# Archer – A Community Distributed Computing Infrastructure for Computer Architecture Research and Education

David Wolinsky
ACIS P2P Group
University of Florida

UF UNIVERSITY of FLORIDA ACIS

# Motivation and Goals

- Computer architecture research is driven by quantitative analysis
  - Often involves many simulator iterations
  - Complex setup and configuration of tools (See Simics)
- Collaborative environments needed though there exists a significant entry barrier
  - Hardware and maintenance costs for local resources
  - Time/funds to train and educate students and staff (deploy, maintain, and use local and remote resources)
- Goal: Simple setup, use, and management of wide-area collaborative cycle-sharing systems

# Outline

- Archer Design Overview
- The Components
  - Condor
  - Virtual Machines
  - Virtual Networking
- The User Experience
  - Joining and Using
  - Learning and Contributing
- Wrap Up

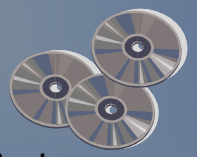
# Outline

- Archer Design Overview
- The Components
  - Condor
  - Virtual Machines
  - Virtual Networking
- The User Experience
  - Joining and Using
  - Learning and Contributing
- Wrap Up

# Archer Design Goals

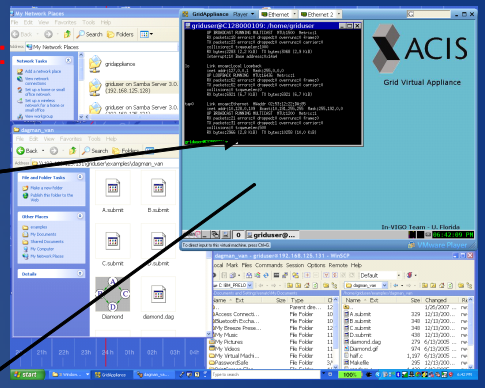
- A computing infrastructure which scales in capacity with community buy-in:
  - Archer starts from a seed set of cluster resources.
  - Each new user seamlessly contributes to its aggregate capacity.
- A system easy for non-experts to join and use:
  - Archer relies on packaging and distribution of self-configuring virtual networks of pre-packaged virtual machines (VMs)
- A repository of reusable simulation environments:
  - Archer allows sharing not only of hardware resources, but also of full-fledged software simulation modules
    - application executables, scripts, input/output data sets

# Archer Overview



**1: Download** appliance

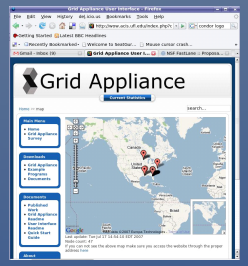
**2. Boot** appliance: automatically joins Archer pool




**Free pre-packaged** Archer Virtual appliances - run on free VMMs (VMware, VirtualBox, KVM)

***Archer Global Virtual Network***

**3. Run** architecture simulation jobs on the Archer pool through Condor




**Portal and Wiki:** Community-contributed content: applications, datasets, tutorials

**Archer seed resources**
300+ cores – Fall 2008

**Simulators:**
Simics, SESC, Simplescalar

**System software:**
Condor scheduler
NFS file systems

UF UNIVERSITY of FLORIDA ACIS

# Use-Case Scenarios

- Easy access to remote cycles:
  - Individual users without access to local hardware resources for simulation
- Local resource pooling shared with community:
  - Groups of users within or across institutions
    - Scheduling can be better managed by Archer's middleware than ad-hoc approaches
    - Increase utilization of local resources, "spill out" to global
- Private resource pools
  - Reuse Archer self-configuring VM and middleware
- Collaborative development and dissemination of tools and experiments

# Usage Models

- Archer Express – Globally public pool, free to test out with minimal seed hardware
- Archer Global – Registered Comp Arch users only, access to Archer seed resources (soon ~400 CPUs)
- Archer Local – Private pool for users or institutions can be shared or not, but it is not globally visible

# Outline

- Archer Design Overview
- The Components
  - Condor
  - Virtual Machines
  - Virtual Networking
- The User Experience
  - Joining and Using
  - Learning and Contributing
- Wrap Up

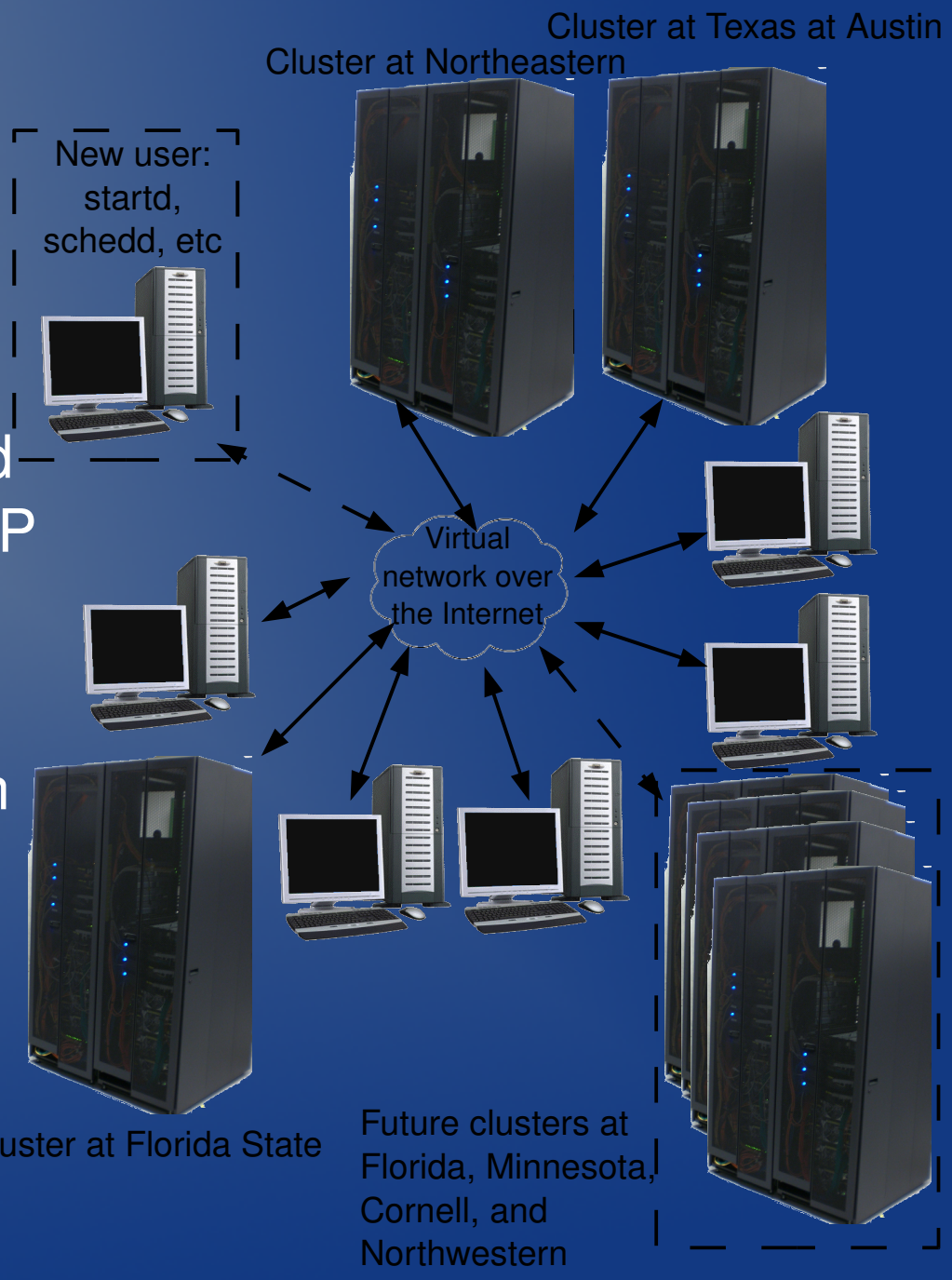
# The Components

- Goal – User-friendly and Autonomic via high-throughput computing over LAN and WAN
- Use Virtualization!
  - Virtual Networks (VPNs) provide a secure, private LAN over WAN; autonomic and capable of firewall/NAT traversal
  - Virtual Machines (VMs) provide sandboxing, software packaging, and homogenous resources
- High-throughput Computing with Condor provides reliable task scheduling, fair sharing of resources, privilege to user owned resources, and unmodified applications (if they run on Linux)

