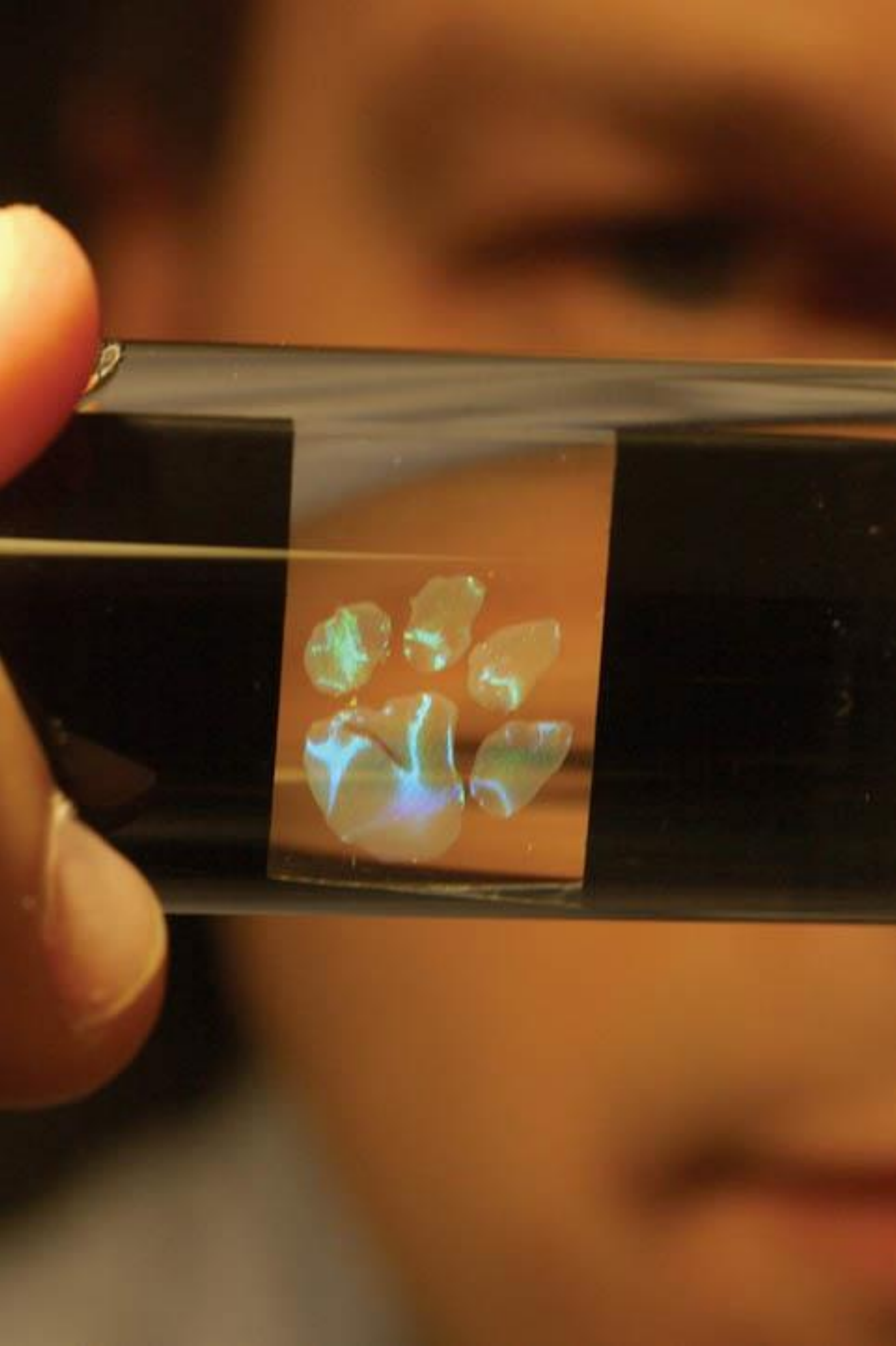


RIAL: Resource Intensity Aware Load Balancing in Clouds

Liuhua Chen and Haiying Shen and Karan Sapra
Dept. of Electrical and Computer Engineering
Clemson University, SC, USA

Outline

- Introduction
- System Design
- Performance Evaluation
- Conclusions

Introduction

- Cloud computing is becoming increasingly popular
  - Leading to an increasingly large number of VMs
- The operational expense of managing VMs represents a significant fraction of overall costs for datacenters
- Determining good VM-to-host mappings
- Multiple resources (CPU, Memory, I/O, Bandwidth)

Introduction (cont.)

- Load balancing methods using VM migration
  - Equal weights: Sandpiper [NSDI'07]
  - Predefined weights: TOPSIS [CoRR'10]
- Load balancing on one resource
  - Storage: Hsiao et al. [TPDS'12]
  - Bandwidth: FairCloud [Sigcomm'12]
- Resource management focuses on initial placement of VMs
  - Power: Lin et al. [Infocom'11]
  - Stochastic Models: Maguluri et al. [Infocom'12]

Introduction (cont.)

- Assign the same or predefined weights
  - Neglect the difference of resources
  - Neglect the time-varying feature
- Do not consider the communication between VMs
- Do not consider VM performance degradation due to migration

$$D_{ijp} = \sum d_{ip} \cdot \int_{t}^{t+\frac{M_{ij}}{B_{ip}}} u_{ij}(t)\mathrm{d}t$$

Introduction (cont.)

