

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

David Wolinsky  
ACIS P2P Group  
University of Florida

# 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

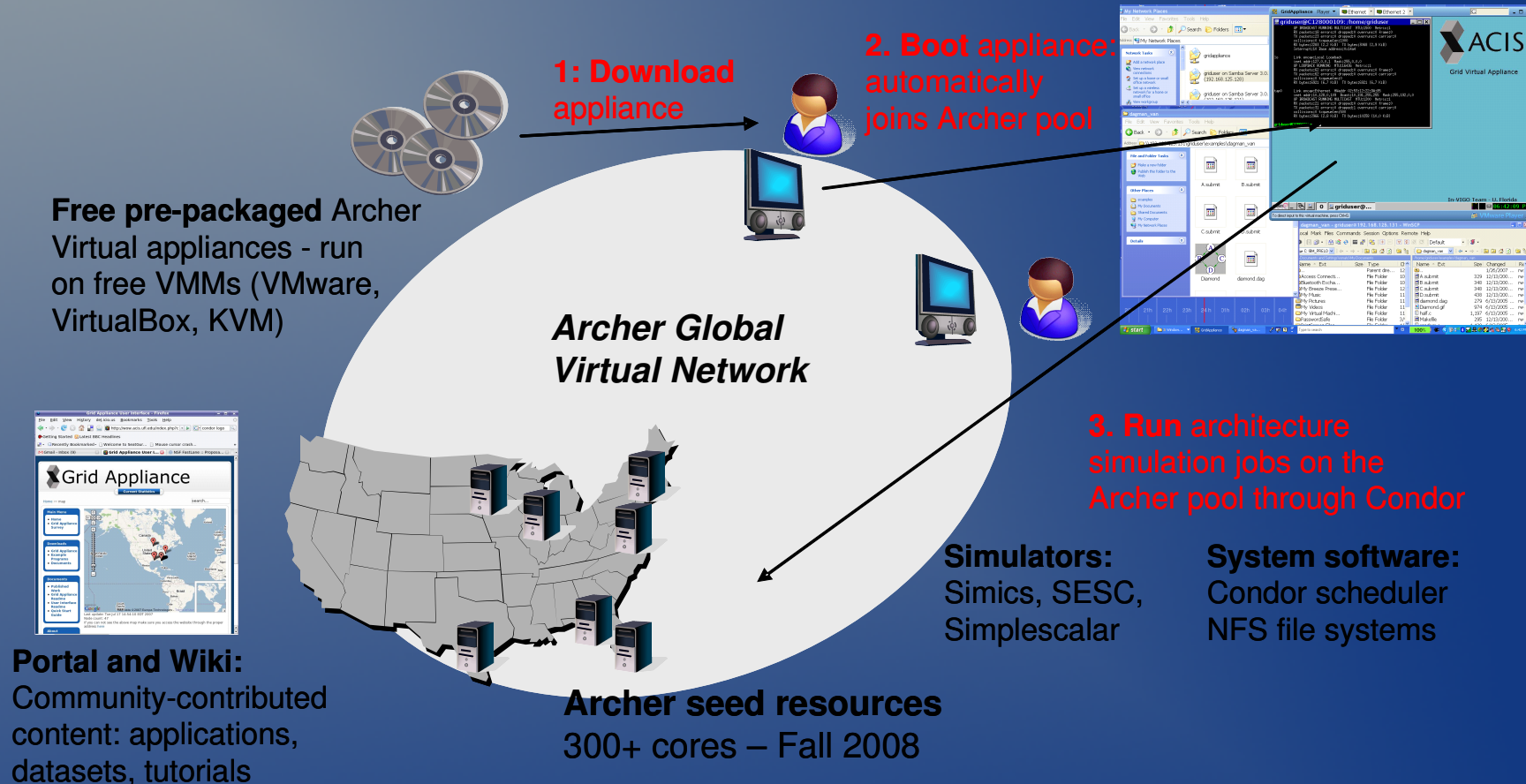
# Outline

- 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





# 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

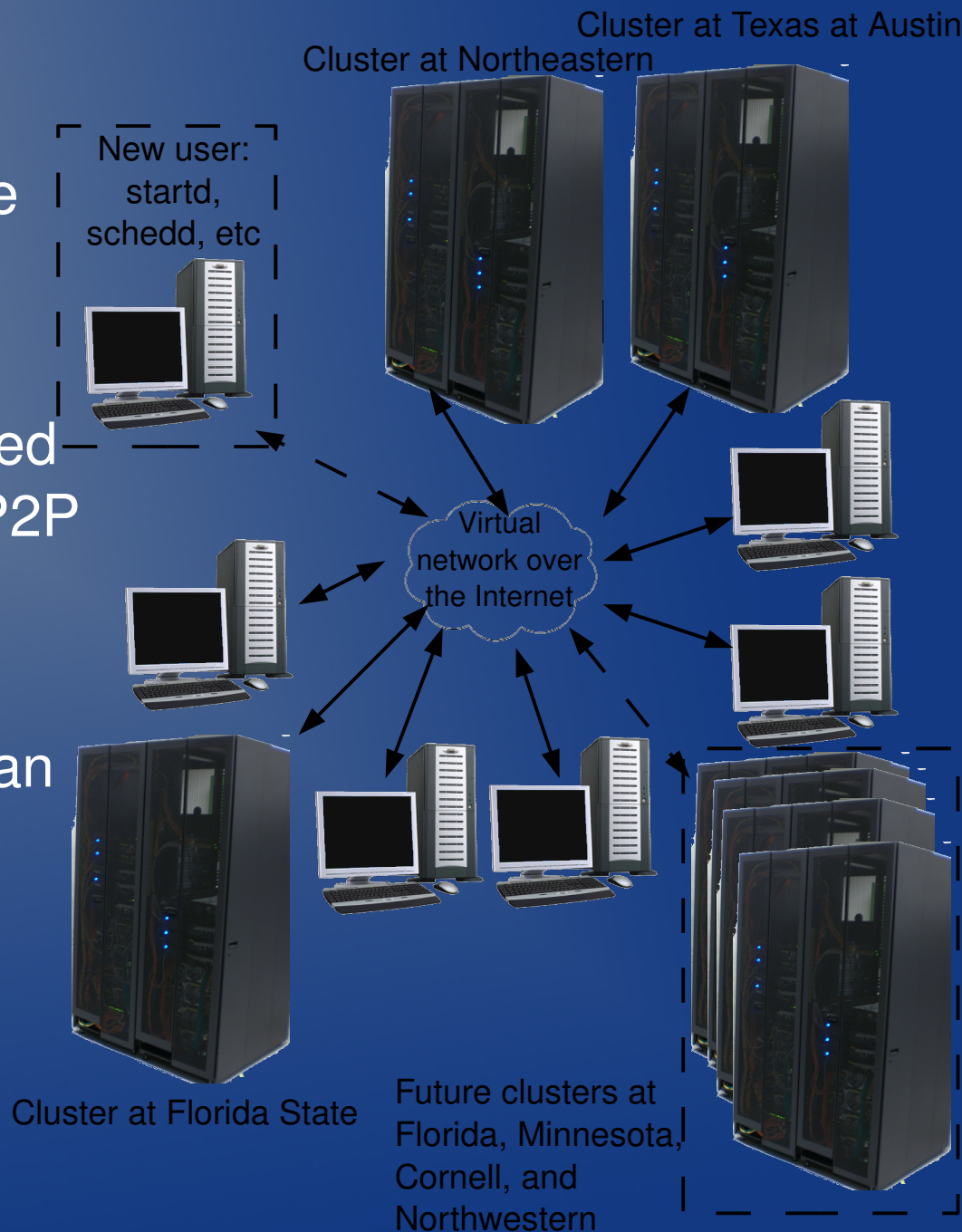
# The Components

- 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



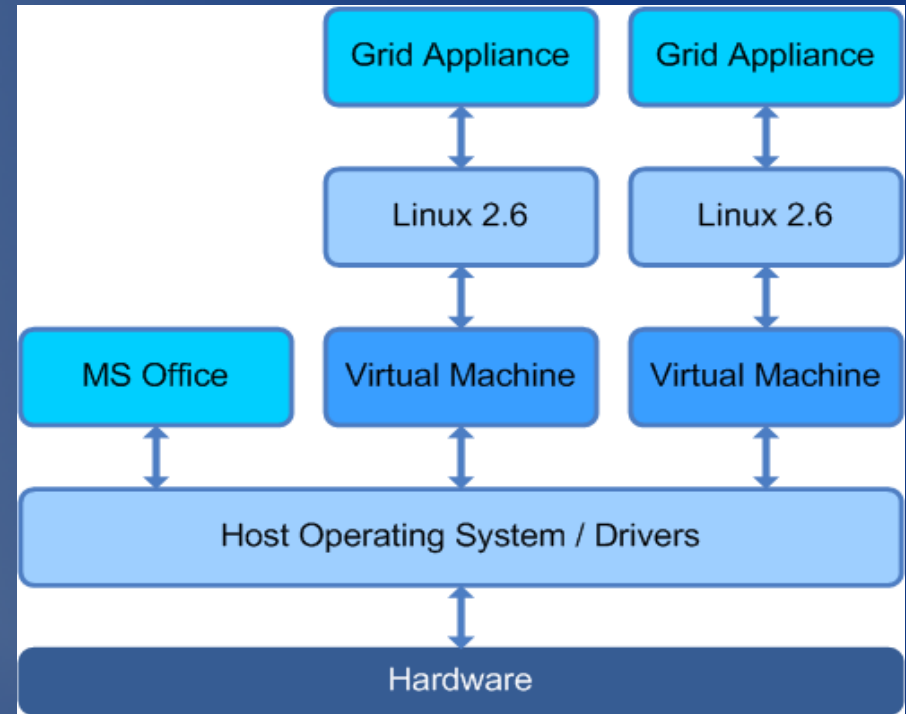
# Condor

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



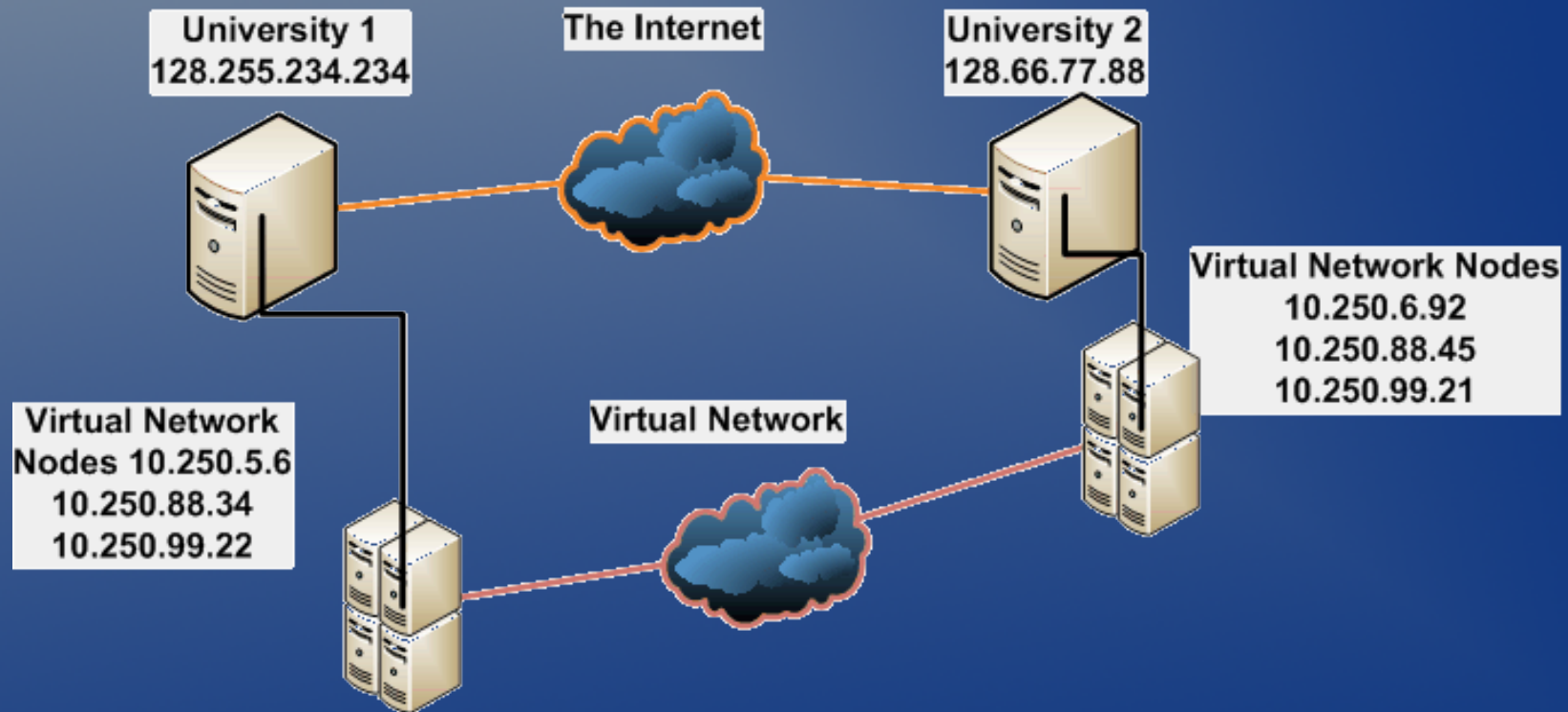


# Configurable Components

- 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





