What is Emulab?

- Software to control network testbeds
  - Instantiates user-requested topologies on available resources
  - Most popular UI is fancy Web interface; XML-RPC
- Emulab “Classic”
  - ~200 PCs in a densely connected cluster
  - Dozens of experiments “swap” in and out each day
- Extended to wide area in late 2002
  - RON testbed and Emulab’s own wide-area nodes
- Now: a testbed with diverse resources
  - Physical, virtual and simulated nodes and links
Why Create "The Portal"?

- Diversify Emulab with new resources
- Explore challenges of integrating with other testbed environments
- Provide PlanetLab users with a powerful but easy-to-use interface
K.I.S.S.

Create a Slice on PlanetLab

Number of nodes 222 or All available nodes or One node at each site

Create it More options

[ Documentation: ] Search [ News ]
### Decisions, Decisions

#### Create a Slice on PlanetLab - Advanced Form

<table>
<thead>
<tr>
<th>Section</th>
<th>Options</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Basic Options</strong></td>
<td>Number of nodes</td>
<td>222 or All available nodes or One node at each site</td>
</tr>
<tr>
<td><strong>Advanced Options</strong></td>
<td>Type of PlanetLab nodes:</td>
<td>Any PlanetLab node (222 available at 105 sites)</td>
</tr>
<tr>
<td></td>
<td>Estimated CPU and memory use:</td>
<td>Very Low</td>
</tr>
<tr>
<td></td>
<td>Retry until nodes with sufficient resources are available:</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>Proceed even if some nodes fail to set up:</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>Auto-terminate slice after:</td>
<td>52 Weeks</td>
</tr>
<tr>
<td><strong>Files to Install and Maintain</strong></td>
<td>Tarball(s) to install:</td>
<td>/tmp <a href="http://www.cs.utah.edu/~lepreau/fooby.tar.gz">http://www.cs.utah.edu/~lepreau/fooby.tar.gz</a></td>
</tr>
<tr>
<td></td>
<td>RPM(s) to install:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Command to run on startup:</td>
<td>/tmp/fooby/lsit</td>
</tr>
</tbody>
</table>

You can also take a look at the widearea node link metrics
Emulab provides all elements of the PlanetLab infrastructure service taxonomy, plus more.
Portal Features (cont'd)

• Monitors sensors to ascertain node characteristics
• Three selection methods: manual, link-, and node-centric
>

**Portal Features (cont'd)**

- **Watchdog process per virtual node**
- **Software upgrades and account updates**
Portal Features (cont’d)

- “Reboot” a single virtual node, or all of them
- Soon: wide-area event system for control
Challenges and Lessons

- Different use models
- State management
- Interface evolution
- Failure
Different Use Models

- Emulab: rapid cycle experiments (mostly)
  PlanetLab: long-running services (mostly)
- Average Emulab experiment duration: five hours
- Building fast/synchronous on delayed/async?

- Delayed, asynchronous interfaces force fast synchronous clients to waste resources
- Exposing lower-level API primitives allows a wider range of service models
State Management

- Locations of data are spread out
- Data coupling issues
  - Identity crisis!
  - Balance between coherency and overhead (age-old problem)

- Persistent & reliable node identifiers are a must
- Should not assume long-term state synchronization
Interface Evolution

- Research infrastructures evolve rapidly
- Tension between PlanetLab goals:
  "Evolving Architecture" $\rightarrow$ change
  vs.
  "Unbundled Management" $\rightarrow$ many services,
  many players

- Internally, use the same API that you export
- Embrace the inevitable: changing APIs.
  Make that code modular
Failure

- All node “liveness” metrics are unreliable
  - Trumpet, Ganglia, Emulab Watchdog …
- Anything can fail
  - Disk space, fds, PLC, …

- Only execution of the application itself indicates node “liveness”!
Conclusions

- Hard to keep it working
- Will people build large systems on other parties’ constantly-changing research systems?