
The Eucalyptus Open-source Cloud Computing System

Daniel Nurmi


Rich Wolski, Chris Grzegorzczuk, Graziano Obertelli, Sunil Soman, Lamia Youseff, Dmitrii Zagorodnov

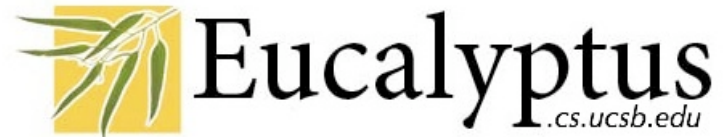
University of California Santa Barbara

Cloud Computing

- **Multiple abstractions and interfaces**
 - Infrastructure (IaaS) - EC2, GoGrid, Joyent, Nimbus
 - Platform (PaaS) - AppEngine
 - Software (SaaS) - Salesforce.com
- **Commonalities**
 - Users provision **dynamic service/resource pools on-demand**
 - **Dynamic** - resource pools can grow and shrink while application is running
 - **On-demand** - no waiting (batch queues)
 - Resource location is irrelevant
 - Interfaces are simple
- **Most cloud computing systems today are commercial**
 - Interfaces and capabilities are clearly defined
 - Implementation is, by design, invisible/unknown

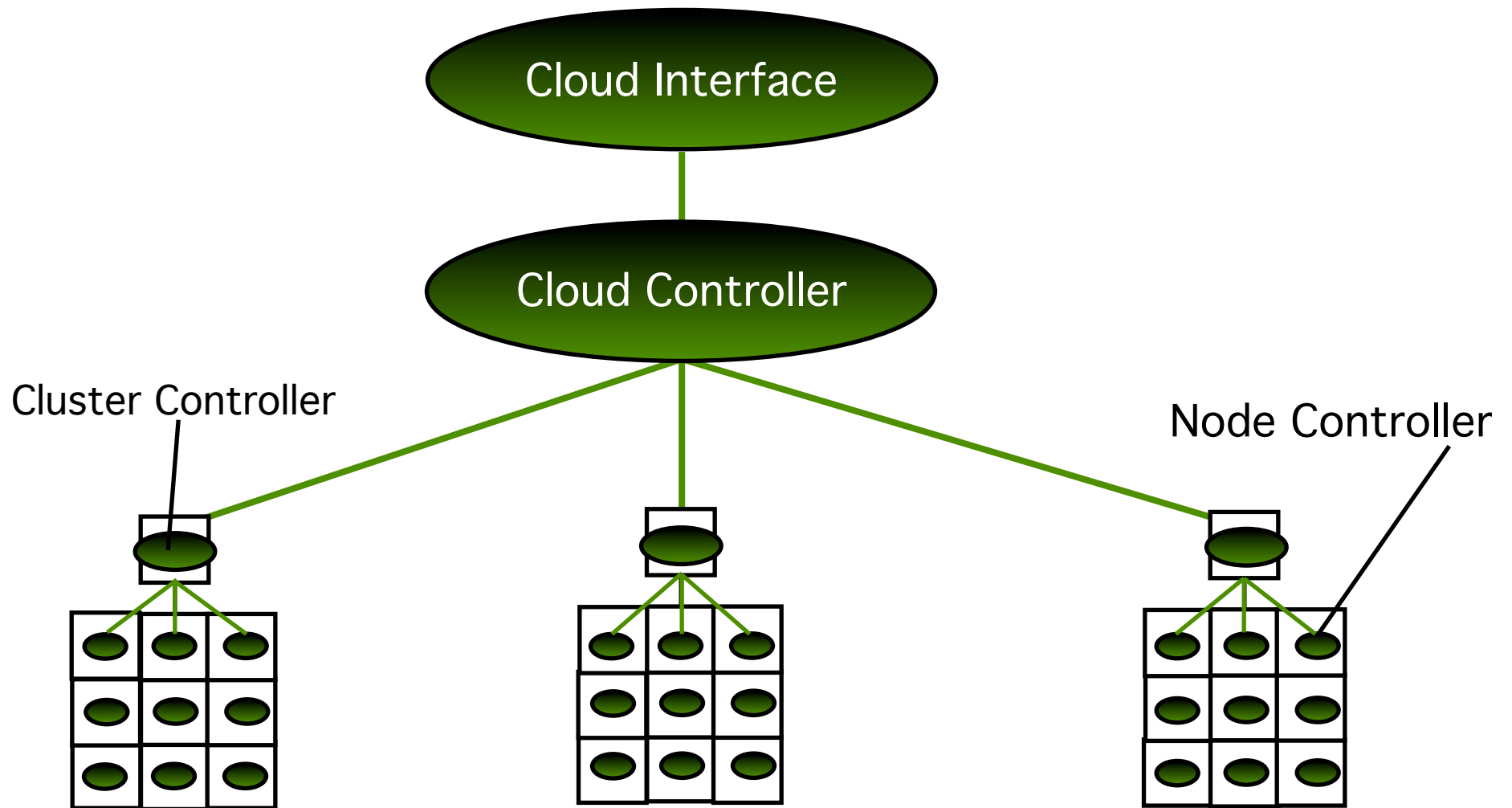
Cloud Computing Research

- **We need a simple, transparent, controllable cloud computing infrastructure to support systems research**
 - What types of interfaces are appropriate for clouds?
 - How should cloud networks be constructed/managed?
