Automated Control in a Cloud Computing Infrastructure

Harold Lim
Duke University
What is Cloud Computing?

- General Overview
What is Cloud Computing?

- 2 Varieties of cloud computing
  - Software as a Service
  - Infrastructure as a Service

- Clouds are an important (and different) context for autonomic control.
  - Elastic Provisioning for variable load
    - Performance
    - Fault Tolerance
    - Interactions among services/modules/tiers
  - Reflective Control
Motivation

- Consider an abstraction of a cloud computing infrastructure service.
- Two kinds of entities: providers and consumers (guests).
Motivation

- Opportunity for guests to monitor deployed application and modulate resources automatically (through slice controllers).
  - Resource efficient (minimize cost)
  - Acceptable quality of service
Motivation

Automated control in a cloud computing infrastructure

- Resources are elastic, but still needs to decide how, where and when to use the resources.
- Can be different from controlling other computing systems.
  - Prior works have studied how to control services (which might also run on cloud platform).
  - Have to consider how to adapt these control loops to the cloud computing context.
What is the Right Control Architecture?

- Autonomic computing for other systems, such as Data Centers: Centralized Control.
  - Has all knowledge and power.
  - Ability to control is unconstrained.
Decoupled Control

In cloud computing, nobody knows and control everything.

Decoupling is fundamental to the architecture.

- Want the cloud to be a general platform for all kinds of (diverse) guests.
- Guests want their services to run on any cloud without having to know about the cloud's internals.
Decoupled Control

- Each entity has an independent controller.
- How do these controllers interact? Limited view of the other entities.

Diagram:
- Slice controllers monitor and acquire resources to respond to their guest's changing resource needs.
- Cloud controllers monitor and arbitrate access to resources by guests.
Service Interface

Constraints

- Guest controller can only control the resources through a service interface.
- Providers discretize (coarse) resources into pre-defined sizes. e.g., Amazon EC2
- Unavailability of certain type of actuators. (e.g., hypervisor-level)
- We introduce a new policy, called proportional thresholding [ACDC09].
  - Dampen control loop.
  - Convert target value to dynamic target range.
Cloud Provider Issues

- What is the right mechanism and policy for arbitration of requests?

- If resources are congested, how does a cloud platform pick which request to grant?

- Provider is a general platform and does not know the particulars of a guest service.
Cloud Provider Issues

- **ORCA**- service-oriented infrastructure that provides resource leasing abstractions.
  - Separates mechanism and policy.
  - Resource provider can modify the arbitration policy. (pluggable)
  - Guests use the ticket mechanism to submit requests.
  - Tickets contain guest-configurable properties.
Other Issues

- Stateful services / services with persistent data
  - Inertia and Lag
    - How to take advantage of elasticity of resources, when services add delay/lag before allocated resources is ready to contribute?
      - Delay before new node is ready. (migration, replication of data)
      - Need to compensate for this delay.
    - Proposed Solution: Cool Standby
Cool Standby

- Periodically sync data with backup nodes.

- When controller needs additional nodes, it simply has to turn on the nodes.

- Reduces delay time, since backup nodes already have most of the data.
Other Issues

What is the right control architecture for complex guest services?

- Composing multiple cloud services.
- Can use decoupled control (further decoupling of the slice controller).
- Decompose controller into local controllers.
  - + Easier to model because each controller only needs to know a piece of the overall service.
  - + Parts are pluggable/ can easily change one of the controllers/services.
  - - May not be optimal
  - - Emergent misbehaviors
Thank You.