Richard Gass
Intel Labs Pittsburgh

Open Cirrus ™ : Tutorial Part I

Development Challenges in Open Cirrus

Taking advantage of the Cloud

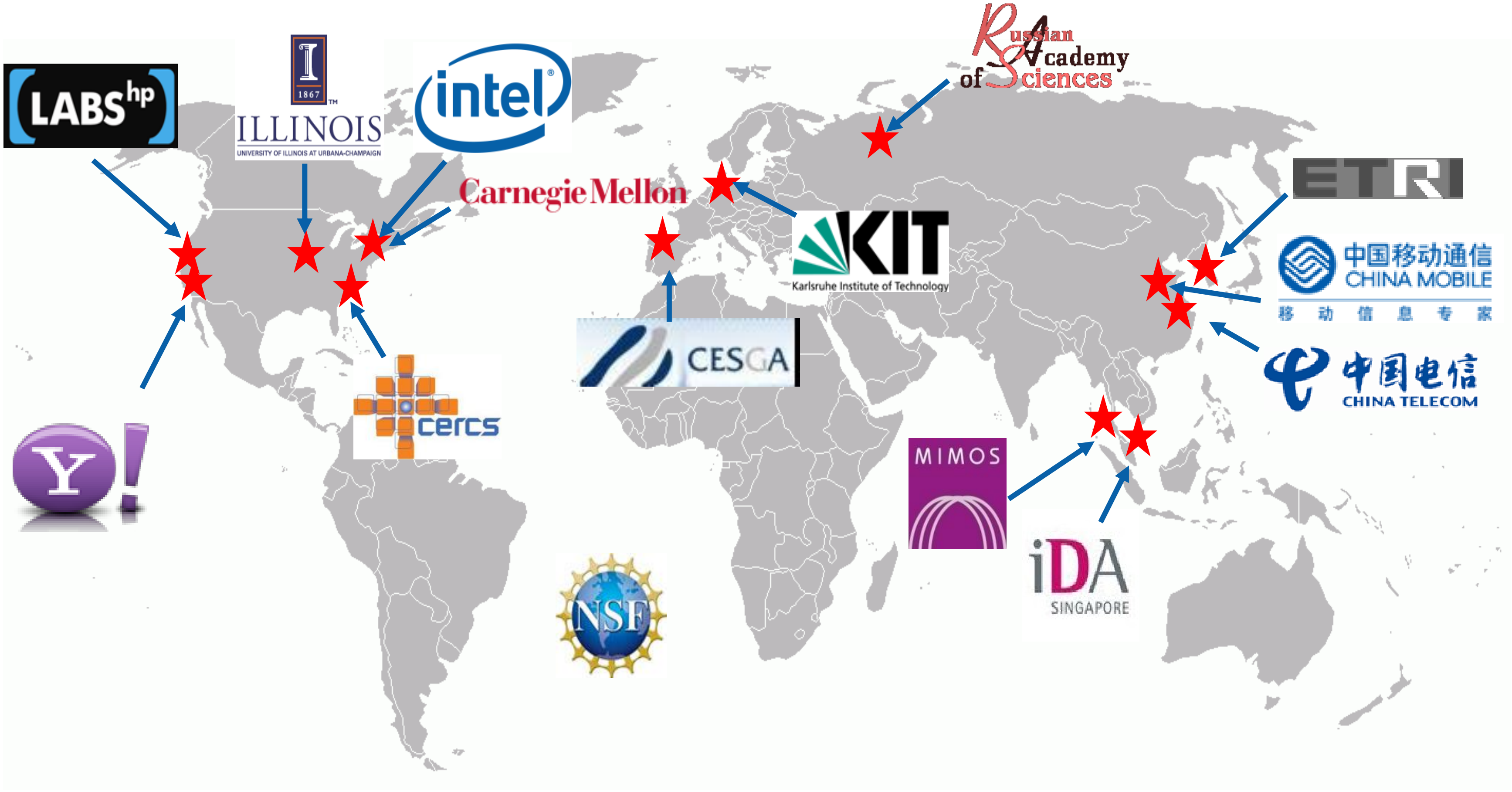
Research in the cloud

- High entry cost to do research in the cloud
- Making the cloud better
  - More efficient
  - Greener

What is Open Cirrus

- Global testbed for cloud computing research
- Sponsored by HP, Intel, and Yahoo! (w/additional support from NSF)
- Launched Sep 2008 with 6 sites
  - 14 sites worldwide today
  - target of ~20
- Each site has 1000-4000 cores
- http://opencirrus.org

Open Cirrus members



Open Cirrus Context

Goals

1. Catalyze **open-source stack** and APIs for the cloud

2. Foster new **systems and services research** around cloud computing

Motivation

- Enable more tier-2 and tier-3 public and private cloud providers

How are we different?

- Support for systems research and applications research
  - Access to bare metal, integrated virtual-physical migration
- Federation of heterogeneous datacenters (eventually)

Cloud Computing Project Philosophy

Research

- Identify needed technology for a developing field (storage, power)

Learn by Doing

- Manage a “production” compute cluster

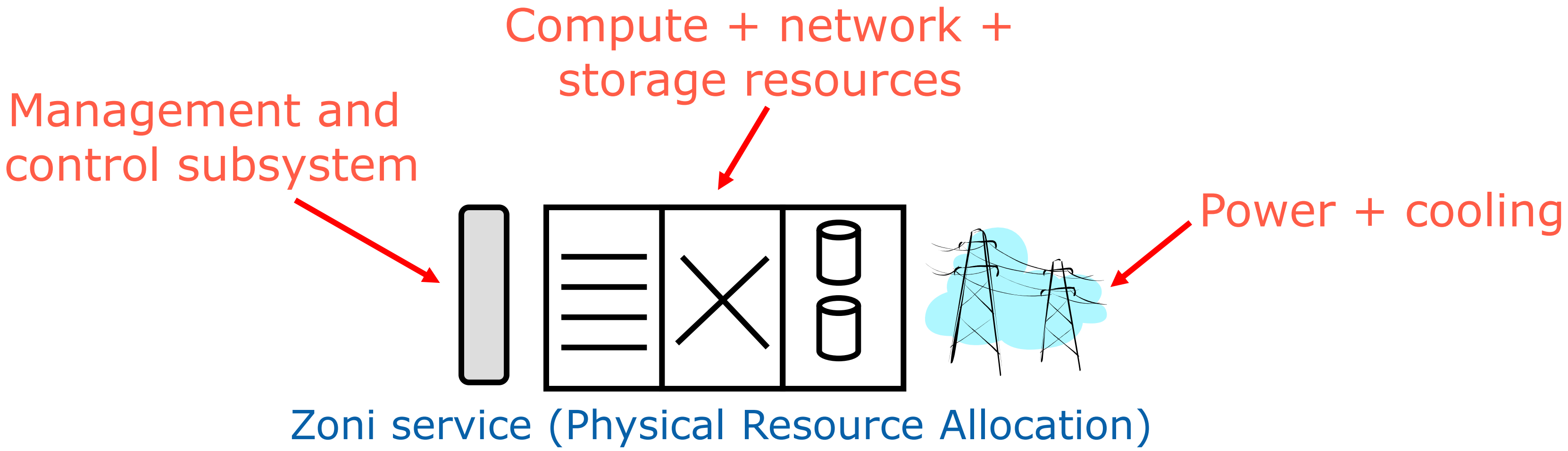
Collaborate

- Share the experiences and costs of developing for cloud computing

Build community

- Leverage open source artifacts
- Contribute artifacts back

Open Cirrus Site Model



Compute + network + storage resources

Management and control subsystem

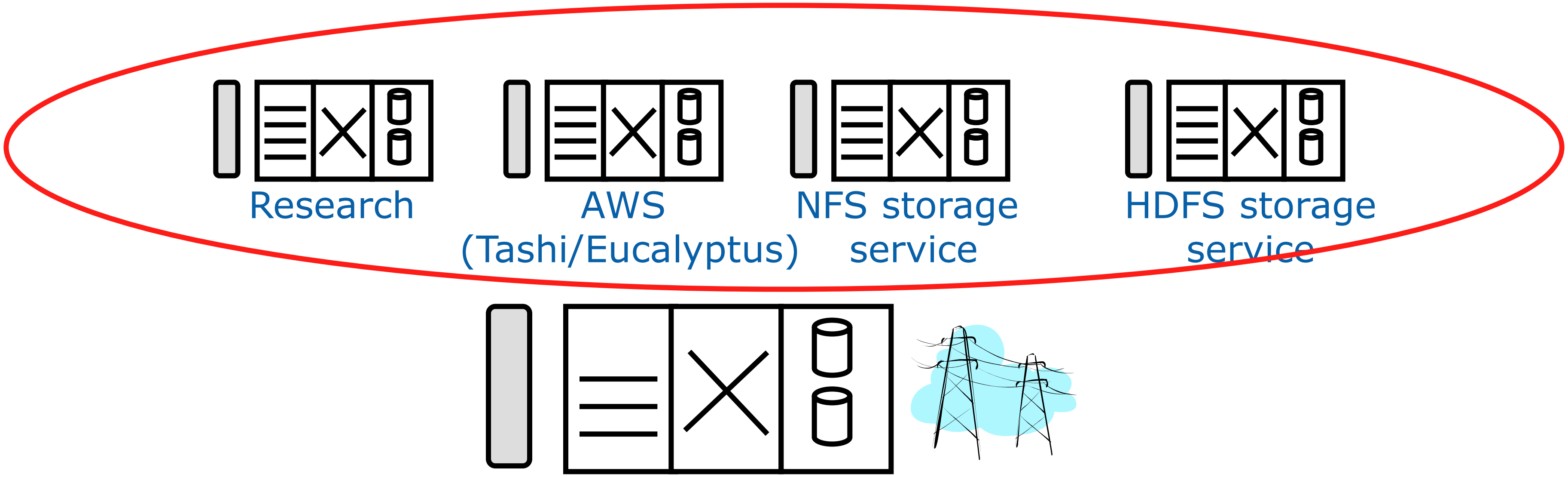
Power + cooling

Zoni service (Physical Resource Allocation)

Credit: John Wilkes (HP)

Open Cirrus Site Model

Zoni clients, each with their
own "physical data center"



Research

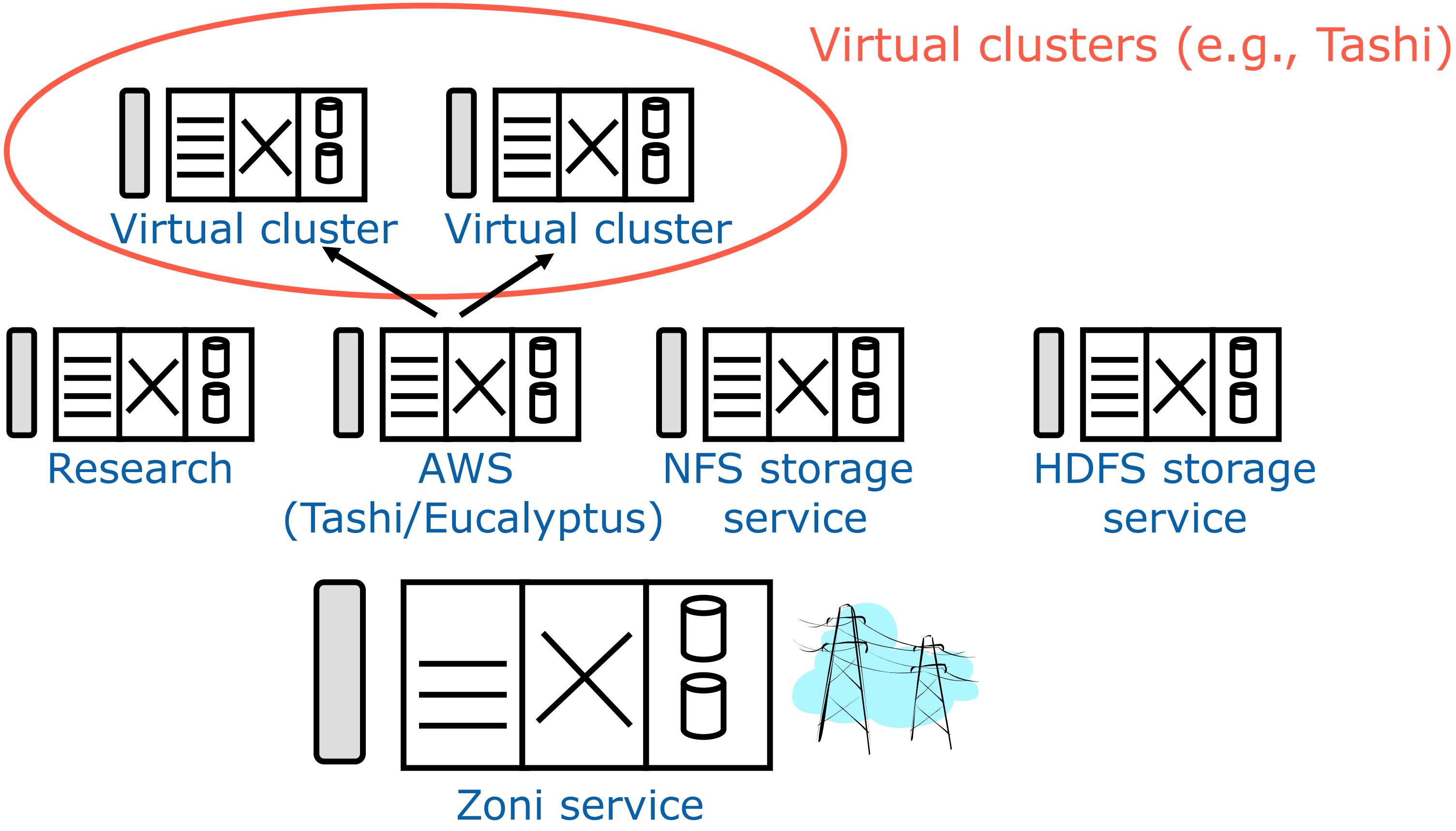
AWS
(Tashi/Eucalyptus)

