Building The PRAGMA International Cloud

2011-

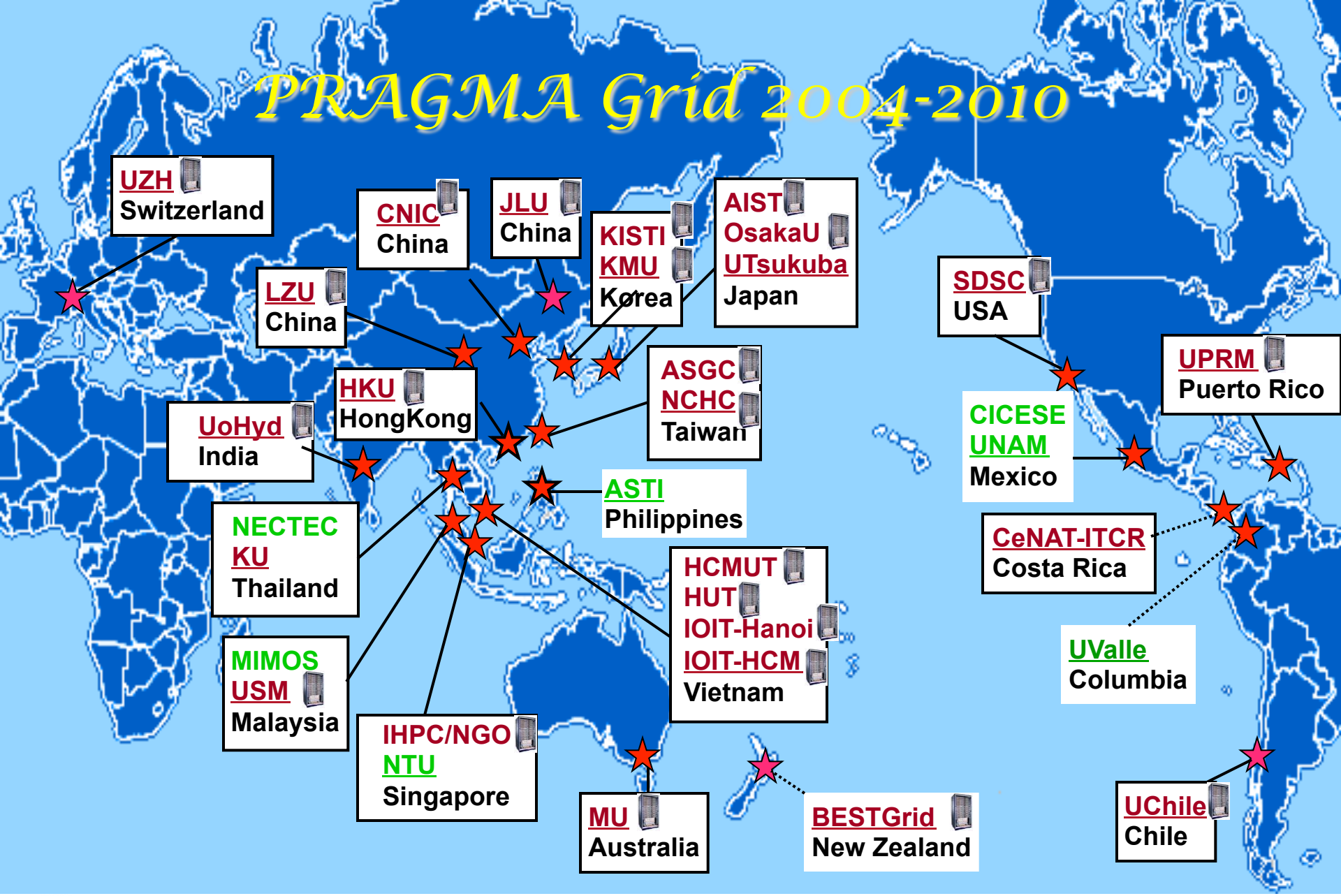
Cindy Zheng

For

Resources Working Group

TOC

- Why Grid to Cloud
- Heterogeneous Cloud needs and solutions
  - Distributed/Common file system
  - Overall strategies
  - Key technologies
    - VM/VC sharing
    - SDN
- Building infrastructure for scientists
  - Lifemapper



