
Puppet for Mac Workstation Configuration Management

Overview

- ◆ Problem Overview
- ◆ Alternatives
- ◆ Architecture
- ◆ Bootstrapping
- ◆ Puppet script examples
- ◆ Tips & Tricks
- ◆ Conclusions

Problem Overview

- ◆ No large scale configuration mgmt solution for Macs
- ◆ Need inventory data
- ◆ Need usage data
- ◆ Need configuration change history
- ◆ Way too much variability on our Mac workstations

Problem Overview - Specifics

- ◆ What Macs have disappeared
- ◆ What Macs need to have Office installed
- ◆ What Macs need memory upgrades before 10.5 install
- ◆ What Macs are at Inglemoor High School
- ◆ What Macs are in a video lab and need a iMovie update
- ◆ Which Macs being taken home
- ◆ How to convert Macs from OD to AD authentication

Alternatives

- ◆ CFEngine
 - ◆ Required too much TLC
 - ◆ No built in feedback mechanism
- ◆ Bcfg2
 - ◆ Good feedback mechanism
 - ◆ Did not seem to be as well tested on Mac
- ◆ Apple tools
 - ◆ Do not scale

Architecture - Puppet Basics

- ◆ Created by Luke Kanies
- ◆ Facter collects facts about a client
- ◆ Resources to represent files, services, scripts, etc.
- ◆ Providers map resources to client OS tools
- ◆ Classes are groups of resources
- ◆ Puppetmaster then uses facts to determine which classes/resources should be applied to a client

