Summary of Recent Research Developments on Using Autonomic Computing to Mitigate the Downsides of Virtualization in the Cloud

Presented by Grady Moran

Virtualization

- Benefits:
  - Helps solve lack of energy proportionality
    - Migration and vertical scaling allow for fine tuning of PM utilization ratio
  - Migration
    - Can recover from failures, allows easy transfer between multiple physical machines
  - Workload consolidation
  - Custom user environment
  - Privacy
  - Process isolation

Virtualization

- Downsides:
  - Overhead associated with running an entire OS on top of server's OS
  - Migration is time consuming and computationally expensive
  - Requires making NP-hard allocation decisions when optimizing for some metric

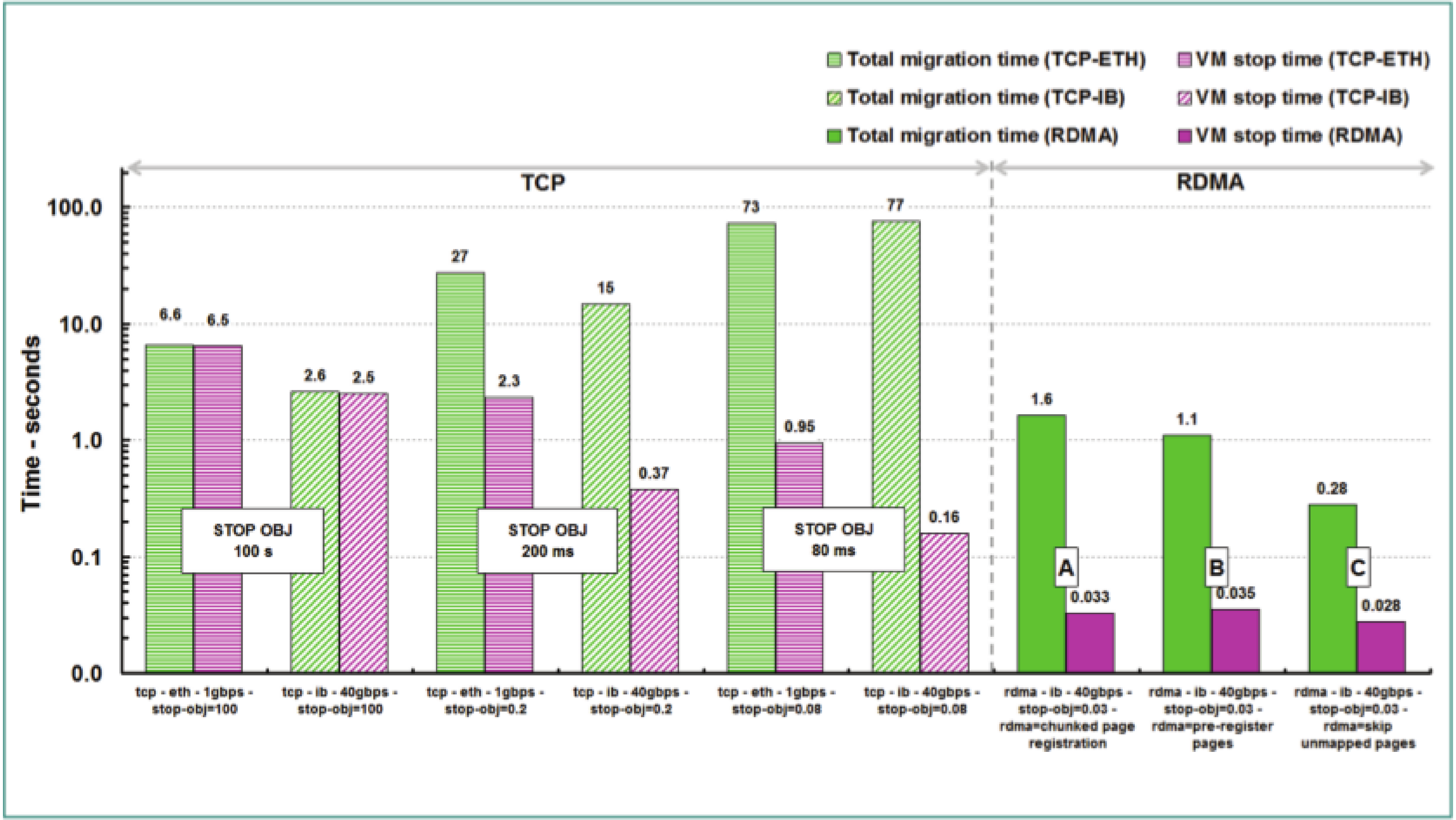
Live Migration

- Works by keeping source VM running while copying information (such as memory state and meta information) to destination
- During this period source VM overwrites some of the transferred memory. This requires transferring more memory pages.
- Once the difference in memory is small enough or a time limit is reached the VM is transferred along with remaining pages and CPU state
  - Mandatory downtime during transfer, but it can be very short
- Authors notice migration increases significantly with higher levels of VM memory utilization

RDMA-based Migration

- Normal migration transfers memory pages via TCP, which requires a lot of overhead.
- Instead, use Remote Direct Memory Access protocol (RDMA) to transfer memory pages
