Coverage-Based Metrics for Cloud Adaptation

Yonit Magid, Rachel Tzoref-Brill, Marcel Zalmanovici

IBM Haifa Research Lab

This project is funded under the European Union’s Seventh Framework Programme (FP7) under grant agreement 610802

© 2016 IBM Corporation
Outline

CloudWave
Cloud Adaptations
Combinatorial Model
Risk Assessment Engine
Preliminary Results
Questions
CloudWave

- Research project developing the next gen Cloud stack
  - Funded by the EU

- Cloud infrastructure and hosted applications collaborate
  - Increase level of service
  - Lower costs

- Dynamically adapts cloud services to their environment
  - Improving service quality
  - Optimizing resource use

- Enables DevOps-like interfaces with the developer
Coordinated Adaptation: Reconfigures the application and infrastructure in real-time to compensate for a variety of performance factors, resulting in an increasingly resilient, automated and optimized cloud deployment.
Outline

- CloudWave
- **Cloud Adaptations**
- Combinatorial Model
- Risk Assessment Engine
- Preliminary Results
- Questions
Cloud Adaptations

- **Migration**
  - Move one (or more) VMs, not necessarily the one(s) reporting problems

- **Scale-out**
  - Add more VM(s)/hardware instances

- **Scale-up**
  - Add more resources to VM(s) running the application

- **Application adaptation**
  - App. changes its resource usage, e.g. reducing/delaying services
Cloud Adaptations (cont.)

- Evaluation Engines
- Target Engines
- Recommendation Engines
- Constraint Engines

Inputs:
- Cost, risk, …
- Hardware, …
- Analytics, Developer input, …
- Cost, physical location, …
Outline

CloudWave

Cloud Adaptations

Combinatorial Model

Risk Assessment Engine

Preliminary Results

Questions
Combinatorial Model

- Using the IBM FOCUS Combinatorial Testing engine traditionally used for testing at the development phase
Combinatorial Model (cont.)

- **Model based on HW features.** May contain any measurable SW/HW features

<table>
<thead>
<tr>
<th>VM₁</th>
<th>VM₂</th>
<th>VMₙ</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>Mem</td>
<td>Net</td>
</tr>
<tr>
<td>Low</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>High</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Test space (size of the cross-product) is huge**
  - E.g. an app with 10 modules (VMs) and 4 HW groups:
    - Each VM: 3 x 3 x 3 x 4. In total: $10^4 = 108^{10} = 2.16 \times 10^{20}$

- **Coverage metric**
  - Pairs (can be any tuple) of parameter values (in our example ~8K)
Combinatorial Model (cont.)

- Example of measures while running tests

<table>
<thead>
<tr>
<th></th>
<th>VM₁</th>
<th></th>
<th>VM₂</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CPU</strong></td>
<td>Mem</td>
<td>Net In</td>
<td>Net Out</td>
</tr>
<tr>
<td>Medium</td>
<td>Medium</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Low</td>
<td>High</td>
<td>Low</td>
<td>Medium</td>
</tr>
<tr>
<td>High</td>
<td>Low</td>
<td>Medium</td>
<td>High</td>
</tr>
<tr>
<td>Medium</td>
<td>Low</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>High</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Medium</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
</tr>
</tbody>
</table>

- Each **row** is a measurement
- Every pair of values is a coverage task
- Each **test** may generate more than one row
Risk Assessment Engine
Risk Assessment Engine - Novelty

- Using Combinatorial testing (CT) outside test design phase
  - Learning new "safe" combinations during production
  - Blurring the boundary between testing and production on the cloud
- Use CT to predict missing coverage of potential adaptations
  - Instead of measuring existing combinatorial coverage
- Using CT for load testing
  - As opposed to its traditional use for functional testing
- Using (load) history to prioritize adaptation actions by risk
Risk Assessment Engine (cont.)

- **Goal:** *estimate* the risk of encountering a bug

- How is it calculated?
  - Inputs
    - Model describing the world (*dimensions*)
    - Test concern (*coverage metric*)
    - Baseline data-points monitored while running tests (*test coverage*)
    - Live monitoring data (*coverage update*)
Risk Assessment Engine (cont.)

- How is it calculated?
  - Calculates
    - \( \text{risk}_{\text{metric}} = \frac{|\{ (X, Y) | (X, Y) \in \text{win} \land (X, Y) \in \text{unseen_combos} \}|}{|\text{unseen_combos}|} \)

    \( \text{unseen_combos} = \{ (X, Y) | Y_{\text{VM}, \text{PID} = \text{trg}, \text{PID}} \} \cup \text{unseen_combos} \)

    - \( \text{risk}_{\text{migration}} = \frac{|\{ (X, Y) | Y_{\text{VM}, \text{PID} = \text{trg}, \text{PID}} \land (X, Y) \in \text{unseen_combos} \}|}{|\text{unseen_combos}|} \)

    \( \nu_1 = |\{ (X, Y) | X \in \text{orig}_{\text{VM}} \land Y \in \text{clone}_{\text{VM}} \}| \)

    - \( \nu_2 = |\{ (X, Y) | X \in \text{clone}_{\text{VM}} \land (X, Y) \in \text{old_model} \land (X, Y) \in \text{unseen_combos} \}| \)

    \( \text{risk}_{\text{scale_out}} = \frac{\nu_1 + \nu_2}{\nu_1 + \nu_2 + |\text{unseen_combos}|} \)

Risk values are taken as part of the world which tests did not cover

Using a BDD-based symbolic computation for scaling
Preliminary Results

- Tested a TelCo application on top of CloudWave
- Risk metric triggering adaptation