

# Introduction To Marrionette Collective

# MCollective

...a framework to build server orchestration or parallel job execution systems...

# What is MCollective

## toolset

actions => manage/command/control...

objects => services/packages/processes/file...

aliases => fabric/func/capistrano

## framework

modular => pluggable core meet your own needs

simpleRPC => write your own agents and clients

# What does it allow

- system discovery
  - inventory collective
  - task execution
  - configuration management
  - command & control
  - else...
- device reports ...

```
[root@tvn ~]# mco ping
-----ping to 55.101.101.101----- time=57.26 ms
-----ping to 55.101.101.101----- time=96.17 ms
-----ping to 55.101.101.101----- time=96.77 ms

---- ping statistics ----
3 replies max: 96.77 min: 57.26 avg: 83.40
```



```
[root@tvn ~]# mco rpc service status service=httpd
```

```
Determining the amount of hosts matching filter for 2 seconds .... 3
```

```
* [ =====> ] 3 / 3
```

```
Service Status: stopped
```

```
Service Status: stopped
```

```
Service Status: stopped
```

```
Finished processing 3 / 3 hosts in 135.73 ms
```

```
[root@tvn ~]# mco rpc runcommand runcommand cmd="uname -n" -v
```

```
Determining the amount of hosts matching filter for 2 seconds .... 3
```

```
* [ =====> ] 3 / 3
```

```
===== : OK  
{:command=>"uname -n", :err=>"", :out=>"=====", :status=>0}
```

```
===== : OK  
{:command=>"uname -n", :err=>"", :out=>"=====", :status=>0}
```

```
===== : OK  
{:command=>"uname -n", :err=>"", :out=>"=====", :status=>0}
```

```
---- runcommand#runcommand call stats ----
```

```
Nodes: 3 / 3
```

```
Pass / Fail: 3 / 0
```

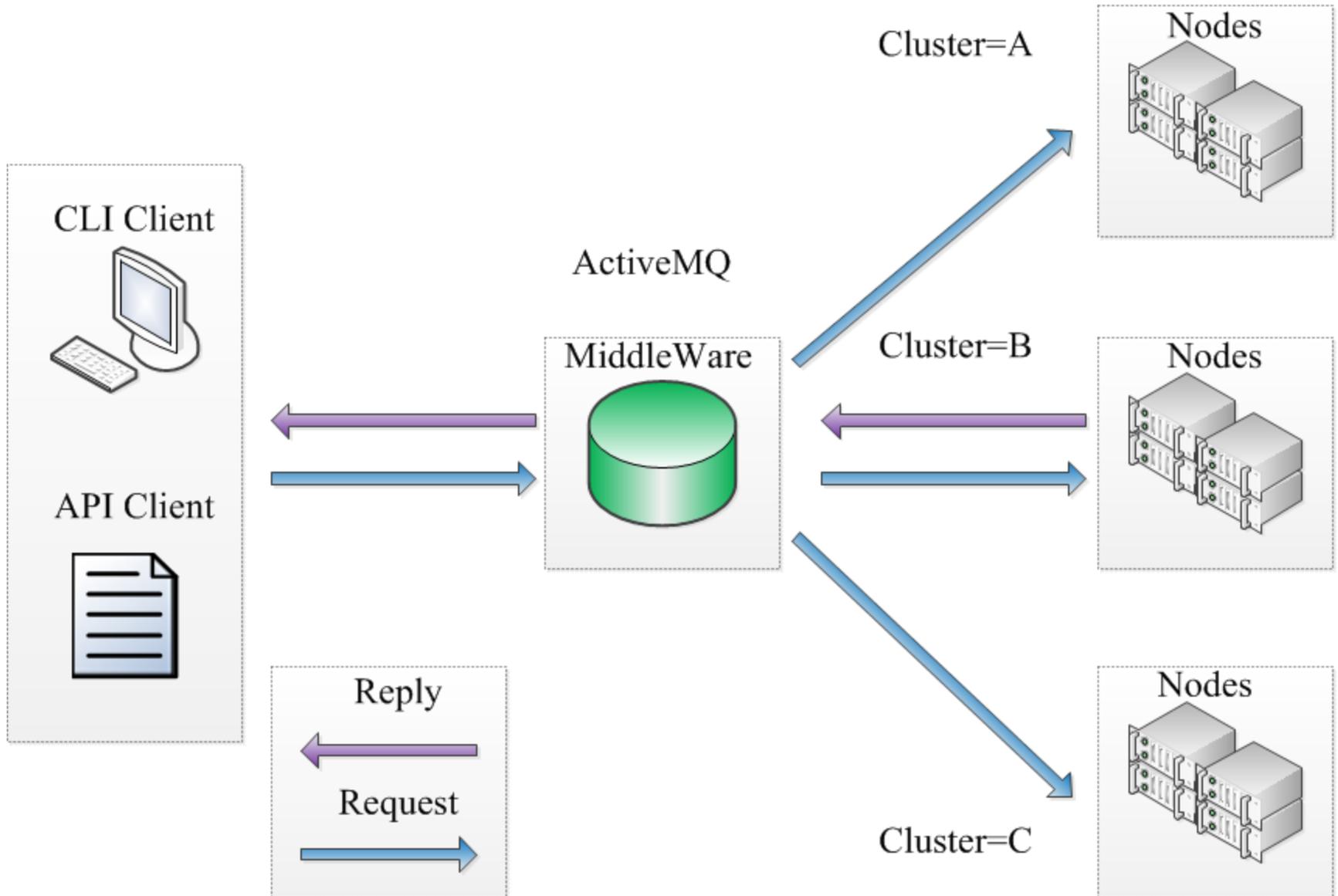
```
Start Time: Thu Aug 09 17:14:13 +0800 2012
```

```
Discovery Time: 2001.25ms
```

```
Agent Time: 117.73ms
```

```
Total Time: 2118.98ms
```

