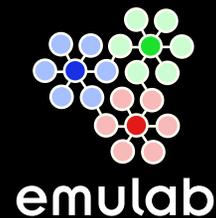
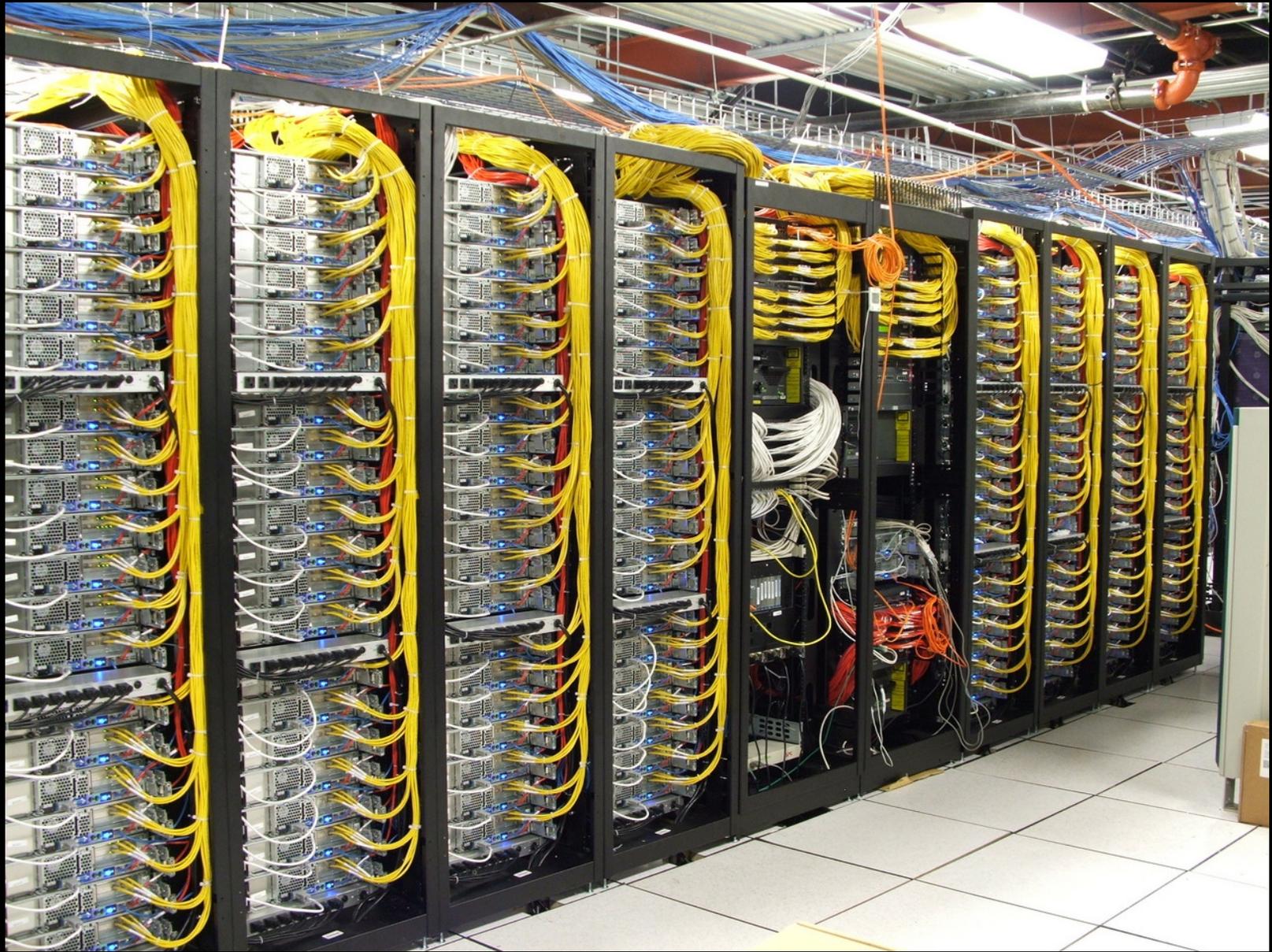


ProtoGENI

and undergraduate courses

Gary Wong







188 Free PCs
6 PCs reloading
13 active users
56 active expts.



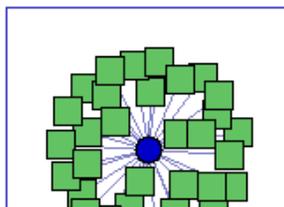
Experiment (tbres/event-test)

'gtw' Logged in.
Sat Jul 07 8:15pm MDT

Experiment Options

- [View Activity Logfile](#)
- [Swap Experiment In](#)
- [Terminate Experiment](#)
- [Modify Experiment](#)
- [Modify Settings](#)
- [Clear Feedback Data](#)
- [Show History](#)
- [Duplicate Experiment](#)
- [Experiment File Archive](#)

188 Free PCs, 6 reloading											
pc600	25	pc850	44	d710	46	pc3000	70	pc2400w	39	pc2000	4
pc3000w	0	d820	0	pc6000	0	pc2400c2	0	pc2400h4	0	gpuhost	0
pc2830qx2	0	pc2400hp	0	d2100	0	rspro	0	e6520	0		