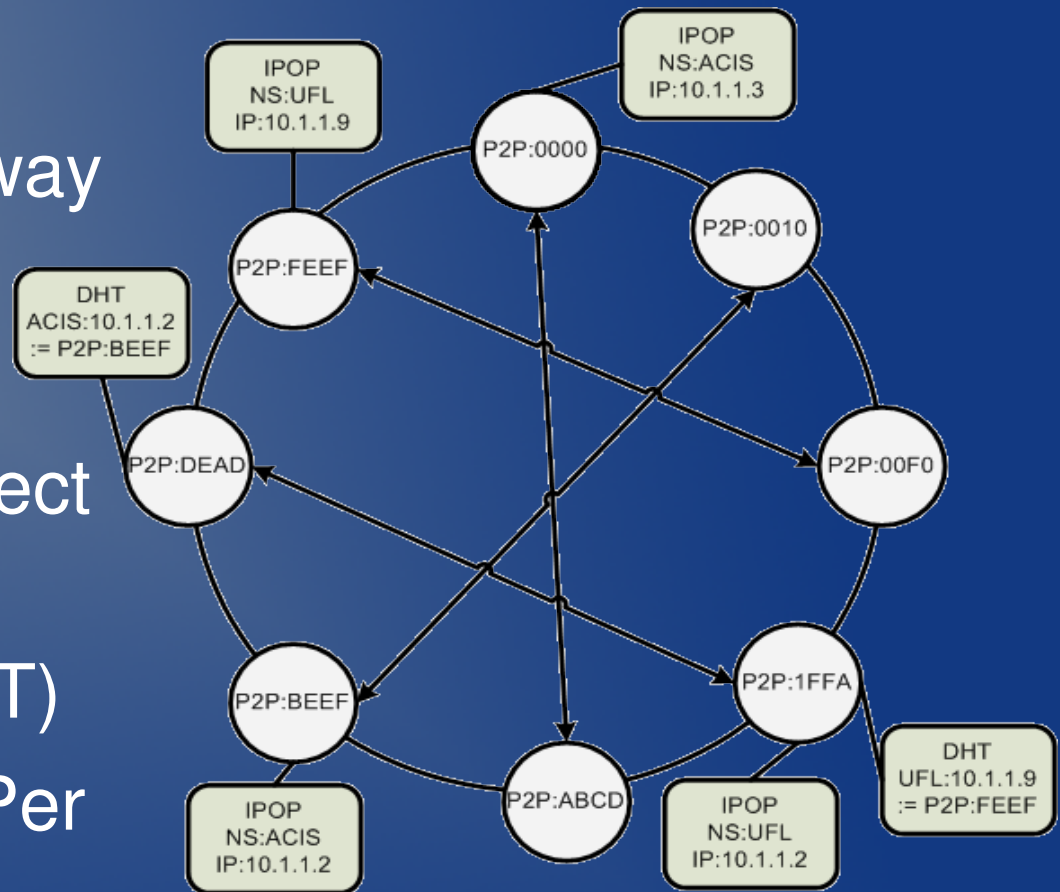
# VN Interfaces

- 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



# 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:
  - Bootstrapping
  - 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

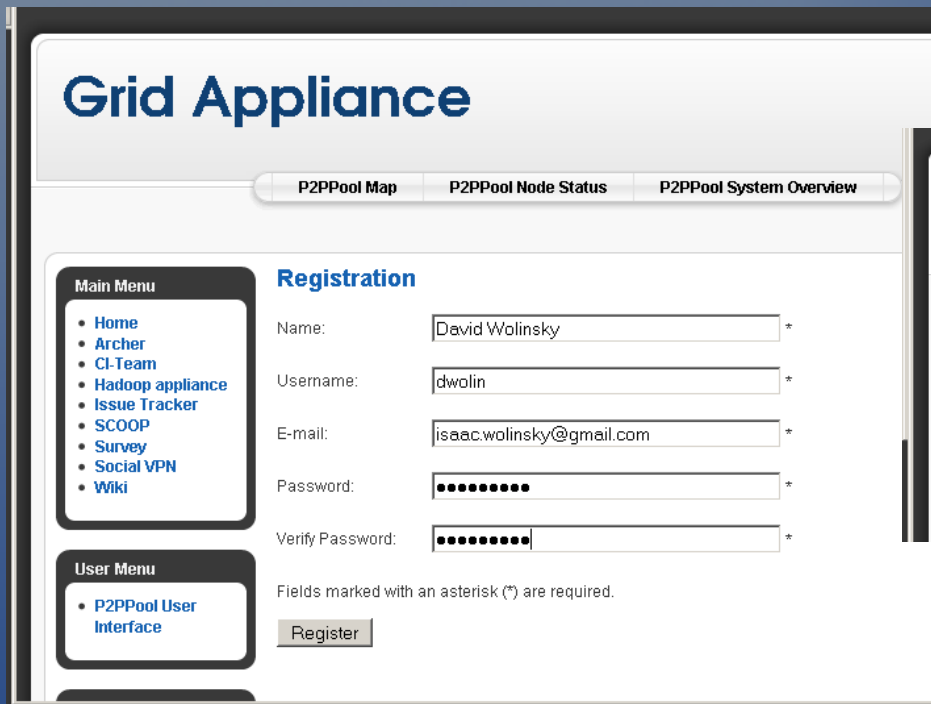
# The User Experience

- Join Archer ([www.grid-appliance.org](http://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](http://www.grid-appliance.org)
- Check your e-mail
- Click activation link



The screenshot shows the 'Grid Appliance' website with the 'Registration' form. The form includes fields for Name, Username, E-mail, Password, and Verify Password, each