

HPC Cloud Prototype Based on Condor VM Universe

Condor Day 2010, Japan

Kunitaka Futatsugi

Kunitaka_Futatsugi@argo-graph.co.jp

ArgoGraphics Inc.

Background

- HPC
 - Multi Threading, MPI
 - Running on clusters
- Preparation for HPC
 - Checking of the Environment
 - Compilation
 - Execution
 - Test


Background

- Clusters in a Grid Environment
 - Each Cluster has its own settings
 - OS (version, kernel parameters)
 - Tools (compiler, libraries)
 - Users must repeat the preparation on each cluster



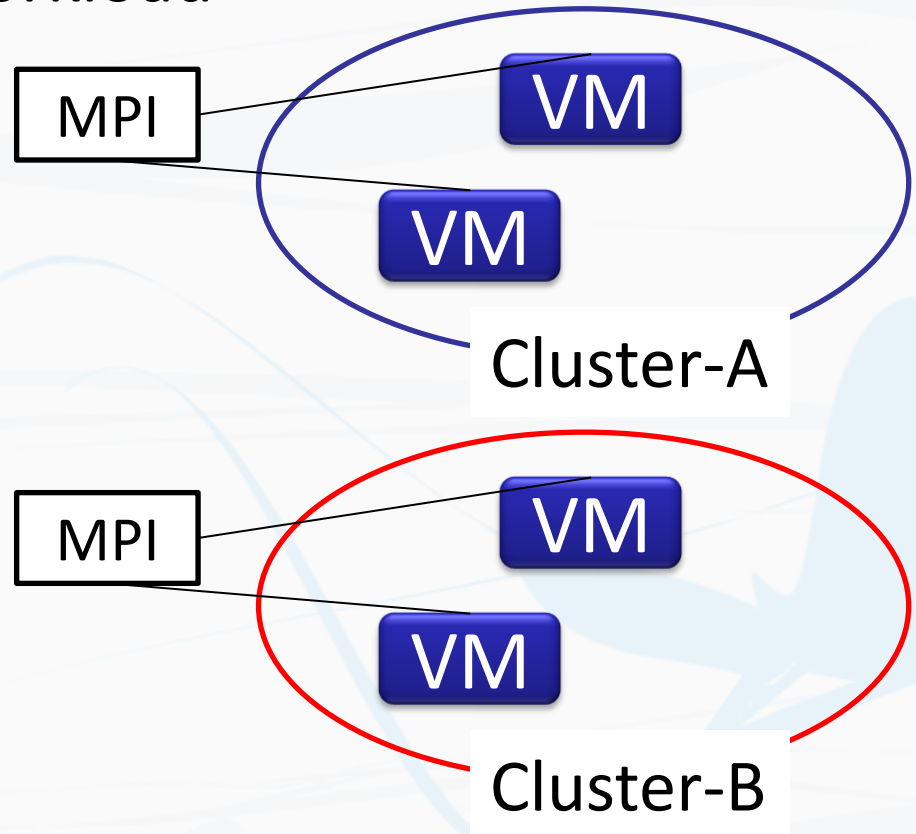
High Workload

Background

- Cloud computing
 - Provide on-demand resources and services over a network
 - computing instance  Virtual Machine
 - computing capacity
 - One VM can be shared by all cluster nodes which have the same VM infrastructure

Research Purpose

- Making HPC environment prototype with Cloud approach
 - Reduce Preparation workload



