

PlanetLab Experimenter Tools

Jeannie Albrecht

GEC 5

(With help/slides from John Hartman)

Cluster B Participants

- PlanetLab
- Enterprise Geni
- Mid-Atlantic Crossroads (MAX)
- GpENI
- Internet Scale Overlay Hosting (SPP)
- Gush
- Raven

Experimenter Tools

- Official "GENI" tools
 - SFI – slice facility interface (UI for the PLC)
 - Gush – short-term, "online" experiments
 - Raven/Stork – long-term, "offline" experiments
- Non-GENI PlanetLab tools
 - CoDeen (CoMon, CoDeploy, CoTop, etc)
 - PLMan
 - ...

Gush

- Experiment/Application Manager
 - Supports experiment planning, deployment, and execution
- Usage overview
 - Define application in simple XML file
 - Define resource aggregates
 - Run application
 - Install software, start execution, monitor execution, recover from failures, gather data, cleanup resources
 - Gush can use Stork to install the necessary software bundled as either an RPM or a tarball

Gush App Spec

```
<gush>
  <project name="simple">
    <software name="SimpleSoftwareName" type="none">
      <package name="Package" type="web">
        <path>http://sysnet.cs.williams.edu/~jeannie/software.tar</path>
        <dest_path>software.tar</dest_path>
      </package>
    </software>
    <component name="Cluster I">
      <rspec>
        <num_hosts>3</num_hosts>
      </rspec>
      <software name="SimpleSoftwareName" />
      <resources>
        <resource type="planetlab" group="williams_gush" />
        <resource type="gpeni" group="gpeni_gush" />
        <resource type="max" group="maxpl_gush" />
      </resources>
    </component>
    <experiment name="simple">
      <execution>
        <component_block name="cb I">
          <component name="Cluster I" />
          <process_block name="p2">
            <process name="cat">
              <path>cat</path>
              <cmdline>
                <arg>software.txt</arg>
              </cmdline>
            </process>
          </process_block>
        </component_block>
      </execution>
    </experiment>
  </project>
</gush>
```

SOFTWARE

DEFINE RESOURCE POOL

DEFINE PROCESSES (EXECUTION)

Gush Resource Directory

```
<gush>
  <resource_manager type="planetlab">
    <user>plc.williams.jeannie</user>
    <port_map slice="plc.williams.gush" port="15413"/>
    <configFile>planetlab_sfi_config</configFile>
  </resource_manager>

  <resource_manager type="gpeni">
    <user>plc.ksu.jeannie</user>
    <port_map slice="plc.ksu.gpeni_gush" port="15414"/>
    <configFile>gpeni_sfi_config</configFile>
  </resource_manager>

  <resource_manager type="max">
    <user>plc.max.maxpl.jeannie</user>
    <port_map slice="plc.max.maxpl.maxpl_gush" port="15415"/>
    <configFile>max_sfi_config</configFile>
  </resource_manager>
</gush>
```

Gush User Interfaces

- Command-line interface used to interact with applications
- Nebula (GUI) allows users to describe, run, monitor, & visualize applications
- XML-RPC interface for managing applications programmatically