 - How are security concerns addressed in “the cloud”?
 - How are various workloads most efficiently farmed out to clouds?
 - What types of applications can run in clouds?
 - What types of service level agreements are appropriate/possible?
- **How do we answer these questions?**
 - Need to be able to perform experiments
 - Need to compare different methodologies and implementations
 - Need flexible cloud computing framework that is aimed at academic researchers
-  **Eucalyptus**
.cs.ucsb.edu



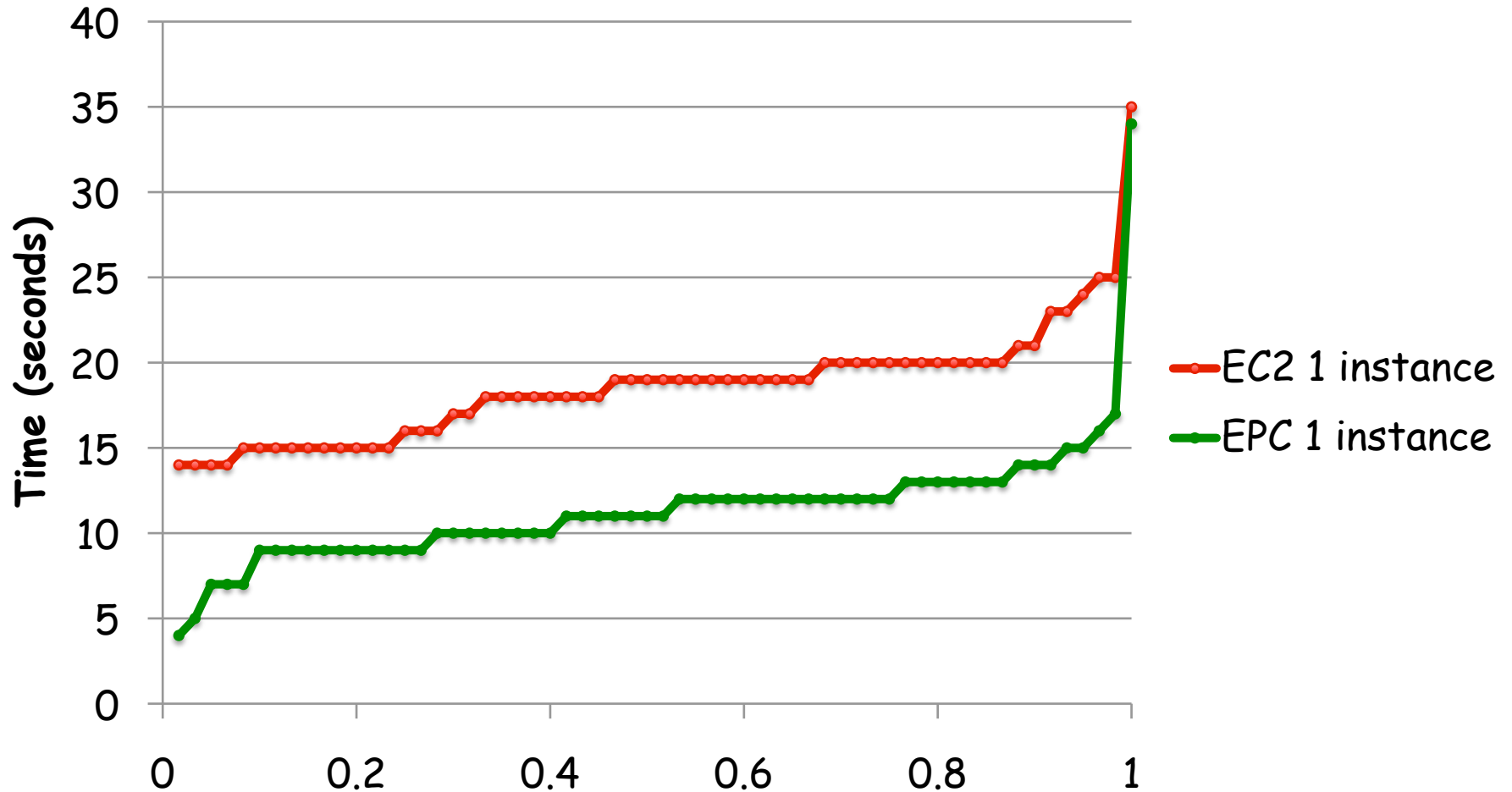
-
- **Overlay software allowing researchers to investigate and experiment with aspects of IaaS style cloud computing**
 - Implements “lowest level” of cloud computing systems
 - Users allocate and de-allocate entire VM instances “on demand”
 - **Designed to easily install on common academic cluster configurations**
 - **Modularized to allow researcher replacement of logical components**
 - **Open-source**
 - Easily instrumented in support of experiments
 - **Flexible user interface (currently compatible with Amazon EC2)**
 - **Researchers can download Eucalyptus and install an EC2 compatible cloud computing system atop existing resources**