# Wide-area Clusters

Archer

- Self-contained VM Appliance
- Configuration through virtual floppy
- Fully distributed, decentralized Virtual Private Network via P2P Overlay
- Job scheduling via Condor
- Customization through Debian and Stacked File system




UF UNIVERSITY of FLORIDA ACIS

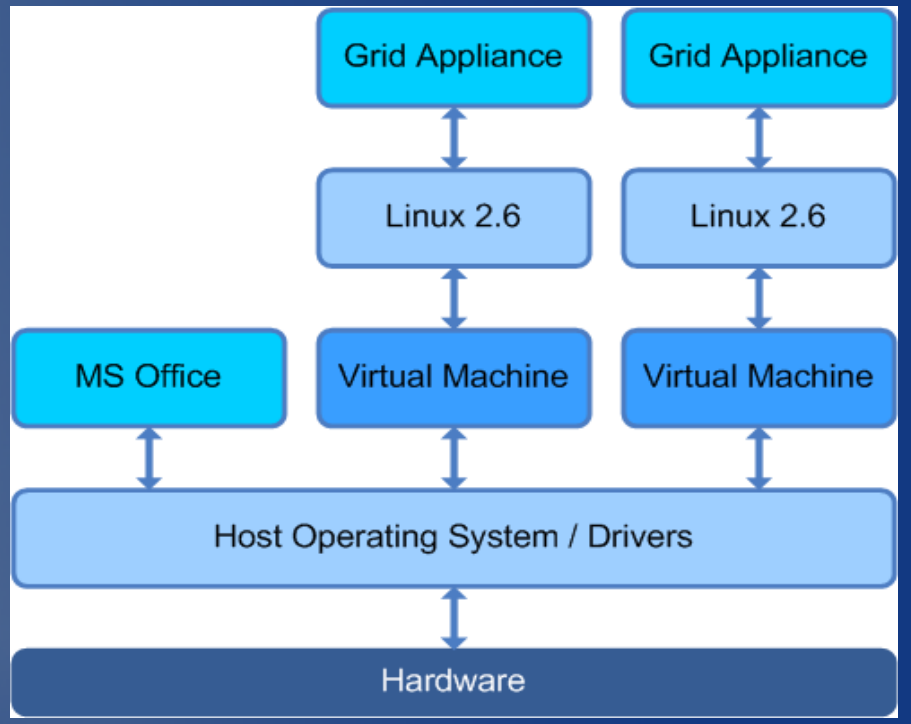
# Condor

Condor is an established distributed computing environment for high-throughput computing

- Research, software development and deployment of Condor for nearly 20 years at U. Wisconsin Madison
- Evolved from a local batch management system into a full-fledged distributed computing environment
  - Wide-area grids, complex workflows, compute-intensive applications
  - Data placement, resource monitoring, job scheduling, and workflow supervision.
  - Both dedicated compute clusters and non-dedicated machines under the control of interactive users or autonomous batch systems.
- Condor has been deployed on more than 100,000 computers in more than 1400 Condor pools
- http://www.cs.wisc.edu/condor

# Hosted virtual machines

- Reuse wealth of O/S tools
  - VM image = files
  - VM  instance = process
- Low CPU Overhead
- Homogenous resources in heterogenous environments
- Easy to install – Requires VMM
  - KVM, VMware
  - VirtualBox, Parallels
- Plug-and-Play
  - Custom tailored configuration
  - All necessary tools, scripts, libraries, and configuration in a single image
- Customizable via downloadable virtual floppy image

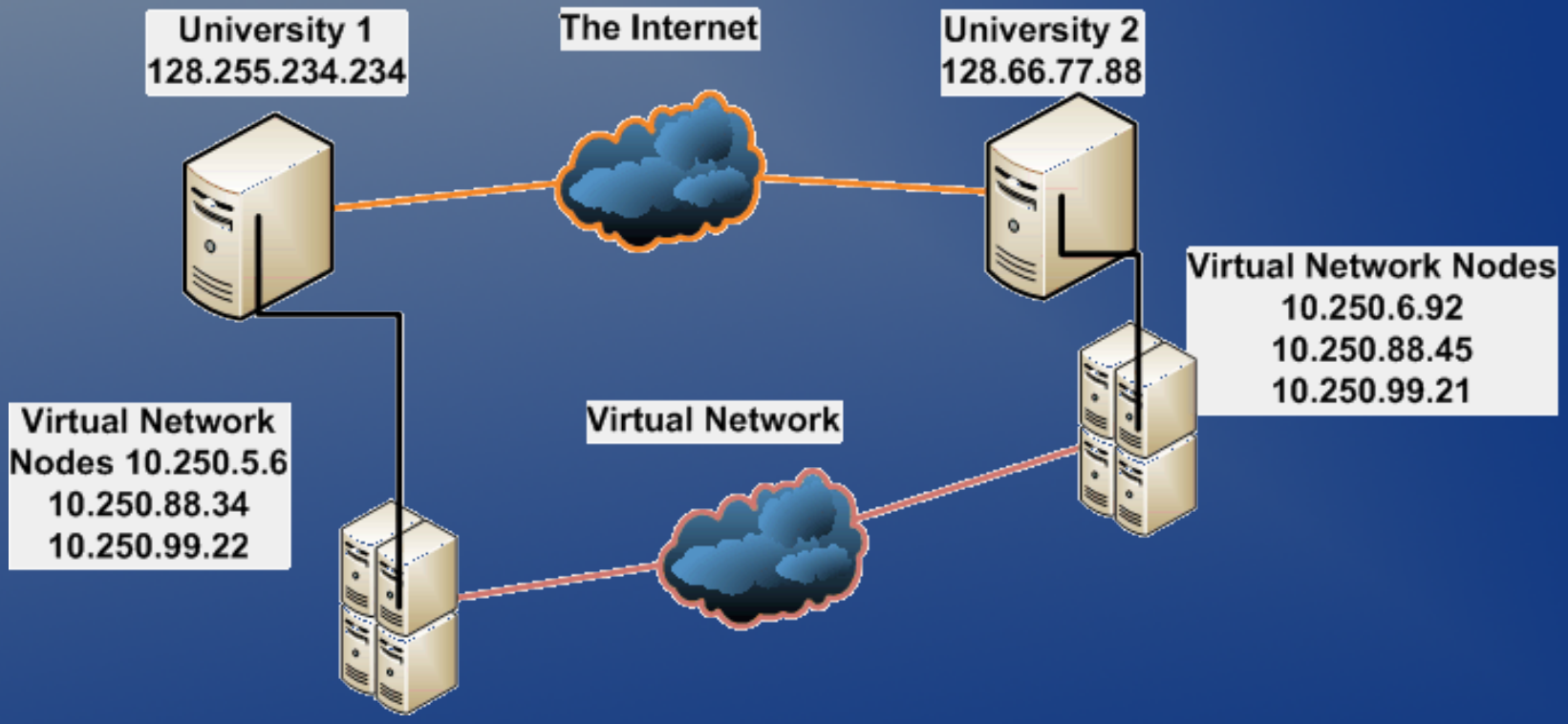


# Configurable Components

- Some Distributed with Floppy, others stored in distributed system
- Condor services to start (machine type)
  - Worker – execution only - startd
  - Client - submission and execution – startd and schedd
  - Master / Server – negotiator and collector
- Which Master / Server to connect or flock to
- User / Group Resource ownership and preemption
- Client can share files via autofs enabled NFS

