Accelerating Debugging In A Highly Distributed Environment

CHEP 2015
OIST Okinawa, Japan
April 28, 2015

Andrew Hanushevsky, SLAC

http://xrootd.org

Motivation

- Deploying a highly distributed system
  - It's federated across dozens of sites
    - Sites are administratively independent
    - The system competes with other site priorities
      - Getting site attention can sometimes be difficult
  - Site expertise is highly variable
  - Sites span almost all time zones
- How do you resolve a site problem?

Problematic Approaches

■ ssh to the site's problem node
  - Multiple policies, accounts, password, etc
    - This is not scalable
    - May not even be given sufficient access
■ E-mail the site admin
  - Painfully slow iterative process
    - The problem may disappear before you are done
■ There must be a better way

SLAC
NATIONAL ACCELERATOR LABORATORY

What Do You Really Need?

- Most of the time resolution simply needs
  - Ability to look at the log files
  - Check the configurations files
  - Verify system settings
  - Perhaps debug with a core file
- Remote access without site peculiarities
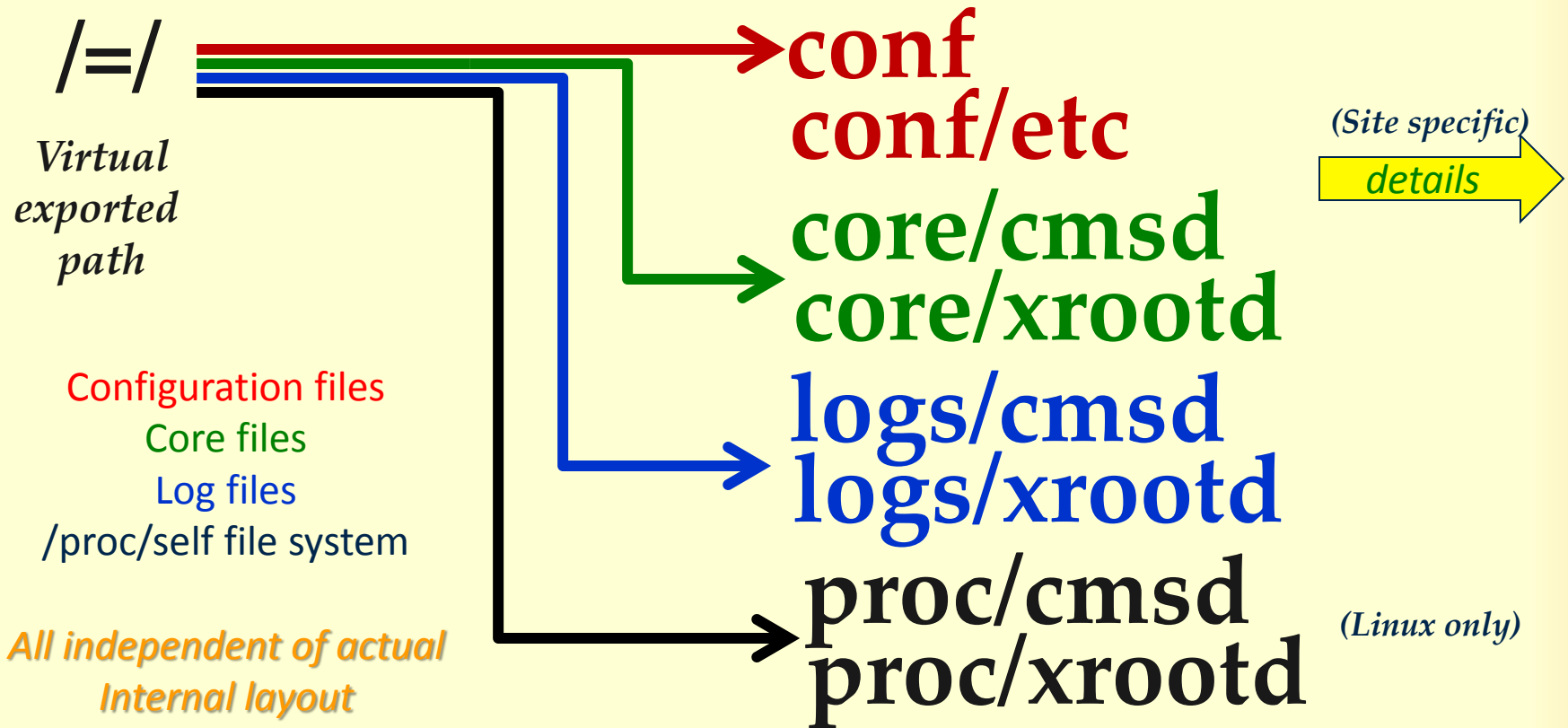 - That is, independent of the site's setup
    - File naming and locations

SLAC
NATIONAL ACCELERATOR LABORATORY

Remote Debugging Is Not New

- Almost all browsers have this capability
  - Standardization being attempted
    - http://remotedebug.org/
- Windows remote desktop
- Remote gdb via gdbserver
- There are many more available schemes
  - Usually overly intrusive
  - Require expertise on both ends to use

SLAC
NATIONAL ACCELERATOR LABORATORY

Minimalist XRootD Approach

- ⌗ An exportable pseudo file system - **digFS**
  - ■ Uniformly lays out all important information
- ⌗ Site admin has complete control
  - ■ What information to export
  - ■ Who can actually see what *details* →
    - ■ Using strong authentication & authorization rules
      - ■ Detail logging of accesses and denials
- ⌗ Information is strictly read-only
  - ■ Nothing in the site can be changed

The **digFS** View

/=/

Virtual exported path

conf
conf/etc

(Site specific)

details →

core/cmsd
core/xrootd

logs/cmsd
logs/xrootd

Configuration files
Core files
Log files
/proc/self file system

proc/cmsd
proc/xrootd

(Linux only)

All independent of actual Internal layout

Deployment Not A Slam-Dunk

- Firewalls may interfere
  - Standard proxies are of little help
    - Looking to see what can be done here
- Resistance to even minimalistic scheme
  - Some sites forbid any kind of remote access
  - Some admins dislike any kind of intrusion
- But, whatever we get is better than nothing

SLAC
NATIONAL ACCELERATOR LABORATORY

What The Brave Would Like

- Additional useful abilities…
  - Change certain directive settings
    - Tracing; perhaps others for a specific duration
  - Generate a gcore or pstack
  - Restart the daemon(s)
- These would be subject to site admin rules

Acknowledgements

- Current Software Contributors
  - ATLAS: Doug Benjamin, Patrick McGuigan, Ilija Vukotic
  - CERN: Lukasz Janyst, Andreas Peters, Sebastien Ponce, Elvin Sindrilaru
  - Fermi: Tony Johnson
  - Root: Gerri Ganis
  - SLAC: Andrew Hanushevsky, Wilko Kroeger, Daniel Wang, Wei Yang
  - UCSD: Alja Mrak-Tadel , Matevz Tadel
  - UNL: Brian Bockelman
  - WLCG: Mattias Ellert, Fabrizio Furano, David Smith
- US Department of Energy
  - Contract DE-AC02-76SF00515 with Stanford University

Now For The Demo!