# Architecture



# Implement

## ➤ discovery mechanism

not ssh for loop, real time discovery of network resources

## ➤ Publish-Subscribe Model

ActiveMQ/RabbitMQ for Message queues

use a broadcast paradigm for request distribution (also direct addressing)

# Implement

## ➤ filter mechanism

Leverages meta-data (facter, ohai...) to Limit Scope

use facts, identities, classes, agents for filters not only  
hostname(identity)

```
$ mco rpc package yum_checkupdates --  
with-fact location=/nan/
```

```
$ mco find --with-fact location=~^nan
```

# Implement

- SimpleRPC-Based framework for parallel jobs and/or server orchestration

client applications

server agents

```
module MCollective
  module Agent
    class Printer < RPC::Agent
      metadata :name          => 'printer',
              :description => 'Printers and printing',
              :author       => 'Gary Larizza',
              :license       => 'BSD',
              :version       => '0.1',
              :url           => 'http://puppetlabs.com',
              :timeout       => 100

      action 'list' do
        reply[:output] = `lpstat -a | cut -d ' ' -f 1`.split("\n")
      end
    end
  end
end
```

# With Puppet

## Controller

puppetd/puppetca/puppetral agent

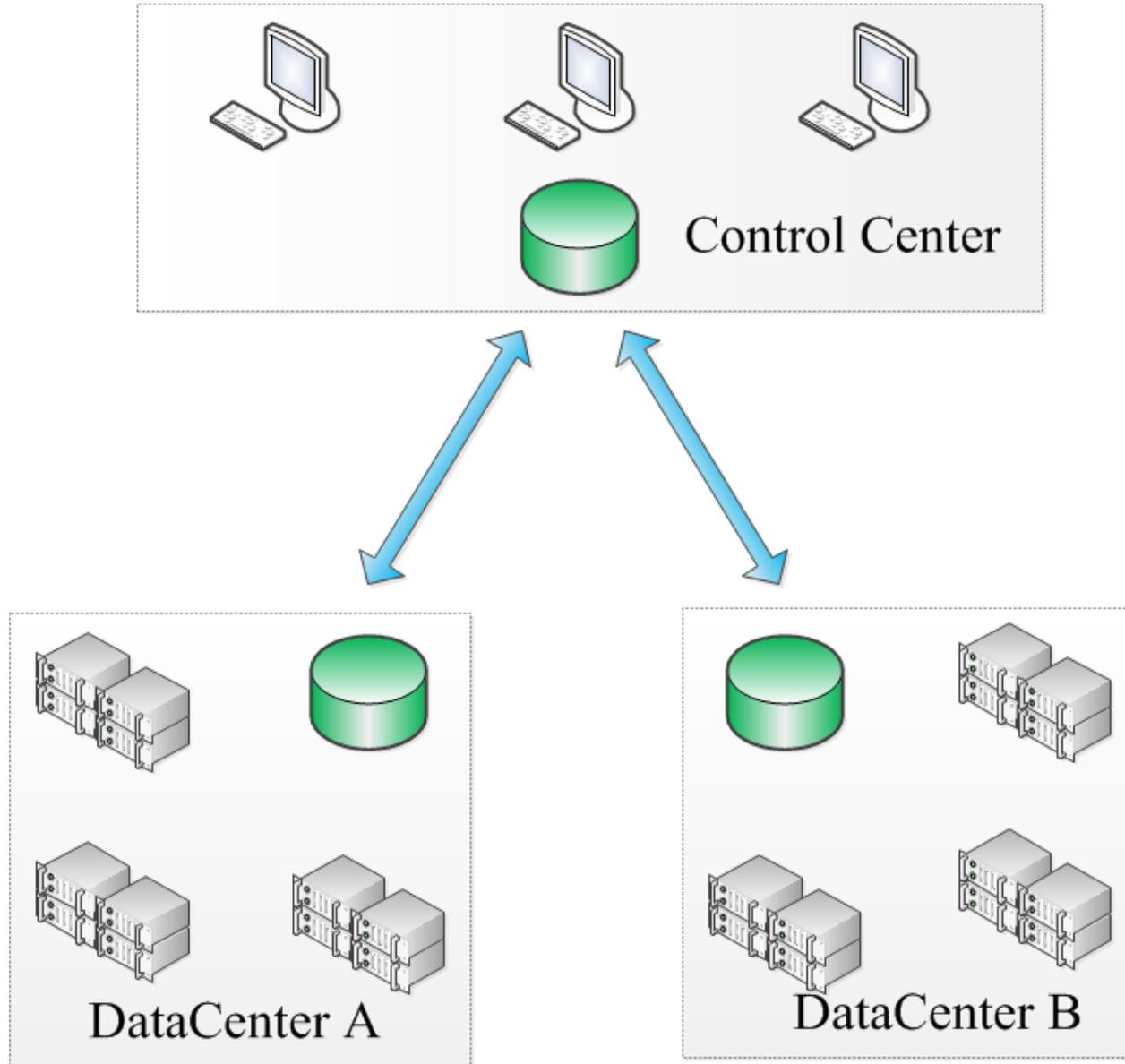
## Classifier

puppet class/puppet facter

## Scheduler

puppetcommander

