LA-UR-16-28013
Approved for public release; distribution is unlimited.

Title:	Proactive Identification and Remediation of HPC Network Subsystem Failures
Author(s):	Coulter, Susan K.
Intended for:	Grace Hopper Conference 2016, 2016-10-19 (Houston, Texas, United States)
Issued:	2016-10-20

Proactive Identification and Remediation of HPC Network Subsystem Failures

Susan Coulter

Los Alamos National Laboratory

2016

ANITA BORG INSTITUTE

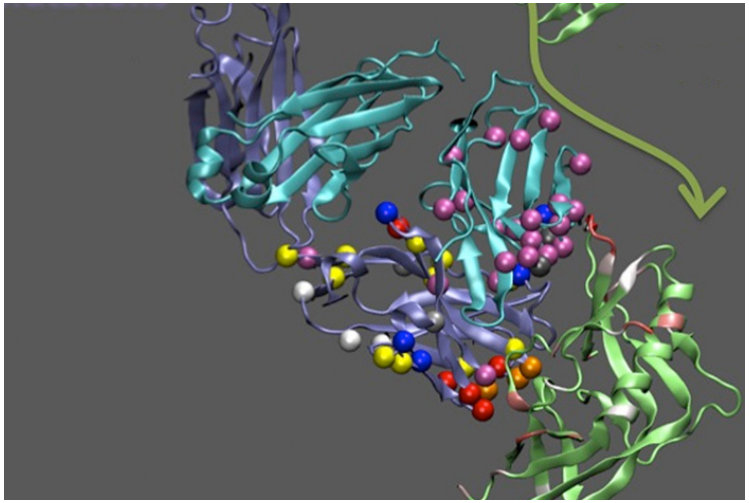
GRACE HOPPER CELEBRATION OF WOMEN IN COMPUTING

#GHC16

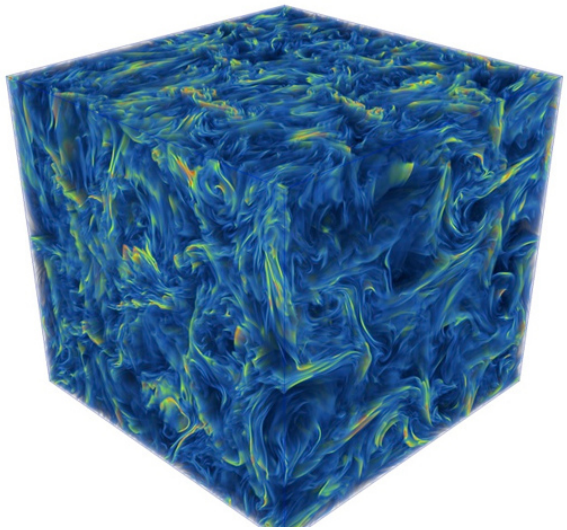
ANITA BORG INSTITUTE

Association for Computing Machinery

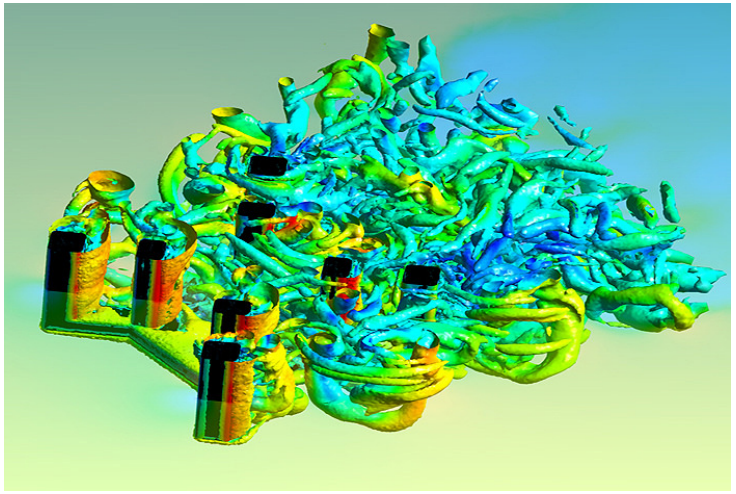
Why and What is High Performance Computing ?



AIDs Virus
Mutation Patterns



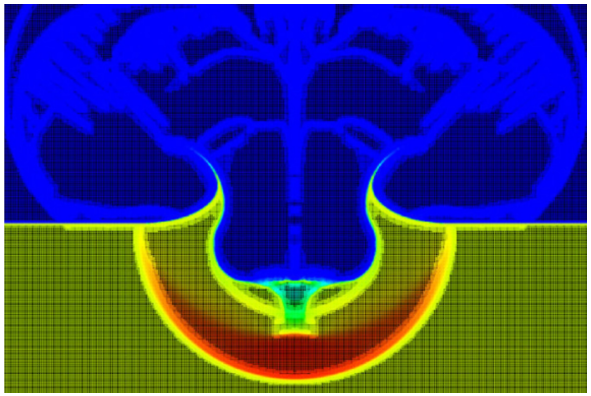
Fast Magnetic Reconnections
(sheds light on solar flares)

