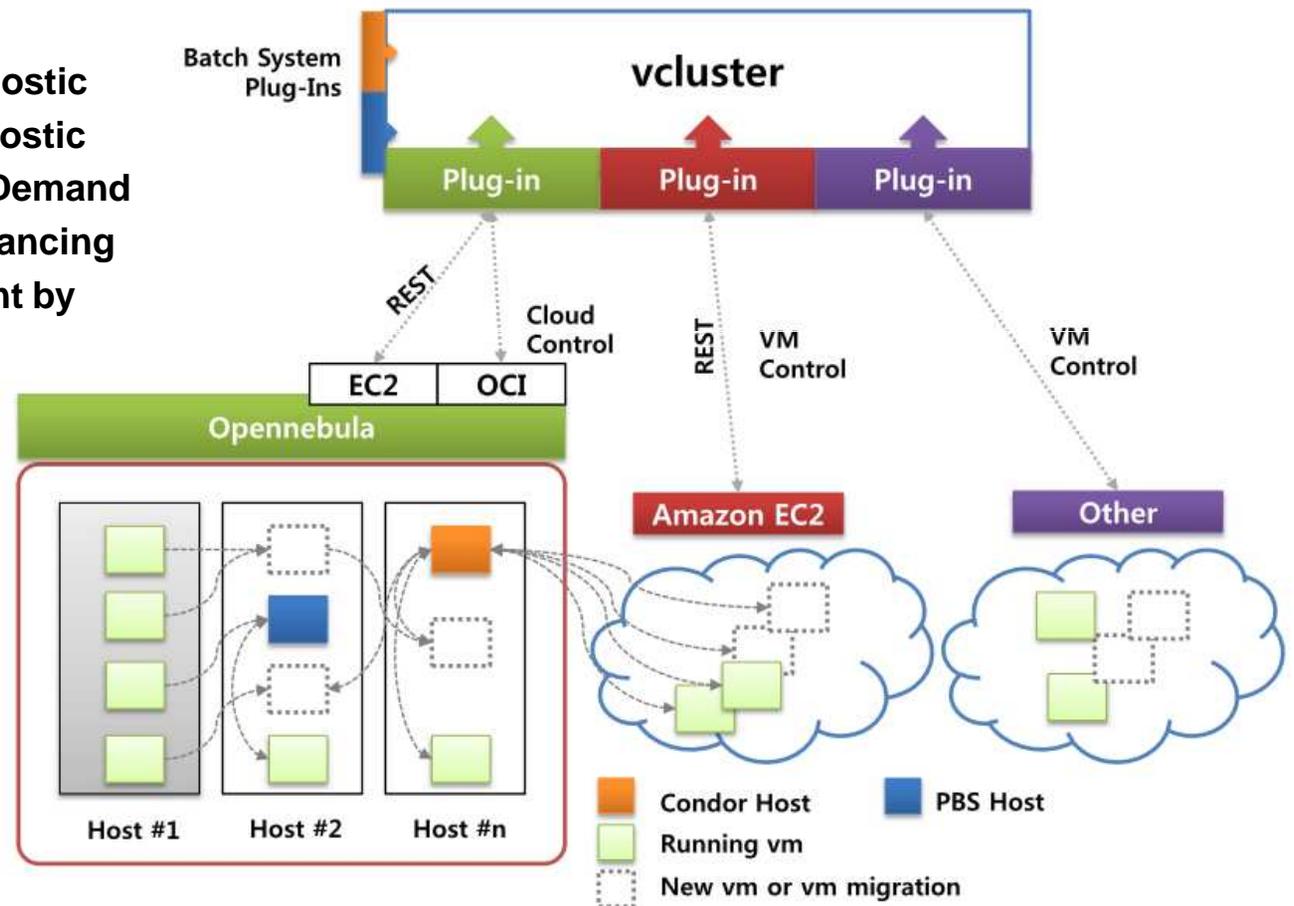
```
[rsyoung@fermicloud009 01.globus-jobs]$ ./getstatus.sh
1 https://cloudgate.fnal.gov:5357/21976/1310008164/ DONE
2 https://cloudgate.fnal.gov:5358/22044/1310008176/ DONE
3 https://cloudgate.fnal.gov:5359/22111/1310008189/ DONE
4 https://cloudgate.fnal.gov:5360/22197/1310008204/ DONE
5 https://cloudgate.fnal.gov:5362/22270/1310008220/ DONE
6 https://cloudgate.fnal.gov:5363/22347/1310008232/ DONE
7 https://cloudgate.fnal.gov:5364/22425/1310008246/ DONE
8 https://cloudgate.fnal.gov:5365/22499/1310008256/ DONE
9 https://cloudgate.fnal.gov:5367/22574/1310008270/ DONE
10 https://cloudgate.fnal.gov:5368/22674/1310008288/ DONE
```