# Virtual Networking

- Unified layer 3 (IP) network for all machines
- Cross-site communication without a middleware broker

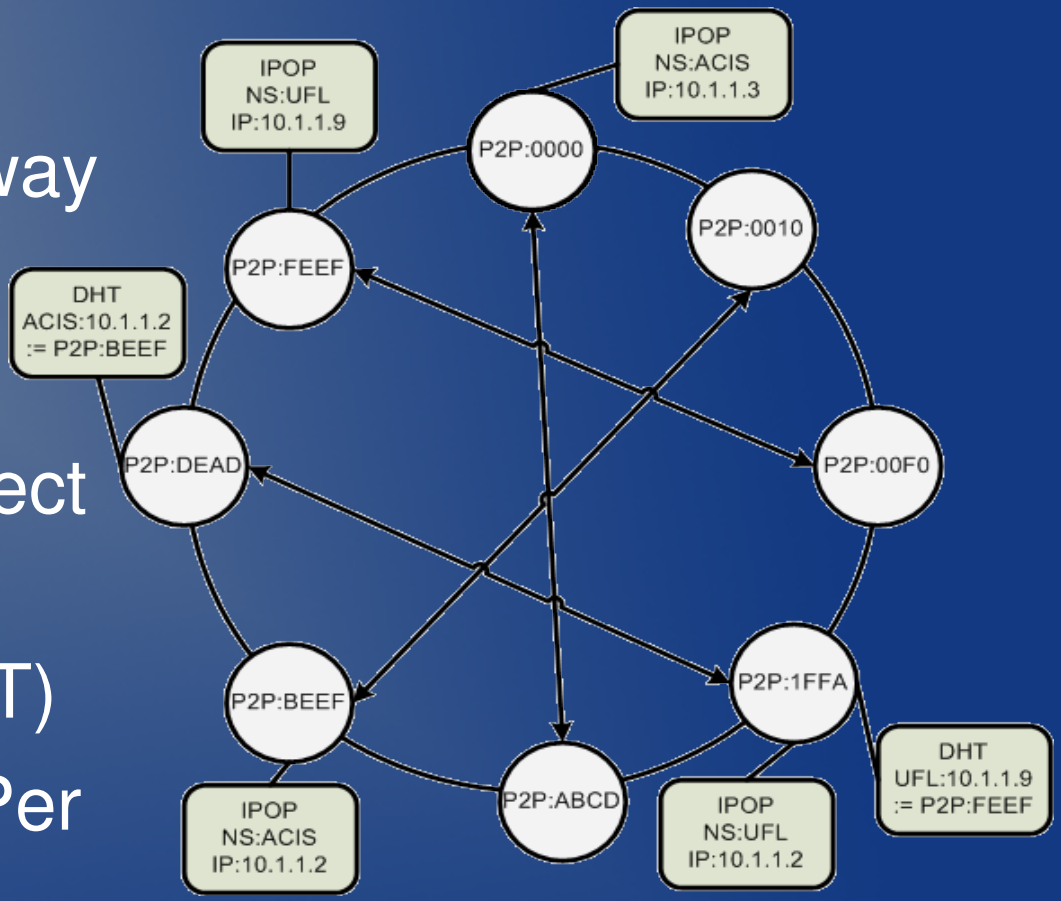


# VN Interfaces

- Each machine has VN Interface running locally
- Machine has VN and “Internet” connectivity

# IPOP

- Open Source
- NAT Traversal (STUN)
- Transparent Subnet Gateway
- Structured P2P Network Overlay
- Provides tunneling and direct shortcuts
- Distributed data store (DHT)
- Multiple Virtual Networks Per Overlay




UF UNIVERSITY of FLORIDA ACIS

# GroupVPN – Securing the System

- IPOP with security enforced by group membership
- Easily allows access to non-Grid Appliance
  - NFS Servers
  - External Grid Resources
  - Physical (non-Virtual) resources
- All communication is encrypted and authenticated
- Runs on Windows and Linux

# P2P Overlay

- Several hundred well distributed nodes
- Assist in:
  - Bootstraping
  - NAT traversal
  - Low latency relaying
- Runs on top of Planetlab

# Two Forms Groups

- GroupVPN
  - Used to ensure everyone in Archer uses a common, secure network
  - Similar to Open/CiscoVPN
- GroupAppliances
  - Each institution or set of users form a group
  - Ensures priority of the resources for members of the group

# Outline

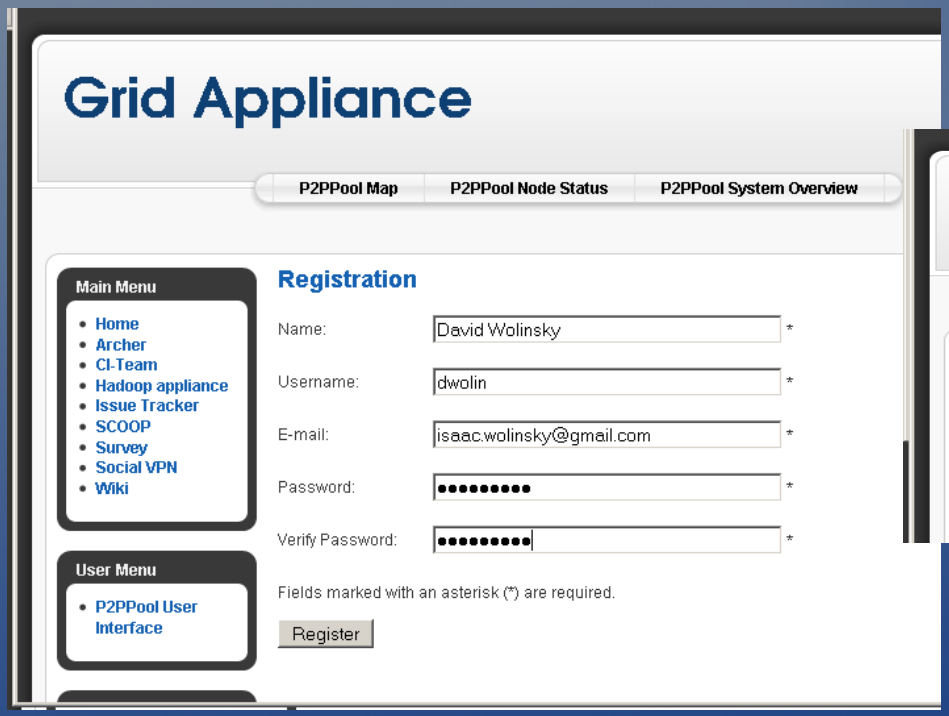
- Archer Design Overview
- The Components
  - Condor
  - Virtual Machines
  - Virtual Networking
- The User Experience
  - Joining and Using
  - Learning and Contributing
- Wrap Up

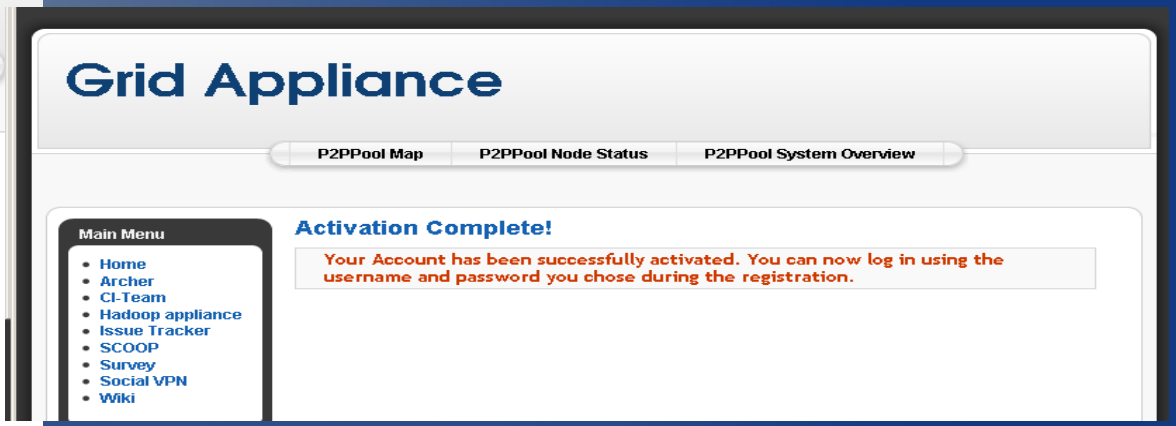
# The User Experience

- Join Archer (www.grid-appliance.org)
- Download Appliance and configuration
- Start VM and check connectivity
- Direct connectivity (i.e. low network overheads)
- Auto NFS
- Simics
- Accessing your data

