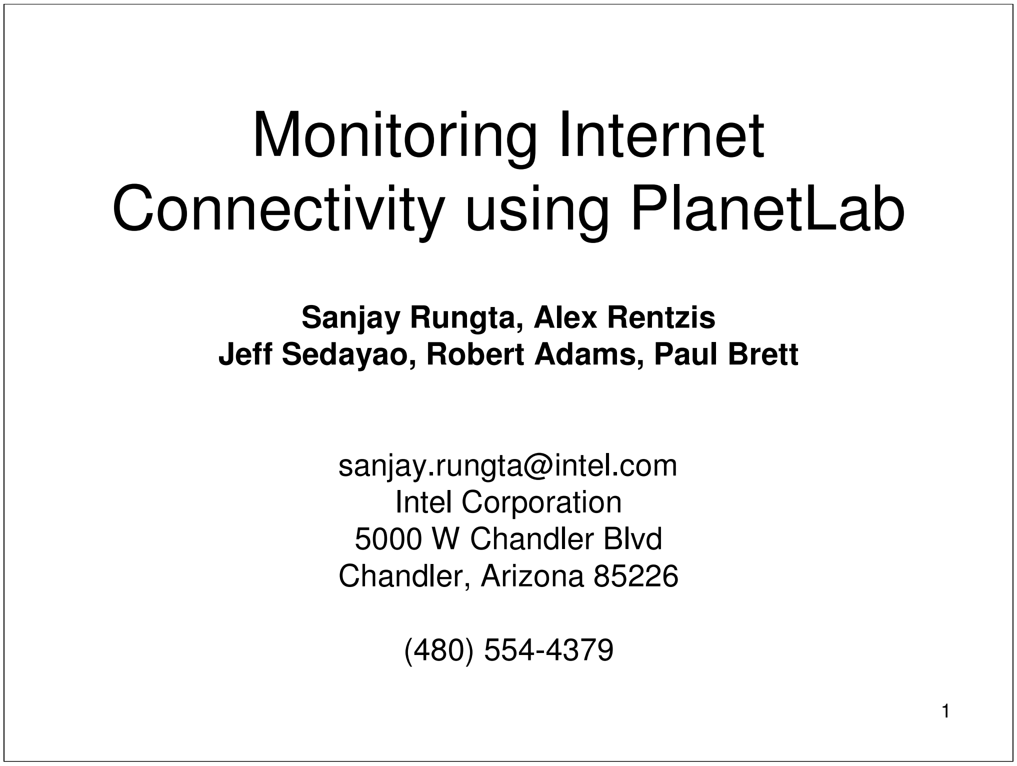


Monitoring Internet Connectivity using PlanetLab

Sanjay Rungta, Alex Rentzis
Jeff Sedayao, Robert Adams, Paul Brett

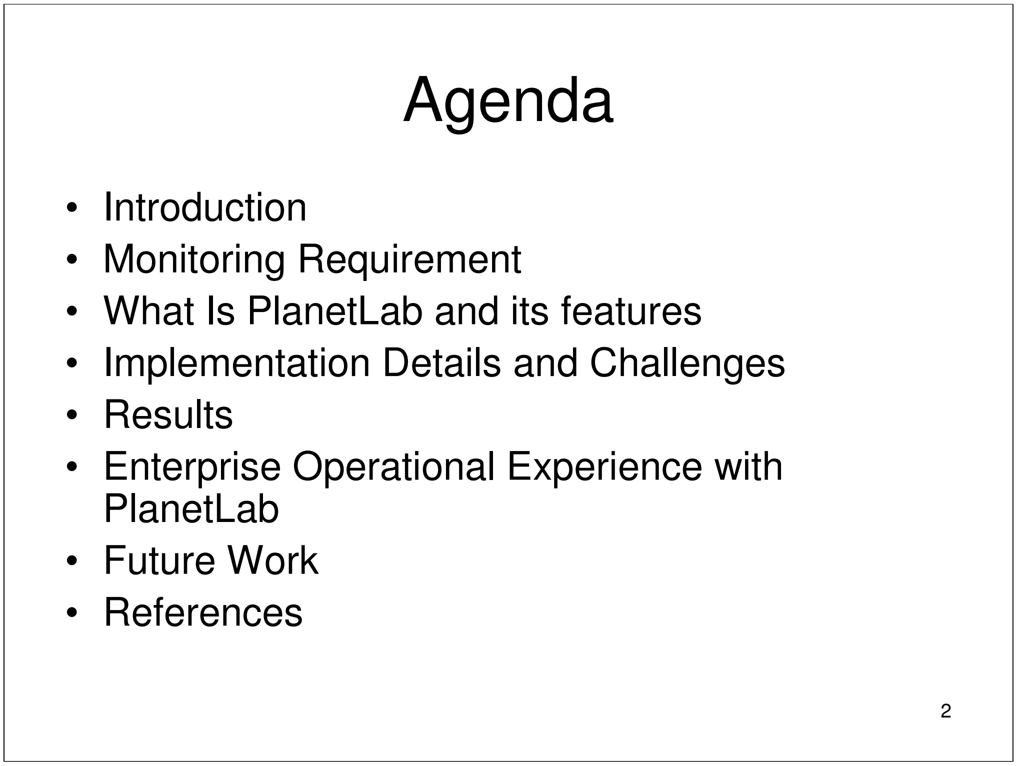
sanjay.rungta@intel.com
Intel Corporation
5000 W Chandler Blvd
Chandler, Arizona 85226

(480) 554-4379

Abstract:

This paper explores one company's use of PlanetLab for a real application. Intel Corporation is a global enterprise with many Internet "DMZs" and thousands of customers around the world who use them. Intel needs to monitor the quality of service received through these Internet connections from many parts of the world. Doing this with available commercial services or by implementing monitoring systems in rented data center space across the globe would be expensive as well as being relatively inflexible. PlanetLab presents a relatively inexpensive and flexible platform for global scale monitoring but poses significant challenges in developing, deploying, and managing such a widely distributed application in an environment where node available and connectivity can change rapidly. We implemented the global DMZ monitor using PlanetLab nodes and the Distributed Service Management Toolkit (DSMT). DSMT provides a way to distribute code for an application and manage it despite node outages, moving the application to geographically appropriate nodes when nodes become unavailable. We position graphs to allow us to correlate data to either geographical local events or Internet wide events. Connectivity events are propagated using the PSEPR eventing system. Our experience with this implementation has shown that it can detect problems Internet connectivity problems. Future work includes using different protocols such as HTTP for monitoring and to extend DSMT services to monitor other conditions.