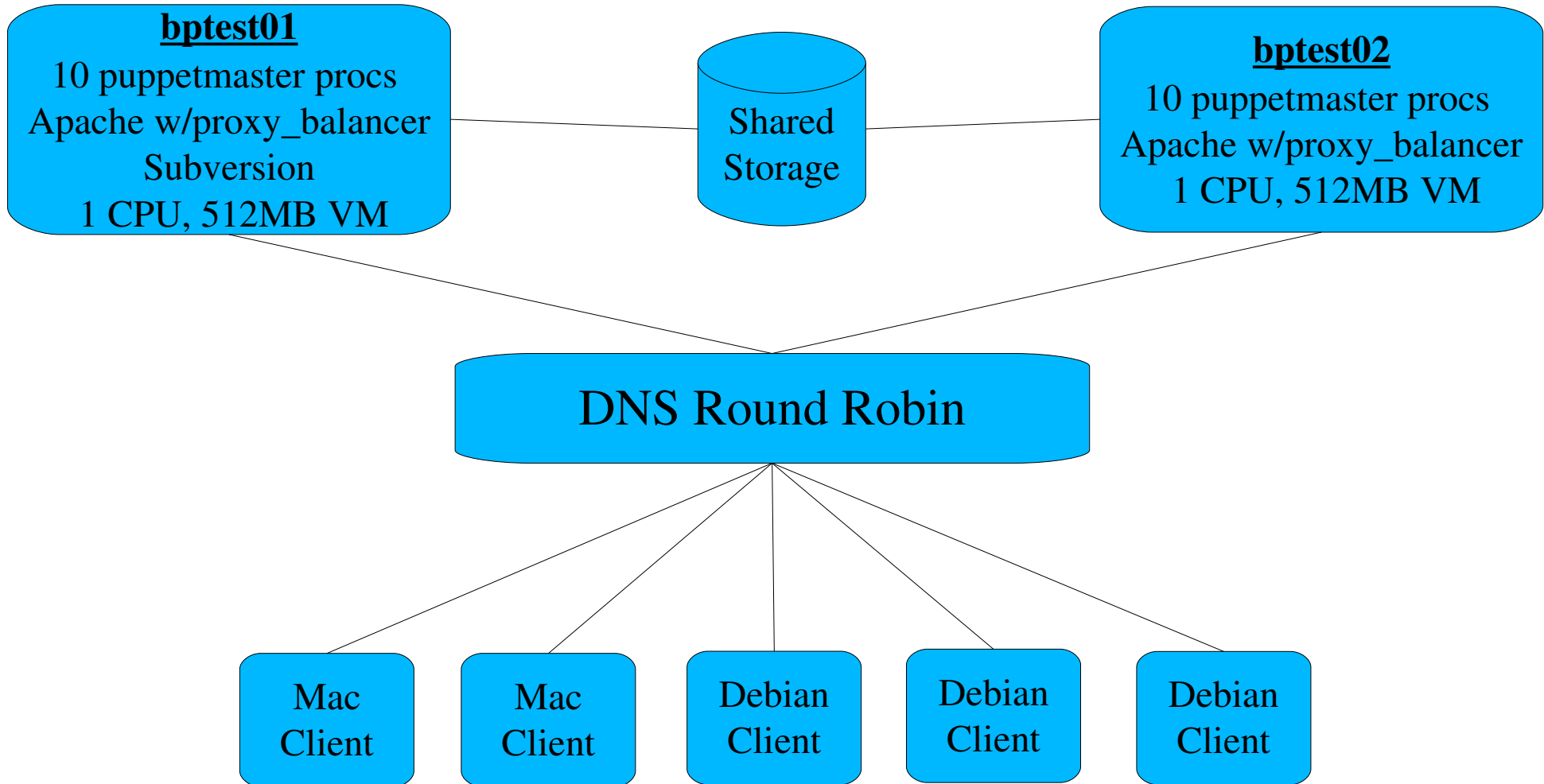
Architecture - Puppet & Ldap

- ◆ Can use Ldap to store classes
- ◆ Found out that you cannot store facts in Ldap
- ◆ Found out that it still uses the disk even if Ldap selected
- ◆ Decided not to use Ldap as no perceived benefits