Eucalyptus Architecture



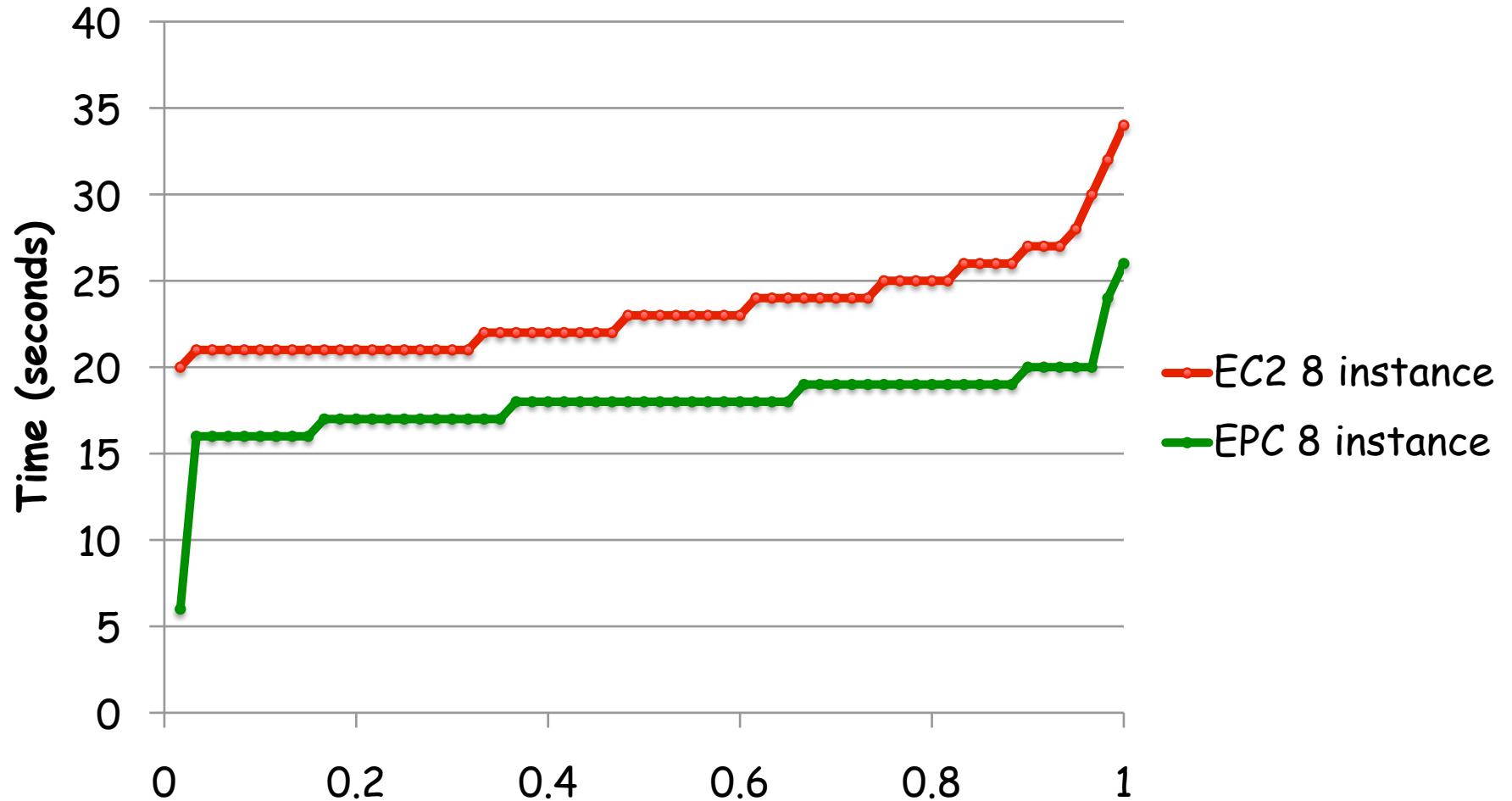
Experimenting with the Cloud

Single Instance Start-up Time (CDF)



