Automated Planning for Configuration Changes

Herry
Paul Anderson
Gerhard Wickler

LISA Conference, December 2011
Boston, US
Outline

• Declarative Approach
• Example: Configuration Problem
  – Solution: Declarative Tool
  – Solution: Our Prototype
• Experiment: Cloud-Burst Problem
  – Demo
• Conclusions
Declarative Approach

• Most commonly used today
• Popular tools: Puppet, Cfengine, Chef, LCFG
• Critical shortcomings
  – Indeterminate order executions of actions
  – Could violates the system’s constraints
Example: Configuration Problem

Current State

Desired State

Constraint:
C must always refer to a running server!
Solution: Declarative Tools

Desired State
- A.running = false
- B.running = true
- C.service = B

Puppet
Cfengine
LCFG
Submit
Implement

Possible sequences of states
1) A.running = false  C.service = B  B.running = true  X
2) C.service = B  A.running = false  B.running = true  X
3) B.running = true  A.running = false  C.service = B  X
4) A.running = false  B.running = true  C.service = B  X
5) C.service = B  B.running = true  A.running = false  X
6) B.running = true  C.service = B  A.running = false  ✓

Highly likely producing the wrong sequence!
Solution: Our Prototype

- All actions must be orchestrated as a workflow to
  - achieve the desired state
  - satisfy the constraints
- Method – using Automated Planning technique

Declarative approach:  

Our Prototype:  

$pre$: preconditions  
$eff$: effects
Solution: Our Prototype (2)

Desired State
- A.running = false
- B.running = true
- C.service = B

Global Constraint
- C.service.running = true

Current State
- A.running = true
- B.running = false
- C.service = A

Actions

<table>
<thead>
<tr>
<th>pre</th>
<th>Action</th>
<th>eff</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>start ( server )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>stop ( server )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>change ( s1, s2, c )</td>
<td></td>
</tr>
</tbody>
</table>
Solution: Our Prototype (3)

Desired State + Constraints

Current State

Actions Database

Planner

Workflow

Execution Agents (ControlTier and Puppet)

Managed Servers
Experiment: Cloud-Burst Problem

• Cloud-Burst
  – Migrate application from private to public cloud
  – Address spikes in demand

• Constraints
  – No down-time
  – Reconfigure the firewall
  – Full migration but not duplication
Experiment: Cloud-Burst Problem

Company’s LAN

Private Cloud

WS-A
VM-X

WS-B
VM-Y

Public

Cloud Provider

WS-A
VM-X

WS-B
VM-Y

Company’s LAN

Actions?

Administrator

Running

Stopping

Firewall

Internet
Demo

• http://goo.gl/Qph7F
• Cloud-Burst problem
Conclusions

• Our prototype
  – Automatically generate the workflow between any two states
  – Achieve the desired state
  – Preserving system’s constraints
  – Enable autonomic reconfiguration
Acknowledgement

• This research is fully supported by a grant from 2010 HP Labs Innovation Research Program (IRP) award
Thank you!