1-4244-0143-7/06/$20.00 ©2006 IEEE



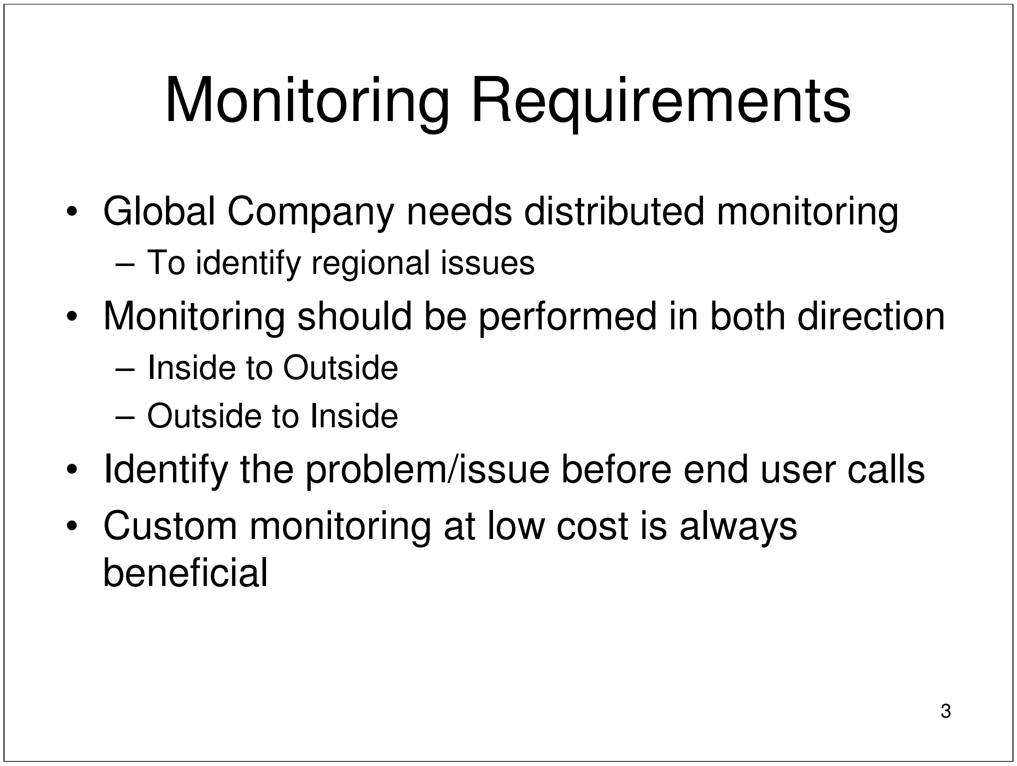
Agenda

- Introduction
- Monitoring Requirement
- What Is PlanetLab and its features
- Implementation Details and Challenges
- Results
- Enterprise Operational Experience with PlanetLab
- Future Work
- References

Monitoring the availability and service quality that an enterprise DMZ receives from its Internet service providers can be complex and difficult. Intel Corporation is a global enterprise with many DMZs and customers that use them all around the world, both inside and outside of the corporate firewalls. Ideally we would monitor the connection from outside of the DMZ from areas around the world. Doing this with available commercial services is expensive and relatively inflexible, as we would be limited to the services and monitoring locations provided. Implementing monitoring systems in rented data center space across the globe would be more flexible, but definitely more expensive. We can achieve this monitoring both inexpensively and with the global coverage that we require through PlanetLab, but this approach poses challenges with keeping the monitoring application going despite node outages.

This paper describes how we implemented a monitoring system that looks at DMZs from a global viewpoint. We first layout the requirements and challenges for Internet service infrastructure monitoring. We provide an overview of PlanetLab, followed by a section on related work. Next, our design and implementation are described, as we highlight the Distributed Service Management Toolkit that enables application deployment and management.. The last section describes our experiences and plans for future work and discusses how other organizations can take advantage of PlanetLab to monitor their own DMZs or globally deployed services.