# With Puppet



# puppetcommander

```
[root@tvn home]# puppetcommanderd --interval 1 --max-concurrent 2
Looping clients with an interval of 1 minutes
Restricting to 2 concurrent puppet runs
Doing discovery start of run
Discovered 3 nodes
Found 3 puppet nodes, sleeping for ~20 seconds between runs
Current puppetd's running: 0
Running agent for [REDACTED]
Sleeping for 19 seconds
Current puppetd's running: 0
Running agent for [REDACTED]
Sleeping for 19 seconds
Current puppetd's running: 0
Running agent for [REDACTED]
Sleeping for 20 seconds
Doing discovery start of run
Discovered 3 nodes
Found 3 puppet nodes, sleeping for ~20 seconds between runs
Current puppetd's running: 0
Running agent for [REDACTED]
```

# Advantages or Features

- with large scale

Interact with small to very large clusters of servers

- fast and scalable

subcollectives + activemq clusters

- asynchronous parallel

# security

## ➤ clients

AES+RSA plugins: encrypt & sign every message

SSL plugins: sign every message

TLS for connection encrypt

## ➤ middleware

topic permissions and subclusters

## ➤ agents

all clients security measures

rpc authorization and auditing

# plug-able and customizable

- plugins

custom agents

security

facts

.....

- mcollective is a platform on which we've shipped a set of rudimentary programs

# disadvantages or difficulties

➤ missing events

➤ adding by you.....