IBM Research

Towards an understanding of oversubscription in cloud

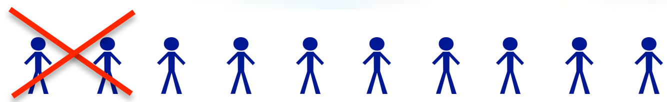
Salman A. Baset, Long Wang, Chunqiang Tang
sabaset@us.ibm.com
IBM T. J. Watson Research Center
Hawthorne, NY

Outline

- Oversubscription background
  - Airlines and cloud
  - What are typical overload symptoms for CPU, memory, disk, and network?
  - Isn't managing oversubscribed cloud the same as 'regular cloud'?
- Mitigating overload: mechanism vs. policy
- Contributions
  - Theoretical basis for oversubscription problem
  - A Markov model for oversubscription
  - SLAs and oversubscription
  - Results on increasing oversubscription in cloud by terminating or live migrating a VM while meeting SLAs
- Ongoing work

Motivation

10 seat capacity



Plans changed last minute

Airline boss: my planes are not flying full. Overbook the seats!

Motivation

10 seat capacity

12 people book seats, 2 cancel.

Airplane flies full

Motivation

10 seat capacity

12 people book seats, 12 show up

PROBLEM!!!!!!!

Refund, vouchers etc

Cloud motivation

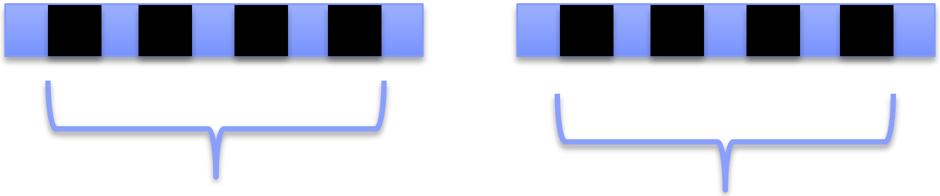
- Studies indicate that VMs do not fully utilize the provisioned resources
- Definitions
  - Provisioned resources
    - e.g., the resources with which a VM is configured. EC2 small instance (1.7 GB memory, 160 GB disk)
  - Used resources
    - e.g., the resources used by a VM at a point time (1 GB memory, 50 GB disk)
  - Overcommitted, oversubscribed
- Can we oversubscribe the resources of a physical machine while meeting the SLAs promised to a customer?

‘Regular’ cloud

8 GB RAM
1 TB disk
Quad core Xeon

8 GB RAM
1 TB disk
Quad core Xeon

VM:
2 GB RAM
500 GB
1 CPU



4 VMs per physical machine

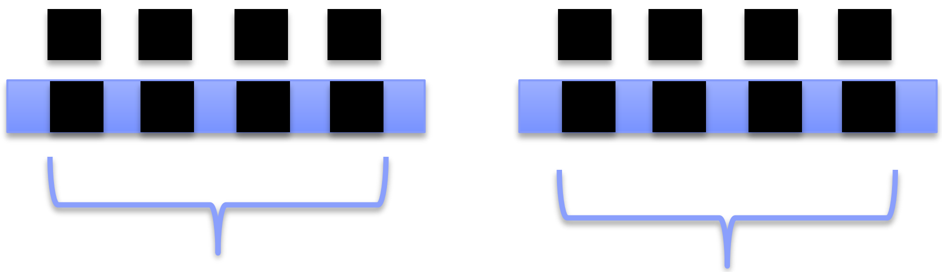
Black box indicates provisioned resources per VM

Oversubscribed cloud

8 GB RAM
1 TB disk
Quad core Xeon

8 GB RAM
1 TB disk
Quad core Xeon

VM:
2 GB RAM
500 GB
1 CPU



8 VMs per physical machine

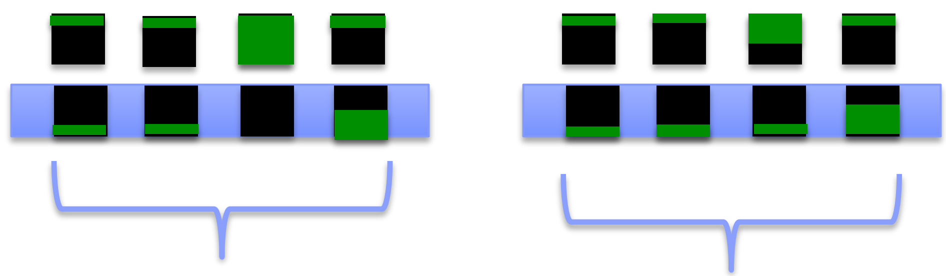
Black box indicates provisioned resources per VM

