PlanetLab

- Overview, History, and Future Directions
- Using PlanetLab for Network Research: Myths, Realities, and Best Practices

2009.12.16
Sung Bum Park
Contents

- Introduction
- Concept of Planetlab
- Architecture
- Myths, realities, and best practices
- Conclusion
- Q & A
Concept of Planetlab

Concept:
- Planetary scale overlay network
- Testbed for developing/accessing network services
- Real world experience
- 1068 nodes over 491 sites
Node Architecture Goals

- Provide a virtual machine for each service running on a node
- Isolate virtual machines
- Allow maximal control over virtual machines
- Fair allocation of resources
  - Network, CPU, memory, disk
Node Architecture

- Virtual machine (VM)
  - Available to run several tests on a node

- Vserver
  - Patched Linux 2.4 kernel
  - Illusion of multiple servers on a single machine
  - Have its own superuser with safe access
Node Architecture

- Slices (VMs)
  - Node manager
    - resource allocation
    - sensors
    - auditing
    - slice bootstrapping
  - Local admin
    - resource limits
    - kill process

VMM: Linux++
Network Architecture

- **Node manager (one per node)**
  - Create slices for service managers
    - When service managers provide valid tickets
  - Allocate resources for vservers

- **Resource Monitor (one per node)**
  - Track node’s available resources
  - Tell agents about available resources
Network Architecture

Agents (centralized)
- Track nodes’ free resources
- Advertise resources to resource brokers
- Issue tickets to resource brokers
  - Tickets may be redeemed with node managers to obtain the resource
Network Architecture

- **Resource Broker (per service)**
  - Obtain tickets from agents on behalf of service managers

- **Service Managers (per service)**
  - Obtain tickets from broker
  - Redeem tickets with node managers to acquire resources
  - If resources can be acquired, start service
Services Run in Slices

PlanetLab Nodes
Services Run in Slices

PlanetLab Nodes

Virtual Machines

Service / Slice A
Services Run in Slices

Service / Slice A
Service / Slice B

PlanetLab Nodes
Virtual Machines
Services Run in Slices
Obtaining a Slice

Agent

Broker

Service Manager
Obtaining a Slice

Agent

Resource Monitor

Broker

Service Manager
Obtaining a Slice

Agent

Resource Monitor

Broker

Service Manager
Obtaining a Slice

Resource Monitor

Agent

- ticket -

Broker

Service Manager
Obtaining a Slice

Agent

Broker

Service Manager

Resource Monitor

Resource Monitor

Resource Monitor
Obtaining a Slice

Agent

Broker

Service Manager

Resource Monitor

Resource Monitor
Obtaining a Slice

Agent
- ticket
- ticket
- ticket

Broker

Service Manager
Obtaining a Slice
Obtaining a Slice

Agent

- ticket
- ticket
- ticket

Broker

Service Manager

22
Obtaining a Slice

Agent

Broker

Service Manager

ticket
ticket
Obtaining a Slice

Agent

Broker

Service Manager

Node Manager

Node Manager

Manager

Ticket

Ticket
Obtaining a Slice
Obtaining a Slice

Agent

Broker

Service Manager
Realities

- Describes widely-cited criticisms for PlanetLab that are entirely true
Reality 1: Results are not reproducible

- Load on networks and on machines varies on every time scale
- An experiment that runs for an hour will reflect only that network condition
- Using the CoMon
- Alternatives: Emulab, Modelnet

- Producing unexpected result in short period is not a bug
Reality 2: PlanetLab nodes are not representative of peer-to-peer network nodes

- PlanetLab is a managed infrastructure and not subject to the same churn as desktop systems
- Cannot scale to millions of machines
Myths that are no longer true

Some who tried to use early versions of PlanetLab found challenges that are no longer problems
Myth 1: PlanetLab is too heavily loaded

- May always be under-provisioned
- Load is especially high before conference deadlines

- Newly determined daemon
- Two brokerage services
  - Sirius and Bellagio
Myth 1: PlanetLab is too heavily loaded

Figure 1: Median, 25th, and 10th percentiles of available CPU across PlanetLab nodes, measured using spin loops.
Myth 2: PlanetLab cannot guarantee resources

- With the release of PlanetLab ver. 3.0
  - Resource guarantees are possible

- By using *Sirius* and *Bellagio*
  - Running slices could receive resource guarantees
Myths falsified by best practices

The following three myths about PlanetLab are not true if best practices are followed:

1. The first two myths address problems using PlanetLab for network measurement, the last, its potential for churn.
Myth 1: Load prevents accurate latency measurement

- Cannot ensure that any slice will be scheduled immediately upon receiving a packet

- Using in-kernel timestamping features of Linux
Myth 2: Load prevents sending precise packet trains

- Sending packets at precise times is more difficult.
- The desired sending times were not achieved, then data sending on PlanetLab simply requires more attempts than on unloaded systems.
Myth2: Load prevents sending precise packet trains

- 11 packets
- The gap is either 1ms or 11ms

1ms or 11ms
Myth 2: Load prevents sending precise packet trains.
Myth 3: PlanetLab experiences excessive churn

- Only three times during the last two years (2004~)
  - Dec. 2003: all nodes were off-line for a week
    - A security incident
  - Nov. 2004: upgraded Ver. 1.0 -> Ver. 2.0
  - Feb. 2005: many nodes off-line for a weekend
    - By a kernel bug
Conclusion

PlanetLab is a global research network that supports the development of new network services.

- Help to develop
  - new technologies for distributed storage
  - network mapping
  - peer-to-peer systems
  - distributed hash tables
  - and query processing.
Q & A