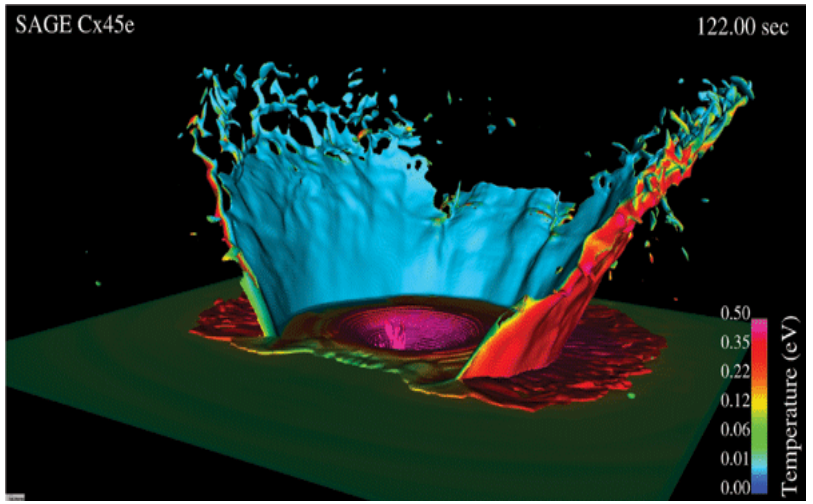


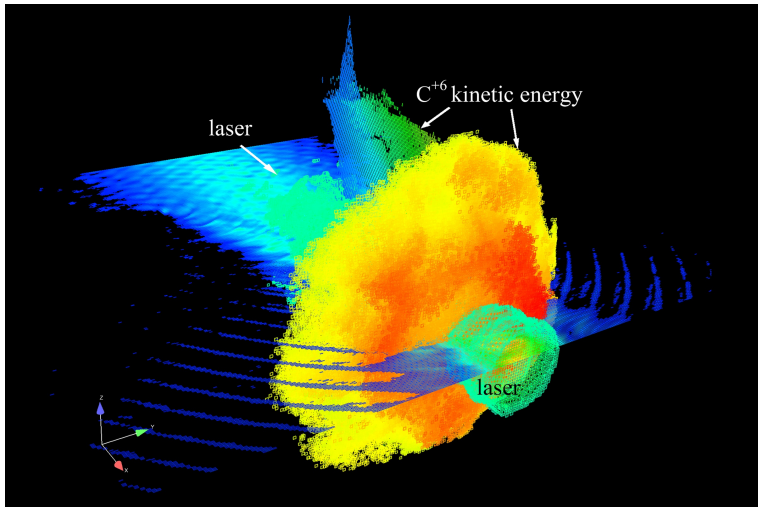
Turbulence Around Oil Rig Platforms

Why and What is High Performance Computing ?