1-4244-0143-7/06/$20.00 ©2006 IEEE



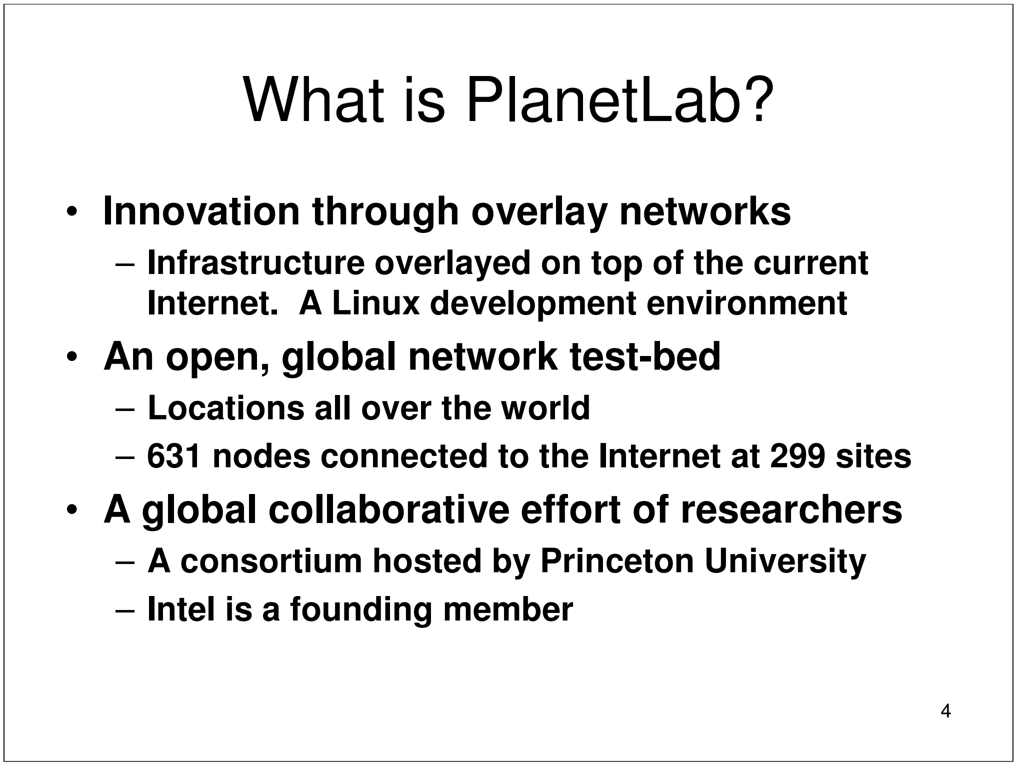
Monitoring Requirements

- Global Company needs distributed monitoring
  - To identify regional issues
- Monitoring should be performed in both direction
  - Inside to Outside
  - Outside to Inside
- Identify the problem/issue before end user calls
- Custom monitoring at low cost is always beneficial

A global organization such as Intel has many Internet users scattered across the planet. Some are Intel customers, some are suppliers, and some are employees. Employees can be within Intel's firewalls or working remotely from home or from customer sites located anywhere on the globe. Services that are utilized include web sites such as Intel's corporate presence at www.intel.com, various e-commerce applications, and VPN connectivity back into Intel. This requirement for global access can result in Intel's Network Operation Center (NOC) receiving complaints about performance from any spot on the planet to any one of Intel's many DMZ zones. For example, the NOC might get a call from a user in China saying that they response for an e-commerce application that they are using is very poor. Is the problem local to China? Is the problem local to the Internet connection in question? Is the problem Internet wide? The NOC needs tools to be able to answer those questions.

A key question that comes from this discussion is from where to monitor. The typical DMZ firewall model lends itself toward monitoring the DMZ systems from within the DMZ. This ends up creating a monitoring model with limited scope that does not address problems with transit from anywhere in the world to the DMZ. An alternative would be an approach that examined web logs for performance problems [1] or looked at traffic flow data using Cisco Netflow [2]. Because of our traffic volume and the fact that we didn't have web servers at all of our Internet DMZs, we ruled out this option. It would be extremely useful to be able to proactively monitor for performance problems all around the world using active measurements. Active measurements from regions in the world could be done from commercial services like Keynote [3] or be done from hosts in data centers strategically placed around the world. Using commercial services would limit the kind of applications we could run to monitor the DMZs and be fairly expensive. Deploying our own hosts in the locations around the world where we want to monitor from would permit much more flexibility, but we would be even more expensive.

PlanetLab [4] presents a relatively inexpensive and flexible platform for global scale monitoring, but poses challenges with software distribution and application management. A key problem is that nodes go up and down, and if the nodes monitoring a particular Internet connection from a particular geography go down, that monitoring perspective is lost. Even as nodes come and go, we need to make sure that the software running the monitor application and any additional configuration files are synchronized.



What is PlanetLab?

- **Innovation through overlay networks**
  - **Infrastructure overlayed on top of the current Internet. A Linux development environment**
- **An open, global network test-bed**
  - **Locations all over the world**
  - **631 nodes connected to the Internet at 299 sites**
- **A global collaborative effort of researchers**
  - **A consortium hosted by Princeton University**
  - **Intel is a founding member**

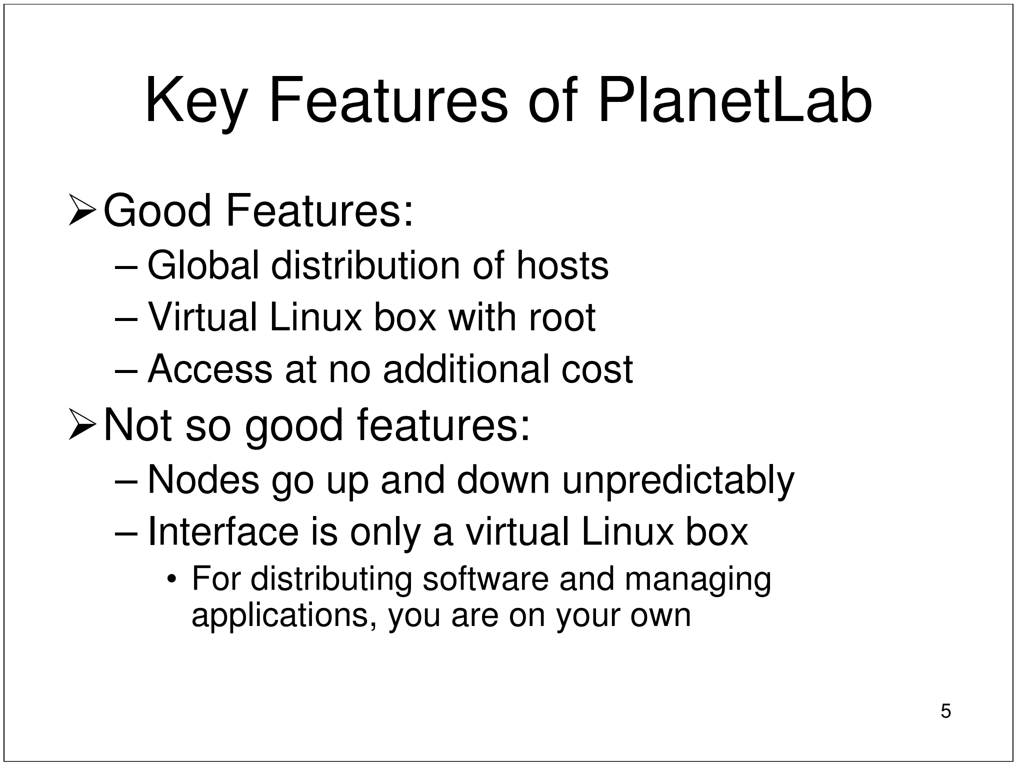
4

PlanetLab [4] was born out of the demands of professionals in the field of distributed system research, who wanted to research global-scale distributed services without investing in separate testbeds. While many of the researchers had ideas for services and experiments that would work on a global scale, there was no truly global testbed to try out and validate those ideas. Another challenge was that the Internet has become so important for every day use that it was impossible to directly experiment with it. Intel joined with leading distributed system researchers and funded the first set of PlanetLab nodes spread across the world. PlanetLab was envisioned as a test ground for next-generation global-scale Internet services. It would be an overlay network–an application and service layer living over the Internet in the same way that the Internet was an overlay on top of the global telephony infrastructure. As the Internet becomes more ossified and harder to change, PlanetLab's design allows its nodes to host innovated new services without affecting other existing services.

