Deployment of Virtual Clusters for Molecular Docking Experiments on the PRAGMA Cloud

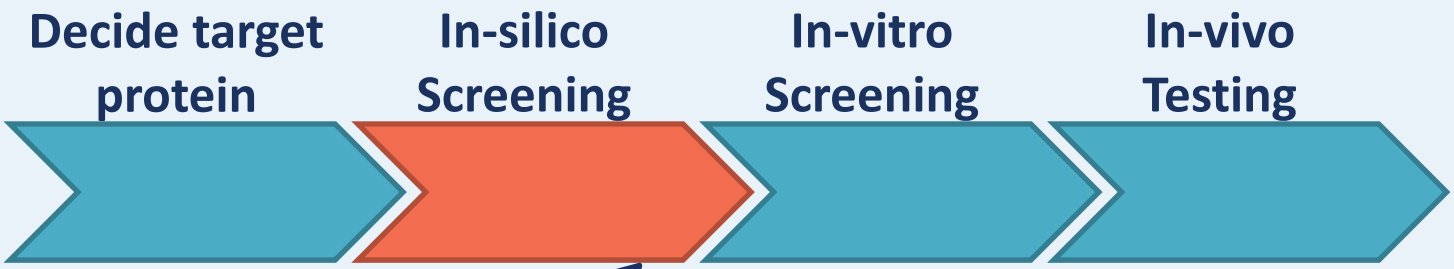
Kohei Ichikawa (NAIST/Osaka U),
Kevin Lam, (Prime2013)
Karen Rodriguez, (Prime2013)
Wen-Wai Yim, (Prime2009)
Jason Haga (UCSD)

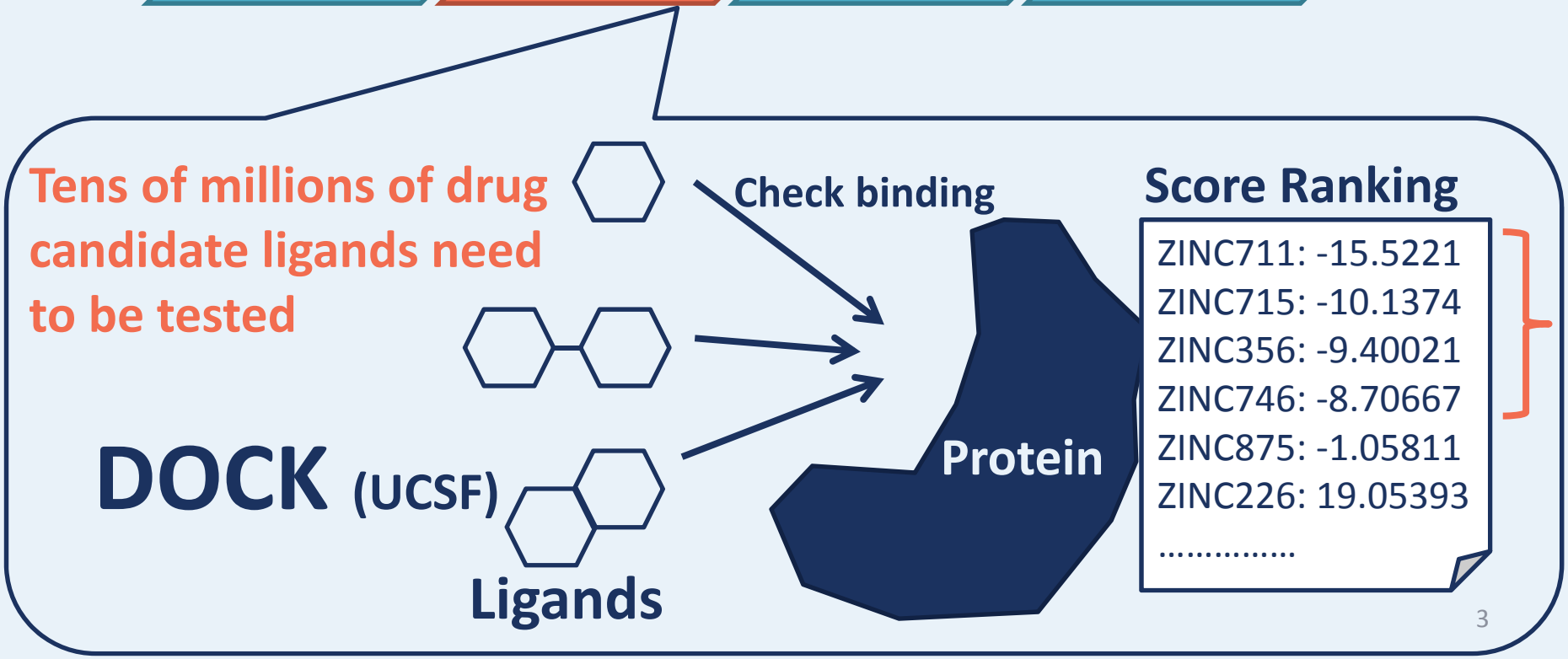
Objective

- Create virtual clusters for use in molecular docking experiments
  - Consistent reproducible, environment for docking studies
  - Simplify the learning curve, scripting, and data management
  - Future potential clinical treatment for disease