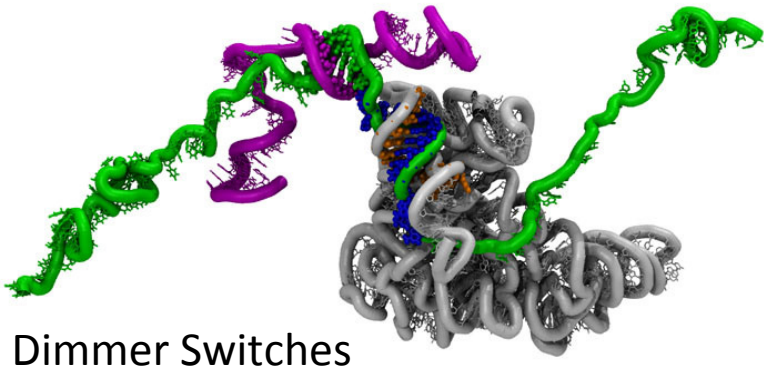
Crater simulations





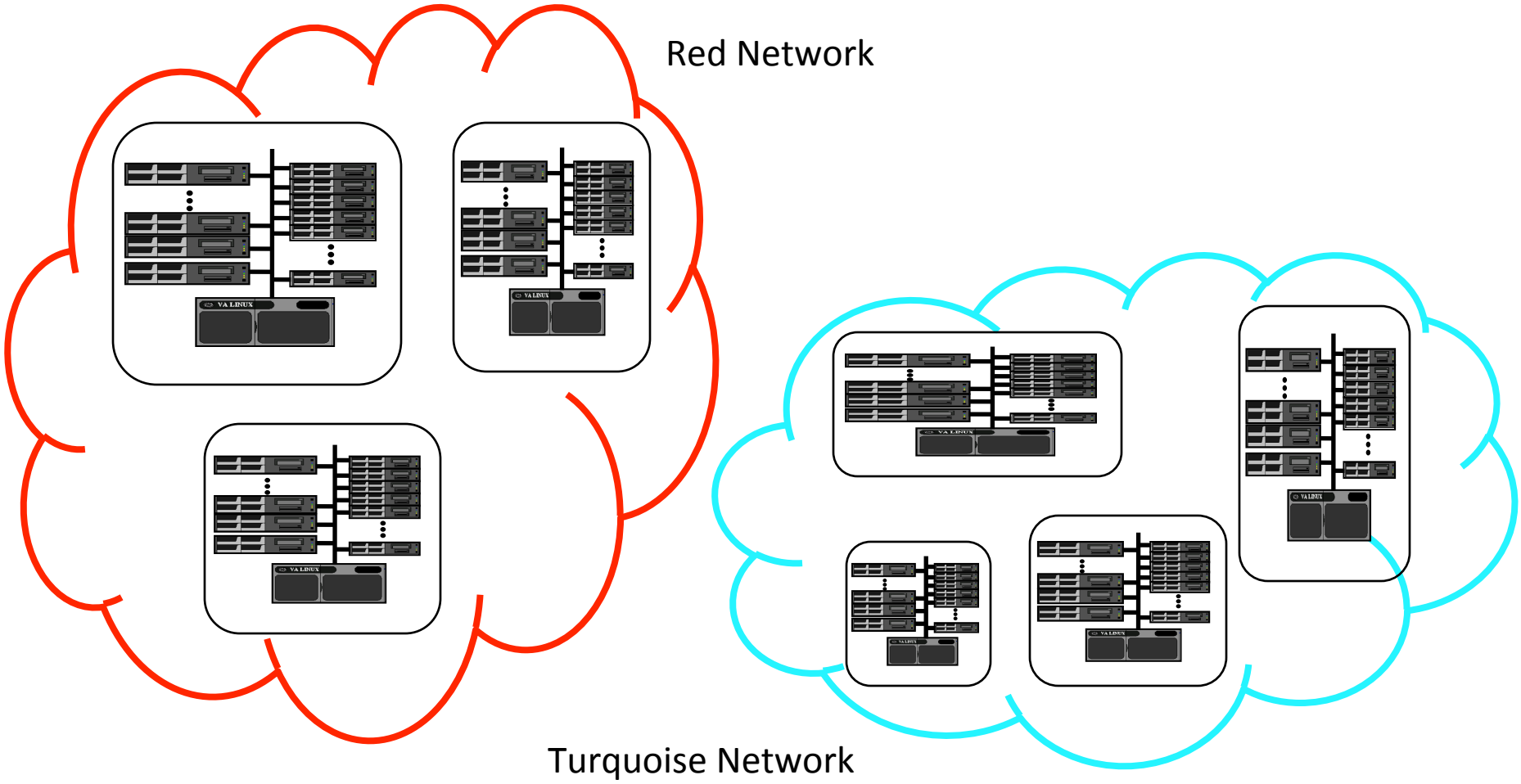


Ion Acceleration

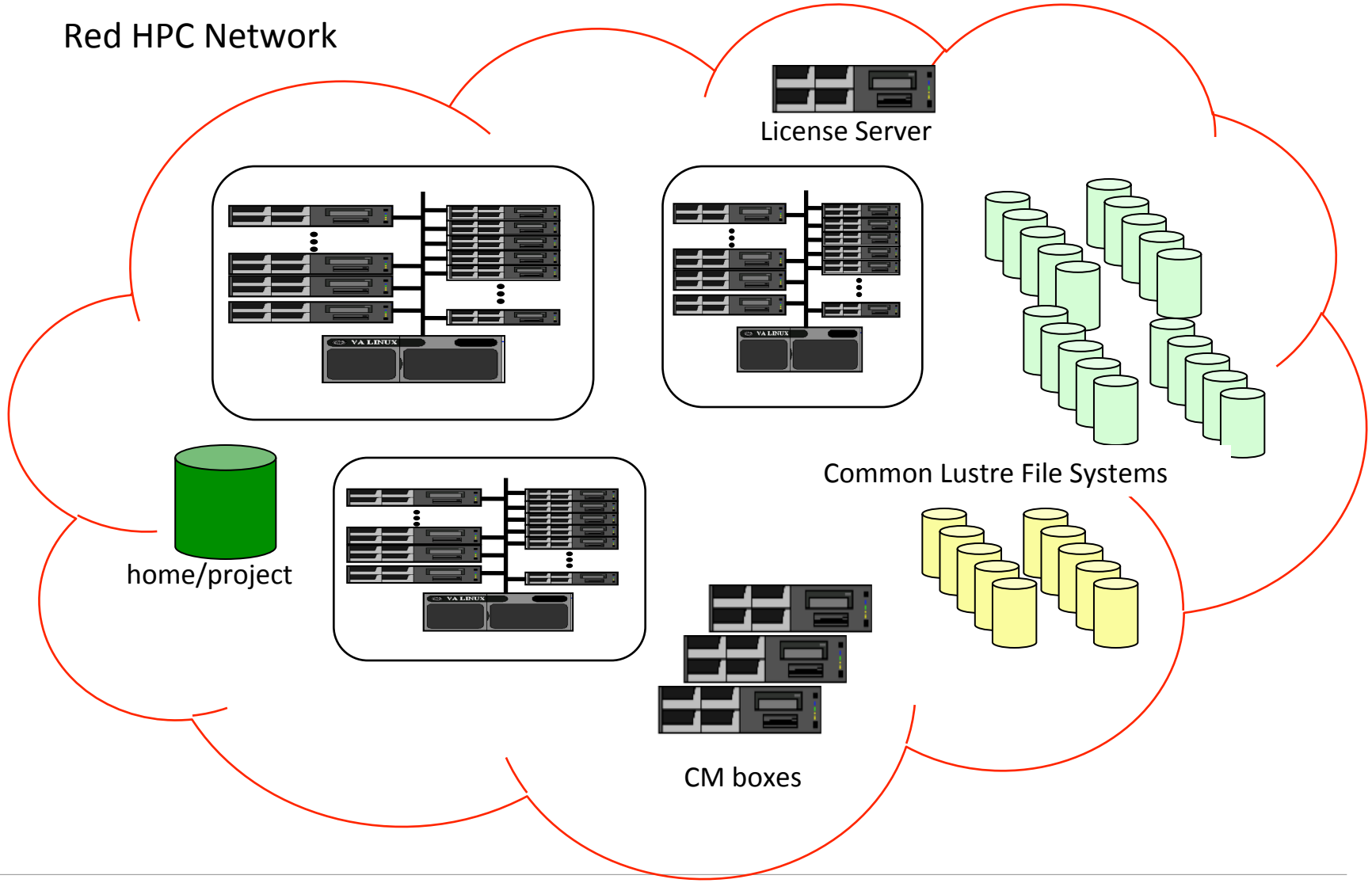


Molecular Dimmer Switches
(controls cellular metabolism)