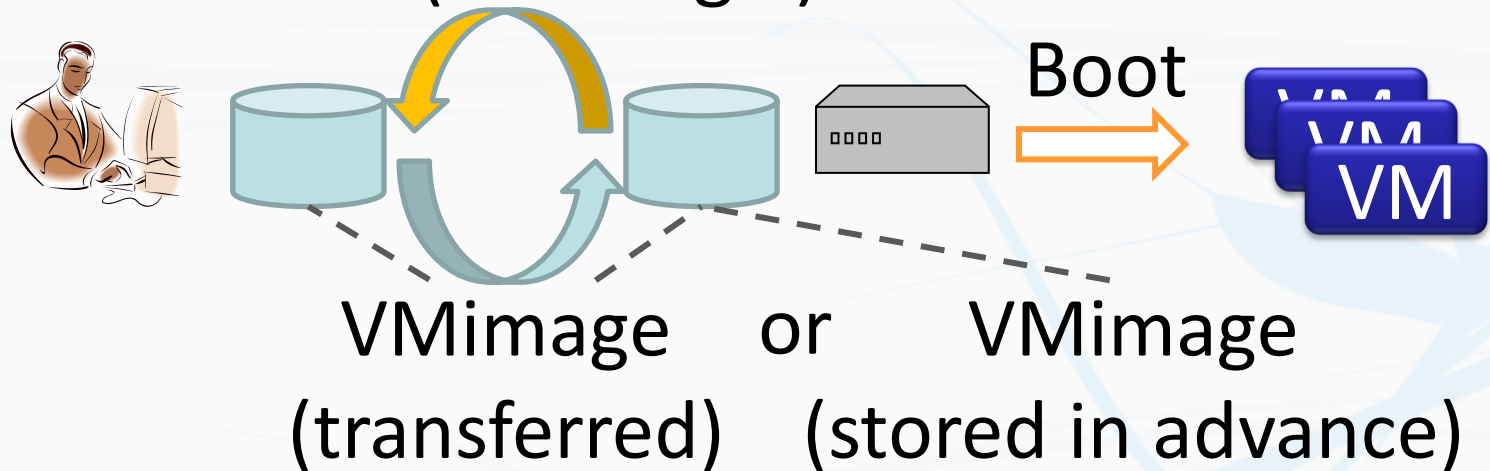
Contents

- Background
- Overview
- System
- Test
- Discussion
- Summary

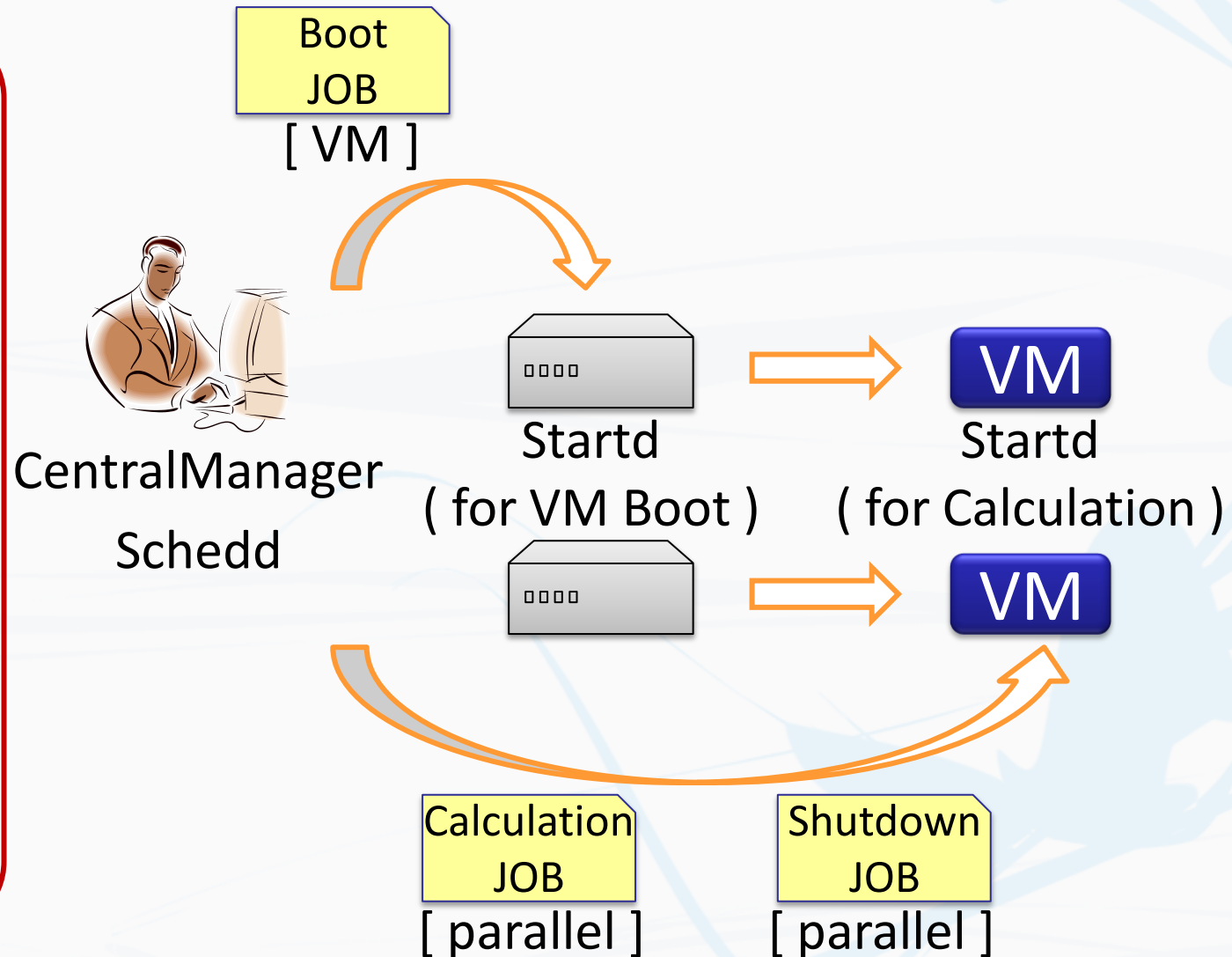
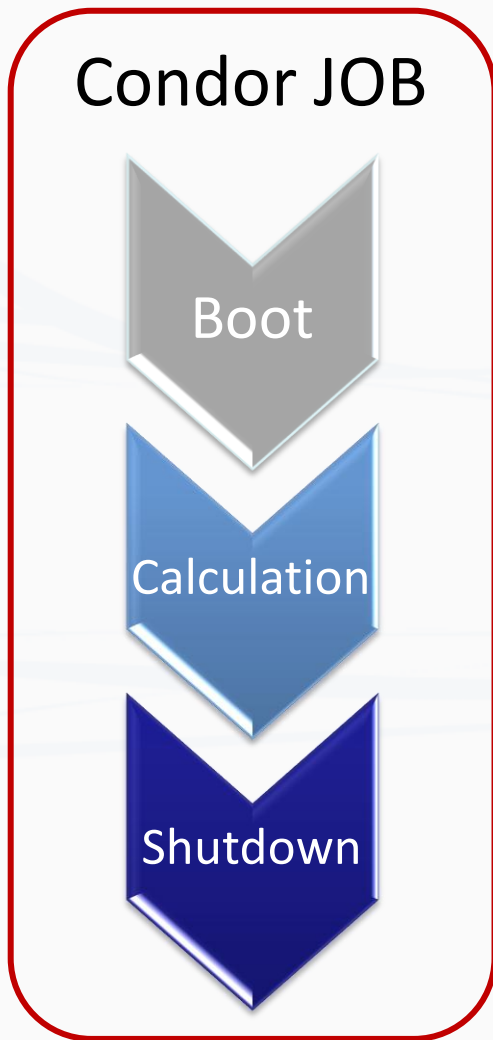
Condor VM Universe

