Towards a Distributed Test-Lab for Planetary-Scale Services

David Culler
UC Berkeley
Intel Research @ Berkeley
Motivation

• A new class of services & applications is emerging that spread over a sizable fraction of the web
  – CDNs as the first examples
  – Peer-to-peer, ...

• Architectural components are beginning to emerge
  – Distributable hash tables to provide scalable translation
  – Distributed storage, caching, instrumentation, mapping, ...

• The next internet will be created as an overlay on the current one
  – as did the last one
  – it will be defined by its services, not its transport
    » translation, storage, caching, event notification, management

• There is NO vehicle to try out the next $n$ great ideas in this area
Guidelines (1)

- Thousand viewpoints on “the cloud” is what matters
  - not the thousand servers
  - not the routers, per se
  - not the pipes
Guidelines (2)

- and you must have the vantage points of the crossroads
  - primarily co-location centers
Guidelines (3)

• Each service needs an overlay covering many points
  – logically isolated
• Many concurrent services and applications
  – must be able to slice nodes => VM per service
  – service has a slice across large subset
• Must be able to run each service / app over long period to build meaningful workload
  – traffic capture/generator must be part of facility
• Consensus on “a node” more important than “which node”
Guidelines (4)

- Test-lab as a whole must be up a lot
  - global remote administration and management
    » mission control
  - redundancy within
- Each service will require its own remote management capability
- Testlab nodes cannot “bring down” their site
  - generally not on main forwarding path
  - proxy path
  - must be able to extend overlay out to user nodes?
- Relationship to firewalls and proxies is key
Guidelines (5)

• Storage has to be a part of it
  – edge nodes have significant capacity

• Needs a basic well-managed capability
  – but growing to the seti@home model should be considered at some stage
  – may be essential for some services
# Initial Researchers (mar 02)

<table>
<thead>
<tr>
<th>Location</th>
<th>Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>Washington</td>
<td>Tom Anderson, Steven Gribble, David Wetherall</td>
</tr>
<tr>
<td>MIT</td>
<td>Frans Kaashoek, Hari Balakrishnan, Robert Morris, David Anderson</td>
</tr>
<tr>
<td>Intel Research</td>
<td>David Culler, Timothy Roscoe, Sylvia Ratnasamy, Gaetano Borriello, Satya, Milan Milenkovic</td>
</tr>
<tr>
<td>Duke</td>
<td>Amin Vadat, Jeff Chase</td>
</tr>
<tr>
<td>Princeton</td>
<td>Larry Peterson, Randy Wang, Vivek Pai</td>
</tr>
<tr>
<td>Rice</td>
<td>Peter Druschel</td>
</tr>
<tr>
<td>Utah</td>
<td>Jay Lepreau</td>
</tr>
<tr>
<td>CMU</td>
<td>Srini Seshan, Hui Zhang</td>
</tr>
<tr>
<td>UCSD</td>
<td>Stefan Savage</td>
</tr>
<tr>
<td>Columbia</td>
<td>Andrew Campbell</td>
</tr>
<tr>
<td>ICIR</td>
<td>Scott Shenker, Mark Handley, Eddie Kohler</td>
</tr>
</tbody>
</table>
Initial Planet-Lab Candidate Sites

- Intel Berkeley
- ICIR
- MIT
- Princeton
- Cornell
- Duke
- UT
- Columbia
- UCSB
- UCB
- UCLAUCLA
- UW
- Intel Seattle
- Intel OR
- UBC
- UCSD
- Intel
- Intel Berkeley
- UCB
- UCSB
- UCLA
- UCSD
- ISI
- St. Louis
- Washu
- KY
- Duke
- GIT
- Intel
- MIT
- Cornell
- Princeton
- Columbia
- Harvard
- St. Louis
- Chicago
- Utah
- Orange
- Cambridge
- Copenhagen
- Amsterdam
- Karlsruhe
- Barcelona
- Melbourne
- Beijing
- Tokyo
- Uppsala
Hard problems/challenges

• “Slice-ability” – multiple experimental services deployed over many nodes
  – Distributed Virtualization
  – Isolation & Resource Containment
  – Proportional Scheduling
  – Scalability

• Security & Integrity - remotely accessed and fully exposed
  – Authentication / Key Infrastructure proven, if only systems were bug free
  – Build secure scalable platform for distributed services
    » Narrow API vs. Tiny Machine Monitor

• Management
  – Resource Discovery, Provisioning, Overlay->IP
  – Create management services (not people) and environment for innovation in management
    » Deal with many as if one

