Experiences Building PlanetLab

Larry Peterson, Andy Bavier, Marc E. Fiuczynski, and Steve Muir
Princeton University

UWCS OS Seminar Discussion
Andy Pavlo
06 November 2006

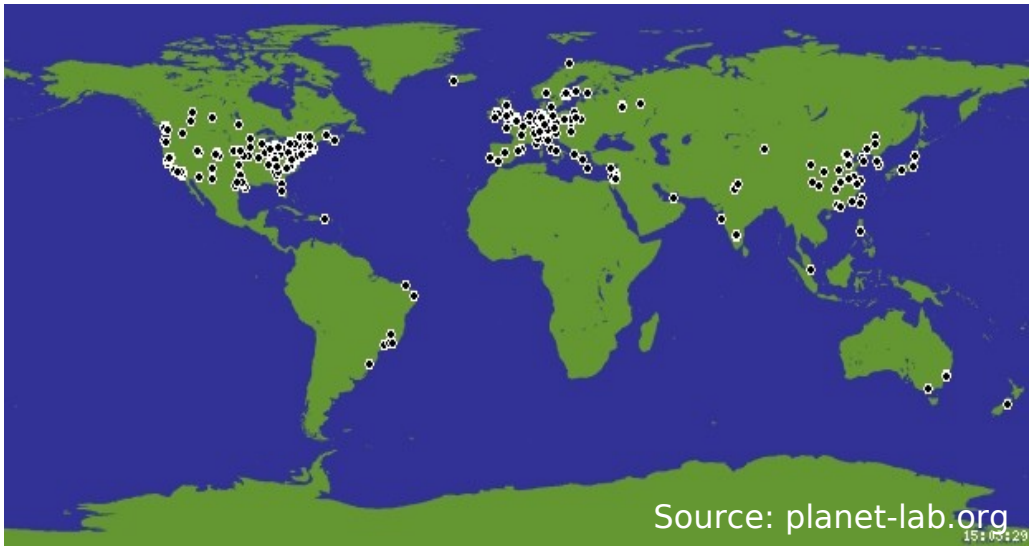
Outline

- Overview
- Implementation
- Resource Allocation
- Discussion

How to enable the development of new, potentially disruptive network technologies?

Overview

- Global network services testbed and deployment platform.
- Launched in 2002.
- Current status:
  - 694 nodes
  - 335 sites
  - 35 countries
- UWCS contributes two nodes.



Overview

- Testbed:
  - Geographically distributed machines.
  - Real network behavior.
  - Realistic client workload.
- Deployment platform:
  - Easy to develop new services.
  - Provide users with access to these services.

Organizing Principles

- Distributed virtualization.
- Unbundled management.
- Preserve chain of responsibility.
- Decentralized control.
- Efficient resource sharing.

Implementation

- Component overview:
  - Nodes
  - Slices
  - Management services
- Design principles:
  - Trust assumptions
  - Delegation
  - Federation

Nodes

- Linux+VServer x86 nodes.
- Booted from immutable file system:
  - Boot manager.
  - Public key of the central manager.
  - Node-specific secret key.
- Nodes authenticate with the PlanetLab central manager before deploying the VMM.
- Sites that contribute more than the minimum can control access.

Slices

- Users request *slices* to run experiments:
  - A set of VMs, each running on a unique node.
  - Each individual VM contains no knowledge about other VMs.
  - Users given remote access to VMs.
- Short-term vs. long-running
- Direct vs. delegated

Management

- Management services deployed in slices with additional privileges:
  - Slice creation
  - Resource brokerage
  - Monitoring
  - Environment instantiation
  - Auditing

Trust Assumption

- PlanetLab Central acts as trusted intermediary that manages nodes.
- Trust relationship:
  - PLC expresses trust to user.
  - User trusts PLC to create slices on their behalf.
  - Node owner trusts PLC to manage slices.
  - PLC trusts owners to keep nodes physically secure.
- Implications?

Federation

- Need a well-defined API between central managers.
- Unique 'root' names for nodes & slices.
- Nodes can simultaneously belong to more than one PlanetLab pools.