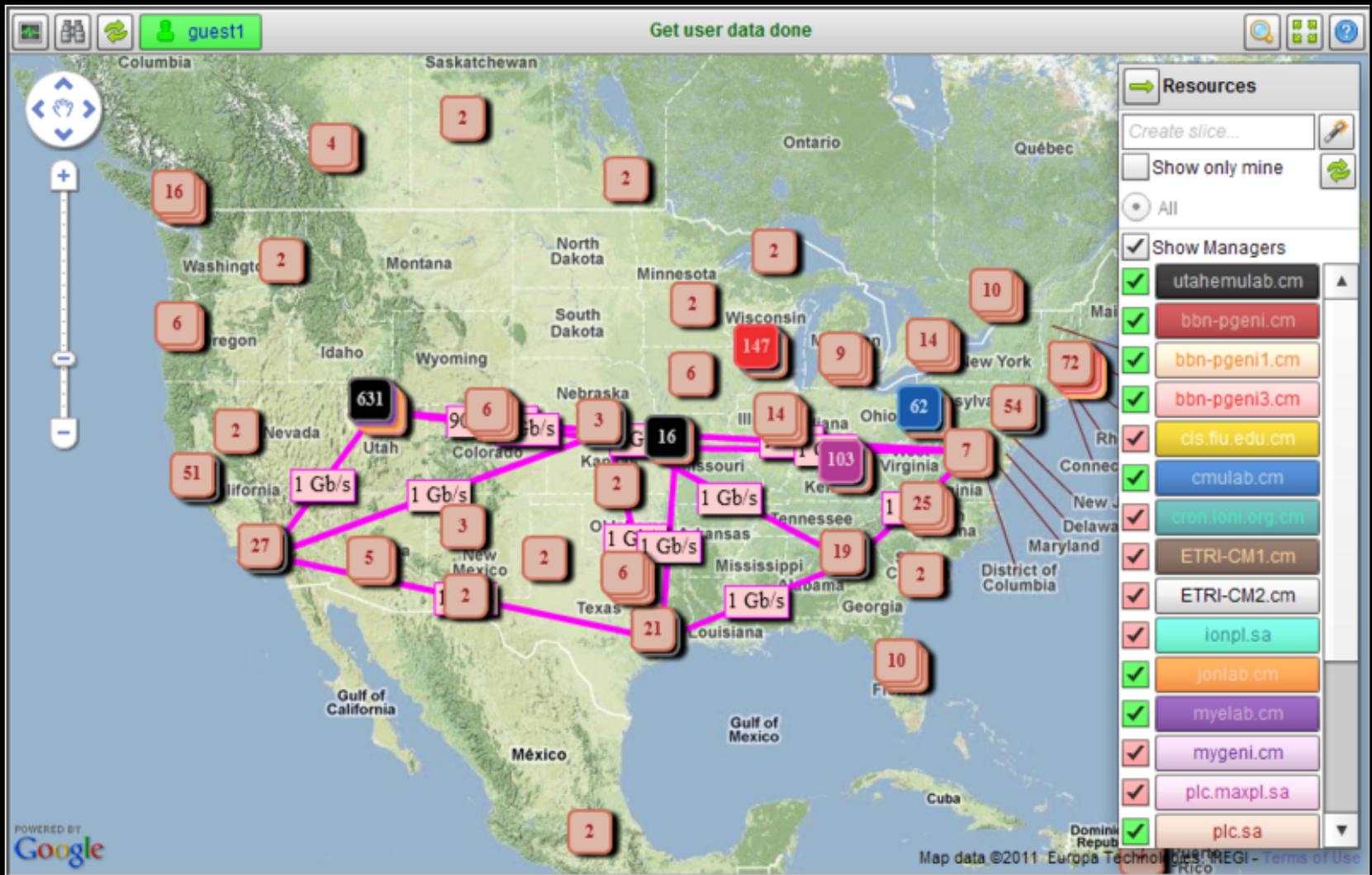
Settings Visualization NS File Details

```

set ns [new Simulator]
source tb_compat.tcl
set maxnodes 32
set tarball "/proj/tbres/duerig/event/node.tar.gz"
set servercmd "/local/node/start-server"
set clustercmd "/local/node/start-cluster clusterd-nofilter"
set clientcmd "/local/node/start-clients 1 5 localhost"
set lanstr ""

# The pubsub server node
set node0 [$ns node]
tb-set-hardware $node0 d710
tb-set-node-os $node0 FBSD72-STD
set server [$node0 program-agent -command "$servercmd"]
set clusters [new EventGroup $ns]
set clients [new EventGroup $ns]

# Agent nodes
for {set i 1} {$i <= $maxnodes} {incr i} {
    set node($i) [$ns node]
    tb-set-hardware $node($i) d710
    tb-set-node-os $node($i) FBSD72-STD
    tb-set-node-tarfiles $node($i) /usr/testbed/lib $tarball
    set cluster($i) [$node($i) program-agent -command "$clustercmd"]
    set client($i) [$node($i) program-agent -command "$clientcmd"]
    append lanstr "$node($i) "
    $clusters add $cluster($i)
    $clients add $client($i)
}
    
```



Getting started

- Professor creates new project
 - approved by testbed admin
- TAs then join that project (group leader)
 - approved by the project head
- Students also join
 - approved by the project head or TA

Education

- Emulab used in a couple of dozen classes
- ProtoGENI used in two

Assignments

- Put everything in an NS file or RSpec, give it to all students
- Can include software packages and dependencies
- More advanced classes can have students create their own topologies

Grading

- Run every student's code in a consistent environment
- Few concerns about side effects or malice

Environment

- Forgiving!
- (Almost) nothing a student can do that can't be recovered easily

Courseware

- Standard environment no matter what university the class is taught at
- Can adapt textbook exercises to GENI/Emulab
 - This was done for a class at Calvin College

<http://www.emulab.net/>

<http://www.protogeni.net/>

<http://users.emulab.net/trac/emulab/wiki/Classes>

<http://cs.calvin.edu/activities/emulab/comer.html>