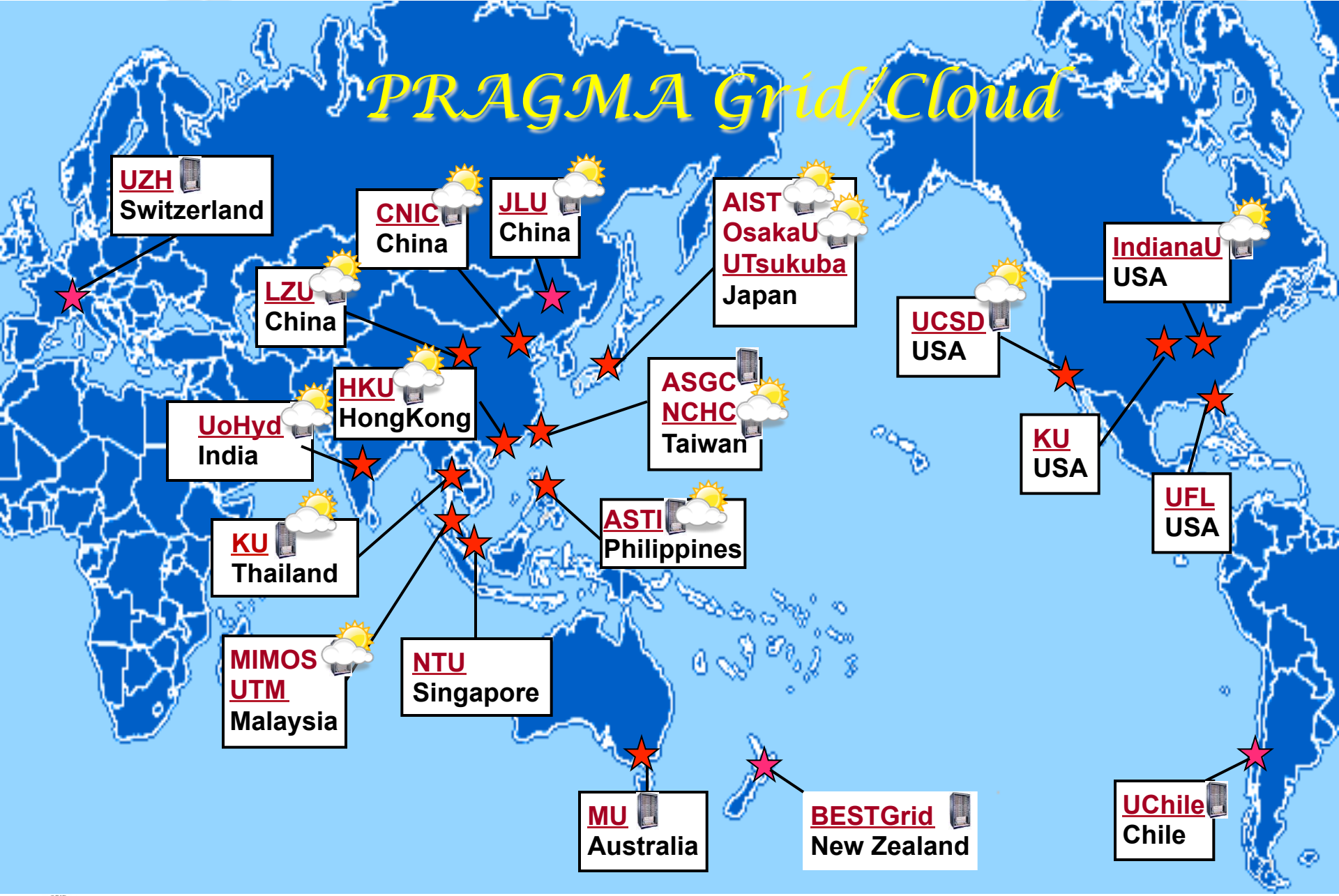
28 institutions in 17 countries/regions, 22 compute sites (+ 7 site in preparation)

Cindy Zheng, Pragma Cloud, 3/20/2013

Shifting Focus to Cloud

http://goc.pragma-grid.net

- Basic goals unchanged
  - Distributed resources
    - Data collection/storage, expertise, computing, …
  - Needs of sharing and collaborations
  - Try to make easier for scientists to collaborate and use distributed resources
- Grid is still hard to use
  - Main difficulty: provide software environments for users
- Cloud can be easier to use
  - VM allow customize/maintain user environment
- But needs work
  - How can users deploy VMs/VCs easily in a heterogeneous global cloud



