Managing Multi-Cloud Systems with CloudMF

Nicolas Ferry, Franck Chauvel, Alessandro Rossini, Brice Morin, Arnor Solberg
SINTEF ICT – MOD Group

NordiCloud 2013
Introduction

- Complex software systems

- **DAS** to tame complexity of managing complex systems
  - Focus on the software

- **Cloud-computing** allows the management of the complete stack
  - Infrastructure, Platform, Software

- Lack of integration between
  - Cloud management solutions
  - DAS technics and methods
The Cloud Modelling Framework (CloudMF)

Two main components:

- A **modelling environment** with a tool-supported domain-specific modelling language (DSML) to model the provisioning and deployment of multi-cloud systems.
- A **models@run-time environment** for enacting the provisioning, deployment and adaptation of these systems.

Relies on solutions for cloud management.
CloudMF architecture

Modeling environment

Editor → CPIM → Refinement engine

Declarative deployment

Reasoning engine

Adaptation

Target CPSM

Current CPSM → Diff → Adaptation engine → Adaptation actions

Monitoring

Models@run-time environment

Provisioning and deployment engine

Imperative deployment
The Modelling environment

- A Model-driven approach with two levels of abstraction
  - Cloud Provider-Independent Model (CPIM)
    - cloud concerns related to the application in a cloud agnostic way
  - Cloud Provider-Specific Model (CPSM)
    - cloud concerns needed to deploy and provision the application on a specific cloud.

- Two main tools
  - A language CloudML along with an editor to manipulate the models
  - A refinement engine to transform CPIM into CPSM (CPIM enriched with provider specific metadata)
CloudML: main concepts

- Generic virtual machine

- Provisioning requirements
  - 2 cores $\leq$ compute $\leq$ 4 cores
  - 2 GiB $\leq$ memory $\leq$ 4 GiB
  - storage $\geq$ 10 GiB
  - location = Europe
CloudML: main concepts

- Generic component of the application
  - a Java servlet of an application
  - a Jetty container
  - a MongoDB database
CloudML: main concepts

- Deployment commands
  - retrieve the Java servlet from cloudml.org
  - configure it
  - run it
CloudML: main concepts

- Deployment dependencies
  - the Jetty container and the MongoDB database have to be deployed before the servlet

- Communication channels
  - a servlet communicates with another servlet through HTTPS on port 443
Models@runtime

- Architectural pattern for DAS
- Using a Causal Link
  - Bidirectional Synchronization
- Benefits
  - Separation of Concerns
  - Reuse of MDE tools for free
  - Free testing space
The Models@runtime environment

- CPSM causally connected to the running system
- A change in the CPSM is reflected on-demand in the running system
- A change in the running system is automatically reflected in the CPSM
Summary

- A framework to **provision, deploy and adapt** multi-cloud systems
  - Bring together DAS technics and classical cloud solutions
  - **Model-driven** approach at runtime and at design-time

- Focus on the IaaS level
Status

- Resources
  - http://cloudml.org
  - https://github.com/SINTEF-9012/cloudml/

- Used in several EU projects

- Future works
  - Introducing PaaS concepts
  - Distributed models@runtime
Thank you!

Contact: nicolas.ferry@sintef.no
Demo

- Templates for SensApp:

  A: SensAppAdmin
  wc: Jetty
  java: Jdk17
  Ubuntu1204: VirtualNode

  S: SensApp
  wc: Jetty
  java: Jdk17
 Ubuntu1204: VirtualNode
  mg: MongoDB