marked with an asterisk to indicate it is required. A 'Register' button is at the bottom of the form. The left sidebar contains a 'Main Menu' with links to Home, Archer, CI-Team, Hadoop appliance, Issue Tracker, SCOOP, Survey, Social VPN, and Wiki, and a 'User Menu' with a link to the P2PPool User Interface. The top navigation bar includes links to P2PPool Map, P2PPool Node Status, and P2PPool System Overview.

**Grid Appliance**

P2PPool Map P2PPool Node Status P2PPool System Overview

**Registration**

Name:  \*

Username:  \*

E-mail:  \*

Password:  \*

Verify Password:  \*

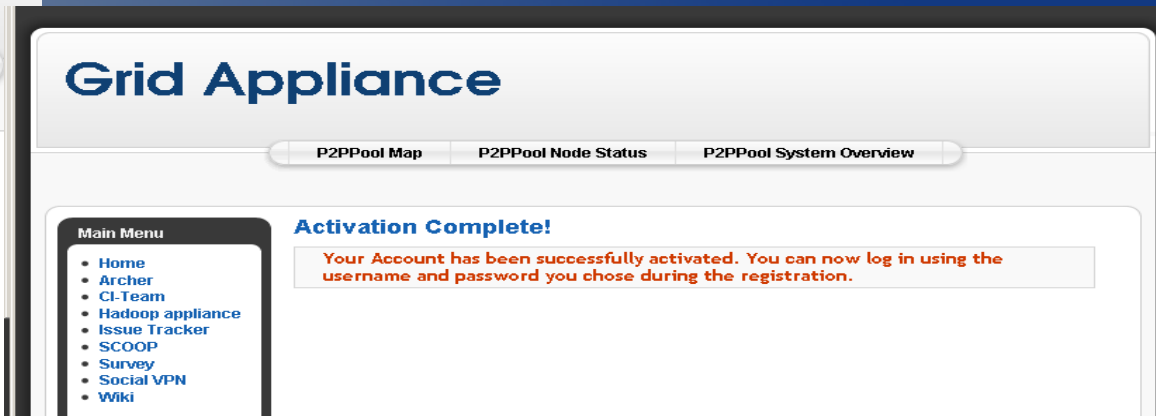
Fields marked with an asterisk (\*) are required.