PlanetLab has evolved to become many things. It is a network and server infrastructure for testing global-scale services and experiments, at comprising 587 nodes at 280 sites (at the time this was written). PlanetLab is truly "planetary scale" as it is geographically spread across five continents and topologically spread across the Internet, Internet2, and other networks. Because of this geographic and network diversity, the test bed provides researchers with a very "real-world" set of opportunities and challenges; specifically it allows the deployment of, experimentation with, and test/measurement of services in a non-simulated network. Significant numbers of papers at leading distributed systems conferences describe work using PlanetLab.

PlanetLab is also a consortium of universities, corporations, and research institutions that run and make available a global testbed. It is a set of technologies and standards for running distributed applications, as well as a platform deploying those applications and services. Finally, it is also a way of driving innovation through the use of overlays and overlay technologies, as well as an open platform encouraging cooperation. It is notable that there are very few non-academic enterprises that are members of PlanetLab. What would enterprise want to do with PlanetLab? We explore that idea in the rest of the paper.

1-4244-0143-7/06/$20.00 ©2006 IEEE



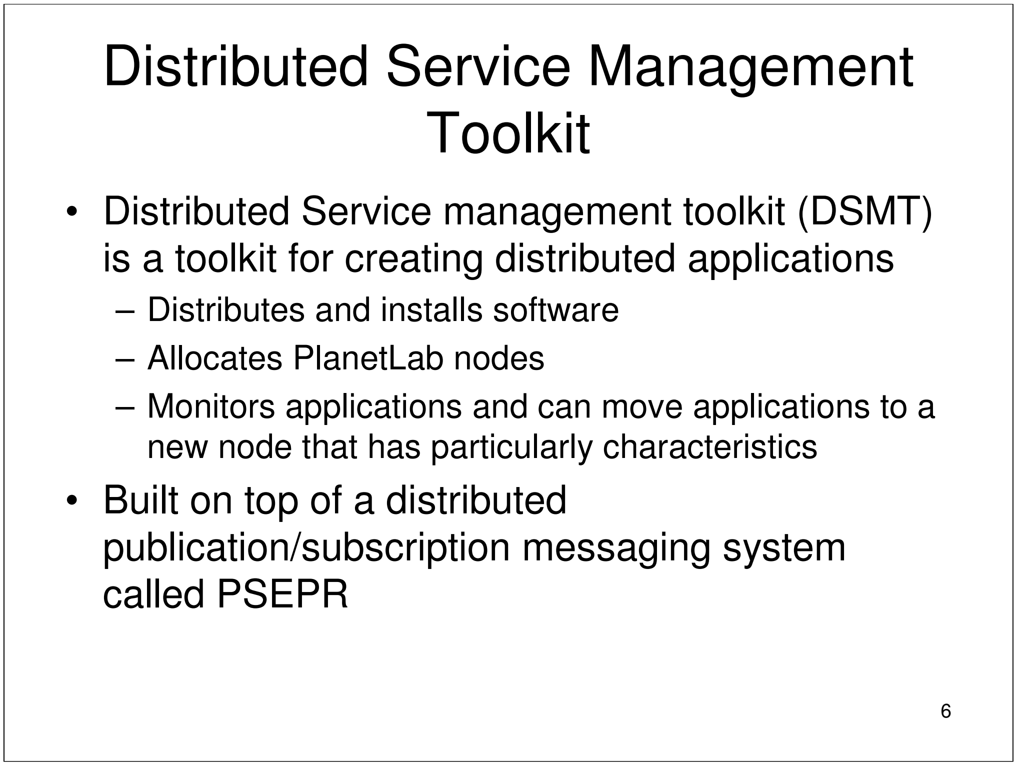
Key Features of PlanetLab

- Good Features:
  - Global distribution of hosts
  - Virtual Linux box with root
  - Access at no additional cost
- Not so good features:
  - Nodes go up and down unpredictably
  - Interface is only a virtual Linux box
    - For distributing software and managing applications, you are on your own

To serve the potentially large number of distributed researchers that would use the PlanetLab infrastructure, the implementers of PlanetLab created the abstraction known as a “slice”. A slice is piece of the PlanetLab infrastructure given to researchers, experimenters, and service implementers to use. To a person implementing a service on PlanetLab, a slice is a set of virtual machines on some set of PlanetLab nodes defined by that person. Each virtual machine appears to be a complete Linux* machine with root access. PlanetLab’s virtualization, through the vserver package, happens at the system call level and allows us to scale to up to 1000 virtual machines per node. PlanetLab allows allocation of a virtual Linux machines at any or even all of the PlanetLab lab nodes across the globe.

While PlanetLab offers tremendous flexibility and global reach for services, there are a number of downsides to the slice paradigm. When you get a slice, all you get is a virtual linux box. The native slice provides no way to distribute and manage applications. Moreover, the linux box you get can go up and down unpredictably. Any programs or data on the slice may disappear when nodes reboot.

* Other brands and names are the property of their respective owners.