NFS storage
service

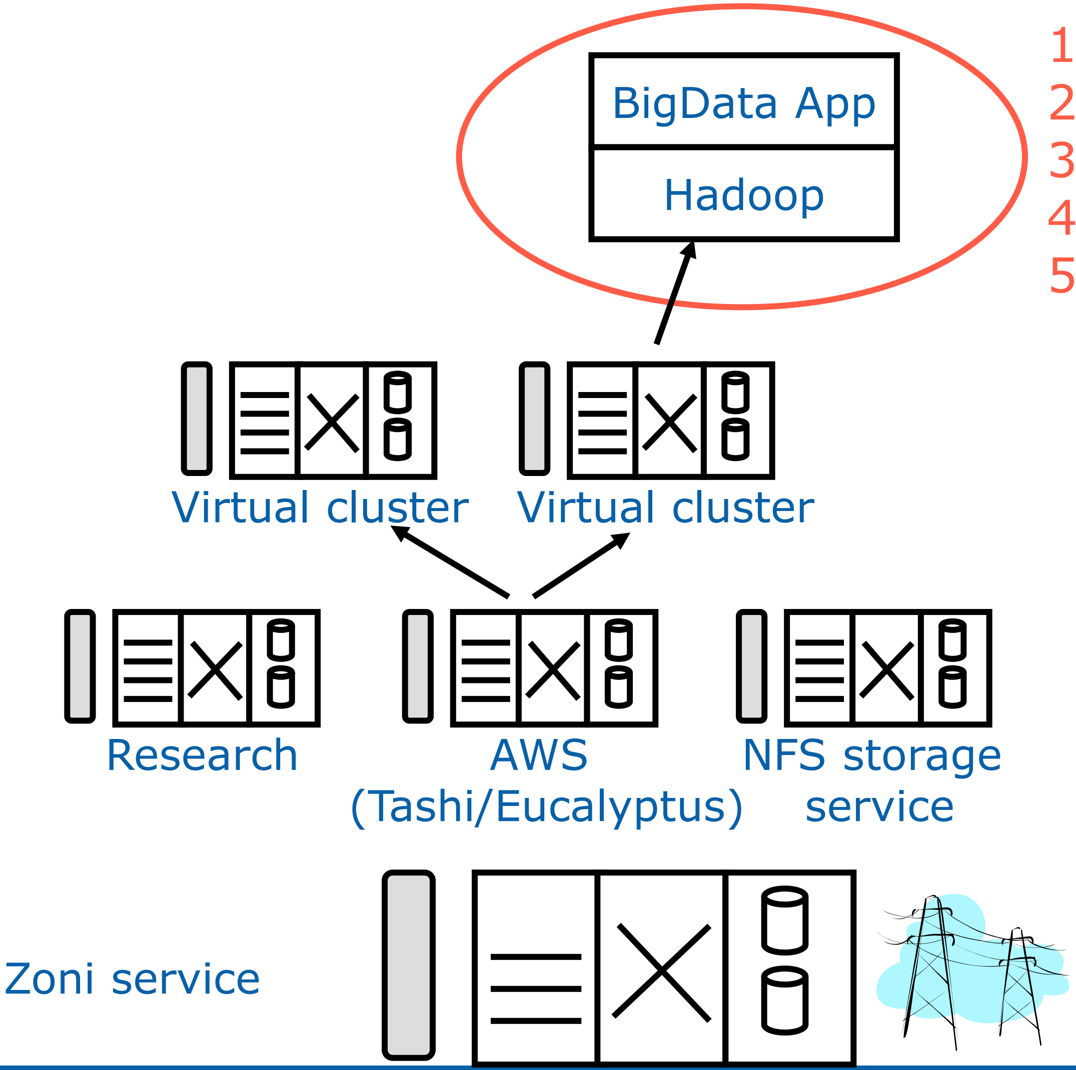
HDFS storage
service

Zoni service

Open Cirrus Site Model

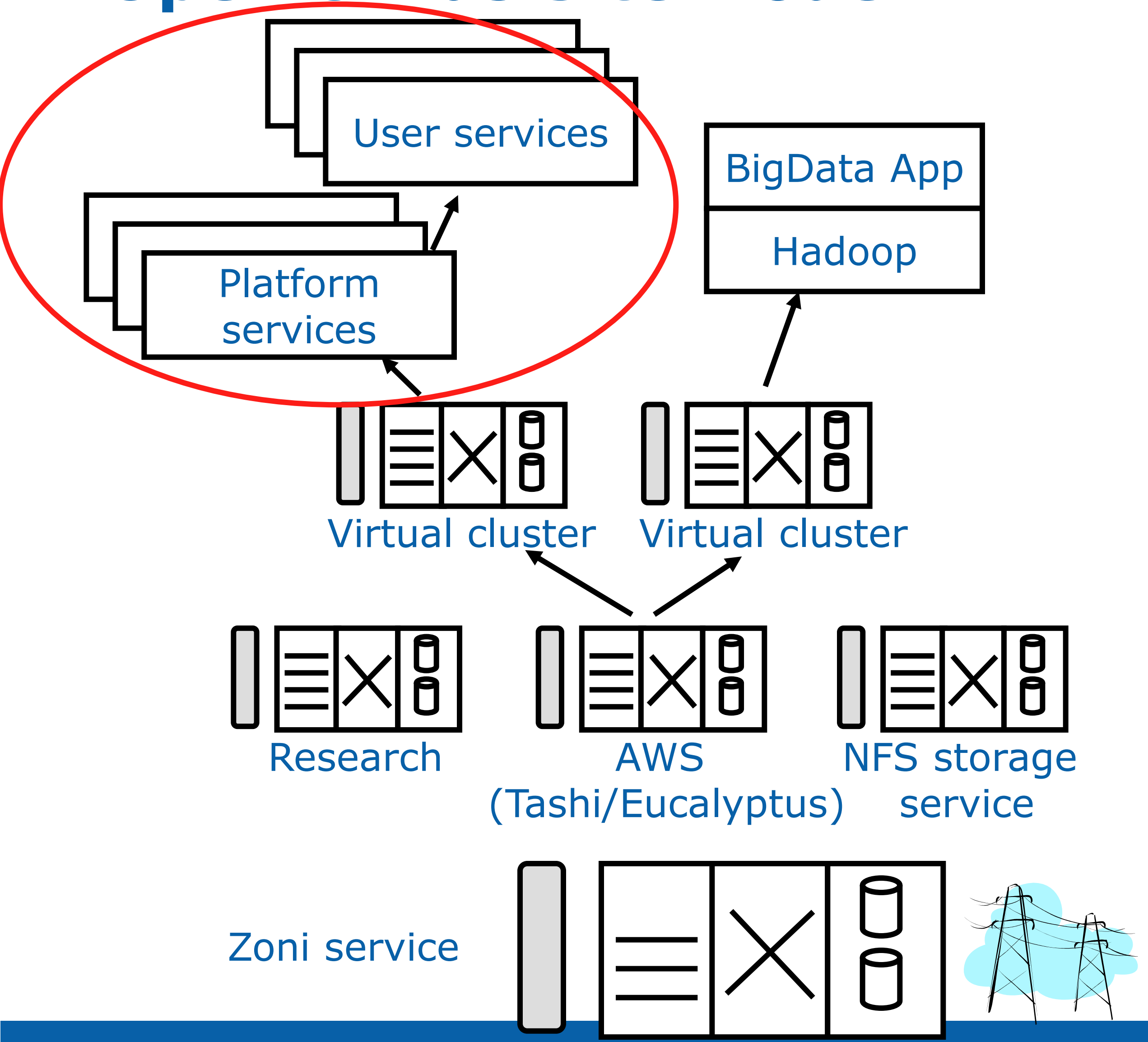


Open Cirrus Site Model



1. Application running
2. On Hadoop
3. On Tashi virtual cluster
4. On a Zoni allocation unit
5. On real hardware

Open Cirrus Site Model



User services

Platform services

BigData App

Hadoop

Virtual cluster

Virtual cluster

Research

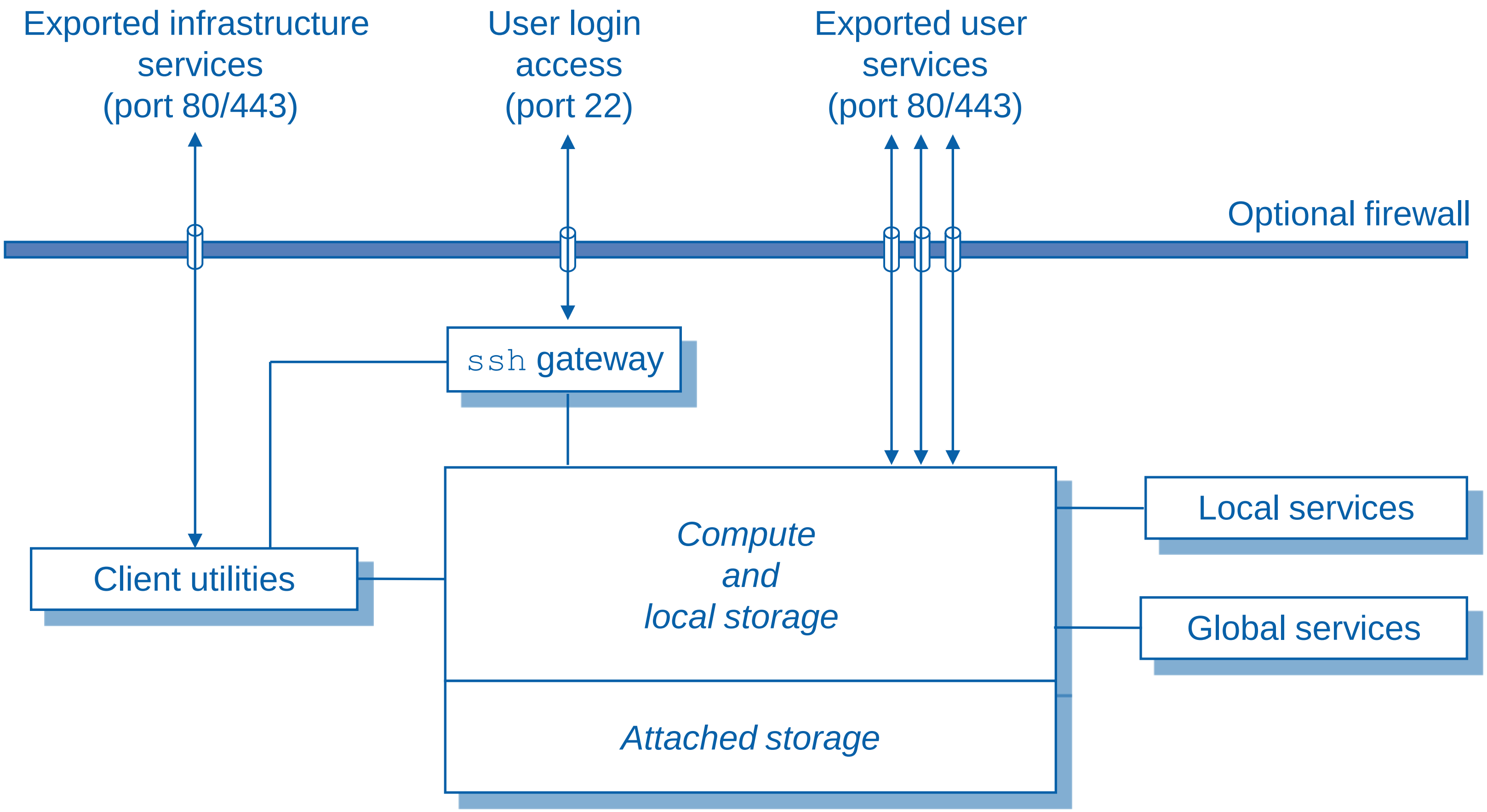
AWS (Tashi/Eucalyptus)

NFS storage service

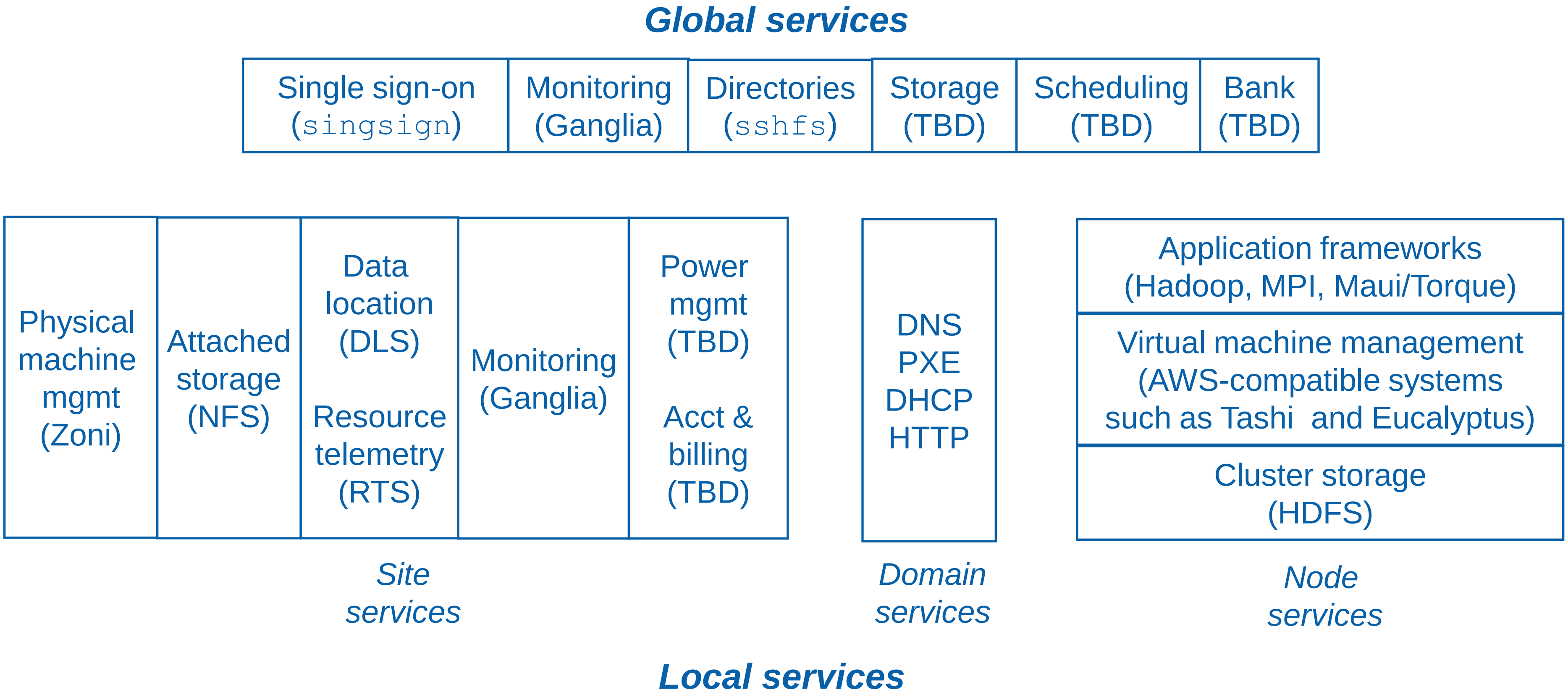
HDFS storage service

Zoni service

Model of an Open Cirrus Site


Exported infrastructure services (port 80/443)
User login access (port 22)
Exported user services (port 80/443)
Optional firewall
ssh gateway
Client utilities
Compute and local storage
Attached storage
Local services
Global services

The Open Cirrus Service Stack



Global services

Single sign-on (`singsign`)

Monitoring (Ganglia)

Directories (`sshfs`)

Storage (TBD)

Scheduling (TBD)

Bank (TBD)

Physical machine mgmt (Zoni)

Attached storage (NFS)

Data location (DLS)

Resource telemetry (RTS)

Monitoring (Ganglia)

Power mgmt (TBD)

Acct & billing (TBD)

Site services

DNS
PXE
DHCP
HTTP

Domain services

Application frameworks (Hadoop, MPI, Maui/Torque)

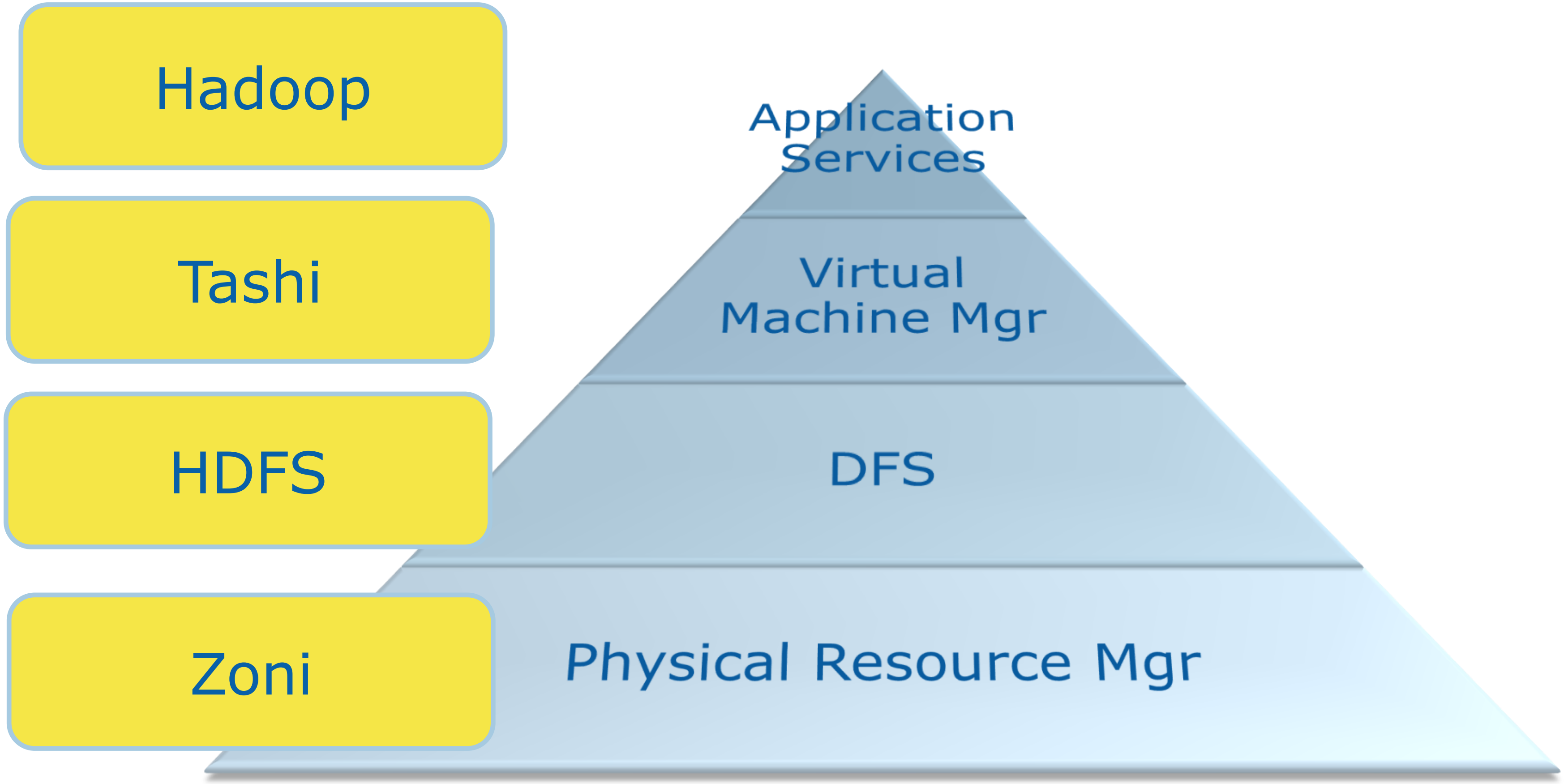
Virtual machine management (AWS-compatible systems such as Tashi and Eucalyptus)

Cluster storage (HDFS)

Node services

Local services

The open source stack



Intel Labs Pittsburgh

One of three Intel labs located near major universities designed to foster industry-university collaborative research.

In 2009, Intel Labs Pittsburgh researchers interacted with ~70 faculty and ~70 students.

Key features:

An Open Collaborative Research Model

Alignment between lab and university interests

Faculty involvement in the lab

Proximity

Intel BigData Cluster

http://opencirrus.intel-research.net

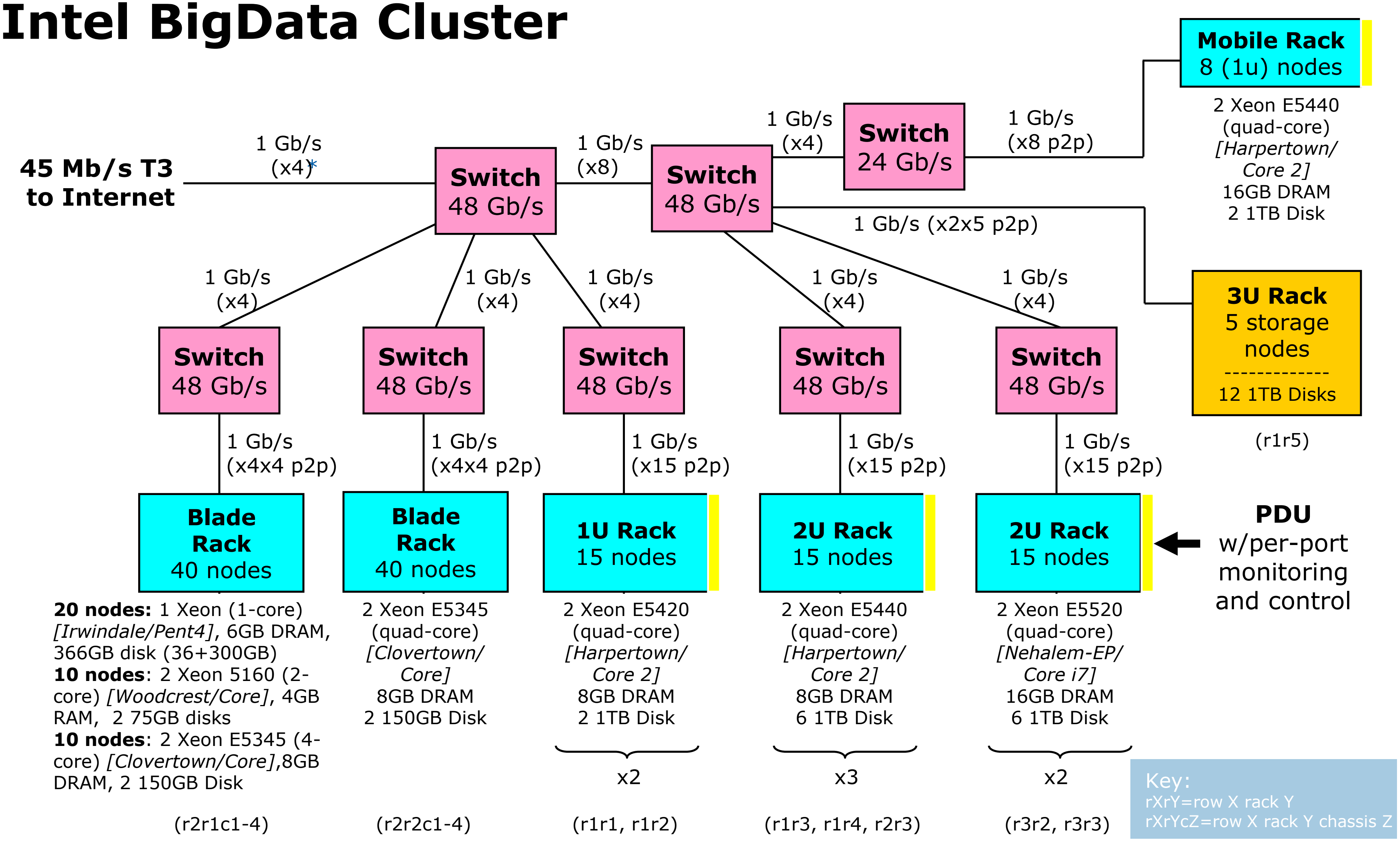
Open Cirrus site hosted by Intel Labs Pittsburgh

- Operational since Jan 2009.
- 200 nodes, 1440 cores, 600 TB disk

Supporting ~75 users, 20 projects

- Intel, CMU, UPitt, Rice, Ga Tech
- *Systems research:*
  - Cluster management, location and power aware scheduling, physical virtual migration (Zoni/Tashi), cache savvy algorithms (Hi-Spade), streaming frameworks (SLIPstream), optical datacenter interconnects (CloudConnect),
- *Applications research:*
  - Programmable matter simulation, online education, realtime brain activity decoding, realtime gesture and object recognition, automated food recognition.