23 institutions in 14 countries/regions, 18 compute sites, 13 Cloud sites

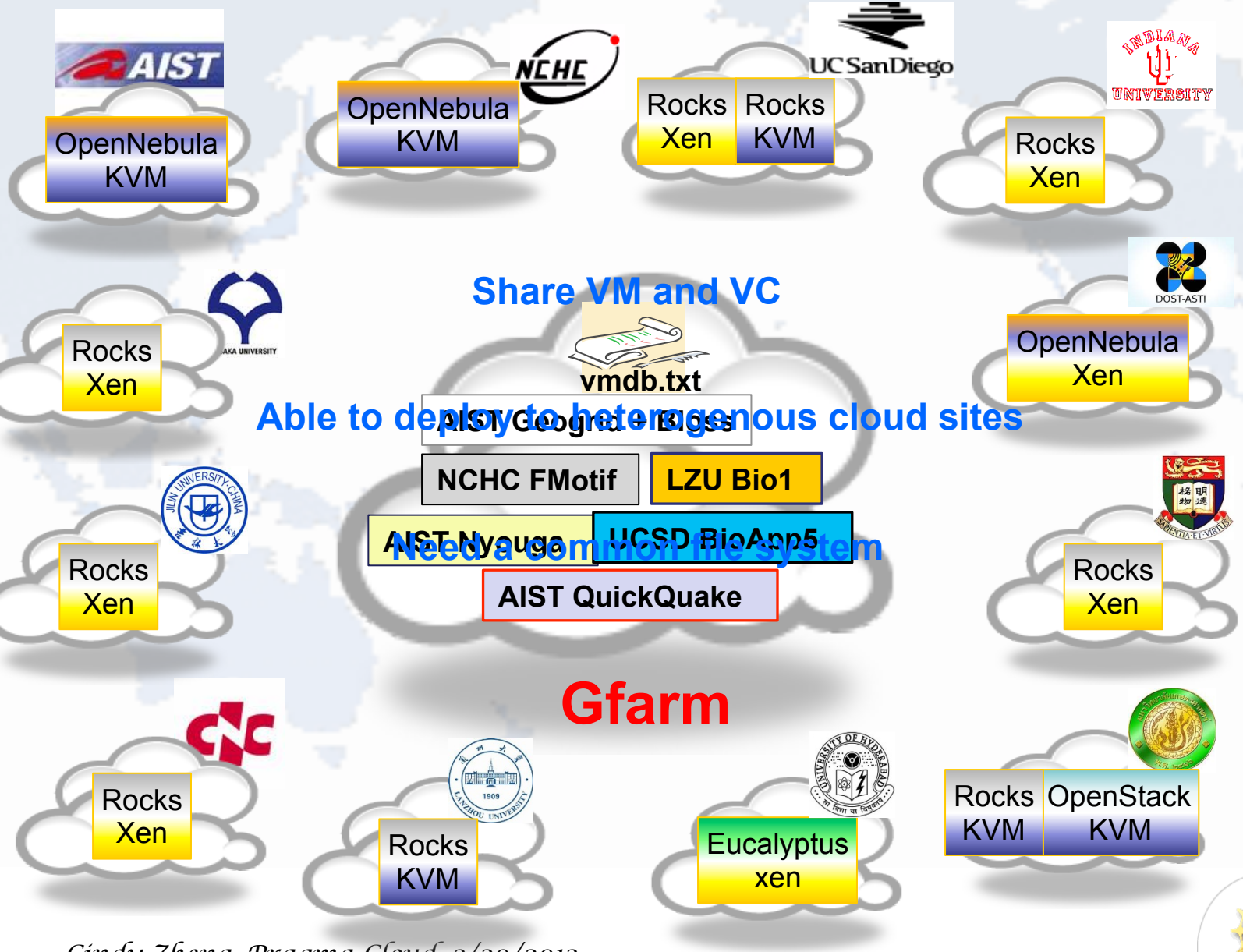
March 4, 2013

Cindy Zheng, Pragma Cloud, 3/20/2013

PRAGMA Cloud Resources

	Institution	Region	Host Name	nodes	CPUs	Memory (GB)	Disk Space (TB)	CPU Model	CPU Speed (MHz)	VM Platform	System Type	VM Manager	OS Release
	AIST	Japan	pragma-v	32	64	768	600	X86_64	2400	KVM	Cluster	OpenNebula	Linux kernel 3.0.4
	ASTI	Phillipines	one	10	80	160	5	X86_64	2000	Xen	Server	OpenNebula	2.6.18-238.19.1.el5
	CNIC	China	cluster	1	2	128	2	X86_64	2400	Xen	Server	Rocks 5.4.3	2.6.18-238.19.1.el5xen
	HKU	Hong Kong	hpcrocks	3	8	12	1	X86_64	2530	Xen	Cluster	Rocks 5.4.3	2.6.18-274.17.1.el5xen
	IU	USA	pragma	8	16	128	12	X86_64	2000	Xen	Cluster	Rocks 5.4	2.6.18-194.17.4.el5xen
	JLU	China	gfarm	1	4	34	1	X86_64	2000	Xen	Server	Rocks 5.4.3	2.6.18-238.19.1.el5xen
	KU	Thailand	yuuko	1	4	16	0.25	X86_64	2500	KVM	Server	Rocks 6.0	2.6.32-220.13.1.el6
	KU	Thailand	akatosh	3	8	6	0.6	X86_64	3000	KVM	Cluster	OpenStack	2.6.32-220.13.1.el6
	LZU	China	vm	1	8	16	1	X86_64	4122	KVM	Server	Rocks 6.0	2.6.18-238.12.1.el5
	MIMOS	Malaysia	vm	6	12	192	1	X86_64	2400	KVM	Cluster	Eucalyptus	2.6.18-194.11.1.el5
	NCHC	Taiwan	Snowfox	8	64	128	2	X86_64	2500	KVM	Cluster	OpenNebula	2.6.18-194.11.1.el5
	OSAKAU	Japan	cider	3	24	34	3	X86_64	2270	Xen	Cluster	Rocks 5.4	2.6.18-194.17.4.el5xen
	UCSD	USA	fiji	18	25	380	10	X86_64	2261	KVM	Cluster	Rocks 6.1	2.6.32-279.14.1.el6.x86_64
	UCSD	USA	rockstar	32	64	512	16	X86_64	2000	Xen	Cluster	Rocks 5.4.3	2.6.18-238.19.1.el5xen
	UCSD	USA	calit2-191-121	4	16	32	6	X86_64	3200	KVM	Server	Rocks 6.1	2.6.32-279.14.1.el6.x86_64
	UoHyd	India	venus	4	4	16	4	X86_64	2400	Xen	Cluster	Eucalyptus	2.6.18-194.11.1.el5
Total	13	9	16	135	403	2562	663.31						