- VM Job

- Starting Boot Up of the VM
- Running VM On
- Completion Shutdown of the VM
- Result Modified VM image (Optional)
 (VMimage')



System Overview



Boot Job (Xen)

- VM image
 - Preparing images for each node and slot
- Submit description file

```
universe          = vm
vm_type           = xen
xen_disk = /... /cent55_64.$$(Machine).slot$$(SlotID).img:xvda:w
...
```

Boot Job (VMware)

- VM image
 - Sharing one image file across a cluster
 - Creating snap shot images on each node
- Submit description file

universe	= vm
vm_type	= vmware
vmware_snapshot_disk	= TRUE
transfer_input_files	= <u>CentOS_5.4.vmx</u>
...	

includes the full path name of the shared image file

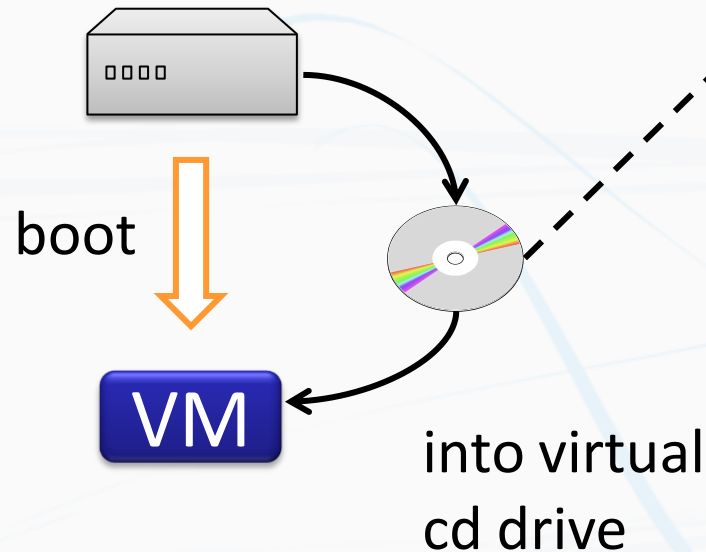
VM Initialization

- VM as Condor resource

- Name resolution (for parallel universe)
- Condor configuration

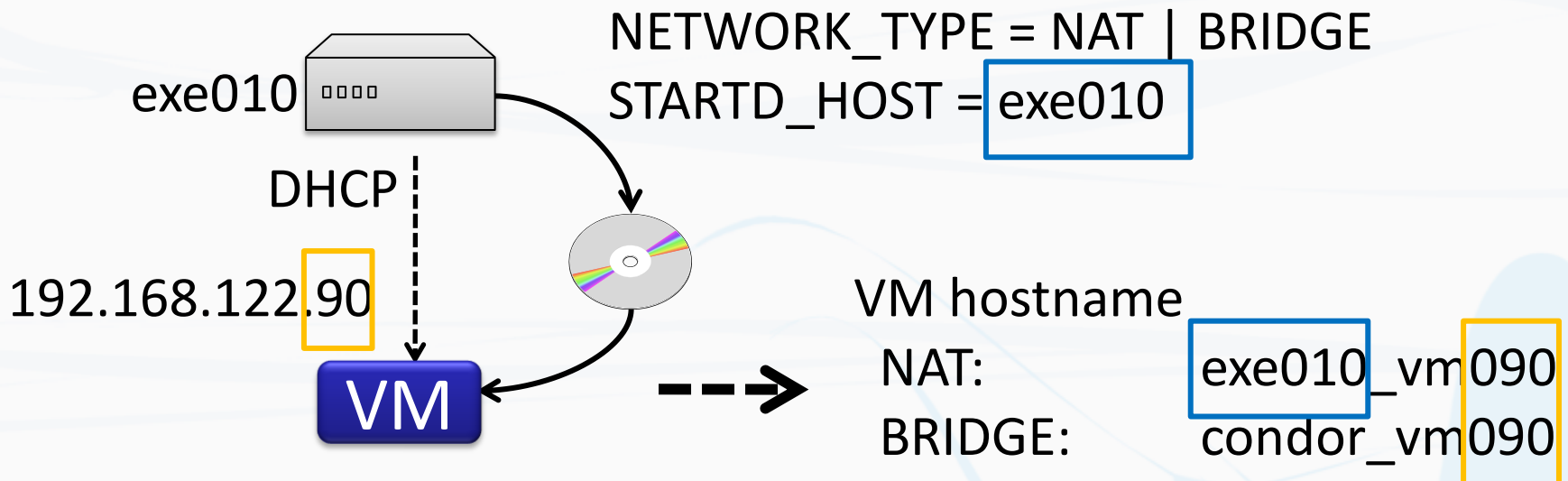
needs Cluster information

PREPARE JOB HOOK



VM Initialization (name resolution)

- Hostname
 - ***PREFIX_vm4thIPAddress***



VM Initialization (name resolution)

