Educational Virtual Clusters for On-demand MPI/Hadoop/Condor in FutureGrid

Renato Figueiredo
Panoat Chuchaisri, David Wolinsky
ACIS Lab - University of Florida

Goals and Approach

- A flexible, extensible platform for *hands-on,* education on parallel and distributed systems
- Focus on *usability* – lower entry barrier
  - Plug and play, open-source
- Virtualization + virtual networking to create educational sandboxes
  - *Virtual appliances*: self-contained, pre-packaged execution environments
  - *Group VPNs*: simple management of virtual clusters by students and educators

Guiding principles

- Full-blown, pre-configured, plug-and-play well-known middleware stacks
  - Condor – high-throughput, workflows
  - MPI – parallel
  - Hadoop – data parallel, Map/Reduce
- Quick user start – minutes to first job
  - Gain access to FutureGrid, or use desktop VMs
  - Shared playground cluster
- Isolated sandbox clusters; flexibility
  - Allow individuals, groups complete control over their virtual cluster - root access

Outline

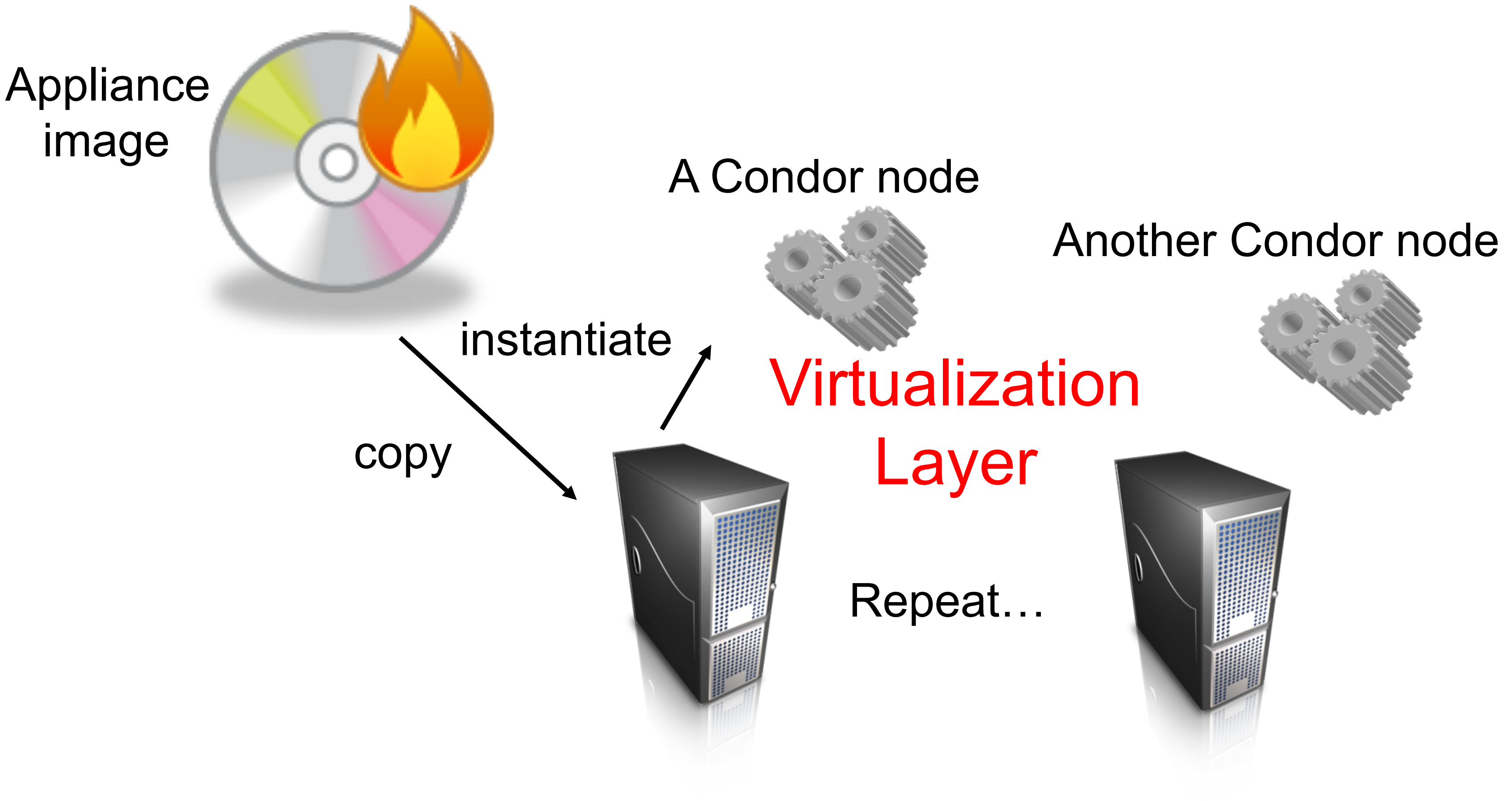
- Overview
  - Virtual appliances, networks
  - FutureGrid resources
  - FutureGrid accounts
- Deploying an appliance and connecting to playground virtual cluster
  - Condor self-configuration
  - Deploying MPI parallel jobs
  - Deploying Hadoop pools

What is a virtual appliance?

