

Puppet dCache Module

How to manage a small dCache instance with Puppet

12 April 2016 | Cristina Manzano, Oleg Tsigenov

Outline

- dCache in Jülich
- Why Puppet
- Puppet dCache module step by step
- Small demo
- What has to be done

dCache in Jülich

LOFAR LTA @ JSC in production since 2011:

10 servers

384 TB disk space

more than 6 PB on tape

ILGD instance in production since 2004:

3 servers, 4 pools, 120 TB tape space

Why Puppet

How to administrate a small site?

Manually

- Lot of time

Scripting

- A bit tricky

At some point you just have to start using a site management tool

Puppet dCache module

Should be:

simple:

```
class { 'dcache': }
```

flexible:

```
class { 'dcache':  
  package_ensure => 'dcache-2.13.29-1',  
}
```

Does not address installing another program/service that requires dCache

```
class { 'dcache':  
  package_ensure => 'dcache-2.13.29-1',  
  require        => Package['java-1.8.0-openjdk'],  
}
```

Puppet dCache module (cont.)

What the module can:

- Control dCache version
- Manage dCache service
- Generate configuration files:
 - dcache.conf, <host_layout_file.conf>, gplazma.conf, poolmanager.conf, pool's "setup"
- Support TSM (Mattias Wadenstein) interface

ToDo: manage dCache domain services

Puppet dCache module elements:

/etc/dcache/dcache.conf

```
class { 'dcache':  
  conf => {  
    'dcache' => {  
      'dcache.user' => 'dcache',  
      'dcache.layout' => '${host.name}',  
      ... },  
    'chimera' => {  
      'chimera.db.user' => 'chimera',  
      ... },  
    'pnfsmanager' => {  
      'pnfsmanager.db.connections.max' =>  
        '50'  
    }  
  }  
}
```

Puppet dCache module elements and Hiera

Separate data from code in Puppet manifests:

```
class { 'dcache':  
  conf => hiera('dcache_conf',  
    'nodef'),  
}
```

data.yaml

```
dcache_conf:  
  dcache:  
    dcache.log.dir      : '/var/log/dcache'  
    dcache.layout       : '${host.name}'  
  chimera:  
    chimera.db.user     : 'chimera'  
  pnfsmanager:  
    pnfsmanager.db.connections.max: '50'
```


Layout yaml

Structure of the **layout** parameter:

```
# global setting for layout
globals:                <layout global settings >
  'dcache.java.memory.heap': '512m'
domains:                <list of domain definitions >
  dCacheDomain:
    domainsettings:    <domain parameters>
      dcache.java.memory.heap: '256m'
    poolmanager:      <cell>
    nfs:
      nfs.version: '4.1' <cell parameters >
```

Layout - dependencies

dCache module will create pool(s) structure, if pool(s) defined in layout.
The parameter “pools_setup” customizes pool configuration.

```
class { 'dcache':  
  conf          => hiera('dcache_conf', 'nodef'),  
  layout        => hiera('dcache_layout', 'nodef'),  
  pools_setup  => hiera('pools_setup', 'nodef'),  
}
```

```
pools_setup:  
  '${host.name}_01':  
    setup: |  
      hsm create osm osm script -command=/opt/ndit/ndit.pl -c:gets=0 -c:puts=5 -c:removes=1  
  .... Lot of setu parameters there  
  '${host.name}_02':  
    setup: |  
      csm set checksumtype ADLER32  
      csm set policy -scrub=off  
  .... Lot of setu parameters there
```

Customizing poolmanager.conf

Customization of poolmanager.conf possible if node layout contains “dCacheDomain”.

```
class { 'dcache':  
  conf          => hiera('dcache_conf', 'nodef'),  
  layout        => hiera('dcache_layout', 'nodef'),  
  pools_setup   => hiera('pools_setup', 'nodef'),  
  poolmanager_conf => hiera('poolmanager_cfg_hash', 'nodef'),  
  # poolmanager_conf => hiera('poolmanager_cfg_txt', 'nodef'),  
}
```

The parameter **poolmanager_conf** can be hash or multiline string

dcache::poolmanager_conf as hash

```
poolmanager_cfg_hash:
  cm: [ 'set debug off', 'set update on', 'set magic on' ]
  rc: [ 'set max threads 2147483647', 'onerror suspend' ... ]
  pm:
    default:
      type: 'wass'
      options:
        sameHostRetry: 'notchecked'
        p2p-oncost : 'yes'
  psu:
    set: [ 'regex off', 'allpoolsactive off' ]
    unitgroups:
      net:
        world-net: [ '0.0.0.0/0.0.0.0' ]
      store:
        any-store: [ '*@*' ]
        ops-ugroup: [ 'ops:DEFAULT@osm' ]
      protocol:
        any-protocol: [ '*/*' ]
    poolgroups:
      ildg-pools: [ 'dcachepool2_01', 'dcachepool1_01', ... ]
      ops: [ 'dcachepool1_00', 'dcachepool2_00' ]
    links:
      ildg-link:
        ug: [ 'world-net', 'ildg-ugroup', 'any-protocol' ]
        set: '-readpref=10 -writepref=0 -cachepref=10 -p2ppref=-1'
        pg: ['ildg-pools']
      ops-link:
        ug: ['ops-ugroup', 'world-net', 'any-protocol']
        pg: ['ops']
```

Welcome to contribute

Lot of things to be done !!!

Small demo :

How puppet deploys dCache, change configurations,
upgrade version.

Thanks !!