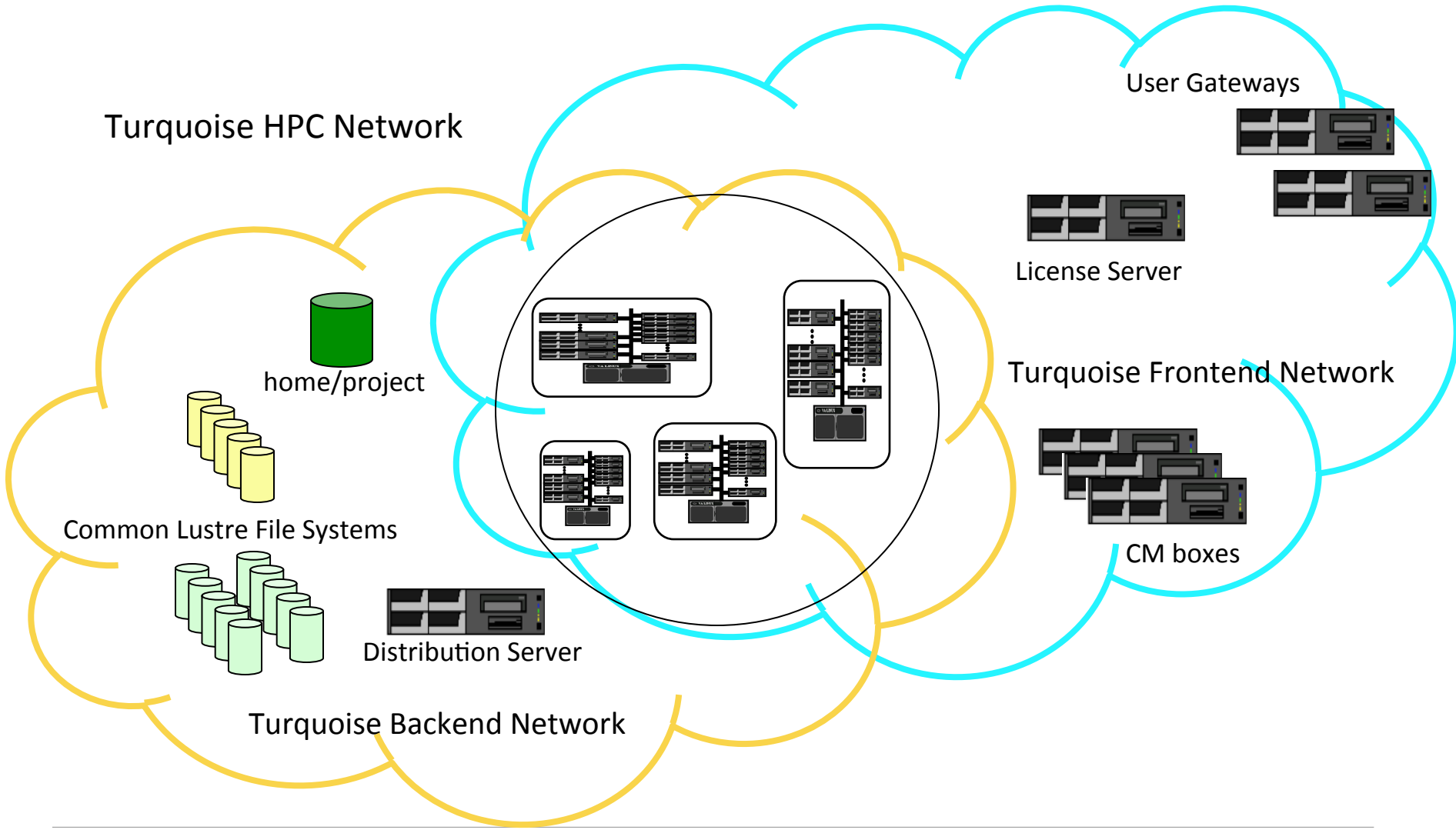
LANL HPC Networks – 30,000 foot view



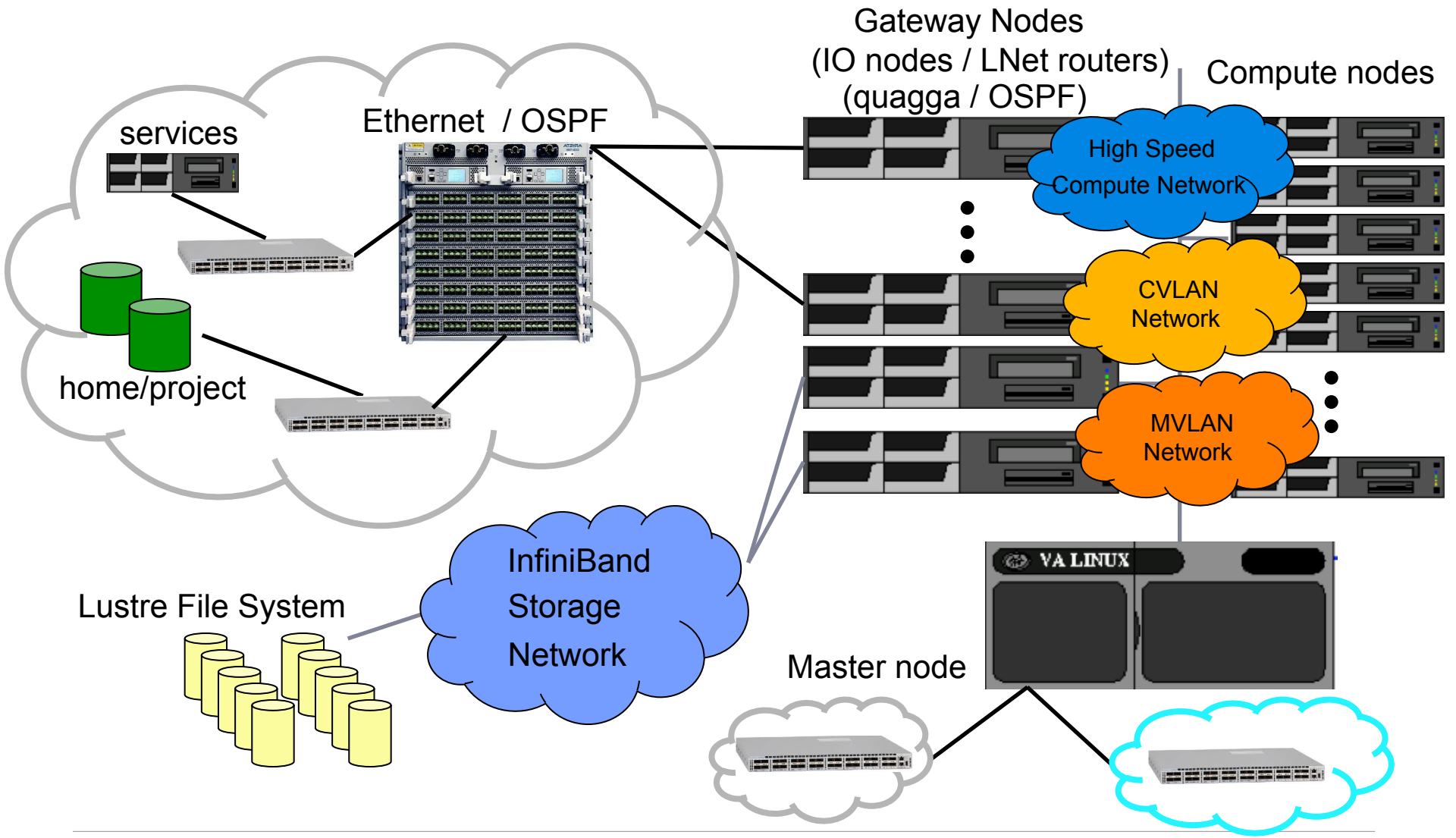
LANL HPC Networks – 15,000 foot view