• Building Blocks and Primitives
  – Ubiquitous overlays

• Instrumentation
Confluence of Technologies

- Cluster-based scalable distribution, remote execution, management, monitoring tools
  - UCB Millennium, OSCAR, ..., Utah Emulab, ...
- CDNS and P2Ps
  - Gnutella, Kazaa, ...
- Proxies routine
- Virtual machines & Sandboxing
  - VMWare, Janos, Denali,... web-host slices (EnSim)
- Overlay networks becoming ubiquitous
  - RON, Detour... Akamai, Digital Island, ....
- Service Composition Frameworks
  - yahoo, ninja, .net, websphere, Eliza
- Established internet ‘crossroads’ – colos
- Web Services / Utility Computing
- Grid authentication infrastructure
- Packet processing,
  - layer 7 switches, NATs, firewalls
- Internet instrumentation

The Time is NOW
Emerging “Killer Apps” and Community

• CDNs and P2Ps are first examples
  – coherent service / application spreads itself over much of the internet

• Researchers looking at key architectural elements
  – Distributed Hash Tables
    » Chord, CAN, Tapestry, Pastry
  – Distributed Storage
    » oceanstore, ...

• Vibrant research community embarking on new direction and none can try out their ideas.

NOW is the Time
Approach: Service-Centric Virtualization

• Virtual Machine Technology has re-emerged for hosting complete desktop environments on non-native OS’s and potentially on machine monitors.
  – ex. VMWare, ...

• Sandboxing has emerged to emulate multiple virtual machines per server with limited /bin, (no /dev)
  – ex. ENSim web hosting

• Network Services require fundamentally simpler virtual machines, can be made far more scalable (VMs per PM), focused on service requirements
  – ex. Jail, Denali, scalable and fast, but no full legacy OS
  – access to overlays (controlled access to raw sockets)
  – allocation & isolation
    » proportional scheduling across resource container - CPU, net, disk
  – foundation of security model
  – fast packet/flow processing puts specific design pressures

• Instrumentation and management are additional virtualized ‘slices’
  - distributed workload generation, data collection
Security: restricted API -> Simple Machine Monitor

• Authentication & Crypto works… if underlying SW has no holes
  ⇒ very simple system
  ⇒ push complexity up into place where it can be managed
    ⇒ virtualized services
• Classic ‘security sandbox’ limits the API and inspects each request
• Ultimately can only make very tiny machine monitor truly secure
• SILK effort (Princeton) captures most valuable part of ANets nodeOS in Linux kernel modules
  – controlled access to raw sockets, forwarding, proportional alloc
• Key question is how limited can be the API
  – ultimately should self-virtualize
    » deploy the next planetlab within the current one
  – progressively constrain it, introducing compatibility box
  – minimal box defines capability of thinix
• Host $\phi_1$ planetSILK within $\phi_2$ thinix VM
Planned Obsolescence of Building Block services

• Community-driven service definition and development
• Service components on node run in just another VM
  – service slices from the beginning
• Team develops bootstrap ‘global’ services - centralized
  – authentication
  – discovery, matching
  – provisioning, overlay allocation
  – higher level resource management provides guidelines and permission to negotiate with nodal resources, but sites ultimately control the actual resources
• These bootstrap services become mere backstop once successful
  – distributed versions of these services replace them
Overlapping Phases of Development

- 2003:
  - 0: seed

- 2004:
  - 1: get API & interfaces right

- 2005:
  - 2: get underlying arch. and impl. right

6/7/2002

planet-lab
Plan

• Success: adoption and growth of the research community and creation of novel network services
  – The Services will define the next internet
  – PlanetLab should take on life of it’s own
  – However, a central operations capability will be required to maintain core components

• Intel Research is already seeding the effort

• Will need to bring in NSF, Darpa, other industry

• Proposal: Create a non-profit or consortium to manage PlanetLab by late 2004
  – Consortium model maintains openness, but provides revenue model
  – Core set of engineers and operations staff
  – Node addition/replacement, bandwidth,
What Planet-Lab will enable

• Create the open infrastructure for invention of the next generation of wide-area (“planetary scale”) services
  – post-cluster, post-yahoo, post-CDN, post-P2P, ...

• Potentially, the foundation on which the next Internet can emerge
  – think beyond TCP/UDP/IP + DNS + BGP + OSPF... as to what the net provides
  – building-blocks upon which services and applications will be based
  – “the next internet will be created as an overlay in the current one” (NRC)

• A different kind of network testbed
  – not a collection of pipes and giga-pops
  – not a distributed supercomputer
  – geographically distributed network services
  – alternative network architectures and protocols

• Focus and Mobilize the Network / Systems Research Community to define the emerging internet