Dynamic configuration of cluster services

WIKIMEDIA
FOUNDATION

How should we change the state of a database in MediaWiki?

WIKIMEDIA
FOUNDATION

How should we switch what datacenter is active for a service?

WIKIMEDIA
FOUNDATION

How should we insert/remove a server from a pool?

WIKIMEDIA
FOUNDATION

My answer:

NOT with a commit to a configuration repository

Configuration vs state

Config or state?

- N. of HHVM threads
- Max exec time
- Is server online
- Master or read-only? (server)
- Master or read-only? (client)
- Weight of a server (client/lb)

What the WMF is doing now

https://wikitech.wikimedia.org/wiki/Conftool

WIKIMEDIA
FOUNDATION

etcd

- Strongly consistent, distributed k-v store
- Small amount of data
- Watch API
- Somewhat limited auth
- Performance issues

We are NOT tied to this specific technology

Direct watch vs confd

WIKIMEDIA FOUNDATION

And what about DNS?

Small scale installs

We want most applications to work without specific knowledge of state management tech

Service discovery

WIKIMEDIA
FOUNDATION

The SOA address book

- What’s the url for the service that is read-write?
- What’s the local url?
- What are the servers for this service?
- Centralized logic, apps need (want to?) only discern local and rw

WIKIMEDIA
FOUNDATION

The SOA address book

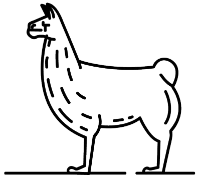
- DNS seems the natural candidate
- CNAME records for easy discovery
- TXT and URI records for the full URL
- Very short TTL - need a performant DNS server

Examples

```
$ dig +short -t TXT api.ro.discovery.wmnet
“https://api.svc.eqiad.wmnet/w/api.php”

$ dig +short -t TXT api.rw.discovery.wmnet
“https://api.svc.codfw.wmnet/w/api.php”

$ dig +short api.rw.discovery.wmnet
api.svc.codfw.wmnet
...
```



State management

WIKIMEDIA
FOUNDATION

Some things do not fit the DNS paradigm

- Complex data
- Aim for more predictable coordination
- Latency
- Third-party applications

Complex data structures

Some applications will need more than just a hostname or a url, but a full data structure. Think of servers in an LVS pool, or databases in a shard: they do have a series of intrinsic properties that go beyond that.

```
{name: db1085, shard: s2, role: slave, api: false, vslow: false}
```

More coordination in changes

While it is possible, I would advise against counting on it. In our experience, it’s not hard to have average propagation times below 1 second, but to ensure you have no higher latencies can be a problem. 5 seconds is probably a conservative timespan for change propagation when using confd / direct polling.

Latency

In some cases, having to resolve any DNS name every time to do service discovery can introduce unneeded latencies. Having confd push the new configuration to the system is a surefire performance gain esp in cases where we're already using DNS caching (e.g. HHVM).

confd

- Multiple backends
- Watch capabilities
- Create file base on go text/template
- Runs validation script on the generated file
- Can run a script after that

WIKIMEDIA
FOUNDATION

For example

- Watch `/conftool/v1/mw/databases`
- Generates a json file
- Sanity checks on the generated json
- Calls a special URL to parse and upload that json to the HHVM APC

Summarizing

- State and configuration should be treated separately
- Wikimedia uses conftool to manage state
- For simple service location/discovery a DNS interface is good and simple, but has limitations
- Confd + templates + scripts should be used when such limitations are an issue.
- MediaWiki: either write a json file and save its value to APC, or write a php file and add it to the source tree?
- Proper checks should be added to ensure the same revision of the config is used everywhere

THANK YOU

WIKIMEDIA

FOUNDATION