- Requires specialized hardware
- Implemented by modifying the hypervisor

Results


Total migration time (TCP-ETH)
VM stop time (TCP-ETH)
Total migration time (TCP-IB)
VM stop time (TCP-IB)
Total migration time (RDMA)
VM stop time (RDMA)
TCP
RDMA
Time - seconds
100.0
10.0
1.0
0.1
0.0
6.6
6.5
2.6
2.5
27
2.3
15
0.37
73
0.95
77
0.16
1.6
0.033
1.1
0.035
0.28
0.028
STOP OBJ 100 s
STOP OBJ 200 ms
STOP OBJ 80 ms
A
B
C
tcp - eth - 1gbps - stop-obj=100
tcp - ib - 40gbps - stop-obj=100
tcp - eth - 1gbps - stop-obj=0.2
tcp - ib - 40gbps - stop-obj=0.2
tcp - eth - 1gbps - stop-obj=0.08
tcp - ib - 40gbps - stop-obj=0.08
rdma - ib - 40gbps - stop-obj=0.03 - rdma=chunked page registration
rdma - ib - 40gbps - stop-obj=0.03 - rdma=pre-register pages
rdma - ib - 40gbps - stop-obj=0.03 - rdma=skip unmapped pages

Distributed Migration Decisions

- Problems associated with single global controller:
  - Does not scale well
  - Single point of failure
- Would like to allow servers to self-migrate in a way that leads the cluster as a whole to a load-balanced configuration

Distributed Migration Decisions

- Network model used assumes not full interconnectivity between servers, instead they are grouped in neighborhoods, and neighborhoods are connected.
- When utilization reaches some critical level, a server retrieves VM allocation lists for all neighbors and creates a load balancing configuration. Then servers migrate VMs as necessary.
  - Some extra behavior for when multiple nodes are executing this process simultaneously

