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

Administrator

Actions?

Desired State

A   B

C

A   B

C
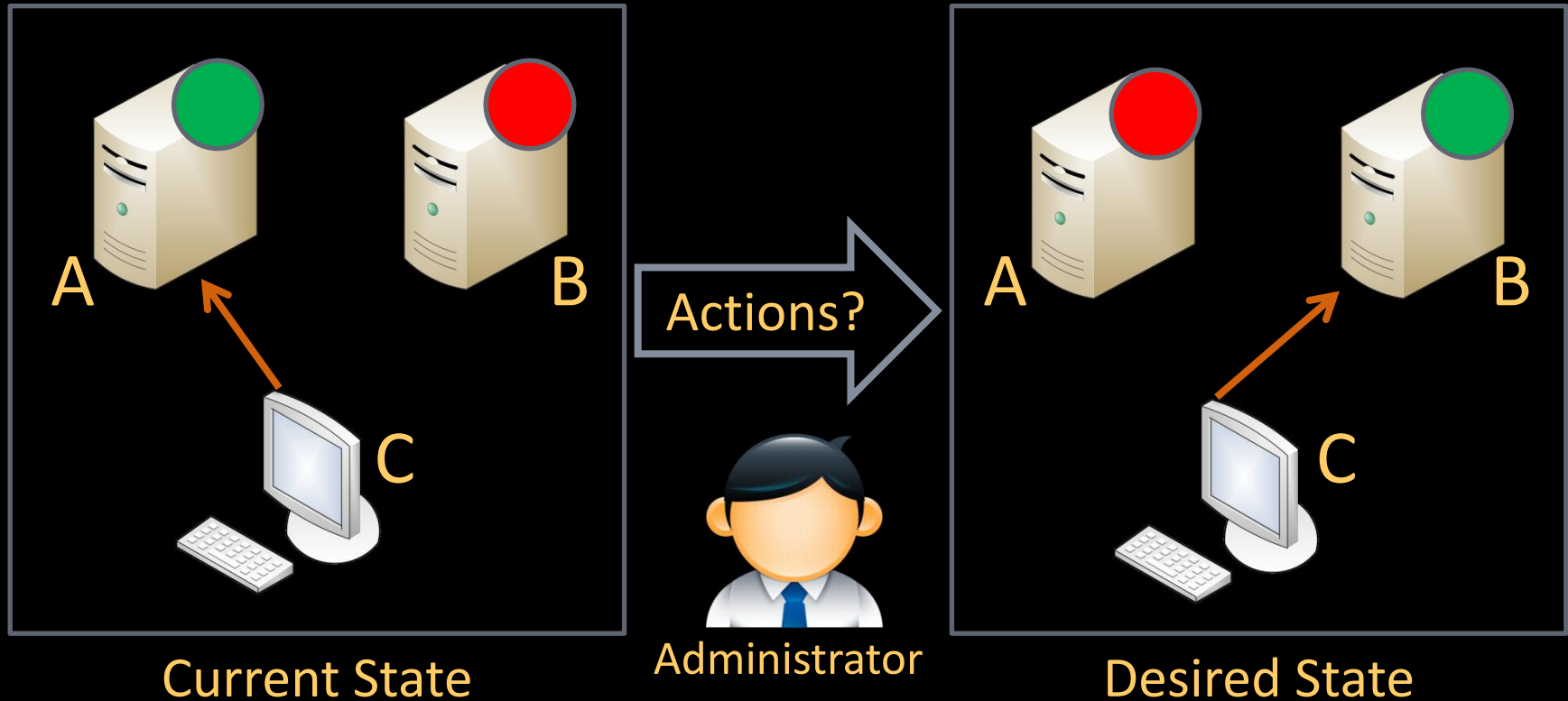
Constraint:
C must always refer to a running server!