**Main Menu**

- Home
- Archer
- CI-Team
- Hadoop appliance
- Issue Tracker
- SCOOP
- Survey
- Social VPN
- Wiki

**User Menu**

- P2PPool User Interface



The screenshot shows the 'Grid Appliance' website with the 'Activation Complete!' message. The message states: 'Your Account has been successfully activated. You can now log in using the username and password you chose during the registration.' The left sidebar and top navigation bar are identical to the registration page.

**Grid Appliance**

P2PPool Map P2PPool Node Status P2PPool System Overview

**Activation Complete!**

Your Account has been successfully activated. You can now log in using the username and password you chose during the registration.

**Main Menu**

- Home
- Archer
- CI-Team
- Hadoop appliance
- Issue Tracker
- SCOOP
- Survey
- Social VPN
- Wiki

# 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

**Main Menu**

- Home
- Archer
- CI-Team
- Hadoop appliance
- Issue Tracker
- SCOOP
- Survey
- Social VPN
- Wiki

**User Menu**

- Your Details
- GroupVPN
- GroupAppliances
- P2PPool User Interface
- Archer User Panel
- Administration
- Logout

Group -	Description	State
<a href="#">Archer</a>	Architecture Research Group	
<a href="#">Clemson_ipop_pool</a>	P2P at Clemson	
<a href="#">Clemson_pool_2</a>	A Second pool connected to Planetlab	
<a href="#">GroupVPN trial</a>	GroupVPN trial	
<a href="#">GroupVPN_1</a>	Try group vpn	
<a href="#">IPOP Test Group</a>	This is a public test group for the purpose of testing IPOP and its GroupVPN	
<a href="#">Jimnet</a>	Test network for Jimmy	
<a href="#">PierreTest</a>		
<a href="#">Public Default Group</a>	A GroupVPN for testing and global usage	
<a href="#">Yonggangtest</a>	for test	
<a href="#">rftest</a>	Testing the creation of a new groupvpn.	

Create a new group

## Introduction and Terminology

Throughout this document, the following terms are used:

- \* "Archer appliance": a virtual machine which hosts the execution of jobs submitted from registered Archer users;
- \* "Archer Wiki": the Web site which hosts content provided by registered Archer users, currently hosted at <http://archer-project.org>;
- \* "Archer Global" (or "Archer"): the distributed system consisting of the Archer Wiki and the collection of Archer appliances deployed by its registered users.

Archer is a community distributed system for the simulation-based computer architecture research and education. It consists of virtual machine appliances deployed at various institutions. Archer administration is currently managed the University of Florida. The use of the Archer Global distributed system is restricted to registered users. Access to Archer Global by its registered users is achieved through deployment of Archer appliances on their own computers. Archer appliances deployed by Archer registered users host the execution of jobs from registered Archer users when idle, enabling the computational

Organization:

Department:

Country:

Phone number:

Ethnicity:

Reason for Account:

# 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

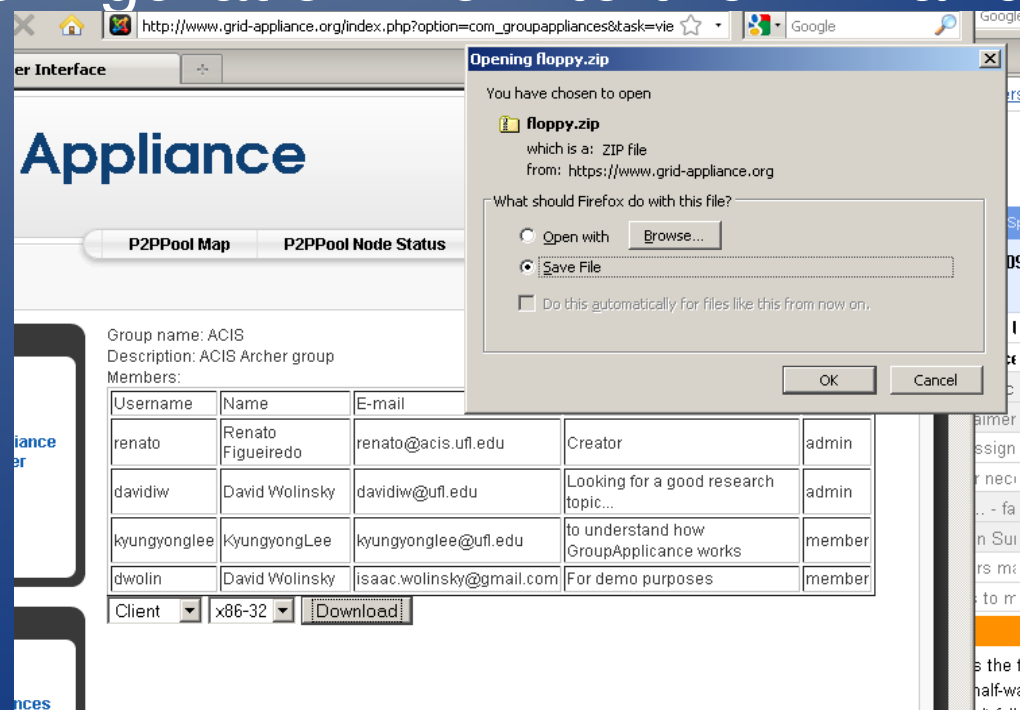
The screenshot shows the 'P2PPool System Overview' page. On the left is a 'Main Menu' with links: Home, Archer, CI-Team, Hadoop appliance, Issue Tracker, SCOOP, Survey, Social VPN, and Wiki. Below it is a 'User Menu' with links: Your Details, GroupVPN, GroupAppliances (highlighted with an orange box), P2PPool User Interface, and Archer User Panel. The main content area features a table of groups:

Group	Description	GroupVPN	Reason for joining	State	Action
ACIS	ACIS Archer group	Archer	For demo purposes		<input type="button" value="Join"/>
Archer Default	Provides access to Archer Global resources.	Archer			<input type="button" value="Join"/>
Public Default Group	Group Distributed with the Grid Appliance	Public Default Group			<input type="button" value="Join"/>

Below the table is a 'Create a new group:' section with a 'Group description' text area, a 'group name' input field, and a dropdown menu currently set to 'rftest'. A 'Create group' button is at the bottom of this section. At the very bottom of the page, there is a list of users: Renato Figueiredo (with a link to 'Re: Preparation') and P. Oscar Boykin (with a link to 'IP2PI Re: Bugs, bugs, bugs').