Results

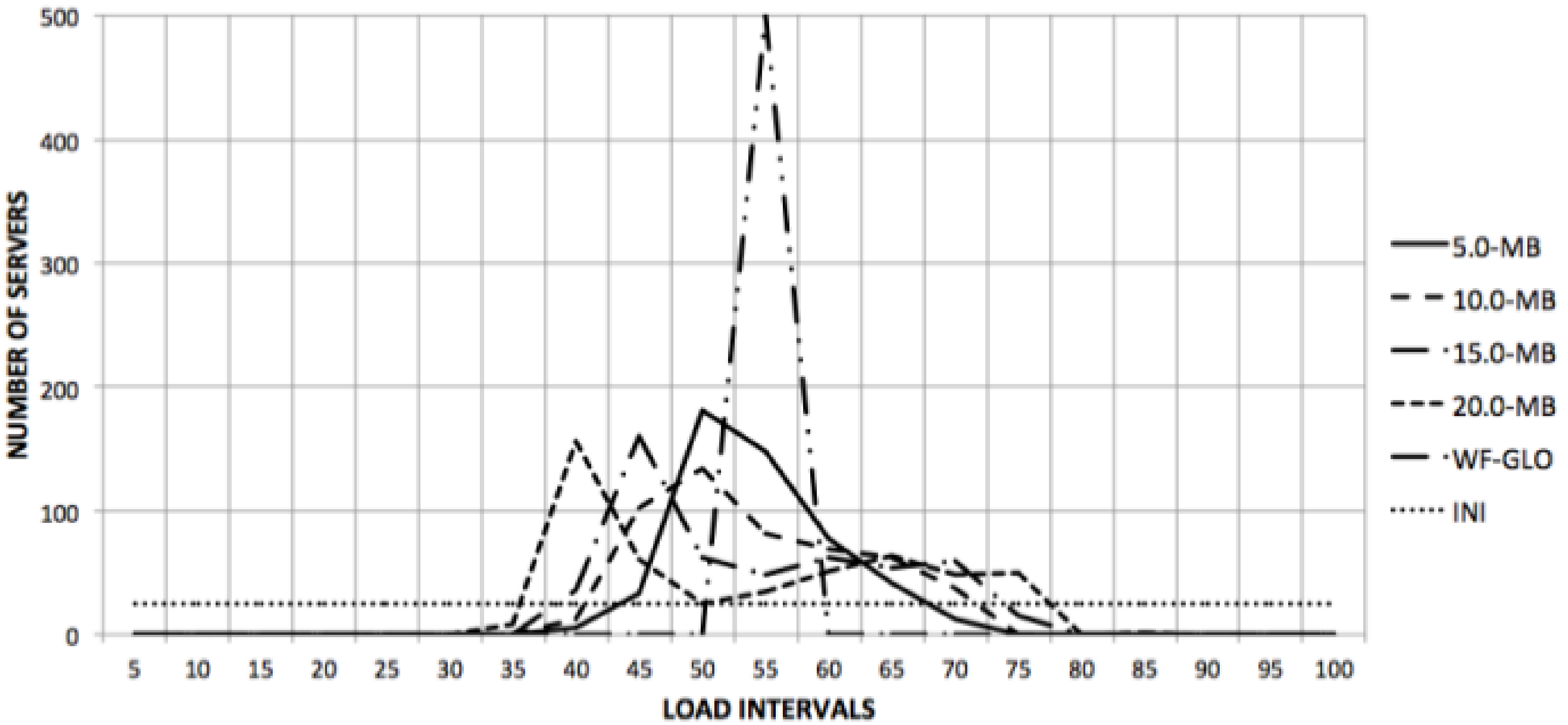


Fig. 4: Distribution of servers on load intervals.

Saving Power with Migration and Vertical Scaling

- Vertically scale VMs when their CPU or Memory utilization levels go outside acceptable bounds
- Increase level of VM resources when usage levels go above an upper threshold parameter
- Decrease level of resources when usage goes below lower threshold parameter

Saving Power with Migration and Vertical Scaling

- Four different migration policies suggested:
  - First Fit: put VMs on PM with least power consumption
  - Round Robin: Allocate one VM on each PM until none remain
  - Monte Carlo: Initially allocate like RR, but then reallocate by associating costs with migrations, overloaded hosts, and empty hosts, and try a set number of random configurations. The lowest cost is applied
  - Vector Packing: Sort incoming VMs by request capacity and use greedy algorithm to group sets of VMs with total capacity near some level, and map each set to a PM

Saving Power with Migration and Vertical Scaling

- Unoccupied physical machines are shut down incrementally at a rate defined by a parameter.
- A minimum of one unoccupied PM is left on to handle workload spikes.
- Additional machines are turned on as necessary