- An appliance that packages software and configuration needed for a particular purpose into a virtual machine “image”
- The virtual appliance has no hardware – just software and configuration
- The image is a (big) file
- It can be *instantiated* on hardware
  - Desktops: VMware, VirtualBox
  - Clouds: FutureGrid, Amazon EC2

Grid appliances

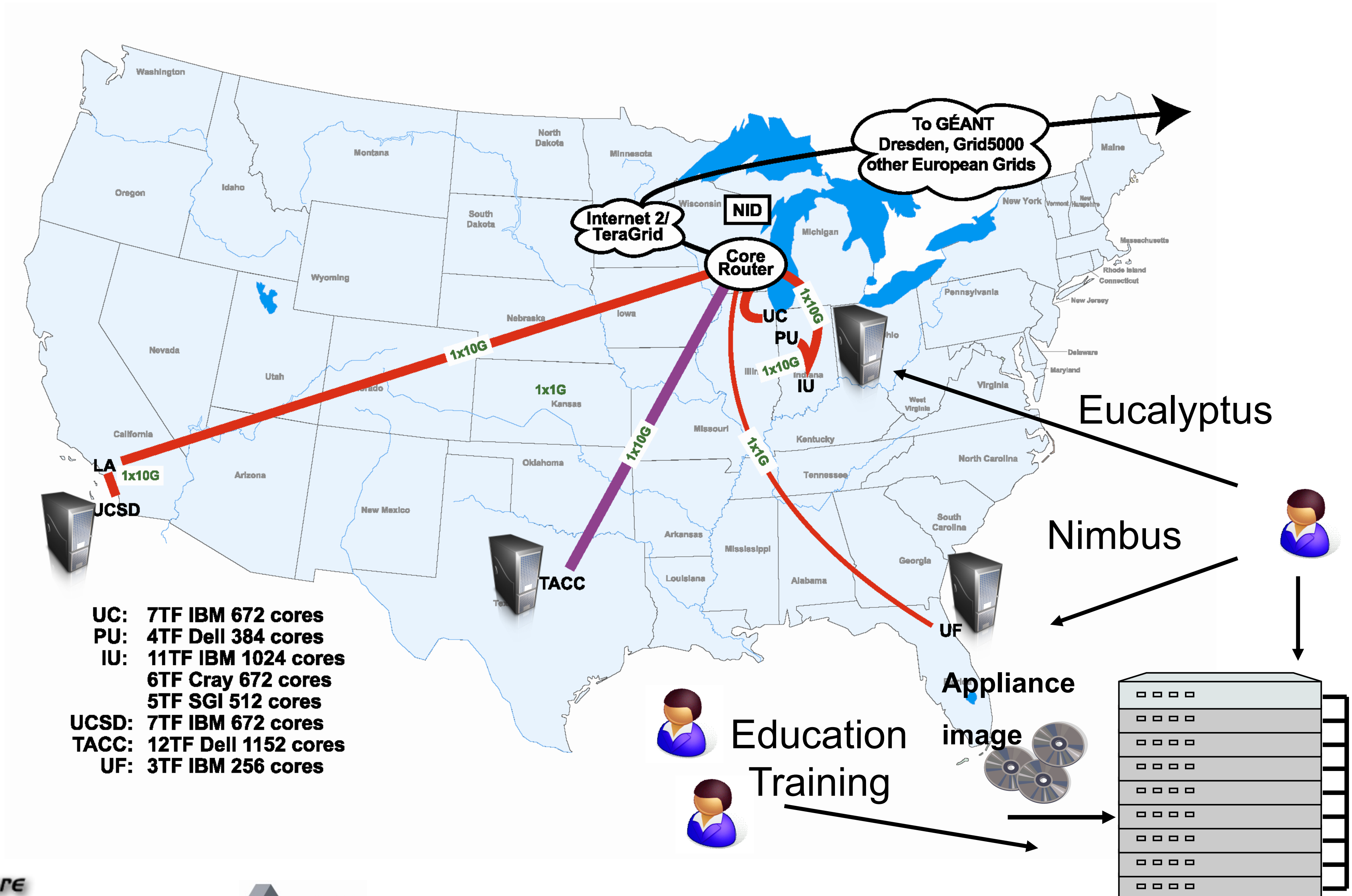
- Baseline image: self-configures Condor