Resource Intensity Awareness

+

Communication Awareness

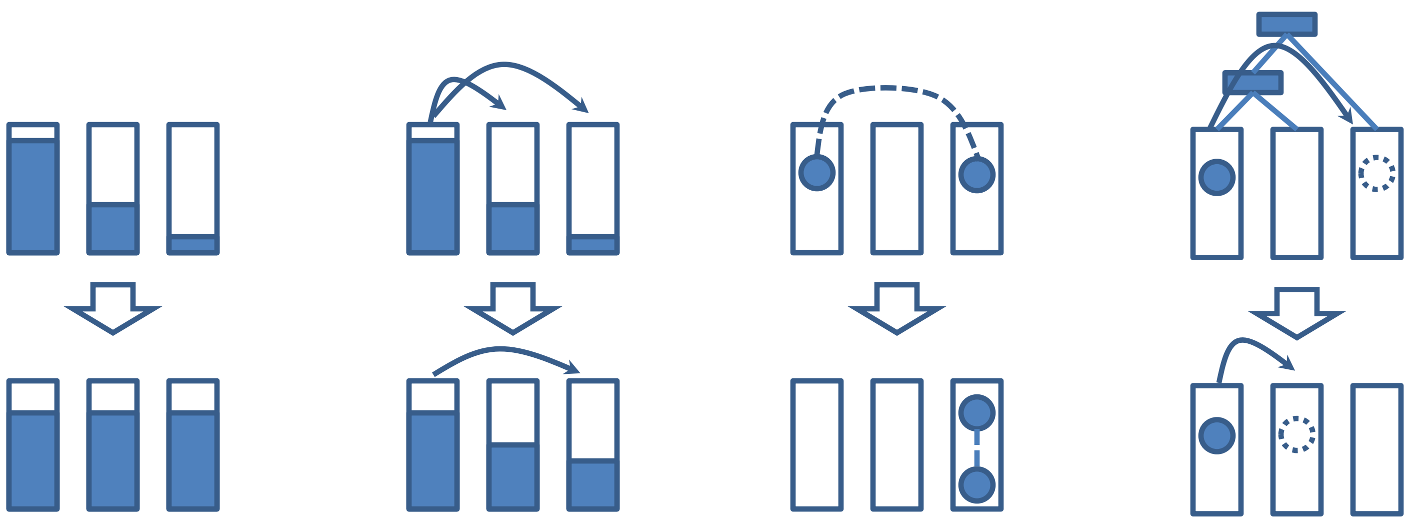
Performance degradation Awareness

↓

RIAL: Resource Intensity Aware Load Balancing in Clouds

System Design: Objective

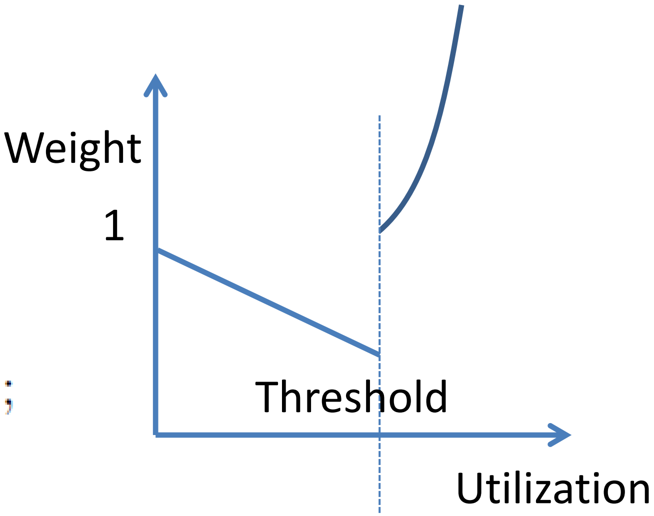
- Avoid overload
- Minimize the number of VM migrations
- Minimize the communication between PMs
- Minimize performance degradation



Design: RIAL

- Selecting VMs to Migrate (based on Multi-Criteria Decision Making Algorithm (MCDM))
  - Assign weights to each resource according to intensity

$$w_{ik} = \begin{cases} \frac{1}{1-u_{ik}}, & \text{if } k \in O, \\ 1 - u_{ik}, & \text{if } k \in L. \end{cases}$$



  - Determine ideal migration VM

$$R_{VM} = \{r_{i1}, ..., r_{i|K|}\}$$

$$= \{(\max_j x_{ijk} | k \in O), (\min_j x_{ijk} | k \in L)\};$$

  - Considering VM communication

$$l_{ij} = \sqrt{\sum_{k=1}^{|K|} [w_{ik}(x_{ijk} - r_{ik})]^2 + [w_t T_{ij}]^2},$$

Design: RIAL (cont.)