Background: Docking Simulation

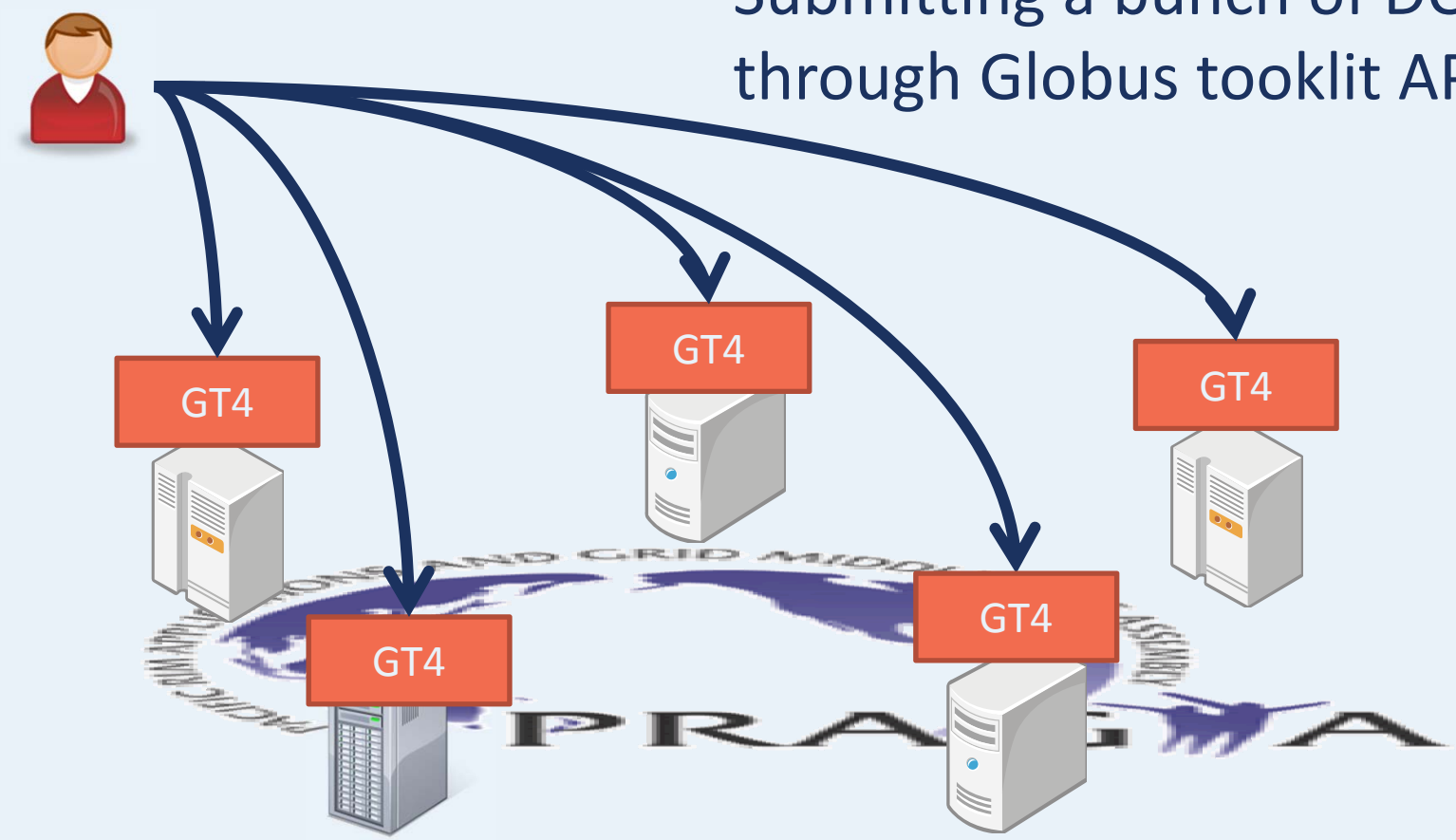
Drug discovery workflow



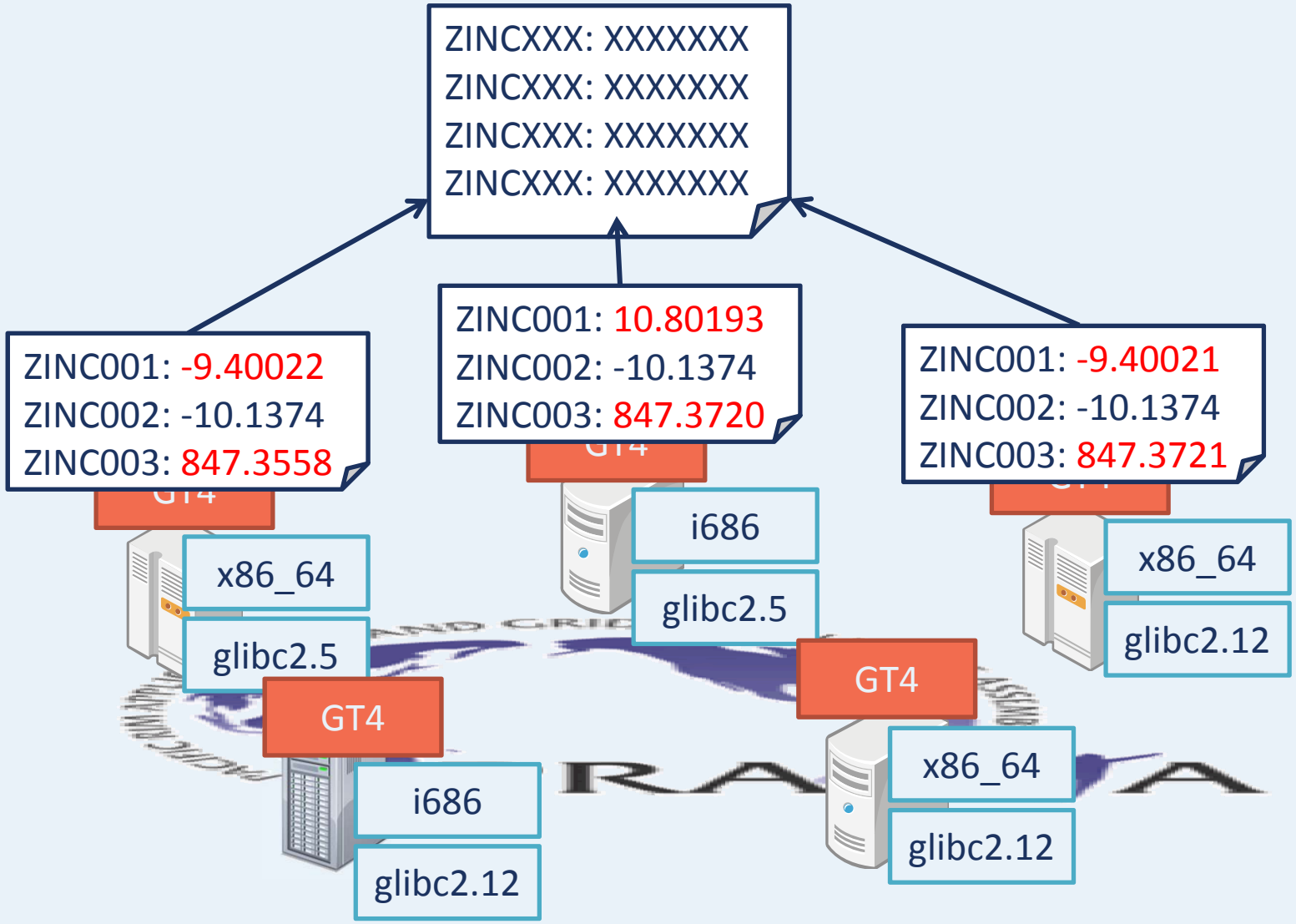


Grid computing environment

Submitting a bunch of DOCK jobs through Globus tooklit API



Problem: Heterogeneous Grid platforms yield inconsistent results

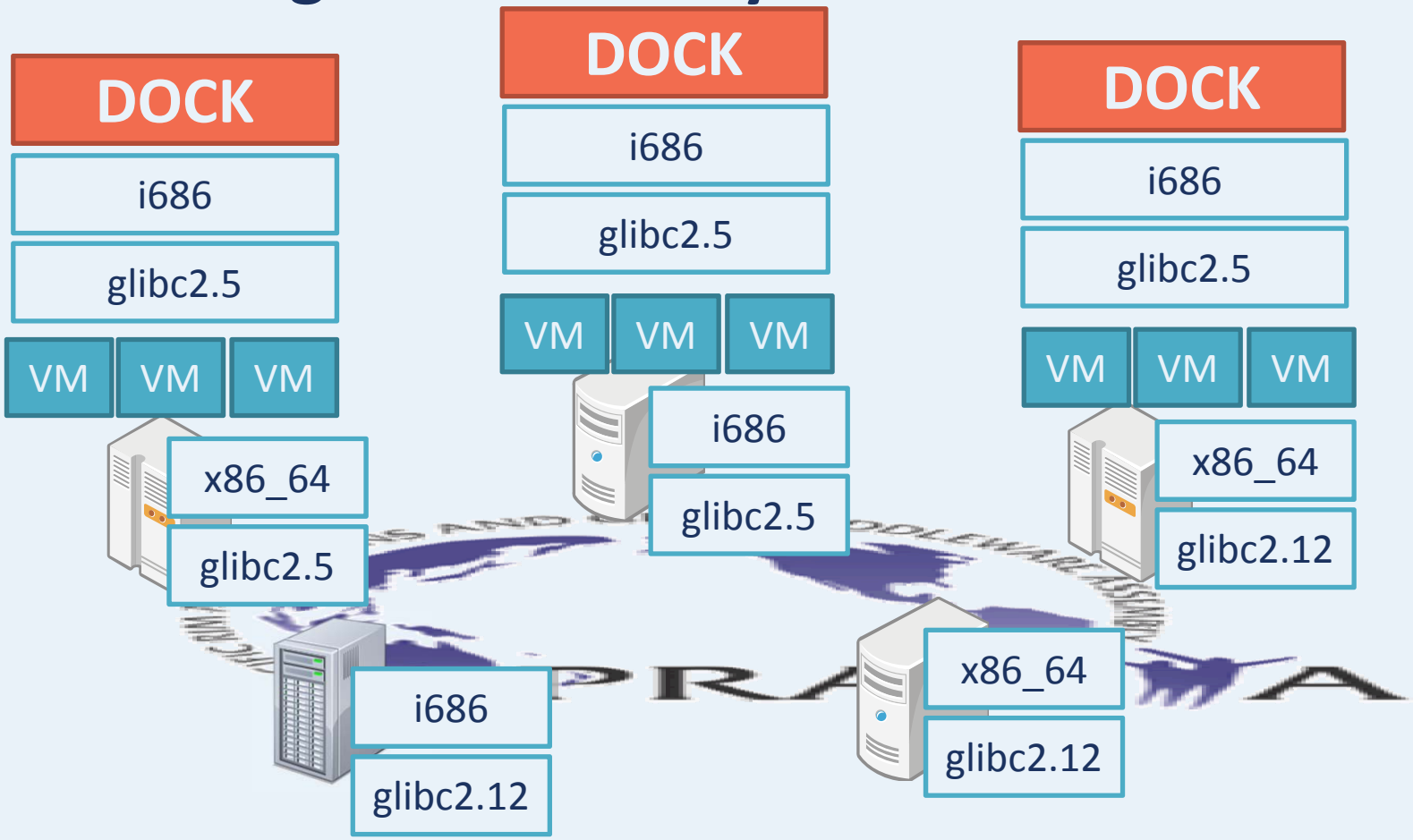


Score comparison between UCSF DOCK Developer Results and Grid results

	manufacturer's	*nacona*	*komolongma*	*ocikbpra*	*aurora*	*cafe01*	*tea01*
OS-bit		64	32	32	32	32	32
machine bit		64	32	32	32	32	32
gcc version		3.4.6	4.1.2	3.4.6	3.4.6	3.4.6	3.4.6
ID	Scores						
ZINC00013564	**10.801939**	-9.40022	10.801939	-9.40021	-9.40021	-9.40021	-9.40021
ZINC00150863	**21.139143**	-12.4672	21.139143	21.13914	21.13914	21.13914	21.13914
ZINC00152265	**30.238361**	30.23803	30.238361	19.05393	19.05393	19.05393	19.05393
ZINC00157111	**-12.916615**	-12.9166	-12.916615	-15.5221	-15.5221	-15.5221	-15.5221
ZINC00157152	**-10.137384**	-10.1374	-10.137384	-10.1374	-10.1374	-10.1374	-10.1374