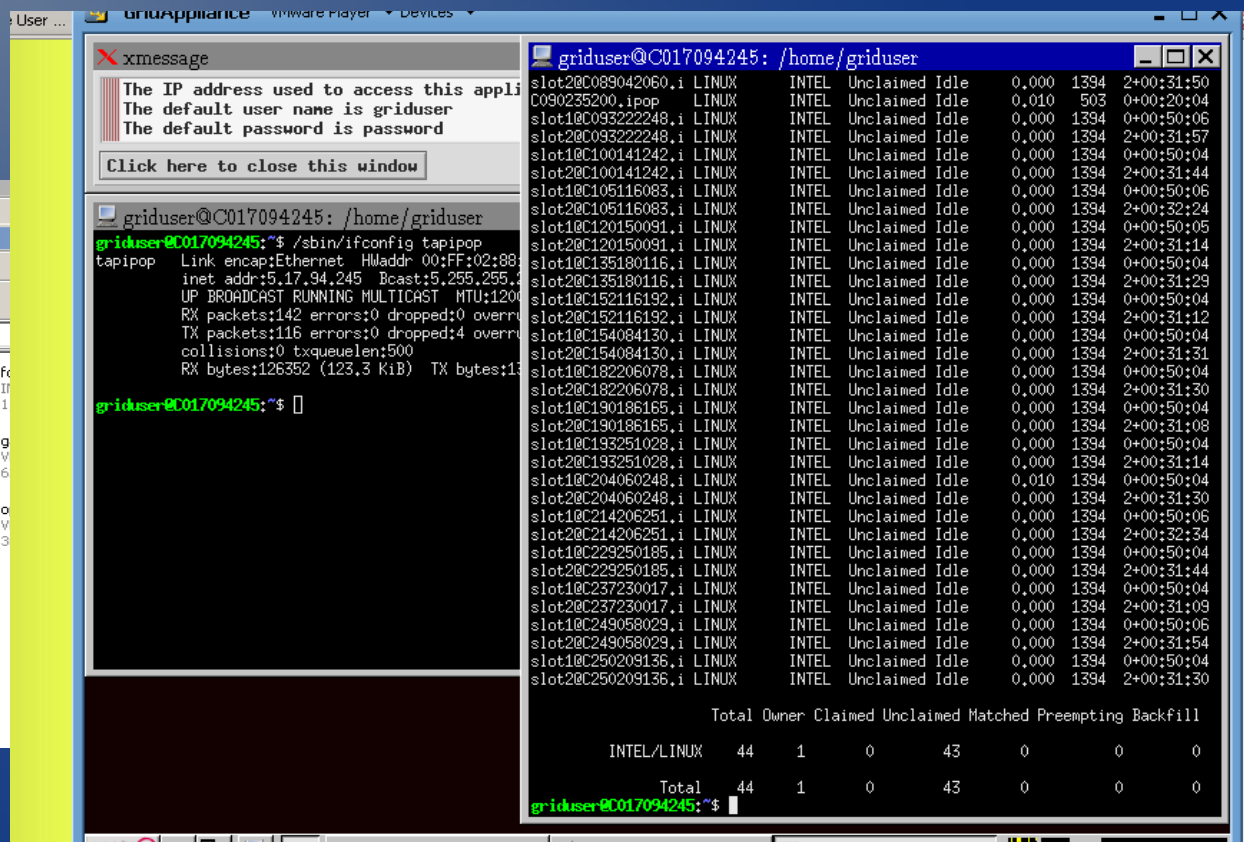
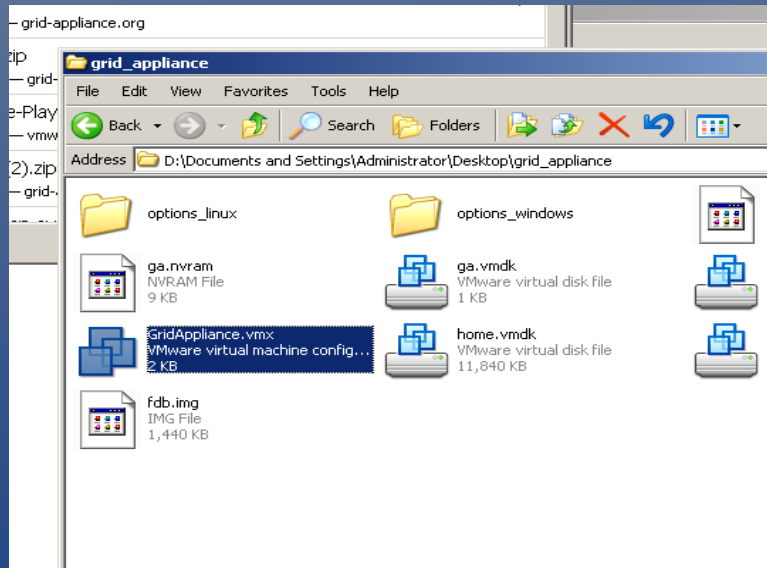
# Download Appliance and config

- Download the latest Appliance from [www.grid-appliance.org](http://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



# 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



# 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

```
griduser@C017094245: /home/griduser
64 bytes from C250209136.ipop (5,250,209,136): icmp_seq=47 ttl=64 time=824 ms
64 bytes from C250209136.ipop (5,250,209,136): icmp_seq=48 ttl=64 time=802 ms
64 bytes from C250209136.ipop (5,250,209,136): icmp_seq=49 ttl=64 time=777 ms
64 bytes from C250209136.ipop (5,250,209,136): icmp_seq=50 ttl=64 time=783 ms
64 bytes from C250209136.ipop (5,250,209,136): icmp_seq=51 ttl=64 time=886 ms
64 bytes from C250209136.ipop (5,250,209,136): icmp_seq=52 ttl=64 time=871 ms
64 bytes from C250209136.ipop (5,250,209,136): icmp_seq=53 ttl=64 time=948 ms
64 bytes from C250209136.ipop (5,250,209,136): icmp_seq=54 ttl=64 time=976 ms
64 bytes from C250209136.ipop (5,250,209,136): icmp_seq=55 ttl=64 time=782 ms
64 bytes from C250209136.ipop (5,250,209,136): icmp_seq=56 ttl=64 time=940 ms
64 bytes from C250209136.ipop (5,250,209,136): icmp_seq=57 ttl=64 time=789 ms
64 bytes from C250209136.ipop (5,250,209,136): icmp_seq=58 ttl=64 time=938 ms
64 bytes from C250209136.ipop (5,250,209,136): icmp_seq=59 ttl=64 time=824 ms
64 bytes from C250209136.ipop (5,250,209,136): icmp_seq=60 ttl=64 time=819 ms
64 bytes from C250209136.ipop (5,250,209,136): icmp_seq=61 ttl=64 time=862 ms
64 bytes from C250209136.ipop (5,250,209,136): icmp_seq=62 ttl=64 time=823 ms
64 bytes from C250209136.ipop (5,250,209,136): icmp_seq=63 ttl=64 time=768 ms
64 bytes from C250209136.ipop (5,250,209,136): icmp_seq=64 ttl=64 time=781 ms
64 bytes from C250209136.ipop (5,250,209,136): icmp_seq=65 ttl=64 time=795 ms
64 bytes from C250209136.ipop (5,250,209,136): icmp_seq=66 ttl=64 time=811 ms
64 bytes from C250209136.ipop (5,250,209,136): icmp_seq=67 ttl=64 time=764 ms
64 bytes from C250209136.ipop (5,250,209,136): icmp_seq=68 ttl=64 time=804 ms
64 bytes from C250209136.ipop (5,250,209,136): icmp_seq=69 ttl=64 time=669 ms
64 bytes from C250209136.ipop (5,250,209,136): icmp_seq=70 ttl=64 time=275 ms
64 bytes from C250209136.ipop (5,250,209,136): icmp_seq=71 ttl=64 time=316 ms
64 bytes from C250209136.ipop (5,250,209,136): icmp_seq=72 ttl=64 time=371 ms
64 bytes from C250209136.ipop (5,250,209,136): icmp_seq=73 ttl=64 time=343 ms
64 bytes from C250209136.ipop (5,250,209,136): icmp_seq=74 ttl=64 time=333 ms
64 bytes from C250209136.ipop (5,250,209,136): icmp_seq=75 ttl=64 time=438 ms
64 bytes from C250209136.ipop (5,250,209,136): icmp_seq=76 ttl=64 time=280 ms
64 bytes from C250209136.ipop (5,250,209,136): icmp_seq=77 ttl=64 time=186 ms
64 bytes from C250209136.ipop (5,250,209,136): icmp_seq=78 ttl=64 time=250 ms
64 bytes from C250209136.ipop (5,250,209,136): icmp_seq=79 ttl=64 time=281 ms
64 bytes from C250209136.ipop (5,250,209,136): icmp_seq=80 ttl=64 time=280 ms
64 bytes from C250209136.ipop (5,250,209,136): icmp_seq=81 ttl=64 time=287 ms
64 bytes from C250209136.ipop (5,250,209,136): icmp_seq=82 ttl=64 time=239 ms
--- C250209136.ipop ping statistics ---
83 packets transmitted, 72 received, 13% packet loss, time 82752ms
rtt min/avg/max/mdev = 186.730/724.727/1336.043/218.985 ms, pipe 2
```