- Selecting Destination PMs (based on MCDM)
  - Assign weights to each resource according to intensity
  - Determine ideal destination PM

$$R'_{PM} = \{r'_1, ..., r'_k, ..., r'_{|K|}\} = \{\min_j x'_{jk} | k \in K\}.$$

  - Considering VM to PM communication

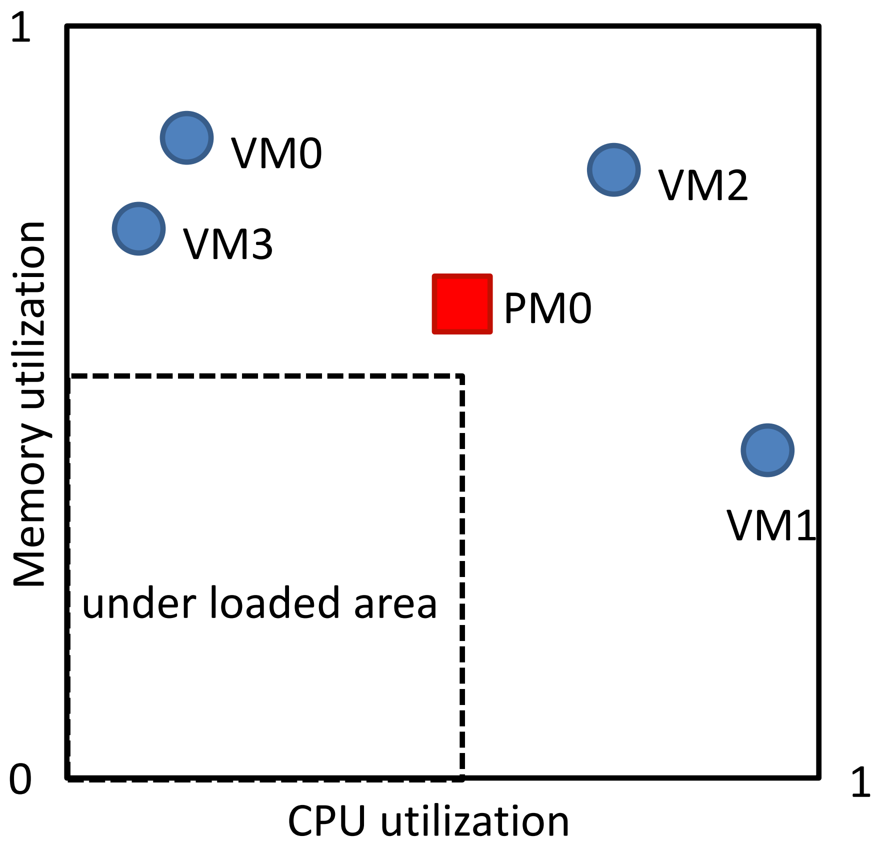
$$[w_t(T_{ijp} - T_{max})]^2$$

  - Considering performance degradation

$$[w_d D_{ijp}]^2$$

$$l_{p,ij} = \sqrt{\sum_{k=1}^{|K|} [w_{ik}(x'_{pk} - r'_k)]^2 + [w_t(T_{ijp} - T_{max})]^2 + [w_d D_{ijp}]^2}$$

Design: example



VM0 (0.20, 0.90)
VM1 (0.90, 0.40)
VM2 (0.75, 0.75)
VM3 (0.10, 0.75)

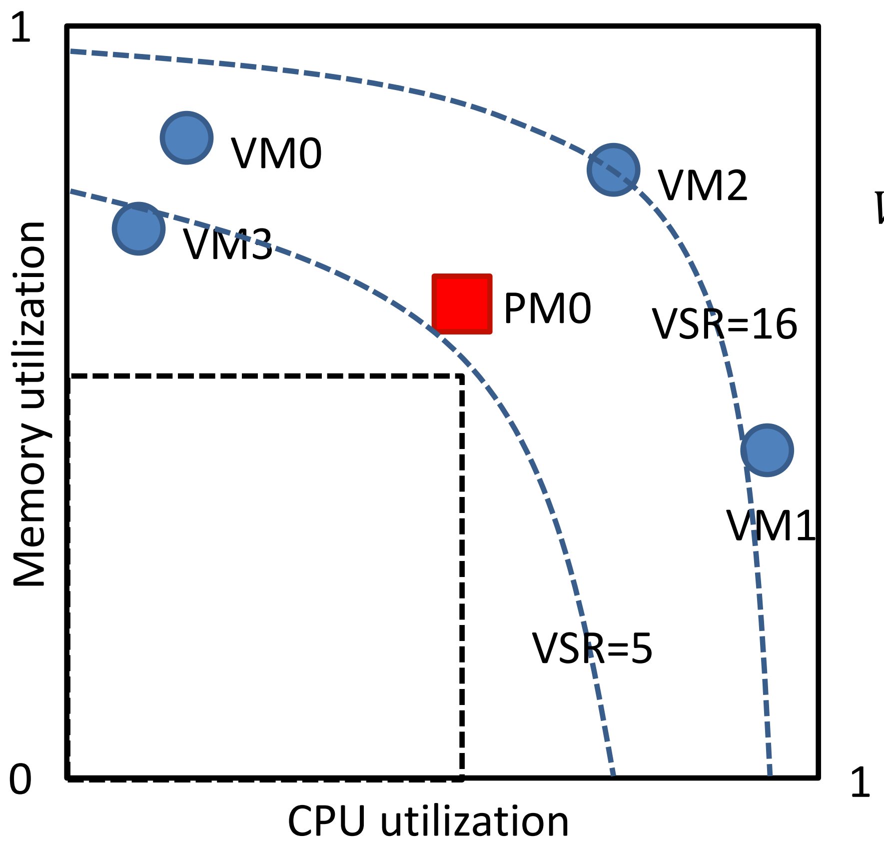
PM capacity = 4 X VM capacity

$$PM0=\frac{\sum(VM\ util \times VM\ Cap)}{PM\ cap}=(0.49, 0.70)$$

Assume Threshold = 0.5

PM0 overloaded!

Design: example (cont.)