ZINC00157402	**168513.625**	168506.4	168513.625	145519.8	145519.8	145519.8	145519.8
ZINC00157467	**-8.706671**	-8.70678	-8.706671	-8.70667	-8.70667	-8.70667	-8.70667
ZINC00157960	**847.37207**	847.3558	847.37207	847.3721	847.3721	847.3721	847.3721
ZINC00158442	**52.56588**	52.56579	52.56588	52.56588	52.56588	52.56588	52.56588
ZINC00158751	**-1.058107**	-1.05816	-1.058107	-1.05811	-1.05811	-1.05811	-1.05811
ZINC01555236	**503.249725**	5.02E+08	503.249725				

Solution: Using virtual machine as a layer of abstraction

Wen-wai Yim (Prime2009) has been working on testing the feasibility.



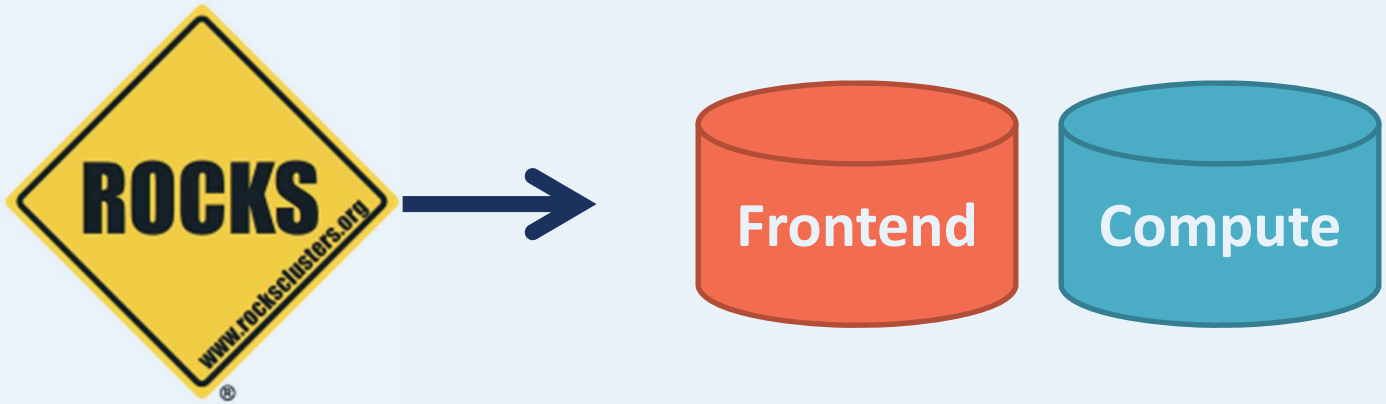
Determined the best configuration for DOCK

Kevin Lam and Karen Rodriguez (Prime2013) have been working on preparing VC images.

VM Tests							
Machine bit		32	64	64	64		64
OS bit		32	32	64	64		64
CentOS		5.9	5.2	5.9	6.4		6.4
gcc		4.1.2	4.1.2	4.1.2	4.7.7		4.4.7
Dock Version		6.2	6.2	6.2	6.2		6.6