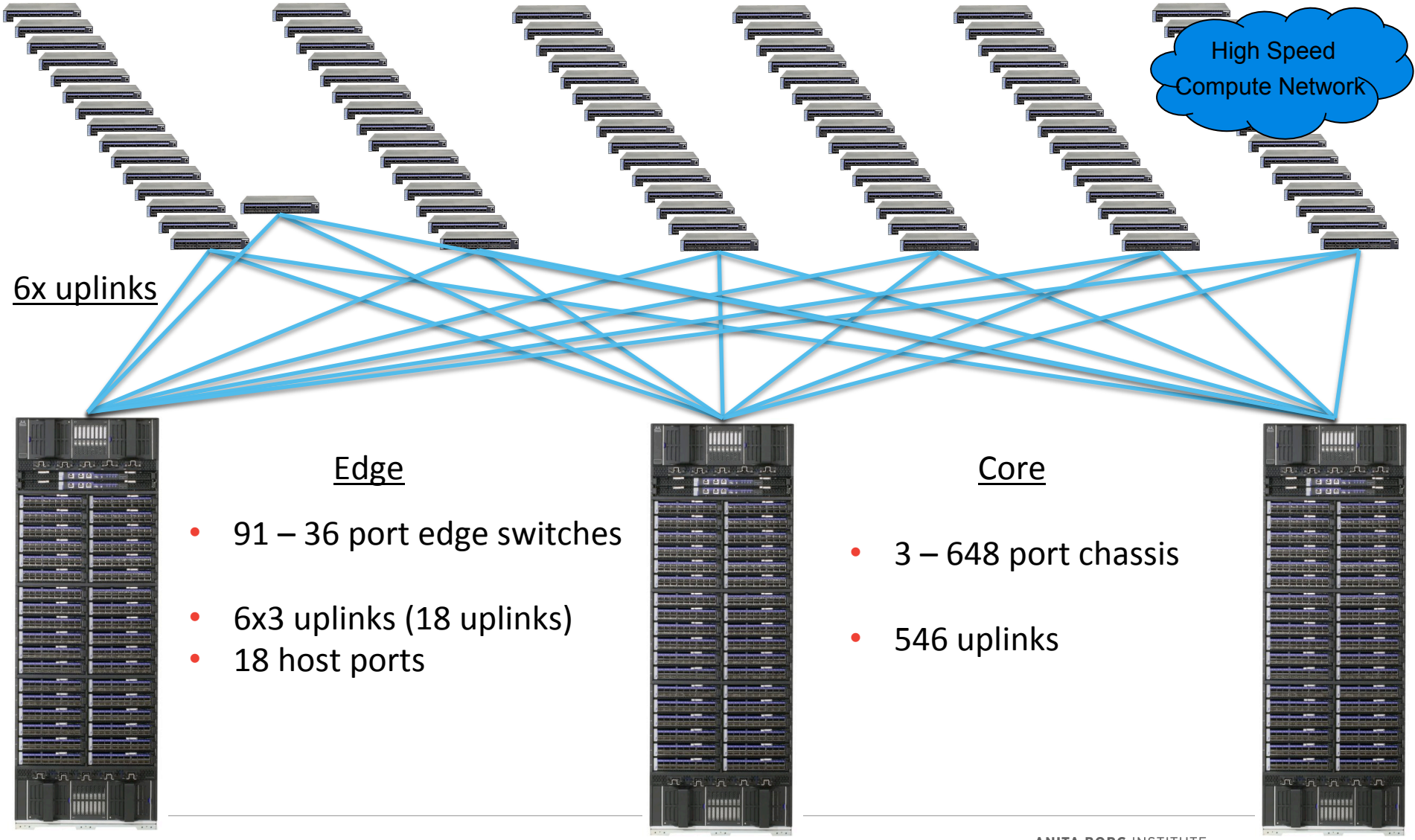
LANL HPC Networks – 15,000 foot view



LANL HPC Network Subsystem – (5,000 foot view)



LANL HPC Network Subsystem – (Cluster HSN)



