Search-Based Genetic Optimization for Deployment and Reconfiguration of Software in the Cloud

35th International Conference on Software Engineering (ICSE 2013)
San Francisco, CA, USA

Sören Frey, Florian Fittkau, and Wilhelm Hasselbring

Software Engineering Group
Dept. Computer Science
Kiel University, Kiel, Germany

May 23, 2013
“Deploying and migrating software applications to the cloud involves many combinations of options that vary widely in their characteristics and performances, from different combinations of CPU, memory, storage, and network options to IT resource management services, to algorithms that can perform dynamic resource scaling. Developing software systems and choosing the most suitable cloud deployment option using these heterogeneous resources is both nontrivial and difficult [...]”

⇒ Impractical to incorporate all cloud environments in an actual comparison.

[Grundy et al., 2012]
Motivation

On premise deployment
(Status quo deployment)
Simple Cloud Deployment Option Example
(No reconfiguration rules, for simplicity)

Motivation

On premise deployment (Status quo deployment)

Cloud Deployment Options (CDOs):
Cloud environment to use

Cloud Environment?

- Amazon EC2
- BitRefinery
- CloudSigma
- Enomaly
- GoGrid
- Hosting.com
- HP Cloud Compute
- IBM SmartCloud
- Joyent
- Lunacloud
- NephoScale
- OpSource
- Rackspace
- ReliaCloud
- Softlayer
- Terremark Enterprise Cloud
Simple Cloud Deployment Option Example
(No reconfiguration rules, for simplicity)

Motivation

On premise deployment
(Status quo deployment)

Cloud Deployment Options (CDOs):
By chance

Cloud Environment?
- Amazon EC2
- BitRefinery
- CloudSigma
- Enomaly
- GoGrid
- Hosting.com
- HP Cloud Compute
- IBM SmartCloud
- Joyent
- Lunacloud
- NephoScale
- OpSource
- Rackspace
- ReliaCloud
- Softlayer
- Terremark Enterprise Cloud
...
Simple Cloud Deployment Option Example
(No reconfiguration rules, for simplicity)

Motivation

On premise deployment (Status quo deployment)

Cloud Deployment Options (CDOs):
Number of VM instances
Simple Cloud Deployment Option Example
(No reconfiguration rules, for simplicity)

Motivation

On premise deployment
(Status quo deployment)

Cloud Deployment Options (CDOs):
VM instance types to use
Simple Cloud Deployment Option Example
(No reconfiguration rules, for simplicity)

Motivation

On premise deployment
(Status quo deployment)

Cloud Deployment Options (CDOs):
Mapping of components to VM images
So, which Cloud Deployment Option (CDO) to use?

A **CDO** defines:

- Cloud environment to use
- Number of VM instances to start
- VM instance types to use
- Mapping of software components to VM images
- Runtime reconfiguration rules
Huge Variety of Cloud Deployment Options

Motivation

So, which Cloud Deployment Option (CDO) to use?

A CDO defines:

- Cloud environment to use
- Number of VM instances to start
- VM instance types to use
- Mapping of software components to VM images
- Runtime reconfiguration rules

Example runtime reconfiguration rule:

Start $R_1 = 2$ new VM instances of VM instance type $R_2 = m1.medium$ if average CPU utilization $\geq R_3 = 80\%$ for $R_4 = 5$ minutes.
Huge Variety of Cloud Deployment Options

Motivation

So, which Cloud Deployment Option (CDO) to use?

A CDO defines:

- Cloud environment to use
- Number of VM instances to start
- VM instance types to use
- Mapping of software components to VM images
- Runtime reconfiguration rules

Example runtime reconfiguration rule:

Start $R_1 = 2$ new VM instances of VM instance type $R_2 = m1.medium$ if average CPU utilization $\geq R_3 = 80\%$ for $R_4 = 5$ minutes.

Which are the best suited values for $R_1$, $R_2$, $R_3$, and $R_4$?
Outline

1 Motivation

2 Project Context

3 CDOXplorer

4 Experiments

5 Conclusion
The CloudMIG Method
[Frey and Hasselbring, 2011]

Project Context

Legend:
A1: Extraction
A2: Selection
A3: Generation
A4: Adaptation
A5: Evaluation
A6: Transformation

Optional
Mandatory

Legend:
A1: Extraction
A2: Selection
A3: Generation
A4: Adaptation
A5: Evaluation
A6: Transformation

Optional
Mandatory
• CDOXplorer and CDOSim: Part of our cloud migration approach CloudMIG