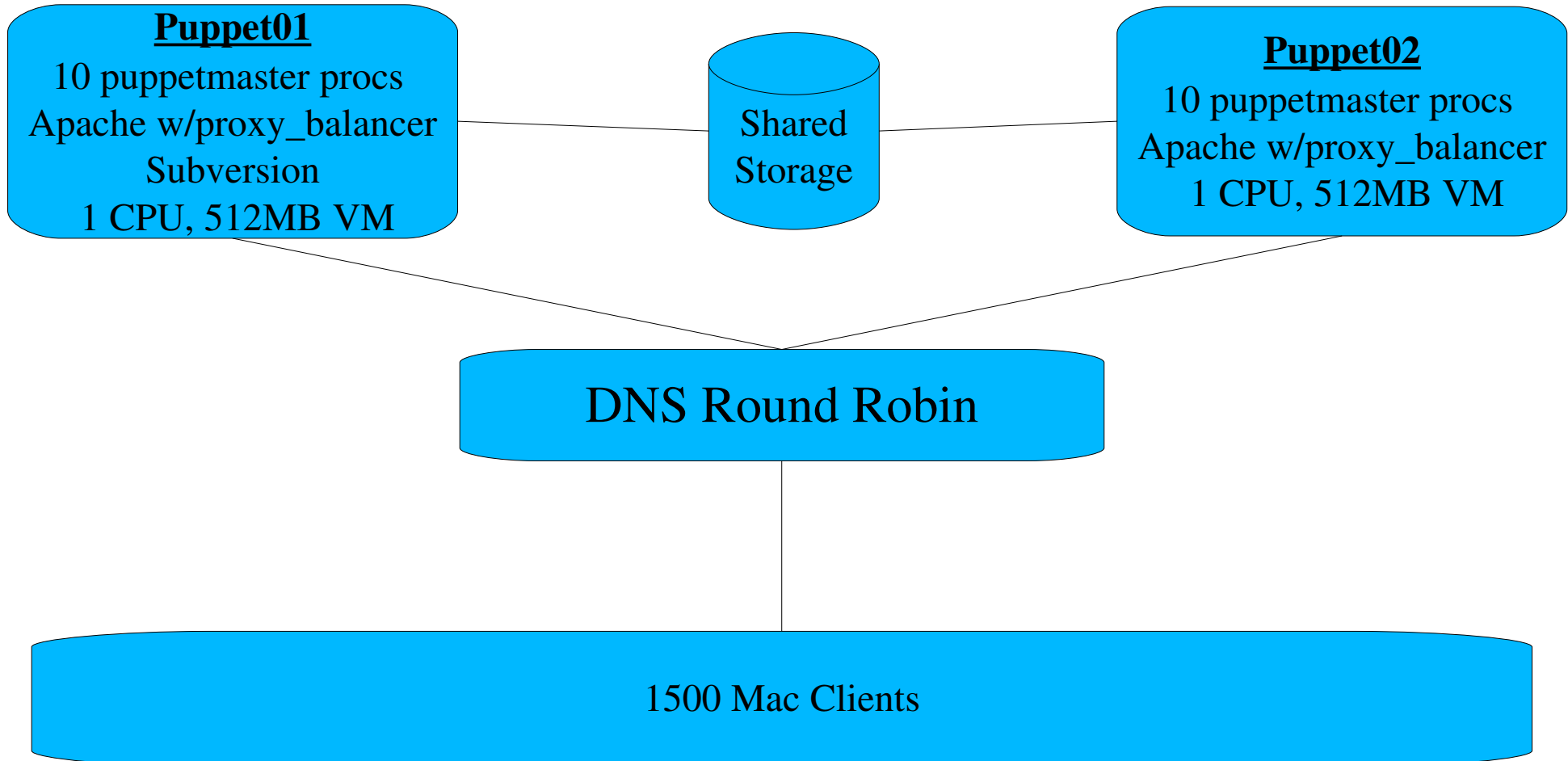
Architecture Ideas

- ◆ Keep machine specific data in local config file
 - ◆ Machine role
 - ◆ Primary & secondary users
 - ◆ Assettag number
- ◆ Push svn.rev with current svn revision
- ◆ /etc/puppet/checkfiles for lock files
- ◆ SVN post-commit hook, updates puppetmasters
- ◆ SVN pre-commit hook to test syntax