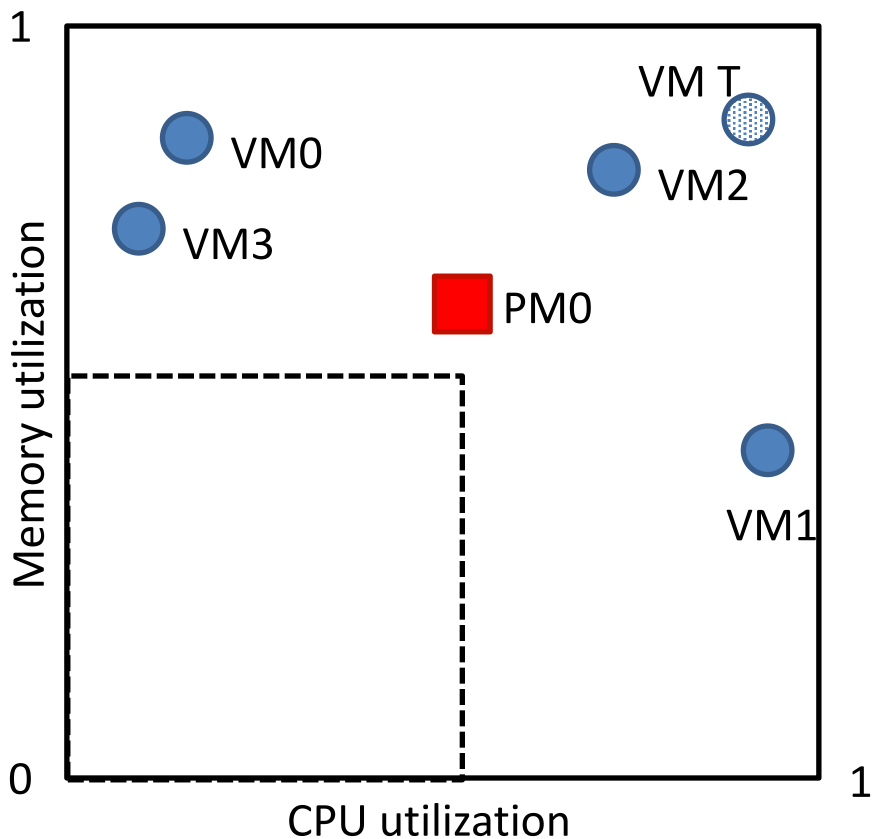


Sandpiper

$$VSR = \frac{Volunme}{Size} \propto \frac{1}{1 - u_{cpu}} \cdot \frac{1}{1 - u_{mem}}$$

VM1 has the highest VSR,
VM1 is selected to migrate out

Design: example (cont.)



TOPSIS

Weights for CPU:MEM is 9:4

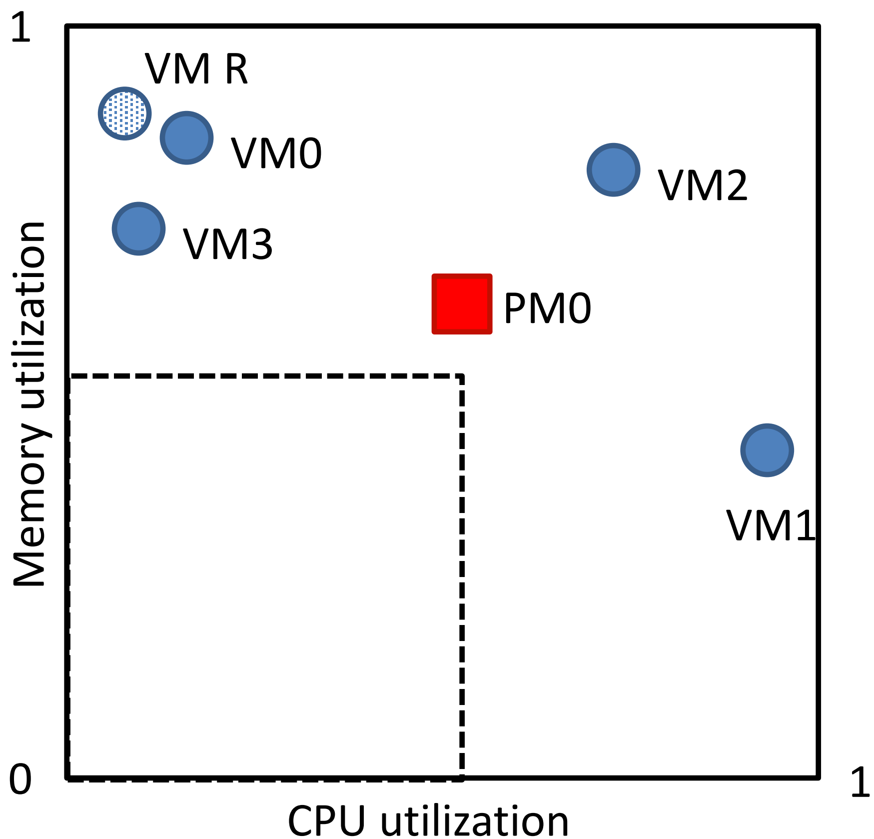
VM0 (0.20, 0.90)
VM1 (0.90, 0.40)
VM2 (0.75, 0.75)
VM3 (0.10, 0.75)

max max

Determine ideal VM T (0.90, 0.90)

VM2 is weighted closest to VM T,
VM2 are selected to migrate out

Design: example (cont.)