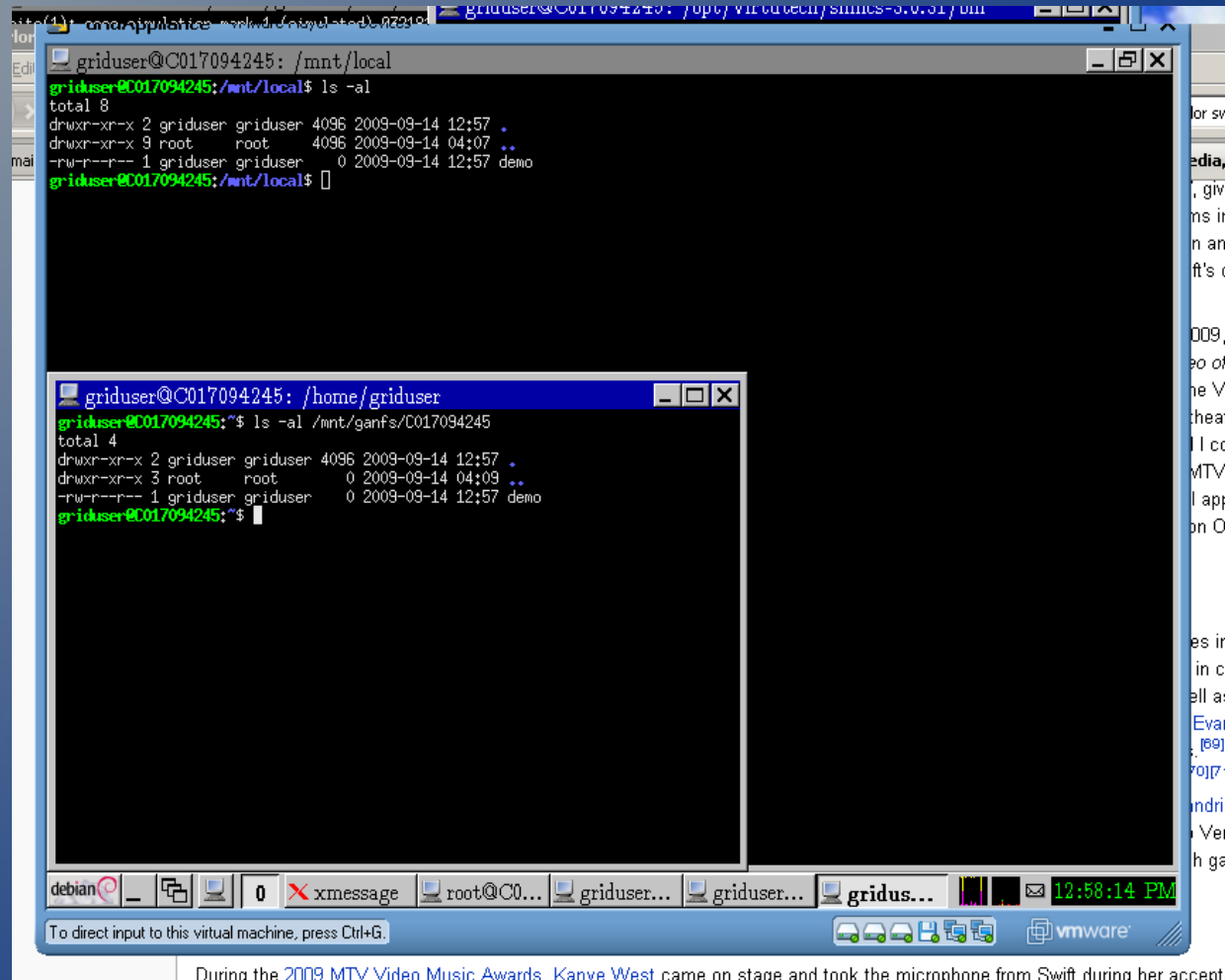


# NFS Data Shares

- 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/Cxxxxyyyzzz (user's appliance name)
- Useful for sharing pre-configured applications among users; large, sparse data sets



# NFS Data Shares in Action



```
griduser@C017094245: /mnt/local
griduser@C017094245:/mnt/local$ ls -al
total 8
drwxr-xr-x 2 griduser griduser 4096 2009-09-14 12:57 .
drwxr-xr-x 9 root      root    4096 2009-09-14 04:07 ..
-rw-r--r-- 1 griduser griduser  0 2009-09-14 12:57 demo
griduser@C017094245:/mnt/local$

griduser@C017094245: /home/griduser
griduser@C017094245:~$ ls -al /mnt/ganfs/C017094245
total 4
drwxr-xr-x 2 griduser griduser 4096 2009-09-14 12:57 .
drwxr-xr-x 3 root      root    0 2009-09-14 04:09 ..
-rw-r--r-- 1 griduser griduser  0 2009-09-14 12:57 demo
griduser@C017094245:~$
```

During the 2009 MTV Video Music Awards, Kanye West came on stage and took the microphone from Swift during her acceptance