Oversubscribed cloud

8 GB RAM
1 TB disk
Quad core Xeon

8 GB RAM
1 TB disk
Quad core Xeon

VM:
2 GB RAM
500 GB
1 CPU



8 VMs per physical machine

Black box indicates provisioned resources per VM

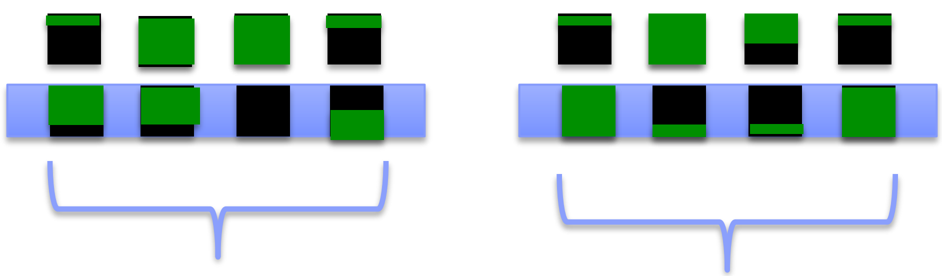
Green box indicates used resources per VM

Overload!

8 GB RAM
1 TB disk
Quad core Xeon

8 GB RAM
1 TB disk
Quad core Xeon

VM:
2 GB RAM
500 GB
1 CPU



VMs requesting more memory than available in physical server.

8 VMs per physical machine

Black box indicates provisioned resources per VM

Green box indicates used resources per VM

What are overload symptoms for CPU, memory, network, disk?

- CPU
- Memory
- Disk
- Network

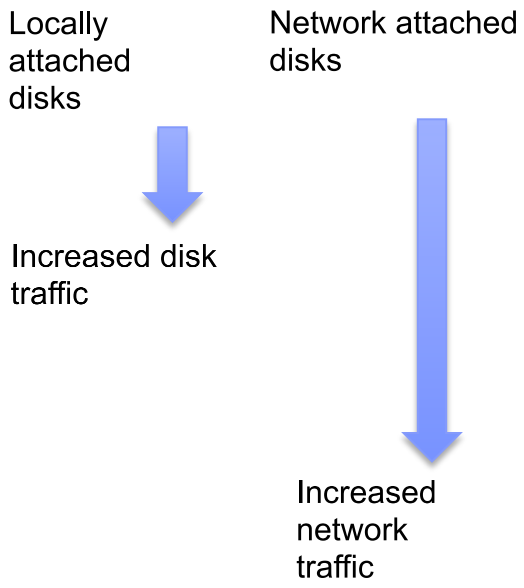
What are overload symptoms for CPU, memory, network, disk?

- CPU
  - less CPU share per VM, long run queues
- Memory
  - Swapping to hypervisor disk, thrashing
- Disk (spinning)
  - Increased r/w latency, decreased throughput
- Network
  - Link fully utilized

Are symptoms related?

What are overload symptoms for CPU, memory, network, disk?

- CPU
  - less CPU share per VM, long run queues
- Memory
  - Swapping to hypervisor disk, thrashing
- Disk (spinning)
  - Increased r/w latency, decreased throughput
- Network
  - Link fully utilized



What are overload symptoms for CPU, memory, network, disk?

- CPU
  - less CPU share per VM

Monitoring agents within VMs and hypervisor may not get a chance to run as per their schedule

- Memory
  - Swapping to hypervisor disk, thrashing
- Disk (spinning)
  - Increased r/w latency, decreased throughput
- Network
  - Link fully utilized

What are overload symptoms for CPU, memory, network, disk?

- CPU
  - less CPU share per VM
- Memory
  - Swapping to hypervisor disk, thrashing
- Disk (spinning)
  - Increased r/w latency, decreased throughput
- Network
  - Link fully utilized

If work of all VMs is I/O bound, a fully utilized link (for one VM) may cause other VMs to sit idle, wasting CPU and memory resources.

Isn't managing oversubscribed cloud the same as 'regular' cloud?

- Regular cloud
  - Only network and disk are susceptible to overload
  - CPU and memory are never oversubscribed
- Oversubscribed cloud
  - CPU, disk, memory, and network are oversubscribed

Mitigating overload

- Mechanism vs. policy
- Mechanisms
  - Stealing
    - Borrow resources from one VM and give it to other
  - Quiescing
    - Terminate a VM. Which VMs to terminate?
  - Migrate
    - Live migration
      - Shared vs. local disk storage
      - VMware VMotion
      - Streaming disks
    - Offline migration
    - Which VMs to live / offline migrate?
  - Network memory
    - Swap space is over network. May work for transient workloads.

Handling overload

- Overload detection
  - Detect that overload is occurring (within VMs or physical server)
  - Hard or adaptive thresholds
- Overload mitigation
  - Mitigate overload by terminating a VM, live migrating it, or using network memory

- It is hard!