	Developer's 6.2	Masternode Slavenode1	Master	Sailboat Kayak	CentOS1	Developer's 6.6	CentOS5
ZINC00158751	138.113892	138.11386	138.113861	138.114029	138.114029	-13.890112	-13.890112
ZINC00157960	21535.20898	21535.209	21535.20898	21535.30859	21535.30859	-17.022007	-17.022007
ZINC00158442	52.565872	52.56588	52.56588	52.565788	52.565788	-0.554128	-0.554128
ZINC00013564	-9.400209	10.801939	10.801939	-9.400218	-9.400218	-15.287382	-15.287382
ZINC01555236	-	503.24973	503.249725	1800544512	1800544512	-	-
ZINC00150863	21.139147	21.139143	21.139143	-12.467216	-12.467216	-13.396927	-13.396927
ZINC00152265	19.053925	30.238361	30.238361	30.238028	30.238028	-15.236704	-15.236704
ZINC00157111	-15.522114	-12.916615	-12.916615	-12.916644	-12.916644	-14.073079	-14.073079
ZINC00157152	-10.13739	-10.137384	-10.137384	-10.137392	-10.137392	-15.55584	-15.55584
ZINC00157402	168513.7031	168513.63	168513.625	168506.4063	168506.4063	-15.105305	-15.105305
ZINC00157467	-8.706671	-8.706671	-8.706671	-8.706783	-8.706783	-14.412565	-14.412565