# Joining Archer

- Sign up at www.grid-appliance.org
- Check your e-mail
- Click activation link

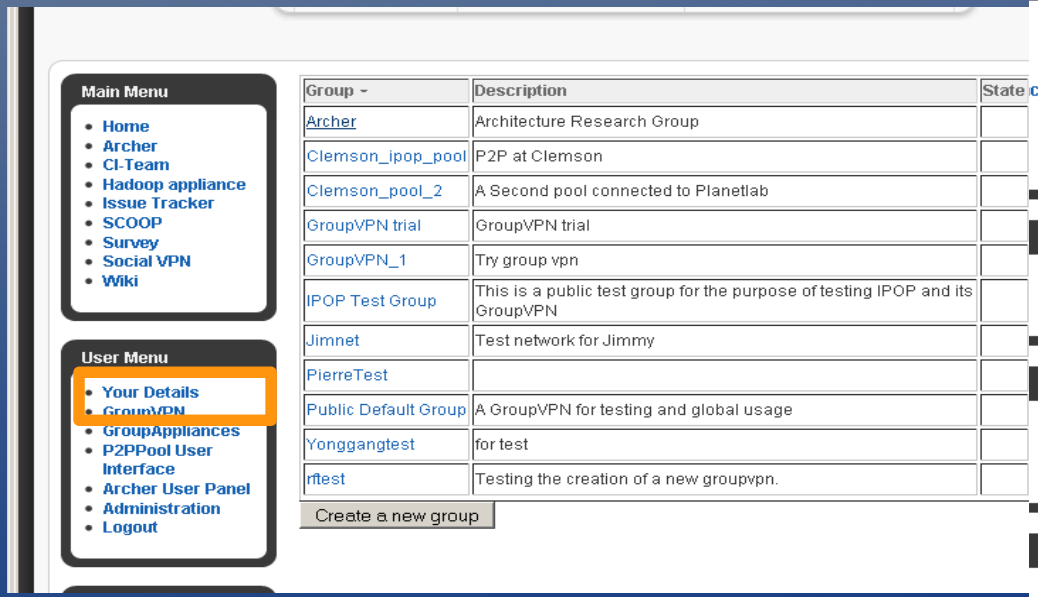


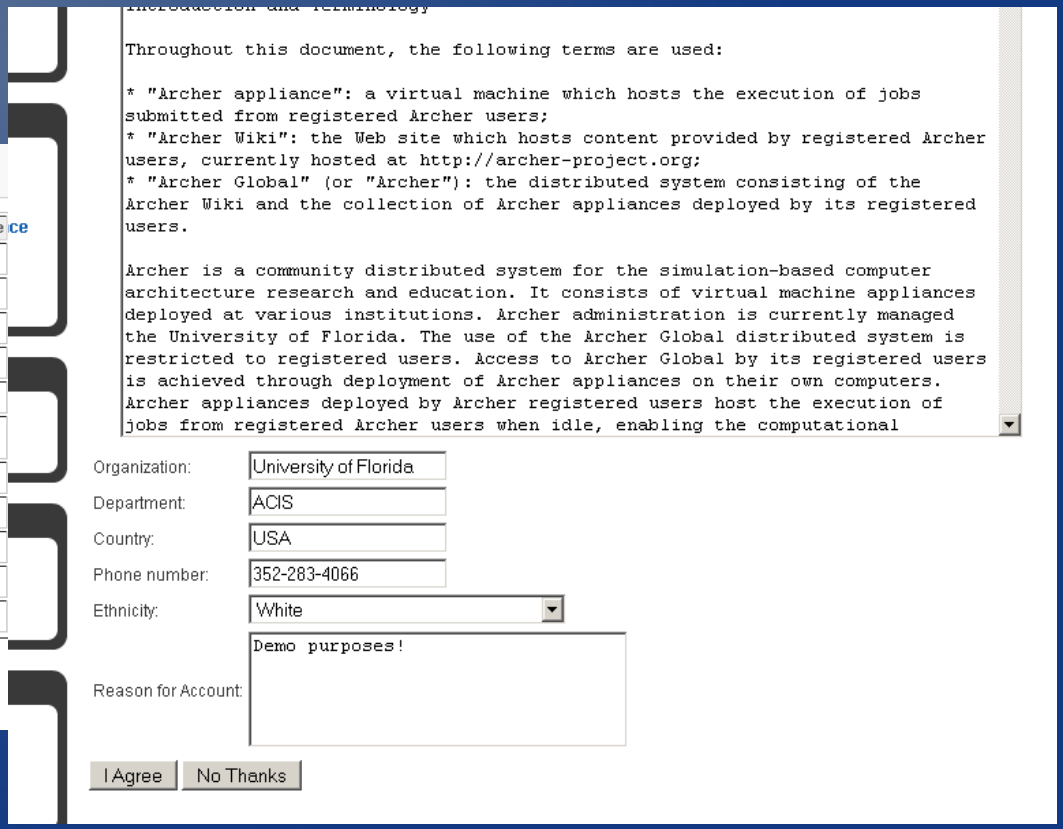




UF UNIVERSITY of FLORIDA ACIS

# Joining Archer

- Register for the Archer GroupVPN
- Detailed information makes NSF happy
- You'll receive an e-mail when the admin for the site (myself or Professor Figueiredo) approves