Overload mitigation policy

- Factors to consider
  - Performance
  - Useful work done
  - Cost
  - Fairness
  - Minimal impact to VMs
  - SLAs

- An optimization problem

Oversubscription and classical problems

- Multiple-constraints single knapsack (FPTAS polynomial in n and 1/e for e > 0)
  - Given n items and one bin (single knapsack)
  - Each item and bin has d dimensions, and each item has profit p(i)
  - Find a packing of n items into this bin which maximizes profit, while meeting bins dimensions
- Multiple knapsacks (bin packing) (PTAS polynomial in 1/e for e > 0)
  - Given n items, and m bins (knapsacks)
  - Each item has a profit, p(i), and size(i)
  - Find items with maximum profit that fit in n bins
- Vector bin packing (no-APTAS cannot find a PTAS for every constant e > 0)
  - Given n items and m bins
  - Each item and bin has d dimensions
  - Find a packing of n into m which minimizes m, while meeting bins dimensions
- Online vector bin packing
  - Same as above
  - but also minimize the total number of moves across bins or VM terminations

The underlying theoretical problem of oversubscription

- Online multiple constraints multiple knapsack problem with costs of moving between knapsacks
  - Given n items (VMs), and m bins (servers)
  - Each VM and server has d dimensions, and each VM has utility u(i)
  - Moving a VM from server i to j has a cost M_{ij}
  - Terminating a VM k has a cost T_k
  - lambda is the rate of arrival of workloads within VMs (iid)
  - Utility of a VM and PM, U_{VM}, U_{PM}, respectively
  - State space:
    - resource consumption of PMs and VMs resources
      - PM resources: CPU, memory, disk, network
      - state tuple: (PM_{i-CPU} , PM_{i-disk} , PM_{i-mem}, $PM_{i-network}$)
      - state space explosion
    - probability of being in that state, given workload distributions
    - Utility of a state

- Given workload distributions, find argmax number of VMs s.t.
  - Total utility (profit) is maximized

SLAs and overload

- Overload must be precisely defined as part of SLAs
- What are the SLAs of public cloud providers?
  - None provide any performance guarantees for compute
  - Uptime guarantees, typically only for data center and not for VMs.

Compute SLA comparison

	Amazon EC2	Azure Compute	Rackspace Cloud Servers	Terremark vCloud Express	Storm on Demand
Service guarantee	Availability (99.95%) 5 minute interval	Role uptime and availability, 5 minute interval	Availability	Availability	Availability
Granularity	Data center	Aggregate across all role	Per instance and data center + mgmt. stack	Data center + management stack	Per instance
Scheduled maintenance	Unclear if excluded	Includ. in service guarantee calc.	Excluded	Unclear if excluded	Excluded
Patching	N/A	Excluded	Excluded if managed	N/A	Excluded
Guarantee time period	365 days or since last claim	Per month	Per month	Per month	Unclear
Service credit	10%			of	1000% for every hour of downtime –
Violation report respon.	Customer	Customer	Customer	Customer	Customer
Reporting time period	N/A	5 days of occurrence	N/A	N/A	N/A
Claim filing timer period	30 business days of last reported incident in claim	Within 1 billing month of incident	Within 30 days of downtime	Within 30 days of the last reported incident in claim	Within 5 days of incident in question
Credit only for future payments	Yes	No	No	Yes	No

Uptime guarantees on a data center (very weak)
Implicit uptime guarantees on a VM

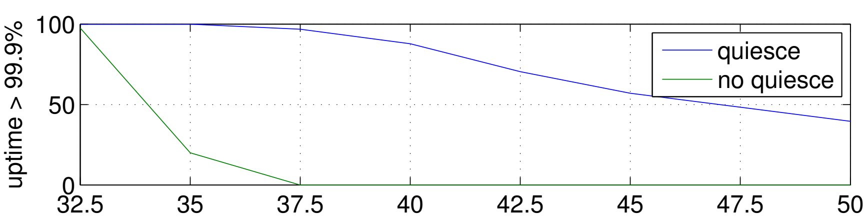
Cloud SLAs: Present and Future. To appear in ACM Operating System Review

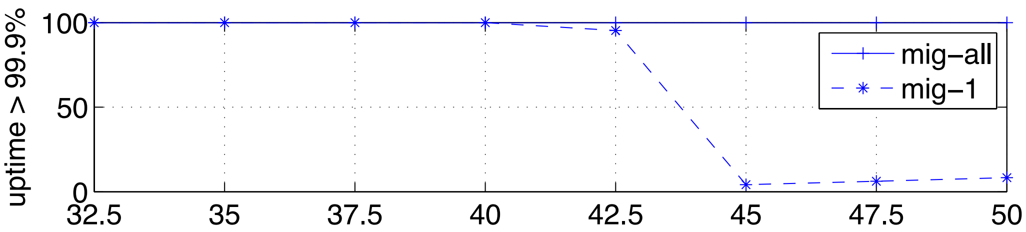
Questions investigated in this paper

- Overload detection interval and request inter-arrival within VM
- Mitigating overload by terminating VMs over a do nothing approach
- Mitigating overload by live migrating a VM, over terminating VMs and do nothing.

