ThrMmt – Threshold - Minimal migration time

Thr

So basically what it does is set some threshold like this

$$L_{threshold} = \text{rate}_{\text{threshold}} \times (1 - rate_{norm}) \times K \times L_{\max}$$

Rate is a utilization floating point value between 1 and 0
K number of VM
L max load

A resource reallocation takes place only if the application's load changes (up or down) is over $L_{threshold}$

Datacenter:
while the system is running **do**
  wait for brokers to submit requests
  if broker requests virtual resource then
    allocate virtual resource to broker;
    start the application on new virtual resource;
  end if
  if broker releases virtual resource then
    revoke the virtual resource released by a broker
  end if
    return the operation result;
end while

Broker:
while the system is running **do**
  wait for the next reallocation cycle;
  initBroker();
  calThreshold();//calculate Threshold
  if currentLoad() > normLoad()+Threshold **then**
    calculate the amount of virtual resources needed;
      send a request to datacenter for more virtual resources;
      wait for response from datacenter;
      add new virtual resources to the local resource list;
  end if
  if currentLoad()<normLoad()-Threshold **then**
    calculate the amount of virtual resources to be released;
    send a request to datacenter to revoke virtual resources;
    remove the released virtual resources from the local resource list;
  end if
end while

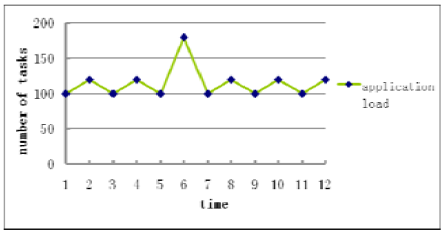

Fig.2. cloud application under oscillating workloads

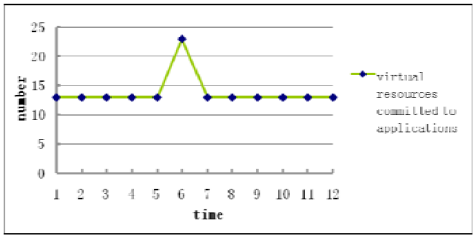

Fig.3 virtual resources committed to cloud application with threshold rate 0.95

MMT

```java
ublic class PowerVmSelectionPolicyMinimumMigrationTime extends PowerVmSelectionPolicy {
    @Override
    public Vm getVmToMigrate(PowerHost host) {
        List<PowerVm> migratableVms = getMigratableVms(host);
        if (migratableVms.isEmpty()) {
            return null;
        }
        Vm vmToMigrate = null;
        double minMetric = Double.MAX_VALUE;
        for (Vm vm : migratableVms) {
            if (vm.isInMigration()) {
                continue;
            }
            double metric = vm.getRam();
            if (metric < minMetric) {
                minMetric = metric;
                vmToMigrate = vm;
            }
        }
        return vmToMigrate;
    }
```

Summary:
Thresholding policy collects/choses the VMs for allocation and then Minimal migration time policy picks the vm order that they will be migrated in. Trigger for thresholding is any change in the vm load (down or up) as far as I understand the thresholding's advantage is stability performance stability.