Cross-layer Optimization for Virtual Machine Resource Management

Ming Zhao, Arizona State University
Lixi Wang, Amazon
Yun Lv, Beihang Universituy
Jing Xu, Google

http://visa.lab.asu.edu

VISA Virtualized Infrastructures, Systems, & Applications ASU

QoS in Clouds

- No support for QoS by existing cloud service providers
- Users have to pay for capacity, not performance
- But ultimately, users care about only performance

Amazon EC2 Service Level Agreement

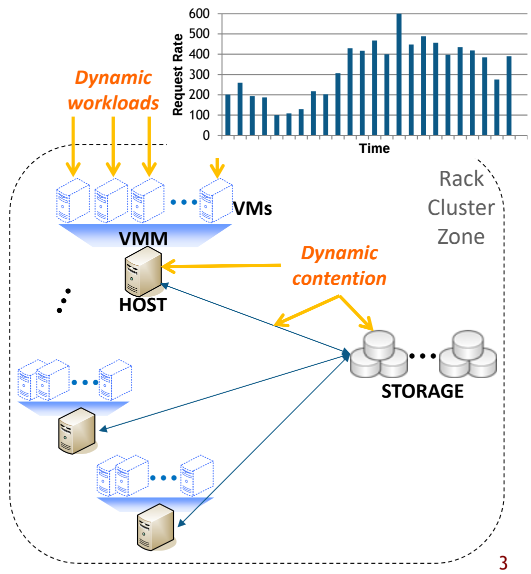
Monthly Uptime Percentage	Service Credit
<99.95% but ≥99.0%	10%
<99.0%	30%

Instance Type	vCPU	Memory (GB)	Storage (GB)	Clock Speed (GHz)
t2.micro	1	1	EBS Only	2.5
t2.small	1	2	EBS Only	2.5
t2.medium	2	4	EBS Only	2.5
…	…	…	…	…

VISA 2

Challenges to QoS Guarantees

- Dynamic VM workloads and dynamic resource contention
- But existing resource management treats VMs as black boxes
  - o Unable to adapt applications according to changing resource availability

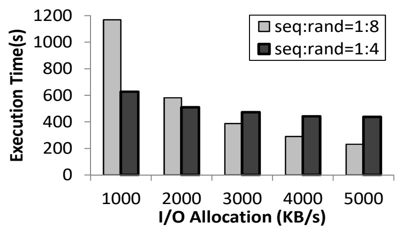


Motivations

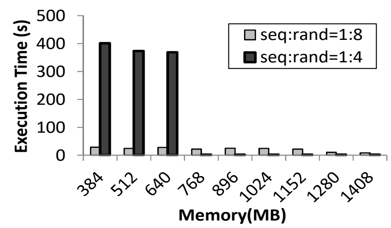
- Many applications need to be tuned according to resource availability, e.g.,
  - o A web server’s number of concurrent threads
  - o A database’s query cost estimation
  - o A search engine’s crawling, indexing, or searching strategy
  - o A simulator’s modeling resolution
  - o …
  - o Only need to tuned once on physical machines
- But when they are virtualized, they are stuck with their initial configurations
  - o VM-level resource contention is hidden to the applications

Motivating Examples

- When the database VM's resource allocation changes, different query optimization leads to different performance
  - E.g., *sequential_page_cost* and *random_page_cost* (seq:rand)



Varying VM disk bandwidth for TPC-H Q8



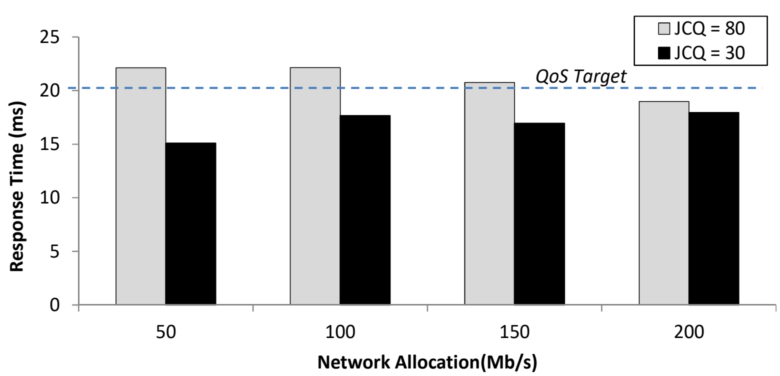
Varying VM memory allocation for TPC-H Q8

VISA

5

Motivating Examples

- When the map service VM's resource allocation changes, different map configuration decides response time
  - E.g., JPEG compression quality (JCQ)