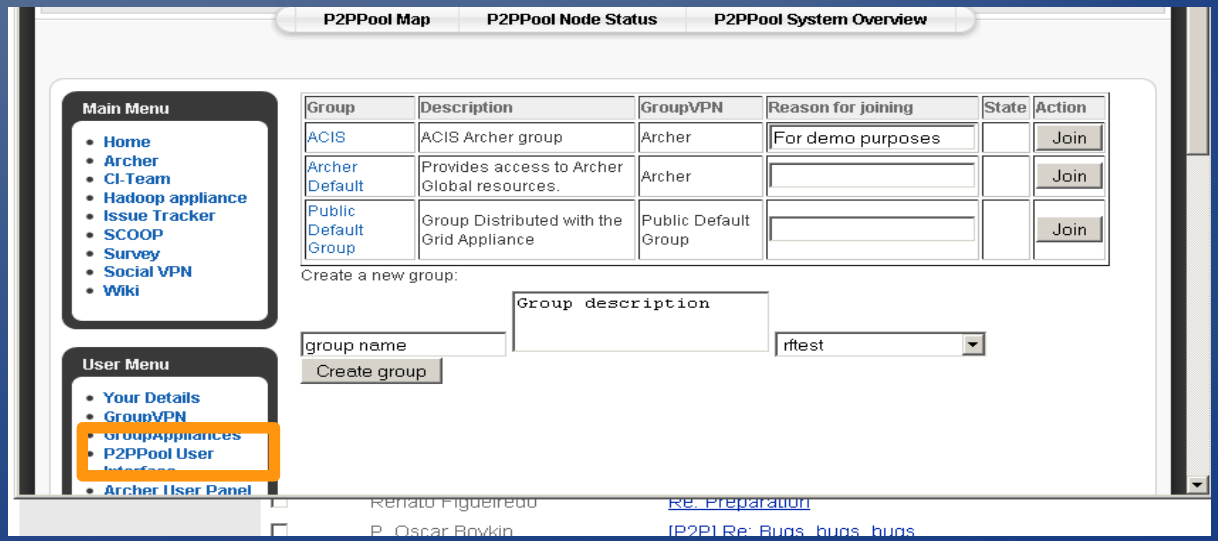



UF UNIVERSITY of FLORIDA ACIS

# Joining Archer

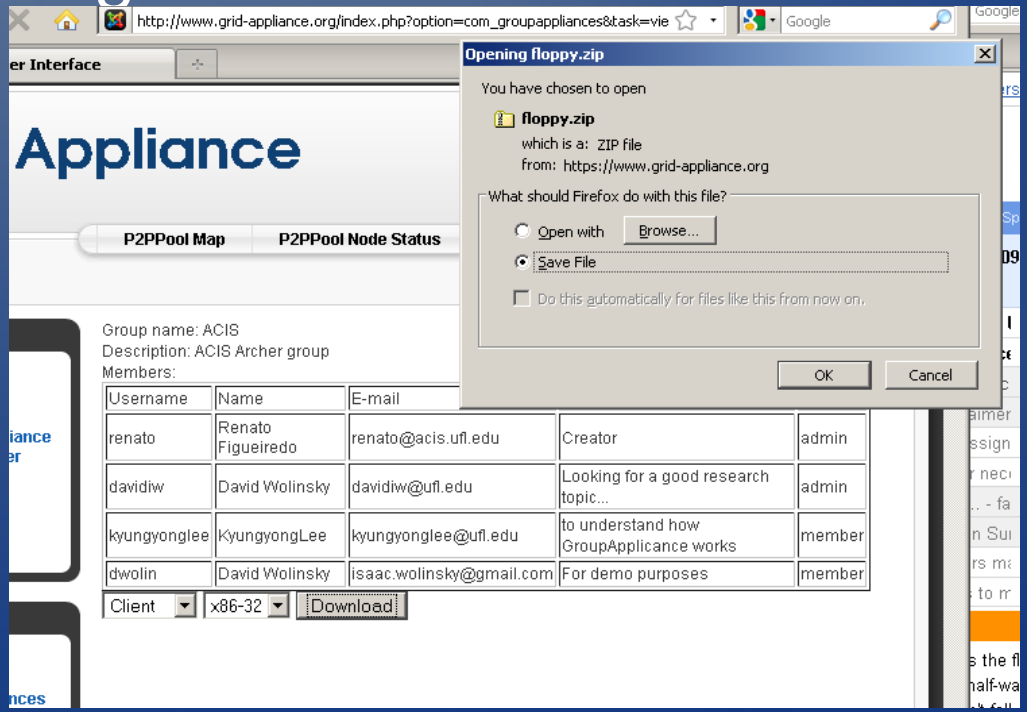
- Join or Create an Archer based GroupAppliance
- If your site has a Group and you are a member of that group, you will have priority in that group over other Archer users
- You'll receive an e-mail when you're approved from the group admin, either site IT or PI




UF UNIVERSITY of FLORIDA ACIS

# Download Appliance and config

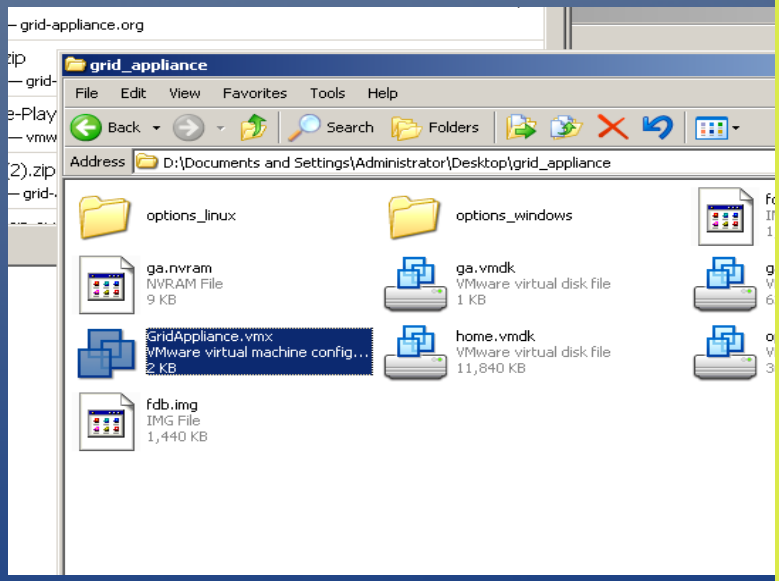
- Download the latest Appliance from www.grid-appliance.org
- Go to your GroupAppliance and download an appropriate configuration file
- Extract the appliance and configuration file
- Place the configuration file into the VM and rename if fdb.img

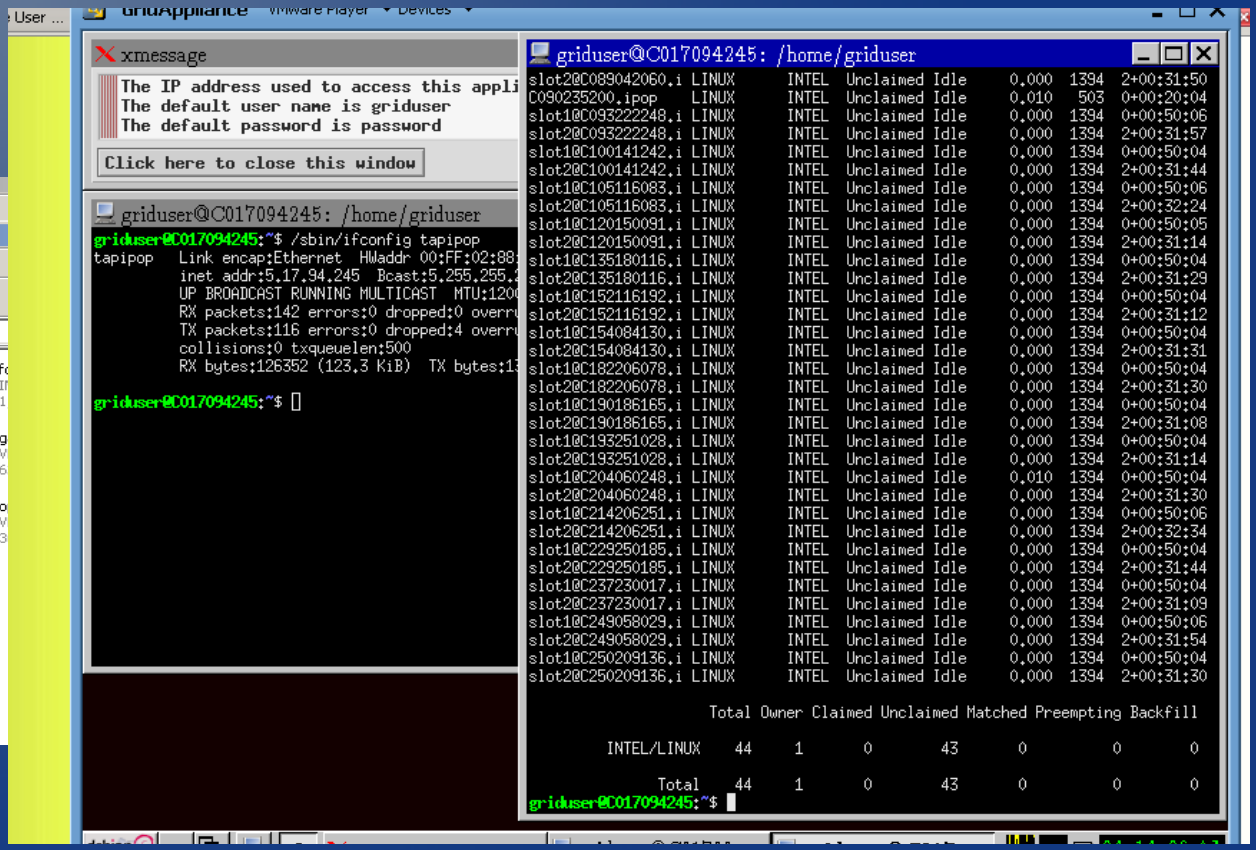



UF UNIVERSITY of FLORIDA ACIS

# Start VM and Check Connectivity

- If using Vmware Player, double click the GridAppliance[.vmx]
- Once booted, type `condor_status` at a console
- Or verify that the “tapipop” Ethernet interface has an IP Address