Experimenting with the Cloud

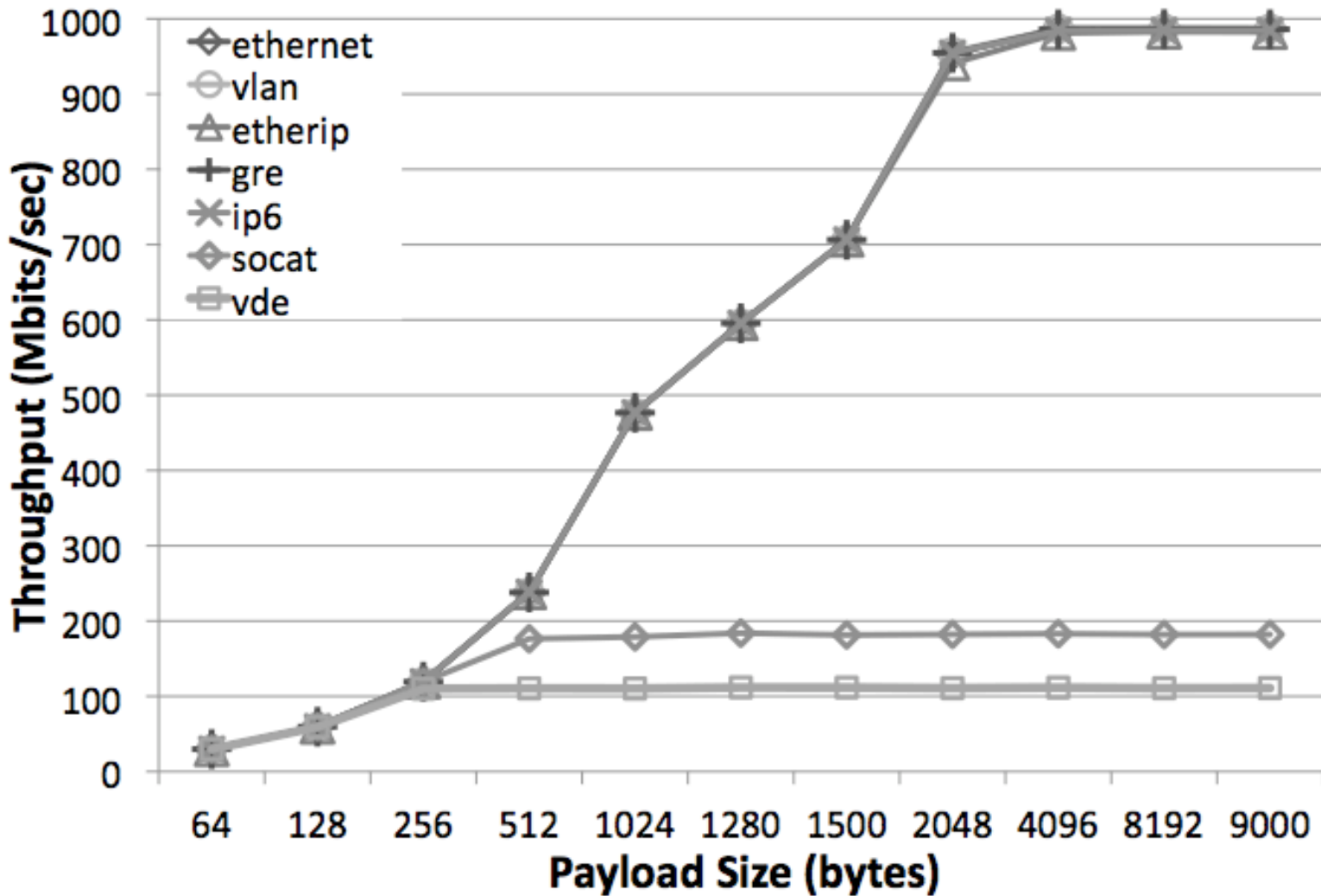
Eight Instance Start-up Time (CDF)



