Asynchronous Real-time Monitoring with MCollective

Jeff McCune - @0xEFF
A little about me

• Sysadmin for 10 years
• Puppet contributor from the beginning
• Puppet Labs road warrior since May 2010
• Co-author of Pro-Puppet
  (Ask me if you’d like a copy)
What is MCollective?

- It’s a framework...
- To make Remote Procedure Calls...
- Asynchronously...
- With metadata (No hostnames!)...
- Collating the results for you.
- And it’s fully pluggable (Ruby)
The middleware is usually ActiveMQ or RabbitMQ.
MCollective (con’t)

• MCollective is written by:
  R.I. Pienaar - @ripienaar

http://www.devco.net
Volcane on irc.freenode.net

• This presentation demonstrates R.I.’s work on using MCollective with NRPE
How does Puppet Fit?

• For this talk, it doesn’t really. =(  

• MCollective was designed to drive Puppet  

• Manage resources on-demand  
  e.g. packages and services  

• Kick off Puppet Runs on-demand  
  With concurrency < N
The Problem

- Something bad happens on $N > 2$ nodes.
- The pager goes: Ding.
- And then: Ding.
- And: Ding.
- Ding. Ding. Ding. Ding. Ding... =(
The Solution

• Something bad happens on \( N > 2 \) nodes.

• Ding.

• ... Ahhhhhhh ...
Demo

$ mco nrpe check_puppet_run

Finished processing 2 / 2 hosts in 117.42 ms

OK: 2
WARNING: 0
CRITICAL: 0
UNKNOWN: 0
What you just saw

- Nodes are *Discovered* (~ 2 seconds)
- Discovered nodes are sent a message
- Agents respond to the message (~ 500ms)
- The results are collated and presented (~ 2.5 seconds total)
Nagios Integration

• Goal: Single executable NRPE plugin

• Handles for us:
  • Discovery of online nodes
  • RPC nrpe call to online nodes
  • Collation of results

• Outputs NRPE compatible OK or FAIL
Demo

$ check-mc-nrpe check_puppet_run

check_puppet_run: OK: 2 \nWARNING: 1 \nCRITICAL: 0 \nUNKNOWN: 0| \ntotal=3 ok=2 warn=1 crit=0 unknown=0\nchecktime=0.172370
WARNING: 
  www21
The Second Problem

- If a node is NOT online...
- Then MCollective can’t discover it...
- And we can’t monitor it
Registration

- Nodes can periodically send a registration message.
- Monitors record registered nodes.
- Monitors alert if a registered node hasn’t registered recently.
- Registration agents aren’t “special.” (Fully pluggable normal agents)
Demo

```bash
# check-mc-nrpe -W monitor check_mcollective

check_mcollective_fast: OK: 0 \ WARNING: 0 \ CRITICAL: 1 \ UNKNOWN: 0| \ total=1 ok=0 warn=0 crit=1 unknown=0 \ checktime=0.114291
CRITICAL:
    monitor101
```
Demo

$ /usr/lib/nagios/plugins/check_mcollective.rb \\ 
  --directory /var/tmp/mcollective \\ 
  --interval 300

CRITICAL: \ 
1 / 3 hosts not checked in within 3 seconds| \ 
totalhosts=3 oldhosts=1 currenthosts=2
What you just saw

• When a node drops out of the collective.
• The monitors notice this after a short period of time.
• Because registration messages are no longer being sent by the node.
• A single alert is generated.
Re-cap (MCollective)

- MCollective is an RPC framework
- Nodes are *Discovered* automatically (No hostname spreadsheets!)
- *Agents* execute RPC messages
- Results are collated and presented
- Easy to script and plug in your own agents
Re-cap (Monitoring)

- N alerts for N nodes is incredibly annoying.
- The nrpe agent executes any NRPE check.
- Offline nodes are *not* checked.
- Registration allows you to alert when nodes are offline.
- *One* alert for N nodes in trouble. =)
• http://www.devco.net

• http://puppetlabs.com/mcollective

• github.com/puppetlabs/mcollective-plugins

• All of the code I’ve demoed was written by R.I.P. and published as OSS on github