Distributed Service Management Toolkit

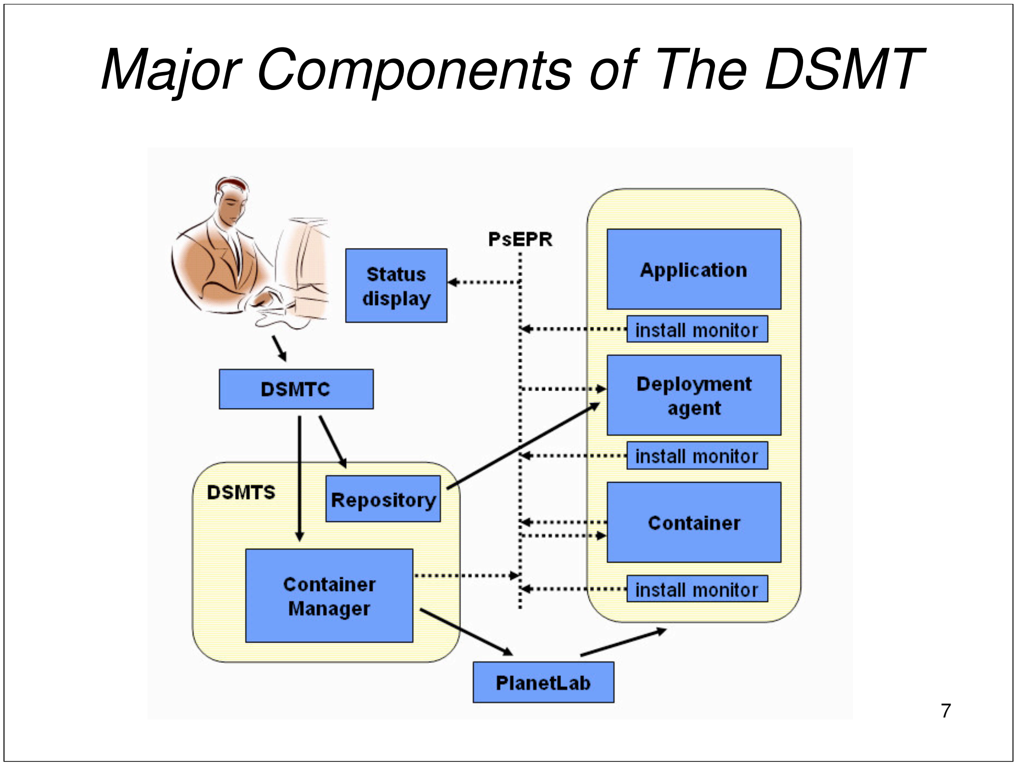
- Distributed Service management toolkit (DSMT) is a toolkit for creating distributed applications
  - – Distributes and installs software
  - – Allocates PlanetLab nodes
  - – Monitors applications and can move applications to a new node that has particularly characteristics
- Built on top of a distributed publication/subscription messaging system called PSEPR

One of the key problems that we mentioned is dealing with nodes that sporadically available. How do we update the software on nodes that may be unreachable? How does monitoring work when nodes go down? To deal with these issues, we turned to the Distributed Service Management Toolkit.

The Distributed Management Toolkit ("DSMT") is a service and collection of tools which, given installable packages and a description of the required PlanetLab node characteristics, will allocate PlanetLab virtual machines on nodes and install, run and monitor the application. Additionally, the DSMT service monitors the operation of the application and, if an instance of the application stops running, will provision and install the application on another suitable node. In the case of our monitoring application, it will pick a node within the geographical coordinates for the region from which want a monitoring perspective.

The mechanism that ties the DSMT with the instances is a loosely coupled, publish/subscribe messaging system. The Planetary Scale Event Propagation and Routing system ("PsEPR" pronounced "pepper")[5] uses an overlay network to move XML messages from senders to receivers who have subscribed to the events. The publish/subscribe model based on around 'channels,' a name space that senders place events in and the receivers subscribe to.

The purpose of the PsEPR eventing system is for the multitude of components to share status information. The Container Manager outputs status messages as it selects and provisions PlanetLab virtual machines. Each of the components in the provisioned system have Installation Monitors which output status on PsEPR channels as the component is installed and instantiated. The Deployment Agent outputs status messages on PsEPR channels as it monitors the operation of the application.



Major Components of The DSMT

DSMTS includes the Container Manager service. This service takes the administrator's node specification, performs the following steps:

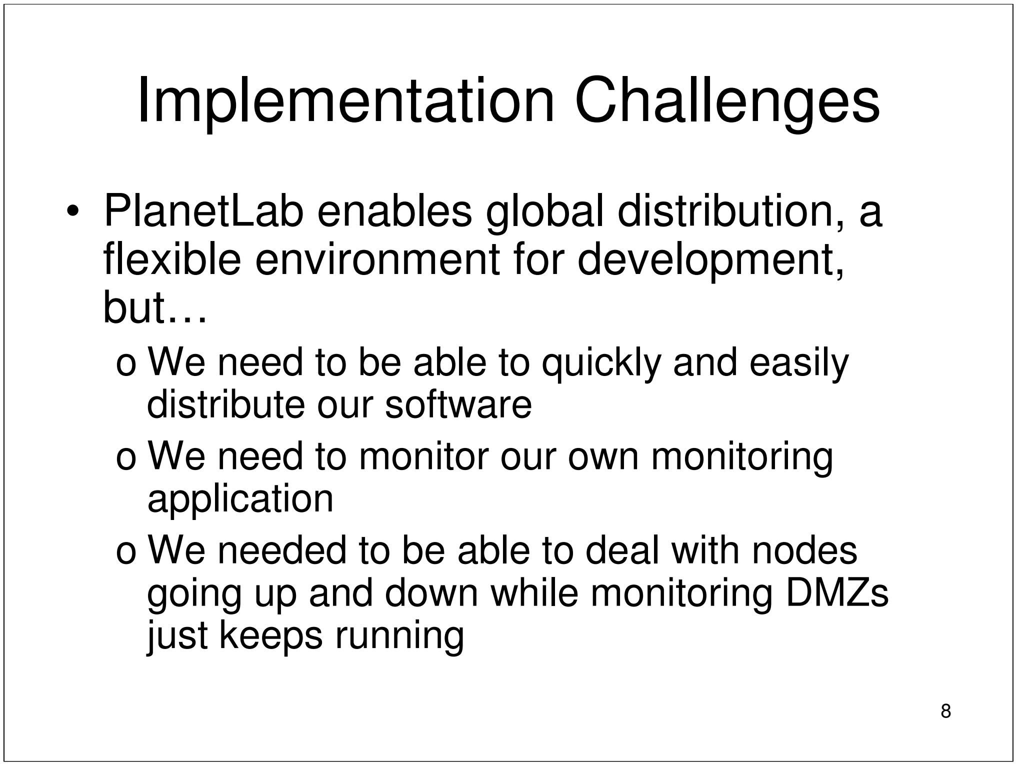
•creates PlanetLab virtual machines on nodes that fit the user's criteria;

•installs a Container controller in the virtual machine;

•installs a Deployment Agent ;

•starts the Deployment Agent.