• Tool support for CloudMIG: Integration of CDOSim in CloudMIG Xpress (based on [Fittkau et al., 2012a])
Simulating Cloud Deployment Options (CDOs)
CDOSim [Fittkau et al., 2012a,b]

Project Context

• Our simulator CDOSim can simulate CDOs and identify costs, response times, and #SLA violations

• Example from previous case study:
  • JPetStore deployed to Amazon EC2 and private Eucalyptus cloud
  • Start new VM instances when CPU utilization > 70% for at least one minute
  • Shut down VM instance when CPU utilization < 30% for at least one minute
Simulating Cloud Deployment Options (CDOs)

CDOSim [Fittkau et al., 2012a,b]

Project Context

• Our simulator CDOSim can simulate CDOs and identify costs, response times, and #SLA violations

• Example from previous case study:
  • JPetStore deployed to Amazon EC2 and private Eucalyptus cloud
  • Start new VM instances when CPU utilization > 70% for at least one minute
  • Shut down VM instance when CPU utilization < 30% for at least one minute

Figure: Simulated CPU utilization
Simulating Cloud Deployment Options (CDOs)

CDOSim [Fittkau et al., 2012a,b]

Project Context

- Our simulator CDOSim can simulate CDOs and identify costs, response times, and #SLA violations
- Example from previous case study:
  - JPetStore deployed to Amazon EC2 and private Eucalyptus cloud
  - Start new VM instances when CPU utilization > 70% for at least one minute
  - Shut down VM instance when CPU utilization < 30% for at least one minute

⇒ High precision
⇒ But: Huge number of possible CDOs
⇒ Simulating all CDOs (sequentially) would last for years
1 Motivation

2 Project Context

3 CDOXplorer

4 Experiments

5 Conclusion
CDO Optimization Approach

Goal:

- Support comparison of cloud deployment options
- Find suitable trade-offs between cost, response times, and SLA violations (number of method timeouts)
Goal:
- Support comparison of cloud deployment options
- Find suitable trade-offs between cost, response times, and SLA violations (number of method timeouts)

Challenges:
- Optimization problem with multiple objectives
- SLA-aware service deployment optimization is NP-hard [Canfora et al., 2005]
- Complex and non-linear correlations of input parameters
CDO Optimization Approach

Goal:

- Support comparison of cloud deployment options
- Find suitable trade-offs between cost, response times, and SLA violations (number of method timeouts)

Challenges:

- Optimization problem with multiple objectives
- SLA-aware service deployment optimization is NP-hard [Canfora et al., 2005]
- Complex and non-linear correlations of input parameters

Approach:

- Explore design space with a genetic algorithm called CDOXplorer
- Use our simulator CDOSim to compute fitness of CDOs (⇒ simulation-based optimization)
CDOXplorer Overview

- IaaS cloud environments (basic building blocks: virtual machines)
- Usage of actual workload data (e.g., from monitoring log files)
- Optimizes runtime reconfiguration rules at design time (migration planning)
CDOXplorer Overview

- IaaS cloud environments (basic building blocks: virtual machines)
- Usage of actual workload data (e.g., from monitoring log files)
- Optimizes runtime reconfiguration rules at design time (migration planning)

Input and Output:
Scaling Types

CDOXplorer

scale out/ scale in

scale up/ scale down

S. Frey, F. Fittkau, and W. Hasselbring
Optimizing the Deployment of Software in the Cloud
May 23, 2013 10 / 20
Cloud Deployment Options

Structure of a Cloud Deployment Option (Phenotype)

Constraints:
1. If a Grow Rule exists, there has to be a Shrink Rule and vice versa
2. If Grow Action is "scaleUp", Shrink Action has to be "scaleDown"
3. If Grow Action is "scaleOut", Shrink Action has to be "scaleIn"
Structure of a Cloud Deployment Option (Genotype)