**All jobs are done!**

# 5. vcluster: Automatic Scalable Virtual Cluster System

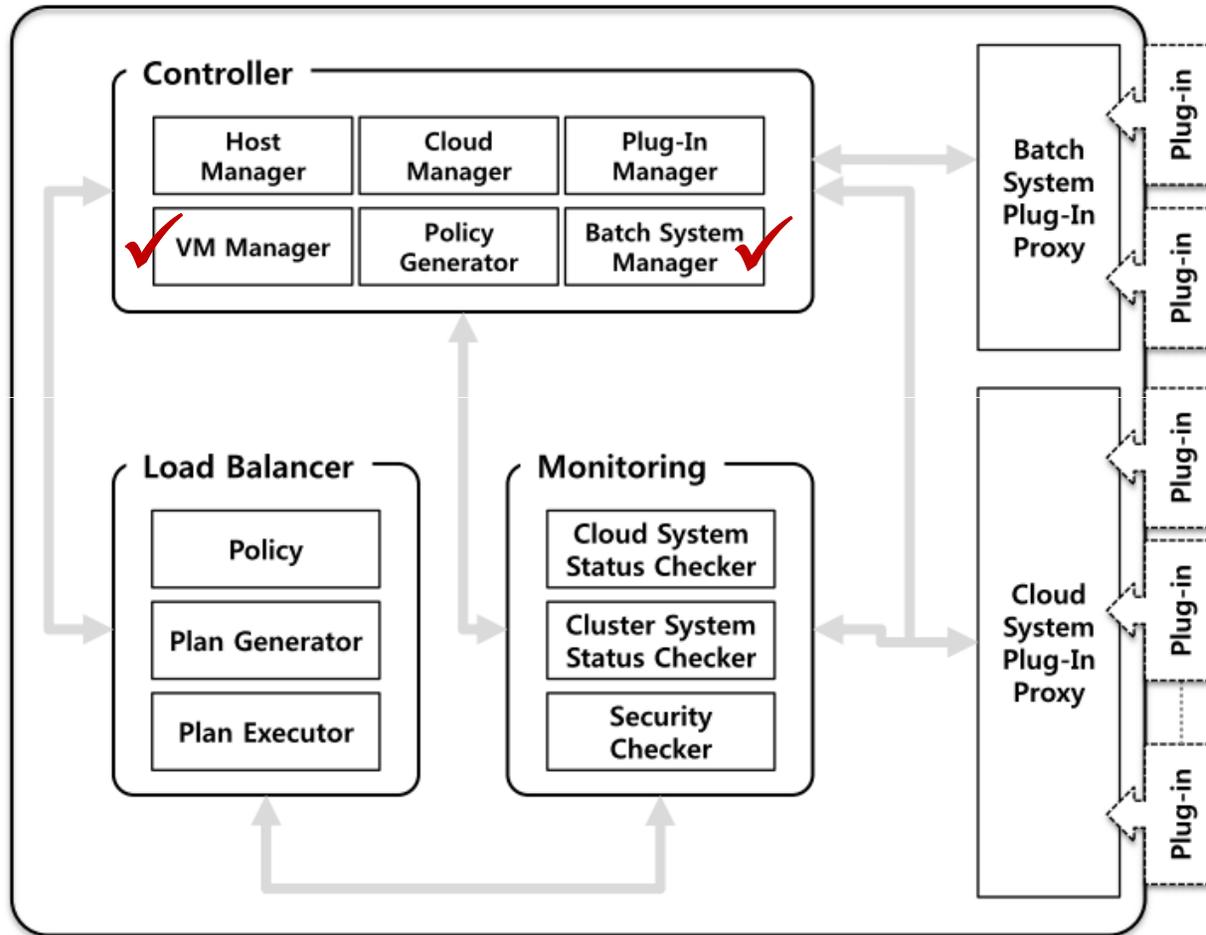


- Design Concept
  - ➔ Cloud System Agnostic
  - ➔ Batch System Agnostic
  - ➔ Auto Scalable on Demand
  - ➔ Dynamic Load Balancing
  - ➔ Power Management by Auto Migration



# 5. vcluster: Automatic Scalable Virtual Cluster System

## ■ Components



# 5. vcluster: Automatic Scalable Virtual Cluster System

- Status

## Demonstration

## 6. Conclusions

- With many supports from GCC Department, I achieved everything that I wanted before coming here.
  
- There are many things that I could not cover in this presentation
  - ➔ Virtual machine submission through Condor
  - ➔ Condor job submission
  - ➔ OSG site configuration
  - ➔ Creating Amazon AMI image
  - ➔ Etc
  
- G-Cloud system will be setup soon at GSDC
  - ➔ There are still many things that I have to get supports from engineers of GCC department
  
- Continue on developing my toy system – vcluster

**Thank You**