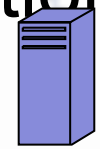
PRAGMA Heterogeneous Cloud

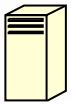


Gfarm

http://goc.pragma-grid.net/wiki/index.php/Gfarm

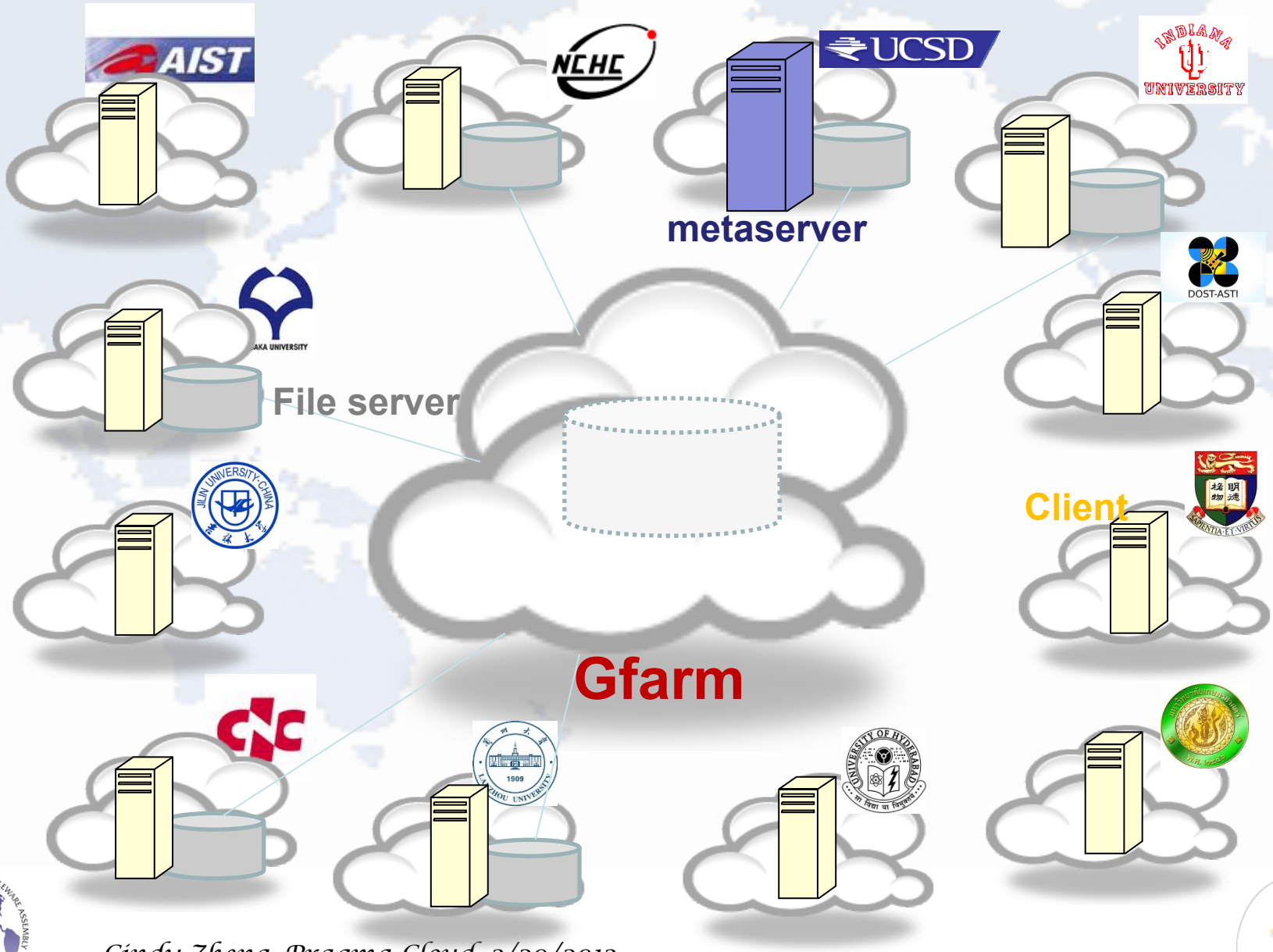
- Open source
- Distributed file system
- Developer – University of Tsukuba
- Used by PRAGMA Grid and some regional networks
- GSI authentication
- Meta-servers
- File servers
- Clients





PRAGMA Cloud Storage – Gfarm

http://goc.pragma-grid.net/wiki/index.php/Gfarm



Overall Strategy

- Goals
  - Easy to use
    - Distributed computing
    - Distributed services
    - Use distributed data
  - Secure
    - Private network span globally
- Technologies
  - Virtualization
    - Sharing VM/VC images among heterogeneous clouds
    - Moving big data is slow
    - Running application where the data is
  - SDN
    - Secure envelopes for projects
- How
  - Pilot team leads experiments
  - Build infrastructure for scientists
    - Work with science team – first Biodiversity

VM/VC Sharing Experiments

- Methedology
  - Pilot team (UCSD, AIST, NCHC)
  - Extended testing team (All PRAGMA Cloud sites)
- Experiments
  - Manually create VMs - pilot
  - Manually deploy VMs - pilot
    - Between KVM and Xen
  - Automate VM deployment – pilot then PRAGMA Cloud
  - Manually create VCs - pilot
  - Manually deploy VCs - pilot
  - Automate VC deployment – pilot then PRAGMA Cloud sites
- Results
  - Pilot team designed VM/VC sharing scheme and interface standards
  - Pilot team developed software and share with all sites
  - Successfully auto-deploy VM/VC among many PRAGMA Cloud sites

Define Roles and Processes

- VM/VC authors
  - Create VM/VC with applications
    - If a VM/VC includes any network-sensitive service/application and needs adjustments when the VM/VC is deployed in a new network environment, the VM/VC author should provide script /root/VMreconfig or /root/VCreconfig to reconfigure the VM and VC
  - Deposit VM/VC images into Gfarm
  - Provide information about the VM/VC
- VM/VC users
  - Request resources for VM/VC deployment
  - Deploy VM/VC use scripts provided by resources providers
  - Run /root/VMreconfig or /root/VCreconfig if provided
  - Manage the VM/VC and run applications
- Resources providers
  - Provide resources and access to users
  - Provide easy means for users to deploy and manage VM/VCs

Implement Easy VM/VC deployment

- Automate deployment
  - Implementation per virtual environment
    - Virtualization engine (KVM, Xen, …)
    - Virtualization manager (Rocks, OpenNebula, …)
  - Share among sites with the same virtual environment
- Standard interface
  - Transparency for users
  - Enable easy implementation among different virtual environment

Design Interface Standards

- VM/VC image depository
- VM/VC datebase
- VC deployment input xml file
- VC deployment output xml file
- VM/VC deployment script command-line

VM/VC Images Depository Standards

- VM/VC images Gfarm depository directory structure

  /vm-images/

  vmdb.txt

  vcdb.txt

  <author institution>/

  VM images

  VC directory

  Frontend image, compute node image, xml file

- Example

```
$ gfls /vm-images/SDSC/calit2-119-222
-rw-r--r-- 1 zhengc   gfarmadm 10293547538 Jan 24 10:12 calit2-119-222.img.gz
-rw-r--r-- 1 zhengc   gfarmadm        1148 Feb 21 13:35 calit2-119-222.xml
-rw-r--r-- 1 zhengc   gfarmadm  1693669044 Jan 24 10:20 hosted-vm-0-0-1.img.gz
```


VM/VC Database Standards

- VM database

```
$ gfexport /vm-images/vmdb.txt
lzu-bio1,kvm-xen,LZU/lzu-bio1.img.gz,418f8b472dc7578bbfd661a71c712591
bioapp5,xen-kvm,SDSC/bioapp5.img.gz,fd555578ddd725c07d3ee378452d8147
fmotif,kvm-xen,NCHC/fmotif.hda.gz,06fff5f211750a38fd0883e8c97102ff
geobloss,kvm-xen,AIST/geobloss.hda.gz,fcf9b1e45098f2e8053afc1e0e15b6a5
nyouga,xen-kvm,AIST/nyouga.img.gz,e4342a37740b9998c781fe2210fa8989
quiquake,xen-kvm,AIST/quiquake.img.gz,db64fc7fcaced46ef3b7baf7e3441418
worker,xen-kvm,AIST/worker_qq.img.gz,83113dddf665e86368a15d0682a3a532
```