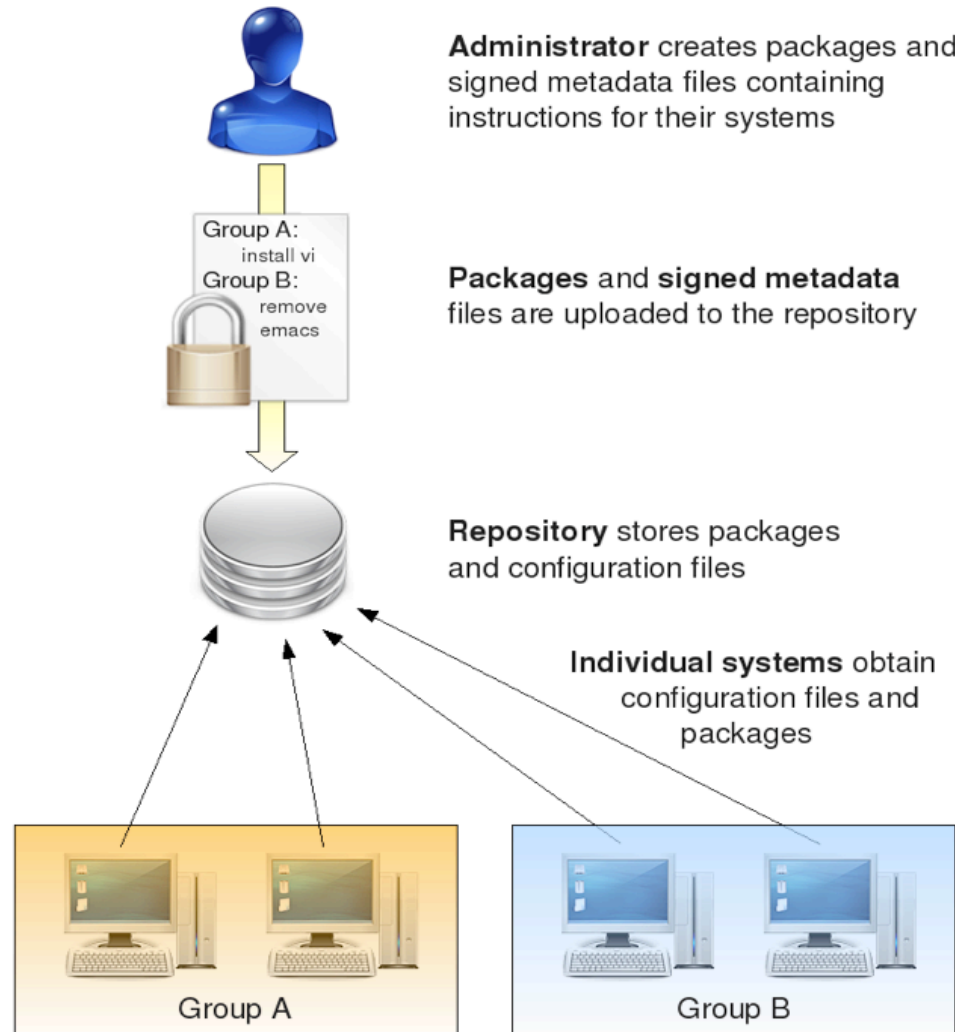
```
Nebula v0.8 - Untitled.xml
File Edit Plush
World View Application View Resource View Host View SSH:planetlab1.cs.duke.edu x
logfile-planetlab1-15415-1178479282.txt logfile-planetlab1-15415-1178664027.txt logfile-planetlab1-15417-1178514401.txt
logfile-planetlab1-15415-1178484137.txt logfile-planetlab1-15415-1178664362.txt
logfile-planetlab1-15415-1178484906.txt logfile-planetlab1-15415-1178664430.txt
[ucsd_plush@planetlab1 ~]$ less logfile-planetlab1-1541
[ucsd_plush@planetlab1 ~]$ ls -ltr
total 6732
-rwxr--r-- 1 ucsd_plush slices 241 Apr 24 17:54 plush.prefs
drwxr--r-- 3 ucsd_plush slices 4096 May 6 03:17 helper-scripts
-rwxr--r-- 1 ucsd_plush slices 6458700 May 6 19:09 client
-rw-r--r-- 1 ucsd_plush slices 293 May 6 19:21 plush-logfile15415-1178479282.txt
-rw-r--r-- 1 ucsd_plush slices 27361 May 6 19:28 logfile-planetlab1-15415-1178479282.txt
-rwxr--r-- 1 ucsd_plush slices 4764 May 6 20:41 bootstrap.pl
-rw-r--r-- 1 ucsd_plush slices 291 May 6 20:42 plush-logfile15415-1178484137.txt
-rw-r--r-- 1 ucsd_plush slices 39787 May 6 20:43 logfile-planetlab1-15415-1178484137.txt
-rw-r--r-- 1 ucsd_plush slices 293 May 6 20:55 plush-logfile15415-1178484906.txt
-rw-r--r-- 1 ucsd_plush slices 37634 May 6 20:57 logfile-planetlab1-15415-1178484906.txt
-rw-r--r-- 1 ucsd_plush slices 280 May 7 05:06 plush-logfile15417-1178514401.txt
-rw-r--r-- 1 ucsd_plush slices 18694 May 7 05:08 logfile-planetlab1-15417-1178514401.txt
-rw-r--r-- 1 ucsd_plush slices 311 May 8 22:40 plush-logfile15415-1178664027.txt
-rw-r--r-- 1 ucsd_plush slices 32749 May 8 22:44 logfile-planetlab1-15415-1178664027.txt
-rw-r--r-- 1 ucsd_plush slices 313 May 8 22:46 plush-logfile15415-1178664362.txt
-rw-r--r-- 1 ucsd_plush slices 32923 May 8 22:46 logfile-planetlab1-15415-1178664362.txt
lrwxrwxrwx 1 ucsd_plush slices 35 May 8 22:47 plush-logfile.txt -> ./plush-logfile15415-1178664430.txt
lrwxrwxrwx 1 ucsd_plush slices 41 May 8 22:47 client.txt -> ./logfile-planetlab1-15415-1178664430.txt
-rw-r--r-- 1 ucsd_plush slices 313 May 8 22:47 plush-logfile15415-1178664430.txt
-rw-r--r-- 1 ucsd_plush slices 168123 May 8 22:48 logfile-planetlab1-15415-1178664430.txt
[ucsd_plush@planetlab1 ~]$ traceroute www.google.com
traceroute: Warning: www.google.com has multiple addresses; using 72.14.205.99
traceroute to www.l.google.com (72.14.205.99), 30 hops max, 38 byte packets
 1 152.3.138.61 (152.3.138.61) 0.330 ms 0.275 ms 0.229 ms
 2 152.3.219.69 (152.3.219.69) 0.353 ms 0.300 ms 0.230 ms
 3 telisp-roti.netcom.duke.edu (152.3.219.54) 0.281 ms 0.333 ms 0.245 ms
 4 te2-1--581.tr01-asbnva01.transitrail.net (137.164.131.173) 7.633 ms 7.663 ms 8.402 ms
 5 tel-2.tr01-sttlwa01.transitrail.net (137.164.129.37) 76.141 ms 84.463 ms 76.121 ms
 6 te4-1--160.tr01-plalca01.transitrail.net (137.164.129.34) 93.630 ms 93.511 ms 93.597 ms
 7 calren-trcust.plalca01.transitrail.net (137.164.131.254) 99.644 ms 97.167 ms 93.723 ms
 8 * * *
 9 209.85.130.4 (209.85.130.4) 95.293 ms 97.987 ms 94.702 ms
10 64.233.174.81 (64.233.174.81) 86.525 ms 86.340 ms 86.495 ms
   MPLS Label=684000 CoS=0 TTL=1 S=1
11 72.14.236.20 (72.14.236.20) 93.077 ms 110.785 ms 93.037 ms
12 72.14.232.113 (72.14.232.113) 100.908 ms 96.452 ms 98.807 ms
13 72.14.232.62 (72.14.232.62) 99.173 ms 72.14.236.142 (72.14.236.142) 95.319 ms 72.14.232.66 (72.14.232.66) 100.434 ms
14 qb-in-f99.google.com (72.14.205.99) 95.983 ms 93.976 ms 107.922 ms
[ucsd_plush@planetlab1 ~]$
```

