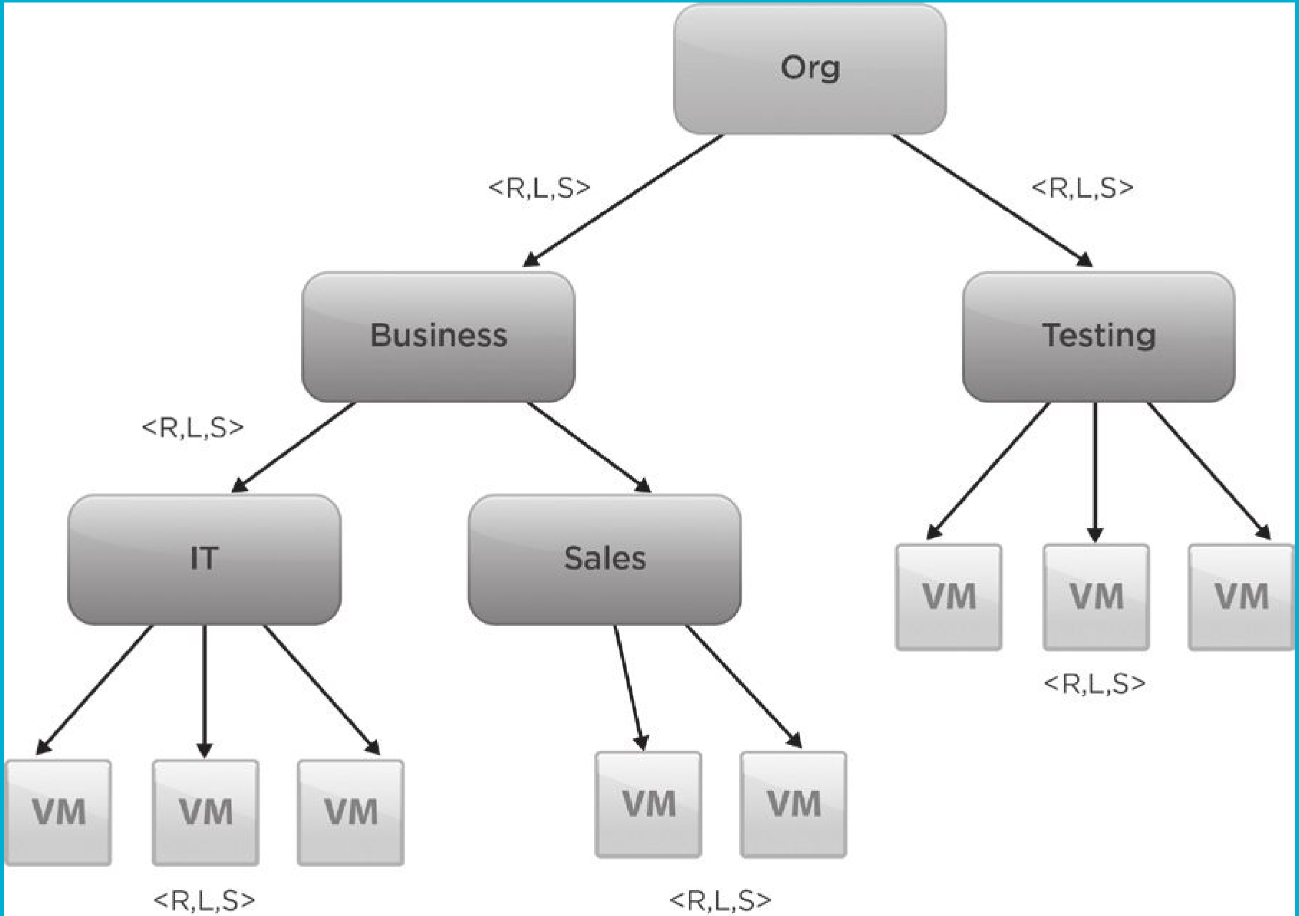
VMware DRS

DRS

- Stands for Distributed Resource Scheduler.
- Responsible for allocation of physical resources to a set of virtual machines deployed in a cluster of hosts.
- It provides
  - Initial placement of a VM when it is powered on
  - Dynamic load balancing based on fluctuating demands of both CPU and memory
  - A maintenance mode which evacuates PM’s by migrating all VMs from it