VM-name boot-order image path checksum

- VC database

```
$ gfexport /vm-images/vcdb.txt
calit2-119-222,SDSC/calit2-119-222/calit2-119-222.xml
```

VC-name XML file path

– All VC files are in the same directory – images, xml, checksum

VC Deployment Input

http://goc.pragma-grid.net/wiki/index.php/Vc-in.xml

VC-in.xml template

- Cluster topology
- KVM or Xen
- Architecture, HVM?
- Memory size, number of CPUs
- Network interface name, type
- How to boot
- …

VC Deployment Output

http://goc.pragma-grid.net/wiki/index.php/Vc-out.xml

VC-out.xml template

- Cluster topology
- KVM or Xen
- Architecture, HVM?
- Number of compute nodes
- Each compute node
  - name, MAC address, IP address
- …

VC Deployment Command-line Interface

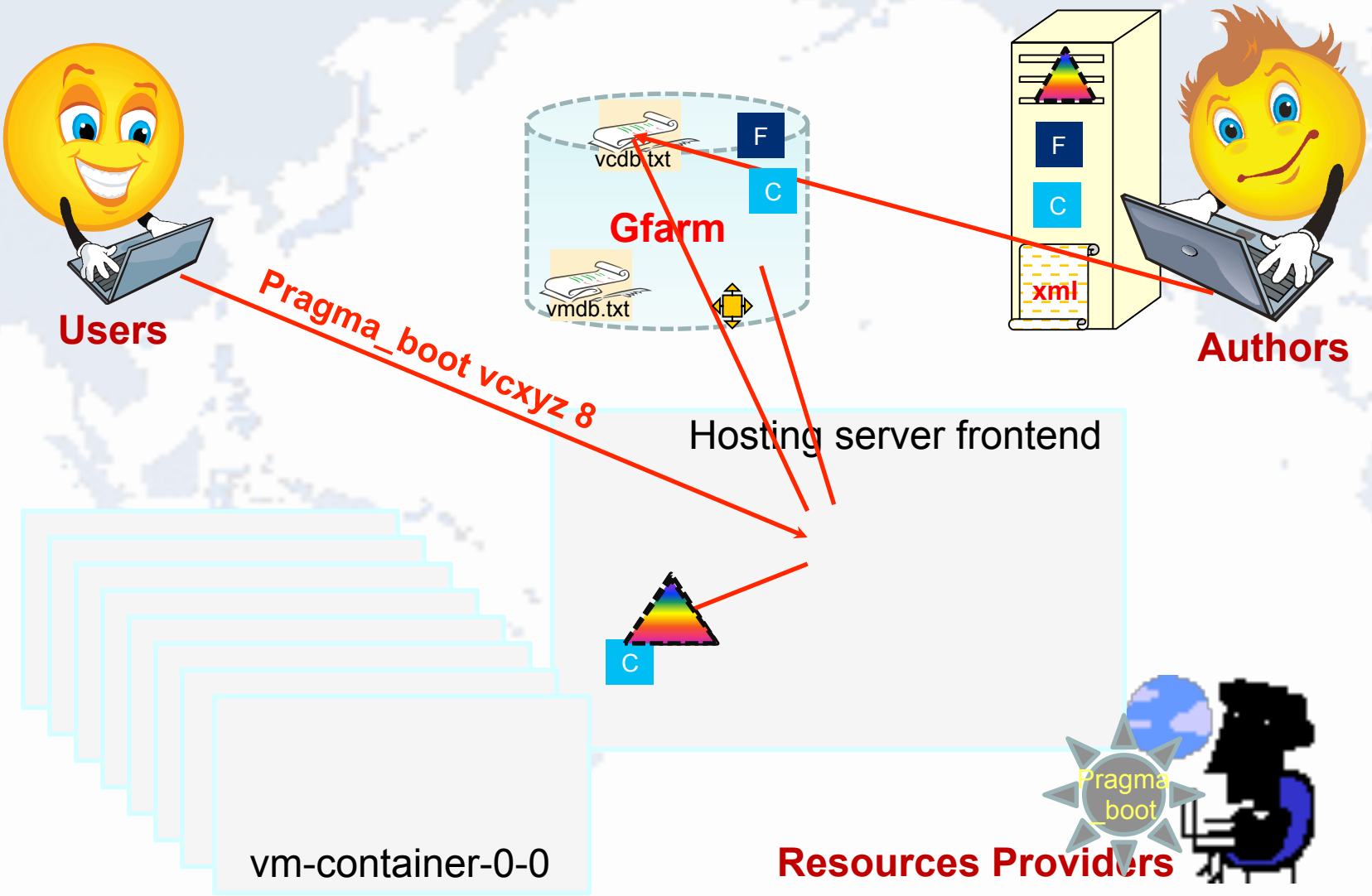
http://goc.pragma-grid.net/wiki/index.php/Vc-interface

$ pragma_boot vc-name [number of compute nodes] [local VC image depository path]

- "pragma_boot" is the script name, should be run in a normal user account
- "vc-name" is the name of the VC to be deployed. It's a required argument and should match the first field of an entry in vcdb.txt
- "number of compute node" is an optional argument. Default is 1.
- "local VC image depository path" is an optional argument. Default is standard gfarm path.