RIAL

Weights for CPU:MEM is 0.51:3.33

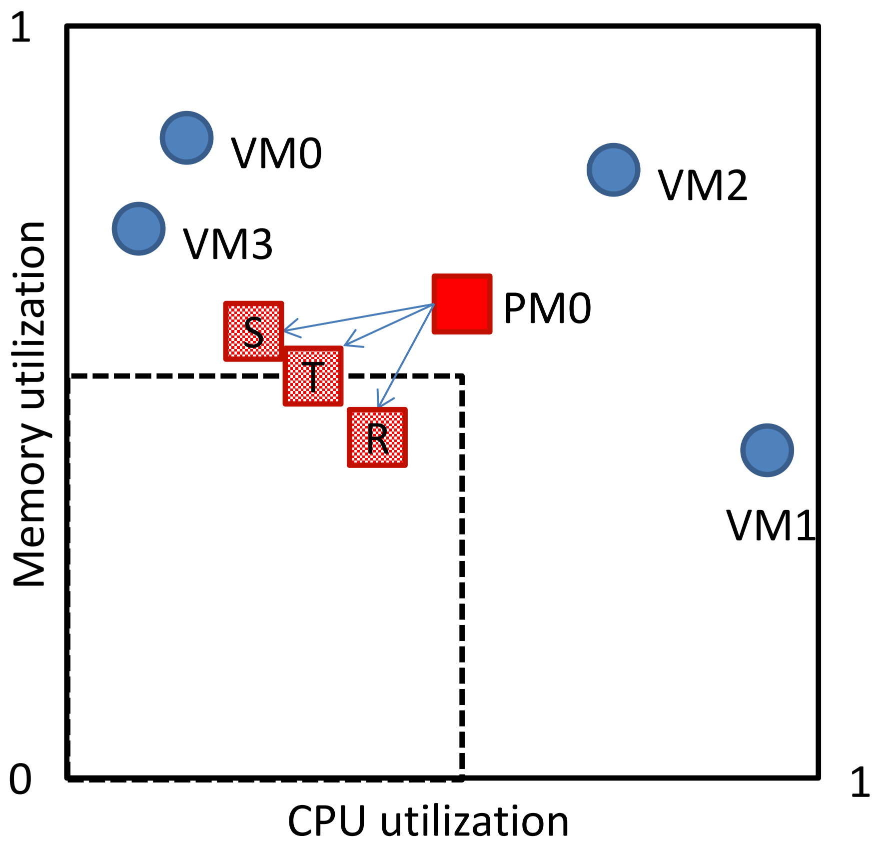
VM0 (0.20, 0.90)
VM1 (0.90, 0.40)
VM2 (0.75, 0.75)
VM3 (0.10, 0.75)

min ↓ ↓ max

Determine ideal VM R (0.10, 0.90)

VM0 is weighted closest to VM R,
VM0 are selected to migrate out

Design: example (cont.)



Sandpiper migrates VM1

TOPSIS migrates VM2

RIAL migrates VM0

Neither Sandpiper nor TOPSIS can eliminate memory overload in PM0, and hence additional VM migration is needed.

Performance Evaluation

- Simulation tool: CloudSim [1]
- Workload
  - CPU: VM utilization trace from PlanetLab
  - MEM: (mean, variance range) (0.2,0.05),(0.2,0.15),(0.3,0.05),(0.6,0.10),(0.6,0.15)
  - BW: random graph G(n,p=0.3) with random communication rate
- Profile
  - PM: 1GHz 2-core CPU, 1536MB memory, and 1GB/s network bandwidth
  - VM: 500Hz CPU, 512MB memory, and 100Mbit/s bandwidth

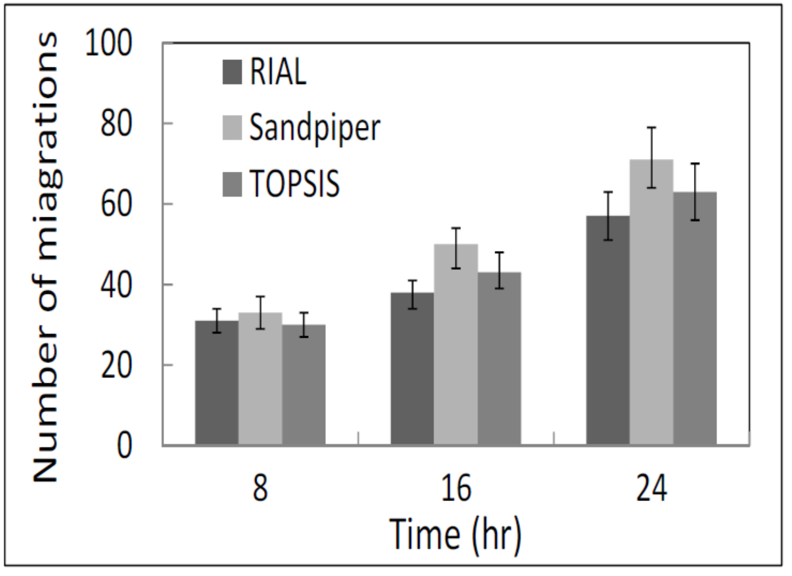
[1] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. D. Rose, and R. Buyya, "Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms." SPE, 2011.