Virtual network, configuration

- P2P overlay used to self-organize virtual private network (VPN) of appliances
  - Akin to Skype
  - Virtual cluster; assign IP addresses in virtual space through DHCP – support existing middleware (Condor, MPI, Hadoop)
- P2P overlay also used to self-configure middleware
  - Akin to Bonjour/UPnP
  - Condor manager advertises itself; Condor workers discover and register with manager

FutureGrid resources



UC:	7TF IBM 672 cores
PU:	4TF Dell 384 cores
IU:	11TF IBM 1024 cores
	6TF Cray 672 cores
	5TF SGI 512 cores
UCSD:	7TF IBM 672 cores
TACC:	12TF Dell 1152 cores
UF:	3TF IBM 256 cores

Using FutureGrid – accounts 101

- Create a portal account
  - Can access and post content, manage profile
  - Identity verification – no resources allocated, but users can interact with portal
    - E.g. cloud computing class community page
- Create or join a project
  - Request needs to be authorized, and resources granted
  - Portal users can then be added to the project
    - E.g. a cloud class instructor submits a project request; students request portal accounts; instructor uses portal to add students to class

Web site – FG account

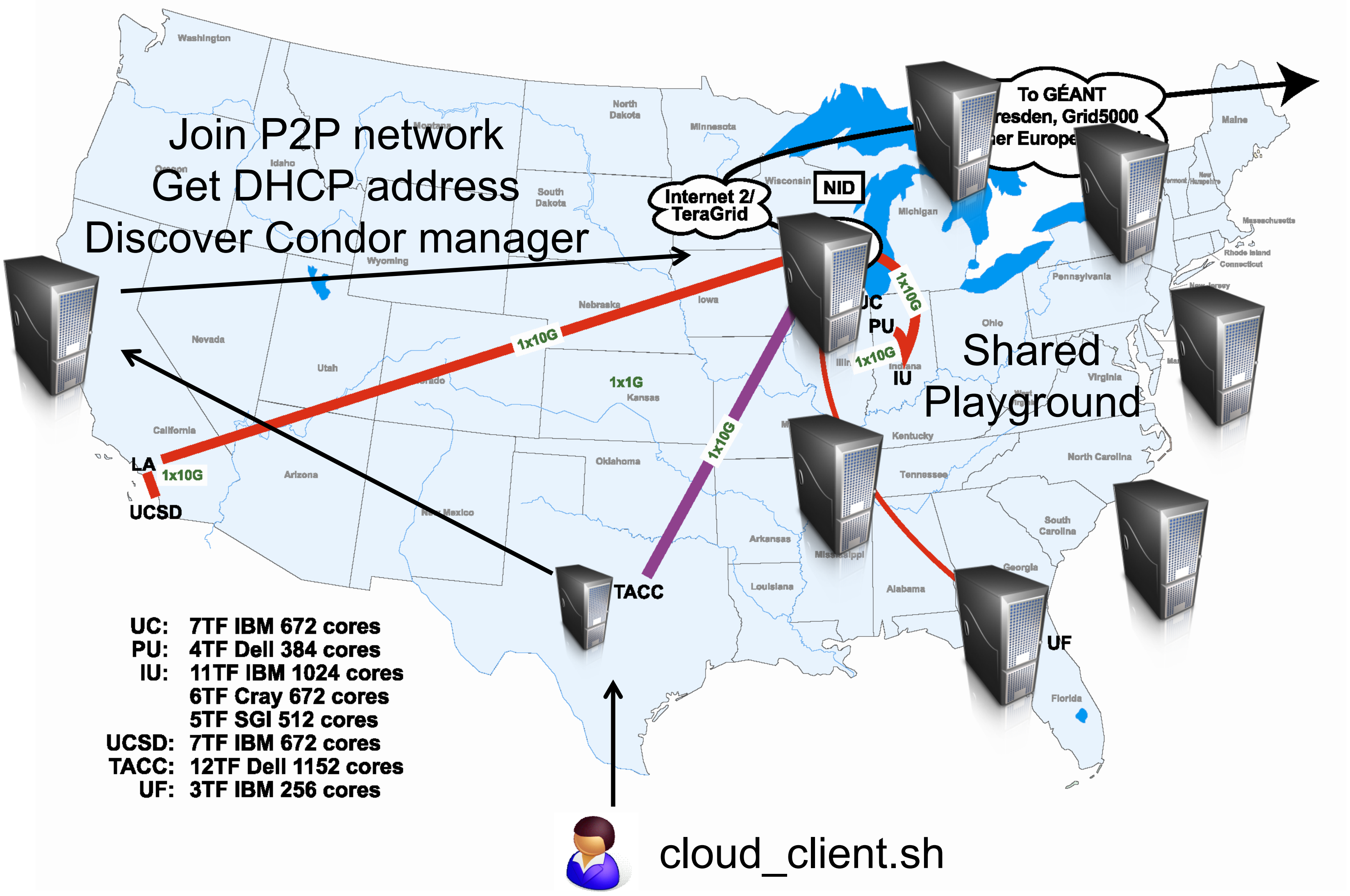
Using FutureGrid – cloud 101

- Once a user has a portal account and project, he/she can use Nimbus or Eucalyptus to instantiate appliances on the different FutureGrid Clouds
    - Tutorials show steps to deploy appliances with a single-line command
- Refer to portal.futuregrid.org
    - Under "User information":
        - Getting started – to get accounts
        - Using Clouds – Nimbus, Eucalyptus
        - Pointers to relevant tutorials