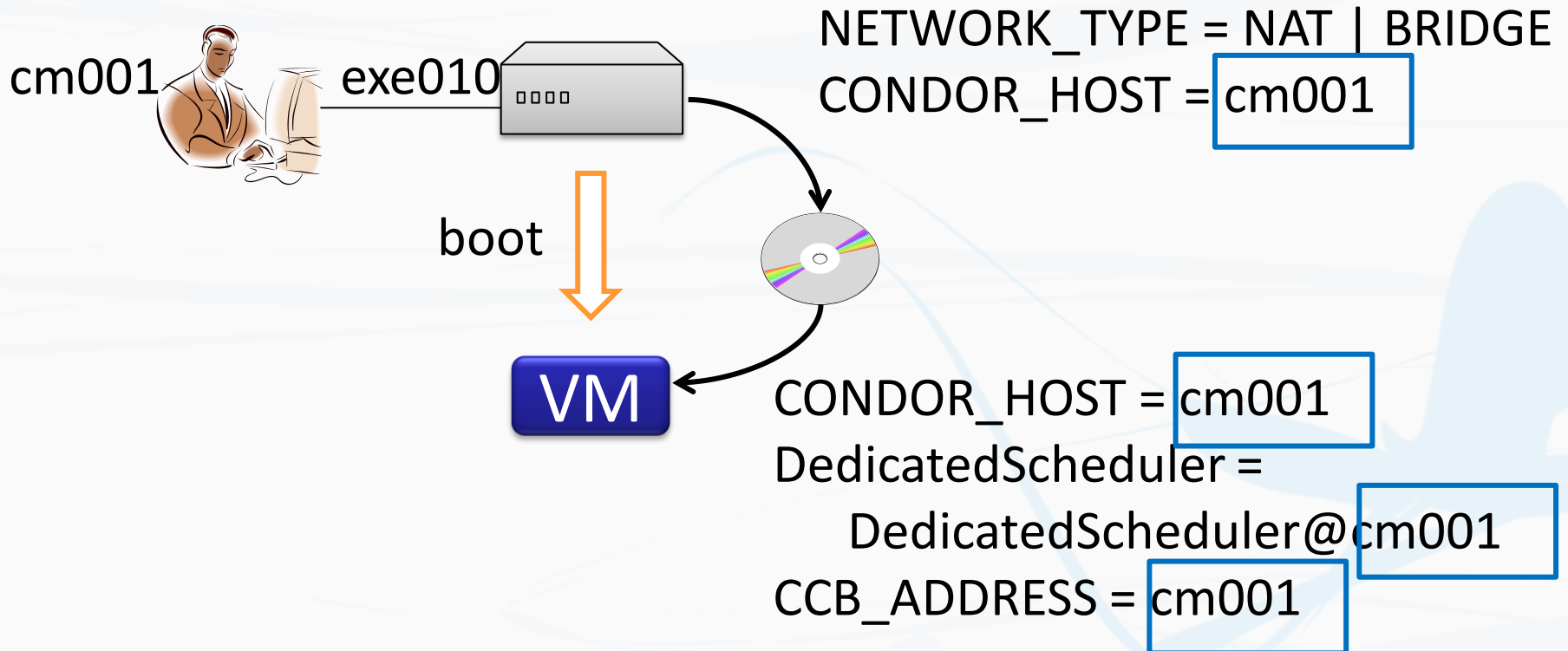
- /etc/hosts
 - networkaddress.x hostname.localdomain hostname
- example

```
192.168.122.1 exe101_vm001.localdomain exe101_vm001
...
192.168.122.254 exe101_vm254.localdomain exe101_vm254
```