# Using Simics

- Download the Simics module from [www.grid-appliance.org](http://www.grid-appliance.org)
- Extract the file to opt.vmdk in your Grid Appliance folder
- Start the Grid Appliance

# Simics in Action

```
root@C017094245: /home/griduser/sesc/build
White(1): sesc_simulation_mark 1 (simulated) @3218
White(1): pondering disabled.
sesc_simulation_mark 2 (simulated) @32267727
White(1): noise level set to 0.
sesc_simulation_mark 3 (simulated) @32295866
White(1): search time set to 99999.00.
sesc_simulation_mark 4 (simulated) @32421265
White(1): verbosity set to 5.
sesc_simulation_mark 5 (simulated) @32445353
White(1): sesc_simulation_mark 6 (simulated) @3290
White(1): search depth set to 2.
sesc_simulation_mark 7 (simulated) @32918413
White(1):
clearing hash tables
depth time score variation (1)
1 ###.## -0.67 axb5 c6xb5
1 ###.## -0.08 a4a5
sesc_simulation_mark 8 (simulated) @37372024
1-> ###.## -0.08 a4a5
2 ###.## -- a4a5
2 ###.## -0.65 a4a5 f6f5
2 ###.## -0.58 axb5 c6xb5 Ne4
2 ###.## -0.46 Rf1c1 f6f5
2 ###.## -0.43 Ra1c1 f6f5
sesc_simulation_mark 9 (simulated) @41422680
2-> ###.## -0.43 Ra1c1 f6f5
time:### cpu:### mat:-1 n:833 n
ext-> checks:18 recaps:4 pawns:0 1re
predicted:0 nodes:833 evals:372
endgame tablebase-> probes done: 0
hashing-> trans/ref:17% pawn:83% v
White(1): Ra1c1
time used: ###.##
sesc_simulation_mark 10 (simulated) @42648444
Black(1): execution complete.
root@C017094245: /sesc/build_sq# cat tt
root@C017094245: /sesc/build_sq# ls
config.h config.log config.status game.001 Make
root@C017094245: /sesc/build_sq# cat
root@C017094245: /sesc/build_sq#

griduser@C017094245: /opt/virtutech/simics-3.0.31/bin
slot20C229250185.i LINUX INTEL Unclaimed Idle 0.000 1394 2+09:02:02
slot10C237230017.i LINUX INTEL Unclaimed Idle 0.020 1394 0+01:20:04
slot20C237230017.i LINUX INTEL Unclaimed Idle 0.000 1394 2+09:01:19
slot10C249058029.i LINUX INTEL Unclaimed Idle 0.000 1394 0+01:20:04
slot20C249058029.i LINUX INTEL Unclaimed Idle 0.000 1394 2+09:02:16
slot10C250209136.i LINUX INTEL Unclaimed Idle 0.000 1394 0+01:20:04
slot20C250209136.i LINUX INTEL Unclaimed Idle 0.000 1394 2+09:01:44

Total Owner Claimed Unclaimed Matched Preempting Backfill
INTEL/LINUX 46 2 0 44 0 0 0
Total 46 2 0 44 0 0 0
griduser@C017094245:~$ cd /opt/
condor/ grid_appliance/ ipop/ virtutech/
griduser@C017094245:~$ cd /opt/virtutech/simics-3.0.31/
griduser@C017094245:~$ cd /opt/virtutech/simics-3.0.31$ ls
bin home LIMITATIONS README.previous scripts x86-linux
config import Makefile.in RELEASNOTES src
configure LICENSE packageinfo RELEASNOTES-3.0.x targets
doc licenses README RELEASNOTES-DML Version
griduser@C017094245:~$ cd /opt/virtutech/simics-3.0.31$ cd bin/
griduser@C017094245:~$ cd /opt/virtutech/simics-3.0.31/bin$ ls
checkpoint-merge dlc simics tfdecode
craff fake-python simics-eclipse workspace-setup
griduser@C017094245:~$ cd /opt/virtutech/simics-3.0.31/bin$ ./simics

The user is deemed to have read and complied with the SLA at http://www.grid-app
liance.org/files/archer/tutorials/simics/NON_COMMERCIAL_SLA

+-----+ Copyright 1998-2007 by Virtutech, All Rights Reserved
+-----+
+-----+ Virtutech Build: 1406 Host: x86-linux
+-----+
+-----+ www.simics.com "Virtutech" and "Simics" are trademarks of Virtutech AB
+-----+

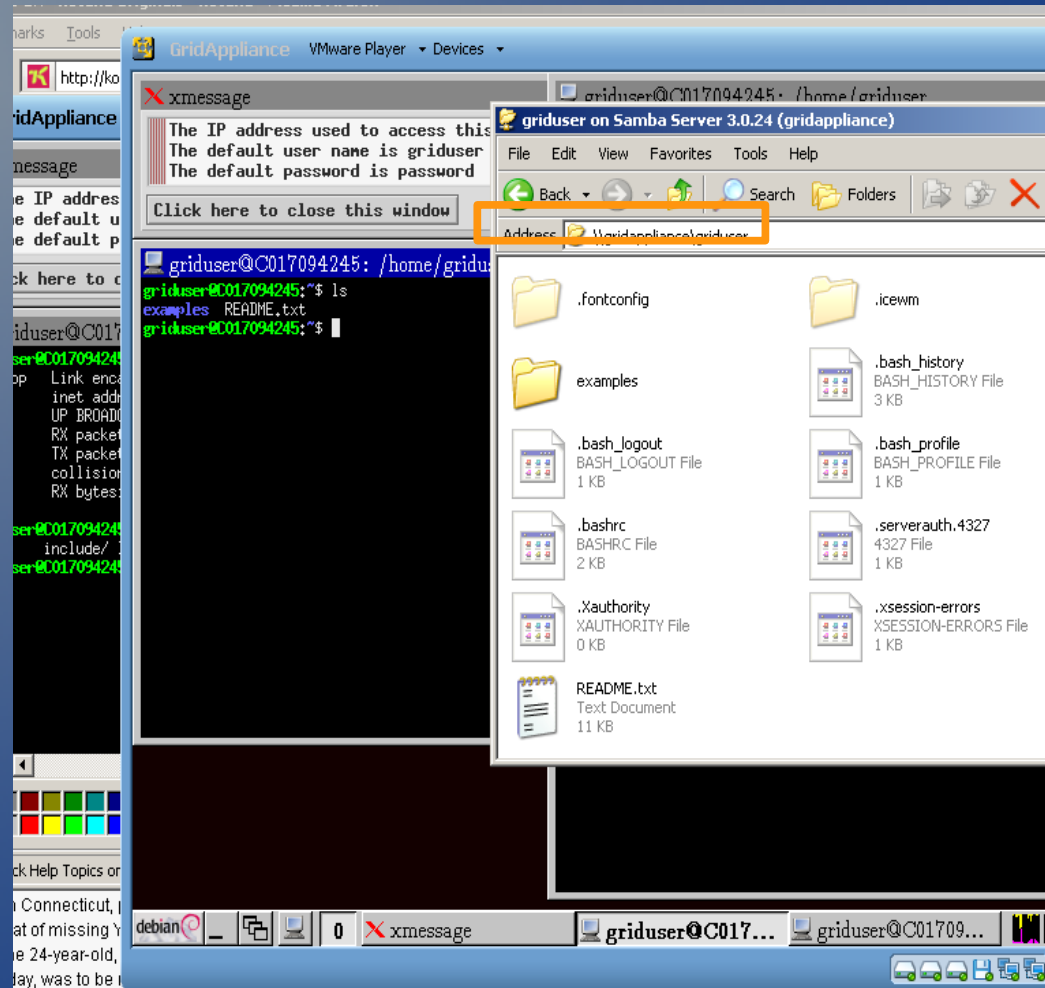
Use of this software is subject to appropriate license.
Type 'copyright' for details on copyright.
Type 'help help' for info on the on-line documentation.

simics>
```

