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.

What is LBclient?

The LBclient is used to detect if a node can be behind an alias. The user defines multiple checks to assess 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

<table>
<thead>
<tr>
<th>alias1</th>
<th>alias2</th>
<th>alias3</th>
</tr>
</thead>
<tbody>
<tr>
<td>rudy.cern.ch</td>
<td>alice.cern.ch</td>
<td>bob.cern.ch</td>
</tr>
</tbody>
</table>

### Alias metrics

<table>
<thead>
<tr>
<th>Metric Type</th>
<th>Expression</th>
<th>Check</th>
<th>Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>afs</td>
<td></td>
<td>check</td>
<td></td>
</tr>
<tr>
<td>collectd</td>
<td></td>
<td>check</td>
<td></td>
</tr>
<tr>
<td>daemon</td>
<td></td>
<td>check</td>
<td></td>
</tr>
<tr>
<td>cmd</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lemon</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>roger</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### LBclient - LBD logic

1. Periodically request status to each node
2. Each node reports load metric for Alias1
3. Calculate best nodes for Alias1
4. DNS update of Alias1
5. Resolve Alias1

### LBclient - LBD diagram flow

- Find IPs of node1
- Find IPs of node2
- Find IPs of nodeN
- Update for alias1
- Update for aliasN

### Deployment

- Agile Infrastructure
- Find & configure aliases
- Node
- Initial configuration
- Deploys
- Metrics configuration reads
- lbclient

### 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 emphasize how aliases are spread over several LBDs.

### DNS and HAPerxy

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

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

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 HAPerxy has been investigated.