Creating virtual cluster images

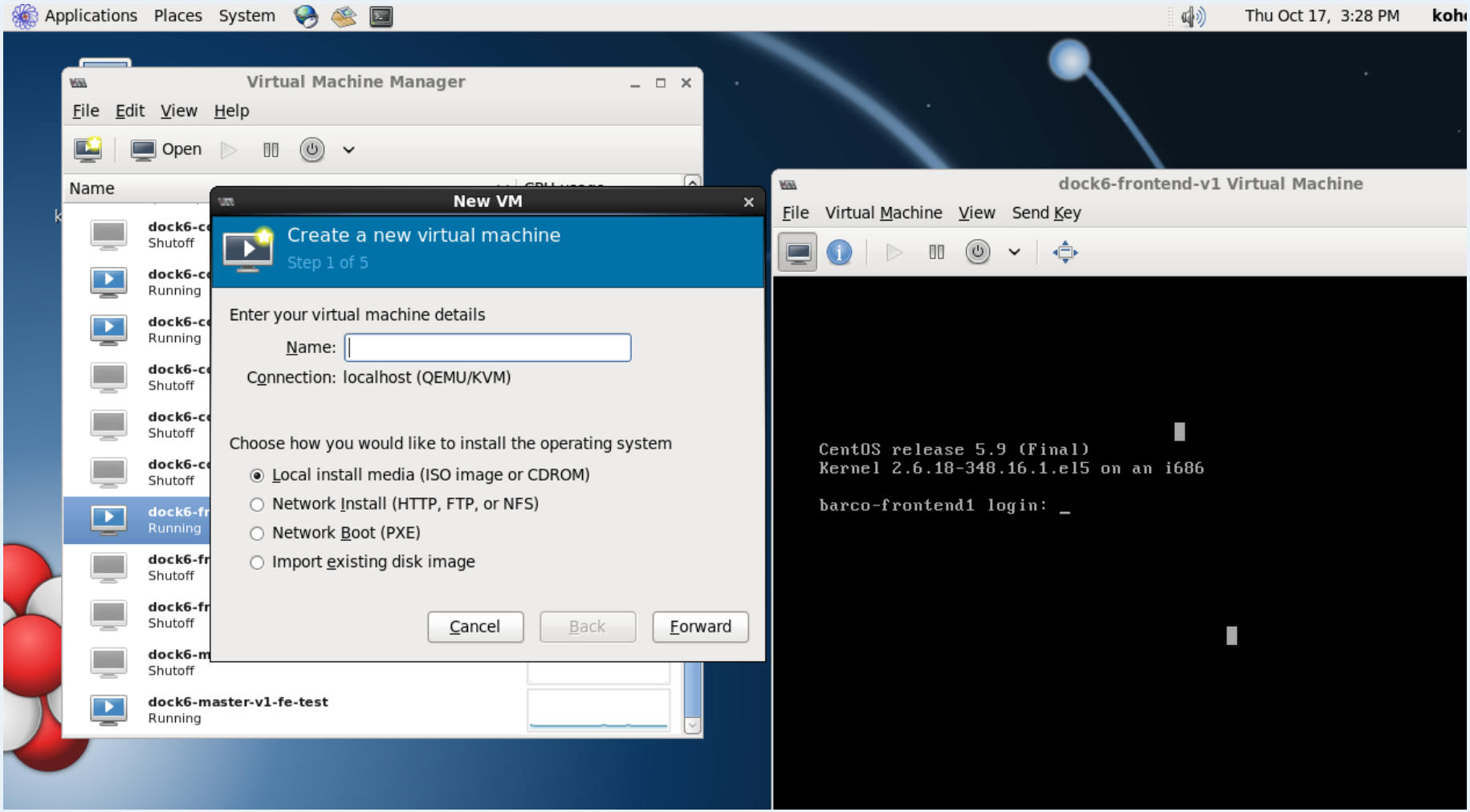
- Using Rocks:



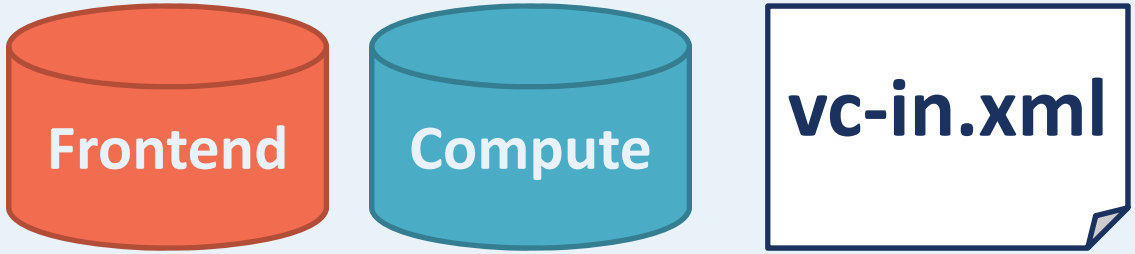
  - Easy to create a template of VC images
  - Not easy to customize OS architecture/libraries