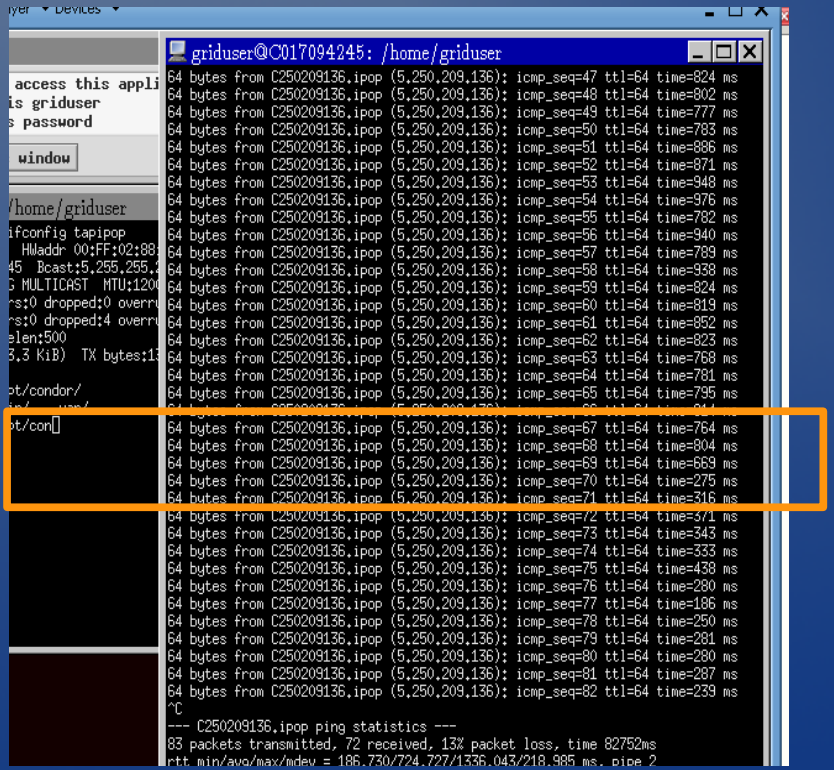



UF UNIVERSITY of FLORIDA ACIS

# Direct Connectivity

- Autonomically attempt to form direct connections with those you frequent
- Similar to Hamachi low overhead connectivity
- Done via cell network… so not necessarily a huge performance boost, though sometimes indirect links can have latency on the order of 5s

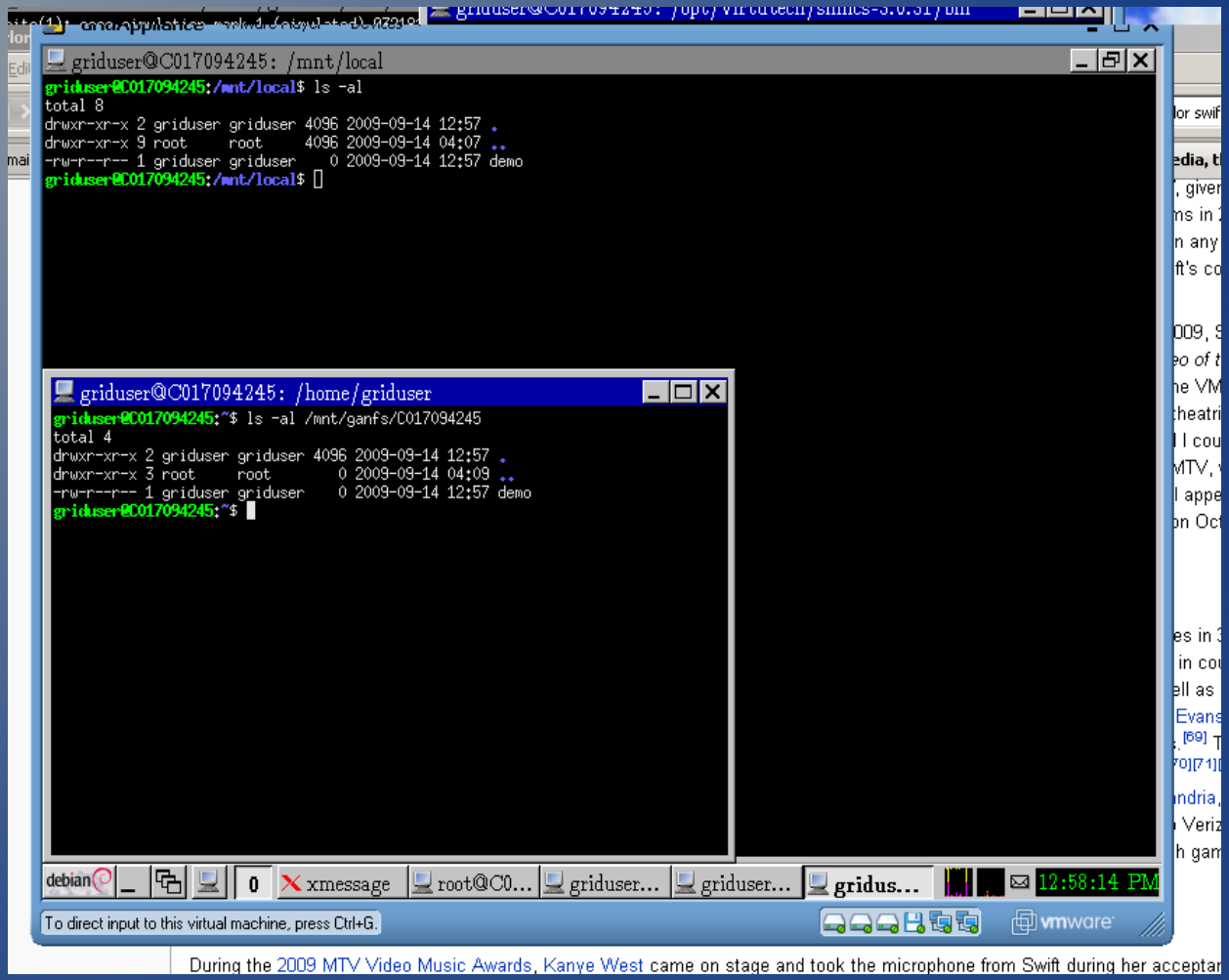



UF UNIVERSITY of FLORIDA ACIS

# NFS Data Shares

- Appliance has folder automatically exported over NFS (read-only) to the pool
  - E.g. Condor binaries, read-only input files
- Simple to the user
  - Copy files over to /mnt/local
  - Files are automatically mounted remotely, on demand, via auto-fs
  - /mnt/ganfs/Cxxxyyyzzz (user's appliance name)
- Useful for sharing pre-configured applications among users; large, sparse data sets

