

XRootD Clustering Amid Container Chaos

XRootD Workshop

IN2P3, Lyon

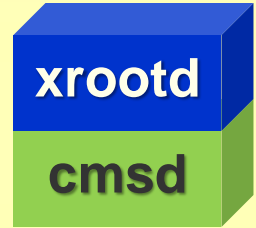
June 11 - 12, 2019

Andrew Hanushevsky, SLAC

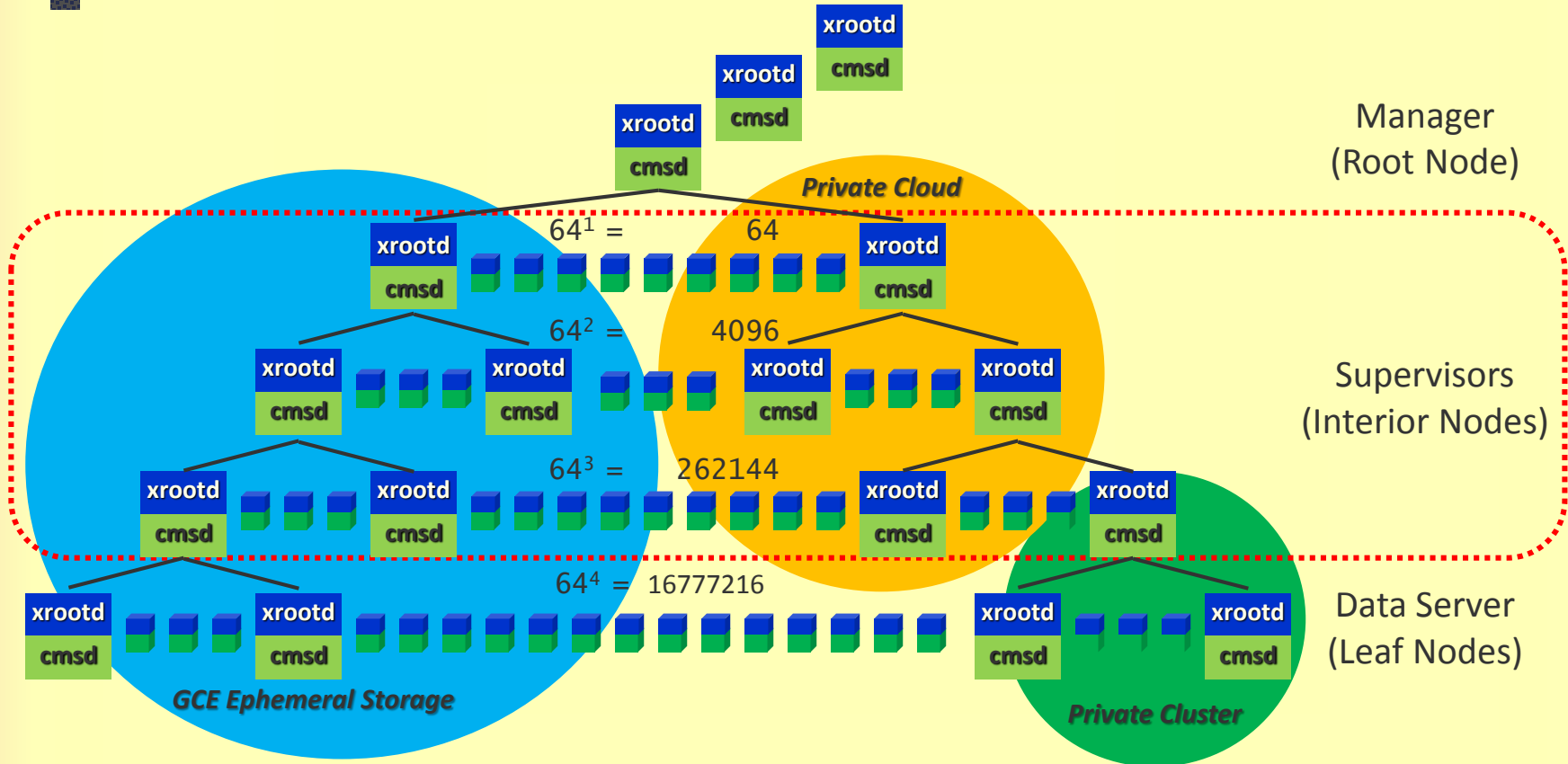
<http://xrootd.org>

Introduction

- # The **XRootD** clustering allows scaling
 - It works extremely well on the LAN
 - Typically used for proxy & cache clustering
 - Today **XRootD** servers are containerized
 - This includes xrootd and cmsd daemons
 - But, container frameworks can hinder reliability
 - Kubernetes, swarm, etc
- # Let's see why and how to solve it



XRootD Cluster Refresher



The cmsd is responsible for tracking file location.

Tracking File Location

- # Each lower level cmsd reports to parent
 - If it actually has the file a client wants
- # The parent caches the response
 - The host name, IP address, file name
- # Over time cluster knowledge is built up
 - The manager knows where desired files are

What's the Problem?

- # Container managements schemes and how they handle container restarts
 - IP addresses are arbitrary and reassigned
 - New host name may be manufactured or not registered in a timely fashion
- # Recall cmsd tracks files by host name & IP
 - Now that becomes undependable
 - Cached location information may be associated with the wrong container (i.e. server) after restart.

Solution?

- # Overlay network with a name space
 - Essentially we construct a virtual network
 - Where each “node” has an internal name
- # The cmsd now can then track file location by a dependable name
 - Does not rely on host name or IP address

The cms.vnid Directive (R 4.9)

`cms.vnid {=id | <path | @libpath [parms]}]`

- *=id*
 - The actual 1-to-64 character name node has
- <*path*
 - File system path of a file containing the name
- @*libpath*
 - Path to plug-in that will supply the name.

What cms.vnid gives you

- # Assigns a name to a resource provider
 - Resources are tracked using the name
 - Specifically, file location
 - cmsd verifies that it's xrootd has the same name
 - They may differ if the cmsd and xrootd are in different containers and erroneously connect to the wrong counterpart after being restarted
 - Unlikely but possible under some scenarios
 - It's essential what the cmsd sees is what the xrootd actually exports

Simply naming servers?

Not at all!

- cmsd doesn't care what server it is as long as it has what it claimed to have in the past
- In practice, the virtual network ID should be assigned to the disk that holds the files
 - It doesn't matter which server exports that disk
 - We just need to pin the file to a disk location
- Recommended practice is to place a file holding the vnid on the disk the server will export
 - Then in the configuration point to that file

Conclusion

- # The `cms.vnid` directive gets around container craziness
 - It insulates the cluster from the vagaries of container management systems
- # If your clusters servers are containerized and managed by any framework you should specify the `cms.vnid` directive
 - Or use host networking though that isn't always a clean or workable solution.