- Simulations
  - Setup
    - 40 PMs (rack of physical machines), each has 64 GB of RAM
    - Only memory overload
    - 30 days of simulated time
    - Number of VMs fixed
    - Request interarrival rate exponentially distributed
    - Request size exponential and pareto – (real data set in progress)
    - Live migration: 1 VM per minute at most (mig-1) or all VMs until overload alleviated (mig-all).
  - Overload definition
    - If memory consumption exceeds 95% of physical server memory for five contiguous minutes, overload occurs.
  - Metrics
    - Percentage of VMs not experiencing overload for given workload arrival rate
    - Number of VMs terminated and migrated

Preliminary results



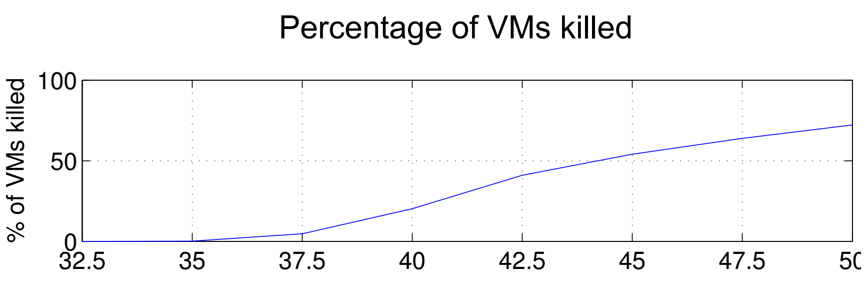


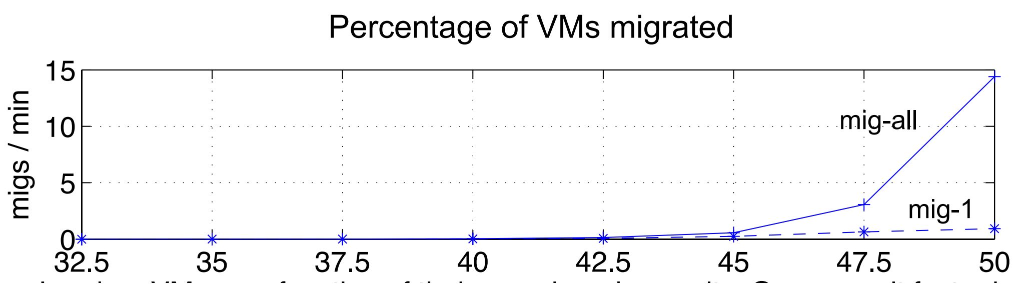
- Overcommit factor is 2.
- All VMs have same provisioned memory, i.e., 2 GB. Physical server has 64 GB memory.
- Average load on VMs as a function of provisioned capacity. E.g., 32.5% of 2 GB = 650 MB
- When average load on all VMs is 50% of provisioned capacity, the physical server memory is exhausted.

- Migration strategy: Select the VM with the largest memory consumption and terminate or live migrate it

- **Insights:**
  - **Terminating a VM improves the uptime performance of all VMs by more than a factor of 2 over a do nothing approach.**
  - **Mig-1 (at most one migration per minute results in a step function like reduction in uptime)**

Preliminary results





- **Insights:**
  - **One or more VMs killed as aggregate memory consumption of all VMs approach physical server memory**
  - **mig-all can overly stress the network**
  - **Always selecting the VM with highest memory consumption for terminating or live migrating is not a good idea!**

Questions under investigation

- To what extent a combination of VM quiescing and live migration schemes perform better than the individual schemes?
- Does asymmetry in oversubscription levels across PMs (within the same rack) and workload distributions lead to a higher overall overcommit level?
- When identical or asymmetric capacity VMs have different SLAs, which overload mitigation scheme gives the best results?
- When the available SLAs are defined per VM group instead of per VM, can it be leveraged to improve the performance of underlying overload mitigation scheme?
- How are the results affected when other resources such as CPU, network, and disk are oversubscribed?
- What is the best strategy for selecting VMs to terminate or live migrate?
- How the SLAs should be defined for oversubscribed environments?
- How can we answer all of the above questions for real workloads in a test-bed or deployed environment?