Cloud Applications: Less Guessing, more Planning and Knowing

Josef Spillner <josef.spillner@zhaw.ch>
Service Prototyping Lab (blog.zhaw.ch/icclab)

23.05.2016 | Universidade de Coimbra
Cruise Ships History

England 1842: P&O → Postship with spare capacity in the Med
Germany 1888: Augusta Victoria → Liner in summer, cruiser in winter time [Klu01]
Winterthur Computer History

Remington Rand (pre-Unisys), Swiss daughter company Mithra [Bru11]:
- M9 (alias Z9), designed around 1953-1954
- using Stibitz coding instead of regular BCD: easier handling of negatives
- users: Winterthur city administration(*), machine producer Rieter in Winterthur, Swissair, Zurich cantonal electricity provider, & many more

- business model:
  - buy
  - lease/rent (with time-based fee)

- (*) only one M9 left world-wide in Museum for Communication, Bern
  - installed in 1961 for computing of bills for utilities
  - using punch cards for input and output (no screen, no printer)
Winterthur Computer History

The M9: Cases and relays [Bru11]
Bringing it all together...

Capacity use: Augusta Victoria → early “virtualisation“
Business model: M9 e.g. in Winterthur → early “PAYG“ service model

+ scaling
+ on-demand provisioning

- risks

= cloud computing infrastructure perspective
... and predicting the future

Fundamentals: resource services

- Compute
  - governed by Moore's Law (for transistors) [CuYo16]
- Networking
  - governed by Keck's Law (for optical fiber) [Hec16]
- Storage
  - governed by Kryder's Law? (for magnetic disks; not quite) [Ros14]

Services on top: not so straightforward - not just speed
Cloud Applications Perspective

Initial onboarding

```
Application Analysis

Application Preparation & Transformation

Virtual Machine Image
  Virtual Machine Hosting

Container Image
  Container Hosting

Framework Package
  Framework Hosting

Single Function File
  Single Function Hosting

Provider Selection

Plan Configuration

Deployment
```
Cloud Applications Perspective

Continuous operation and re-engineering
Planning and Knowing

A-priori safeguarding: design for risks/threats...

- failure / unavailability
- unpredictable popularity
- leaks / surveillance
- dependency changes and issues

A-posteriori knowledge

- monitoring
- event processing
- log file analysis
- incident detection
Novel Techniques

- Cloud-Native Applications
- Stealth Computing
- Active Service Management / Fault Injection
Technique: Cloud-Native Applications
Technique: Stealth Computing

Stealth layer: Coverable cloud service evolution

s: cloud services

[Image sources: dreamstime.com, bitrebels.com, suitsof armour.com]
Technique: Stealth Computing

resource & service multiplexing

local resources

Storage resource service

Network resource service

network multiplexer/interface

File storage interface

Stream storage interface

File storage service

Database service

Stream storage service

Event stream processing interf.

Application service (SaaS)

Application interface

Lambda interface

stealth layer

Compute resource service

local resources
Technique: Stealth Computing

- **Compute**
  - **Stealth Processing**
    - Algorithms
    - Optimised execution
    - Programming model

- **Networking / Storage**
  - **Stealth Transmission**
    - Data relation schemas
    - Data transport
  - **Stealth Distribution**
    - Distribution assignment
    - Fragment ordering
  - **Stealth Coding**
    - Combined coding
    - Steganography
    - Encryption
    - Compression
    - Recursive coding
    - Forward error correction

- **Availability / NFF Stability**
  - Techniques: Stealth search and arithmetics
  - Technique: Redundancy search

- **Sampling**
  - Relation schemas: DRS, FRS
  - Technique: PICav+

- **Control / Vertical**
  - Technique: Bit expansion
  - Technique: Bit splitting

**Usage pipeline/vertical**

**Static service selection**

**Policies**

**Service access**

**Operator distribution**

**Configuration vertical**
Technique: Active Service Management

MC-EMU from SPLab
- library of common faults
- failure patterns
- injection plugins

OpenStack Component X

OpenStack Component Y

OpenStack Component Z

Operating System/ Hardware Resources

Watchtower from ICCLab
- monitoring integration
- rules matching
- actions for error handling
Tools

Existing tools for planning and knowing
- network simulators and emulators
- CloudSim [Buyya et al.]
- EMUSIM [Calheiros et al.] for performance
- Cloud Workflow Emulator [Senna et al.] for resources/performance
Tools

MC-EMU: Multi-Cloud Emulation

4 targets:
- no-op (sim.)
- web/file server (storage)
- OS container (compute)
- L4 proxy (network)

3 properties:
- availability
- slowness
- popularity

3 behavioural models:
- convergence
- incident
- replay/library

Emulation of Service Unavailability on Scenario 'picav'

- Google Drive/0.99900
- Amazon S3/0.98800
- AT /0.99500
- Linode/0.99951
- Apple iCloud/0.99650
Tools

MC-EMU example: storage/availability/convergence
Tools

MC-EMU example: compute/popularity/spikes

(above) without failures
(below) with induced failures in a CNA application
Service Prototyping Lab - Research

Service Prototyping Lab (SPLab)

- Research Theme: Pervasive Services
  - Service Tooling
  - Continuous Deployment for Cloud Services
  - Operations

- Research Theme: Service-Based Applications
  - Cloud Robotics
  - Cloud Native Application Design
  - Active Service Management

this talk

InIT Cloud Computing Lab (ICCLab)

- Research Theme: Energy
- Research Theme: Infrastructure
- Research Theme: Platform
Service Prototyping Lab - Events

Open Cloud Day
15.06.2016
Winterthur

Cloud Computing Summer School
4.-15.7.2016
Winterthur

IEEE/ACM UCC
Shanghai
Conclusion

Service Prototyping Lab + Cloud Computing Lab

blog.zhaw.ch/icclab
github.com/serviceprototyinglab

Active Service Management research initiative
→ enforce predictable application behaviour
→ designs, methods and tools

Obrigadão!

2016
Sources


(Rosenthal's discussion of Kryder's Law @ UNESCO)
http://www.theregister.co.uk/2014/11/10/kryders_law_of_once_cheaper_storage_disproven/?page=2