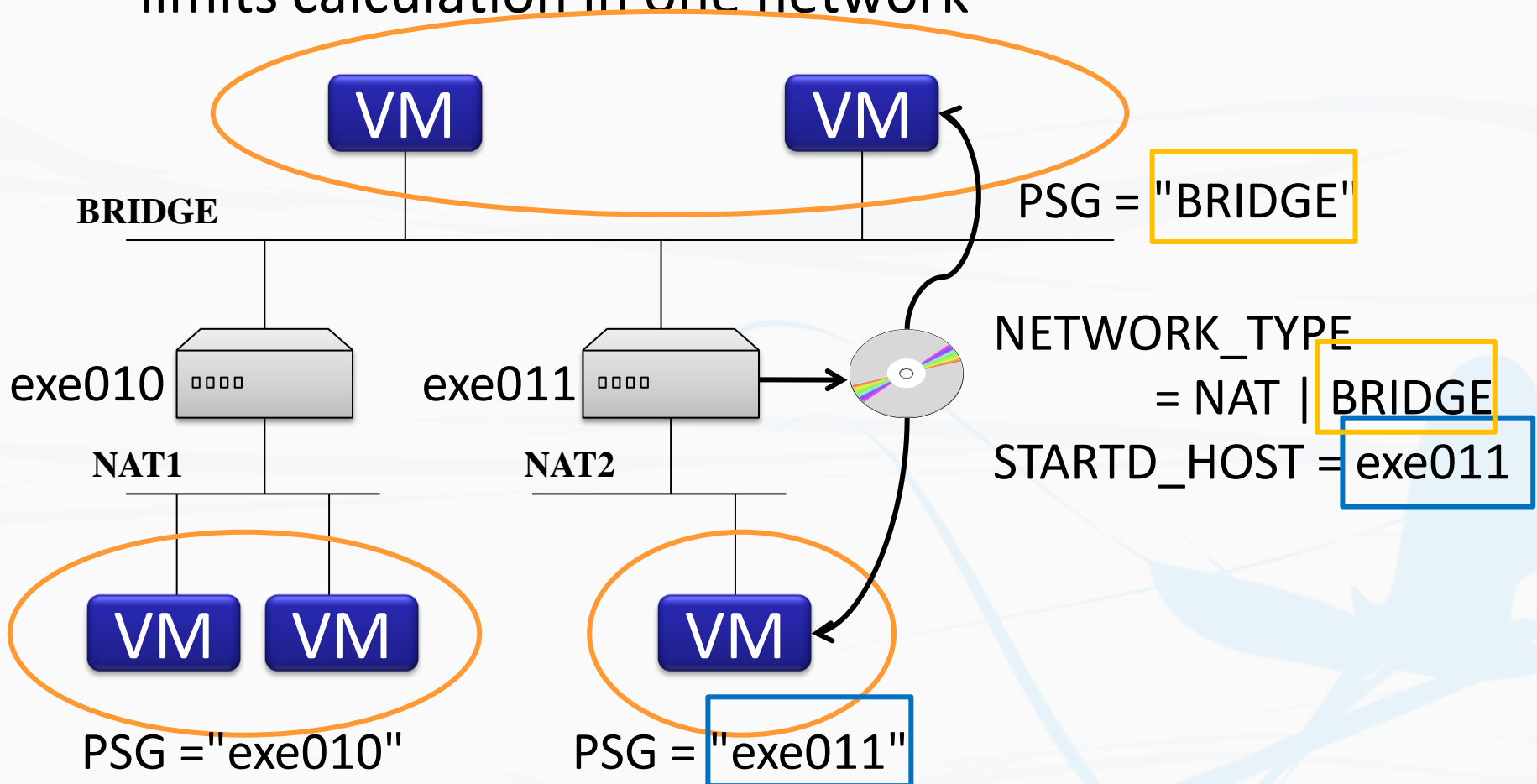
VM Initialization (Condor configuration)

- CONDOR_HOST
- DedicatedScheduler
- CCB_ADDRESS(only NAT)



VM Initialization (Condor configuration)

- ParallelSchedulingGroup (PSG)
 - limits calculation in one network



Additional Condor configuration in VM

- Static configuration (independent of Cluster)

```
VM_Pool          = TRUE
STARTD_ATTRS     = $(STARTD_ATTRS), VM_Pool
NUM_SLOTS_TYPE_1 = 1
SLOT_TYPE_1      = 100%
STARTER_ALLOW_RUNAS_OWNER = FALSE
SLOT1_USER       = condor
```

Calculation Job

- Submitting Parallel job to VM Pool
 - one or more jobs can be submitted
- Submit description file

```
universe           = parallel
machine_count      = 2      # number of VM
requirements       = VM_Pool == TRUE
+WantParallelSchedulingGroup = TRUE
...
```

Shutdown Job

- Submitting parallel job to the VM Pool
- Submit description file

```
universe           = parallel
machine_count      = 2           # number of VM
executable         = shutdown.sh
requirements       = VM_Pool == TRUE
+ParallelShutdownPolicy = "WAIT_FOR_ALL"
...
```

- shutdown.sh

```
#!/bin/sh
cd /opt/condor/libexec
at -f poweroff.cmd now + 1minutes _ _ _ _ _ sudo /usr/bin/poweroff
```

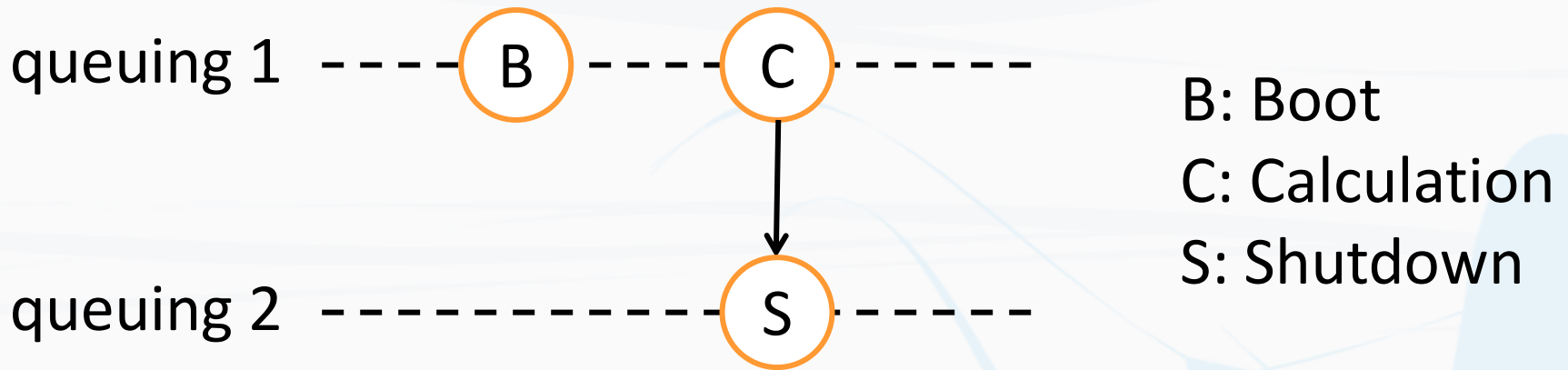
Grouping Jobs with DAGMan

- One group of jobs
 - Boot (VM universe)
 - Calculation (parallel universe)
 - Shutdown (vanilla universe)
- Condor DAGMan (Directed Acyclic Graph Manager)
 - Represents a set of Jobs
 - Manages dependencies between Jobs