1 Cloud Environment

+ Node Configuration

1 Service Composition
1 Initial Start Config

? Grow Rule
? Shrink Rule

1 CE 1 SE 1 IT 1 NI 1 GA 1 G1 1 G2 1 G3 1 G4 1 G5 1 SA 1 S1 1 S2 1 S3 1 S4 1 S5

Crossover Points

CP1  CP2  CP3  CP4

Further
Node Configurations
### Gene Design

#### CDOXplorer

---

**Gene Range Description Chromosome**

<table>
<thead>
<tr>
<th>Gene</th>
<th>Range</th>
<th>Description</th>
<th>Chromosome</th>
</tr>
</thead>
<tbody>
<tr>
<td>CE</td>
<td>N</td>
<td>Cloud environment id</td>
<td>Cloud Environment</td>
</tr>
<tr>
<td>SE</td>
<td>N</td>
<td>Service id</td>
<td>Service Composition</td>
</tr>
<tr>
<td>IT</td>
<td>N</td>
<td>VM Instance type id</td>
<td>Initial Start Configuration</td>
</tr>
<tr>
<td>NI</td>
<td>N</td>
<td>Nr. of VM instances to start initially</td>
<td>Initial Start Configuration</td>
</tr>
<tr>
<td>GA</td>
<td>0,1</td>
<td>Grow action; 0: scale up, 1: scale out</td>
<td>Grow Rule</td>
</tr>
<tr>
<td>G1</td>
<td>N</td>
<td>Minimum nr. of VM instances</td>
<td>Grow Rule</td>
</tr>
<tr>
<td>G2</td>
<td>1.1-3.0</td>
<td>MIPIPS multiple in steps of 0.1</td>
<td>Grow Rule</td>
</tr>
<tr>
<td>G3</td>
<td>0,1</td>
<td>Condition scope; 0: single VM, 1: all VMs</td>
<td>Grow Rule</td>
</tr>
<tr>
<td>G4</td>
<td>0.05-1.0</td>
<td>Condition median utilization in steps of 0.05</td>
<td>Grow Rule</td>
</tr>
<tr>
<td>G5</td>
<td>5-60</td>
<td>Condition time period in steps of 5 minutes</td>
<td>Grow Rule</td>
</tr>
<tr>
<td>SA</td>
<td>0,1</td>
<td>Shrink action; 0: scale down, 1: scale in</td>
<td>Shrink Rule</td>
</tr>
<tr>
<td>S1</td>
<td>N</td>
<td>Minimum nr. of VM instances</td>
<td>Shrink Rule</td>
</tr>
<tr>
<td>S2</td>
<td>0.1-0.9</td>
<td>MIPIPS multiple in steps of 0.1</td>
<td>Shrink Rule</td>
</tr>
<tr>
<td>S3</td>
<td>0,1</td>
<td>Condition scope; 0: single VM, 1: all VMs</td>
<td>Shrink Rule</td>
</tr>
<tr>
<td>S4</td>
<td>0.0-0.95</td>
<td>Condition median utilization in steps of 0.05</td>
<td>Shrink Rule</td>
</tr>
<tr>
<td>S5</td>
<td>5-60</td>
<td>Condition time period in steps of 5 minutes</td>
<td>Shrink Rule</td>
</tr>
</tbody>
</table>

---

S. Frey, F. Fittkau, and W. Hasselbring

Optimizing the Deployment of Software in the Cloud

May 23, 2013 13 / 20
Cloud Deployment Options - Examples

- Cloud Environment: Amazon EC2

Reconfiguration grow rule:
- Grow action: Scale-up
- Grow rule, min. nr. VM instances: 1
- Grow MIPIPS multiple: 2.3
- Grow condition scope: Single VM
- Grow when median CPU utilization > 80% for 20 minutes

Reconfiguration shrink rule:
- Shrink action: Scale-down
- Shrink rule, min nr. VM instances: 1
- Shrink MIPIPS multiple: 0.5
- Shrink condition scope: All VMs
- Shrink when median CPU utilization < 40% for 45 minutes
Cloud Deployment Options - Examples

- Cloud Environment: Amazon EC2

Reconfiguration grow rule:
- Grow action: Scale-up
- Grow rule, min. nr. VM instances: 1
- Grow MIPS multiple: 2.3
- Grow condition scope: Single VM
- Grow when median CPU utilization > 80% for 20 minutes

Reconfiguration shrink rule:
- Shrink action: Scale-down
- Shrink rule, min nr. VM instances: 1
- Shrink MIPS multiple: 0.5
- Shrink condition scope: All VMs
- Shrink when median CPU utilization < 40% for 45 minutes
4 sub crossover operators corresponding to 4 crossover points

<table>
<thead>
<tr>
<th>Crossover point</th>
<th>Sub operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CP1</td>
<td>CE</td>
<td>Swap cloud environments</td>
</tr>
<tr>
<td>CP2</td>
<td>CI</td>
<td>Swap initial start configurations</td>
</tr>
<tr>
<td>CP3</td>
<td>CG</td>
<td>Swap grow rules</td>
</tr>
<tr>
<td>CP4</td>
<td>CS</td>
<td>Swap shrink rules</td>
</tr>
</tbody>
</table>
### Crossover Operators