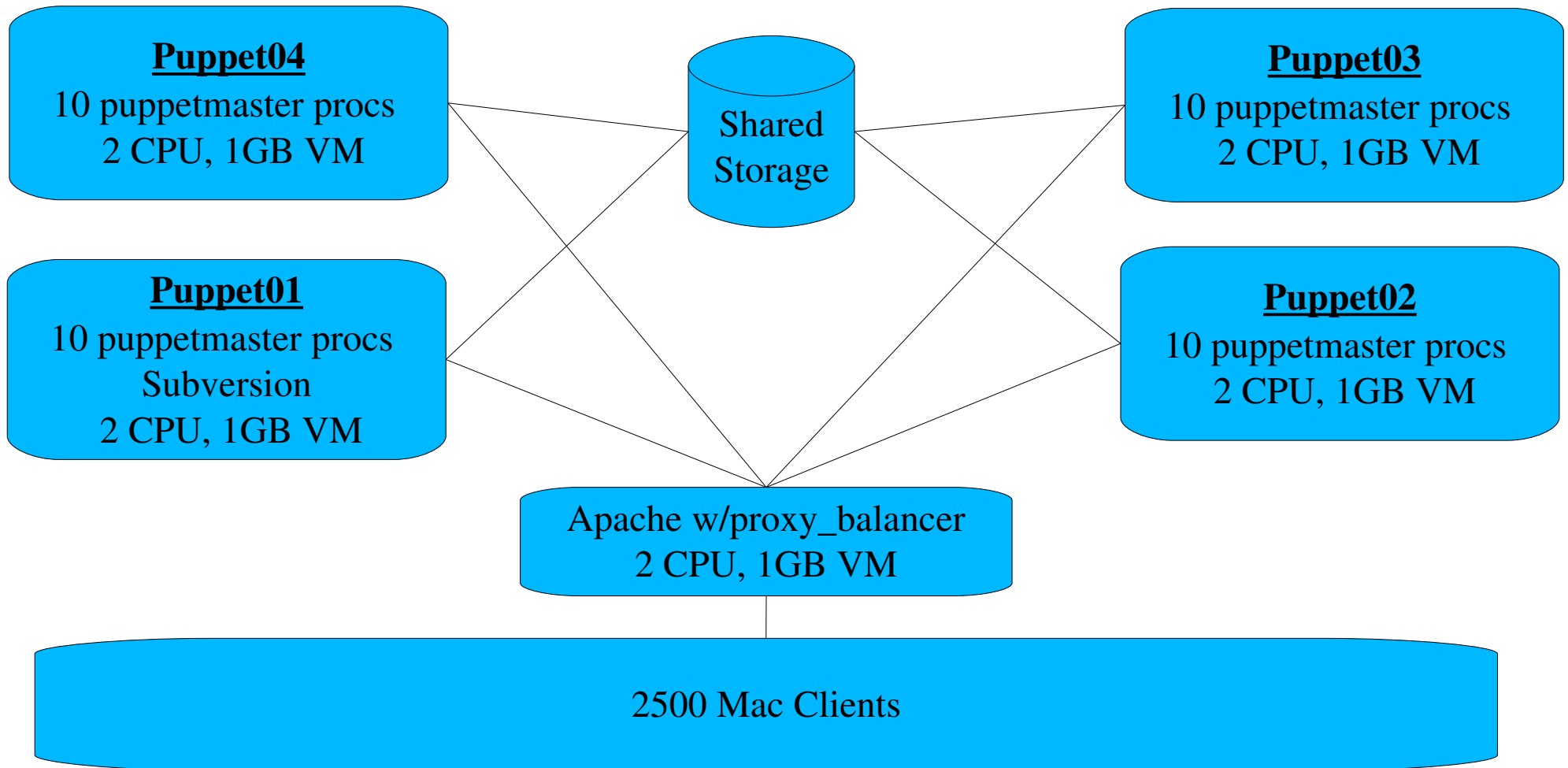
Architecture - Testbed



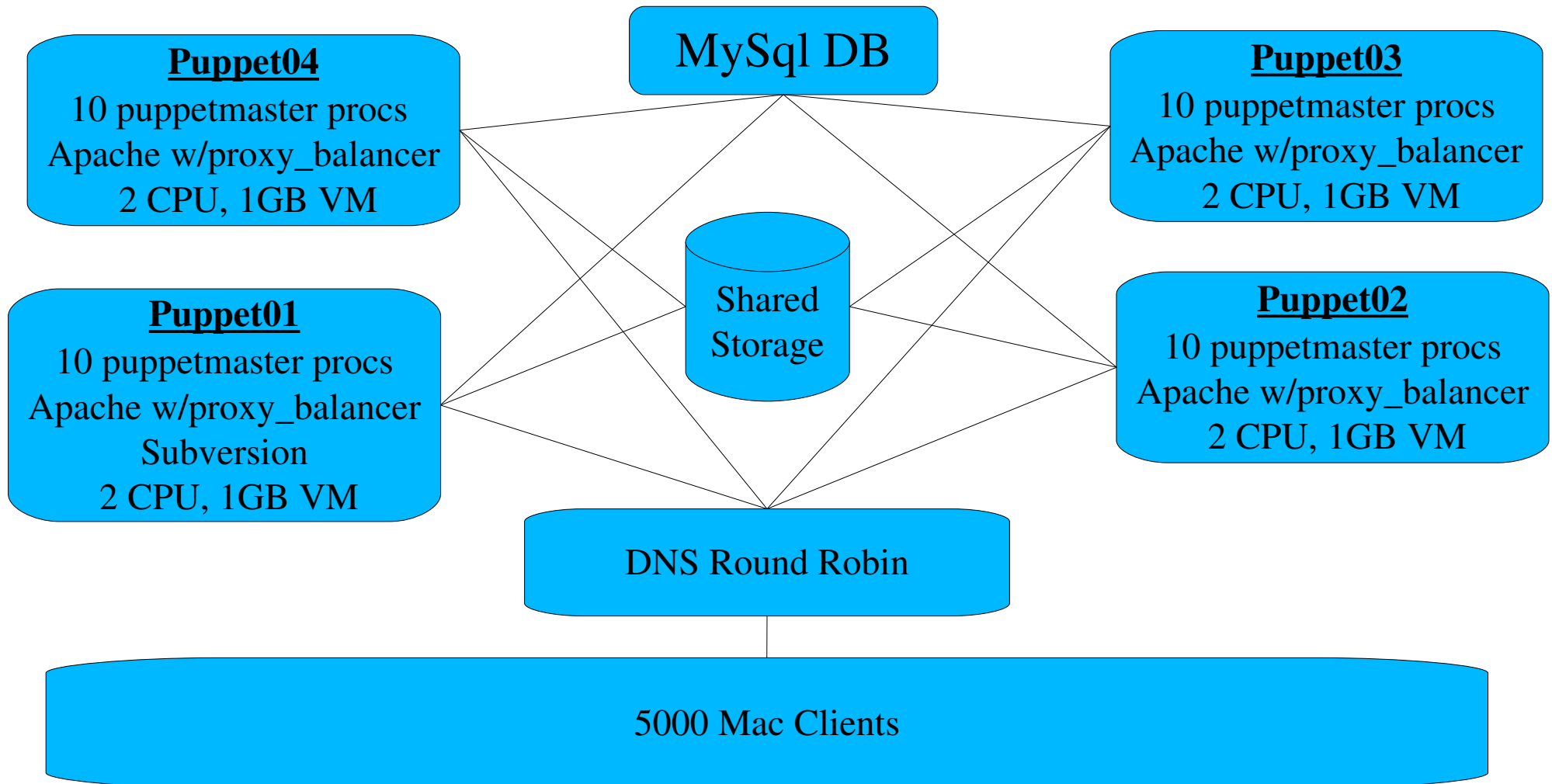
Architecture - Production system V1



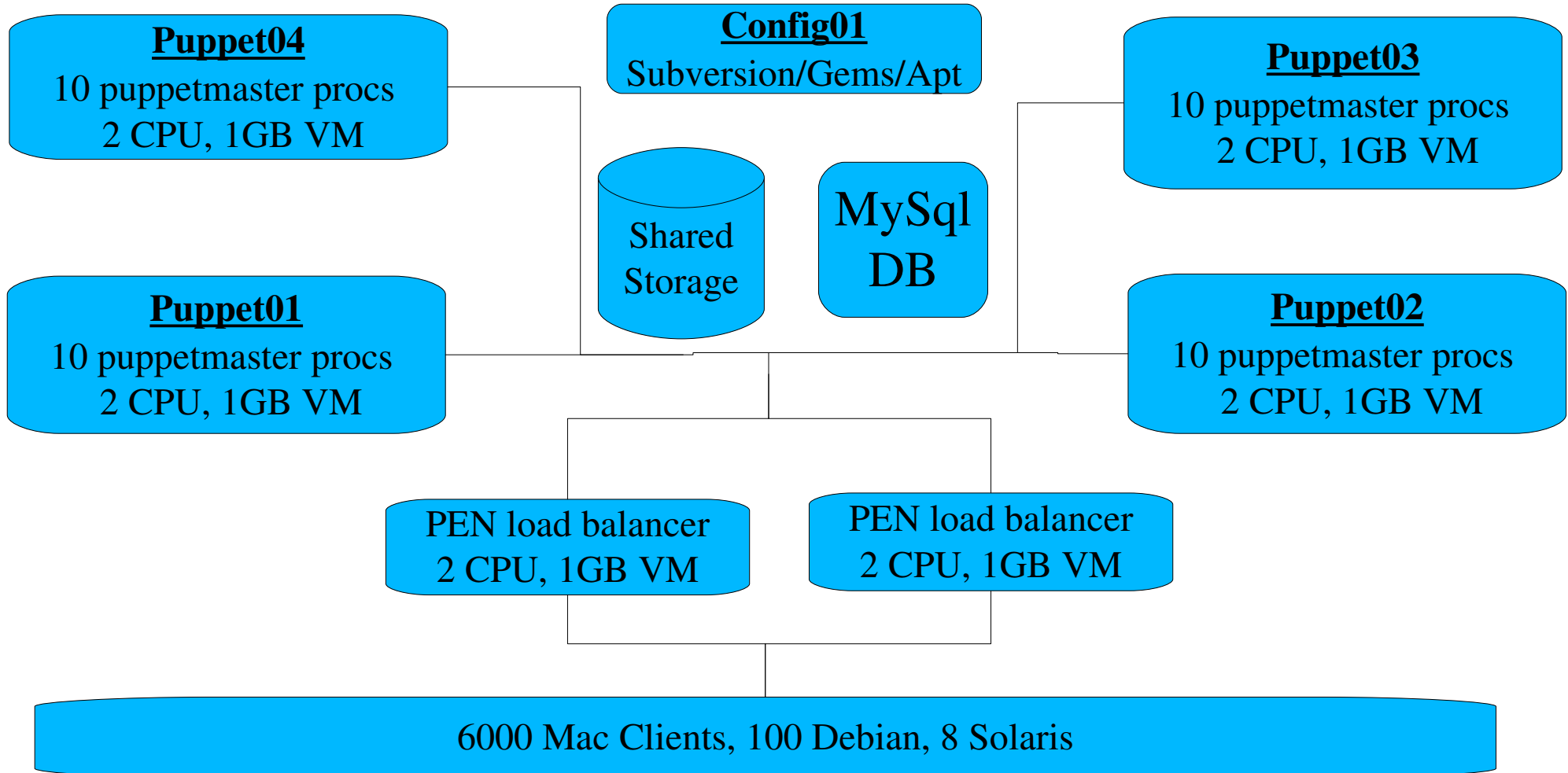
Architecture - Production system V2



Architecture - Production system (Current)



Architecture - Production system (Future)



Bootstrapping

- ◆ Auto fill in nsdfacts file from computer name
- ◆ Handle different ruby location on 10.3 Macs
- ◆ Copy systemsetup & networksetup to common directory
- ◆ Recover from partial installs
- ◆ Create puppet user/group
- ◆ Install ruby-gems, facter, & puppet
- ◆ Separate scripts for staff & student Macs
- ◆ Use ARD to push out

Bootstrapping Issues

- ◆ Workstations not turned on
- ◆ 10.3 default ruby did not work
- ◆ Operator error
- ◆ Wide range of passwords on Mac
- ◆ rc.local did not work for starting on reboot
- ◆ ARD is very unstable