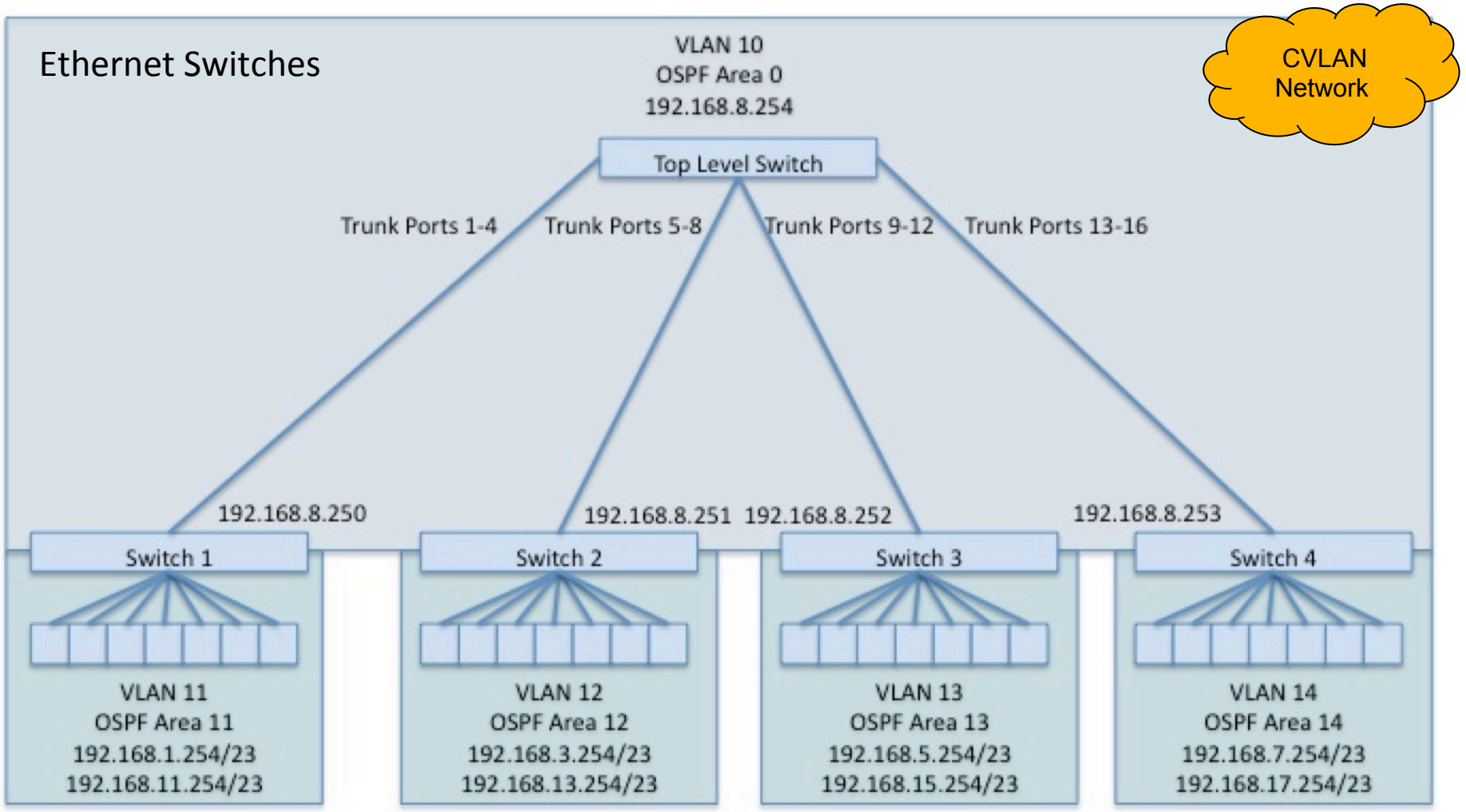
Edge

- 91 – 36 port edge switches
- 6x3 uplinks (18 uplinks)
- 18 host ports

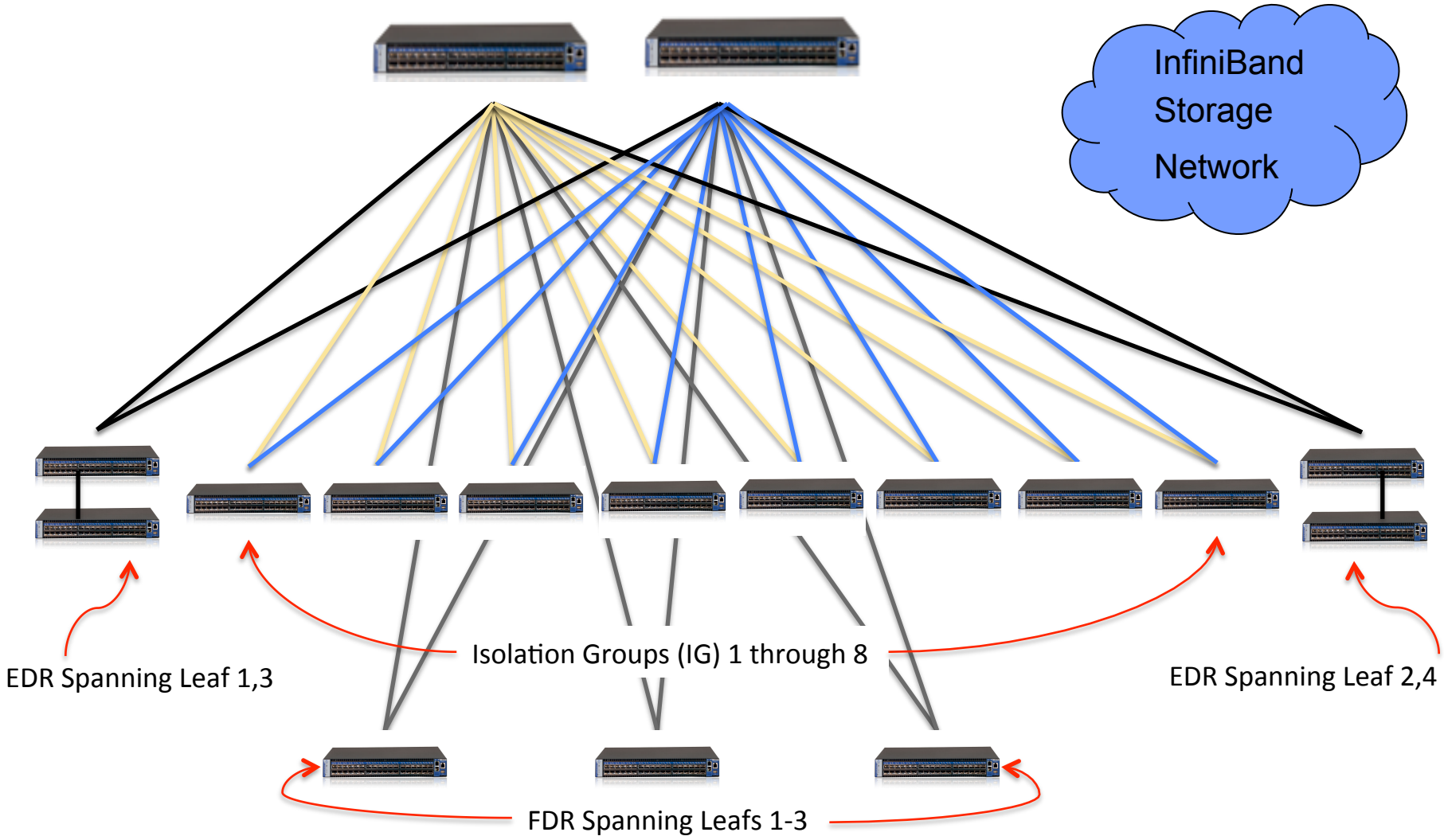
Core

- 3 – 648 port chassis
- 546 uplinks

LANL HPC Network Subsystem – (Cluster VLAN)



LANL HPC Network Subsystem – (Storage SAN)