Cloud Networking

- **Goal is to provision a network, similar to the way we provision virtual machines**
 - Should behave like an actual physical network that the user controls
- **Eucalyptus currently provides software overlay network to each user**
 - Layer 2 (ethernet)
 - Spans federated resources (even when IPs are private/unroutable)
 - Near wire speed (gigabit ethernet) within a cluster
- **We use Eucalyptus to explore the problem of cloud networking**

Experimenting with the Cloud



Eucalyptus Impact

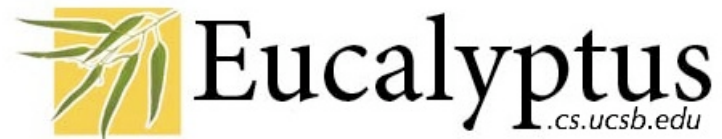
- **Active user community discussion board**
- **Passes EC2 compatible ‘litmus tests’**
 - Amazon tools
 - Rightscale
 - Elastra
 - Elastic Fox
- **Open source AppEngine emulation**
 - Provides Google AppEngine in EC2 or Eucalyptus
- **What are some hot topics?**
 - Hybrid clouds
 - Cloud development
 - Instrumentation
 - Security

Eucalyptus Status

- **Version 1.3 is released**
 - RPM binary packages
 - Rocks binary distribution
 - Source tarball
 - Eucalyptus Public Cloud
 - ~200 users in a few months and growing
 - Active discussion board and growing user community
- **Version 1.4 release in early November**
 - User uploadable images (using Amazon AMI tools)
 - S3 Emulation (Walrus) component for persistent storage
 - More flexible VM networking
 - User defined security rules
 - Dynamic assignment of public IPs to instances
 - Stability, performance improvements