- Hide heterogeneity from users
- Needs minimum one implementation per virtual environment (virtualization engine, virtualization manager)

The Process


Users
Pragma_boot vcxyz 8
vcdb.txt
F
C
Gfarm
vmdb.txt
F
C
xml
Authors
Hosting server frontend
C
vm-container-0-0
Pragma
_boot
Resources Providers

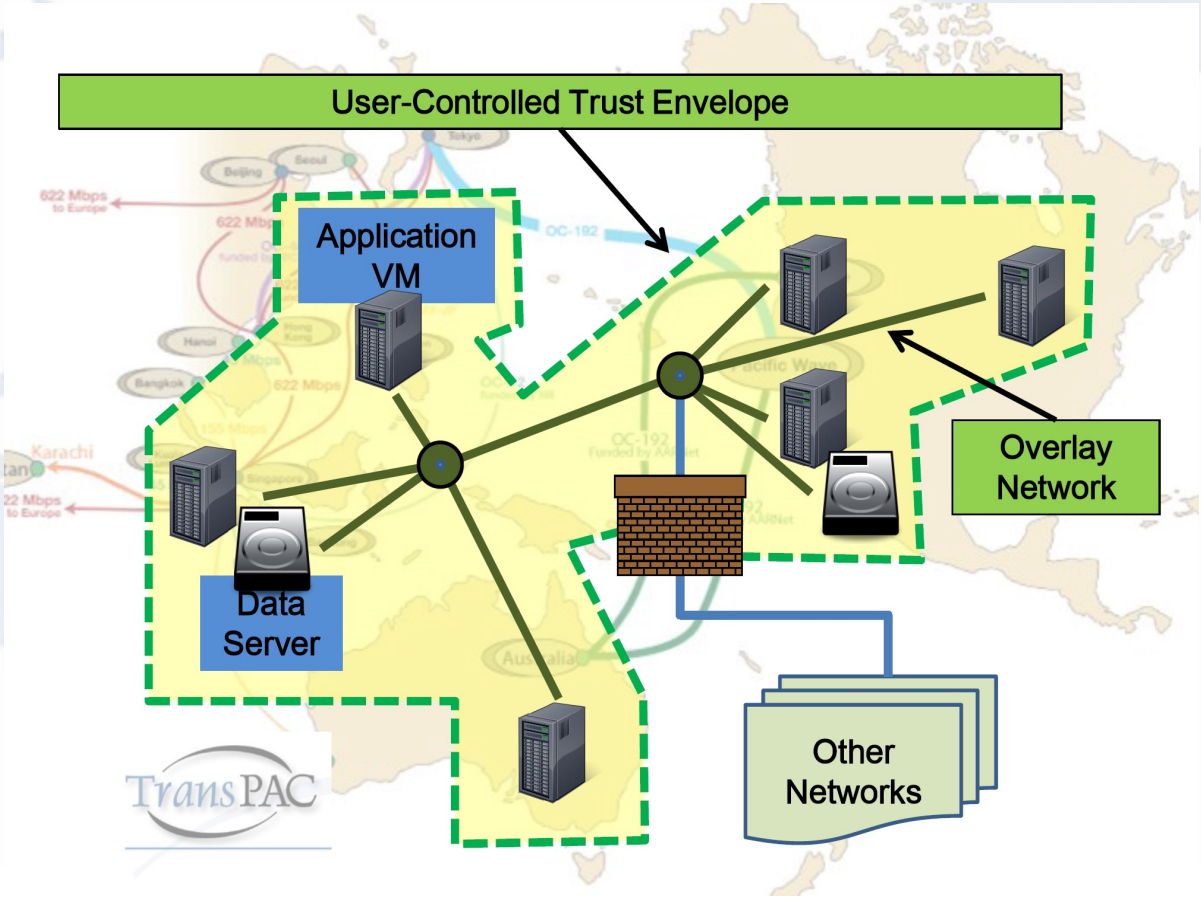
Automated VM/VC Deployment Solutions

- VM Deployment Script
  - Rocks/Xen (UCSD), installed and ran on 7 sites
    http://goc.pragma-grid.net/wiki/index.php/Vm-deploy-multi
  - OpenNebula/KVM (AIST)
    http://goc.pragma-grid.net/wiki/index.php/Auto-deploy_with_KVM/OpenNebula
- VC Deployment Script – demo tomorrow
  - Rocks/KVM (UCSD)
    http://goc.pragma-grid.net/wiki/index.php/Auto-deploy_VC_on_Rocks
  - OpenNebula/KVM (AIST)
- Next phase
  - Expand usage to more Rocks/KVM-Xen and OpenNebula/KVM sites
  - More implementation in other virtual environment