# 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



# 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

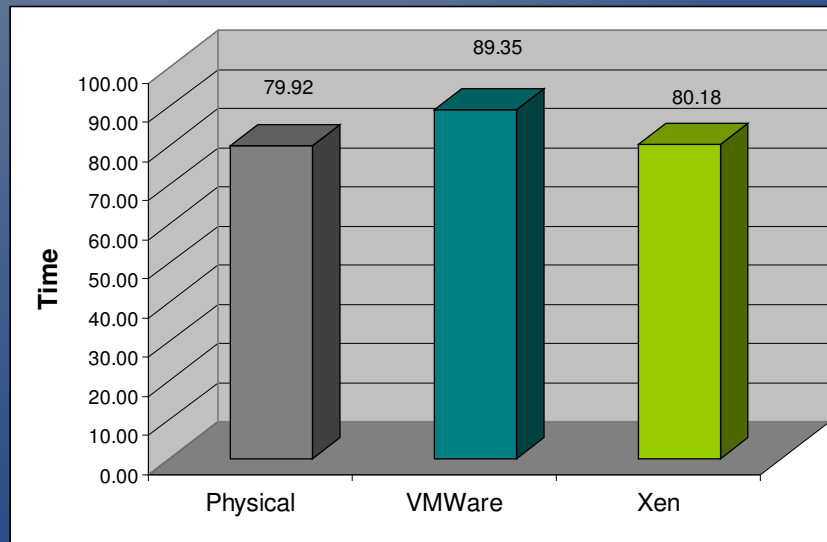
# Outline

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



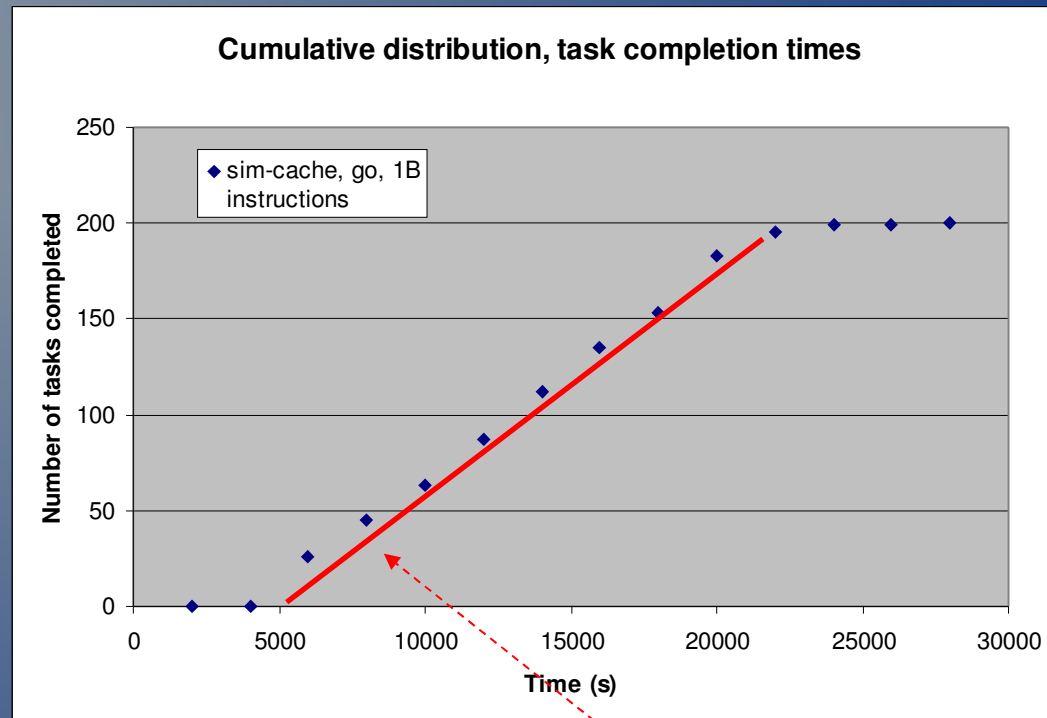
# System Overheads

- 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

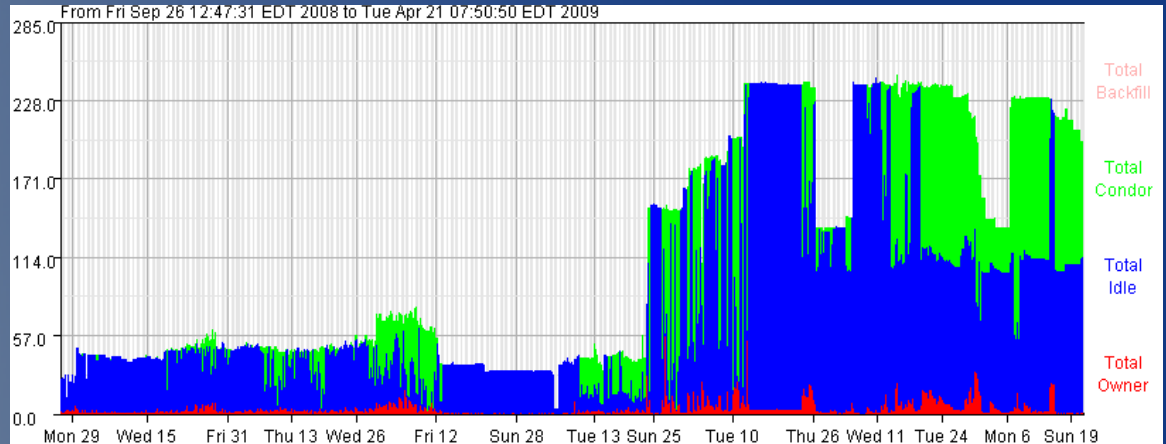


- 200 sim-cache simulations
  - Submitted from a laptop, behind ISP
  - Executed on 56-node Archer prototype
  - In steady state: completing jobs at rate of one job every 90s
    - Compared to one job per 42min of a single job on a single resource.

# 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 Information:
  - Grid Appliance, Archer – [grid-appliance.org](http://grid-appliance.org)
  - IPOP – [ipop-project.org](http://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.

# 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