User perspective – first steps

- Deploying the baseline Grid appliance:
  - Nimbus:
    - cloud-client.sh --run --name grid-appliance-2.04.29.gz --hours 24
  - Eucalyptus:
    - euca-run-instance -k mykey -t c1.medium emi-E4ED1880
  - Wait a few minutes
  - ssh root@machine-address
  - You are connected to a pre-deployed ‘playground’ Condor cluster
    - condor_status

Joining Condor pool



Join P2P network
Get DHCP address
Discover Condor manager

Shared Playground

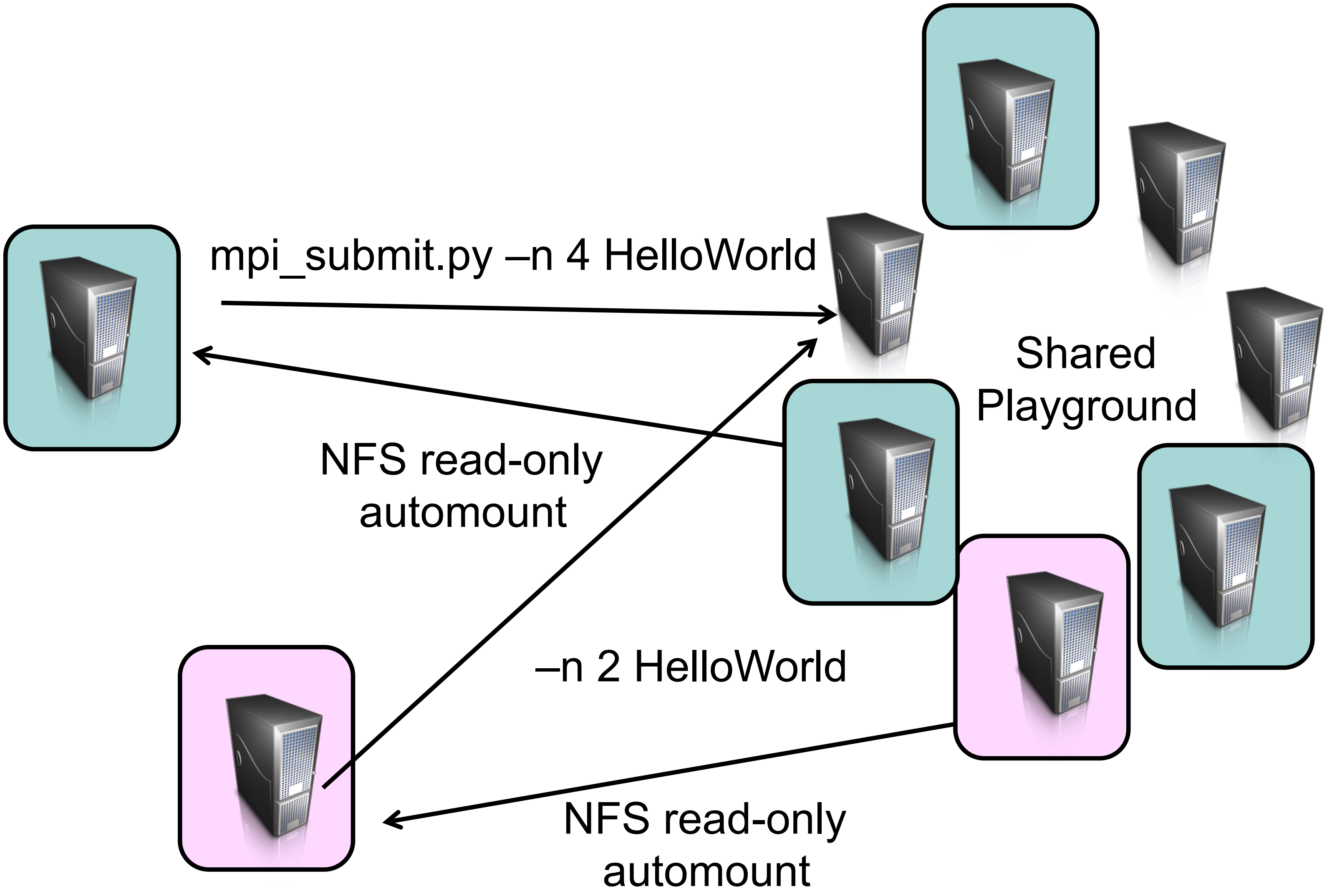
UC:	7TF IBM 672 cores
PU:	4TF Dell 384 cores
IU:	11TF IBM 1024 cores
	6TF Cray 672 cores
	5TF SGI 512 cores
UCSD:	7TF IBM 672 cores
TACC:	12TF Dell 1152 cores
UF:	3TF IBM 256 cores