Network Overlay Experiment

http://goc.pragma-grid.net/wiki/index.php/Network_Overlay

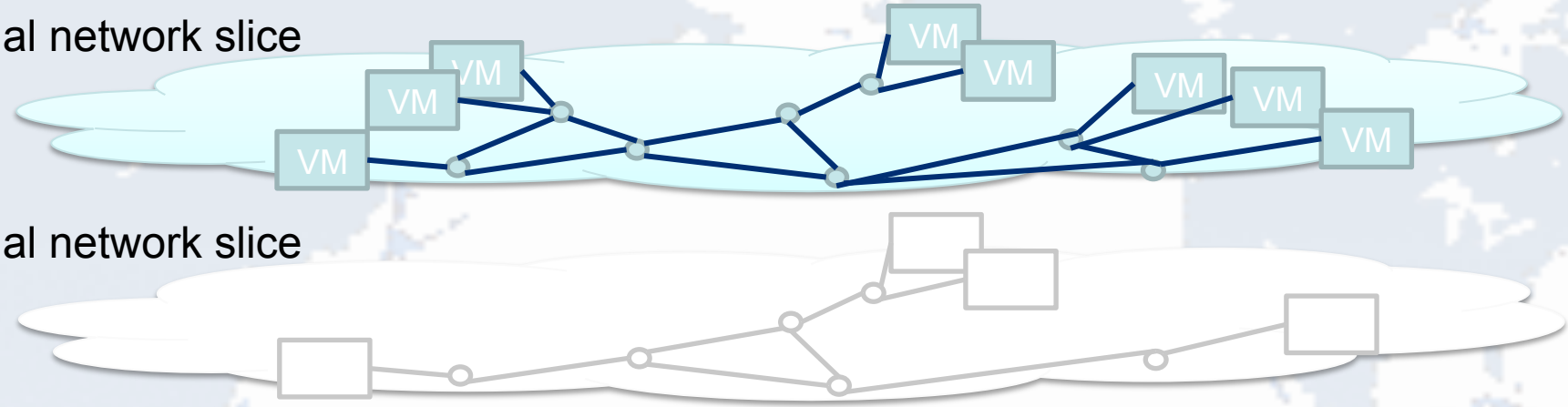
- OpenFlow
- ViNe



Openflow network environment

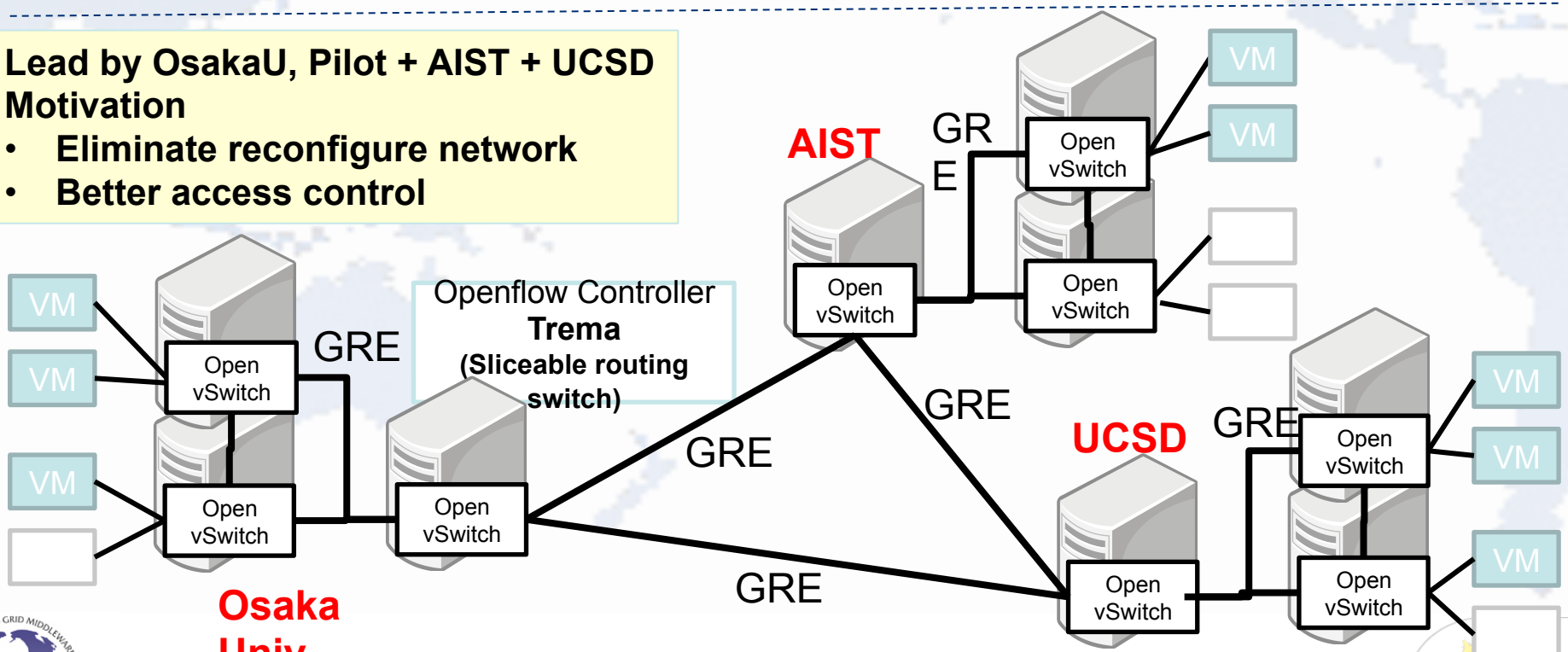
Virtual network slice A

Virtual network slice B



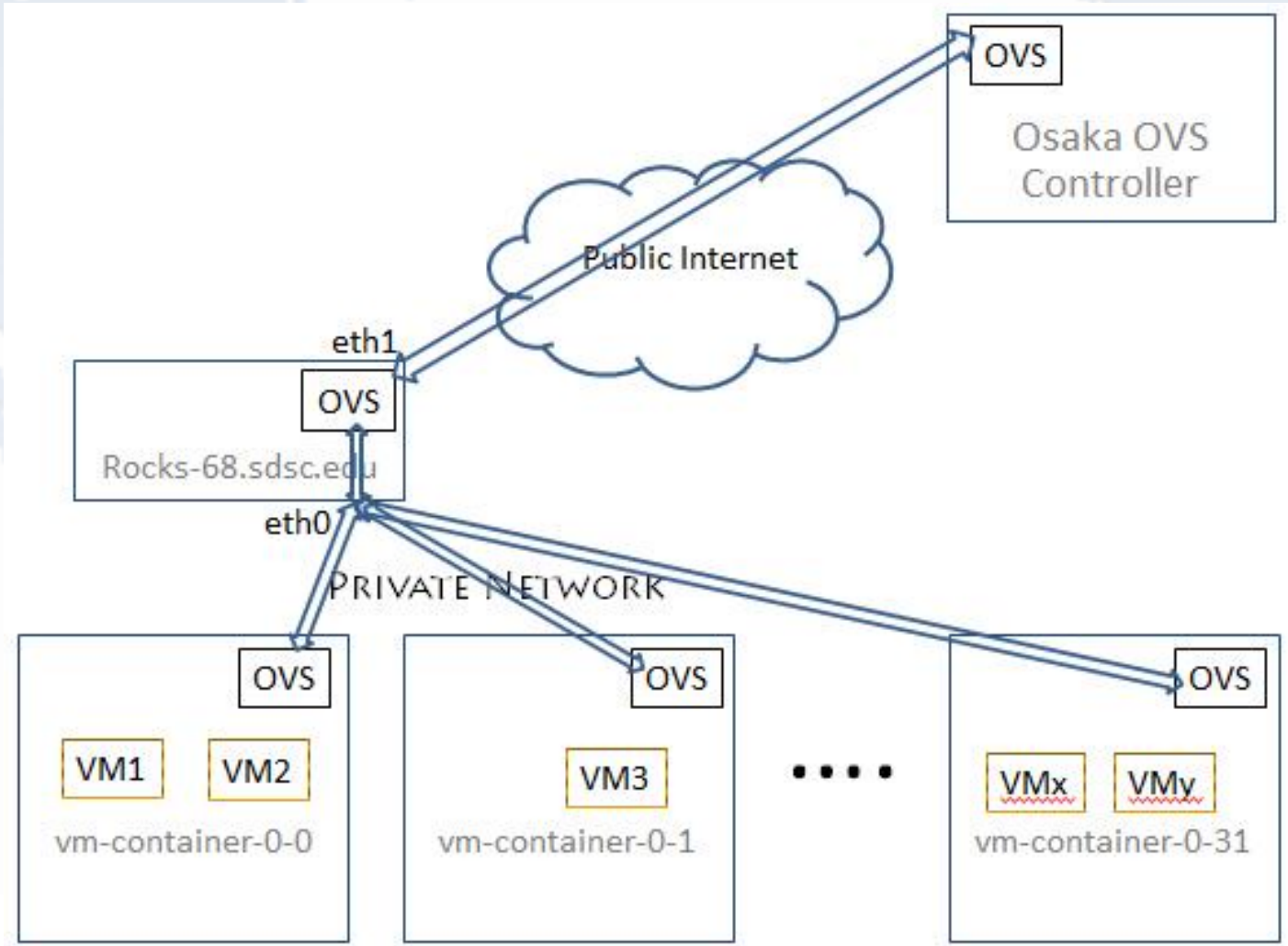
Lead by OsakaU, Pilot + AIST + UCSD
Motivation

- **Eliminate reconfigure network**
- **Better access control**



Site Setup Example For Software OpenFlow

http://goc.pragma-grid.net/wiki/index.php/Network_overlay_setup_at_UCSD



OpenFlow Experiments In PRAGMA Cloud

- First experiment by pilot team (OsakaU, AIST, UCSD)
  - Successfully setup a OpenFlow subnet
  - Easy deployment of Geogrid VMs as condor workers
  - Successfully ran Geogrid applications
- OpenFlow activities fourishing in PRAGMA Cloud
  - OsakaU leads expansion and further research on performance (demo)
  - UCSD getting ready to test hardware OpenFlow (OpenFlow switch) functionalities and performance
  - NTU, KU-Thailand setup OpenVswitch at their sites and connecting to PRAGMA Cloud OpenFlow network
  - LZU built a Openflow switch test environment use NetFPGA at their site

Another Network Overlay – ViNe Experiment

http://goc.pragma-grid.net/pragma-doc/pragma23/posters/vine-Maur%c3%adcioTsugawa.pdf

- Lead by UFL
- First experiment, +AIST
  - Setup ViNe subnet
  - Successfully tested VM live migration
- Second experiment, +UCSD+IU
  - Setup ViNe subnets

    http://goc.pragma-grid.net/wiki/index.php/UCSD_implementation

    http://goc.pragma-grid.net/wiki/index.php/IU_implementation
  - Successfully tested VM live migration
  - Successfully deploy VC and run condor job (demo)