Raven

- Raven experiment
 - Uses software bundled as either an RPM or a tarball and exp spec (defined via Tempest config files)
 - Software is installed on appropriate nodes automatically
- Tempest
 - Allows the user to specify groups of nodes and install packages based on those groups
 - Can use dynamic CoMon queries to form the groups, e.g. "all nodes with load < X"

Raven

Collective Package Actions



Individual systems **determine which instructions apply to them** and install or remove packages accordingly. Offline systems as well as new systems perform all actions when brought online

Specifying Package Actions

```
<PACKAGES>  
  <CONFIG GROUP="A">  
    <INSTALL PACKAGE="vi" VERSION="2.2"/>  
  </CONFIG>  
  <CONFIG GROUP="B">  
    <REMOVE PACKAGE="emacs"/>  
  </CONFIG>  
  <CONFIG>  
    <UPDATE PACKAGE = "firefox"/>  
  </CONFIG>  
</PACKAGES>
```

Specifying Groups

```
<GROUPS>  
  <GROUP NAME="A">  
    <INCLUDE NAME="planetlab1.arizona.net"/>  
    <INCLUDE NAME="planetlab2.arizona.net"/>  
  </GROUP>  
  <GROUP NAME="B">  
    <INCLUDE SLICE="arizona_stork"/>  
  </GROUP>  
  <UNION NAME="Both" GROUP1="A" GROUP2="B"/>  
</GROUPS>
```

Questions

- How are experiments specified in Cluster B?
What are the declarative and procedural aspects of these specifications?
 - Gush experiment/application specification (procedural + declarative)
 - Gush resource directory (declarative)
 - Raven packages and groups (declarative)

Questions

- How is this specification used by the tool chain available to experimenters?
 - Gush app spec defines execution and resources
 - Raven package spec defines mapping of software packages to groups
 - Raven group spec defines groups of resources

Questions

- What are the experimenter tools in your cluster that might be ported to other control frameworks?
 - Gush and Raven can both be ported to other control frameworks
 - Not sure about Non-GENI tools

Questions

- What assumptions do your tools make about the control framework? What assumptions might be specific to your control framework?
 - Gush and Raven both use geniwrapper GID/certificate infrastructure to authenticate
 - Previously Gush only assumed SSH access to resources
 - SSH to machines, start client process, make TCP connection, exec processes