Performance Evaluation

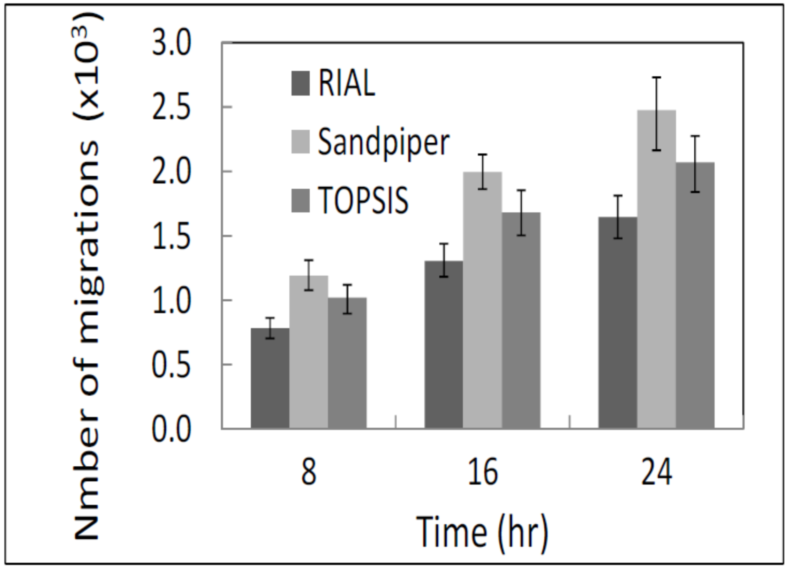
- Infrastructure: tree-like topology
- Two scale
  - Small scale: 250 VMs running on 100 PMs
  - Large scale: 5000 VMs running on 1000 PMs
- Comparison methods
  - Sandpiper [2]: assign the same weight to different resources
  - TOPSIS [3]: assign predefined weights to different resources

[2] T. Wood, P. J. Shenoy, A. Venkataramani, and M. S. Yousif, "Blackbox and gray-box strategies for virtual machine migration." in Proc. of NSDI, 2007.
[3] M. Tarighi, S. A. Motamedi, and S. Sharifian, "A new model for virtual machine migration in virtualized cluster server based on fuzzy decision making." CoRR, 2010.

