Case Study:
Puppet 3 to Puppet 4

by Ryan Whitehurst

Source: https://github.com/thrnio/puppet3-to-puppet4

Ryan Whitehurst

SysOps Engineer, Puppet Labs

Email	rw@puppetlabs.com
IRC	thrnio
Twitter	@thrnio

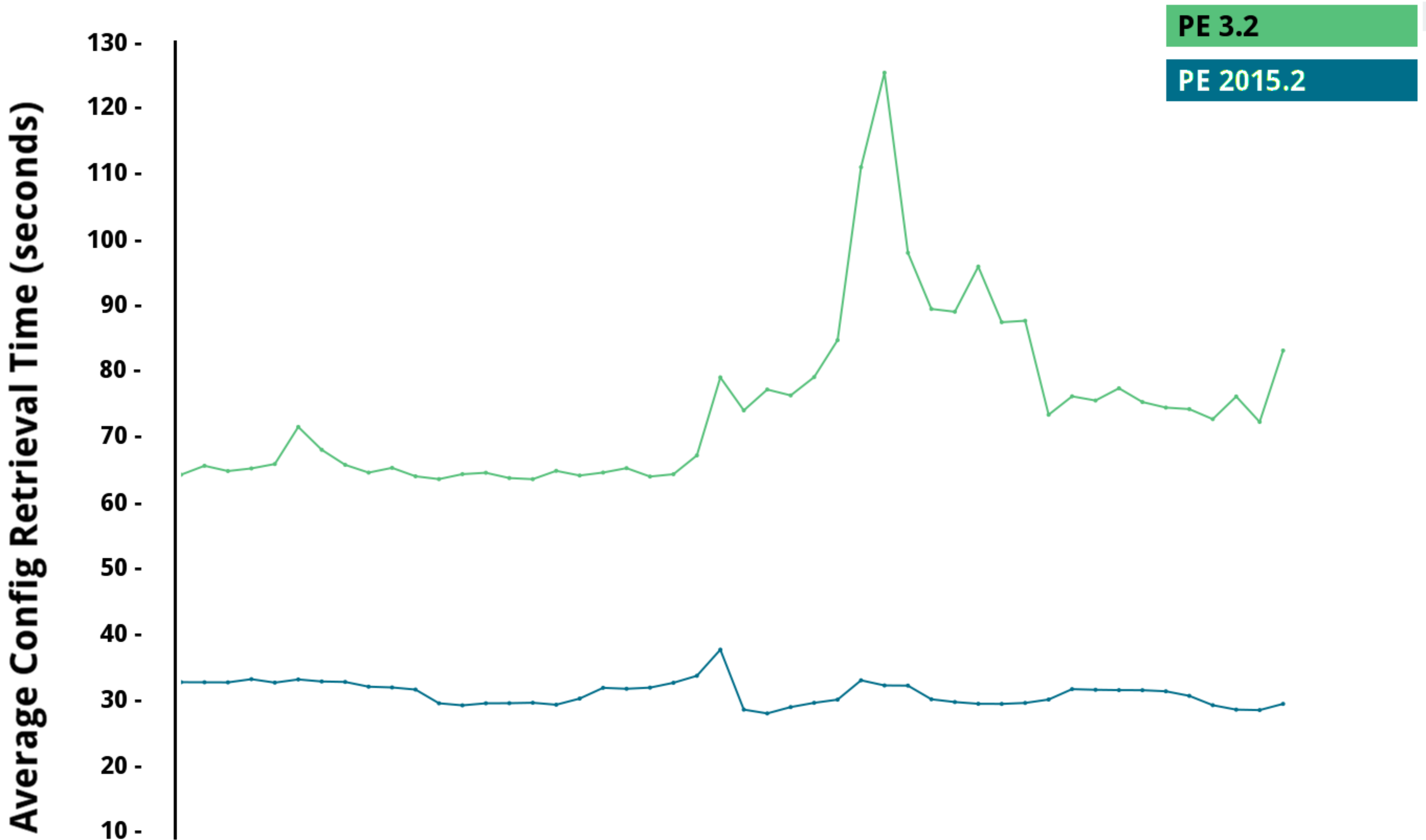
The Plan

1. Overview
2. Building a test environment
3. Testing catalogs
4. Performance
5. The migration

Our infrastructure

- ~500 nodes
- ~50 environments
- ~150 public modules
- ~40k lines of custom puppet code per environment
- ~128k total lines of puppet code per environment