The Deployment Agent installs the application instance on the node by pulling the application installation file from the repository. Because the PlanetLab virtual machines run a version of Linux, the application is supplied as an RPM. The application's RPM is copied by DSMTC to the repository for later access by the Deployment Agent. There are currently Deployment Agents for several different types of distribution systems

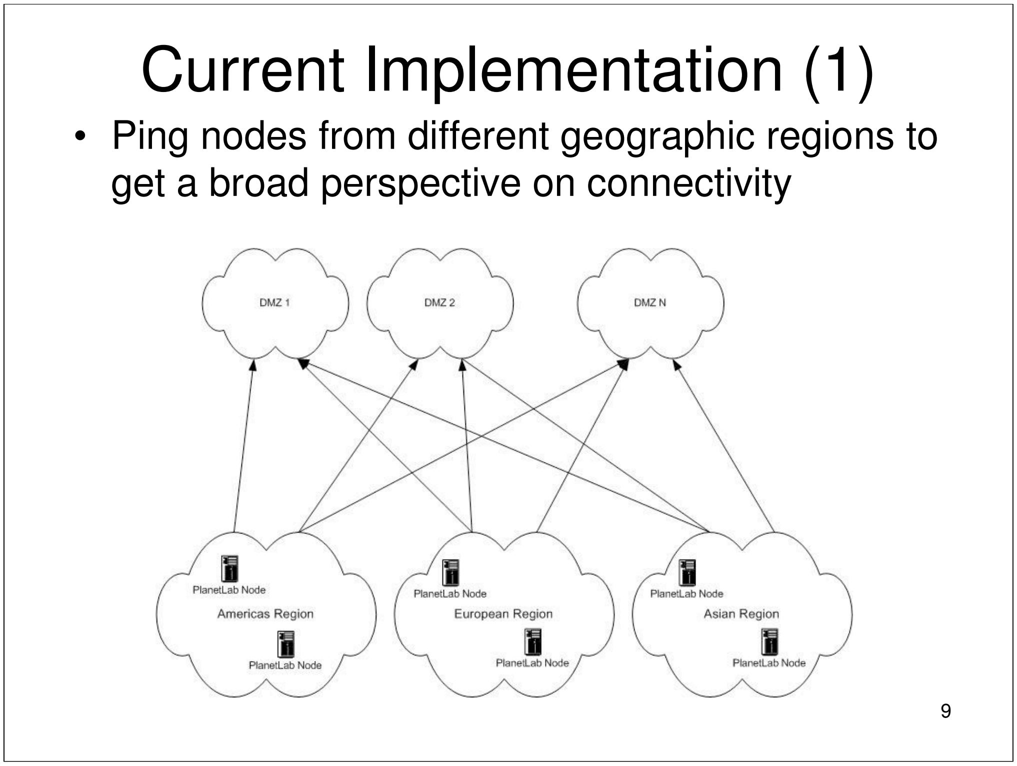


Implementation Challenges

- PlanetLab enables global distribution, a flexible environment for development, but…
  - o We need to be able to quickly and easily distribute our software
  - o We need to monitor our own monitoring application
  - o We needed to be able to deal with nodes going up and down while monitoring DMZs just keeps running

Using PlanetLab presents a number of unique challenges to enterprise IT staff. Most IT staff are used to working within a typical IT environment where there is a Service Level Agreement (SLA) governing the availability and responsiveness of individual hosts. Penalties and remedies are available if the SLA is not met, giving much incentive to keep server hosts up and functioning. PlanetLab, on the other hand, has no guarantees for individual nodes and requires a totally different way of creating and managing applications. The first impulse of the IT staff implementing the monitor application was simply to write the application and deploy to a fixed set of nodes. The dynamic nature of PlanetLab soon doomed this approach and necessitated use of DSMT.

PlanetLab presents a relatively inexpensive and flexible platform for global scale monitoring, but poses challenges with software distribution and application management. A key problem is that nodes go up and down. If the nodes monitoring a particular Internet connection from a particular geography go down, that monitoring perspective is lost. Even as nodes come and go, we need to make sure that the software running the monitor application and any additional configuration files are synchronized.



Current Implementation (1)

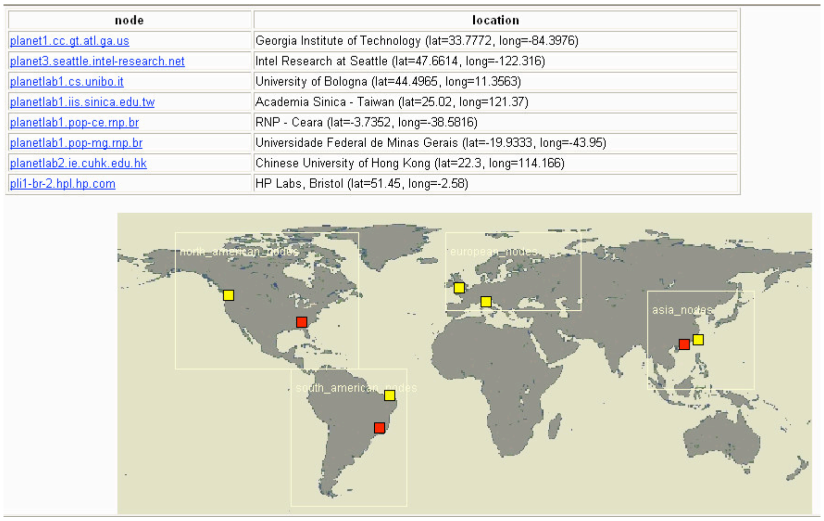
- Ping nodes from different geographic regions to get a broad perspective on connectivity

To get global coverage we started monitoring the DMZ from multiple sites -- two from Asia, two from Europe and few from North America using ping, as shown above. Each of the nodes monitors all DMZs worldwide and graphical representation is created to assist first line support staff.

