School of informatics | cisa | Centre for Intelligent Systems and their Applications

THE UNIVERSITY OF EDINBURGH

Configuring the Cloud

Inside and out

Paul Anderson
<dcspaul@ed.ac.uk>

http://homepages.inf.ed.ac.uk/dcspaul/publications/mysore-2010-talk.pdf

Configuration in the Cloud

The "customer-visible" service interface remains the same, but the underlying implementation continually changes ...

- If a hosting provider goes out of business, applications can easily switch to a different provider
- Processing/storage resources are acquired and released as the demand changes
- Processes/storage migrate in search of cheaper providers
- Storage migrates to be closer to the processing

The components remain largely the same - their number, location and relationships ("configuration") change

Other Layers of Configuration

The layers of infrastructure which support the cloud services require a similar configuration ...

- Network
  - *routing, address assignment, name services*
- System
  - *OS, authentication services, file services*
- Virtualisation
  - *virtual/physical mapping, storage mapping*
- Services
  - *load balancing of a web service*

Overview

It is tempting to treat cloud services as another independent layer with its own configuration procedures and technologies

But I'd like to argue that it is very useful to consider a more holistic approach to configuration of all the layers ...

- I'll explain why I think this is a good idea
- I'll give some examples of configuration issues at different layers
- I'll abstract some of the important problems and show how they might be relevant to cloud service configuration
- I'll say a little about my configuration research

A More Holistic Approach to Configuration ?

Layer Dependencies

How realistic is it to expect to configure layers efficiently without any knowledge/control of the other layers?

- Co-locating highly dependent cloud services requires knowledge/control of the physical locations
- Migrating a virtual machine involves changes in the network routing and possibly other services
- Responsive load balancing requires some knowledge of the service characteristics
- Deploying a redundant service needs to guarantee that the copies do not depend on any common infrastructure

A More Holistic Approach

A more holistic approach to specifying configurations would ...

- Provide a common "language" for different layers to share configuration requirements
- Support the explicit specification of inter-layer dependencies
- Support automatic reasoning and validation across the layers
- Allow configuration problems at one layer to benefit from solutions to similar problems at other layers
- Speed the development of underlying theories & tools

In The Meantime ...

Many different communities and perspectives are involved, so there won't be a "universal" configuration paradigm at any time soon!

- But we can learn useful lessons from looking at the problems and approaches in the other layers
  - *these may provide good models and examples for configuration at the cloud service level*
  - *we can better understand the potential problems with inter-layer dependencies*

The Configuration Problem at Other Layers

Network Configuration

Network configuration operates at the layer of "switches" and "routers" and "addressing" etc ..

- A move towards dynamic configuration under the control of central policies
  - *DHCP instead of fixed IP addresses*
  - *Dynamic DNS*
- Distributed configuration
  - *DNS*
  - *dynamic routing, rather than fixed source-routing*
- Formal verification of properties
  - *for reliability, security etc..*

System Configuration

System configuration operates at the layer of "servers" and "workstations" and "clusters" and the system software that runs on them ...

- Specification languages are important
  - *describing complex entities*
- Deployment is important
- Autonomics
- Mostly centralised, rather than distributed
  - *although there is a need for a more distributed approach*
- A need for continuous, incremental changes
  - *don't "undeploy" the dns service*

Virtualisation Configuration

Virtualisation configuration operates at the layer of "virtual machines", "virtual networks" and "virtual storage"

- Both the physical, and the virtual machines need to be configured
  - *currently, this usually happens independently*
- There are lots of dependencies
  - *between the VMs, network and storage*
- Autonomics is important
  - *for failure recovery*
  - *for load balancing*

Service Configuration

Service configuration operates at the layer of "web servers" and "load balancers" and "database services" and the relationships between them ...

- Dependencies between services are important
- "Composition" is important
  - *services depend on one another in complex ways*
- Deployment and undeployment sequencing are important
- Elastic services change configuration dynamically in response to demand

Some Common Underlying Issues

Degrees of Automation

There is often a tension between the desire for a fully "autonomic" service, and the need to have some aspects confirmed by a human decision
We are interested in a framework which would support an integrated range of approaches ...

- Fully autonomic decisions
  - *guided by "constraints" or "policies"*
- Selection from a set of canned solutions
- Fully manual decisions
  - *authorised by various different people*
- Some combination of all of the above

Separation of Concerns

Many different people (or automatic systems) may have requirements on some aspect of a configuration

- Note that an "aspect" is "cross-cutting" concern
  - *not simply a question of modularity*
- How do we compose these requirements without involving manual negotiation?
- What happens if they conflict?
- How do we manage security

Specifying Configurations

There are many advantages to "declarative" specifications which describe the desired state, rather than the sequence of steps needed to achieve it

- Policies can be implemented directly
  - *Service X must not run on the vendor as service Y*
- Declarative aspects can be composed
  - *Service Y must run on vendor A,B or C*
- Properties can be proven much more easily
- But this is a difficult implementation problem
  - *potentially a large constraint satisfaction problem*

Deploying Configurations

Deploying declarative configurations is a planning problem

- Moving a service from one vendor to another may mean moving many related components
  - *the transfer has to be sequenced so that the overall service does not breakl at any point during the transfer*
- Deployment can be compounded by uncertainty
  - *on a large scale, there will always be broken services*
  - *or are they just "slow" ?*
  - *or is the monitoring service broken ?*

Human Factors

Configuration errors are responsible for a large proportion of downtime. Most of these are due to misunderstandings & human error.

- Configuration languages need to be clear and simple for a wide range of different users
- Extreme automation can have unpredictable effects
  - *why has the company research experiment just been migrated to the competitor's cloud service ?*
- Automatic systems need to explain decisions
- A flexible framework allows gradual automation

Distributed Configuration

For performance and reliability reasons, it is not always practical to evaluate the entire configuration centrally

- We may not be able to plan the composition and deployment of an entire service in advance
- It is fairly obvious that we can introduce a hierarchical model
  - *but it is not clear how to specify this*
  - *particularly considering the previous issues such as cross-cutting concerns*
- This problem analogous to moving from conventional to distributed programming

Finally ...

Summary

- Exploitation of cloud technology for large-scale applications will entail some difficult configuration issues
- Many of these issues are alreday problems for other layers of the supporting infrastructure
- There would be many advantages to a unified configuration framework across the layers
- But this is not likely to happen soon
- However, we can learn pehaps something from the problems and solutions at the other layers

Some Research Issues

- Declarative languages and constraints
- Frameworks for integrating manual and automatic decision making
- Planning configuration change sequences
- Distributed configuration negotiation
- Human issues

School of informatics cisa | Centre for Intelligent Systems and their Applications

THE UNIVERSITY OF EDINBURGH

Configuring the Cloud

Inside and out

Paul Anderson
<dcspaul@ed.ac.uk>

http://homepages.inf.ed.ac.uk/dcspaul/publications/mysore-2010-talk.pdf