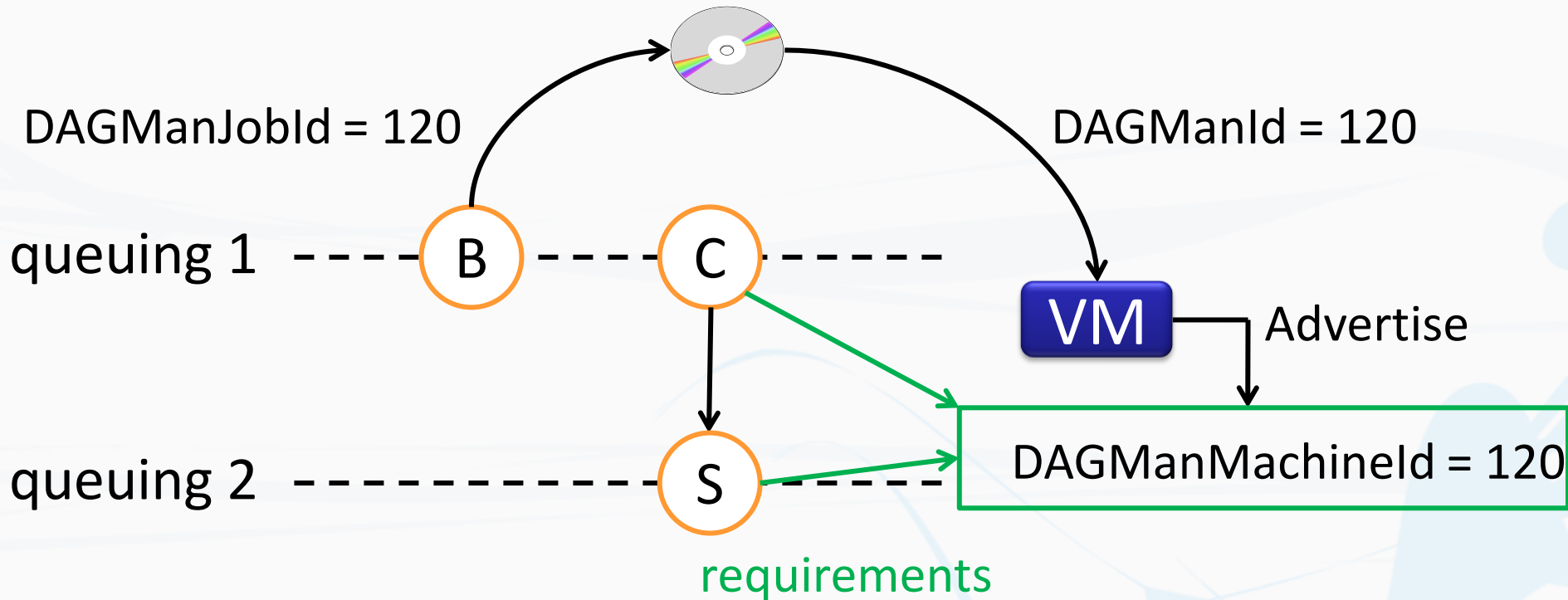
Grouping Jobs with DAGMan

- DAG for this group of jobs
 - The boot and calculation Jobs are submitted together
 - After the calculation, the shutdown Job is submitted



Grouping Jobs with DAGMan

- Relations in the group of jobs with DAGManJobId



Grouping Jobs with DAGMan

- DAG Input File
 - Control job with variable

```
JOB    B    job-vm
JOB    C    job-hpl
JOB    S    job-shut
VARS   B    slots="2" vos="cent55_64" net_type="bridge"
vmem="4096" vcpus="4"
VARS   C    slots="2" did="$(DAGManJobId)"
VARS   S    slots="2" did="$(DAGManJobId)"
PARENT          C    CHILD S
```

Grouping Jobs with DAGMan

- Submit description file (VM)

```
universe                = vm
vm_type                 = xen
vm_memory               = $(vmen)
vm_networking_type     = $(net_type)
xen_disk = /... /$(vos).$$ (Machine).slot$$ (SlotID).img:xvda:w
+JobVM_VCPUS           = $(vcpus)
...
queue $(slots)
```

Grouping Jobs with DAGMan

- Submit description file (Calculation)

```
universe           = parallel
machine_count      = $(slots)
executable         = hpl.sh
requirements       = VM_Pool == TRUE && DAGManMachineld
== $(did)
+WantParallelSchedulingGroup = TRUE
...
```

Grouping Jobs with DAGMan

- Submit description file (Shutdown)

```
universe                = parallel
machine_count           = $(slots)
executable               = shutdown.sh
requirements             = VM_Pool == TRUE && DAGManMachineld
== $(did)
+ParallelShutdownPolicy = "WAIT_FOR_ALL"
...
```

TEST

- Purpose
 - Evaluation of prototype
- Checking
 - Preparation workload
 - Overhead time
 - Performance