- Creating from scratch
  - Using KVM & libvirt on Linux
  - Install OSs on the VMs manually

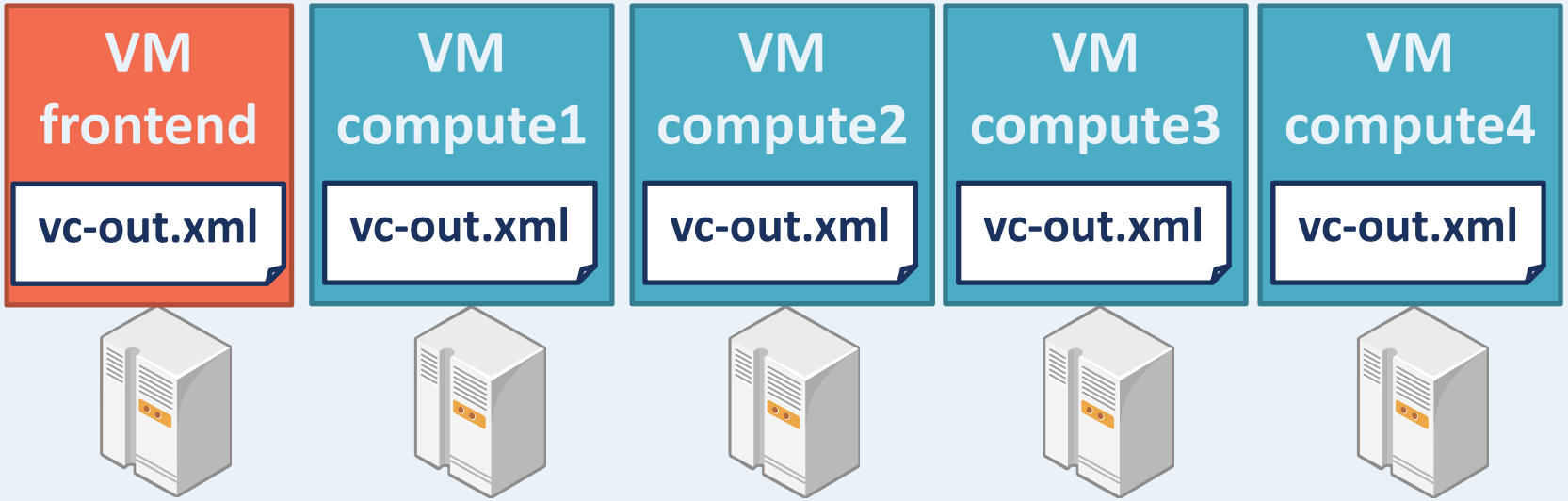
Creating virtual cluster images from scratch



Deployment of VC using pragma_boot




```
# pragma_boot  –vcname  dock6  –num_cpus  4
```



Preparing vc-in.xml

```
<vc version='0.1'>
 <virtualization engine='kvm' type='hvm'
arch='x86_64'/>
 <driver>rocks6_client</driver>
 <frontend>

 </frontend>
 <compute>
  <boot_dependency parent='frontend'>
   <wait type='clock' value='300'/>
  </boot_dependency>

 </compute>
 <networks>
  <network name='private'>
   <ipaddress>10.1.1.1</ipaddress>
   <netmask>255.255.0.0</netmask>
  </network>
  <frontend>
    <public>eth1</public>
  </frontend>
 </networks>
</vc>
```

Copy & Paste

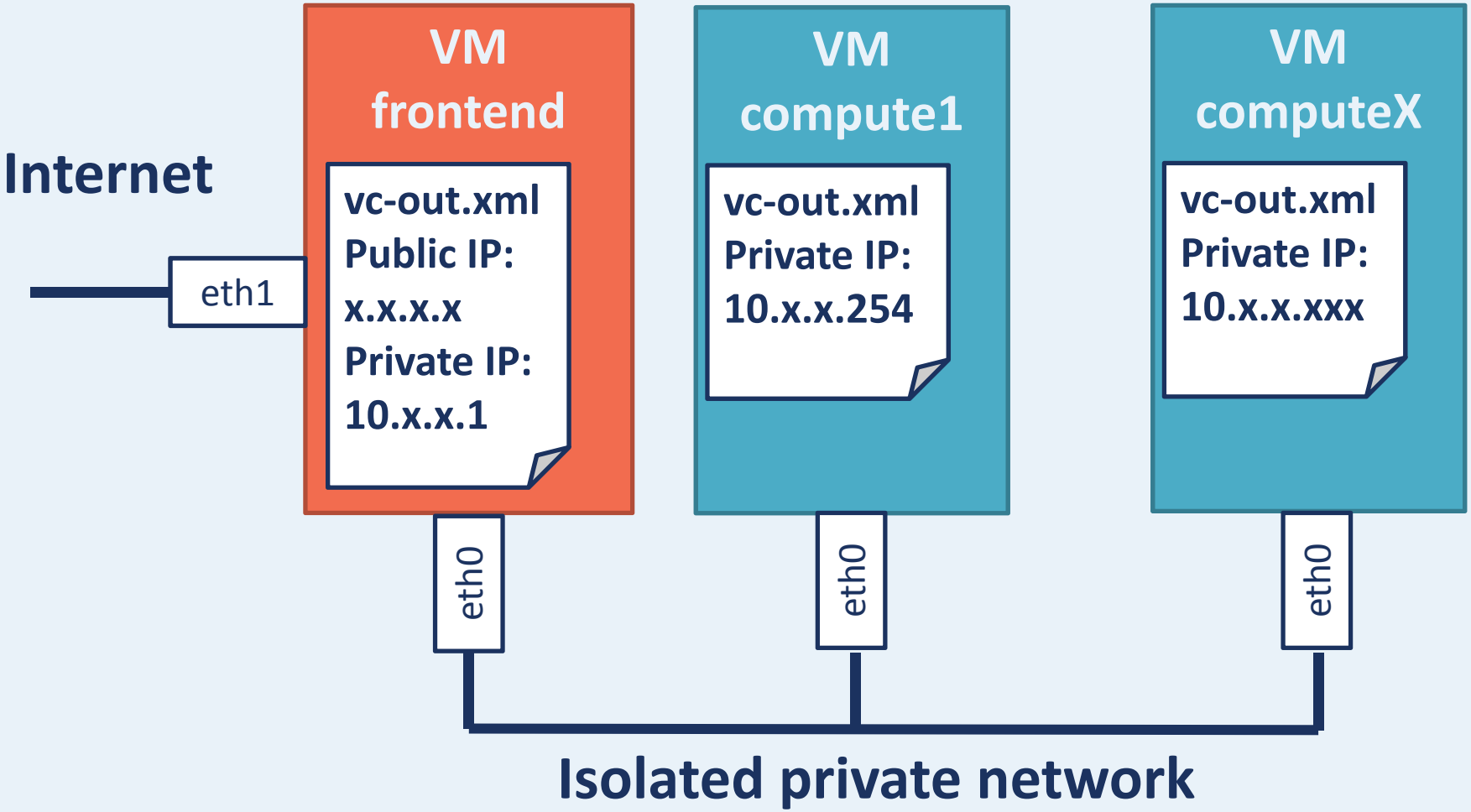
/etc/libvirt/qemu/dock6-frontend.xml

```
<domain type='kvm'>
 <name>
  dock6-frontend
 </name>
……
```