cloud_client.sh

User perspective – running MPI

- User can install MPI on their appliance
  - “Vanilla” MPI – just run a script to build
  - Advanced classes - user can also deploy custom MPI stacks
- Condor is used to bootstrap MPI rings on demand with help of a script
  - Takes executable and number of nodes
  - Dispatches MPI daemons as Condor jobs
  - Waits for all nodes to report
  - Creates configuration based on nodes
  - Submits MPI task
  - Nodes auto-mount the MPI binaries over NFS

MPI dynamic pools



mpi_submit.py –n 4 HelloWorld

NFS read-only
automount

Shared
Playground

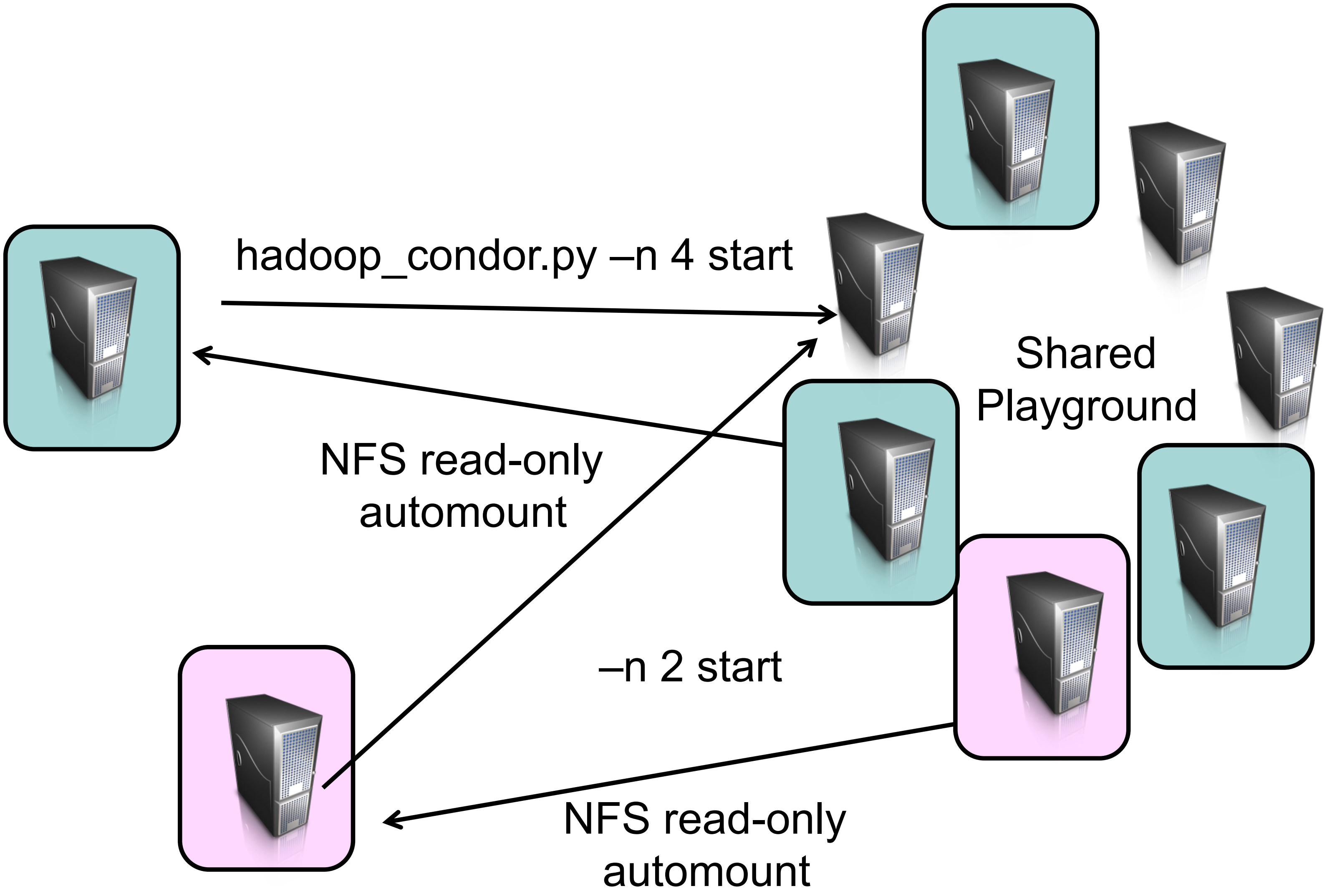
–n 2 HelloWorld

NFS read-only
automount

User perspective – running Hadoop

- User can install Hadoop on their appliance
  - “Vanilla” Hadoop – pre-installed
  - Advanced classes - user can also deploy custom Hadoop stacks
- Condor is used to bootstrap Hadoop pools
  - Takes number of nodes as input
  - Dispatches namenodes, task trackers
  - Waits for all nodes to report
  - Creates configuration based on nodes
  - Nodes auto-mount the Hadoop binaries over NFS
  - After pool is configured, submit tasks, use Hadoop HDFS