The Number of Migrations



100 PMs and 250 VMs

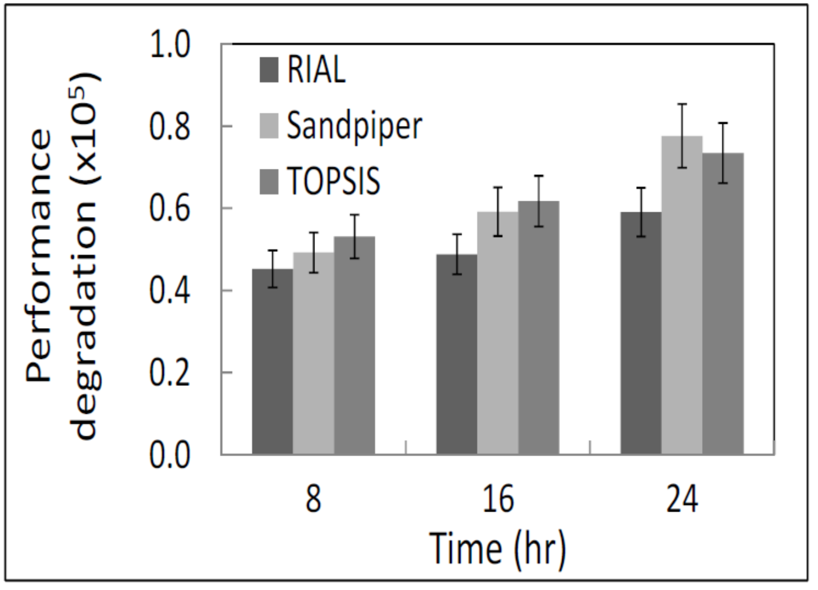


1000 PMs and 5000 VMs

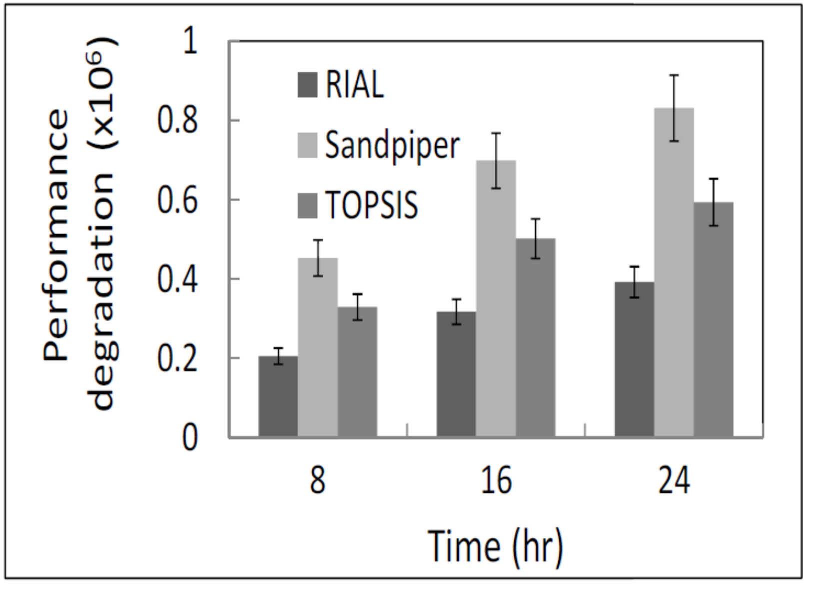
RIAL < TOPSIS < Sandpiper

The number of migrations increases with time and scale

VM Performance Degradation



100 PMs and 250 VMs

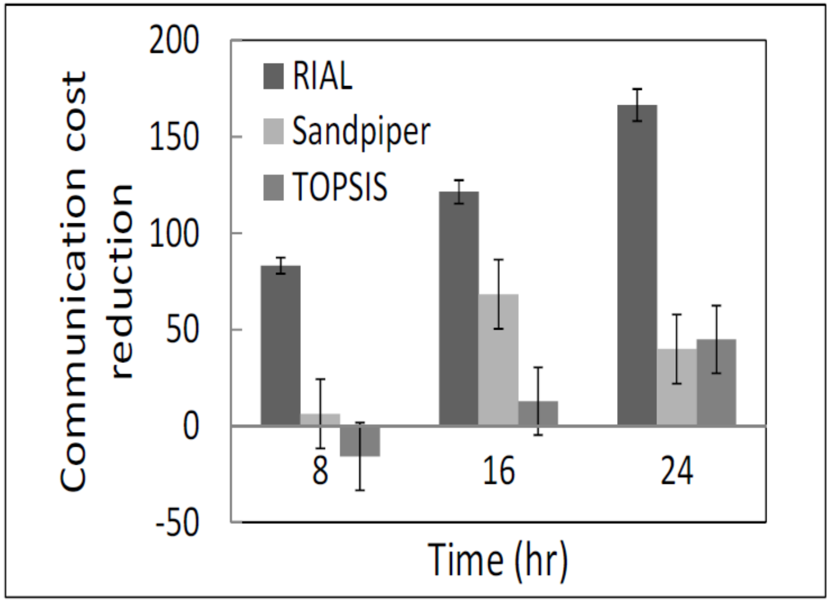


1000 PMs and 5000 VMs

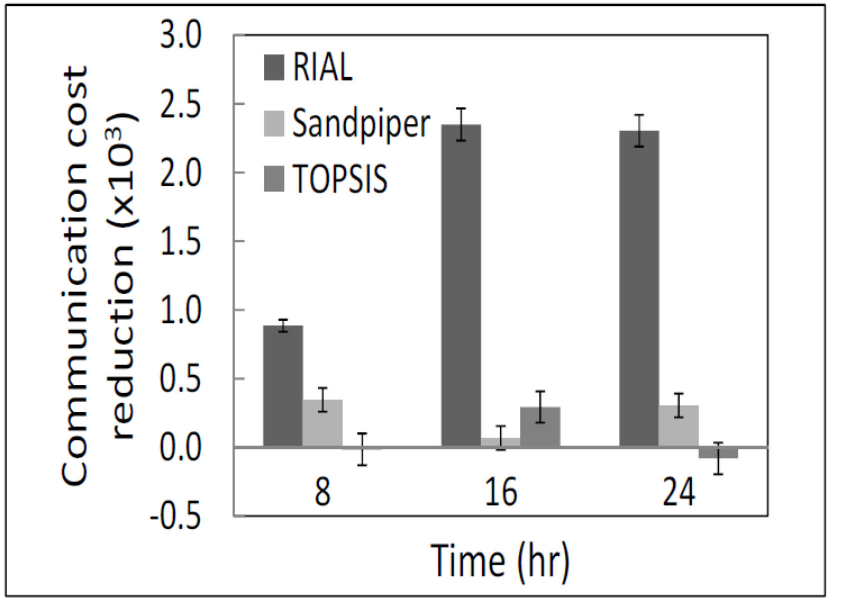
RIAL < TOPSIS < Sandpiper

$$D_{ijp} = \sum d_{ip} \cdot \int_{t}^{t+\frac{M_{ij}}{B_{ip}}} u_{ij}(t)\mathrm{d}t$$