Running   Stopping

Solution: Declarative Tools

Administrator

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

Submit →

Puppet
Cfengine
LCFG

↓ 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:

action

Our Prototype:

pre	action	*eff*

pre: preconditions
eff: effects

Solution: Our Prototype (2)

Administrator — Define →

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

Global Constraint
- C.service.running = true

Monitoring Agent — Retrieve →

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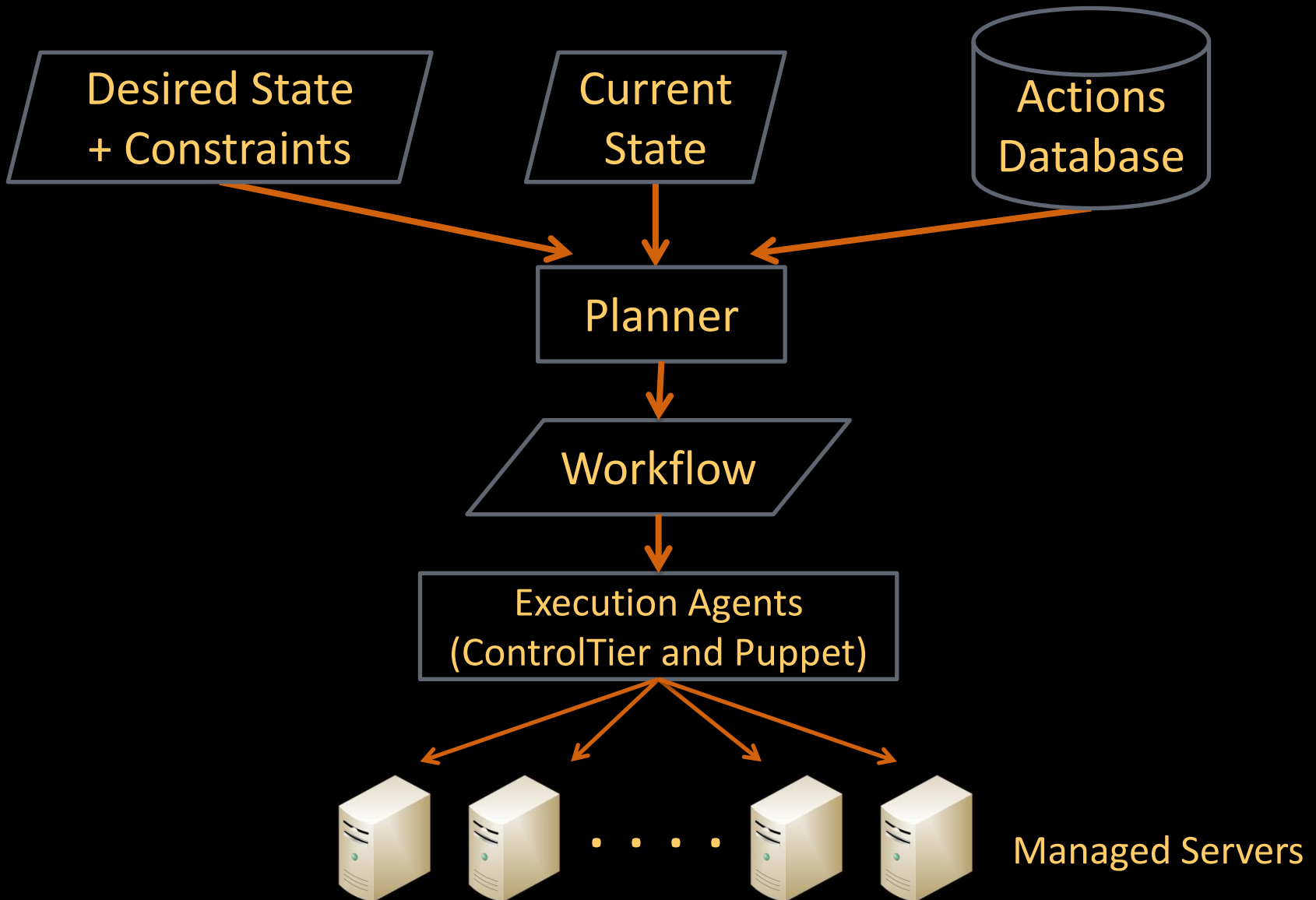- A.running = true
- B.running = false
- C.service = A

Experts, Engineers — Define →

Actions

pre	start (server)	*eff*
pre	stop (server)	*eff*
pre	change (s1, s2, c)	*eff*

Solution: Our Prototype (3)

```
Desired State          Current              Actions
+ Constraints           State              Database
       │                  │                   │
       └────────────┐     │     ┌─────────────┘
                    ▼     ▼     ▼
                  ┌───────────────┐
                  │    Planner     │
                  └───────────────┘
                          │
                          ▼
                   ╱──────────────╲
                  ╱    Workflow     ╱
                 ╱──────────────╱
                          │
                          ▼
            ┌───────────────────────────┐
            │     Execution Agents       │
            │  (ControlTier and Puppet)  │
            └───────────────────────────┘
             ╱       │        │        ╲
            ▼        ▼        ▼         ▼
```