Resource Allocation

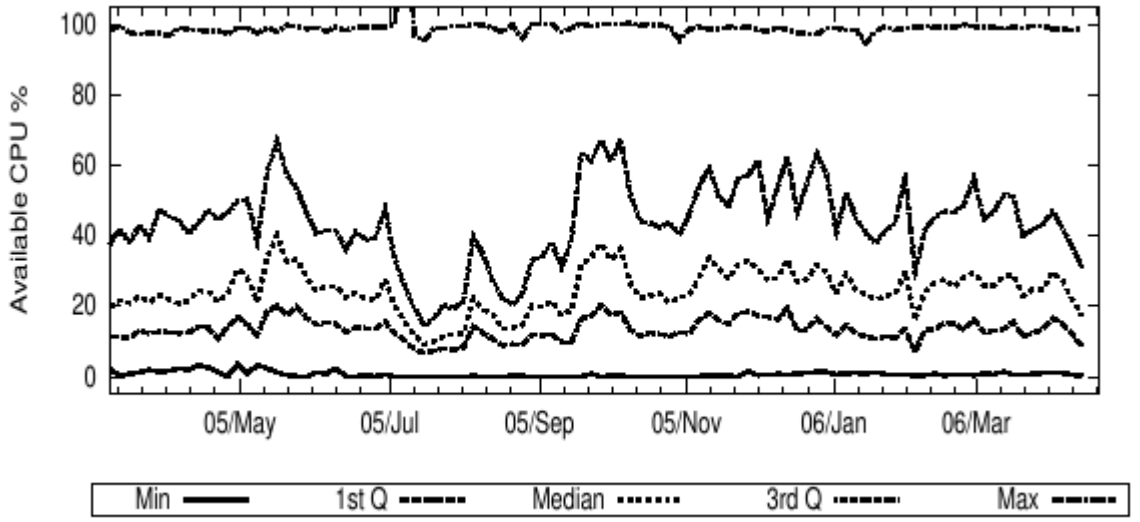
- Over-allocation of resources
- Components:
  - CPU Scheduling
  - Memory
  - Network Bandwidth
  - Disk Usage

Workload

- No imposed limits to number of VMs.
- Gracefully degrade node performance.
- Monitoring has shown that nodes on average support up to 90 *active* VMs and 25 *live* VMs.
  - Active: Contains a process.
  - Live: Contains process that used at least 0.1% (300ms) of CPU in last 5 minutes.

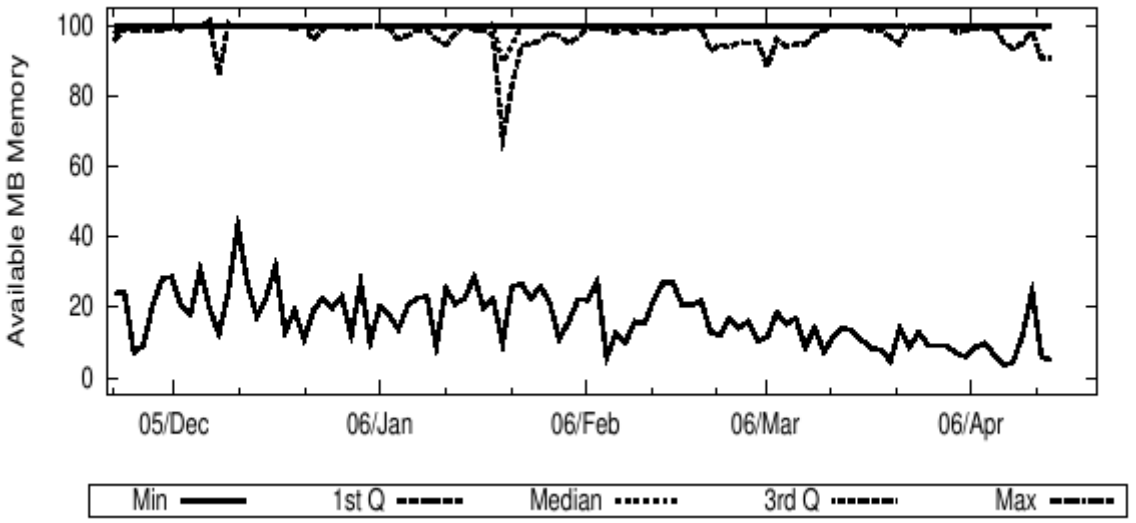
Resource: CPU

- Deployed four different schedulers.
- Current version:
  - Token bucket filter on top of standard Linux CPU scheduler.
  - Reservations vs. Shares
  - Exploring techniques to overcome latency.



