Automated Planning for Configuration Changes

Herry (h.herry@sms.ed.ac.uk)  Paul Anderson (dcspaul@ed.ac.uk)  Gerhard Wickler (g.wickler@ed.ac.uk)

Service Reference Problem

Current State
- DNS-A: run = true, dns-server = dns-a
- DNS-B: run = false

Desired State
- DNS-A: run = false, dns-server = dns-a
- DNS-B: run = true

Using Puppet, Cfengine, or LCFG?

Using Our Prototype?

Solution for Service Reference Problem

Current State
- dns-a.run = true
- dns-b.run = false
- client.dnsserver = dns-a

Desired State
- dns-a.run = false
- dns-b.run = true
- client.dnsserver = dns-b

Actions
- start(dns-b)
- change(dns-a,dns-b,client)
- stop(dns-a)

Using Puppet, Cfengine, or LCFG?

Using Our Prototype?

Solution for Service Reference Problem

Current State
- dns-a.run = true
- dns-b.run = false
- client.dnsserver = dns-a

Desired State
- dns-a.run = false
- dns-b.run = true
- client.dnsserver = dns-b

Actions
- start(dns-b)
- change(dns-a,dns-b,client)
- stop(dns-a)

Automated Planning Process

Generated Workflow

Solution for Cloud-Burst Problem

Current State
- vm-a.cloud = priv-cloud (run = true)
- ws-a.cloud = pub-cloud (firewall = fw)
- ws-a.fport = 8080
- pc.service = ws-a
- vm-b.cloud = priv-cloud (run = false)

Desired State
- vm-a.cloud = pub-cloud (run = true)
- ws-a.cloud = pub-cloud (firewall = fw)
- ws-a.fport = 8080
- pc.service = ws-a
- vm-b.cloud = priv-cloud (run = true)

Generated Workflow

Our Prototype:
- can automatically generate a workflow between any two declarative states
- enables unattended, autonomic reconfiguration for failure recovery or other reasons
- can achieve the desired state as well as preserving system constraints during reconfiguration
- shows that it is possible to build the practical tool using production-quality tools for the deployment

Future Works
- Investigating more distributed and localised approaches to improve the system’s resilience

Reference