

# Brainstorming Thin-provisioned Tier 3s



- Puppet is fairly straightforward to set up in an “masterless” mode.
  - Simply distribute Puppet rules to nodes via VCS (git, mercurial, SVN, whatever)
  - Some more reading:  
<http://bitfieldconsulting.com/scaling-puppet-with-distributed-version-control>
- Common software installed via git repository, applied locally. essentially:
  - `$ yum install puppet`
  - `$ git clone http://github.com/ATLASConnect/Tier3`
  - `$ vim Tier3/init.pp`
  - `$ puppet apply Tier3/init.pp`
- Voila?



- Large knowledge base to draw upon
  - MWT2 and UC ATLAS Tier 3 are managed via Puppet
  - Happy to share these puppet rules with the community
- Will be available via Github repo
  - Contributions welcome
- Easy to cherrypick certain features
- We are planning to build a few T3 interactive/analysis boxes and will lay the groundwork for this



- Transparent kernel block layer cache.
  - Problem:
    - SSDs = Fast but low capacity
    - Spinning disks = Slow but high capacity
  - Solution: Put SSDs in front of RAID array to cache reads/writes
    - Totally transparent to the filesystem
    - Write-through, write-back, and write-around modes available
    - Save \$\$\$ by moving high-end hardware RAID features into kernel space
- Considered stable in kernel 3.10+
  - Might be too aggressive for some conservative sites
  - At MWT2 we routinely use kernel 3.11 on interactive nodes.



- Configure T3 nodes with a large disk array, optionally use kernel bcache for performance boost.
- Use FAX to read data off of the grid
- Cache data locally with XRootD sub-file level caching
  - New feature in XRootD 4



- T3 puppet rules are pre-configured to install Condor & flock to ATLAS Connect
- Need to figure out authentication
  - Should aim to be as lightweight as possible
  - X509 is greatly not-preferable, as a host cert/key pair has to be generated for each site.
  - Shared password viable? Seems reasonable if password is stored in a private repository on GitHub



- Couple of different approaches we could take
  - Red Hat Spacewalk
    - Build a SL image for ATLAS Tier 3 nodes
    - Essentially deprecated at this point. RH is focused on cloud deployments, i.e., Openstack + Puppet/Chef/Ansible/Configuration management tool *du jour*
  - ATLAS T3 RPMs
    - Wrap up some common configs and RPMize them. i.e.,
      - ATLAS Connect condor config?  
→ condor-atlas-connect.rpm
      - Preconfigured XRootD Cache w/ FAX upstream redirection?  
→ xrootd-atlas-fax-cache.rpm
    - More flexible but probably overall more work for little gain
  - Centrally managed Puppet
    - Puppet daemon is advantageous since it runs every 15 minutes, easy to push out updates
    - Easy enough to write a cron to do similar task