Intel BigData Cluster



45 Mb/s T3 to Internet

1 Gb/s (x4)*

Switch 48 Gb/s

1 Gb/s (x8)

Switch 48 Gb/s

1 Gb/s (x4)

Switch 24 Gb/s

1 Gb/s (x8 p2p)

Mobile Rack
8 (1u) nodes

2 Xeon E5440
(quad-core)
*[Harpertown/
Core 2]*
16GB DRAM
2 1TB Disk

1 Gb/s (x2x5 p2p)

3U Rack
5 storage nodes
----
12 1TB Disks

(r1r5)

1 Gb/s (x4)

Switch 48 Gb/s

1 Gb/s (x4x4 p2p)

Blade Rack
40 nodes

20 nodes: 1 Xeon (1-core) *[Irwindale/Pent4]*, 6GB DRAM, 366GB disk (36+300GB)
10 nodes: 2 Xeon 5160 (2-core) *[Woodcrest/Core]*, 4GB RAM, 2 75GB disks
10 nodes: 2 Xeon E5345 (4-core) *[Clovertown/Core]*,8GB DRAM, 2 150GB Disk

(r2r1c1-4)

1 Gb/s (x4)

Switch 48 Gb/s

1 Gb/s (x4x4 p2p)

Blade Rack
40 nodes

2 Xeon E5345
(quad-core)
*[Clovertown/
Core]*
8GB DRAM
2 150GB Disk

(r2r2c1-4)

1 Gb/s (x4)

Switch 48 Gb/s

1 Gb/s (x15 p2p)

1U Rack
15 nodes

2 Xeon E5420
(quad-core)
*[Harpertown/
Core 2]*
8GB DRAM
2 1TB Disk

x2

(r1r1, r1r2)

1 Gb/s (x4)

Switch 48 Gb/s

1 Gb/s (x15 p2p)

2U Rack
15 nodes

2 Xeon E5440
(quad-core)
*[Harpertown/
Core 2]*
8GB DRAM
6 1TB Disk

x3

(r1r3, r1r4, r2r3)

1 Gb/s (x4)

Switch 48 Gb/s

1 Gb/s (x15 p2p)

2U Rack
15 nodes

2 Xeon E5520
(quad-core)
*[Nehalem-EP/
Core i7]*
16GB DRAM
6 1TB Disk

x2

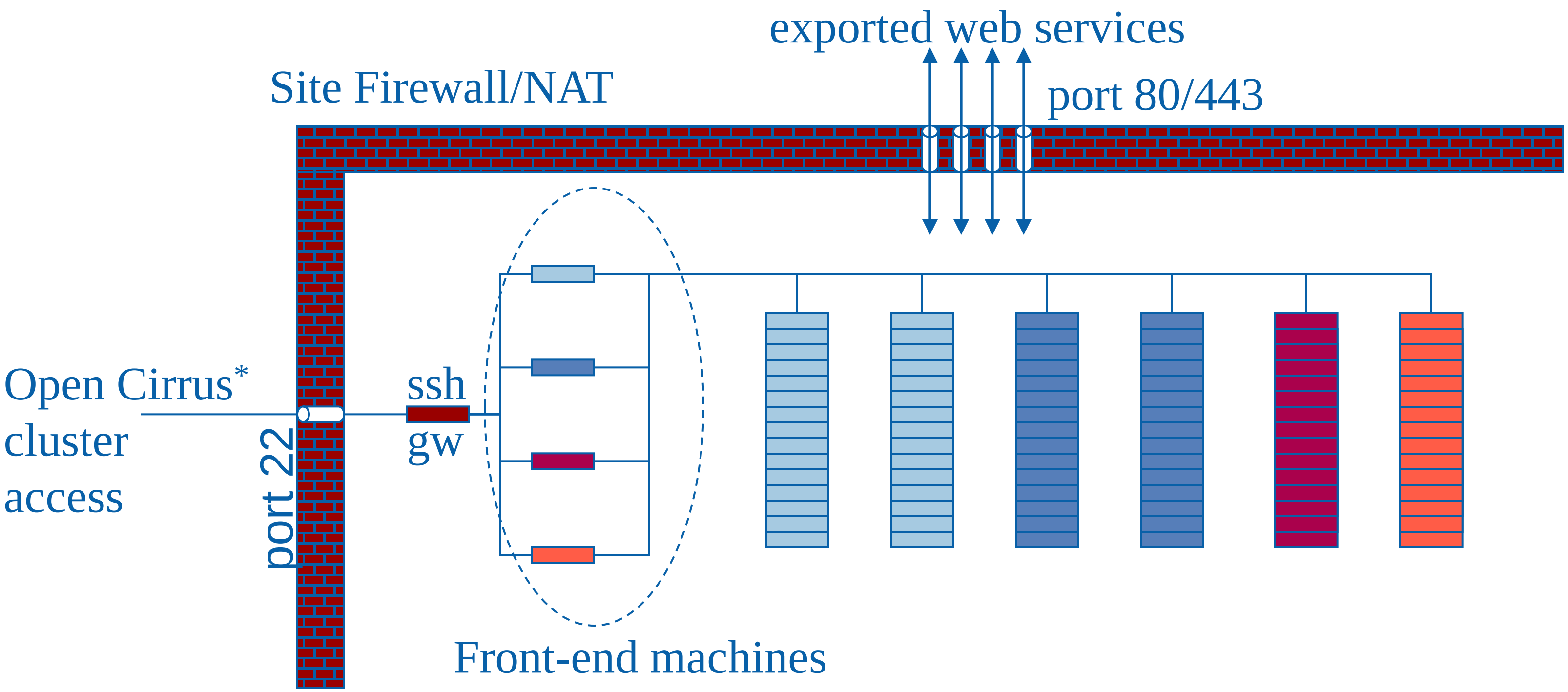
(r3r2, r3r3)

PDU
w/per-port
monitoring
and control

Key:
rXrY=row X rack Y
rXrYcZ=row X rack Y chassis Z

	r2r1c1-4	r2r2c1-4	r1r1 r1r2	r1r3 r1r4 r2r3	r3r2 r3r3	mobile	storage	TOTAL
Nodes	40	40	30	45	30	8	5	**198**
Cores	140	320	240	360	240	64		**1364**
DRAM (GB)	240	320	240	360	480	128		**1768**
Spindles	80	80	60	270	180	16	60	**746**
Storage (TB)	12	12	60	270	180	16	60	**610**

Access Model



Getting access to the Open Cirrus cluster

- Request an account
  - http://opencirrus.intel-research.net/access.html
  - Identify the PI for your project
  - Fill out project proposal and user account forms
- Export Control (48 hours max)
- Intel IT (1 weeks max)
- Account created

Key Services

- Tashi: Primarily a VM instantiation service right now. Particularly useful to users with custom software stacks.
  - E.g. MPI jobs

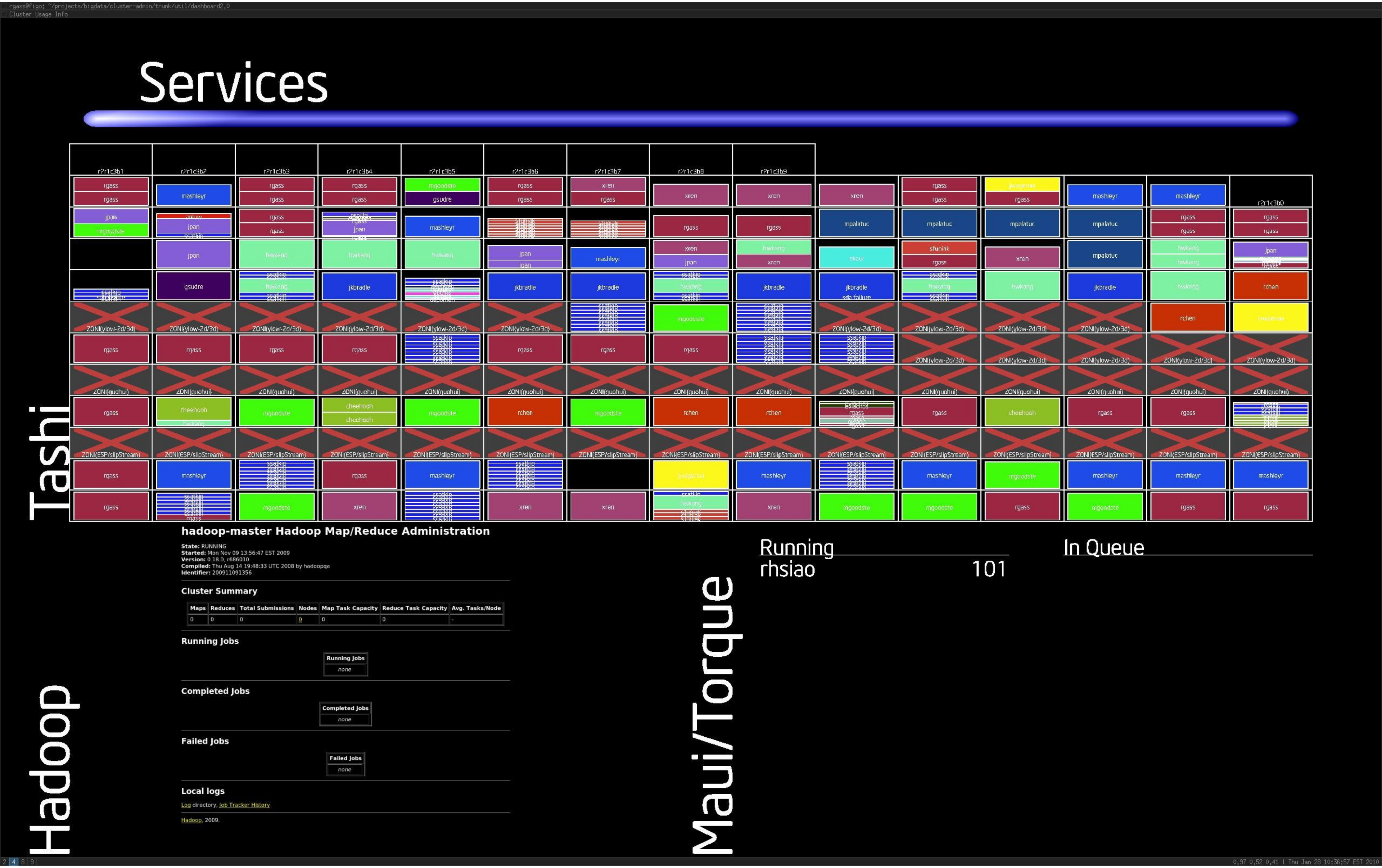
- Maui/Torque: Job submission service. Used primarily by users with “straightforward” applications.
  - E.g. simulation runs

- Hadoop: Service for users who want to experiment with MapReduce.
  - E.g. evaluation of new HDFS management strategies

- Zoni: Used to allocate nodes for special purposes.
  - E.g. power management projects

We actually run the Maui/Torque pool primarily from Tashi.
This enables us to elastically provision Maui/Torque on demand.

Services



rgass@figo: ~/projects/bigdata/cluster-admin/trunk/util/dashboard2.0

Cluster Usage Info

Tashi

r2r1c3b1	r2r1c3b2	r2r1c3b3	r2r1c3b4	r2r1c3b5	r2r1c3b6	r2r1c3b7	r2r1c3b8	r2r1c3b9						r2r1c3b0
rgass rgass	mashleyr	rgass rgass	rgass rgass	mgoodste gsudre	rgass rgass	xren rgass	xren	xren	xren	rgass rgass	jwagenaa rgass	mashleyr	mashleyr	
jpan mgoodste	anluw jpan ssatkin	rgass rgass	pspillai rgass jpan	mashleyr	sfuniak [illegible]	sfuniak [illegible]	rgass	rgass	mpalatuc	mpalatuc	mpalatuc	mpalatuc	rgass rgass	rgass rgass
	jpan	hwkang	hwkang	hwkang	jpan jpan	mashleyr	xren jpan	hwkang xren	skaul	sfuniak rgass	xren	mpalatuc	hwkang hwkang	jpan [illegible]
ssatkin [illegible]	gsudre	ssatkin hwkang ssatkin	jkbradle	ssatkin [illegible]	jkbradle	jkbradle	ssatkin hwkang ssatkin	jkbradle	jkbradle sda failure	ssatkin hwkang ssatkin	hwkang	jkbradle	hwkang	rchen
ZONI(ylow-2d/3d)	ZONI(ylow-2d/3d)	ZONI(ylow-2d/3d)	ZONI(ylow-2d/3d)	ZONI(ylow-2d/3d)	ZONI(ylow-2d/3d)	ssatkin	mgoodste	ssatkin	ZONI(ylow-2d/3d)	ZONI(ylow-2d/3d)	ZONI(ylow-2d/3d)	ZONI(ylow-2d/3d)	rchen	jwagenaa
rgass	rgass	rgass	rgass	ssatkin	rgass	rgass	rgass	ssatkin	ssatkin	ZONI(ylow-2d/3d)	ZONI(ylow-2d/3d)	ZONI(ylow-2d/3d)	ZONI(ylow-2d/3d)	ZONI(ylow-2d/3d)
ZONI(guohui)	ZONI(guohui)	ZONI(guohui)	ZONI(guohui)	ZONI(guohui)	ZONI(guohui)	ZONI(guohui)	ZONI(guohui)	ZONI(guohui)	ZONI(guohui)	ZONI(guohui)	ZONI(guohui)	ZONI(guohui)	ZONI(guohui)	ZONI(guohui)
rgass	cheehooh hwkang	mgoodste	cheehooh cheehooh	mgoodste	rchen	mgoodste	rchen	rchen	[illegible] rgass [illegible]	rgass	cheehooh	rgass	rgass	[illegible]
ZONI(ESP/slipStream)	ZONI(ESP/slipStream)	ZONI(ESP/slipStream)	ZONI(ESP/slipStream)	ZONI(ESP/slipStream)	ZONI(ESP/slipStream)	ZONI(ESP/slipStream)	ZONI(ESP/slipStream)	ZONI(ESP/slipStream)	ZONI(ESP/slipStream)	ZONI(ESP/slipStream)	ZONI(ESP/slipStream)	ZONI(ESP/slipStream)	ZONI(ESP/slipStream)	ZONI(ESP/slipStream)
rgass	mashleyr	ssatkin	rgass	mashleyr	ssatkin		jwagenaa	mashleyr	ssatkin	mashleyr	mgoodste	mashleyr	mashleyr	mashleyr
rgass	ssatkin rgass	mgoodste	xren	ssatkin	xren	xren	ssatkin hwkang sfuniak	xren	mgoodste	mgoodste	rgass	mgoodste	rgass	rgass

Hadoop

hadoop-master Hadoop Map/Reduce Administration

State: RUNNING
Started: Mon Nov 09 13:56:47 EST 2009
Version: 0.18.0, r686010
Compiled: Thu Aug 14 19:48:33 UTC 2008 by hadoopqa
Identifier: 200911091356

Cluster Summary

Maps	Reduces	Total Submissions	Nodes	Map Task Capacity	Reduce Task Capacity	Avg. Tasks/Node
0	0	0	0	0	0	-

Running Jobs

Running Jobs
none

Completed Jobs

Completed Jobs
none

Failed Jobs

Failed Jobs
none

Local logs

Log directory, Job Tracker History

Hadoop, 2009.

Maui/Torque

Running	
rhsiao	101

In Queue

2 4 8 9

0.97 0.52 0.41 | Thu Jan 28 10:36:57 EST 2010

Summary

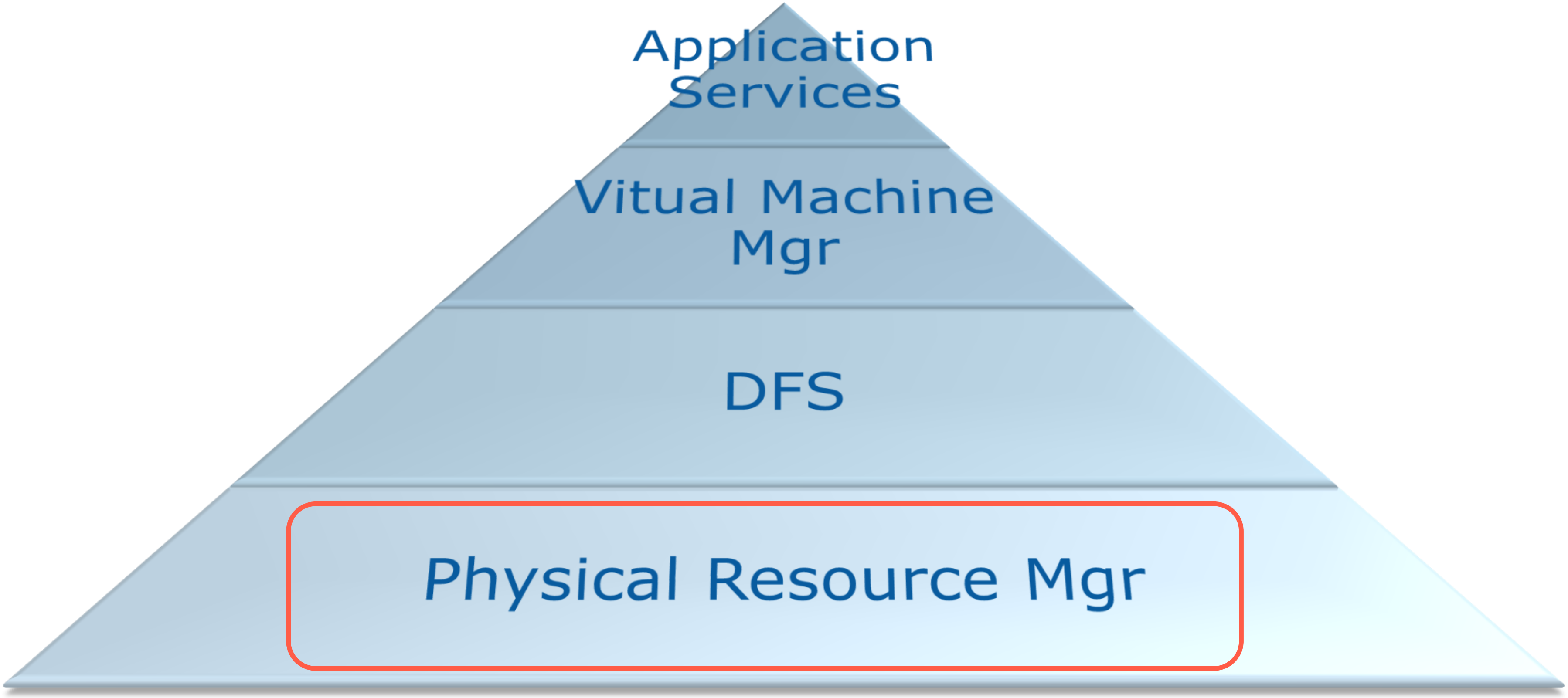
- Open Cirrus is a global research testbed for cloud computing
- Catalyze system level and application research in the cloud
- Accounts are available for research
  - http://opencirrus.intel-research.net/access.html
- Become an open cirrus member
  - Next summit is in Moscow in June 2011
- Join us for the tutorial!

Richard Gass

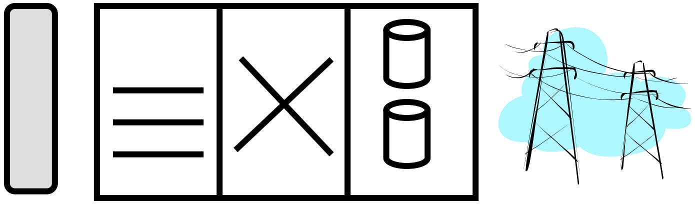
Tutorial : Part I

Open Source Stack/Zoni

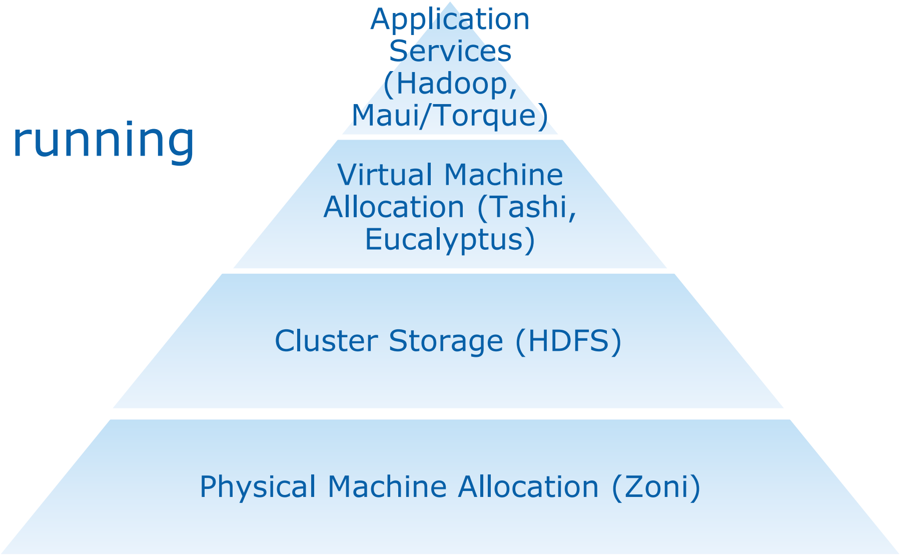
Open source stack



What is Zoni



- A foundation service for the Open Cirrus software stack
- Bootstraps and manages system resources for cloud computing infrastructures
- Enables partitioning of clusters into isolated domains of physical resources (mini-clusters)
- Manages system allocations
- Zoni is transparent once systems are running



Application Services (Hadoop, Maui/Torque)
Virtual Machine Allocation (Tashi, Eucalyptus)
Cluster Storage (HDFS)
Physical Machine Allocation (Zoni)

Zoni Functionality

- Isolation
  - Allow multiple mini-clusters to co-exist without interference
- Allocation
  - Assignment of physical resources to users
- Provisioning
  - Installation or booting of specified OS
- Debugging
  - Out of band access to systems (console access)
- Management
  - Out of band mangement (Remote power control)

Why do I need Zoni

- Eases system administration
  - 1 admin – 200 systems
- Allows rapid provisioning
  - Domains/Pools/switch configurations
  - Usable system setup within minutes
- Isolates systems
  - Multiple groups can coexist without interfering with each other
- Allows access to bare metal
  - Share physical hardware (GPUs)
- Zoni is an Intel open source project hosted by the Apache incubator
  - Your request for changes can get into the code
  - I am very motivated to work with you

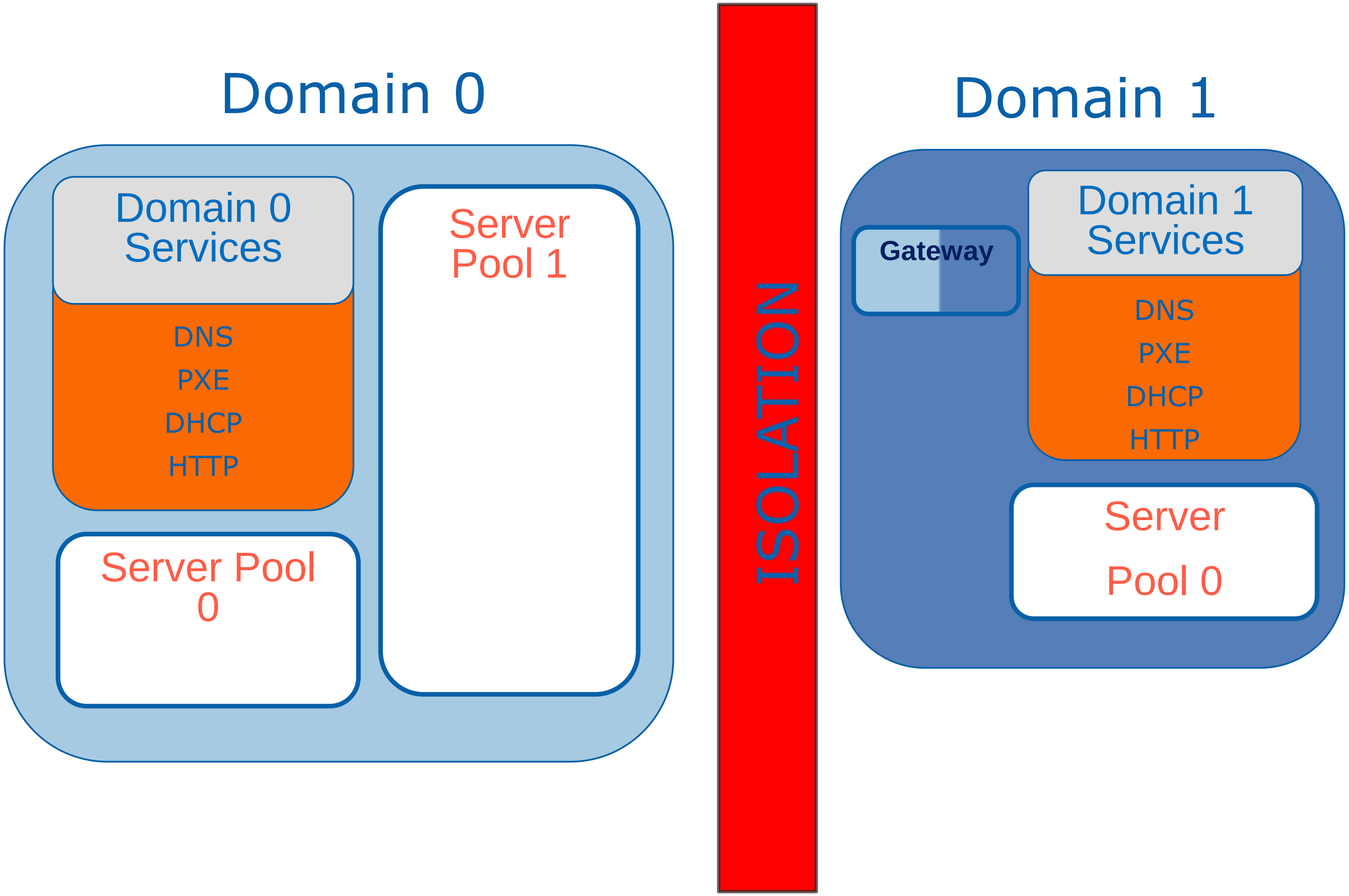
Zoni Goals

- Reduce complexity in allocating physical resources
  - Allow running without virtualization overhead
- Enable systems research in this space
- Provide isolated mini-datacenters to users
- Show users that we can efficiently allocate/deallocate resources and gain user confidence
  - Stop system squatting
    - Incentives
      - HP’s tycoon (economic model)
      - Simple points scheme for good behavior or early return

Zoni Components

- DHCP Server
- PXE Server
- HTTP Server
- Image Store (NFS)
- DNS Server (optional)
- Configurable switches (Layer 2, VLAN support)
- Remote hardware access method allowing management and debugging
  - IPMI /iLO/DRAC
  - IP-addressable PDUs

Zoni Domains



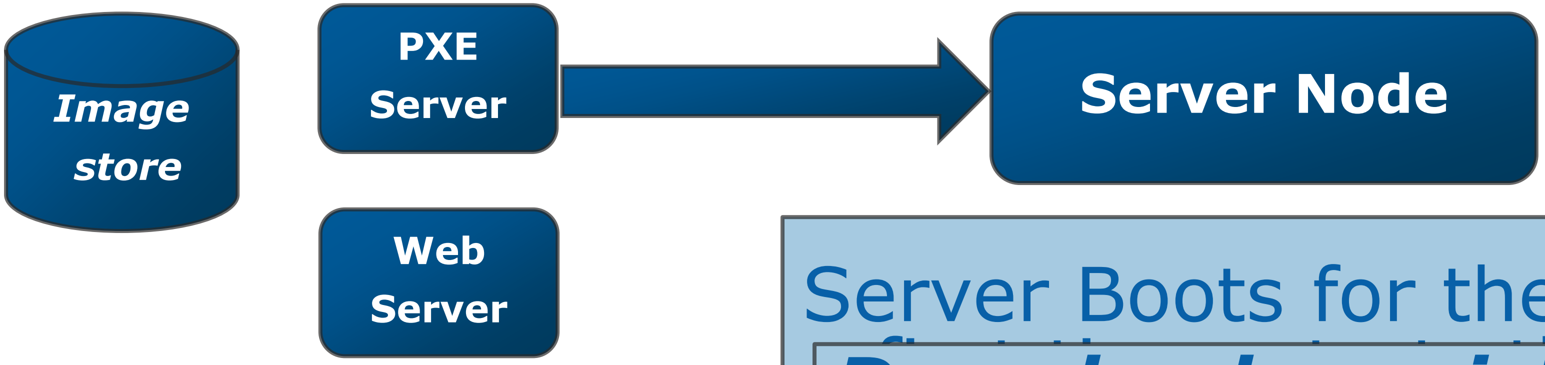
Zoni node registration

- Gather unique identifier from system
  - Mac Address / Dell Tag
- Assign hostname (r1r2u24)
- Assign Switch/PDU port info

Example

UID	Hostname	Domain0 IP	ImageName	Switch	PDU
J3GPGD	r1r2u24	172.16.129.100	tashi_nm	sw0-r1r2:9	pdu0-r1r2:18

Zoni Registration



Server Boots for the

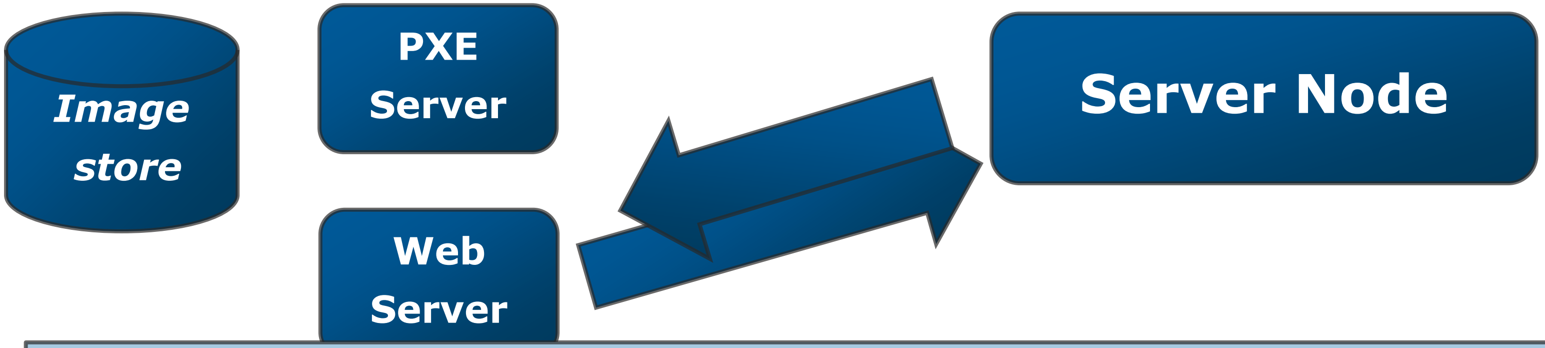
Downloads register kernel and initrd from pxe server

1. Gather unique identifier from sy
   1. Mac Address / Dell Tag
2. ***Assign hostname (r1r2u24)***
3. Get switch/pdu info

Example

J3GPGD r1r2u24 172.16.129.100 tashi_nm sw0-r1r2:9 pdu0-r1r2:18

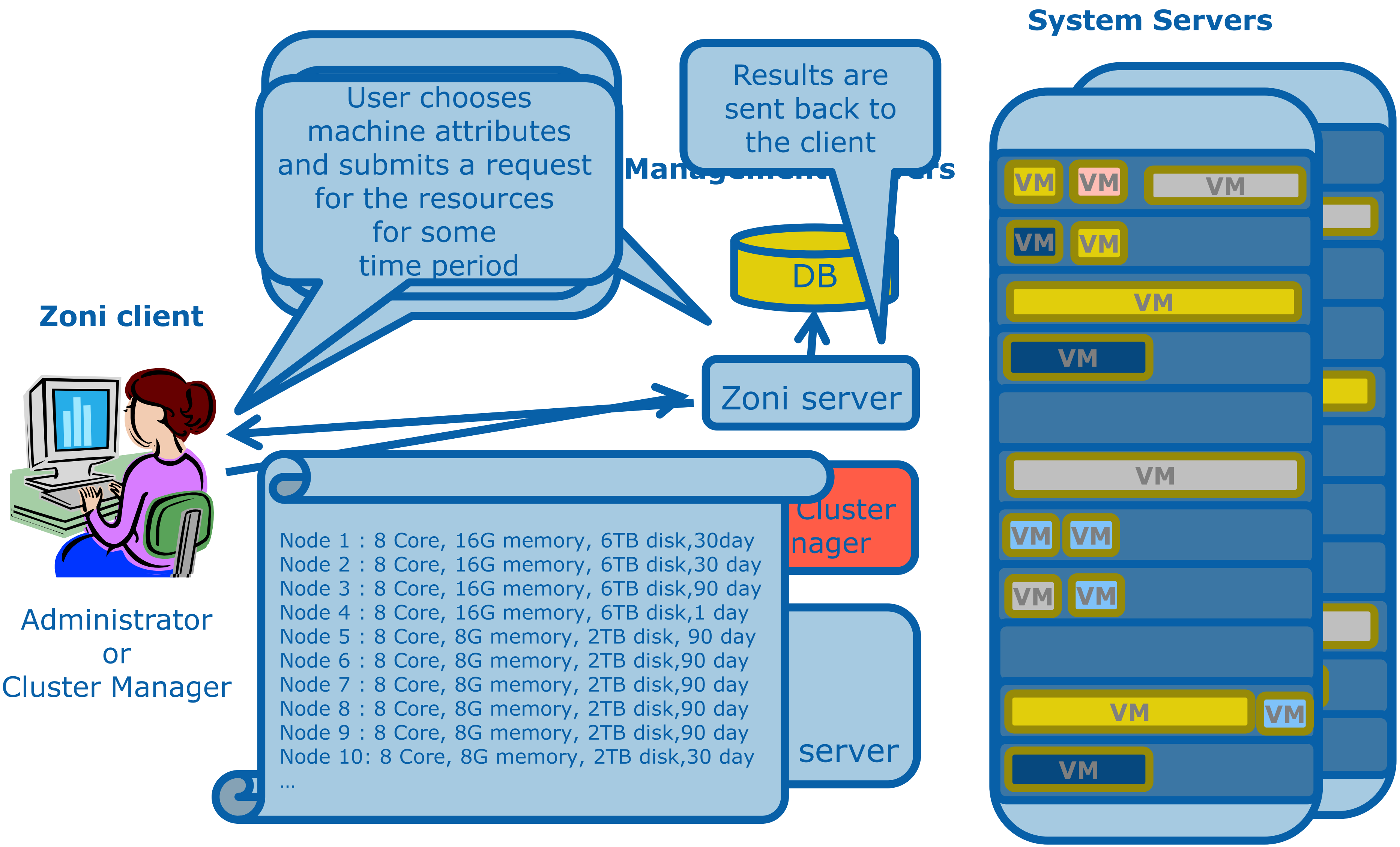
Zoni Registration



Register_node scrapes for system information and populates Zoni database

Final Server Prep

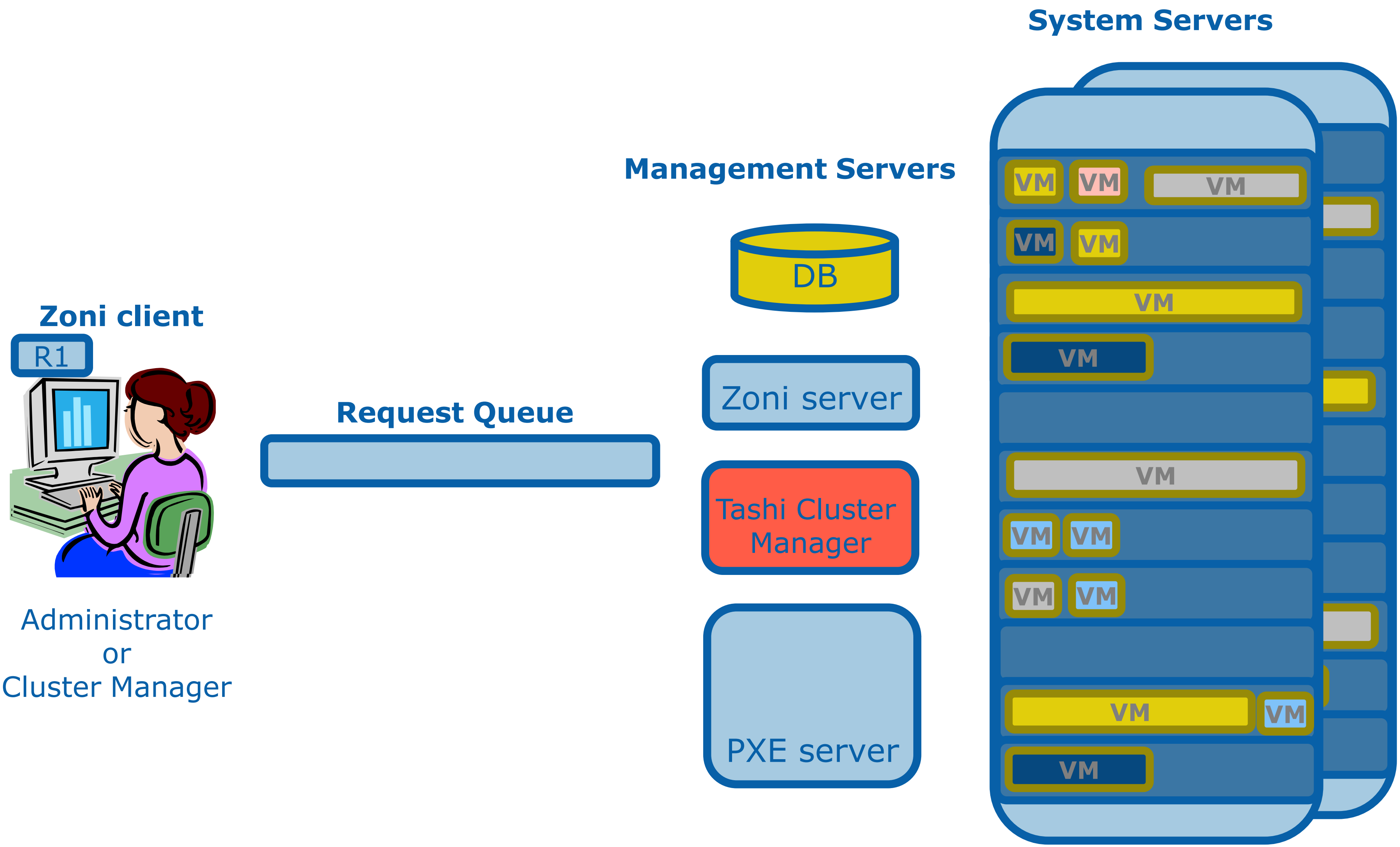
- ***Wipe disks***
- Configure IPMI (IP/admin accounts)
- ***Register node with DNS/DHCP***
- Assign image
- ***Reboot***



User chooses machine attributes and submits a request for the resources for some time period

Results are sent back to the client

Node 1 : 8 Core, 16G memory, 6TB disk,30day
Node 2 : 8 Core, 16G memory, 6TB disk,30 day
Node 3 : 8 Core, 16G memory, 6TB disk,90 day
Node 4 : 8 Core, 16G memory, 6TB disk,1 day
Node 5 : 8 Core, 8G memory, 2TB disk, 90 day
Node 6 : 8 Core, 8G memory, 2TB disk,90 day
Node 7 : 8 Core, 8G memory, 2TB disk,90 day
Node 8 : 8 Core, 8G memory, 2TB disk,90 day
Node 9 : 8 Core, 8G memory, 2TB disk,90 day
Node 10: 8 Core, 8G memory, 2TB disk,30 day
…



System Servers

Management Servers

DB

Zoni client

R1

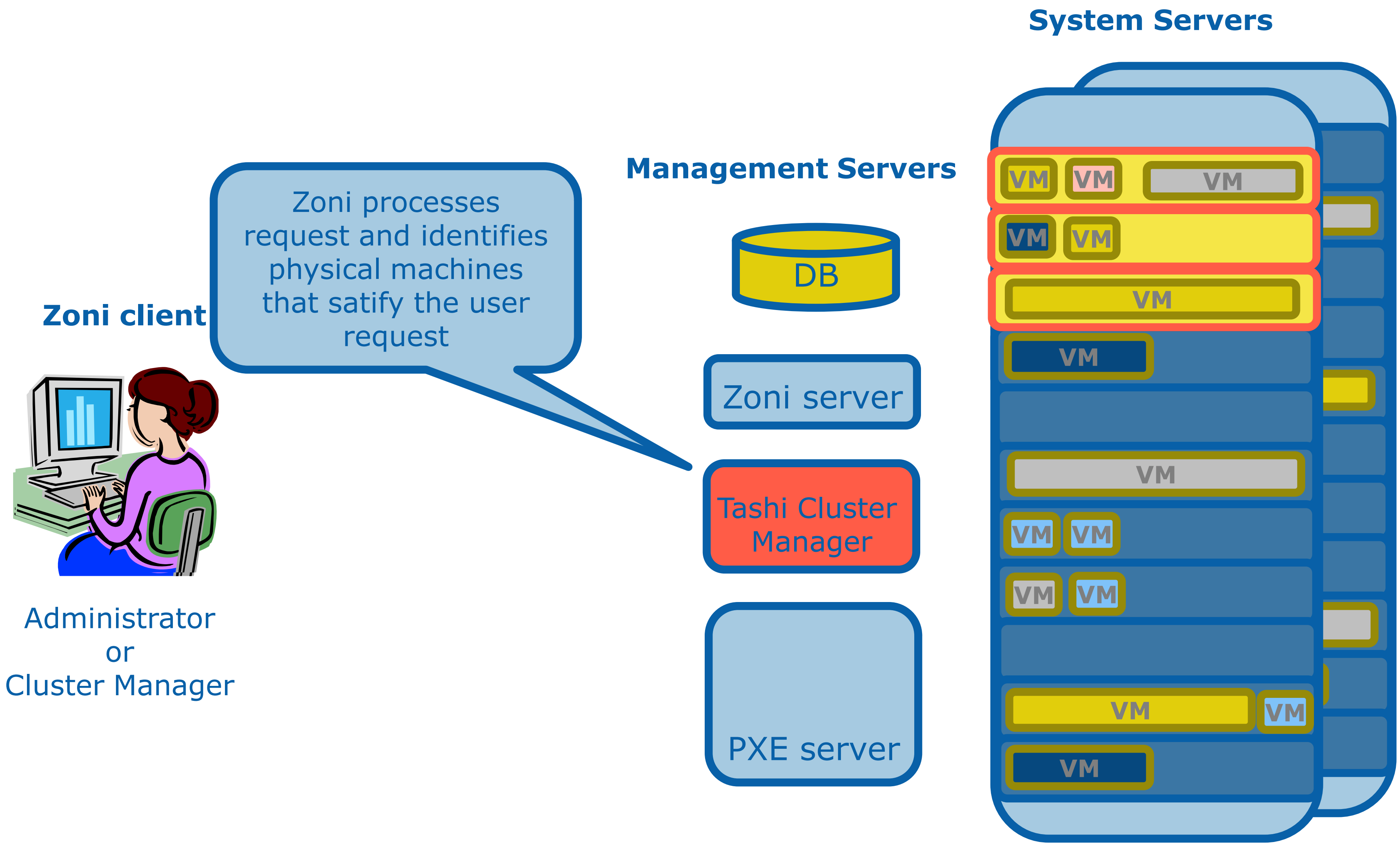
Request Queue

Zoni server

Tashi Cluster Manager

PXE server

Administrator
or
Cluster Manager



System Servers

Management Servers

Zoni client

Zoni processes request and identifies physical machines that satisfy the user request

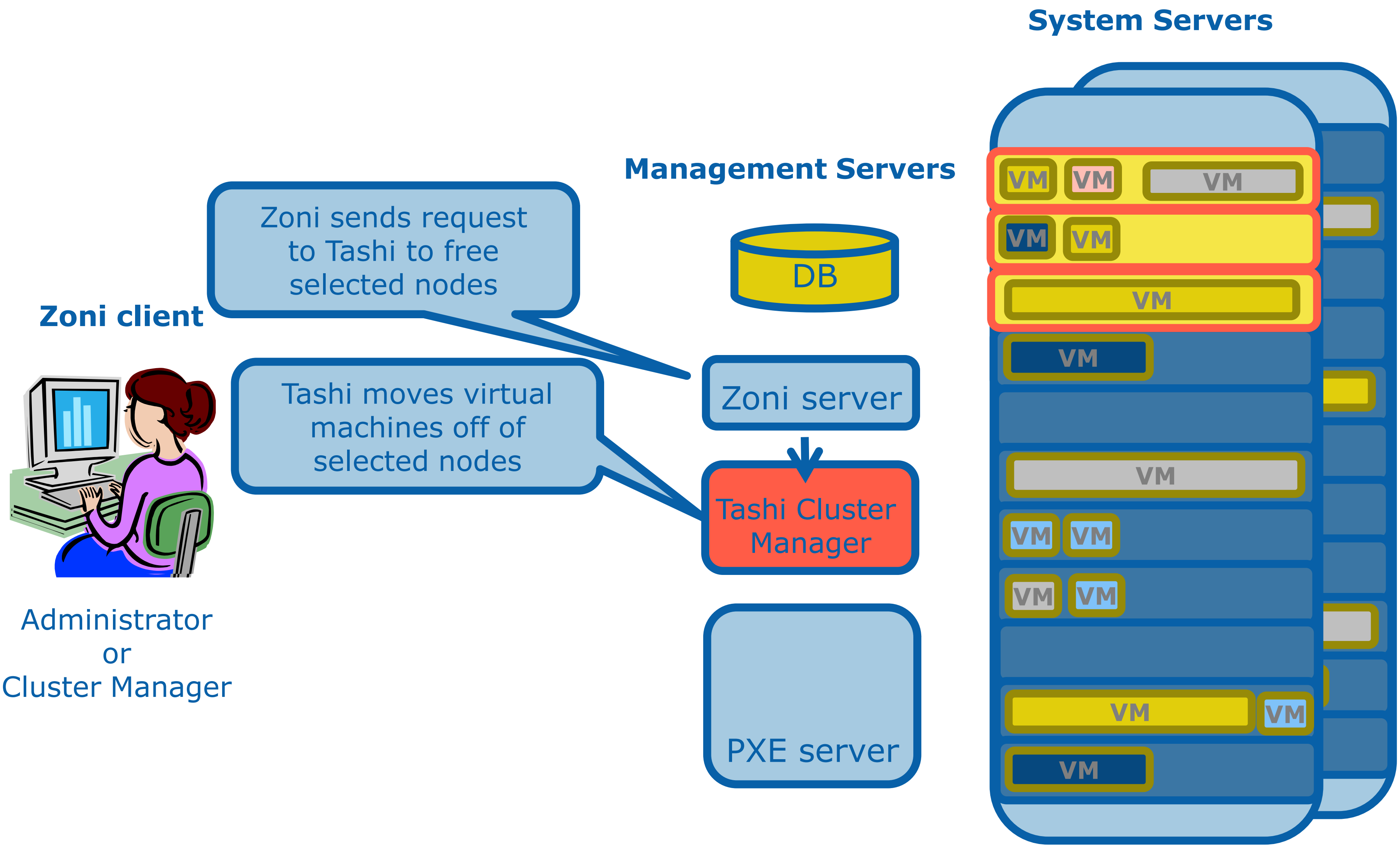
DB

Zoni server

Tashi Cluster Manager

PXE server

Administrator
or
Cluster Manager



Zoni client

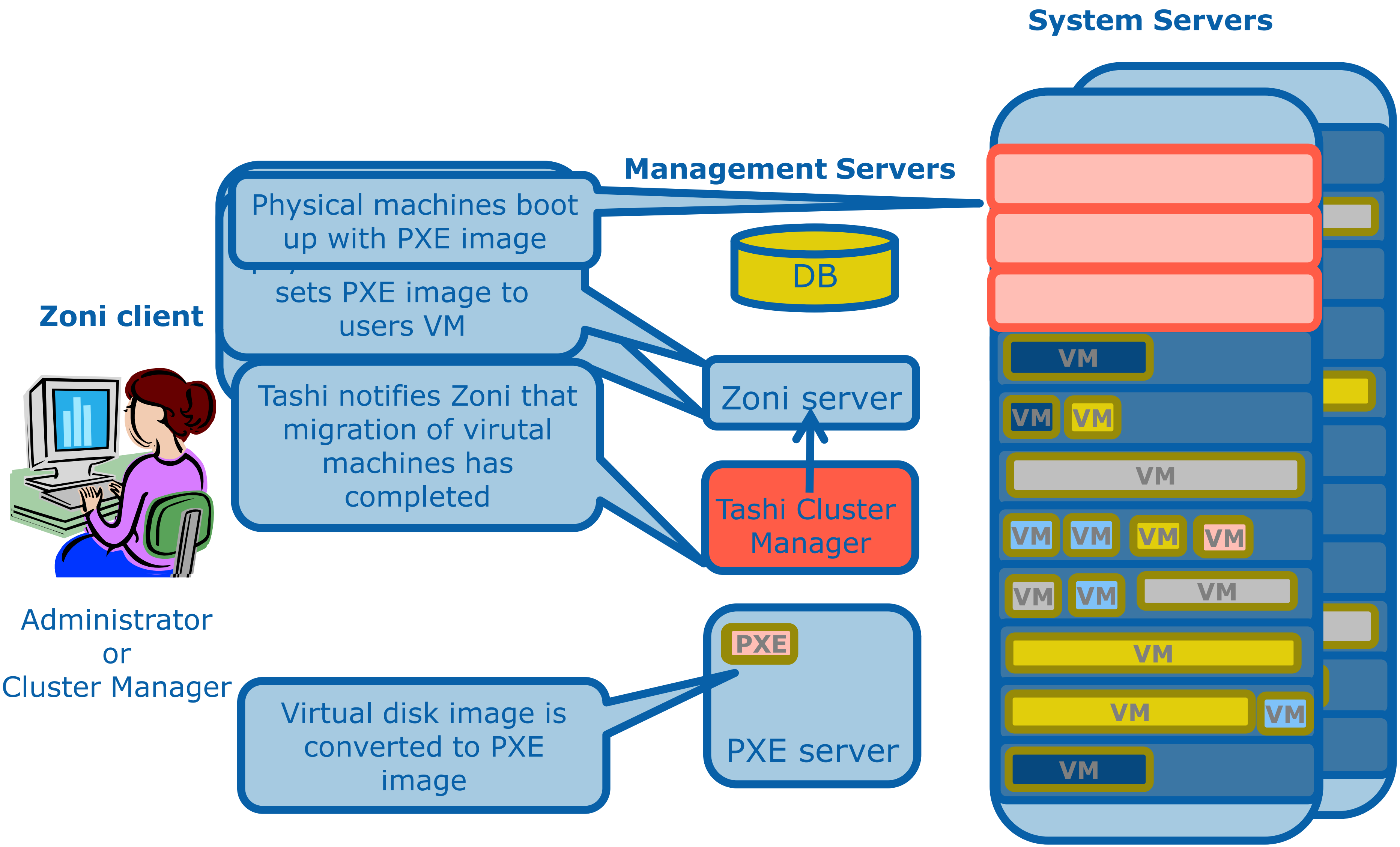
Administrator
or
Cluster Manager

Zoni sends request to Tashi to free selected nodes

Tashi moves virtual machines off of selected nodes

Management Servers

System Servers



System Servers

Management Servers

Physical machines boot up with PXE image

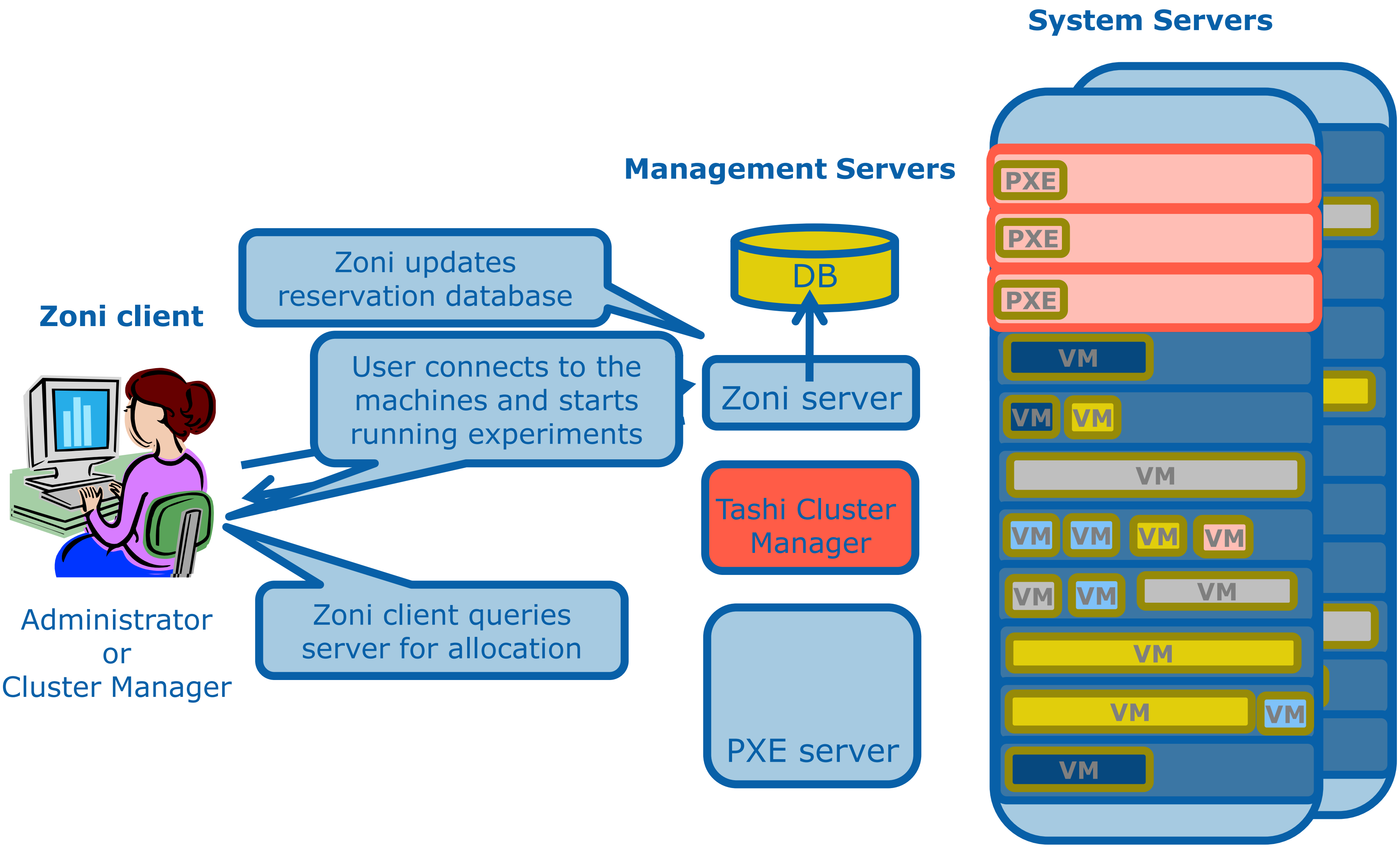
sets PXE image to users VM

Zoni client

Tashi notifies Zoni that migration of virutal machines has completed

Virtual disk image is converted to PXE image

Administrator
or
Cluster Manager



Zoni client

Administrator
or
Cluster Manager

Zoni updates reservation database

User connects to the machines and starts running experiments

Zoni client queries server for allocation

Management Servers

DB

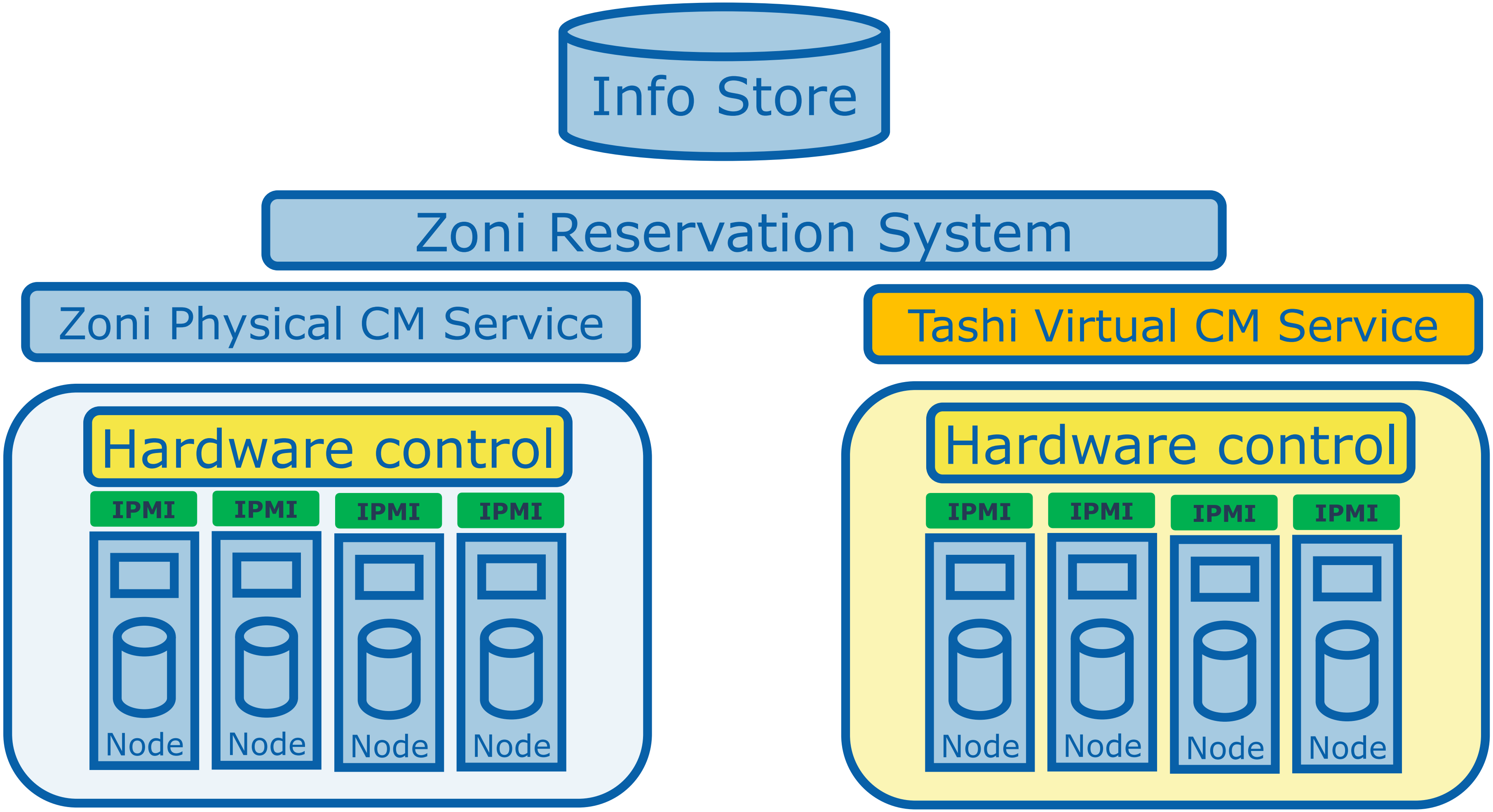
Zoni server

Tashi Cluster Manager

PXE server

System Servers

Physical CM to Virtual CM integration

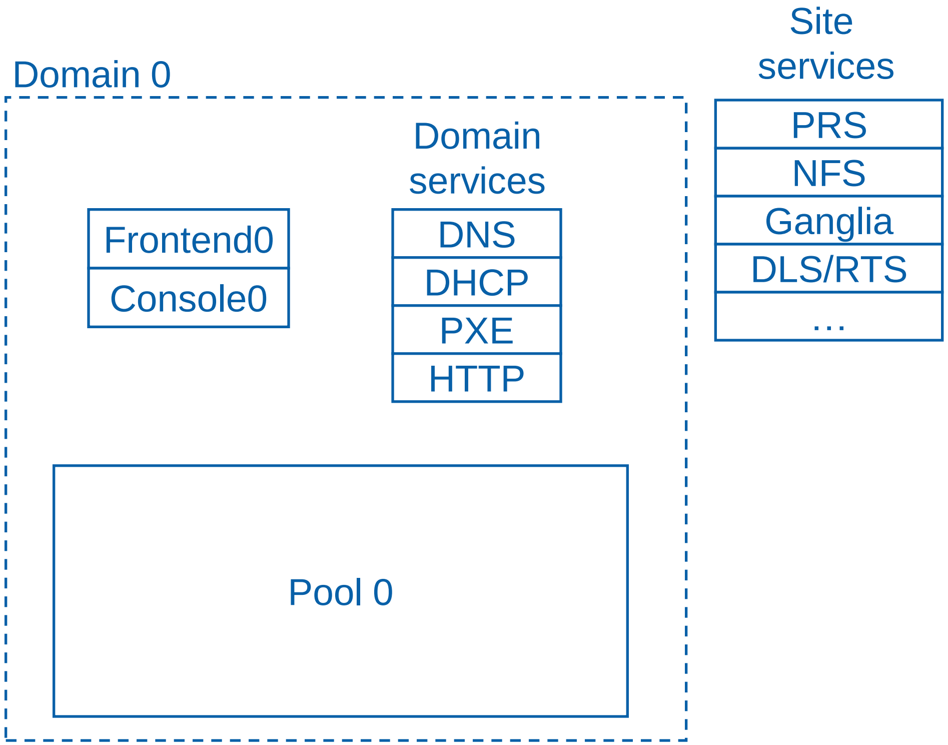


Future Plans

- V1.0 – Cluster admin interface (CLI)
  - Installation
  - Documentation
- V1.1 – Client/Server Interface
  - User can request creation of Domains and Pools
  - Admin still in loop
- V1.1a – Intelligent scheduling
  - Zoni capable of scheduling and reserving resources
  - Automatic reclamation of allocated resources
- V1.5a – Integration with Tashi
  - Zoni makes calls to tashi to move virtual resources
  - Tashi makes calls Zoni to get resources as needed
- V2.0 – Fully automated user-submitted requests

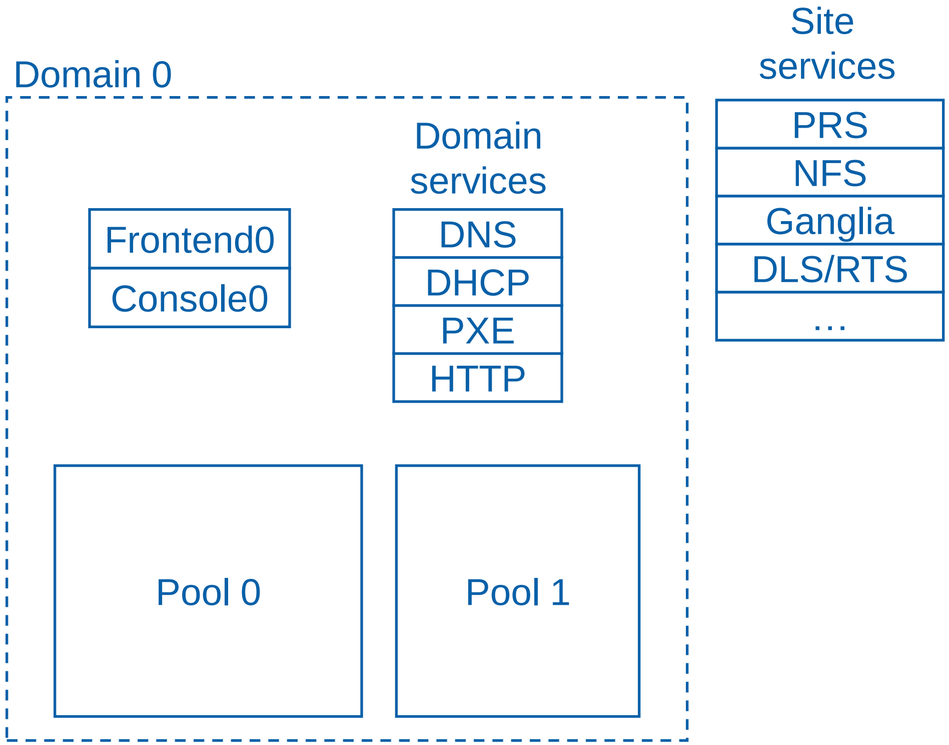
Apache incubator project under Tashi
(http://incubator.apache.org/tashi)

Initial Site Configuration



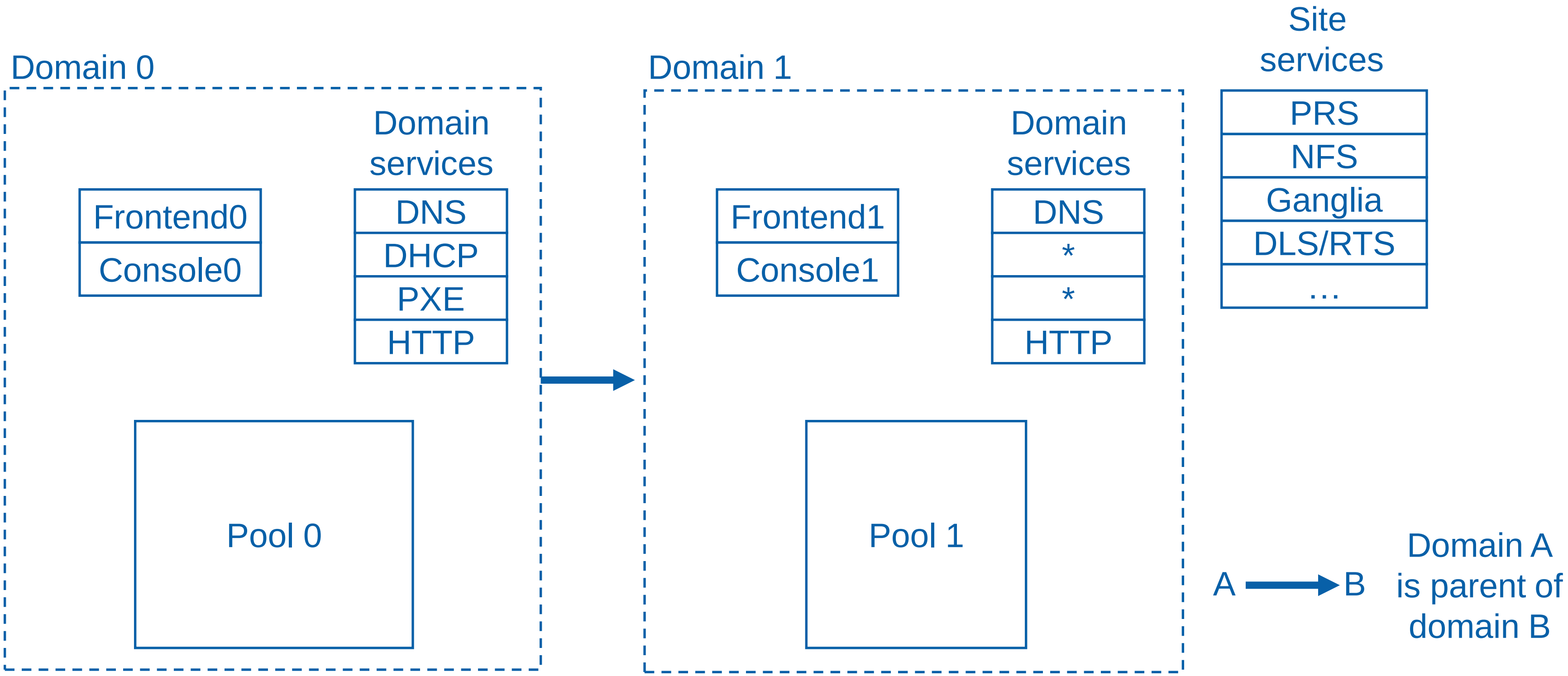
- One frontend machine and console per domain
- One set of domain services per domain
  - DNS: name service
  - PXE/DHCP: kernel and initrd service
  - HTTP: root file system image service
- One set of site services per site

Creating new server pool in same VLAN domain



New server pool shares the same frontend, console, and domain services.

Creating new server pool in different VLAN domain



Child domain can inherit some or all of parent’s domain services.

2-level domain hierarchy.

Zoni Summary

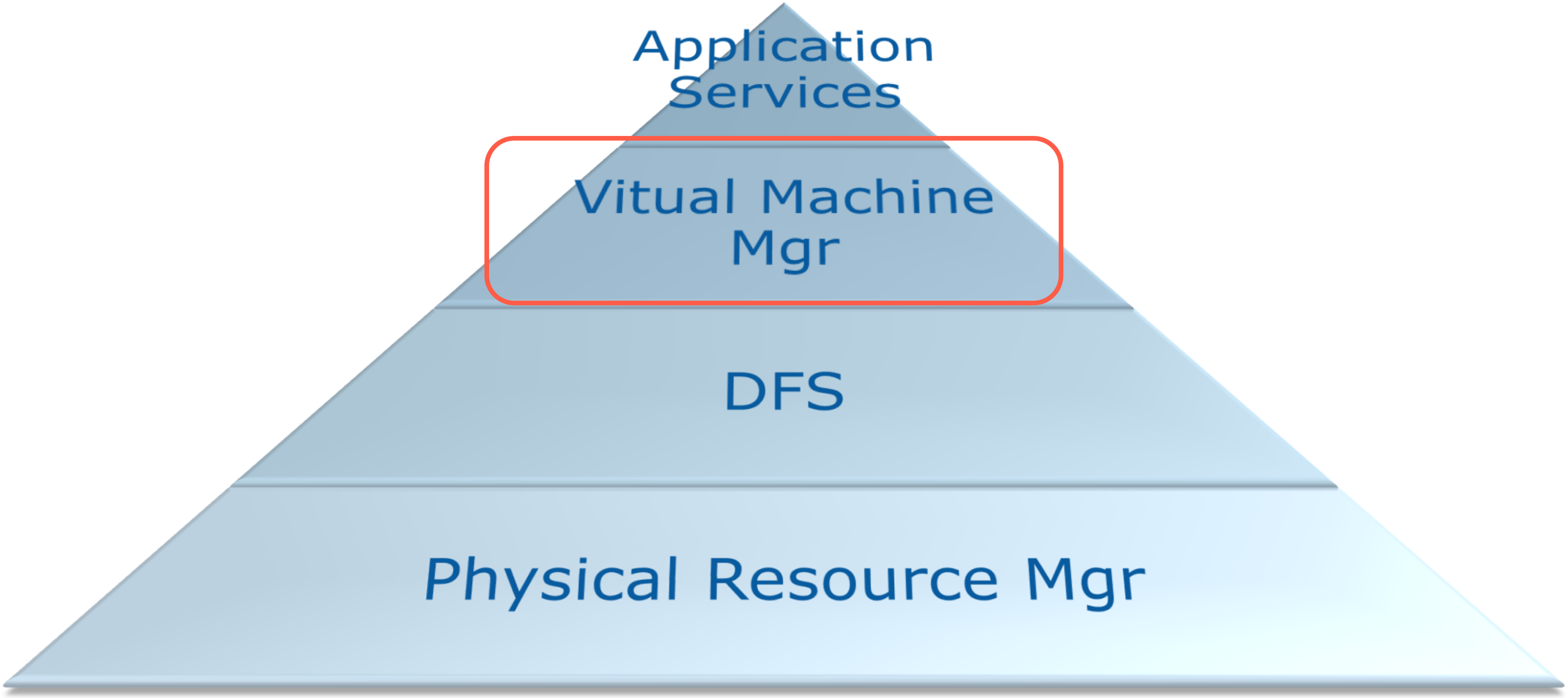
- Zoni functionality
  - Isolation
  - Allcation
  - Provisioning
  - Debugging
  - Remote management
- Dynamic reconfiguration of physical resources
- Quick turn around time to acquire resources

Richard Gass

Tutorial Part II

Tashi

Open source stack



Why Virtualization?

Ease of deployment

- Boot many copies of an operating system very quickly

Cluster lubrication

- Machines can be migrated or even restarted very easily in a different location

Overheads are going down

- Even workloads that tax the virtual memory subsystem can now run with a very small overhead
- I/O intensive workloads have improved dramatically, but still have some room for improvement

Open Cirrus Stack - Tashi

An open source Apache Software Foundation project sponsored by Intel, CMU, and HP.

Research infrastructure for investigating cloud computing on Big Data

- Implements AWS interface
- Daily production use on Intel cluster for 9 months
  - Manages pool of 100 physical nodes
  - ~20 projects/40 users from Intel, CMU, UPitt, Rice
- http://incubator.apache.org/projects/tashi

Research focus:

- Location-aware co-scheduling of VMs, storage, and power.
- Integrated physical/virtual migration (using Zoni)

Tashi System Requirements

Provide high-performance execution over Big Data repositories

➔ Many spindles, many CPUs, co-location

Enable multiple services to access a repository concurrently

Enable low-latency scaling of services

Enable each service to leverage its own software stack

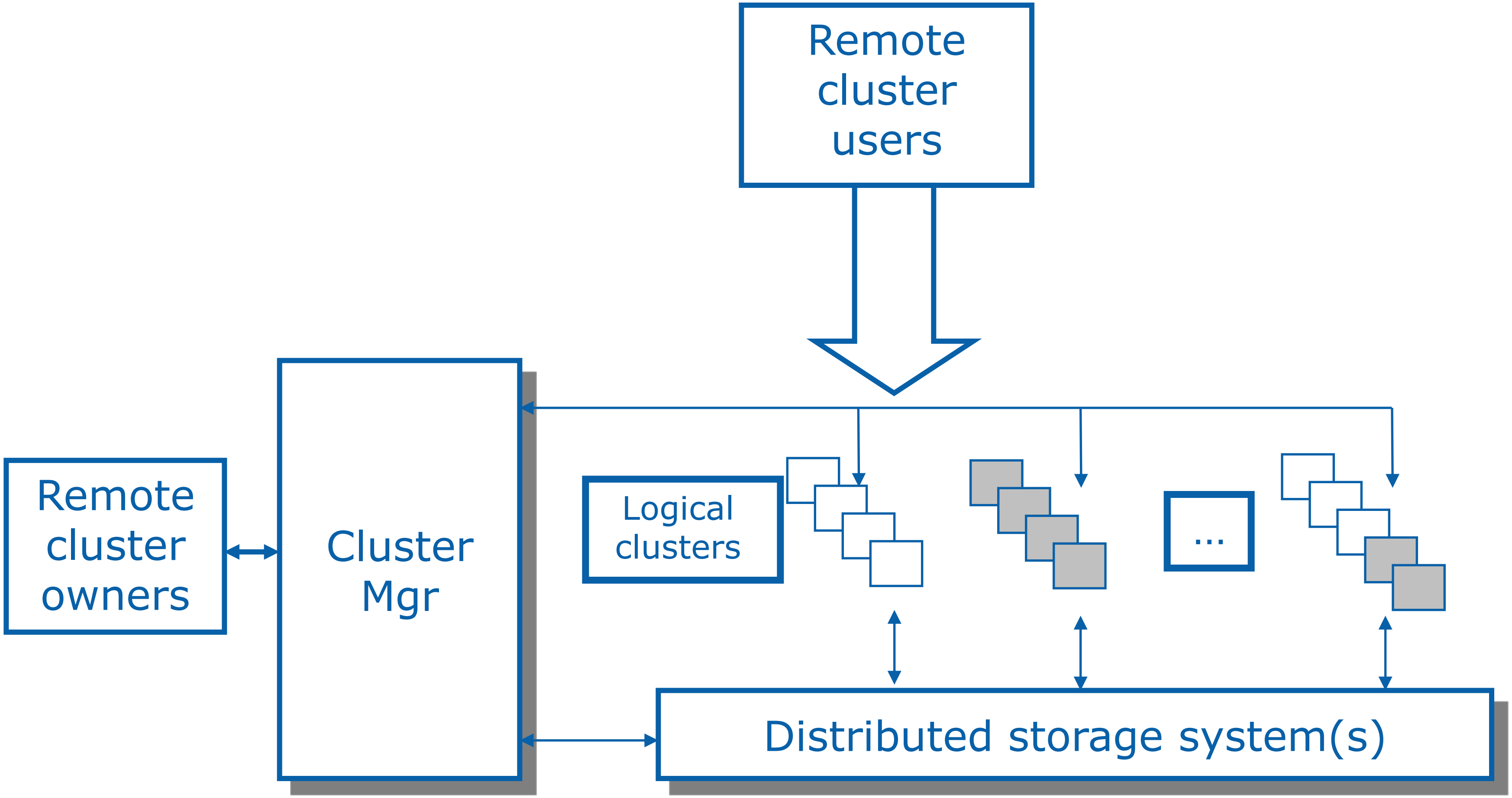
➔ Virtualization, file-system protections

Enable slow resource scaling for growth

Enable rapid resource scaling for power/demand

➔ Scaling-aware storage

Tashi High Level Architecture



Tashi Organization

Each cluster contains one Tashi Cluster Manager (CM)

The CM maintains a database of:

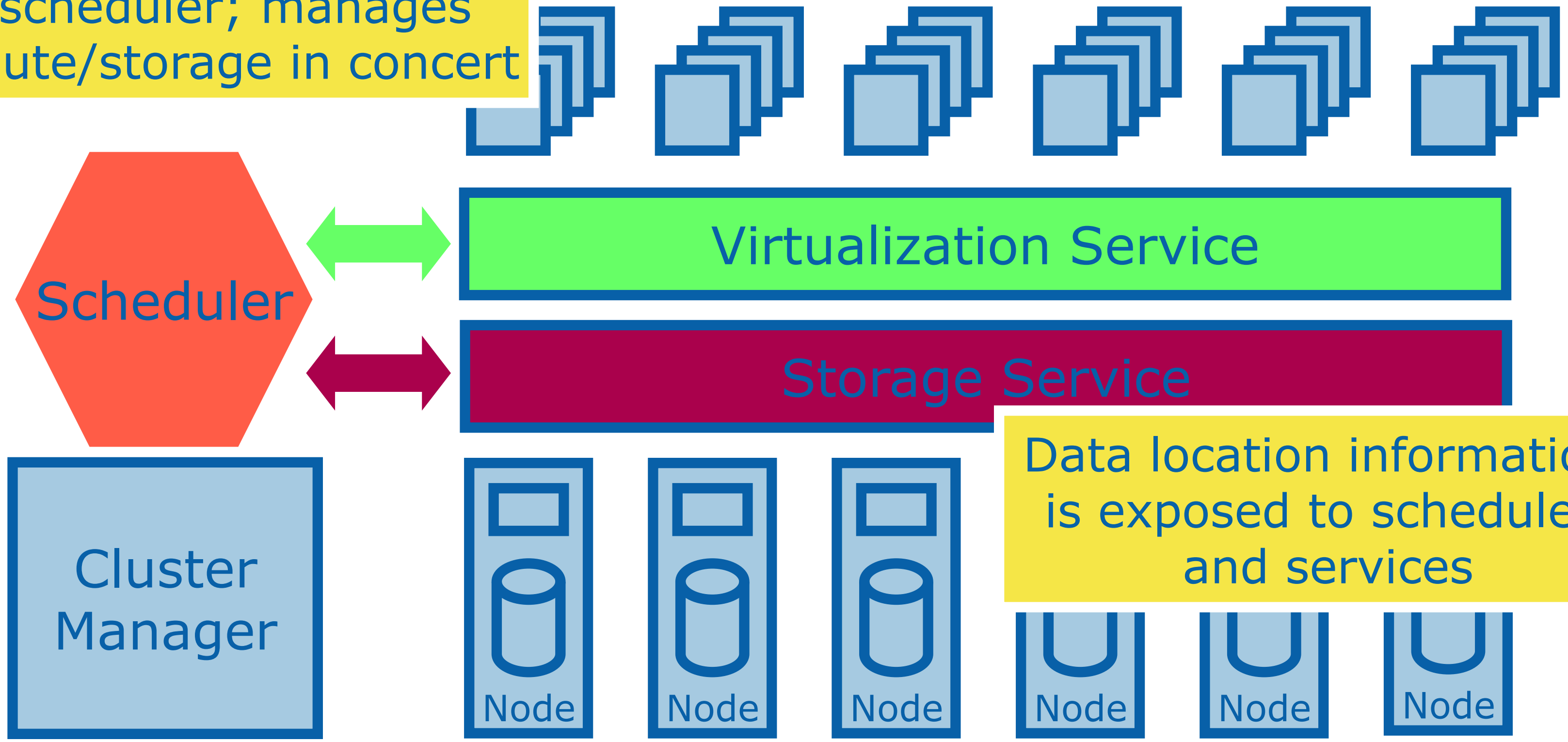
- Available physical resources (nodes)
- Active virtual machines
- Pending requests for virtual machines
- Virtual networks

Users submit requests to the CM through a Tashi Client

The Tashi Scheduler uses the CM databases to invoke actions, such as VM creation, through the CM

Each node contains a Node Manager that carries out actions, such as invoking the local Virtual Machine Manager (VMM), to create a new VM, and monitoring the performance of VMs

Tashi Components



Most decisions happen in the scheduler; manages compute/storage in concert

Services are instantiated through virtual machines

Data location information is exposed to scheduler and services

CM maintains databases and routes messages

Cluster nodes are assumed to be commodity machines

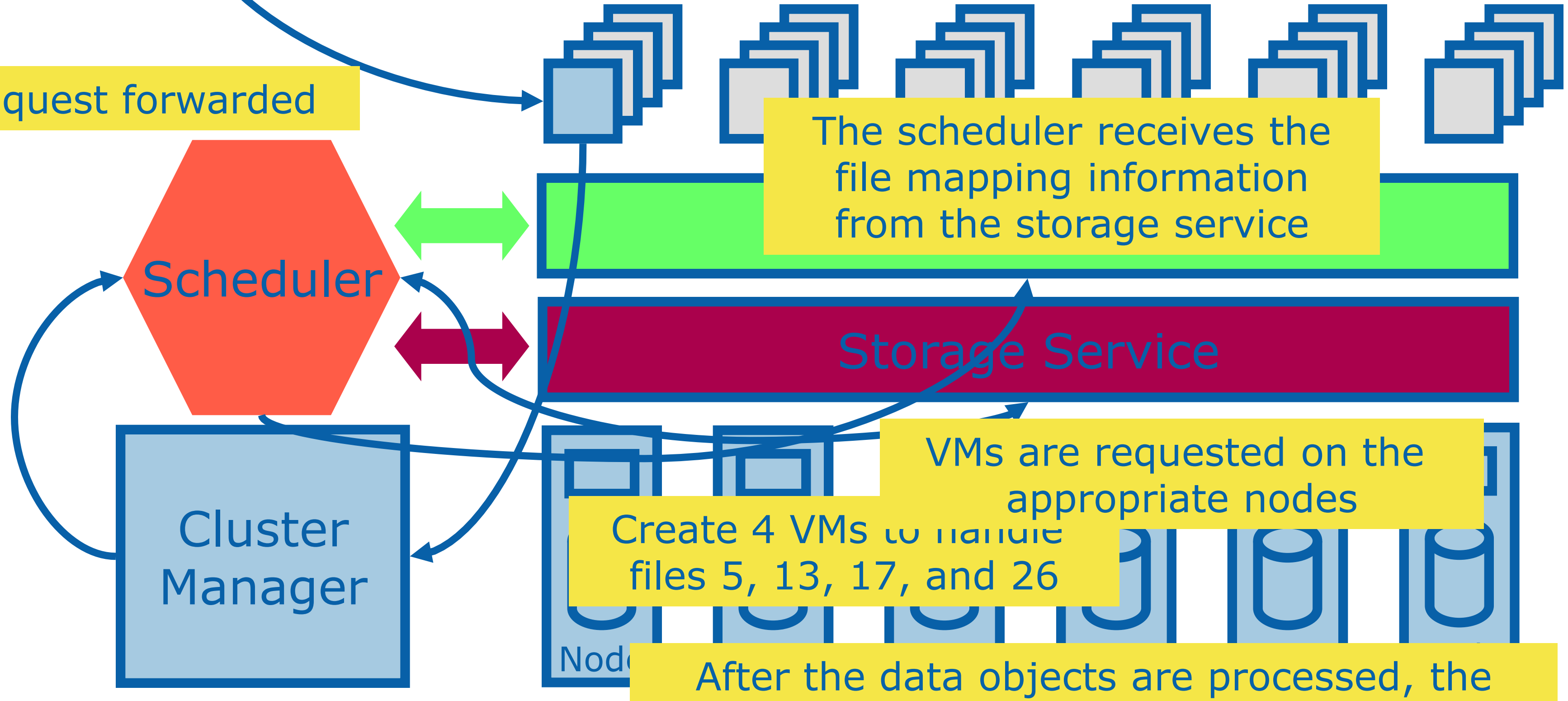
Tashi Operation

A query arrives

answers.opencirrus.net
web server running in 1 V

The web server converts the query into a parallel data processing request

Acting as a Tashi client, a request for additional VMs is submitted



Request forwarded

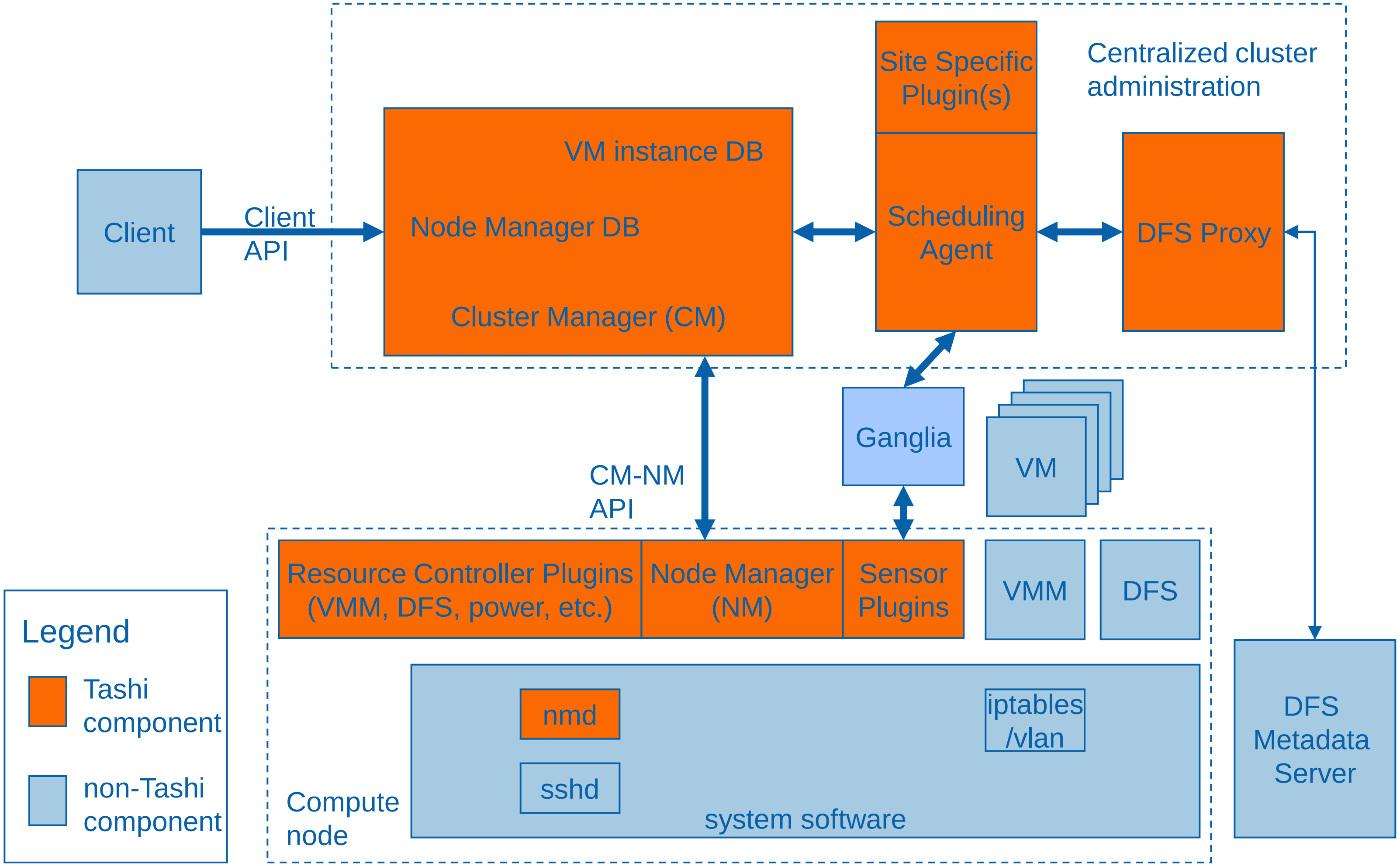
The scheduler receives the file mapping information from the storage service

Create 4 VMs to handle files 5, 13, 17, and 26

VMs are requested on the appropriate nodes

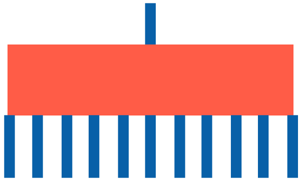
After the data objects are processed, the results are collected and forwarded to Alice. The VMs can then be destroyed

Tashi Software Architecture

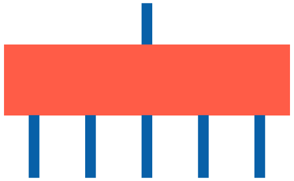


Far vs Near Analysis

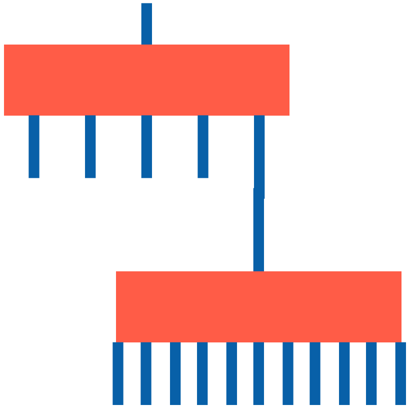
Scenario 1:
11 Racks
@ 4 Gbps



Scenario 2:
5 Racks
@ 8 Gbps



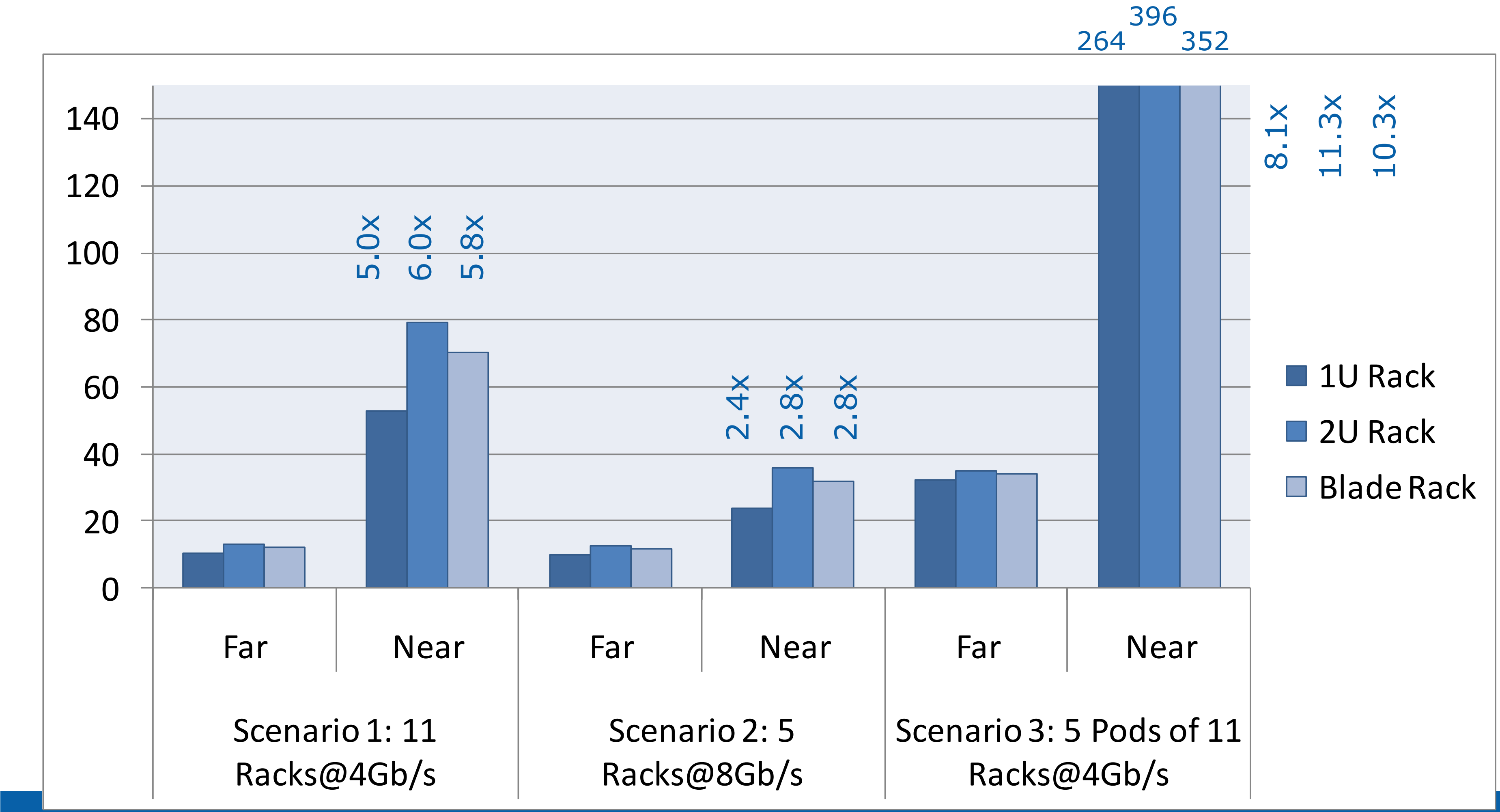
Scenario 3:
5 Pods
@ 8Gbps
of
11 Racks
@ 4 Gbps



Far vs Near Methodology

1. Assume I/O bound (scan) application
2. One task per spindle, no CPU load
3. In the far system, data is consumed on a randomly selected node
4. In the near system, data is consumed on the node where stored
5. Average throughput, no queueing model

Far vs Near Access Throughput



Demo

- Images
- Clustermanager/Nodemanager
- Client
  - Create vm
  - Destroy vm
  - Vmmspecificcall

Tashi Native Client Interface (I)

VM Creation/Destruction Calls (Single Version)

- createVm [--userId <value>] --name <value> [--cores <value>] [--memory <value>] --disks <value> [--nics <value>] [--hints <value>]
- destroyVm --instance <value>
- shutdownVm --instance <value>

VM Creation/Destruction Calls (Multiple Version)

- createMany [--userId <value>] --basename <value> [--cores <value>] [--memory <value>] --disks <value> [--nics <value>] [--hints <value>] --count <value>
- destroyMany --basename <value>

Creating a VM

```
tashi createVm --name mikes-vm --cores 4 --memory 1024 --disks hardy.qcow2
```

--name specifies the DNS name to be created

--disks specifies the disk image

Advanced:

[--nics <value>]

[--hints <value>]

Tashi: Instances

An instance is a running VM

- Each disk image may be used for multiple VMs if the ‘persistent’ bit is not set
- A VM may be booted in persistent mode to make modifications without building an entirely new disk image

getMyInstances Explained

tashi getMyInstances

This lists all VMs belonging to your userId

This is a good way to see what you're currently using

getVmLayout Explained

tashi getVmLayout

This command displays the layout of currently running VMs across the nodes in the cluster

```
 id  name     state   instances                       usedMemory memory usedCores cores
----------------------------------------------------------------------------------------
 126 r3r2u42  Normal  ['bfly3', 'bfly4']              14000      16070  16        16
 127 r3r2u40  Normal  ['mpa-00']                      15360      16070  8         16
 128 r3r2u38  Normal  ['xren1', 'jpan-vm2']           15480      16070  16        16
 129 r3r2u36  Normal  ['xren3', 'collab-00']          14800      16070  16        16
 130 r3r2u34  Normal  ['collab-02', 'collab-03']      14000      16070  16        16
 131 r3r2u32  Drained []                              0          16068  0         16
 132 r3r2u30  Normal  ['collab-04', 'collab-05']      14000      16070  16        16
 133 r3r2u28  Normal  ['collab-06', 'collab-07']      14000      16070  16        16
```

Tashi Native Client Interface (II)

VM Management Calls

- suspendVm --instance <value>
- resumeVm --instance <value>

- pauseVm --instance <value>
- unpauseVm --instance <value>

- migrateVm --instance <value> --targetHostId <value>

- vmmSpecificCall --instance <value> --arg <value>

Tashi Native Client Interface (III)

Bookkeeping Calls

- getMyInstances
- getInstances
- getVmLayout
- getUsers
- getNetworks
- getHosts

Creating Multiple VMs

```
tashi createMany –count 10 --basename mikes-vm --cores 4 --memory 1024 --disks hardy.qcow2
```

--name specifies the DNS name to be created

--disks specifies the disk image

Advanced:

[--nics <value>]