1-4244-0143-7/06/$20.00 ©2006 IEEE

Current Implementation (2)

Global Console for first line support staff

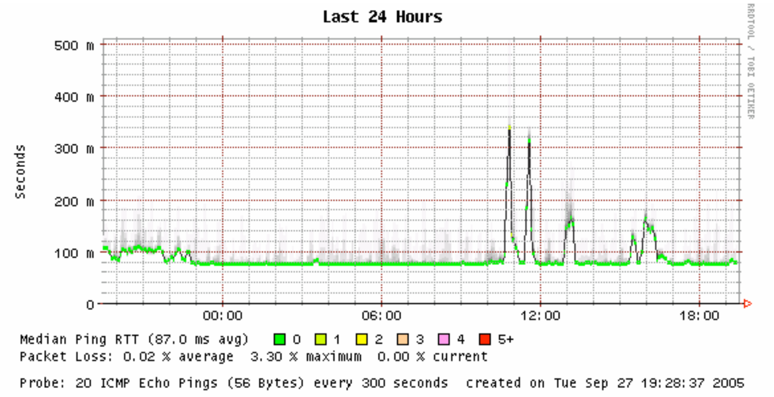


node	location
planet1.cc.gt.atl.ga.us	Georgia Institute of Technology (lat=33.7772, long=-84.3976)
planet3.seattle.intel-research.net	Intel Research at Seattle (lat=47.6614, long=-122.316)
planetlab1.cs.unibo.it	University of Bologna (lat=44.4965, long=11.3563)
planetlab1.iis.sinica.edu.tw	Academia Sinica - Taiwan (lat=25.02, long=121.37)
planetlab1.pop-ce.rnp.br	RNP - Ceara (lat=-3.7352, long=-38.5816)
planetlab1.pop-mg.rnp.br	Universidade Federal de Minas Gerais (lat=-19.9333, long=-43.95)
planetlab2.ie.cuhk.edu.hk	Chinese University of Hong Kong (lat=22.3, long=114.166)
pli1-br-2.hpl.hp.com	HP Labs, Bristol (lat=51.45, long=-2.58)

1-4244-0143-7/06/$20.00 ©2006 IEEE

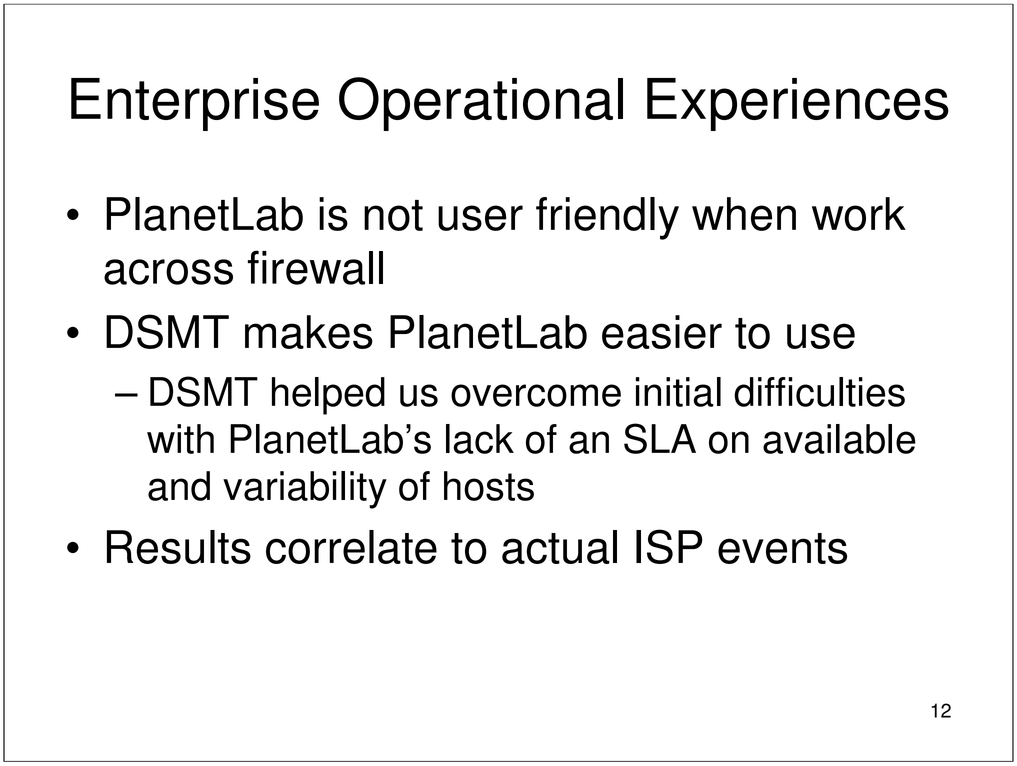
Current Implementation (3)

- We use feed DSMT with our geographic criteria and our software manifest
  - DSMT deploys the application
  - DSMT then monitors it
- Graph data and send alert e-mail if thresholds are crossed:



The administrator deploying the DMZ monitoring application communicates with the DSMT services ("DSMTS") though a command interface ("DSMTC"). Through this interface, the administrator supplies the installable instance of the application and a ruleset specifying the nodes to install the application instances.

Based upon the active monitoring data, the statistical average and standard deviation is calculated for each DMZ. When the active monitoring data exceeds a threshold (we use the mean plus one standard deviation), an alert is sent to proactively notify operations staff. NOC staff can look if a problem is Internet-wide, confined to a region, or specific to a particular DMZ.



