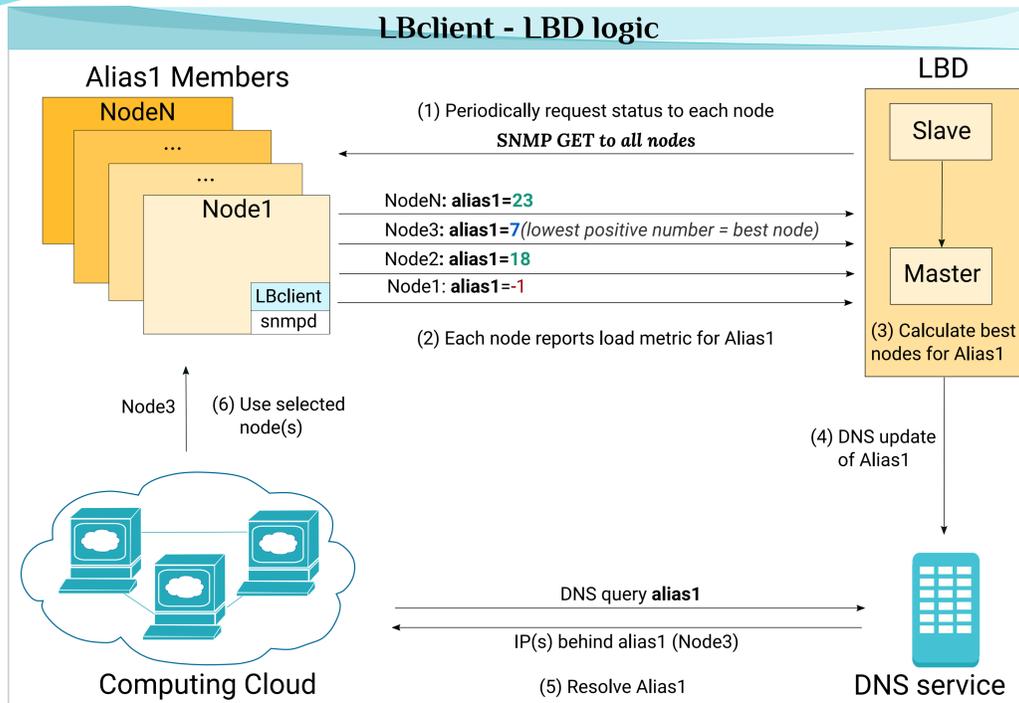


CERN uses an increasing number of DNS-based load balanced aliases. This is the GoLang based concurrent implementation of the Load Balancing Daemon (LBD), how it is being progressively deployed using Puppet and how concurrency greatly improves scalability, ultimately allowing a single master-slave couple of Openstack VMs to server all LB aliases.



The LBclient has been rewritten to integrate Collectd metrics. The new GoLang version introduces concurrency features that reduce the real time needed for the health check of the node.

The LBD server acts as an arbiter, getting feedback on load and health from the backend nodes using SNMP, and deciding which ones will be presented by the alias.

This architecture is well-established at CERN for DNS based aliases. The LBD code can be extended to drive other load balancers, so a prototype with HAProxy has been investigated.

## What is LBclient?

The LBclient is used to detect if a node can be behind an alias. The user defines multiple checks to asses the node health, and a load to sort the healthy nodes.

## What is Load Balance Daemon?

The Load Balance Daemon (LBD) contacts the different nodes behind the aliases and updates the DNS accordingly. The LBD is a pleasingly parallel workload. GoLang subroutines can be used to improve the execution time.

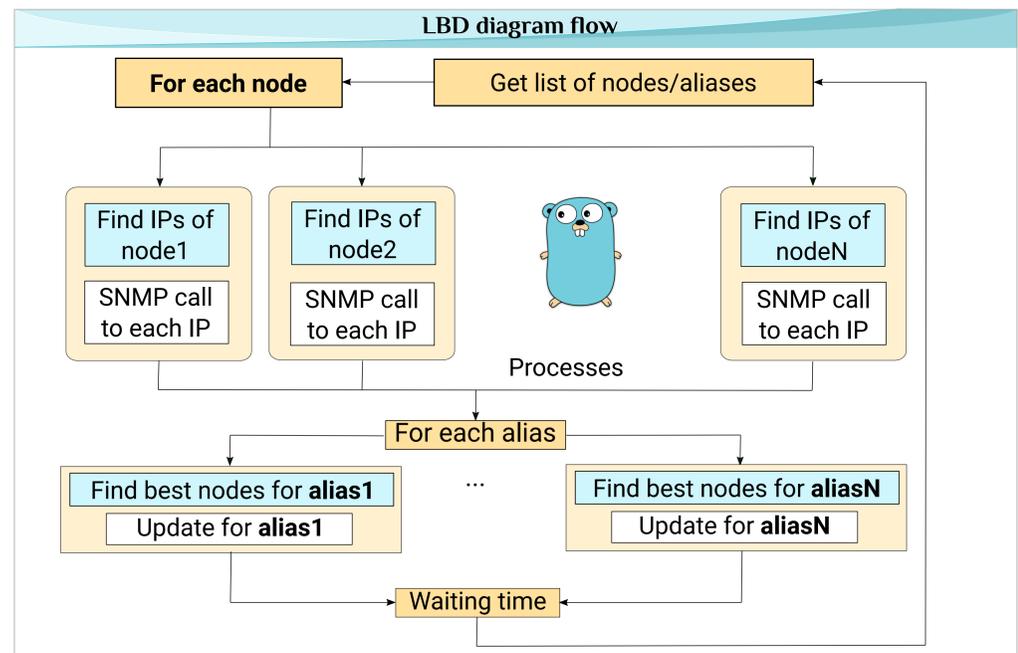
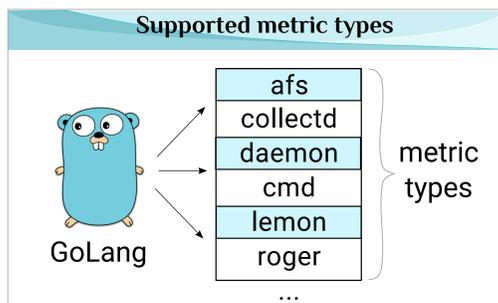
## Configuration

```
Alias names
...
lalias = rudy.cern.ch
lalias = alice.cern.ch
lalias = bob.cern.ch
...
```

```
Alias metrics
...
check <metric_type> <expression>
check <metric_type> <expression>
load <metric_type> <expression>
...
```

