Cloud Scale Resource Management: Challenges & Techniques

Ajay Gulati, Ganesha Shanmuganathan
Anne Holler, Irfan Ahmad
Distributed Resource Management Team
VMware, Inc.
Existing Solutions

- Elasticity
- Efficiency

Small scale solutions
i.e. VMware DRS

Desired

Amazon EC2

Goal: EC2’s *Elasticity* + DRS’ *Resource management*
DRS Resource Controls

- **Resource pools**: hierarchical resource allocation
- Reservation(R), Limit(L), Shares(S) per VM and RP node
- Allows statistical multiplexing, fair over-commitment of resources
- Very useful for Cloud, each org can own a resource pool

Resource pool tree (CPU)

- Finance: <R=3 GHz, L=150, S=10>
- Engineering: <R=7 GHz, L=150, S=20>
- VMs: <R=10 GHz, L=200GHz, S=1000>
DRS Primer

- Computes VM entitlement based on R,L,S
- Maps resource pools to hosts
- Initial placement of VMs
- Load balancing across hosts
  - Invoked every 5 mins

Resource pool tree

- <R,L,S>
- ESX hosts
- VMs

R: reservation
L: limit
S: share
How DRS works

- Uses *dynamic entitlement* as load metric for each resource
  \[ E_{VM} = F(R, L, S, demand) \]
  \[ R \leq E_{VM} \leq L \]

- Computes host normalized entitlement
  - \( C_H \) is host capacity
  \[ N_H = \frac{\sum E_{VM}}{C_H} \]

- Reduces \( N_H \) using Greedy-hill climbing

- Multi-dimensional balancing problem

- Does cost-benefit analysis for each move

- Finds incremental improvements, complete re-shuffle is not an option
Efficient Cloud Scale Resource Management Challenges

- **Heterogeneity of resources**
  - VM/Host compatibility matrix needs to be maintained
  - Islands formed due to storage and network connectivity
  - Multiple versions of hardware co-exist

- **High frequency of operations**
  - Centralized scheme may have high latency due to lock serialization
  - Distributed scheme may have stale inventory snapshot
  - Update host & VM stats/states periodically

- **Failure tolerance**
  - Failures are common at cloud scale
  - Need to handle/retry failed operations
Techniques

- Three proposals
  - Hierarchical Scaling
  - Flat Scaling
  - Statistical Scaling
Hierarchical Scaling

- Build a load balancer on top of clusters
- Cluster level metrics don’t aggregate well
- Scale of certain operations may be limited to a cluster
Flat Scaling

- Build a overlay of virtualized hosts
- Do initial placement and load balancing using
  - Distributed aggregation of stats (SDMIS, Sigcomm’04)
  - Queries on the overlay (SAAR, NSDI’07)
- Hard and challenging to build, debug
- No consistent views
Statistical Scaling

- Create small clusters dynamically
- Run DRS on this small cluster
- **Key idea:** Obtain large scale resource management by doing small scale optimizations
- **Well known result:** Power-of-two choices
Conclusions

A cloud solution needs to provide

**Elasticity + Efficient resource management**

- **Challenges:** scale, heterogeneity, rate of operations and failure handling

- **Techniques:**
  - Can we break the problem hierarchically?
  - Can we use completely decentralized approach?
  - Can we use local optimizations for global efficiency?