Stairway to Heaven: An Architecture-Level Characterization of Cloud Migration Strategies

Nabor C. Mendonça
Programa de Pós-Graduação em Informática Aplicada (PPGIA)
Universidade de Fortaleza (UNIFOR)
Fortaleza, Ceará, Brazil
nabor@unifor.br, nabor.mendonca@gmail.com
https://sites.google.com/site/nabormendonca/

SEIF Brazil Workshop 2013, November 25-26 | Rio de Janeiro
• Cloud computing (quick) overview
• Cloud migration strategies
• Concluding remarks
Cloud Computing

- Essential characteristics

Ref: The NIST Definition of Cloud Computing

On-demand self-service
Ubiquitous network access
Location transparent resource pooling
Rapid elasticity
Measured service with pay per use

Source: http://aka.ms/532/
Cloud Computing

- Delivery models (vs. traditional IT)

Source: Microsoft presentation.
Cloud Computing

• Risks and benefits

Source: http://www.tridens.si/
Talk Outline

• Cloud computing

• **Cloud migration strategies**

• Concluding remarks
Motivation

• Cloud migration is still a difficult process for many organizations

Source: http://www.cloudtweaks.com
Motivation

• Existing approaches are either too broad in scope…

Source: http://www.neevtech.com/
Motivation

- ... or too narrow in terms of architectural alternatives

Goal (and Non-Goal)

• Identify and characterize useful cloud migration strategies from an architectural perspective

• Specific focus on *application components* and their *service dependencies*

• How each strategy would affect (or be affected by) other factors, both technical and non-technical, is beyond the scope of our present work
Basic Assumptions

• Components as the unit of migration

• Cloud migration means that a component will benefit from the cloud somehow, not necessarily run on it

• After migration the target component must be able to resolve all its (original) services dependencies, including those to other external components and the underlying execution environment
We use a simple graphical notation (loosely based on UML component diagrams) to illustrate our cloud migration strategies in terms of key architectural elements:

- Component
- Service dependencies
- Execution environment
- Cloud environment
Component and Service Dependencies

Component

External Services

Environment Services

External Services
Execution Environment

Environment Services

T1 T2 Tn

Execution Environment
Cloud Environment

Diagram showing Cloud Environment and Public Services connected to Cloud Services.
Cloud Migration Characterization

- Relocation
- Cloudification
- Compensation
Cloud Migration Characterization

- **Relocation**
- **Cloudification**
- **Compensation**
Cloud Migration Characterization

- Relocation
- Cloudification
- Compensation

![Diagram of migration process from Local Environment to Cloud Environment]

After migration
Cloud Migration Characterization

- Relocation
- Cloudification
- Compensation

Before migration
Cloud Migration Characterization

- Relocation
- **Cloudification**
- Compensation

![Diagram showing cloud migration from local environment to cloud environment](diagram.png)

After migration
Cloud Migration Characterization

- Relocation
- Cloudification
- Compensation

Before migration
Cloud Migration Characterization

• **Relocation**

• **Cloudification**

• **Compensation**

After migration
Cloud Migration Characterization

- **Relocation**
- **Cloudification**
- **Compensation**

After migration
Strategies Classification

- Each of those strategies are further classified according to the following criteria:
  - Implementation mechanisms
  - Service compatibility requirements
  - Migration effort
# Strategies Classification

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Implementation Mechanisms</th>
<th>Compatibility Requirements</th>
<th>Migration Effort</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>ID</td>
<td>Type</td>
</tr>
<tr>
<td>Relocation</td>
<td>Re-binding</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Adaptation</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td></td>
<td>Transformation</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Cloudification</td>
<td>Replacement</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Adaptation</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td></td>
<td>Transformation</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Compensation</td>
<td>Incorporation</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Suppression</td>
<td>□</td>
<td>□</td>
</tr>
</tbody>
</table>

Legend: **CC** (Component Change), **CM** (Component Migration), **DM** (Data Migration), **SC** (Cloud Service Configuration), **AR** (Architecture Reconfiguration)
Re-binding

Before Migration

C1

T1

Local Environment

C2

Cloud Environment

After Migration

C1

T1

Local Environment

C2

T1

Cloud Environment
Relocation Mechanisms

- Adaptation

Before Migration

After Migration
Relocation Mechanisms

- **Transformation**
Cloudification Mechanisms

• Replacement

Before Migration

After Migration
Cloudification Mechanisms

- Adaptation (with local adapter)

Before Migration

- C1
- C2
- T1
- T2
- Local Environment
- Cloud Environment

After Migration

- C1
- Adapter
- T1
- T2
- Local Environment
- Cloud Environment
Cloudification Mechanisms

- Adaptation (with cloud adapter)

Before Migration

After Migration
Cloudification Mechanisms

• Transformation

Before Migration

C1

T1

C2

Local Environment

Cloud Environment

Cloud Service

T2

After Migration

C1

T2

T2

Cloud Service

Local Environment

Cloud Environment
Compensation Mechanisms

- Incorporation (of external service)
Compensation Mechanisms

- Incorporation (of environment service)
Compensation Mechanisms

• Suppression (of external service)
Compensation Mechanisms

• Suppression (of environment service)
Talk Outline

- Cloud computing (quick) overview
- Cloud migration strategies
- Concluding remarks
In Conclusion

• Cloud migration requires a careful examination of the target application’s runtime architecture

• Our proposed characterization can support developers in this analysis by highlighting several useful component migration mechanisms as well as their implementation requirements
Ongoing Work

• Implementing some of our proposed cloud migration mechanisms in the form of non-intrusive code transformations (“cloud detours”)
  
  – Early prototype uses AOP and generic cloud APIs (e.g., jclouds) to automatically transform (“cloudify”) a file-based legacy application to use a cloud-based storage service

• Winner of a SEIF 2013 Ward
Future Work

• Refine the proposed characterization / classification
  – Additional migration scenarios and purposes
  – Additional implementation mechanisms

• Cloud-bound architecture recovery and conformance-checking

• Other suggestions?
What About the ‘Stairway’ from the Title?

“There's a lady who's sure all that glitters is gold
And she's buying a stairway to heaven. ...”

Stairway to Heaven
(Jimmy Page – Robert Plant)
Led Zeppelin, 1971

Source: http://timtirelli.com/
Not all that glitters in the cloud is gold.
Understanding your architectural choices and migration alternatives can be a good first step towards building a gentle stairway to (cloud) heaven!

Source: http://www.enterprisecloudtoday.com/
Thank You!
Stairway to Heaven: An Architecture-Level Characterization of Cloud Migration Strategies

Nabor C. Mendonça
Programa de Pós-Graduação em Informática Aplicada (PPGIA)
Universidade de Fortaleza (UNIFOR)
Fortaleza, Ceará, Brazil
nabor@unifor.br, nabor.mendonca@gmail.com
https://sites.google.com/site/nabormendonca/