Results

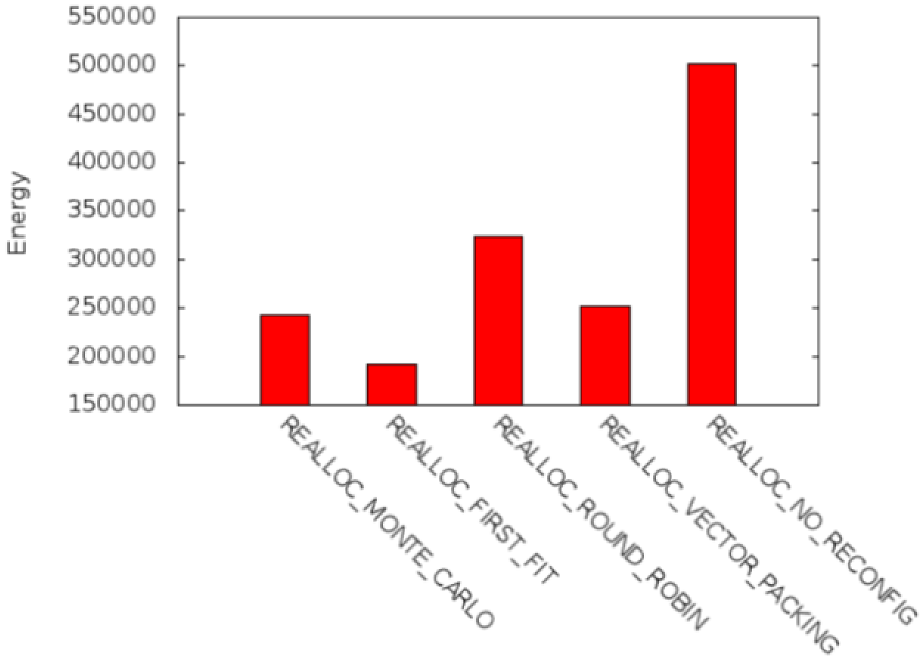


Figure 9: Energy consumption for Experiment 1

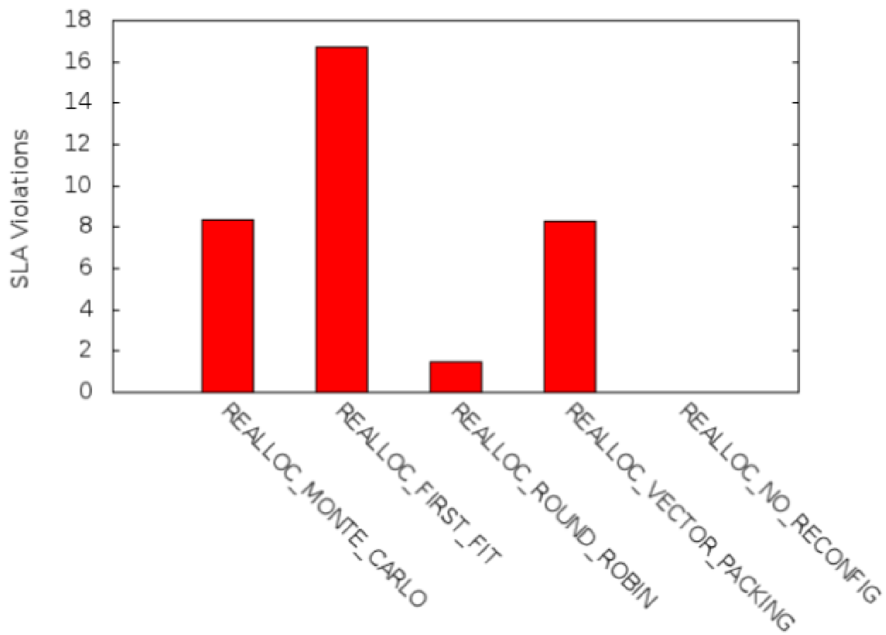


Figure 10: SLA violations for Experiment 1

SmartScale

- Allows a user to decide how to change allocation of resources when workloads change.
- Specifically, when workloads increase, should the instances be scaled vertically or horizontally? And vice versa.
- Vertical scaling: fast and cheap, low range
- Horizontal scaling: slow and expensive, high range

SmartScale

- Solution is to split the problem in two: decide optimal horizontal scaling level, then vertical scaling level.
- Calculate number of servers necessary to provide the requested level of performance
- Calculate the allocation requirements for each VM by using a model of performance, cost, and migration penalty
- Combine the two results and apply to system