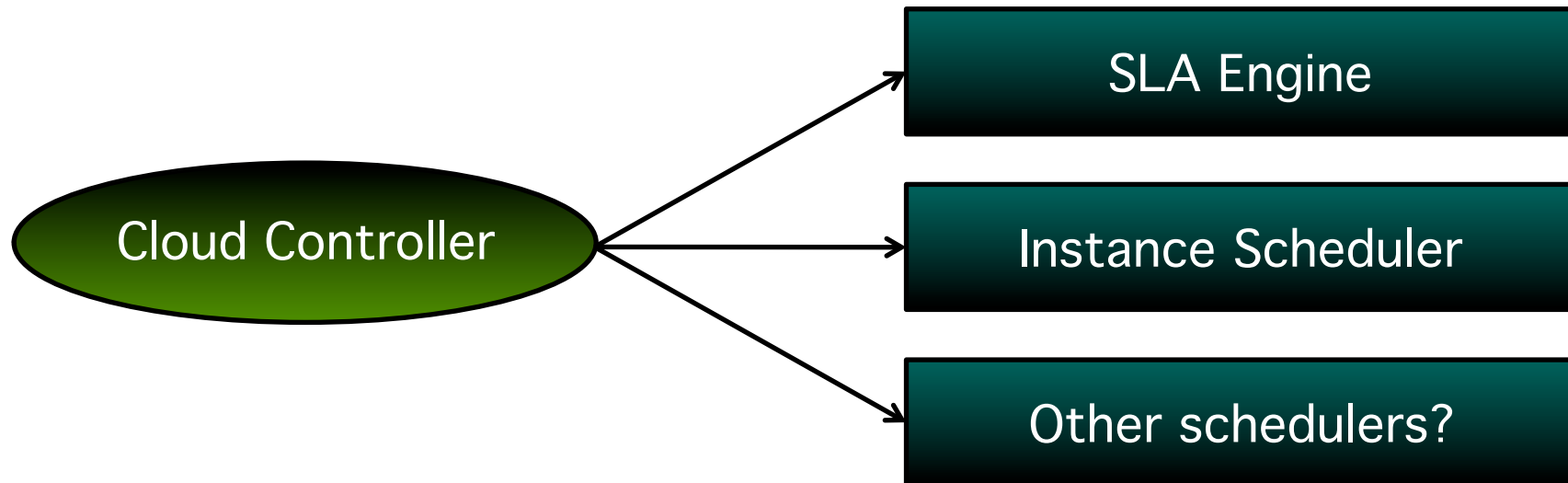
Thank You

- Eucalyptus Project Web-page [<http://eucalyptus.cs.ucsb.edu>]
- Eucalyptus Public Cloud [<https://mayhem9.cs.ucsb.edu:8443>]
- Team Members
 - Rich Wolski (project director) – rich@cs.ucsb.edu
 - Daniel Nurmi (presenter) – nurmi@cs.ucsb.edu
 - Chris Grzegorzczuk – grze@cs.ucsb.edu
 - Graziano Obertelli – graziano@cs.ucsb.edu
 - Sunil Soman – sunils@cs.ucsb.edu
 - Lamia Youseff – lyouseff@cs.ucsb.edu
 - Dmitrii Zagorodnov – dmitrii@cs.ucsb.edu