    http://goc.pragma-grid.net/wiki/index.php/Condor-ViNe_test

Building Infrastructure For Scientists

http://goc.pragma-grid.net/wiki/index.php/Building_CI_For_Scientists

First case - Lifemapper

Mount Kinabalu specimen data storage

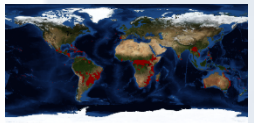
(1)

GeoPortal at UTM

(2)

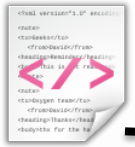
(3)

LifeMapper WebService at UKansas



Predicted Habitat

(4)



Metadata

(5)

GeoPortal at IU PRAGMA node

(1) A script queries GeoPortal instance at UTM (Malaysia)to retrieve raster data
(2) Retrieves raster data from stored dataset
(3) Submit dataset to LifeMapper as occurrences/experiments
   I. Prepare occurrence dataset
   II. Post occurrence set
   III. Post experiment
(4) LifeMapper outputs prediction result as metadata (EML)
(5) Catalogs LifeMapper predicted output metadata in GeoPortal at IU (US)

Source: Uma Pavalanthan

Infrastructure for Lifemapper

- Needs
  - Private network among biodiversity sites
    - UFL, KU, IU (USA), UTM (Malaysia)
  - Distribute data services
    - Currently only KU, UTM
  - Distribute computing
    - Currently only KU
- Solutions
  - Overlay network
    - Setup and tested ViNe subnet among UFL, IU and UCSD
    - Nest step, add other sites into the subnet
  - Virtualize Lifemapper components
    - First – virtualize Lifemapper compute component

Build a Virtual Cluster for Lifemapper

http://goc.pragma-grid.net/wiki/index.php/Lifemapper_VC

- Team: Kansas University (KU), UCSD
- KU: modulate compute component
- UCSD: build VC, install software
- Team: Testing
- Successfully run lifemapper jobs on VC (demo)
- Next step: deploy the VC to other sites, use automated VC deployment solutions

Welcome Everyone To Get Involved