Comparison of Config Retrieval Times for PE 3.2 and PE 2015.2



0 -
hourly datetime buckets

What we did

- PE 3.2 to PE 2015.2
- Puppet 3.4 to Puppet 4.2
- PuppetDB 1.x to PuppetDB 3.x
- Split install to monolithic master of masters
- Passenger to Puppet Server
- 3x parser to 4x ("future") parser
- Switch to directory environments

The Plan

1. Overview
2. Building a test environment
3. Testing catalogs
4. Performance
5. The migration

First attempts

Puppet Server crashes… sometimes

Our solution:

Automate ALL THE THINGS

...except automating PE is hard

The Puppet CA

Duplicate the Puppet CA

1. delete the SSL directory from the new MoM
2. rsync the SSL directory from the old CA to the new MoM
3. bump the next serial number by a lot:
   `/etc/puppetlabs/puppet/ca/serial`
4. reissue all puppet certificates for the new install:
   https://docs.puppetlabs.com/puppet/latest/reference/ssl_regenerate_certificates.html

Problems with CA duplication

- Inaccurate inventory
- Certificate revocation list divergence
- Non-contiguous serial numbers

Other CA options

- Switch to new CA
- Point new infrastructure at old CA
- Use a separate SSL directory for testing
- Use an external CA service

The Plan

1. Overview
2. Building a test environment
3. Testing catalogs
4. Performance
5. The migration

The easy way:

`catalog_preview`

https://forge.puppetlabs.com/puppetlabs/catalog_preview

Manually compile catalogs for all nodes

Attempt 1:

```
puppet master --compile
```

Attempt 2:

```
POST /puppet/v3/catalog
```

~~cronjob to sync facts cache~~

~~source from PuppetDB~~

Let the agent request the catalog

```
if $::osfamily != "windows" {
  cron { 'pe agent':
    ensure  => present,
    command => join(['/opt/puppet/bin/puppet agent',
                     '--no-daemonize --onetime',
                     ], ' '),
                   minute  => [fqdn_rand(25), fqdn_rand(25) + 30],
  }

  cron { 'pe agent noop run':
    ensure  => present,
    command => join(['/opt/puppet/bin/puppet agent',
                     '--test --no-use_srv_records --noop',
                     '--catalog_cache_terminus=""',
                     '--server puppet-next.ops.puppetlabs.net;',
                     '/opt/puppet/bin/puppet plugin download',
                     ], ' '),
    minute  => [fqdn_rand(25) + 4, fqdn_rand(25) + 34],
  }
}
```

Making use of data

- PE Console
- Reports processors
- PuppetDB
- `catalog_preview` module

The Plan

1. Overview
2. Building a test environment
3. Testing catalogs
4. Performance
5. The migration

JRuby worker threads

- `max-active-instances`
- `max-requests-per-instance`
- `environment_timeout`

PE Console class sync

```
classifier_syncronization_period = 0
```

```
path /puppet/v3/resource_type
method find, search
auth yes
# start OPS-7229 changes
# Only allow PE Console to sync class data for the production environment
environment production
# end OPS-7229 changes
allow pe-internal-dashboard, pe-internal-classifier
```

CPU and compile times

Metrics

The Plan

1. Overview
2. Building a test environment
3. Testing catalogs
4. Performance
5. The migration

Things to consider

- Service discovery via PuppetDB
- CA switchover
- How to switch the nodes
- Agent version

Two days before our migration...

Don't cache --noop catalogs

```
puppet agent -t --noop --catalog_cache_terminus=''
```

Overall outcome:

SUCCESS!

Questions?

Source: https://github.com/thrnio/puppet3-to-puppet4