Hadoop dynamic pools - create



hadoop_condor.py –n 4 start

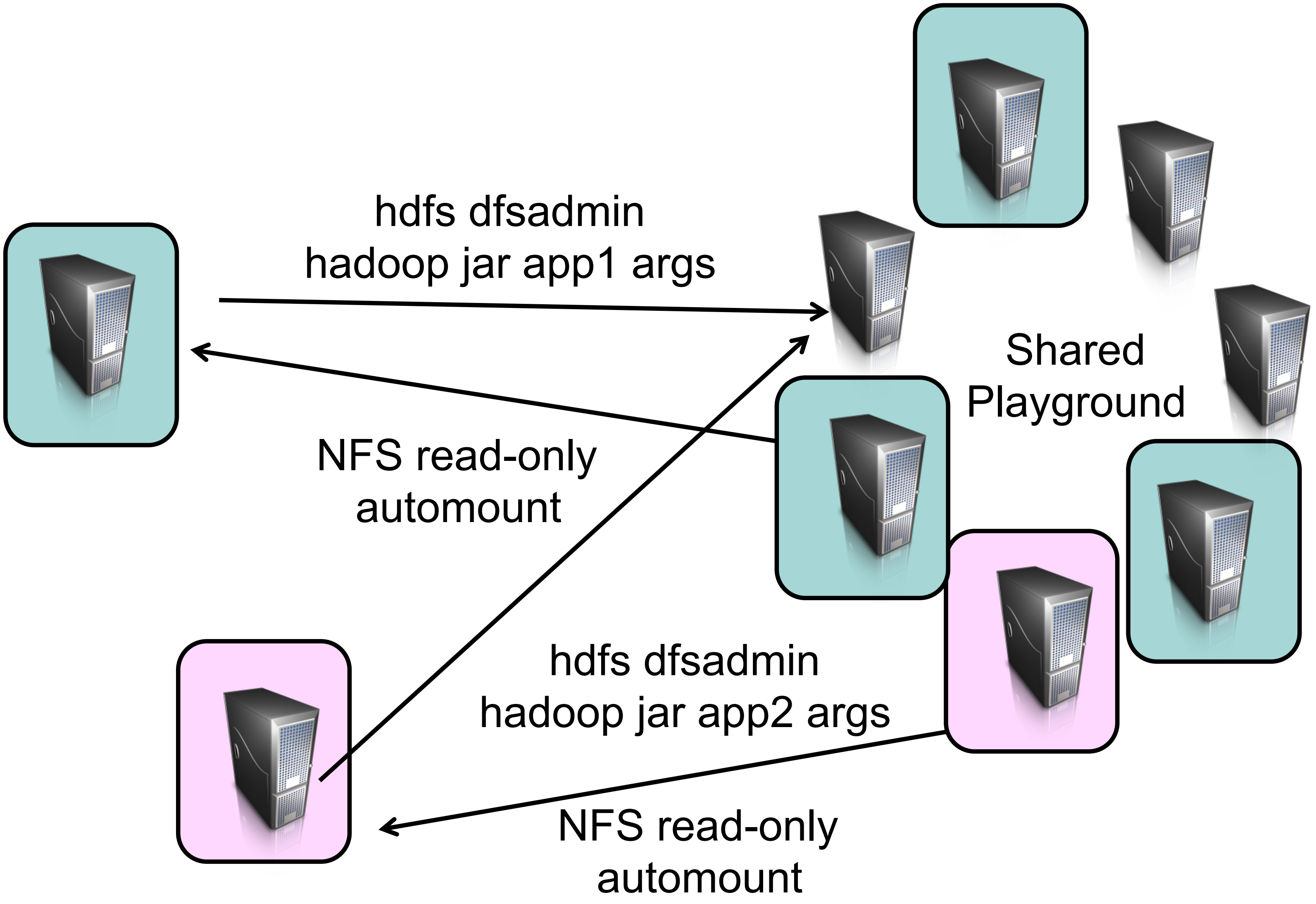
Shared Playground

NFS read-only automount

–n 2 start

NFS read-only automount

Hadoop dynamic pools - run



hdfs dfsadmin
hadoop jar app1 args

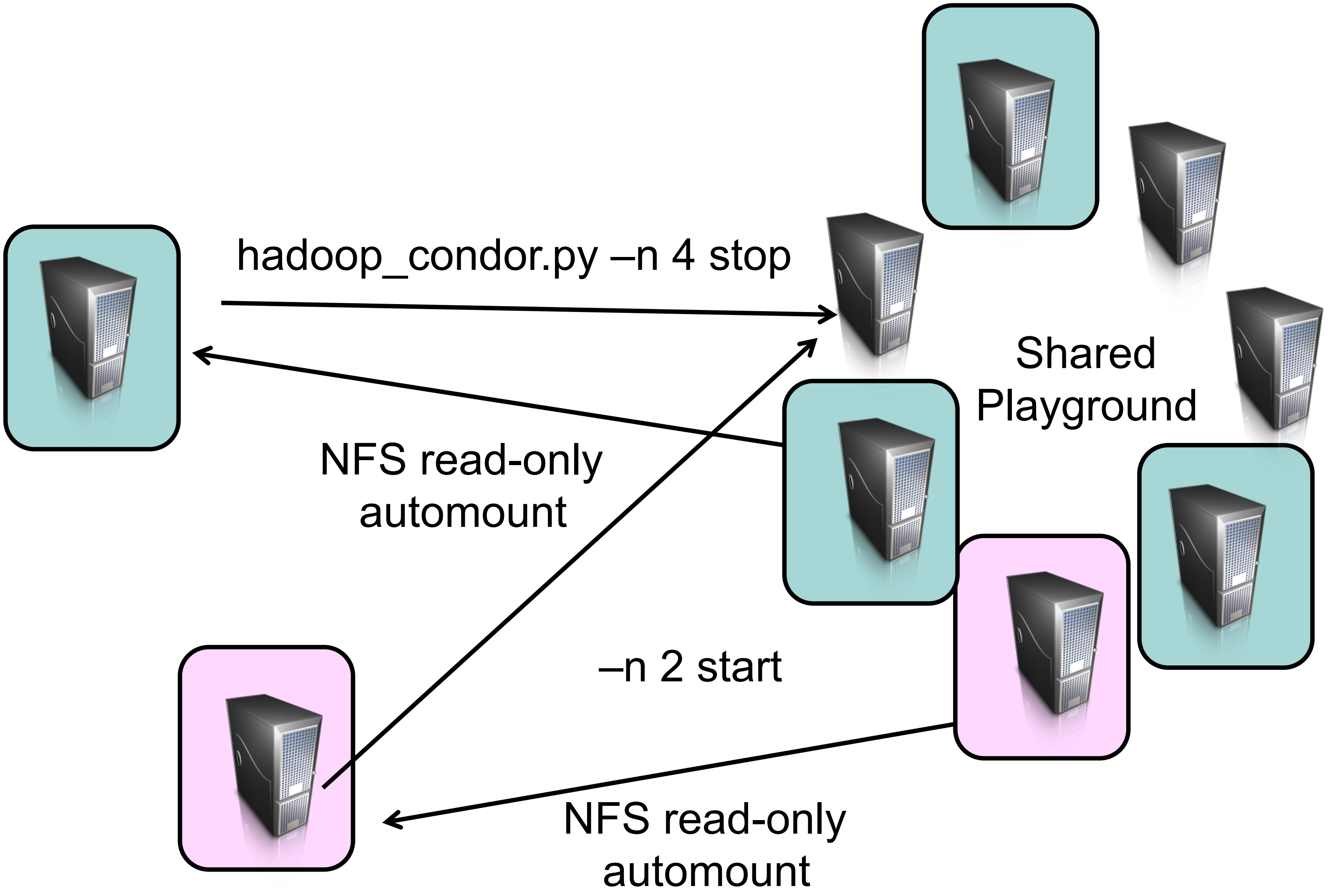
NFS read-only
automount

Shared
Playground

hdfs dfsadmin
hadoop jar app2 args

NFS read-only
automount

Hadoop dynamic pools - teardown



hadoop_condor.py –n 4 stop

Shared Playground

NFS read-only automount

–n 2 start

NFS read-only automount

One appliance, multiple ways to run

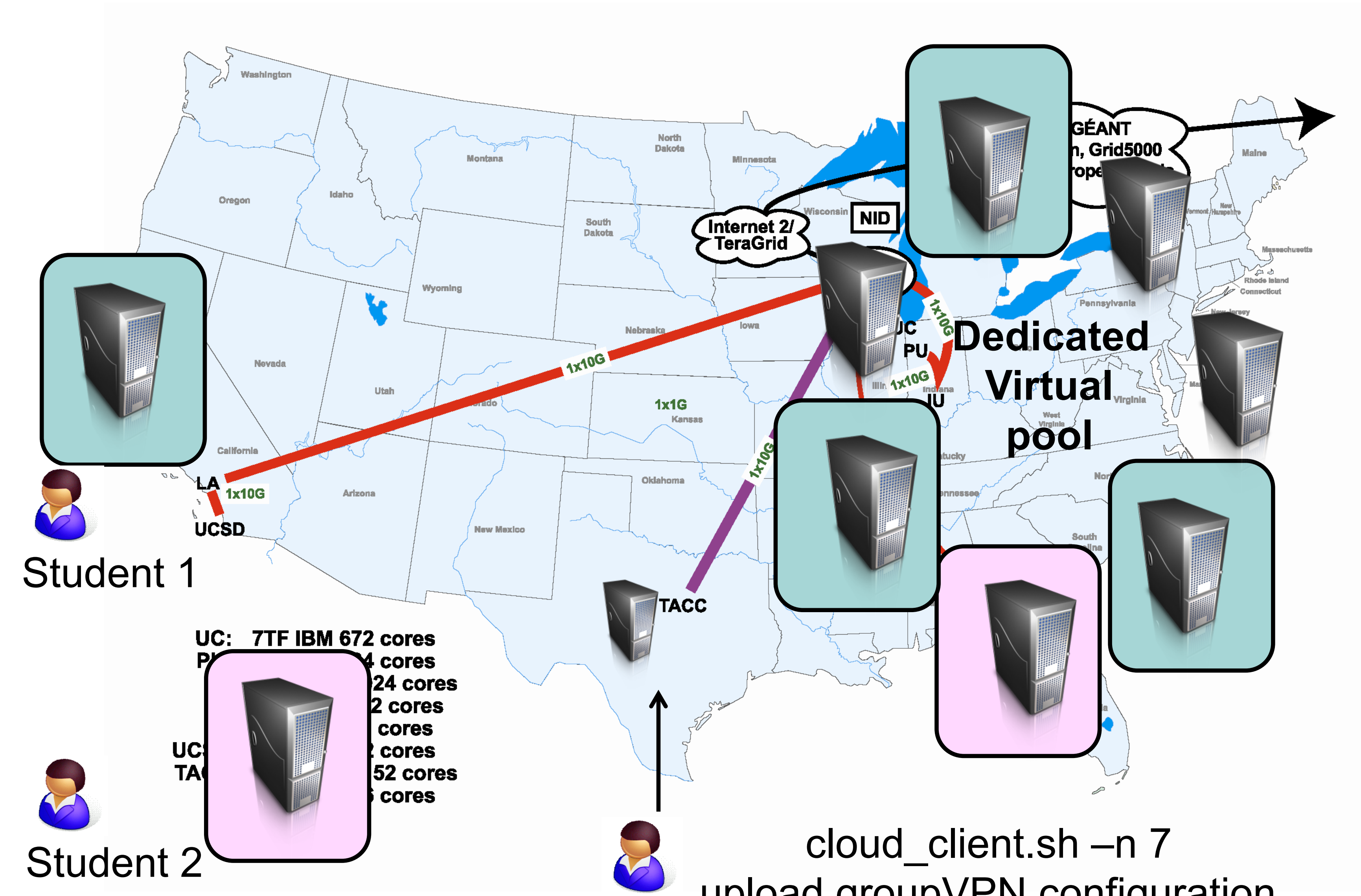
- Allow same logical cluster environment to instantiate on a variety of platforms
  - Local desktop, clusters; FutureGrid; EC2