Puppet Scripts – darwin.pp

```
class darwin {
  notice("Reached class: darwin")
  notice("site=${ site };role=${ role };assettag=${ assettag };majorver=${
{ macosx_productversion_major };ipaddress=${ ipaddress }")
  file { "/etc/puppet/checkfiles":
    ensure => directory,
  }
  file { "svn.rev":
    name => "/etc/puppet/checkfiles/svn.rev",
    source => "puppet://puppet/files/svn.rev",
  }
  file { "puppet.conf":
    name => "/etc/puppet/puppet.conf",
    source => "puppet://puppet/files/workstation/app/puppet/puppet.conf",
  }
}
```


Puppet Scripts - darwin.pp

```
include pmset
include darwinpw
case $macosx_productversion_major {
  "10.3": { include removecfengine }
}
file { ["/Library/StartupItems/Puppet":
  source => $macosx_productversion_major ? {
    "10.4" => "puppet://puppet/files/workstation/app/puppet/Puppet.10.4",
    "10.3" => "puppet://puppet/files/workstation/app/puppet/Puppet.10.3",
  },
  recurse => true,
  ensure => directory,
  owner => root,
  group => wheel,
}
}
```

Puppet Scripts – removecfengine.pp

```
class removecfengine {
  exec { "/usr/bin/killall -9 cfagent":
    onlyif => "/bin/ps ax | /usr/bin/grep cfagent | /usr/bin/grep -v grep" }
  exec { "/usr/bin/killall -9 cfenvd":
    onlyif => "/bin/ps ax | /usr/bin/grep cfenvd | /usr/bin/grep -v grep" }
  exec { "/usr/bin/killall -9 cfexecd":
    onlyif => "/bin/ps ax | /usr/bin/grep cfexecd | /usr/bin/grep -v grep" }
  exec { "/usr/bin/killall -9 cfservd":
    onlyif => "/bin/ps ax | /usr/bin/grep cfservd | /usr/bin/grep -v grep" }
  exec { "/bin/rm -f /var/cron/tabs/*":
    onlyif => "/bin/test -e /var/cron/tabs/root" }
  file { "/etc/rc":
    name => "/etc/rc",
    source => "puppet://puppet/files/workstation/etc/rc",
    owner => root,
    group => wheel,
  }
}
```