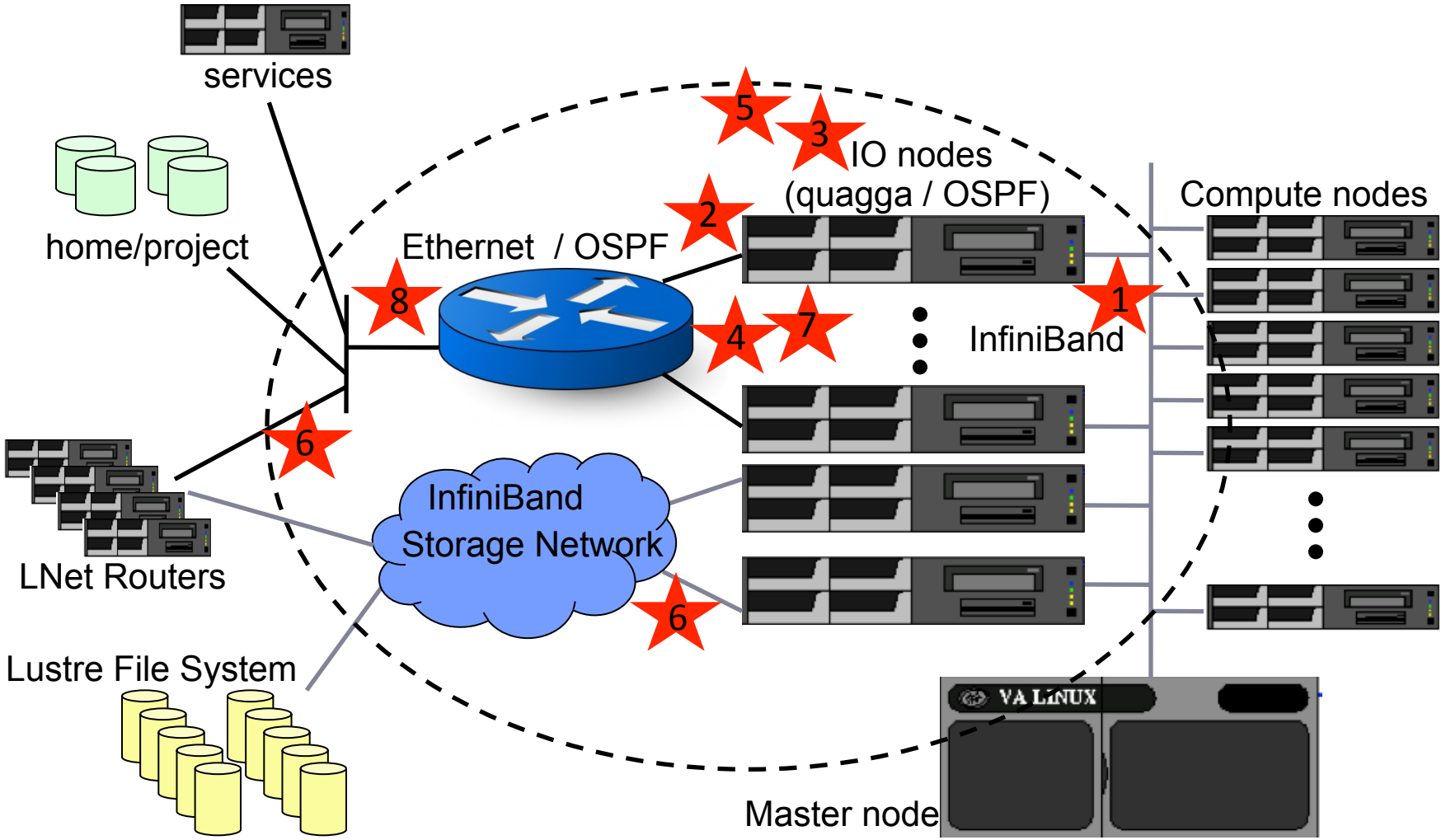


Dead Gateway Detection (DGD)

Purpose: Proactive mitigation of IO subsystem faults

- Define critical and/or weak points in the system
- Monitor those points
- Track any faults in a stateful way
- Report on status
- Perform mitigation actions when faults cross thresholds
  - Automatic process
  - With notifications
  - Transparent to running jobs

DGD: LANL HPC Network Subsystem



DGD: Functional Overview

Process: Daemon runs on boot up

- Set Up
  - Set static variables and function array
  - Read configuration file
  - Pull information on
    - Compute Nodes
    - IO nodes
    - LNet Routers
  - Initialize Health Arrays
- Run
  - Loop through tests as defined in the configuration file
    - Run tests
    - Check results of tests and take action if necessary
- Sleep

DGD: Functional Details

Technology: How it works

- Written in Perl
- Starts on boot up via standard service management files
- Gets hints/directions from a configuration file
  - Each cluster is slightly different
- Tests each component for health
- Tracks complete and partial network connectivity failures
- Some tests launched from the Master Node
- Some tests launched from a random set of compute nodes
- Stateful test results compared to allowed thresholds

DGD: Tests

1. Functionality of High Speed NIC on IO node
2. Functionality of Ethernet NIC or Bond on IO node
3. Status of OSPFD on IO node
4. Ability to reach Gateway on Ethernet backbone switch
5. Error messages from dmesg/syslog of IO node
6. Connectivity of LNet Routers
7. Functionality of Secondary Ethernet IPs on NIC or Bond on IO node
8. Ability to reach Secondary gateway on Ethernet backbone switch

DGD: Bits and Pieces

rpm -qi dgd

Name : dgd
Version : 2.0.0
Release : 2
Install Date: Wed 14 Sep 2016 03:08:40 PM MDT
Group : Applications/System
Size : 43059

Relocations: (not relocatable)
Vendor: (none)
Build Date: Tue 06 Sep 2016 03:41:44 PM MDT
Build Host: ls-fey.lanl.gov
Source RPM: dgd-2.0.0-2.src.rpm
License: LANL

Signature : RSA/SHA1, Mon 12 Sep 2016 08:52:30 AM MDT, Key ID 37cb057e92f976b0
Packager : Susan Coulter <skc@lanl.gov>
Summary : DeadGatewayDetection Package
Description :
Deploys a number of tests, driven by a configuration file, to test various parts of an HPC IO subsystem.

rpm -ql dgd

/etc/init.d/dgd
/usr/sbin/dgd.pl
/usr/sbin/dgd_init.pm
/usr/share/dgd
/usr/share/dgd/dgdcfg.sample

DGD: Capabilities

- Signal handling
  - Dump current health status
  - Re-read the configuration file
  - Wake up from sleep mode and run tests now
  - Force pause
- Code reuse and simplification
  - Functions and arguments in an array
  - Same function for connectivity tests
- Robust
  - Scope of variables
  - Timeouts on all commands

DGD: Code reuse - abstraction

- Reduced the number of lines of code from 97 to 41

```
#=============================================
# check the health of the network subsystem
#=============================================

sub run_tests {
  my $fname = "run_tests";
  my ($cvlan_ping,$node,$test);

# use array of functions to run in configuration defined order
# check status between each set of tests

  for ($test=0; $test<$NUMTESTS; $test++) {

    if ($cfg{DEBUG}) {
      &syslog_write("info","$fname: >>> Test is $TESTS[$test]");
    }

# functions called multiple times with calculated arguments
#    add ssh command to the first element of the argument list array

     if ($TESTS[$test] =~ /TEST_COMPUTE_TO/) {

        my $c;
        my $base_parm = $AL{$TESTS[$test]}[0];
        for ($c=0; $c<$SAMPLE; $c++) {
          my $node = rand($NUM_COMP_NODES);
          my $cvlan_ping = "$SSH_COMMAND $COMPUTE_CVLAN_IPS[$node - 1] ";
          $AL{$TESTS[$test]}[0] = $base_parm.$cvlan_ping;
          $FN{$TESTS[$test]}->(@{$AL{$TESTS[$test]}});
        }
        $AL{$TESTS[$test]}[0] = $base_parm;
```

```
# functions with simple/no arguments

     } else {

        $FN{$TESTS[$test]}->(@{$AL{$TESTS[$test]}});
     }

     &check_for_state_change;
     &reset_counters(@{$AL{$TESTS[$test]}});
```