TEST

- Environment
 - Submit and Execution node : 1
 - Execution only node : 1
 - node spec
 - CPU : Xeon X5680 3.3GHz 2 cpus (total 12 cores)
 - MEMORY : 32GB
 - OS : Redhat 5.5
 - Condor 7.4.3
 - Xen 3.0.3
 - VM : CentOS5.5_64bit(paravirtualization)

TEST

- Calculation
 - High Performance Computing Linpack Benchmark
HPL 1.0a
 - MPI : OpenMPI 1.4
 - Compiler : GCC 4.1.2
 - BLAS library : GSL 1.13
 - 2 nodes, 24 processes MPI calculation

TEST

- DAGManFILE

```
JOB    B    job-vm
JOB    C    job-hpl
JOB    S    job-shut
VARS   B    slots="2" vos="cent55_64" net_type="bridge"
vmem="20480" vcpus="12"
VARS   C    slots="2" did="$(DAGManJobId)"
VARS   S    slots="2" did="$(DAGManJobId)"
PARENT C    CHILD S
```

VM (12 Cores 20GB Memory) × 2

TEST

- Observed behavior
 - Boot of 2 VMs
 - The calculation runs on 2 VMs
 - Shutdown of 2 VMs

- initial status

queue

ID	OWNER	SUBMITTED	RUN_TIME	ST	PRI	SIZE	CMD
----	-------	-----------	----------	----	-----	------	-----

node

Name	OpSys	Arch	State	Activity	LoadAv	Mem	ActvtyTime
slot1@hs2202	LINUX	X86_64	Unclaimed	Idle	0.220	32768	0+00:00:32
slot1@hs2203	LINUX	X86_64	Unclaimed	Idle	0.030	32768	0+00:00:04

TEST

- Observed behavior
 - Boot of 2 VMs
 - The calculation runs on 2 VMs
 - Shutdown of 2 VMs

queue

ID	OWNER	SUBMITTED	RUN_TIME	ST	PRI	SIZE	CMD
257.0	agse	11/2 11:45	0+00:00:32	R	0	7.3	condor_dagman
258.0	agse	11/2 11:45	0+00:00:12	R	0	0.0	cent55_64-bridge
258.1	agse	11/2 11:45	0+00:00:12	R	0	0.0	cent55_64-bridge
259.0	agse	11/2 11:45	0+00:00:00	I	0	0.0	hpl.sh

TEST

- Observed behavior
 - Boot of 2 VMs
 - The calculation runs on 2 VMs
 - Shutdown of 2 VMs

node

Name	OpSys	Arch	State	Activity	LoadAv	Mem	ActvtyTime
slot1@hs2202	LINUX	X86_64	Claimed	Busy	0.180	32768	00:00:30
slot1@hs2203	LINUX	X86_64	Claimed	Busy	0.060	32768	00:00:03

TEST

- Observed behavior
 - Boot of 2 VMs
 - The calculation runs on 2 VMs
 - Shutdown of 2 VMs

node

Name	OpSys	Arch	State	Activity	LoadAv	Mem	ActvtyTime
slot1@condor_vm181	LINUX	X86_64	Unclaimed	Idle	0.360	20480	00:00:04
slot1@condor_vm182	LINUX	X86_64	Unclaimed	Idle	0.310	20480	00:00:04
slot1@hs2202	LINUX	X86_64	Claimed	Busy	0.180	32768	00:00:30
slot1@hs2203	LINUX	X86_64	Claimed	Busy	0.060	32768	00:00:03

TEST

- Observed behavior
 - Boot of 2 VMs
 - The calculation runs on 2 VMs
 - Shutdown of 2 VMs

queue

ID	OWNER	SUBMITTED	RUN_TIME	ST	PRI	SIZE	CMD
257.0	agse	11/2 11:45	0+00:01:24	R	0	7.3	condor_dagman
258.0	agse	11/2 11:45	0+00:01:04	R	0	0.0	cent55_64-bridge
258.1	agse	11/2 11:45	0+00:01:04	R	0	0.0	cent55_64-bridge
259.0	agse	11/2 11:45	0+00:00:03	R	0	0.0	hpl.sh

TEST

- Observed behavior
 - Boot of 2 VMs
 - The calculation runs on 2 VMs
 - Shutdown of 2 VMs

node

Name	OpSys	Arch	State	Activity	LoadAv	Mem	ActvtyTime
slot1@condor_vm181	LINUX	X86_64	Claimed	Busy	0.270	20480	00:00:02
slot1@condor_vm182	LINUX	X86_64	Claimed	Busy	0.180	20480	00:00:03
slot1@hs2202	LINUX	X86_64	Claimed	Busy	0.180	32768	00:00:30
slot1@hs2203	LINUX	X86_64	Claimed	Busy	0.060	32768	00:00:03

TEST

- Observed behavior
 - Boot of 2 VMs
 - The calculation runs on 2 VMs
 - Shutdown of 2 VMs

node

Name	OpSys	Arch	State	Activity	LoadAv	Mem	ActvtyTime
slot1@condor_vm181	LINUX	X86_64	Unclaimed	Idle	0.360	20480	00:00:04
slot1@condor_vm182	LINUX	X86_64	Unclaimed	Idle	0.310	20480	00:00:04
slot1@hs2202	LINUX	X86_64	Claimed	Busy	0.180	32768	0+00:00:30
slot1@hs2203	LINUX	X86_64	Claimed	Busy	0.060	32768	0+00:00:03

