

Seattle: Tools and Services

Tom Anderson and **Justin Cappos**
Computer Science and Engineering
University of Washington



What is Seattle?

PlanetLab@Home



What is Seattle?

- P2P hosting platform (it's a potluck!)
 - You contribute some resources
 - You consume what others provide
 - Others consume what you provide
- Contributed resources
 - Remotely accessible
 - Requested through a sharing site
 - Run in a virtual machine (vessel)



Seash (Seattle Shell)

- Modeled loosely on PLUSH / GUSH
- Acts on groups of vessels
- Some example commands
 - Start / stop program
 - Upload / download files
 - List the status of programs
 - Retrieve vessel logs
 - Find node location information
 - Split / join vessels
- Limitations
 - Cannot acquire resources from sharing site
 - Isn't scriptable
 - No support for notifications
- **Classroom tested!**



Seattle Experiment Library

- Programmable interface for Seattle
- Interfaces with both vessels and SeattleGENI
- Acts on groups of vessels
- Example functionality
 - Acquire / release vessels
 - Renew vessels
 - Similar functionality to Seash
- Limitations
 - Not tested by novice users



AppMap Demo

- Built on Seattle Experiment Library
- Basic idea
 - Acquires vessels
 - Determines coordinates
 - Runs allpairsping program
 - Gathers ping latency (httpretrieve library)
 - Releases vessels (implicit stop)
 - Displays webpage (httpserver library)



Service Monitor

- Built on Seattle Experiment Library
- Allow a long running service to persist
- Basic idea
 - Detect failed vessels
 - Relinquish and redeploy to good vessels
 - Notify if 'oddities' are observed



Long Running Services

- Time Service
 - NTP information for firewalled vessels
- NAT Forwarding Service
 - Relays connection information for NATed nodes
- Lookup Service
 - DHT-style storage
 - Digital Object Registry
- Controlled Node Communication (Beta)
 - Provides membership information



Secure Software Updates

Building a secure software update system is very hard

TUF: a library for securing software update systems

Looking for GENI projects to participate

Ideally would have either:

1. Preexisting software update system implemented in Python
2. No software update system

<http://www.updateframework.com>

<http://groups.geni.net/geni/wiki/SecureUpdates>



Thanks

Giridhar Manepalli and Larry Lannom at CNRI

Gary Wong and Rob Ricci at Emulab

Justin Samuel, Evan Meagher, Monzur Mohammed,
Yafete Yemeru, and Steven Sievers

All of the instructors and students who have used the
platform

Ivan Beschastnikh, Arvind Krishnamurthy, undergrads



Summary

Seattle Tools and Services

- Seash shell
 - Used by many students
- Seattle Experiment Library
- Long Running Services
 - NTP, NAT, Lookup, CNC

<https://seattle.cs.washington.edu/>

- Secure Software Updates

<http://www.updateframework.com/>



Demonstration

- Registration
- Download Installer
- Acquire resources
 - Use seattlegen website
- Deploy all pairs ping
 - Use shell to locate and control resources