. . . .   Managed Servers
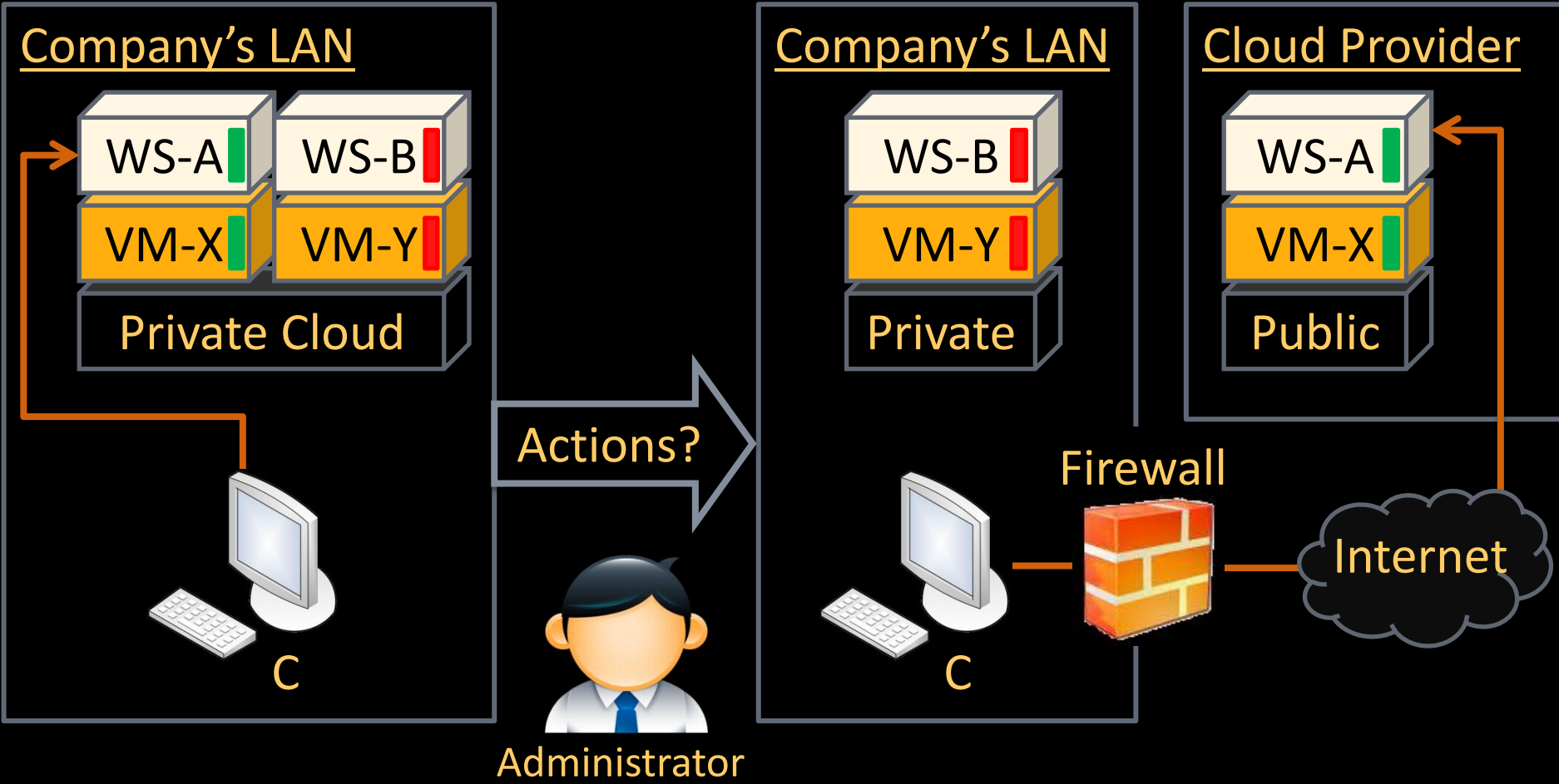
Experiment: Cloud-Burst Problem

- 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

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!