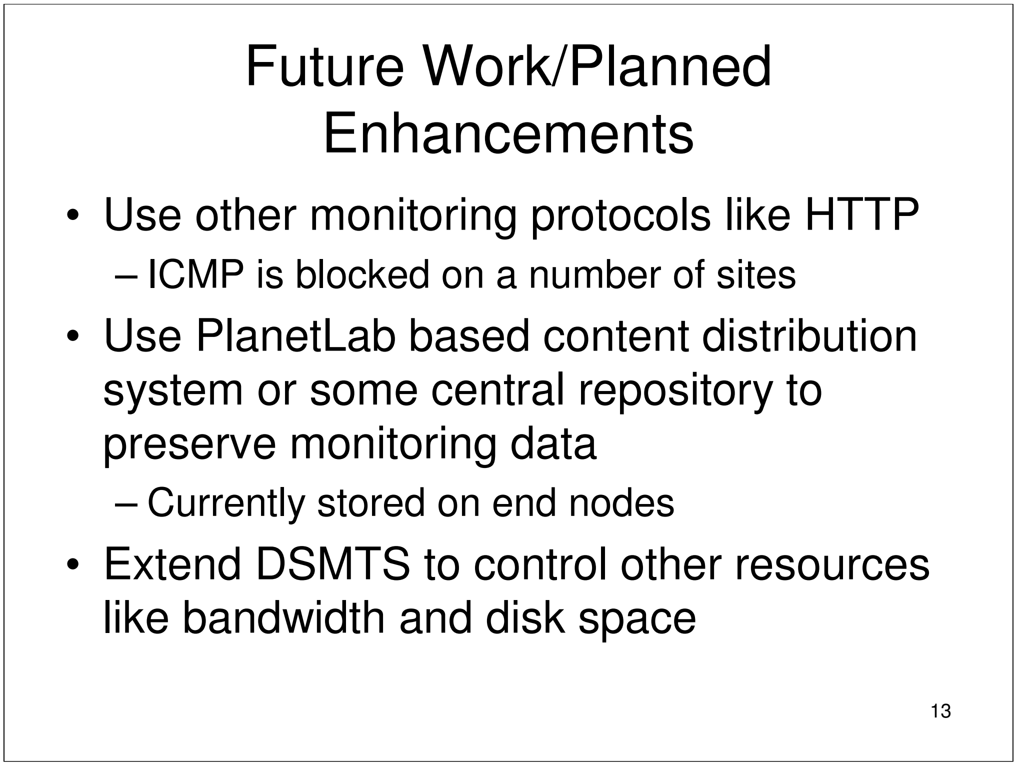
Enterprise Operational Experiences

- PlanetLab is not user friendly when work across firewall
- DSMT makes PlanetLab easier to use
  - DSMT helped us overcome initial difficulties with PlanetLab's lack of an SLA on available and variability of hosts
- Results correlate to actual ISP events

DSMT showed that installing the deployment service and application runtimes themselves is significantly more effort than deploying the applications. This provides a good justification for provisioning services.

Because of the disperse and dynamic nature of PlanetLab, the only real info on whether a service will run on a box is to try it - 'predictive' information is of limited value. Several services are available on PlanetLab which attempt to detect whether a particular node is 'available'. With DSMT, is was found to be easier and, in the long term, more reliable to attempt to install and run the application on a node and merely restart the installation on another node if unsuccessful.

Does the monitoring work? Intel's IT staff has correlated Internet Service Provide availability events to identifiable parts of the graph, so the monitoring approach has been validated



Future Work/Planned Enhancements

- Use other monitoring protocols like HTTP
  - ICMP is blocked on a number of sites
- Use PlanetLab based content distribution system or some central repository to preserve monitoring data
  - Currently stored on end nodes
- Extend DSMTS to control other resources like bandwidth and disk space

We have found that different PlanetLab sites filter different protocols, and it is likely that some rate limit other protocols. We would like to enhance our monitoring by using other protocols other than ICMP, such as HTTP.

The graphs depicting DMZ connection quality are currently displayed from the PlanetLab nodes doing the measuring. While DSMT will migrate the monitoring function if the nodes fail, previous historical data is lost. We want to evaluate ways to store and preserve historical monitoring data in a robust way and also use Content Distribution Networks to speed the display of connection quality graphs. We will also like to evaluate in bringing all the data back in enterprise (enterprises dislike exposing monitoring data) and use PlanetLab solely as data collection platform.

DSMTS currently contains the Container Manager service which selects and manages a set of virtual machine resources. DSMTS will be extended to control other resources (bandwidth, disk space, etc) and allow the selection and management of multiple resource types. This will allow better control of the environments of the selected nodes and will additionally integrate with various resource allocation systems.

References

[1] Bickerstaff, Cindy, Ken True, Charles Smothers, Tod Oace, Jeff Sedayao, and Clinton Wong. “Don`t Just Talk About the Weather--Manage It! A System for Measuring, Monitoring, and Managing Internet Performance and Connectivity.” First Conference on Network Administration (NETA '99). Santa Clara, 1999.

[2] Cisco Corporation. Netflow,. http://www.cisco.com/warp/public/732/Tech/nmp/netflow/index.shtml

[3] Keynote. http://www.keynote.com/.

[4] PlanetLab. http://www.planet-lab.org/.

[5] P. Brett, R. Knauerhase, M. Bowman, R. Adams, A. Nataraj, J. Sedayao, and M. Spindel. “A Shared Global Event Propagation System to Enable Next Generation Distributed Services.” WORLDS ’04: First Workshop on Real Large Distributed Systems, San Francisco, CA 2004

[6] Yum. http://linux.duke.edu/projects/yum/

[7] L. Wang, K. Park, R. Pang, V. Pai, and L. Peterson. “Reliability and Security in the CoDeen Content Distribution Network”. Proceedings of the USENIX 2004 Annual Technical Conference, Boston, June 2004.

[8] J. Cappos, and J. Hartman: “Stork.” http://www.cs.arizona.edu/stork/.

1-4244-0143-7/06/$20.00 ©2006 IEEE