Experiment Reproducibility in Planetlab

RP 1.1 Project Presentation
Sudesh Jethoe
Experiment reproducibility in a distributed environment

- Processing power
- Load of individual servers
- Available bandwidth
- Latency between servers

Under influence of
- Network topology changes
- Occupation of resources
- Bandwidth management
Planetlab?

Distributed Network Research Environment
- Application Development for the Internet
- CDN's (Content Distribution Networks)
- P2P (Peer to peer systems)
- Resource Distribution / Load balancing
- Currently 1076 nodes at 544 sites*

* https://www.planet-lab.org/
Node:
   A machine in the planetlab network
Slice:
   A collection of virtual machines spread over several nodes
Sliver:
   A single virtual machine residing on a single node in the network

Slices can be configured to contain VM's on any node through an API or web interface.
Planetlab 101

Research Questions

"What changes does the PlanetLab infrastructure experience and how will these changes influence experiments conducted in the Planetlab distributed environment?"

“Which factors can be identified that influence the latency between nodes and to what extent?”

“Which factors can be identified that influence the bandwidth between nodes and to what extent?”
Planetlab monitoring system

CoMon, passive monitoring, published via webinterface

- **Node centric daemon**
  - current load of node
  - # active slices on node
  - TX/RX rates
  - disk I/O

- **Slice centric daemon**
  - Relative resource usage per slice
Initial approach

- Dynamically acquire sets of nodes in different regions.
- Retrieve passive monitoring data at fixed intervals.
- Transfer data (100 MB) to each node.
- Measure latency before, during and after the data transfer.
- Collect, store and analyze data.
Experimental setup

Set up multiple experiments in different regions.

Have a central node run a python script which:
● Collect monitoring data from remote servers
● Run iperf client against remote servers
● Run ping against remote servers
● Send data to local machine for analysis.

4 Remote nodes, of which:
1 Remote node in the vicinity of central node
3 Nodes in the same "region" (country).
Implementation Issues

● Planetlab Central API:
  ○ Acquire nodes
  ○ Release nodes
  ○ Retrieve new nodes
● Returns faulty nodes
● Manual selection

● Python script:
  ○ Run experiment and measurements repeatedly
  ○ In multiple regions
  ○ Analyze data
● No concurrent execution of threads due to GIL
● Meta_script which launches separate processes
● Manual analysis of data
Final Implementation

- Gather data
- Run experiment
- Stop gathering data
- Sync data
- Repeat

```bash
while time < ##:
    planetlab.py monitor &
    planetlab.py ping &
    sleep 60 seconds
    planetlab.py iperf (main)
    sleep 60 seconds
    touch $stop_monitoring
    rsync data my_server:~/data
    done
```
Results

RTT distribution BRlist 27 June 19.00

RTT

Time in execution rounds
Results

CoMon RX_rate data of receiving nodes in Brazil

RX rate averaged by minute

Time in execution rounds (1 round per minute)
Results

![Graph showing RX rate over time for plab2.larc.usp.br per round. The graph includes data points for different execution rounds, with markers for 0rx_plab2.larc.usp.br, 1rx_plab2.larc.usp.br, and 2rx_plab2.larc.usp.br.]
Results

Graph showing the rate of execution per round for different conditions. The x-axis represents time in execution rounds, while the y-axis shows the Rx rate. Different conditions are indicated by different colors and markers.
Results
Iperf tests don't show up due to limited resolution of CoMon

Bandwidth limits on nodes limit longer tests for now

Longer measurements are needed to gather enough CoMon data.
Questions

RP 1.1 Project Presentation
Sudesh Jethoe