# NFS Data Shares in Action

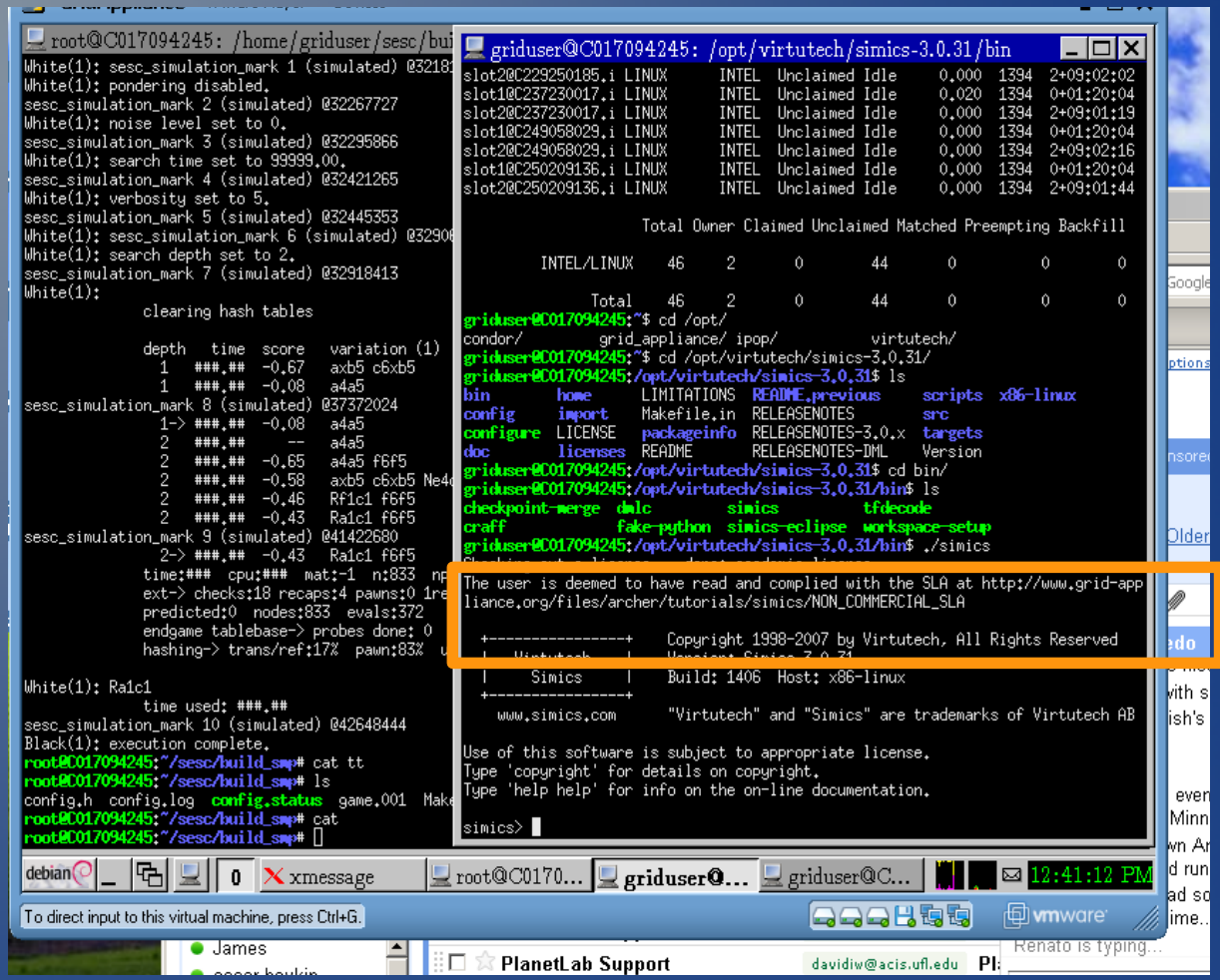



UF UNIVERSITY of FLORIDA ACIS

# Using Simics

- Download the Simics module from www.grid-appliance.org
- Extract the file to opt.vmdk in your Grid Appliance folder
- Start the Grid Appliance

# Simics in Action

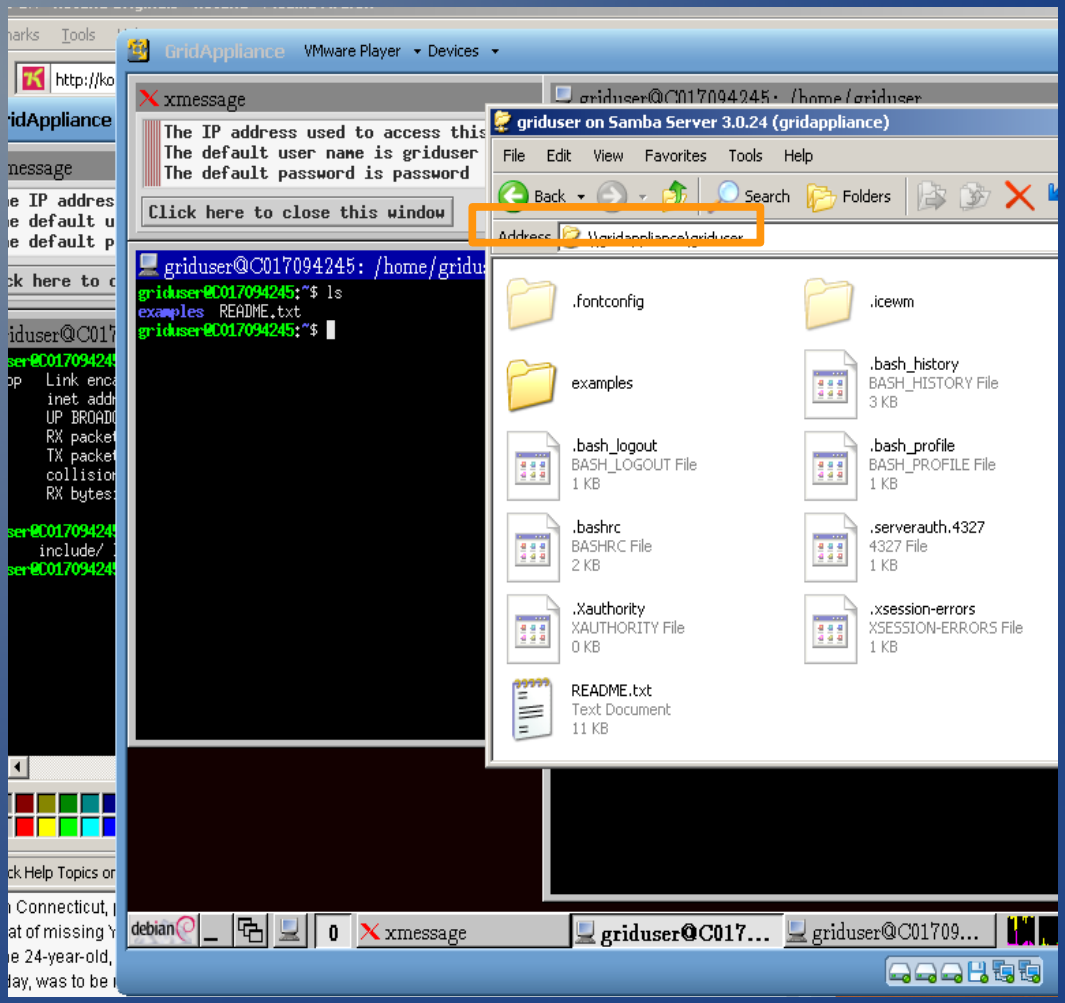



UF UNIVERSITY of FLORIDA ACIS

# Accessing Your Data

- Samba for Linux, Windows, and Mac direct file system access
- SSH and SFTP to perform actions and transfer files via the command-line
- Sharing it over the wide-area via NFS and GroupVPN

# Samba in Action




UF UNIVERSITY of FLORIDA ACIS

# SESC Live Demo

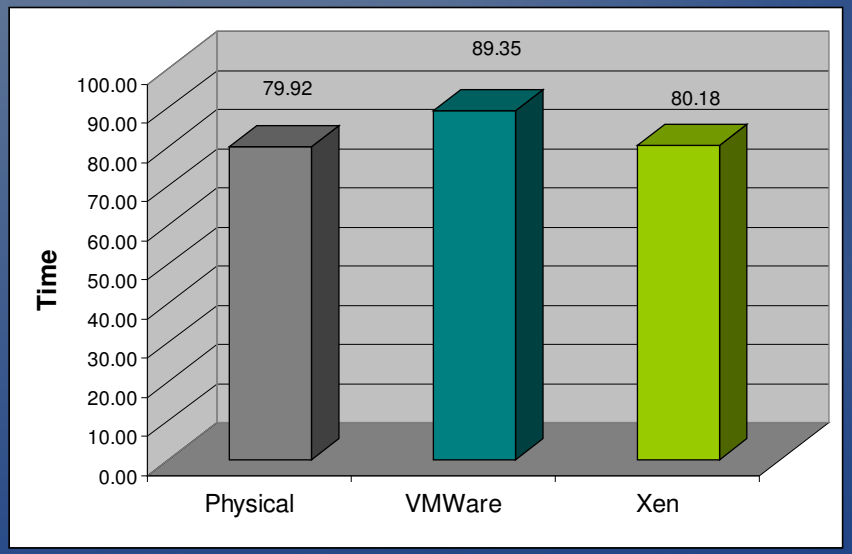
- SESC is a cycle accurate architectural simulator
- Supports single processors, CMPs, PIMs, and thread level speculation
- Most options are compile-time for performance purposes
- See: http://www.grid-appliance.org/wiki/index.php/Archer:SESC for detailed tutorial of SESC