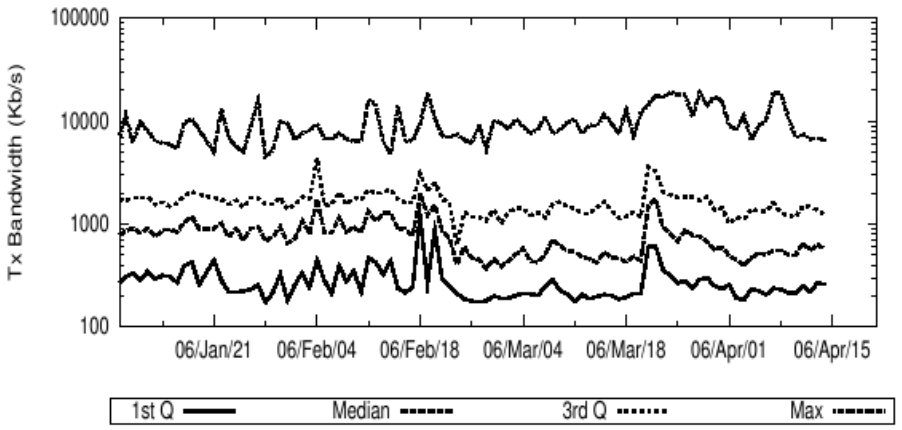
Resource: Memory

- Four possible design choices.
- Current version:
  - Dynamically allocate memory to slices on demand
  - Reset slices using the most physical memory when swap is filled.

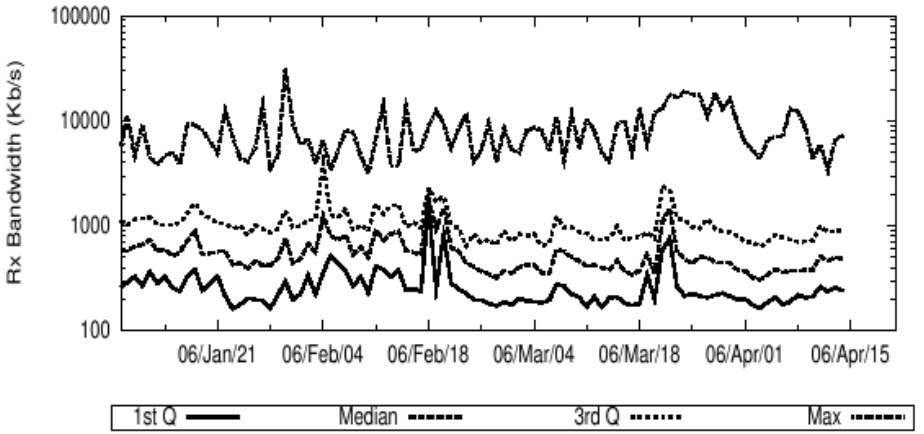


Resource: Bandwidth

- Fair share using token bucket.
- Current version:
  - Slices are allocated a fixed amount of outbound traffic per day.
  - Once this limit is surpassed, the rate is capped until the end of the day.
  - No control of inbound traffic at this time.