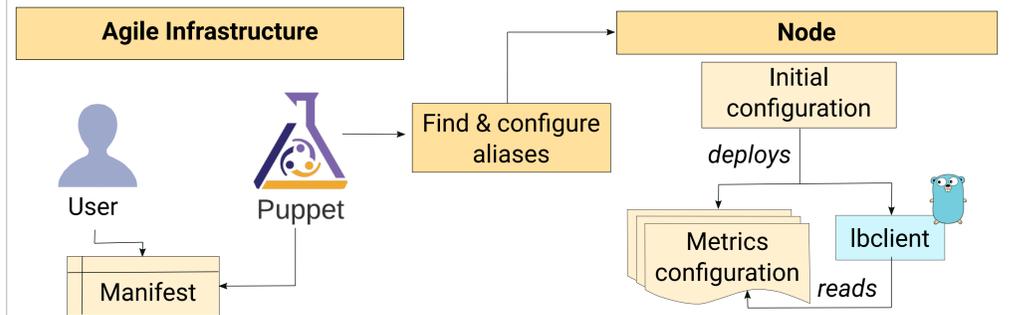
```
LBD alias definition and members
...
parameters alias1 = behaviour#mindlessbest_hosts#2external#nopolling_interval#300
clusters alias1 = node1 node2 node3
...
```

Each node can have multiple aliases. The user can have one configuration file per alias or a shared one across multiple aliases. The files contain the expressions to be used to select the healthy nodes.



- check:** boolean specifying if the node can be included in the alias.
- load:** integer specifying the responsiveness of the node. The nodes with the lower positive number will be selected by the LBD server.
- expression** (example):  $((\langle \text{metric\_X} \rangle * 2.15) / \langle \text{metric\_Y} \rangle) > 1 \ \&\& \ \langle \text{metric\_Z} \rangle \neq 0$

## Deployment

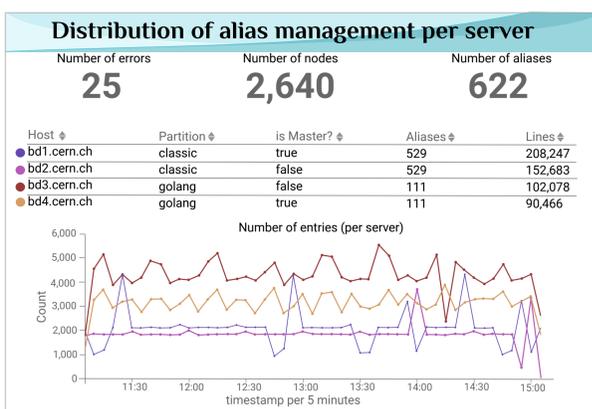
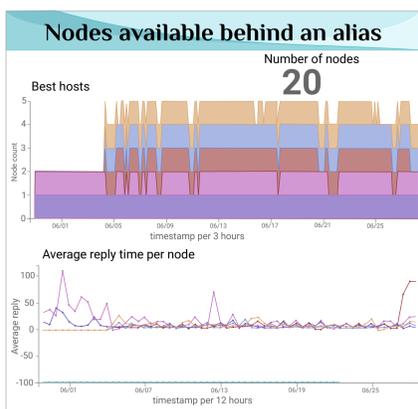


## High Availability

The LBD runs on two types of servers:  
**Master:** it contacts the nodes and updates the DNS.  
**Slaves:** they contact the nodes and check if the master is alive. If the master is not alive, they update the aliases.

## Monitoring

Monitoring the status of the aliases is crucial. The left plots show the load of the different nodes for a particular alias, and which nodes were included. The right plots emphasis how aliases are spread over several LBDs.



## DNS and HAProxy

Most of the load balancing logic can be reused between the DNS and the HAProxy aliases. Gathering the results from the nodes and selecting the best node remains the same.

The HAProxy update are done through the node weights. The information on the health and load of the nodes can be extended with HAProxy for cases such as sticky sessions and faster dynamics.