# Outline

- Archer Design Overview
- The Components
  - Condor
  - Virtual Machines
  - Virtual Networking
- The User Experience
  - Joining and Using
  - Learning and Contributing
- Wrap Up

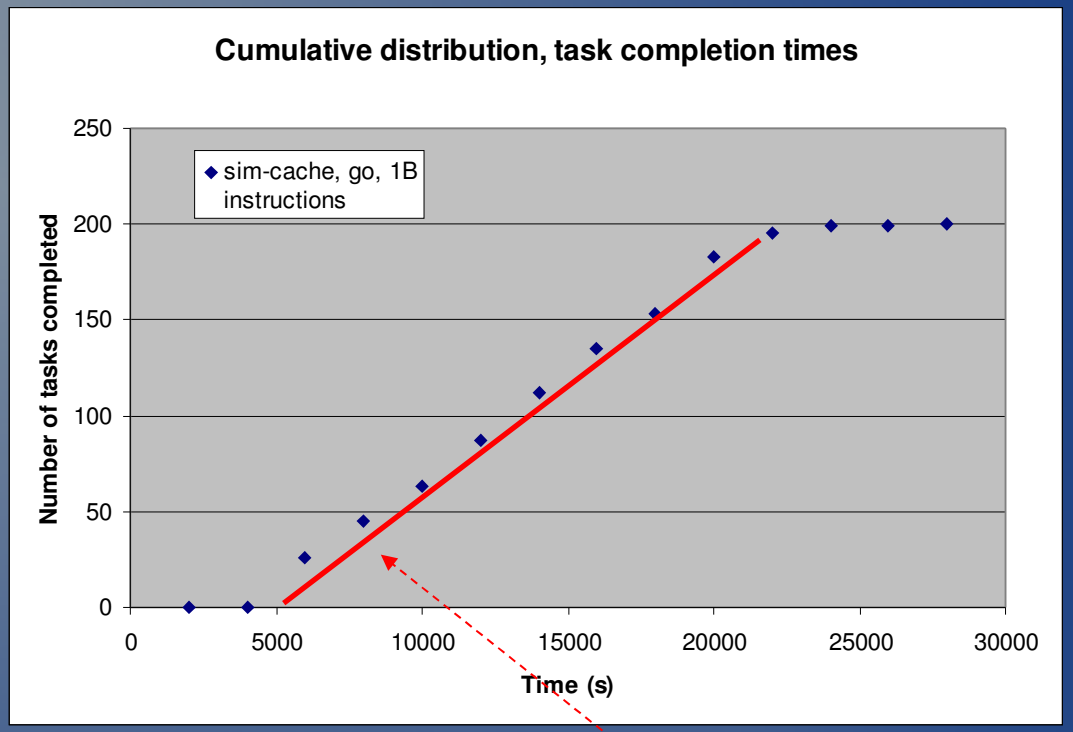
# System Overheads

- User-level C# IPOP implementation (UDP):
  - Link bandwidth: ~55Mbit/s (encrypted)
  - Latency overhead: ~0.5ms
- Connection times:
  - ~5-10s to join overlay and obtain DHCP address
  - ~10s to create NAT-traversed connections




SimpleScalar 3.0
(cycle-accurate CPU simulator)

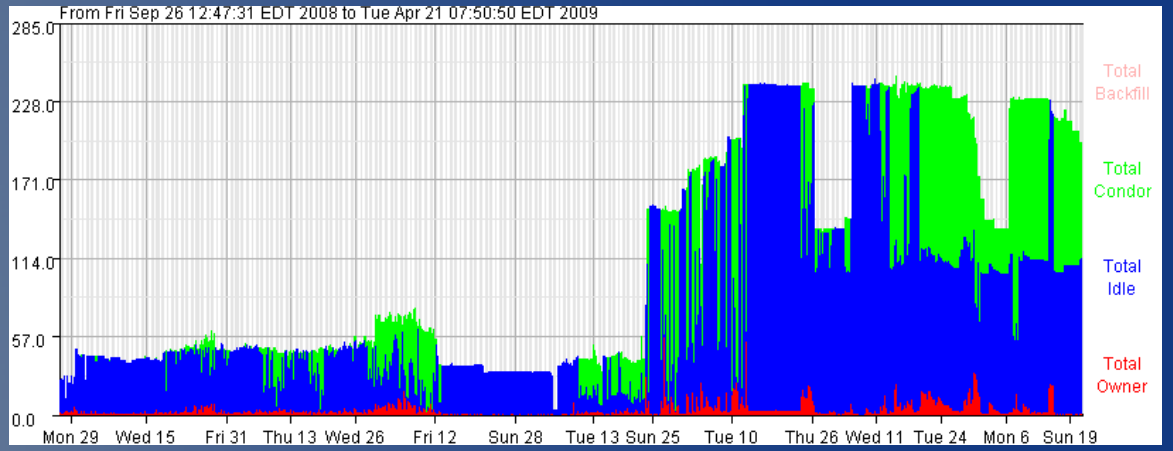
# Job Execution Rate




- 200 sim-cache simulations
  - Submitted from a laptop, behind ISP
  - Executed on 56-node Archer prototype
  - In steady state: completing jobs at rate of one job every 90s
    - Compared to one job per 42min of a single job on a single resource.

# What Else?

- Lot's of cycles to use!
- Tutorials for Simics w/ FeS2 and Gems, SESC, SimpleScalar
- Up and coming:
  - Distributed File System for reduced overhead when performing repetitions of the same task
  - Decoupled appliance setup from a VM – have the Appliance run on bare metal hardware



# Questions

- More Informatin:
  - Grid Appliance, Archer – grid-appliance.org
  - IPOP – ipop-project.org
- Acknowledgements
  - ACIS P2P Group
  - Condor Group – Ben Burnett and Alain Roy
  - Effort sponsored by the NSF under grants OCI-0721867 and CNS-0751112.
  - Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

UF UNIVERSITY of FLORIDA ACIS

# User Feedback

The best thing we like about Archer is its computing power and easy accessibility and usability. We can easily access 50 or more Intel Xeon cores remotely for our computing needs to accomplish something we cannot do before. For example, in our recent data prefetching and cache management studies, we used more than 10000 simulation hours on Archer.

Besides the huge computational resources, Archer is unbelievably easy to use. All the needed software is packaged in a virtual machine, including the IPOP, which provides communication among the grid. It is all transparent to users. Users just need to download the Grid Appliance package, and is ready to go. Archer even provides a couple of popular CPU simulators by default, like Simics, SimpleScalar and PTLsim, etc. Our group uses Simics frequently and glad to see it is available on Archer. Besides, Archer employs Condor to manage all the tasks and resources, which makes it easy to deploy/monitor the tasks and need not worry about the resources. With the newly added feature of NFS interface, we can do more in a customized way. It allows mounting a local virtual disk to the grid, and sharing user-specific files, i.e., large Simics checkpoints.  All the grid nodes can access the files shared in the NFS file system. This feature helps build our own simulation environments efficiently.

- Jih-Kwon Peir