- Avoid dependence on host environment
  - Make minimum assumptions about VM and provisioning software
    - Desktop: VMware, VirtualBox; KVM
    - Para-virtualized VMs (e.g. Xen) and cloud stacks – need to deal with idiosyncrasies
  - Minimum assumptions about networking
    - Private, NATed Ethernet virtual network interface

Creating private clusters

- The default ‘playground’ environment allows new users to quickly get started
- Users and instructors can also deploy their own private clusters
  - The Condor pool becomes a dedicated resource
- Same appliance – what changes is a configuration file that specifies which virtual cluster to connect to
- Web interface to create groups

Web site – GroupVPN

Deploying private virtual pools



cloud_client.sh –n 7

upload groupVPN configuration

Summary

- Hands-on experience with clusters is essential for education and training
- Virtualization, clouds simplify software packaging/configuration
- Grid appliance allows users to easily deploy hands-on virtual clusters
- FutureGrid provides resources and cloud stacks for educators to easily deploy their own virtual clusters
- Towards a community-based marketplace of educational appliances

Thank you!

- More information:
  - http://www.futuregrid.org
  - http://grid-appliance.org

- This document was developed with support from the National Science Foundation (NSF) under Grant No. 0910812 to Indiana University for "FutureGrid: An Experimental, High-Performance Grid Test-bed." Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF