CloudLab

Tutorial
GEC 22
March 2015

Today's Plan

- Everyone will build their own clouds
  - Using an OpenStack profile supplied by CloudLab
  - Each is independent, with it's own compute and storage resources

- Log in using GENI accounts
- Create a cloud
- Explore the CloudLab interface
- Use your cloud
- Administer your cloud

- **CloudLab is about more than OpenStack**

Prerequisites

- Account on the GENI portal (sent to you as "pre work")
- Optional, but will make your experience better:
  - SSH keypair associated with your GENI portal account
  - Knowledge of how to use the private SSH key from your laptop

- Known to work best in Chrome and Firefox browsers
- Tablets might work, but not well tested

Crash Course in CloudLab

- Underneath, it's GENI
  - Same APIs, same account system
  - Even many of the same tools
  - Federated (accept each other's accounts, hardware)
- Physical isolation for compute, storage (shared net.*)
- Profiles are one of the key abstractions
  - Defines an environment – hardware (RSpec) / software (images)
  - Each "instance" of a profile is a separate
  - Provide standard environments, and a way of sharing
  - Explicit role for domain experts
- "Instantiate" a profile to make an "Experiment"
  - Lives in a GENI slice

* Can be dedicated in some cases

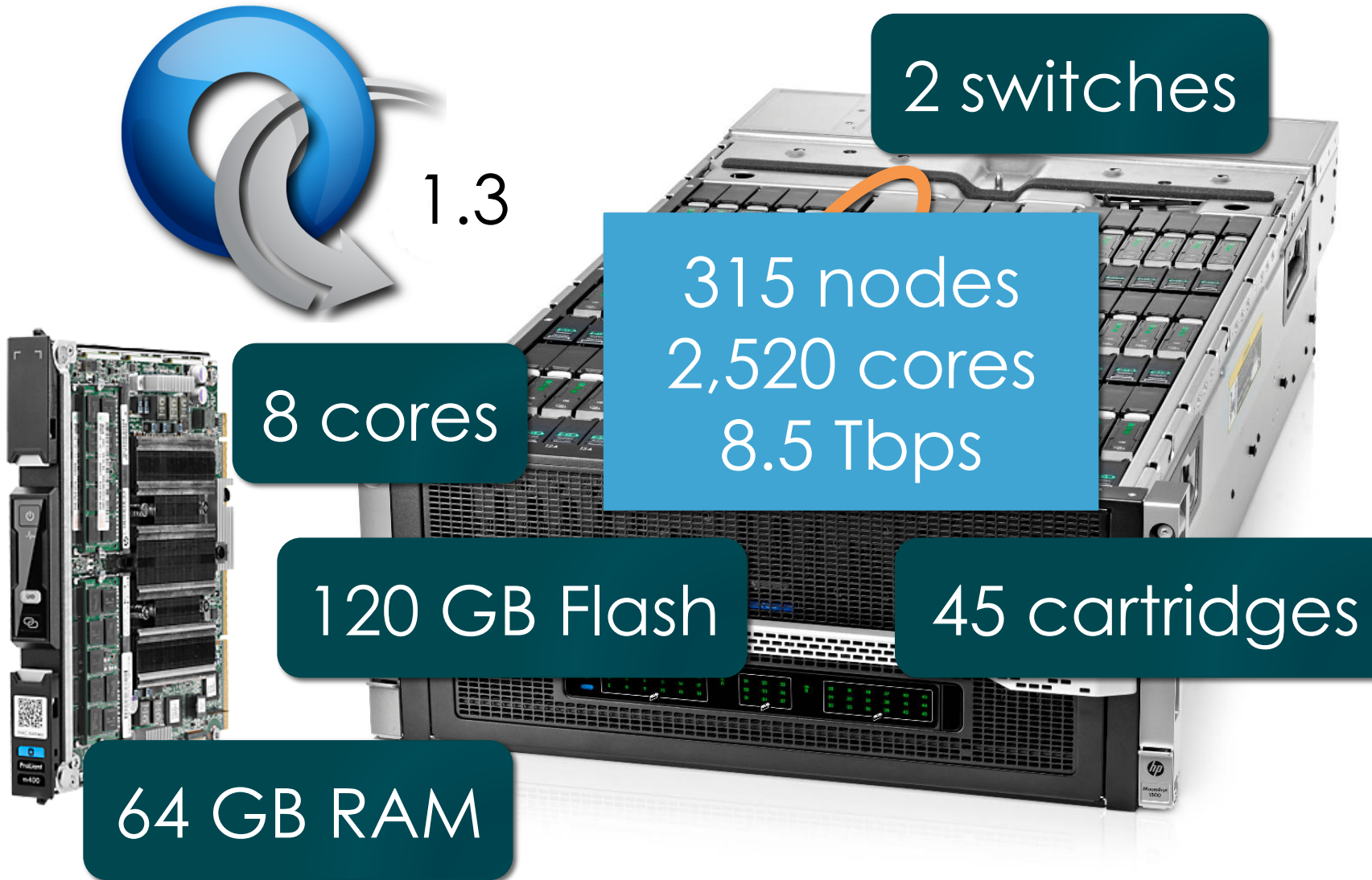# cloudlab.us/tutorial

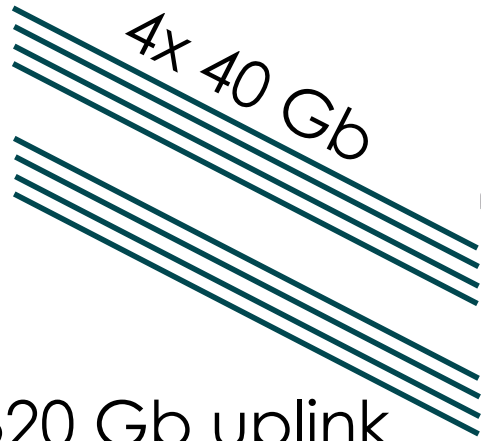# CloudLab Hardware
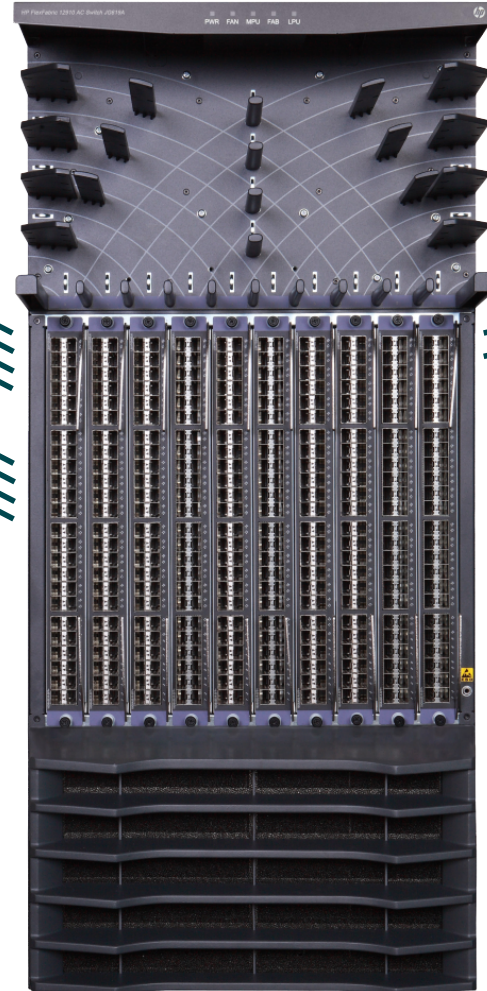
Utah/HP: Very dense

Utah/HP: Low-power ARM64

1.3

2 switches

315 nodes
2,520 cores
8.5 Tbps

8 cores

120 GB Flash

45 cartridges

64 GB RAM

# Utah/HP Network: Core switch

4x 40 Gb

2x 10 Gb

320 Gb uplink

x7

Utah - Suitable for experiments that:

- … explore power/performance tradeoffs
- … want instrumentation of power and temperature
- … want large numbers of nodes and cores
- … want to experiment with RDMA via RoCE
- … need bare-metal control over switches
- … need OpenFlow 1.3
- … want tight ARM64 platform integration

Wisconsin/Cisco

Nexus 3172PQ

8X10G

Nexus 3132Q

40G

40G

Nexus 3172PQ

2X10G

20X12 servers
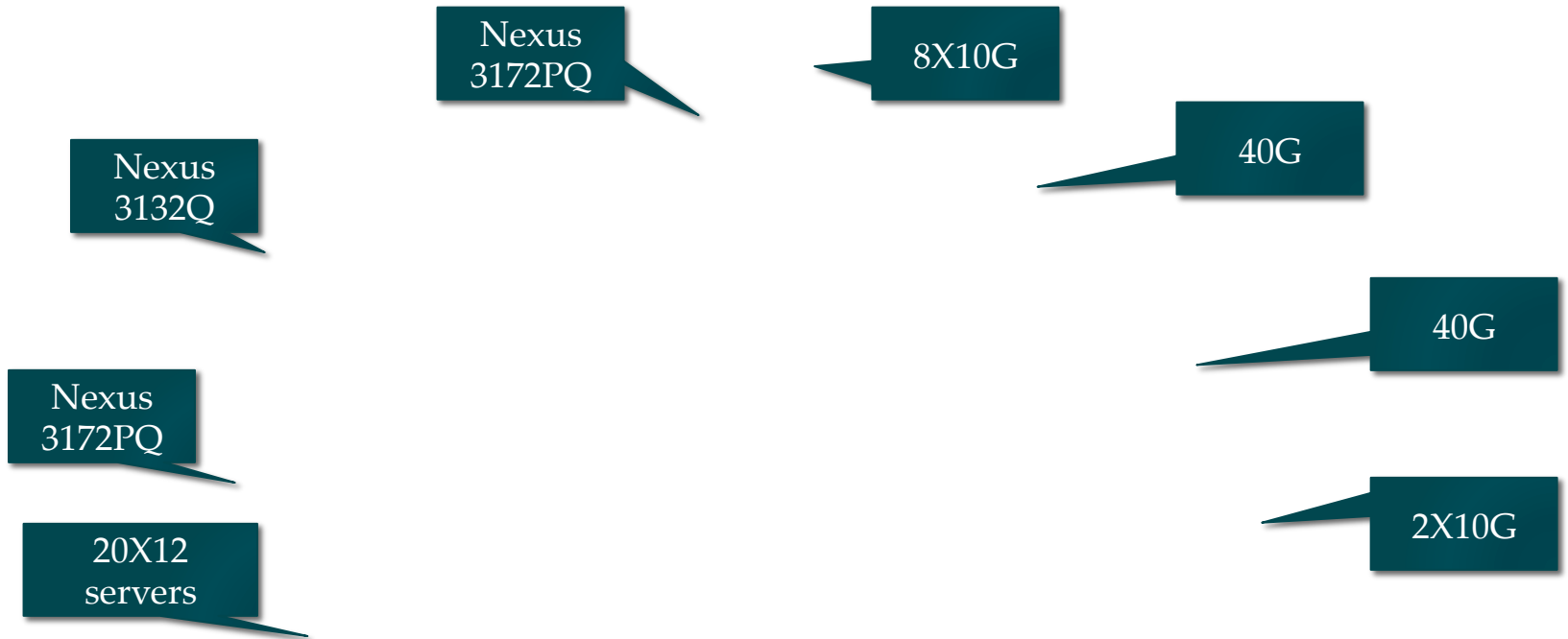
Compute and storage

90X Cisco 220 M4

10X Cisco 240 M4

- 2X 8 cores @ 2.4GHz
- 128GB RAM
- 1X 480GB SSD

- 2X 1.2 TB HDD

- 1X 1TB HDD
- 12X 3TB HDD
(donated by Seagate)

Over the next year: ≥ 140 additional servers;
Limited number of accelerators, e.g., FPGAs, GPUs (planned)

Networking

Nexus 3132q   Nexus 3172pq

- OF 1.0 (working with Cisco on OF 1.3 support)
- Monitoring of instantaneous queue lengths
- Fine-grained tracing of control plane actions
- Support for multiple virtual router instances per router
- Support for many routing protocols

Experiments supported

Large number of nodes/cores, and bare-metal control over nodes/ switches, for sophisticated network/memory/storage research

- … Network I/O performance, intra-cloud routing (e.g., Conga) and transport (e.g., DCTCP)
- … Network virtualization (e.g., CloudNaaS)
- … In-memory big data frameworks (e.g., Spark/Shark)
- … Cloud-scale resource management and scheduling (e.g., Mesos; Tetris)
- … New models for Cloud storage (e.g., tiered; flat storage; IOFlow)
- … New architectures (e.g., RAM Cloud for storage)

Clemson/Dell: High Memory, IB

20 cores/node

1 x 40 Gb IB/node

8 nodes/chassis

2*x 10 GbE OF/node

10 chasses/rack

2*x 1 GbE OF/node

256 GB RAM/node

2 x 1 TB drive/server

* 1 NIC in 1st build

Clemson/Dell Network: IB + 10 GbE
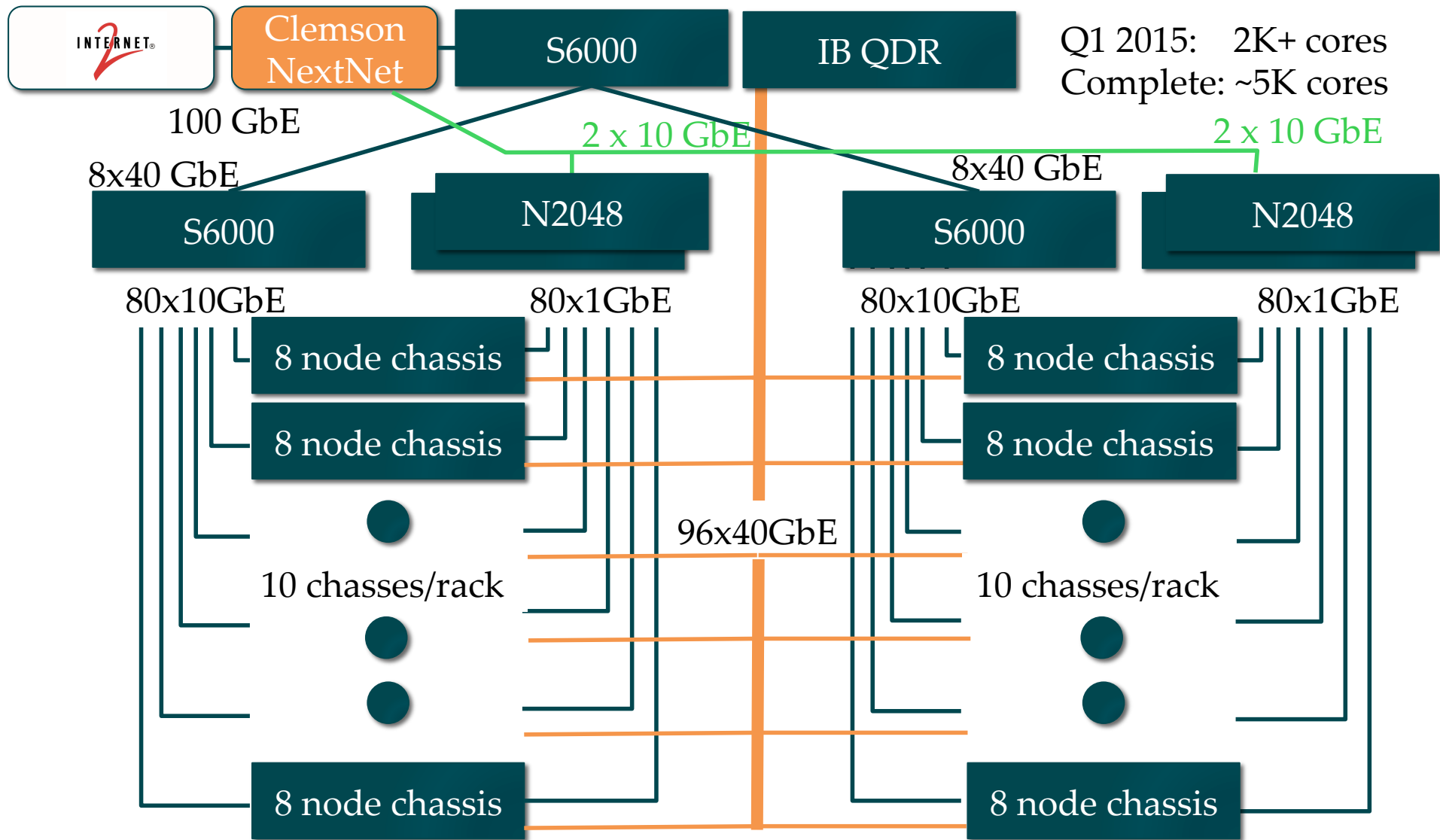
Clemson - Suitable for experiments that:

- … need large per-core memory
  - e.g., High-res media processing
  - e.g. Hadoop
  - e.g., Network Function Virtualization
- … want to experiment with IB and/or GbE networks
  - e.g., hybrid HPC with MPI and TCP/IP
  - e.g., cyber physical system
- … need bare-metal control over switches
- … need OpenFlow 1.3

Building Profiles

Copy an Existing Profile

Use a GUI (Jacks)

Write Python Code (geni-lib)

two-vm.py (~/Desktop) - VIM

```python
import geni.rspec.pg as pg

rspec = pg.Request()

# Create XenVM nodes
node1 = pg.XenVM("node1")
node2 = pg.XenVM("node2")

# Create interfaces for each node.
iface1 = node1.addInterface("if1")
iface2 = node2.addInterface("if2")

rspec.addResource(node1)
rspec.addResource(node2)

# Create a link with the type of LAN.
link = pg.LAN("lan")

# Add both node interfaces to the link.
link.addInterface(iface1)
link.addInterface(iface2)
```

CloudLab

Build From Scratch

Home   Manual   Actions ▾   rpruser logged in   Logout

Create Profile

Name ❓   [   ]   **Project**   [flux   ▲▼]

Your rspec   [Choose file] [Topology] [Source]

Description ❓   [   ]

Instructions ❓   [   ]

☐ List on the home page for anyone to view.

Who can instantiate your profile?

○ *Anyone* on the internet (guest users)

● *Only registered users of the website*

Sign Up

Sign Up At CloudLab.us