DGD: Robust – discipline

- Avoids unpredictable errors and difficult debugging

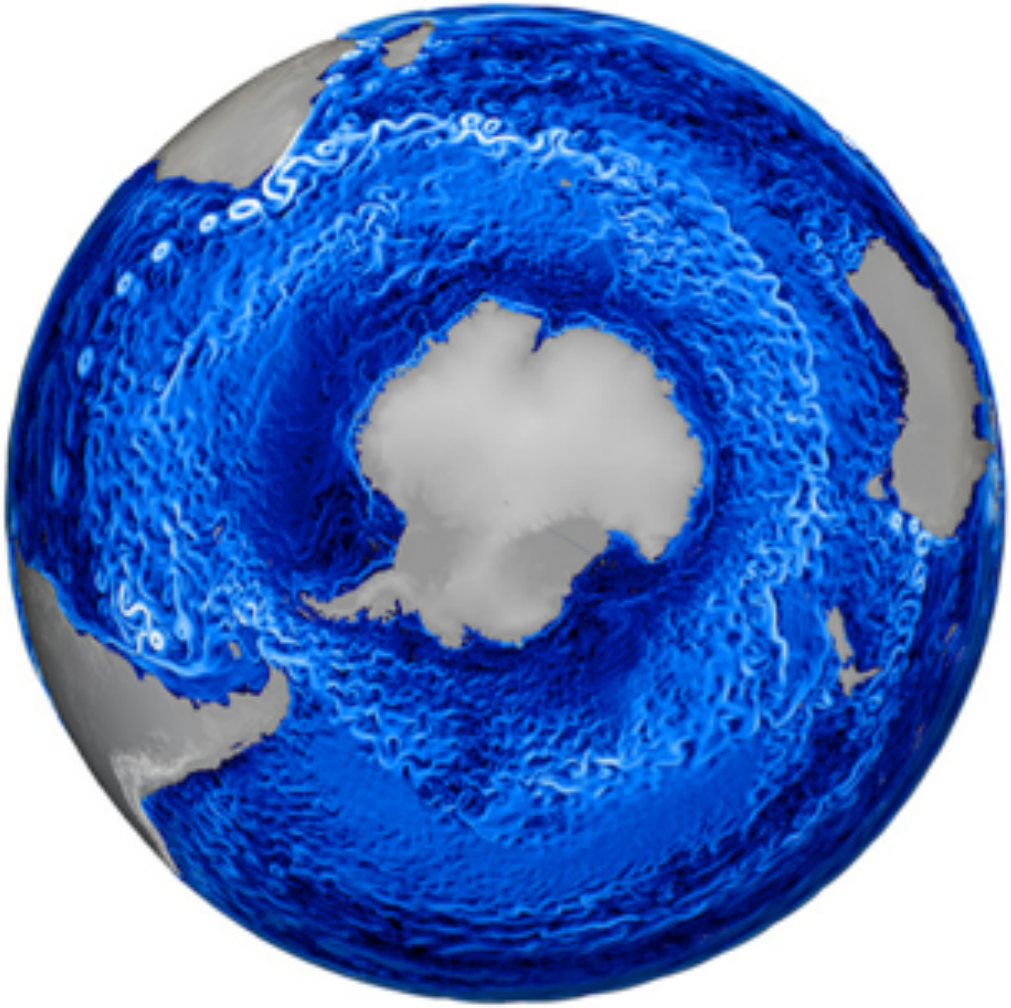
use strict

Strict use of variables:
generates a compile-time error variables are not declared
insures the scope of the variable

Strict use of symbolic references:
avoids confusion
array names and scalar names are distinct

Strict use of “bare words”:
avoids confusion
function names and strings are distinct

Ocean Currents and Eddies Around Antarctica



Thank you

ANITA BORG INSTITUTE
GRACE HOPPER CELEBRATION
OF WOMEN IN COMPUTING

Feedback?

Rate and review the session on our mobile app

Download at http://bit.ly/ghc16app
or search GHC 16 in the app store

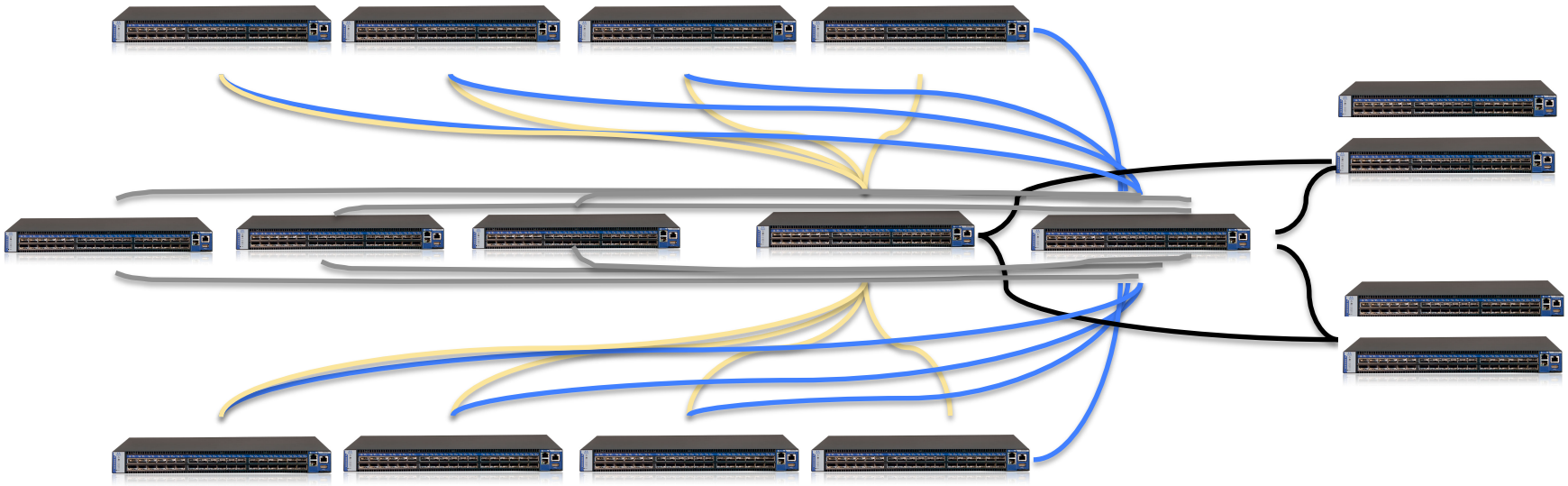
ANITA BORG INSTITUTE
WOMEN TRANSFORMING TECHNOLOGY

acm
Association for Computing Machinery

Backup Slides

HPC Network Subsystem – (Damselfly)

InfiniBand
Storage
Network



HPC Network Subsystem – (Damselfly Details)