/etc/libvirt/qemu/dock6-compute.xml

```
<domain type='kvm'>
 <name>
  dock6-compute
 </name>
……
```

Parsing vc-out.xml & configuring network

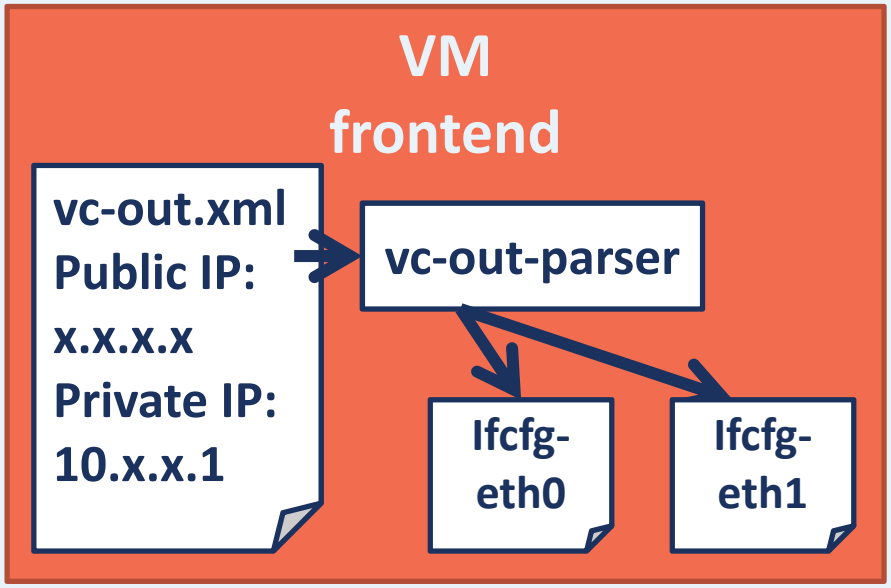


vc-out parser

- https://github.com/pragmagrid/vc-out-parser

Developed by Luca

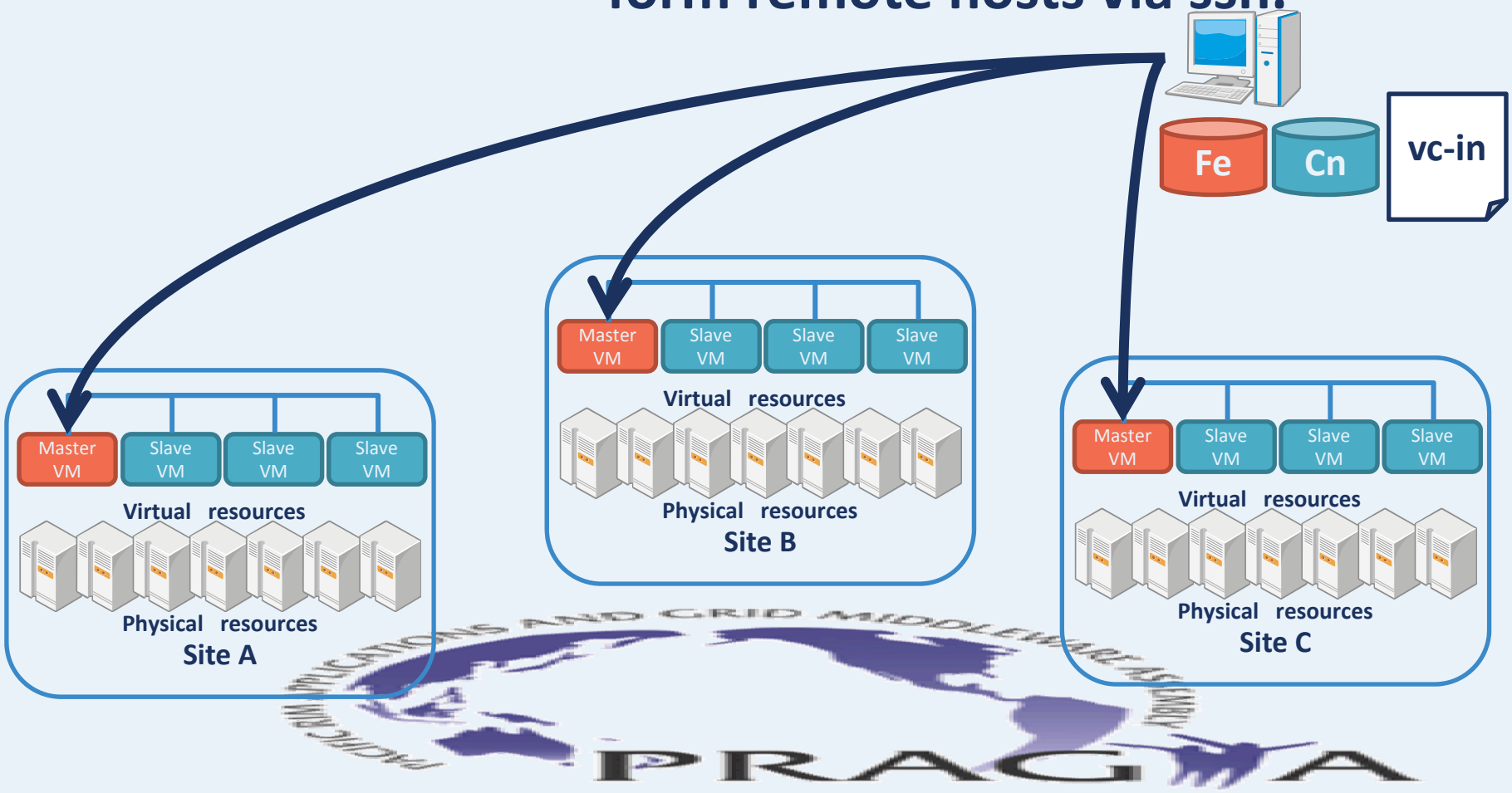
  - Subset of configuration tools for Rocks VC (dynip)
  - Generate basic network configuration for Redhat based Linux by parsing vc-out.xml



vc-out-paser generates network configurations in an early stage of the booting of the VM

Virtual clusters on Pragma Cloud

Virtual clusters can be accessible form remote hosts via ssh.



Demo

Booting DOCK VC

```
# pragma_boot  –vcname  dock6  –num_cpus  8
```

OpenNebula Sunstone

Virtual Machines 9 TOTAL 9 ACTIVE 0 OFF 0 PENDING 0 FAILED oneadmin

Create Search

- Dashboard
- System
- Virtual Resources
  - Virtual Machines
  - Templates
  - Images
  - Files & Kernels
- Infrastructure
- Marketplace
- OneFlow

ID	Owner	Group	Name	Status	Host	IPs	VNC
123	oneadmin	oneadmin	frontend_1_(service_92)	RUNNING	asccmp050	10.2.0.1 163.220.57.212	
124	oneadmin	oneadmin	compute_0_(service_92)	RUNNING	asccmp049	10.2.0.2	
125	oneadmin	oneadmin	compute_1_(service_92)	RUNNING	asccmp048	10.2.0.3	
126	oneadmin	oneadmin	compute_2_(service_92)	RUNNING	asccmp047	10.2.0.4	