TEST

- Observed behavior
 - Boot of 2 VMs
 - The calculation runs on 2 VMs
 - Shutdown of 2 VMs

queue

ID	OWNER	SUBMITTED	RUN_TIME	ST	PRI	SIZE	CMD
257.0	agse	11/2 11:45	0+00:04:14	R	0	7.3	condor_dagman
258.0	agse	11/2 11:45	0+00:03:54	R	0	0.0	cent55_64-bridge
258.1	agse	11/2 11:45	0+00:03:54	R	0	0.0	cent55_64-bridge
260.0	agse	11/2 11:49	0+00:00:00	I	0	0.0	shutdown.sh

TEST

- Observed behavior
 - Boot of 2 VMs
 - The calculation runs on 2 VMs
 - Shutdown of 2 VMs

node

Name	OpSys	Arch	State	Activity	LoadAv	Mem	ActvtyTime
slot1@hs2202	LINUX	X86_64	Claimed	Busy	0.180	32768	0+00:00:30
slot1@hs2203	LINUX	X86_64	Claimed	Busy	0.010	32768	0+00:03:45

TEST

- Observed behavior
 - Boot of 2 VMs
 - The calculation runs on 2 VMs
 - Shutdown of 2 VMs

queue

ID	OWNER	SUBMITTED	RUN_TIME	ST	PRI	SIZE	CMD
257.0	agse	11/2 11:45	0+00:05:17	R	0	7.3	condor_dagman
258.0	agse	11/2 11:45	0+00:04:57	R	0	0.0	cent55_64-bridge
258.1	agse	11/2 11:45	0+00:04:57	R	0	0.0	cent55_64-bridge

TEST

- Observed behavior
 - Boot of 2 VMs
 - The calculation runs on 2 VMs
 - Shutdown of 2 VMs

queue

ID	OWNER	SUBMITTED	RUN_TIME	ST	PRI	SIZE	CMD
257.0	agse	11/2 11:45	0+00:06:22	R	0	7.3	condor_dagman

TEST

- Observed behavior
 - Boot of 2 VMs
 - The calculation runs on 2 VMs
 - Shutdown of 2 VMs

queue

ID	OWNER	SUBMITTED	RUN_TIME	ST	PRI	SIZE	CMD
----	-------	-----------	----------	----	-----	------	-----

node

Name	OpSys	Arch	State	Activity	LoadAv	Mem	ActvtyTime
slot1@hs2202	LINUX	X86_64	Unclaimed	Idle	0.220	32768	0+00:00:32
slot1@hs2203	LINUX	X86_64	Unclaimed	Idle	0.030	32768	0+00:00:04

TEST

- Calculation Result

```
[agse@hs2202 dag-job-1]$ cat out.259
```

```
...
```

```
=====
```

T/V	N	NB	P	Q	Time	Gflops
WR00L2L2	10000	256	6	4	149.13	4.471e+00

```
-----
```

```
||Ax-b||_oo / ( eps * ||A||_1 * N ) = 0.0862480 ..... PASSED  
||Ax-b||_oo / ( eps * ||A||_1 * ||x||_1 ) = 0.0204089 ..... PASSED  
||Ax-b||_oo / ( eps * ||A||_oo * ||x||_oo ) = 0.0045618 ..... PASSED
```

```
=====
```

```
...
```

```
[agse@hs2202 dag-job-1]$
```

TEST

- Calculation Result without VM
 - execute hpl manually with mpiexec command
 - same conditions except for not using VM

```
=====
T/V          N  NB  P  Q          Time          Gflops
-----
WR00L2L2    10000 256 6 4          145.60          4.580e+00
-----
||Ax-b||_oo / ( eps * ||A||_1 * N ) =    0.0862480 ..... PASSED
||Ax-b||_oo / ( eps * ||A||_1 * ||x||_1 ) =    0.0204089 ..... PASSED
||Ax-b||_oo / ( eps * ||A||_oo * ||x||_oo ) =    0.0045618 ..... PASSED
=====
```

TEST RESULT

- Preparation workload
 - creation image : 0.5 day
- Overhead time
 - boot and Job assign: 1 min 20 sec
 - shutdown and DAGMan completion : 2 min 28 sec
- Performance
 - no large difference between VM and non VM environment (not enough case)

Discussion

- Preparation Workload for Cluster usage
 - User
 - creating VM image (1 / VM environment) **heavy**
 - copying VM to Cluster **light**
 - Xen (nodes / Cluster)
 - VMware(1 / Cluster)
 - making Job related file (SDF, scripts)
 - Administrator (Cluster side)
 - Installation and setting (VM environment, Condor)
 - DHCP, DNS (only BRIDGE)

Discussion

- VM is independent of cluster
 - Dynamic initialization
 - Name resolution
 - Condor configuration
 - Cluster specific information can be retrieved from VM boot node via cd-rom



High Reusability



reduce **heavy** preparation load
(only **light**)

Discussion

- Usability
 - One submit command to execute calculation
 - DAGMan Input
 - Submit Description File
 - Low overhead time
 - no image transfer
- VM Maintainability
 - Once VM modified all VM placed on Cluster must be updated

Discussion

- Performance
 - Additional test is required

Summary

- Created HPC cloud Prototype
- Calculations run on VMs
- VMs can be reused on other clusters
 - reduce preparation load
- A calculation sequence (Boot, Execution, Shutdown) is achieved by one submit command
- VM delivery mechanism must be considered in the future

Acknowledgement

- This work is supported by Collaborative Research of AIST and ArgoGraphics.