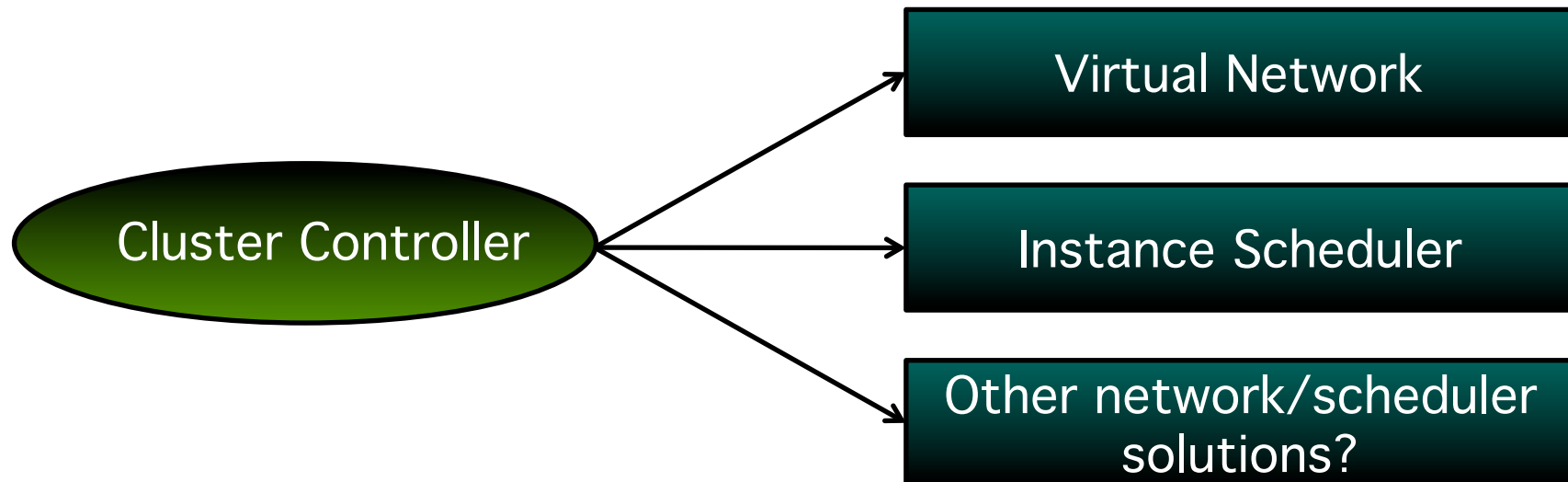
Cloud Controller

- **Web service**
 - Java
- **Handles distribution of VM instance control events to clusters of resources**



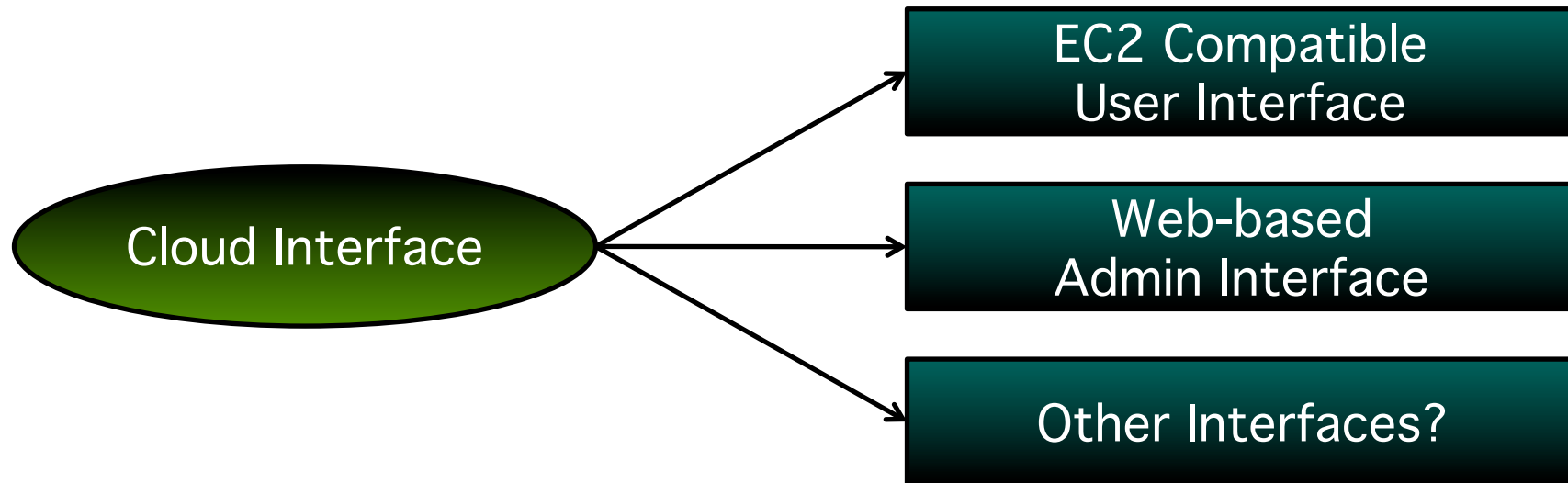
Cluster Controller

- **Web service**
 - C based
- **Handles distribution of VM instance control events to individual of resources**
- **Investigation: what are the performance, invasiveness and functionality tradeoffs of various cloud networking strategies?**



Cloud Interface

- **Web service based**
 - **Written in Java**
- **Handles user event to cloud VM instance control translation**
- **Experiment: instance creation time**



Node Controller

- **Web service**
 - C based
- **Handles management of individual VM instances**