Results

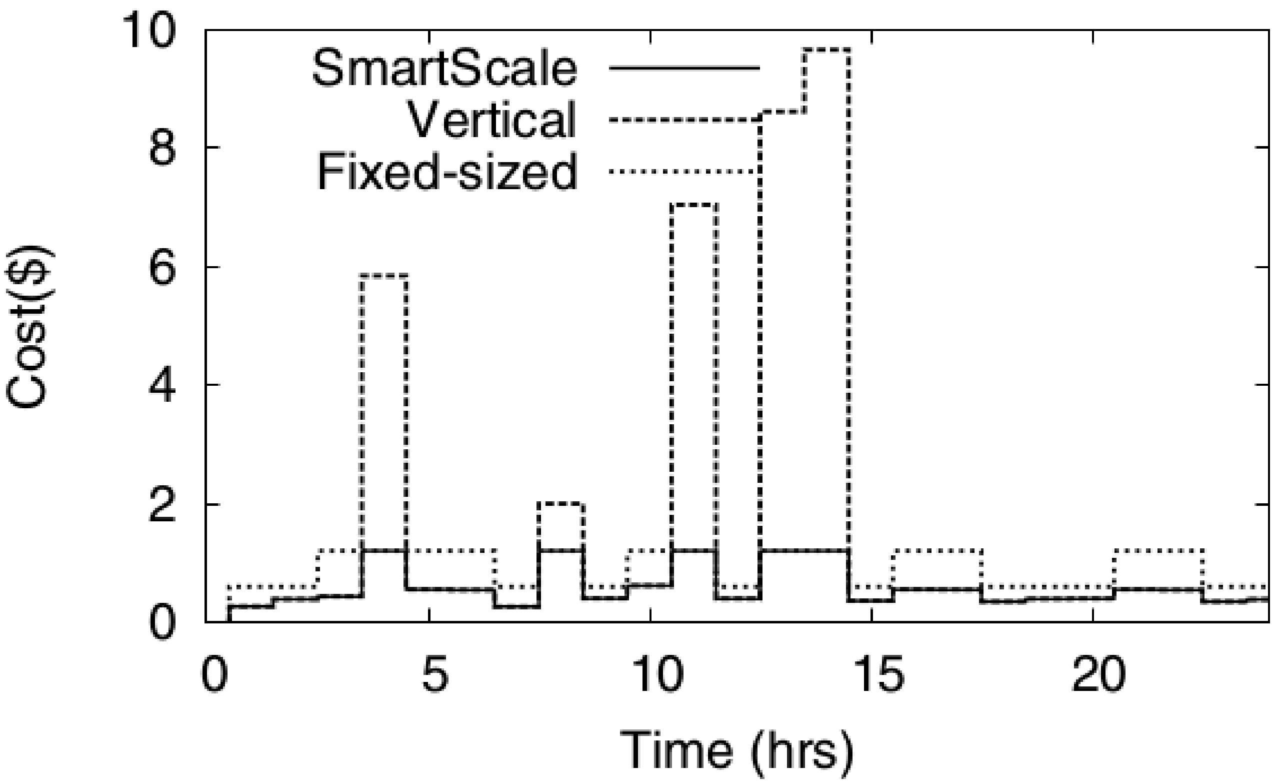


Fig. 9. Total Cost with time

Further Research

- Can easily modify the distributed load balancing work to consolidate workloads and encourage servers to either allocate near some optimal level or shut off
- Possibility to autonomically tune VM stop time to control the ratio of downtime to migration preparation time

References

- C. Isci, J. Liu, B. Abali, J. O. Kephart, and J. Kouloheris, “Improving server utilization using fast virtual machine migration,” IBM Journal of Research and Development, vol. 55, pp. 4:1–4:12, Nov 2011.
- D. Borgetto, M. Maurer, G. Da-Costa, J. M. Pierson,and I. Brandic, “Energy-efficient and sla-aware management of iaas clouds,” in 2012 Third International Conference on Future Systems: Where Energy, Computing and Communication Meet (e-Energy), pp. 1–10, May 2012.
- D. Loreti and A. Ciampolini, “A distributed self-balancing policy for virtual machine management in cloud datacenters,” in 2014 International Conference on High Performance Computing Simulation (HPCS), pp. 391–398, July 2014.
- S. Dutta, S. Gera, A. Verma, and B. Viswanathan, “Smartscale: Automatic application scaling in enterprise clouds,” in 2012 IEEE Fifth International Conference on Cloud Computing, pp. 221–228, June 2012.