127	oneadmin	oneadmin	compute_3_(service_92)	RUNNING	asccmp046	10.2.0.5	
128	oneadmin	oneadmin	compute_4_(service_92)	RUNNING	asccmp045	10.2.0.6	
129	oneadmin	oneadmin	compute_5_(service_92)	RUNNING	asccmp044	10.2.0.7	
130	oneadmin	oneadmin	compute_6_(service_92)	RUNNING	asccmp043	10.2.0.8	
131	oneadmin	oneadmin	compute_7_(service_92)	RUNNING	asccmp042	10.2.0.9	

10 Showing 1 to 9 of 9 entries « ‹ 1 › »

OpenNebula 4.2.0 by C12G Labs.

Login the VC and run applications

capitan@dock6-frontend-v2:~/demo-pragma25

```
koheі@kooh ~
$ ssh capitan@163.220.57.212
Last login: Wed Oct 16 18:28:36 2013 from 159.226.189.177
[capitan@dock6-frontend-v2 ~]$ cd demo-pragma25/
[capitan@dock6-frontend-v2 demo-pragma25]$ ./demo.sh
machinfile:
dock6-frontend-v2
dock6-compute-v2-0-0-0-0
dock6-compute-v2-0-0-0-1
dock6-compute-v2-0-0-0-2
dock6-compute-v2-0-0-0-3
dock6-compute-v2-0-0-0-4
dock6-compute-v2-0-0-0-5
dock6-compute-v2-0-0-0-6
dock6-compute-v2-0-0-0-7


execute:
nohup mpirun -np 9 -machinefile machinefile dock6.mpi -i dock.in -o dock.out
nohup: appending output to `nohup.out'
[capitan@dock6-frontend-v2 demo-pragma25]$
```

Acknowledgements

- Luca Clementi, Nadya Williams, Philip Papadopoulos (UCSD)
- Pongsakorn U-chupala (NAIST)
- Yoshio Tanaka, Akihiko Ota (AIST)