VM Communication Cost Reduction



100 PMs and 250 VMs



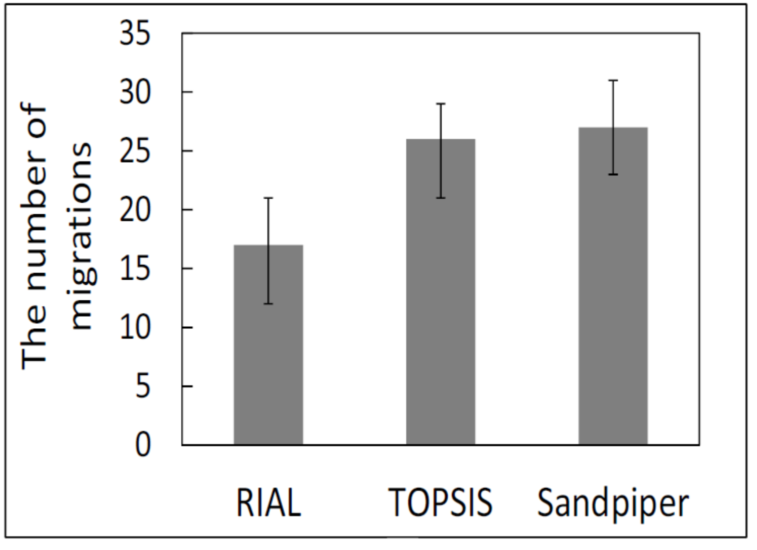
500 PMs and 1250 VMs

RIAL > TOPSIS
RIAL > Sandpiper

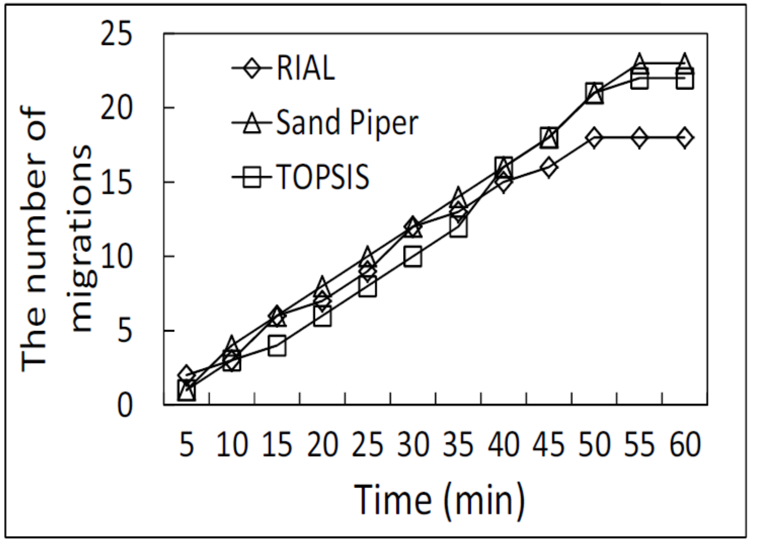
Cost = Communication rate X transmission delay

Performance Evaluation

- Testbed: 7 PMs cluster with XenServer 6.2
- Workload: lookbusy [4]



Number of migrations.

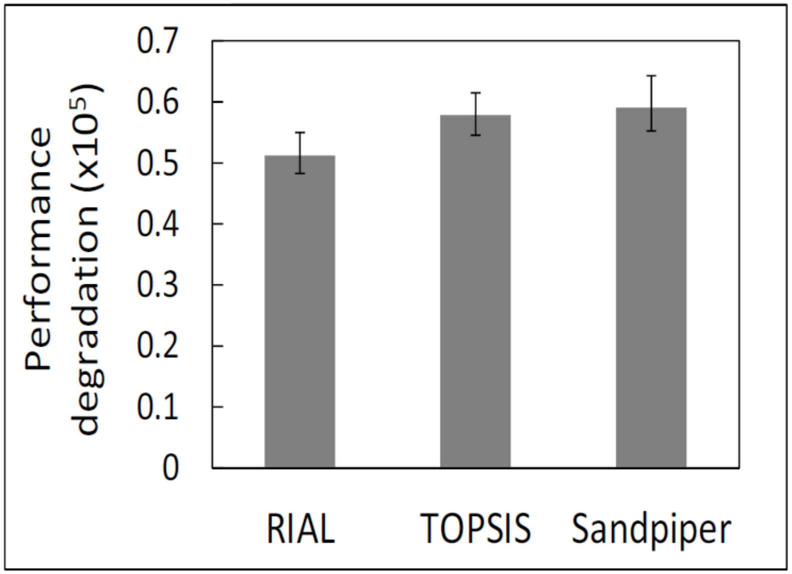


Accumulated # of migrations.

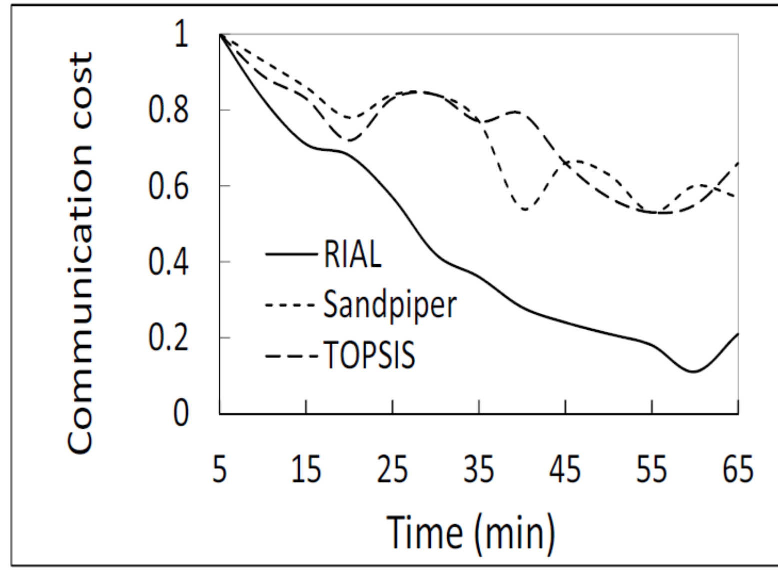
RIAL < TOPSIS < Sandpiper

[4] "lookbusy," http://devin.com/lookbusy/.

Performance Evaluation



Performance degradation.



Communication cost.

RIAL < TOPSIS < Sandpiper

RIAL < TOPSIS
RIAL < Sandpiper

Conclusions

- Propose a resource intensity aware load balancing algorithm
  - Assigning weights based on intensities
- Consider communication dependencies
  - Reduce communication cost
  - Reduce performance degradation
- Provide trace driven simulation and real-testbed experiment

- Future Work
  - Study performance with heterogeneous PMs
  - Achieve optimal tradeoff between overhead and effectiveness

Thank you!

Questions & Comments?

Liuhua Chen

liuhuac@clemson.edu

PhD candidate

Pervasive Communication Laboratory

Clemson University