
Decentralized DRS

Objectives of DRS

- Distribute resources based on the overall cluster load and importance of each VM
 - Decentralized
-

Components

- A monitoring service on each host
- A memory balancing service on each host
- A Load Balancing service on each host

The Monitoring Service

- Runs on each host
 - Regularly (say every 1 second) collects CPU usage and memory usage data from each VM. Keeps an average of resource usage of past say 1 minute. If this average usage changes by a threshold, sync to a database or distributed key value store like etcd.
 - Also monitors the overall usage of the host. If the load profile changes significantly, then a migration decision has to be taken. Use the method discussed in <http://weblab.ing.unimo.it/papers/cloudcomp09.pdf>
-

The Memory Balancing Service

- Runs on each host
 - Distributes available memory on the host to all the VMs present using ballooning.
 - If a new VM is to be migrated to this host, make space for that.
-

The Load Balancing Service

- Runs on each host
 - When the monitoring service notifies this of a change in load profile, it fetches the the load of all the other physical machines, calculates resource entitlement of all the VMs on this machine based on their importance/shares.
 - If the resource entitlement cannot be satisfied on the this machine, choose a candidate to be migrated.
-

Choosing Migration Candidate

- Cost benefit analysis. Cost takes into account the number of migration in previous hour(to prevent a chain of migrations), still to be decided.
 - The destination chosen will be the one which has most free resources(greedy technique, best chance to prevent future migrations)
-

Problems

- How to consolidate?
 - What metric to use for measuring effectiveness of DRS?
 - If no suitable host found for migration, but some capacity is free, then what?
-

Implementation Details

- Can be implemented using openstack or even without it.
 - Which hypervisor to use? Xen or KVM?
-

Other similar works

-