Puppet Scripts – darwinpw.pp

```
class darwinpw {
  file { "RootChgPW.exp":
    name => "/usr/bin/RootChgPW.exp",
    source =>
    "puppet://puppet/files/workstation/bin/RootChgPW.exp",
  }
  file { "OFPW":
    name => "/usr/bin/OFPW",
    source => "puppet://puppet/files/workstation/bin/OFPW",
  }
  exec { "/usr/bin/RootChgPW.exp root *****":
    creates => "/etc/puppet/checkfiles/password"
  }
  exec { "/usr/bin/RootChgPW.exp nsadmin *****":
    creates => "/etc/puppet/checkfiles/password"
  }
  exec { "/usr/bin/OFPW -pass *****;/usr/bin/OFPW -mode 0":
    creates => "/etc/puppet/checkfiles/password"
  }
}
```

Puppet Scripts - darwinpw.pp

```
exec { "/usr/bin/touch /etc/puppet/checkfiles/password":  
  creates => "/etc/puppet/checkfiles/password",  
  unless => "/bin/ps ax | /usr/bin/grep -e cf[eas]"  
}  
}
```

Puppet Scripts – pmset.pp

```
class pmset {
  case $site {
    ihs: { $time = "5:55:00" }
    bhs: { $time = "5:50:00" }
    whs: { $time = "5:45:00" }
    ....
    default: { $time = "5:00:00" }
  }
  exec { "/usr/bin/pmset repeat wakeorpoweron M $time shutdown F 17:00:00":
    creates => "/etc/puppet/checkfiles/pmset"
  }
  exec { "/usr/bin/touch /etc/puppet/checkfiles/pmset":
    creates => "/etc/puppet/checkfiles/pmset"
  }
}
```

Tips & Tricks & Lessons

- ◆ puppet -parseonly before commit
- ◆ puppet -verbose -report -debug -no-daemonize to debug
- ◆ Beware of caching by puppetmaster, may need restart it
- ◆ Needed more testing before going into production on 10.3
- ◆ Can use puppet to install its own config file
- ◆ Variances during bootstrap process surprised us

Current status

- ◆ Installed on 5000 macs (1000 to go)
- ◆ 10.3 macs may still have problems
 - ◆ Need to wait until weekend reboot due to ARD bug
- ◆ Facts stored in mysql database
- ◆ Trigger on database create log table of old fact values
- ◆ Passwords, cfengine removal, startup times working

Future Plans

- ◆ Manage sshd on workstations
- ◆ Change all sharing names to asset tag number
- ◆ Change mac authentication from OD to AD
- ◆ Email alerts when machine not checking in
- ◆ Application to allow changes to nsdfacts file
- ◆ Install software as needed
- ◆ Change MS Office from site to per machine licensing
- ◆ Track file changes in database

Conclusions

- ◆ Mac workstations were more difficult than expected
- ◆ Works well once it is installed
- ◆ Database feedback is really cool
- ◆ Still has several rough edges, but very usable
- ◆ Ability to write own facts allows easy customization
- ◆ Looking forward to Luke's tool to get status reports

Some Puppet Users

- ◆ www.avvo.com
- ◆ www.google.com
- ◆ www.photoblog.com
- ◆ www.tvtservices.ch
- ◆ www.wordtracker.com
- ◆ www.citysafe.org
- ◆ www.fab-it.eu
- ◆ www.sitesuite.com.au
- ◆ www.joost.com
- ◆ www.fedora.com
- ◆ www.speakeasy.net
- ◆ www.arces.net
- ◆ www.ingent.net
- ◆ www.maxnet.co.nz
- ◆ www.powerset.com
- ◆ www.sans.org

References

- ◆ Puppet website: <http://reductivelabs.com/trac/puppet>
- ◆ Puppet on radio: <http://www.redmonk.com/cote>
- ◆ Complete setup: <http://live.lab42.it/puppetinfrastructure>
- ◆ Google at Macworld:
 - ◆ <http://code.google.com/p/puppet-mw08/downloads/list>
- ◆ “Pulling Strings with Puppet: Configuration Management made Easy” by James Turnbull

Special thanks to Mark Foster at BitPusher for setting up NSD's Puppet architecture and Kristi Rimkus for bootstrapping Puppet