Resource Model

- Each VM has a few resource control parameters
  - **Reservation:** specifies a minimum guaranteed amount of a particular resource
  - **Limit:** specifies the maximum amount of resource that can be allocated to that VM
  - **Shares**: specify the relative importance of VM. VM can consume resources proportional to its share allocation
- Resource Pools
  - Used to allow flexible resource management policies for a groups of VMs
  - A hierarchical tree whose leaves are the actual VMs
  - The sum of reservations of a pool’s children must not exceed its own reservation


Org
<R,L,S>
<R,L,S>
Business
Testing
<R,L,S>
IT
Sales
VM
VM
VM
<R,L,S>
VM
VM
VM
VM
VM
<R,L,S>
<R,L,S>

Resource Pool Divvy

- The process of computing reservation, limit and shares of children of a pool is called divvying.
- It is performed in a hierarchical manner by dividing the resources of parent amongst its children.
- Divvying calculates resource entitlements for each pool/VM based on the resource control settings and overall load on the cluster.
- DRS carries out these divvy operations periodically(default every 5 minutes)

Demand Calculation

- To distribute resources among the VMs, calculating demand is important to get the total load on the cluster.
- Active memory of a VM is used as its memory demand. It is calculated by sampling a random set of pages and computing how many of them were touched in a specific time interval.
- CPU demand is computed as its actual CPU consumption plus a scaled portion the time it was ready to execute, but was queued due to contention.

$$CPU_{demand} = CPU_{used} + \frac{CPU_{run}}{CPU_{run} + CPU_{sleep}} * CPU_{ready}$$

Divvying

- The divvying algorithm works in two phases
- In first phase, it aggregates demand values of VMs from leaves up to root.
- At each step, the demand values are updated to be not less than reservation and not more than the limit.
- The second phase proceeds in top-down manner, resources(limit and reservation) of parents at each level are divided such that they are in proportion to shares of the siblings.
- If the sum of children's demand is greater than the quantity that is being divvied, the limit value of the children is replaced with the demand value.

Load Balancing

- DRS load-balancing metric is dynamic entitlement.
- It is equal to the demand of VM if demands of each VM in the cluster can be met, otherwise it is scaled down value of VM’s demand based on its shares.
- Computed by divvying cluster capacity at the root of the tree.
- For each host, normalized entitlement is calculated as follows

For a host h, normalized entitlement N_h is defined as sum of per VM entitlement of all VMs running on h divided by the host capacity.

$$N_h = \Sigma E_i / C_h$$

- If N_h < 1, then the host has some unused resources, otherwise it is overloaded.
- DRS calculates cluster wide imbalance I_c as the standard deviation of all N_h values.
- The cluster wide imbalance considers both CPU and memory imbalance using their weighted sum, where weights are decided by resource contested.
- Weights -
    - If memory is highly contested, its weight is more.
    - If CPU's is highly contested, its weight is more.
    - If none are highly contested, equal weights.
    - Weights are 3:1 which is determined by experimentation

Algorithm

- The algorithm aims to minimize cluster wide imbalance I_c by considering all possible VM migrations.
- The best VM migration is selected, applied to cluster’s state, and another move is selected.
- This proceeds till no beneficial moves remain, or enough moves have been selected for this pass, or the cluster imbalance is below a threshold.

Cost Benefit Analysis

- DRS uses this to filter out bad moves.
- Benefit is computed as how much the VMs on source benefit from resources that are emptied and how much the migrated VM benefits on the destination.
- Cost includes the time taken for VM migration.
- Based on its memory size and the time taken for previous migration, it calculates the time taken for a single round of copying.
- DRS then computes how much the workload inside the VM would suffer due to migration by measuring the page dirtying rate and the time taken for page transfer.
- The cost includes the impact of workload change on VMs on the destination.

References

- http://www.waldspurger.org/carl/papers/drs-vmtj-mar12.pdf