

Managed Control of Composite Cloud Systems

Christopher C. Lamb, Pramod A. Jamkhedkar, Gregory L. Heileman,
and Chaouki T. Abdallah

Department of Electrical and Computer Engineering
University of New Mexico

June 10, 2011



Outline

- ① UNM Informatics
- ② Usage Management and Cloud Systems
- ③ Example Systems

Areas of Study

Our group:

- *UNM Informatics*: Information security, theory, and architectures; this work is specific to information security
- *Usage Management*: Control of how an artifact is used, covering everything *after* access as well as controlling access itself

Areas of Study

Our group:

- *UNM Informatics*: Information security, theory, and architectures; this work is specific to information security
- *Usage Management*: Control of how an artifact is used, covering everything *after* access as well as controlling access itself

Motivation: We believe people should have control over their own information. Or past motivation for DRM work was to provide content control to content creators. Doing so provides incentive for innovation, and improves quality of life for individuals and society as a whole over time. We believe Usage Management provides the same benefits, and should be unobtrusive.

This motivation holds in this domain as well.

Areas of Study

Our group:

- *UNM Informatics*: Information security, theory, and architectures; this work is specific to information security
- *Usage Management*: Control of how an artifact is used, covering everything *after* access as well as controlling access itself

Motivation: We believe people should have control over their own information. Or past motivation for DRM work was to provide content control to content creators. Doing so provides incentive for innovation, and improves quality of life for individuals and society as a whole over time. We believe Usage Management provides the same benefits, and should be unobtrusive.

This motivation holds in this domain as well.

Acronyms:

- *UM*: Usage Management
- *PMR*: Personal Medical Record (this is also electronic, in this case)

Why is this important?

Utility computing will certainly be the most pervasive future computing model

Why is this important?

Utility computing will certainly be the most pervasive future computing model

- *Mainframes won!*

Why is this important?

Utility computing will certainly be the most pervasive future computing model

- *Mainframes won!*
 - Well, end devices are powerful
 - Cloud computing pervasive for *convenience*, not *technical necessity*.
 - *Still resembles centralized models of the past*

Why is this important?

Utility computing will certainly be the most pervasive future computing model

- *Mainframes won!*
 - Well, end devices are powerful
 - Cloud computing pervasive for *convenience*, not *technical necessity*.
 - *Still resembles centralized models of the past*
- *People should control what they own*
 - Access
 - Retention
 - Distribution

Why is this important?

Utility computing will certainly be the most pervasive future computing model

- *Mainframes won!*
 - Well, end devices are powerful
 - Cloud computing pervasive for *convenience*, not *technical necessity*.
 - *Still resembles centralized models of the past*
- *People should control what they own*
 - Access
 - Retention
 - Distribution

Organizations should control what they pay for

- Systems
- Data
- Records

Problems

So this is what we would like to see, but problems abound

- *Scalability, performance, usability, infrastructural support...*

Problems

So this is what we would like to see, but problems abound

- *Scalability, performance, usability, infrastructural support...*

Started examining automation and ability to combine service level agreements (SLAs)

- *Automation*

How we can automate control and enforcement

- *Combine*

How we can combine multiple SLAs into single SLAs

Problems

So this is what we would like to see, but problems abound

- *Scalability, performance, usability, infrastructural support...*

Started examining automation and ability to combine service level agreements (SLAs)

- *Automation*

How we can automate control and enforcement

- *Combine*

How we can combine multiple SLAs into single SLAs

Surprisingly difficult...

- *NP-Complete*

Simple generalized SLAs are equivalent to *SAT*

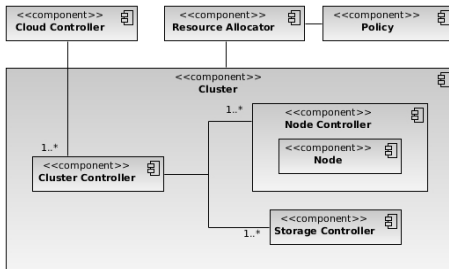
- *Multiple Providers*

Difficult constant factors related to latency, etc.

Single Provider, Feedback

- *Performance:* We will be adjusting a system within specific soft real-time frames. Ergo, we need to be able to collect feedback measurements, process those measurements, and make decisions about how to respond to those measurements quickly in order to avoid falling out of compliance with any performance parameters to which we must adhere.
- *Accessibility:* In order to control component systems, we must be able to access those systems. In order to do so within time constraints, we must be able to access those systems electronically as well; physical access requirements simply will not scale into this performance domain.
- *Controllability:* We must be able to access the appropriate control primitives on the systems we need to tune. This will include accessing compute node generation and termination capabilities. It would help if we could access node performance information and tune those nodes as well, though this is not required; we can emulate this by terminating nodes in one configuration and creating nodes with another to more adequately address performance needs.

Single Provider, Feedback



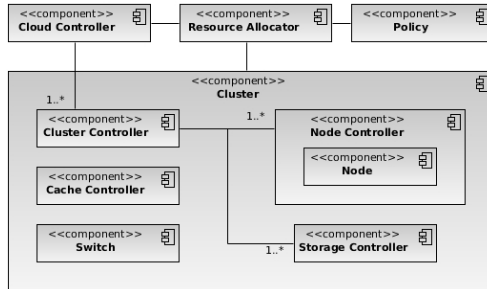
Single Provider, Feedback

- *Cloud Controller* Provides an initial interface to administrative users to control the cloud.
- *Cluster Controller* Managed by the cloud controller, the cluster controller manages the resources of a single cluster. A given cloud may contain multiple clusters.
- *Storage Controller* Provides storage of system images and for other general storage needs. This controller component is highly I/O sensitive.
- *Node Controller* Responsible for allocating, delivering, and managing individual compute nodes upon which client software runs.
- *Node* The compute node delivering services to end users and managed by the cluster's control infrastructure. This is the primary computational resource accessed by users accessing managed cloud resources.
- *Policy* Quality of service terms the cloud provider has agreed to honor for the cloud customer with respect to system delivery, provisioning, and overall performance.
- *Resource Allocator* The component responsible for real-time tuning of the cloud system to maintain defined quality of service.

Single Provider, Feedback with UM

- *Accessibility*: Data streamed through the cloud system must be able to be monitored and the accessibility of that stream needs to be dynamically tunable. This implies that we need to be able to control routing and caching of all streaming data according to user specified conditions. This also implies that we need to be able to control exactly which *Node Controllers* are able to spawn which *Nodes*.

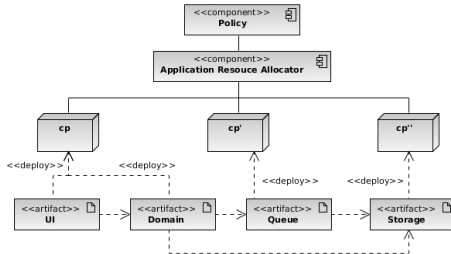
Single Provider, Feedback with UM



Single Provider, Feedback with UM

- *Cache Controllers*: Streaming network data, specifically media-centric streams, can and are cached by strategically located cache systems. In order to control the read access of network data, we must be able to exercise explicit control over any caching systems in our infrastructure.
- *Switch*: Really any kind of hardware that controls the delivery of network data. This component includes switches and routers primarily. In order to control how data is accessed we must be able to control the locations to which it is delivered.

Multiple Providers



Conclusions