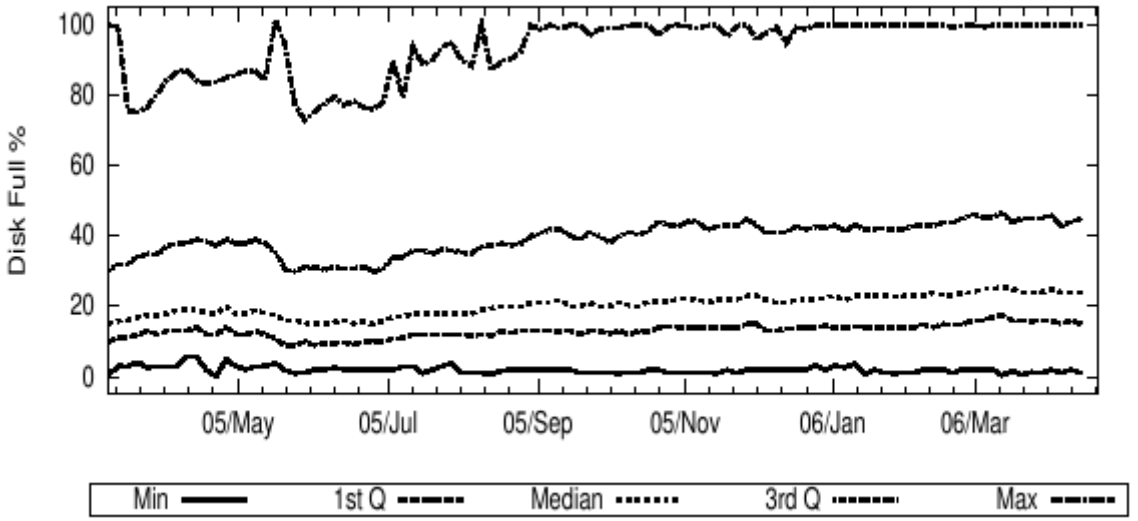
outbound



inbound

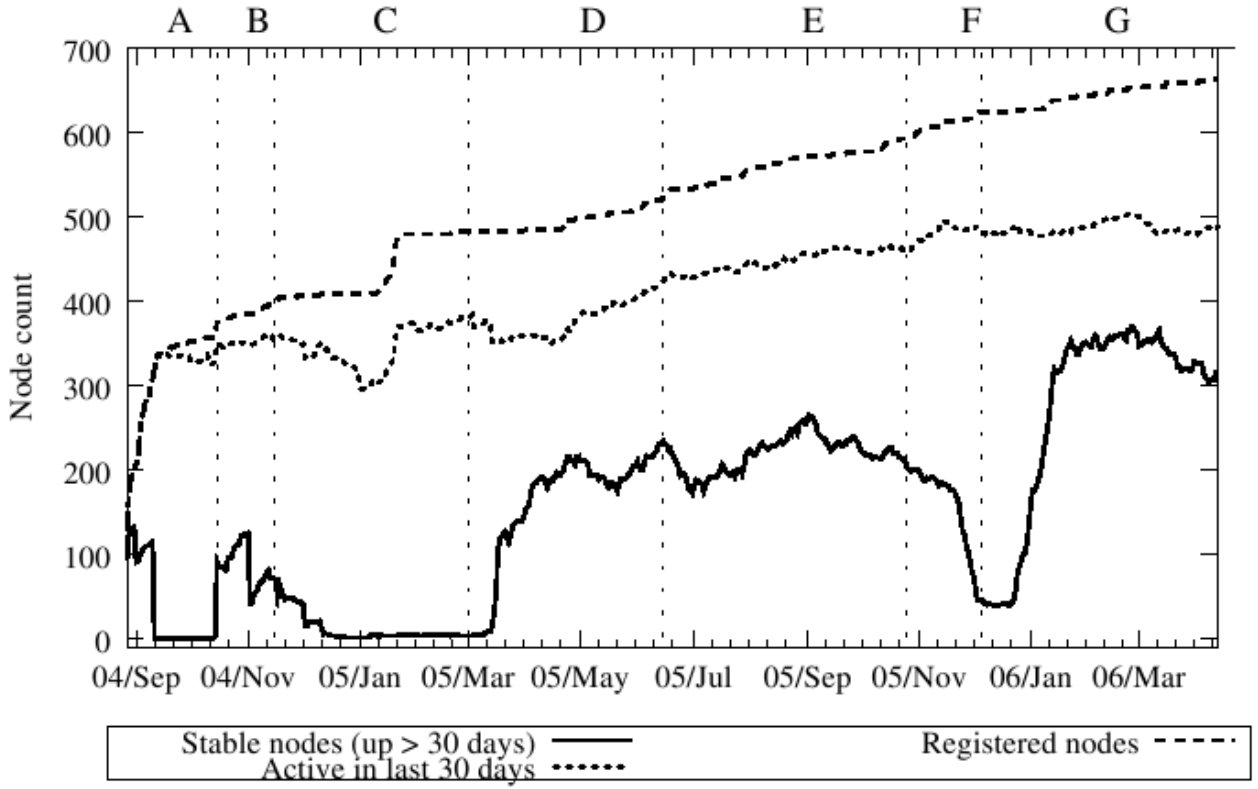
Resource: Disk

- Nodes do not provide any permanent storage.
- Each slice is given a 5GB quota.



System Stability

- Overall strategies:
  - API changes only for management services.
  - Leverage existing software components.
  - Incremental updates.



Unique Problem Space

- Shares characteristics and problems with three similar system domains:
  - ISPs
  - Hosting centers
  - The Grid

PlanetLab vs. Condor

- Testbed vs. Batch System
- Unreliable environment
- Virtual machine support
- Resource allocation
- Localized, monolithic pools
- Supports more OSes/Archs

Discussion

- PlanetLab a good platform?
- Pros:
  - Easy to deploy new services.
- Cons:
  - Difficult to perform accurate time measurement.
  - Lack of permanent storage.
- Comparison to other testbeds:
  - Internet2, Emulab, ABONE, XBONE