4 sub crossover operators corresponding to 4 crossover points

<table>
<thead>
<tr>
<th>Crossover point</th>
<th>Sub operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CP1</td>
<td>CE</td>
<td>Swap cloud environments</td>
</tr>
<tr>
<td>CP2</td>
<td>CI</td>
<td>Swap initial start configs</td>
</tr>
<tr>
<td>CP3</td>
<td>CG</td>
<td>Swap grow rules</td>
</tr>
<tr>
<td>CP4</td>
<td>CS</td>
<td>Swap shrink rules</td>
</tr>
</tbody>
</table>

**Example: CI**

![Diagram of CDO1, CDO2, and Offspring with crossover points and operators]
Case Study: Apache OfBiz

Experimental Setting:

- Single-cloud scenario ($SC_S$): Amazon EC2
- Multi-cloud scenario ($SC_M$): Amazon EC2, Microsoft Windows Azure, Eucalyptus
- Comparing CDOXplorer with three alternative search methods:
  - SI-RS: Simple random sampling
  - SY-RS: Systematic random sampling
  - SI-AN: Simulated annealing

Metrics Hypervolume ($HV$) and Inverted Generational Distance ($IGD$):
Baseline (best-known) pareto-optimal front for $SC_S$: 

![Diagram showing the baseline (best-known) pareto-optimal front for $SC_S$. The diagram plots SLA violations against cost and response times. The x-axis represents cost in dollars, the y-axis represents SLA violations, and the z-axis represents response times in milliseconds. The data points are distributed across the plot, indicating the trade-offs between cost and SLA violations.]
Baseline (best-known) pareto-optimal front for $SC_S$: 

![Graph showing baseline pareto-optimal front for SC_S]
Exemplary results for $SC_S$:

<table>
<thead>
<tr>
<th>Metric</th>
<th>CDOXplorer</th>
<th>SI-RS</th>
<th>SY-RS</th>
<th>SI-AN</th>
</tr>
</thead>
<tbody>
<tr>
<td>I.G. Distance</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>2.70E-02</td>
<td>3.67E-02</td>
<td>4.11E-02</td>
<td>3.28E-02</td>
</tr>
<tr>
<td>SD</td>
<td>2.10E-03</td>
<td>2.13E-03</td>
<td>3.61E-03</td>
<td>2.85E-03</td>
</tr>
<tr>
<td>Median</td>
<td>2.72E-02</td>
<td>3.65E-02</td>
<td>4.21E-02</td>
<td>3.20E-02</td>
</tr>
<tr>
<td>Min (best)</td>
<td>2.16E-02</td>
<td>3.34E-02</td>
<td>3.40E-02</td>
<td>2.76E-02</td>
</tr>
<tr>
<td>Max (worst)</td>
<td>3.03E-02</td>
<td>4.07E-02</td>
<td>4.83E-02</td>
<td>3.95E-02</td>
</tr>
<tr>
<td>Hypervolume</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>4.48E-01</td>
<td>4.41E-01</td>
<td>4.41E-01</td>
<td>4.44E-01</td>
</tr>
<tr>
<td>SD</td>
<td>2.08E-03</td>
<td>1.96E-03</td>
<td>2.89E-03</td>
<td>2.09E-03</td>
</tr>
<tr>
<td>Median</td>
<td>4.48E-01</td>
<td>4.40E-01</td>
<td>4.41E-01</td>
<td>4.44E-01</td>
</tr>
<tr>
<td>Min (worst)</td>
<td>4.44E-01</td>
<td>4.36E-01</td>
<td>4.35E-01</td>
<td>4.40E-01</td>
</tr>
<tr>
<td>Max (best)</td>
<td>4.54E-01</td>
<td>4.46E-01</td>
<td>4.46E-01</td>
<td>4.48E-01</td>
</tr>
</tbody>
</table>
CDOXplorer advantage relative to other approaches:
1 Motivation

2 Project Context

3 CDOXplorer

4 Experiments

5 Conclusion
Conclusion

- **Context:**
  - CDOs can be simulated with our simulator CDOSim (estimation of future costs, response times, and SLA violations)
  - There exists a vast amount of CDOs, simulating all (sequentially) lasts *thousands of years*

- **Approach:**
  - Genetic algorithm CDOXplorer allows to find best-suited CDOs (pareto optimal set)
  - Part of cloud migration approach CloudMIG
  - CDOSim constitutes fitness function (simulation-based optimization)

- **Tool support:**
  - CloudMIG Xpress
  - [http://www.cloudmig.org](http://www.cloudmig.org)