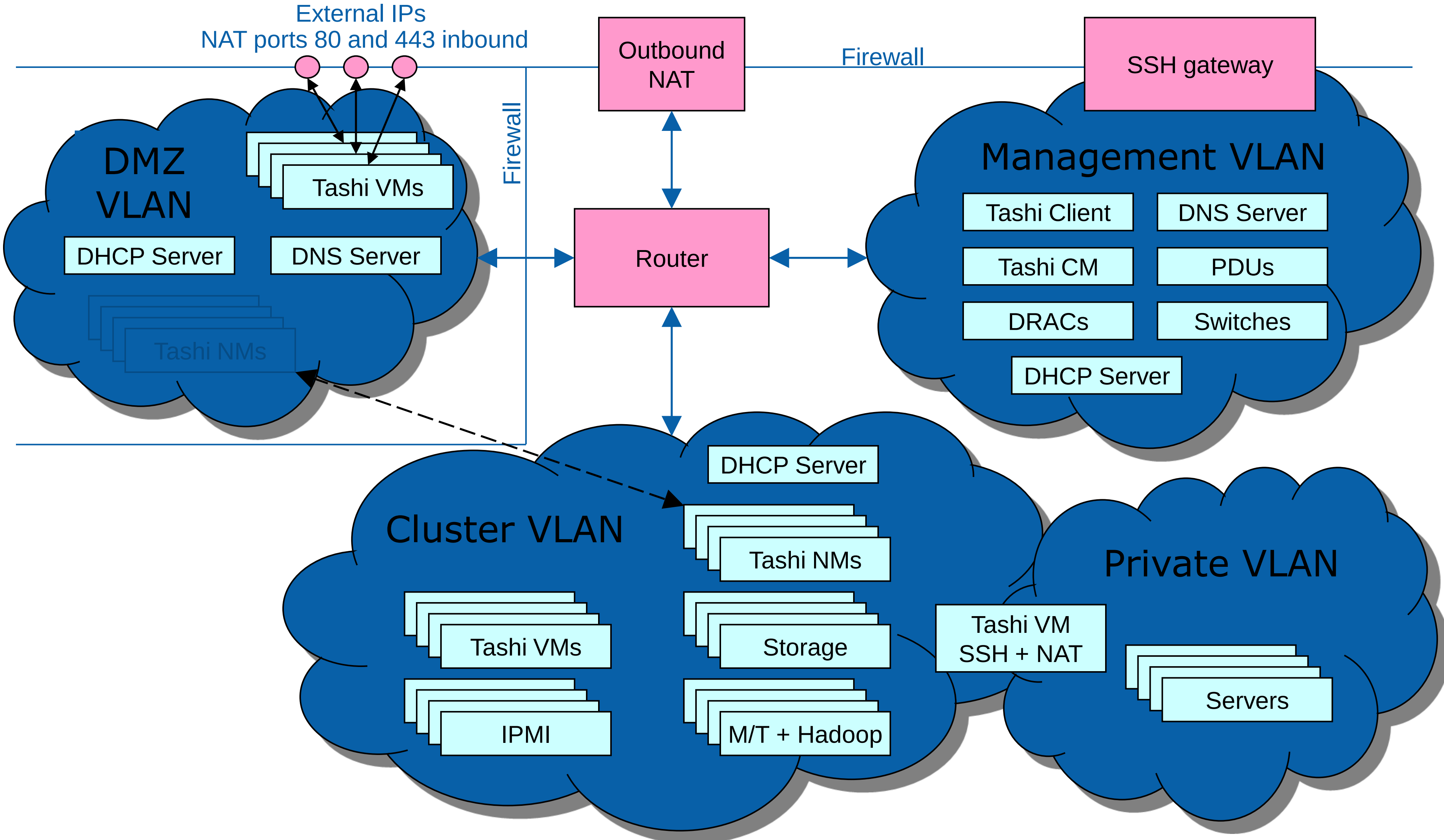
[--hints <value>]

Tashi Deployment

Intel Labs Pittsburgh

- Tashi is used on the Open Cirrus site at ILP
- Majority of the cluster
  - Some nodes run Maui/Torque, Hadoop
- Primary source of computational power for the lab
- Mix of preexisting batch users, HPC workloads, Open Cirrus customers, and others

Intel BigData Cluster - Networking



Tashi Summary

- Tashi is a location aware cluster management system
- Data sets are too large so we need to move the processing closer to the data
- Open source

Questions/Comments

Richard Gass - <richard.gass@intel.com>

Backup

Cloud todo list

Community must cooperate to create open source service stack

- Including access model, local services, global services, application frameworks

Location- and power-aware workload scheduling are open problems.

Storage models are still a problem.

- GFS-style storage systems not mature, impact of SSDs unknown

Need integrated physical/virtual allocations to combat cluster squatting.

Need to investigate new application frameworks

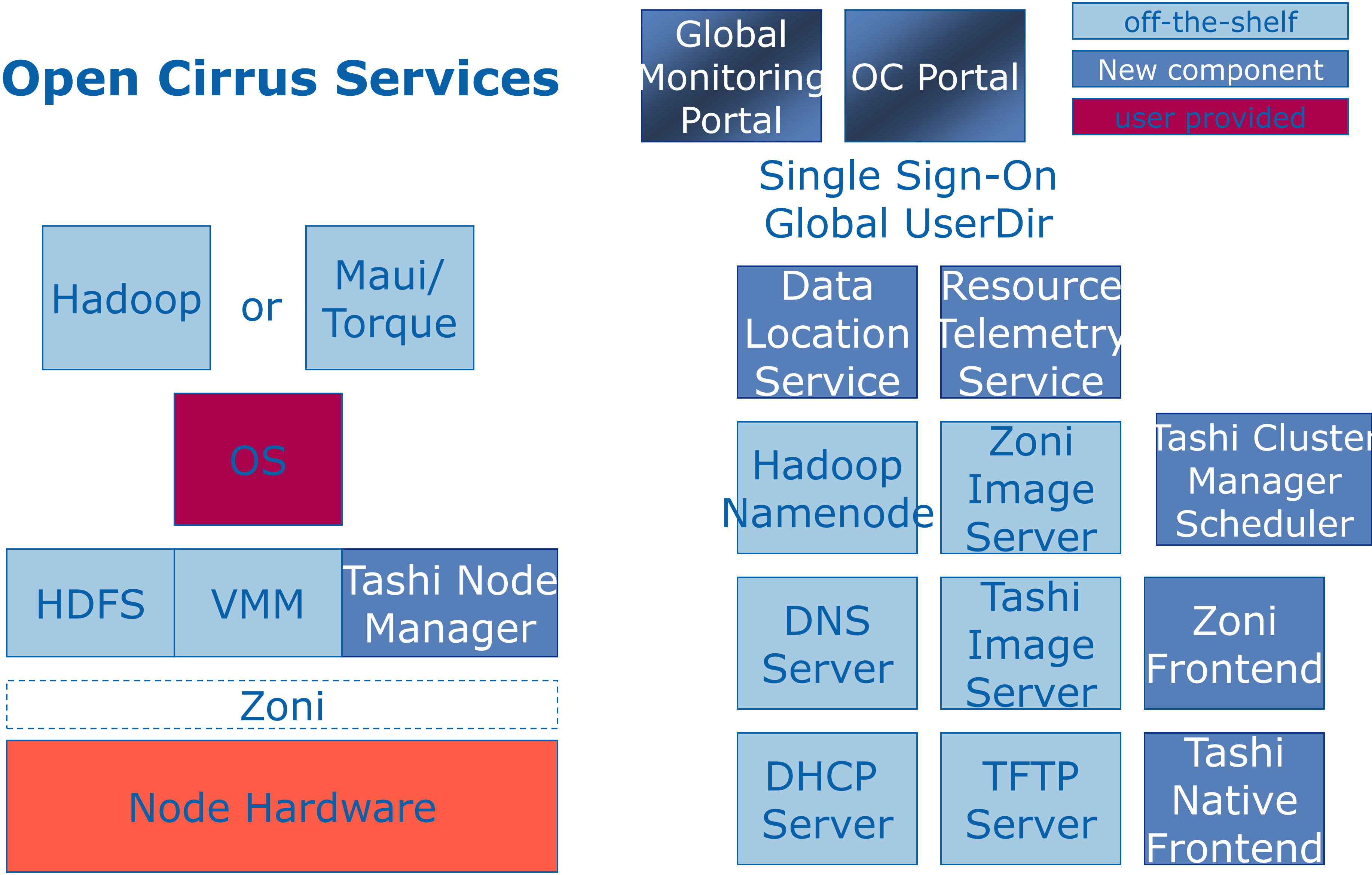
- Map-reduce/Hadoop not always appropriate

Using the cloud as an accelerator for interactive streaming/big data apps is an important usage model.

Zoni installs

- MIMOS (Malaysian Institute of Microelectronic Systems)
  - February 2010
- ETRI (Electronics and Telecommunications Research Institute)
  - May 2010
- HP labs
  - Q4
- CMU
  - Q4
- Intel IT
  - ??

Open Cirrus Services



off-the-shelf

New component

user provided

Global Monitoring Portal

OC Portal

Single Sign-On
Global UserDir

Hadoop or Maui/ Torque

OS

HDFS

VMM

Tashi Node Manager

Zoni

Node Hardware

Data Location Service

Resource Telemetry Service

Hadoop Namenode

Zoni Image Server

Tashi Cluster Manager Scheduler

DNS Server

Tashi Image Server

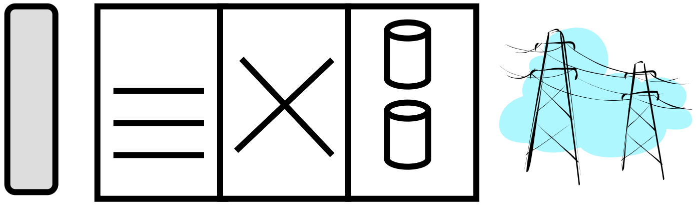
Zoni Frontend

DHCP Server

TFTP Server

Tashi Native Frontend

Open Cirrus Stack - Zoni



An open source Apache Software Foundation project sponsored by Intel, CMU, and HP (included with Tashi distribution)

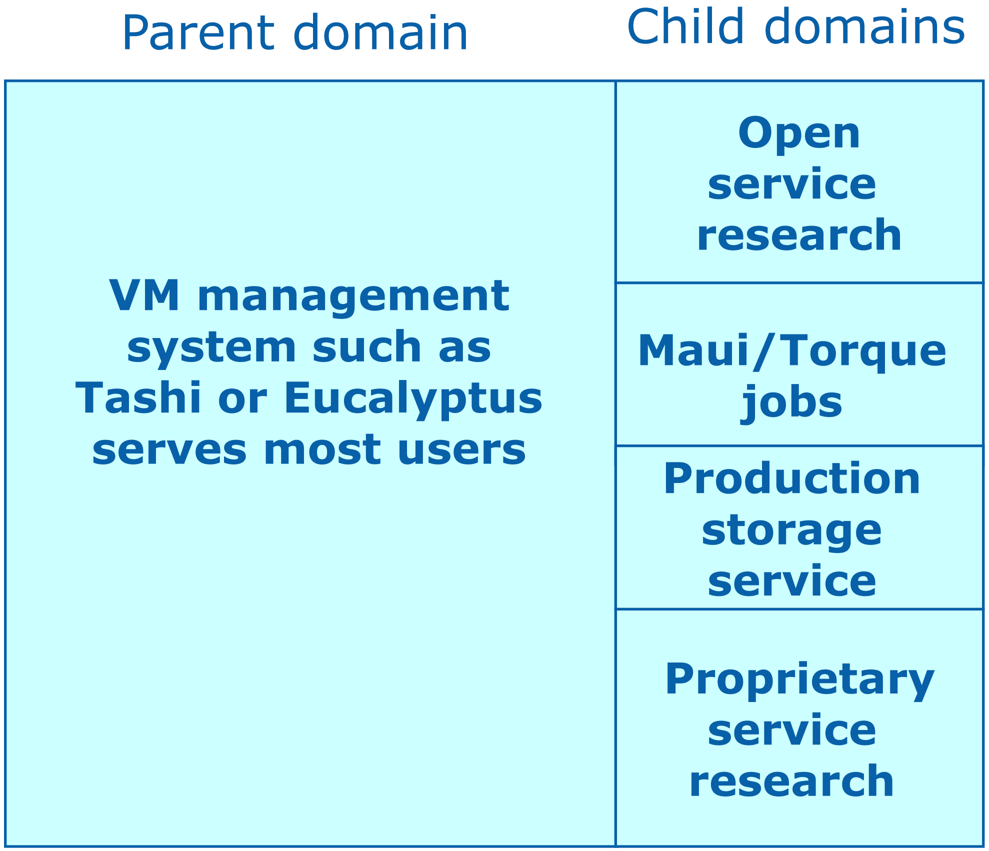
Zoni service goals

- Provide mini-datacenters to researchers and service providers
- Isolate experiments from each other
- Stable base for other research

Zoni service approach

- Allocate sets of physical co-located nodes, isolated inside VLANs.

Example Site Configuration



Parent domain

Child domains

VM management system such as Tashi or Eucalyptus serves most users

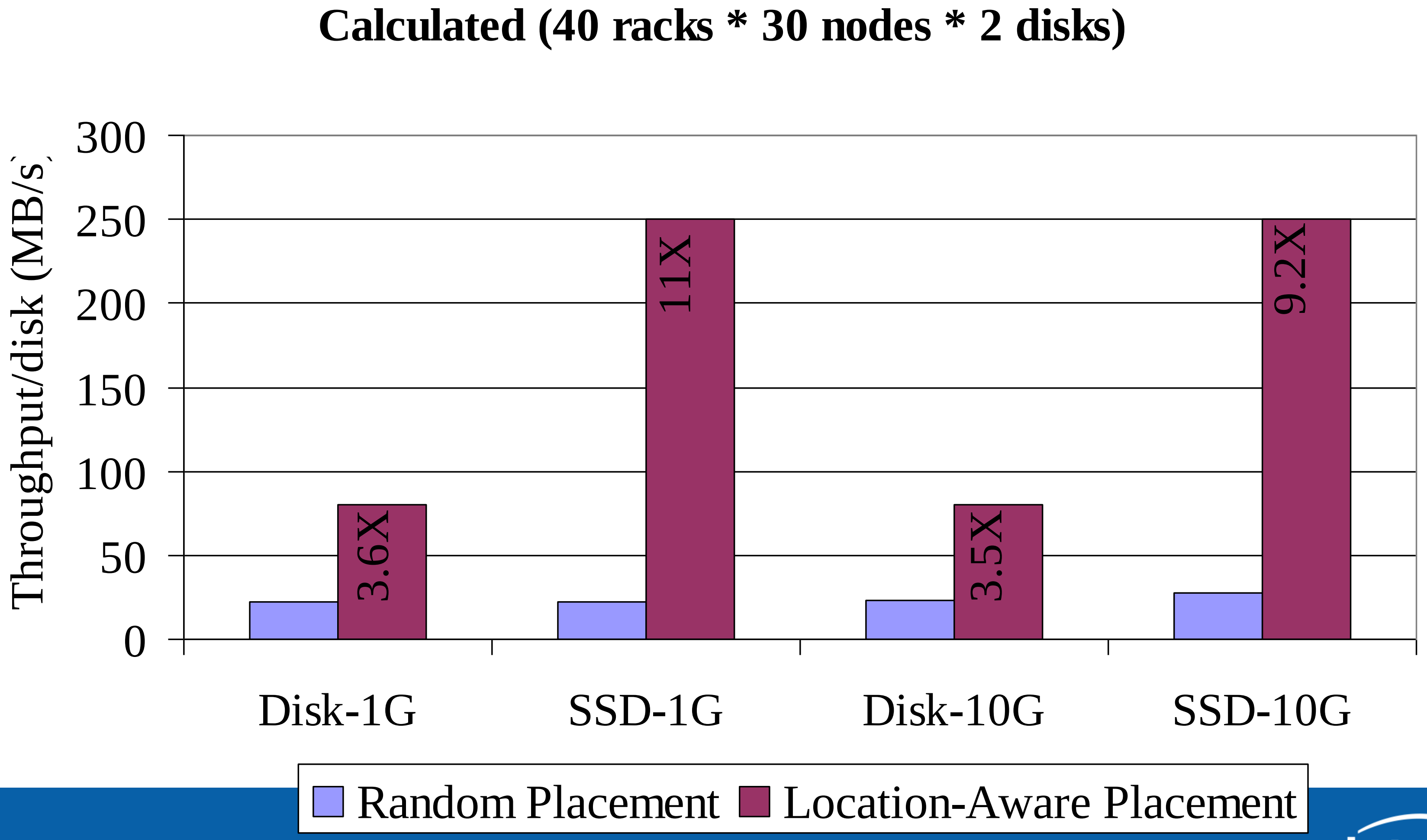
Open service research

Maui/Torque jobs

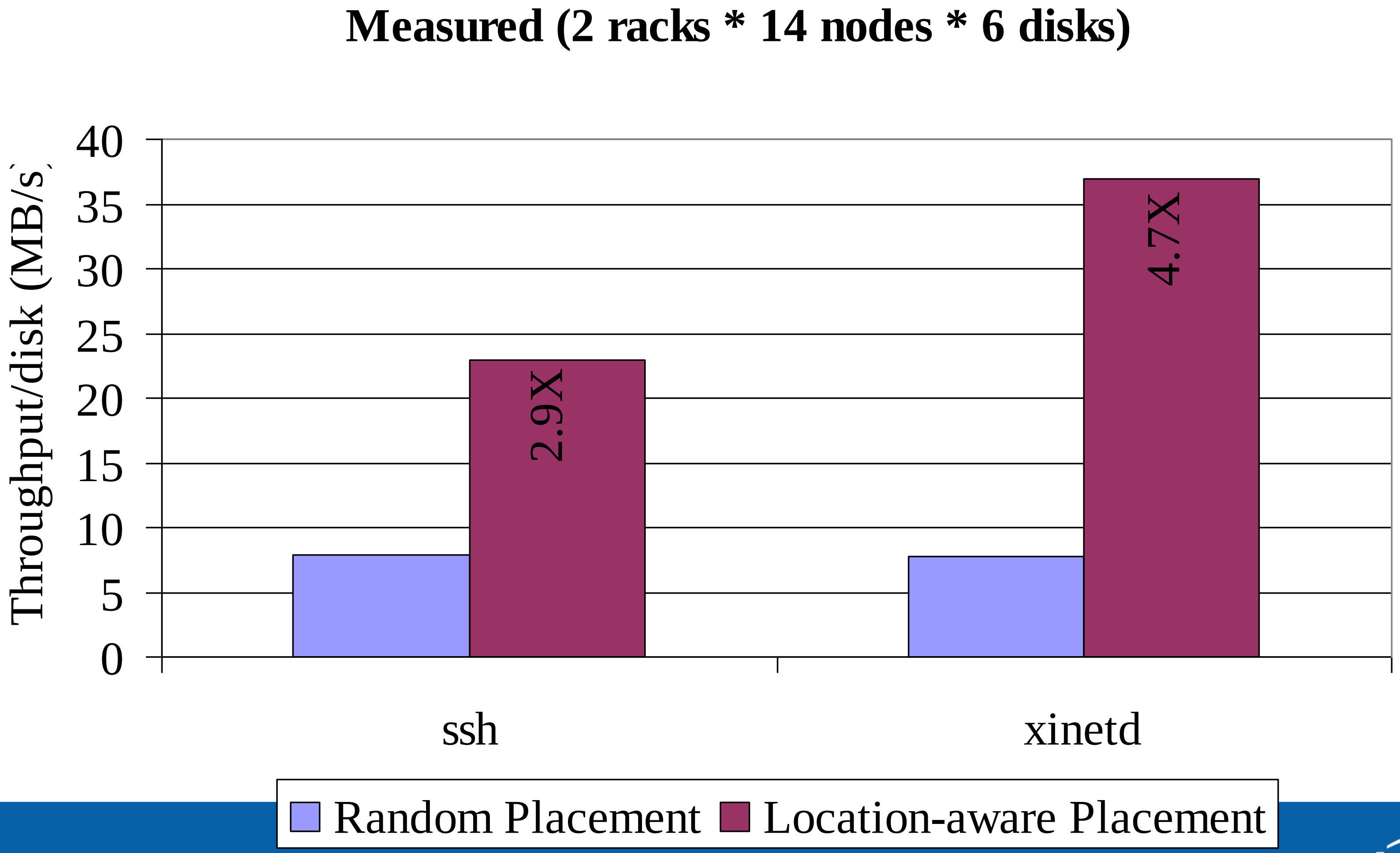
Production storage service

Proprietary service research

Location Matters (calculated)



Location Matters (measured)


Measured (2 racks * 14 nodes * 6 disks)