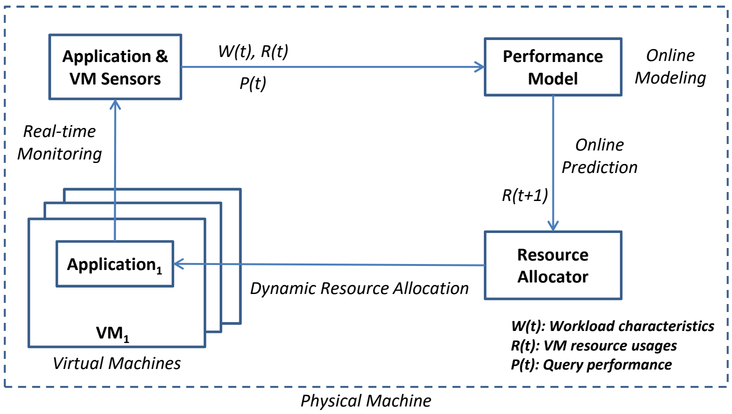


VISA

6

Typical VM Resource Management

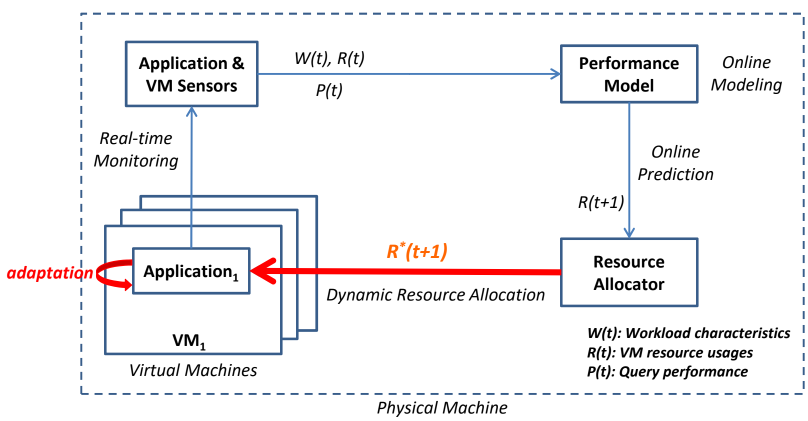
- VMs managed as blackboxes
- Applications unware of VM resource variability



VISA 7

Solution: Cross-layer Optimization

- Shares VM resource availability with the guests
- Adapts application configurations accordingly



VISA 8

Outline

- Introduction
- **Approach**
- Results
- Conclusions

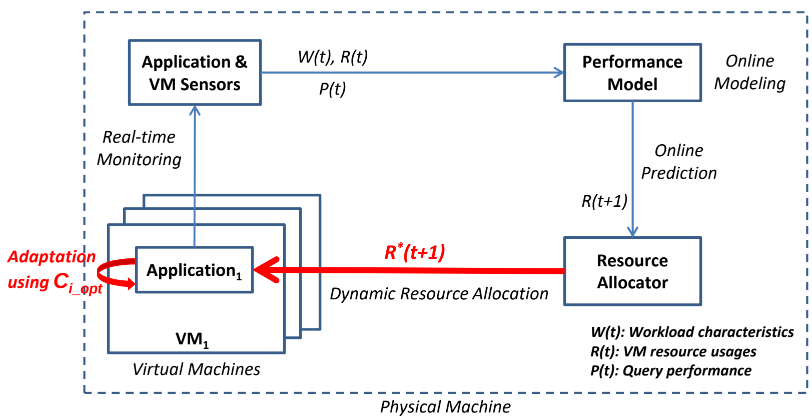
Research Questions

- *How to pass VM resource info from host to guest?*
  - o Through middleware running host and guests
  - o No change to applications or VM systems
- *How to adapt applications accordingly?*
  - o *What configuration parameters to tune?*
    - Based on application knowledge
    - Using machine learning methods (e.g., PCA)
    - Not the focus of this study
  - o *How to tune the parameters according to the VM resource availability?*

General Approach

- Goal: find the optimal configuration C_{i_opt} for application *i* that maximizes its performance P_i for a given VM resource allocation R_i
- Method:
  - o Create a model $C_i \rightarrow P_i$ for different resource allocations R_i
    - E.g., using regression analysis
  - o Use this mapping to find C_{i_opt} for any given resource allocation R_i

Dynamic Application Adaptation



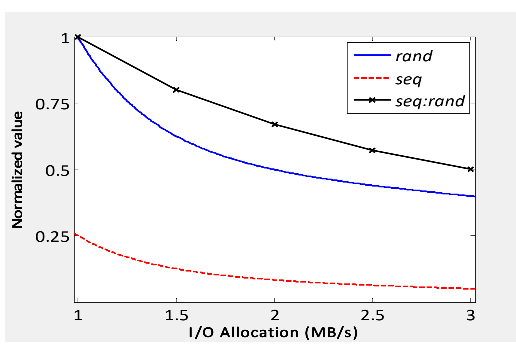
- Host-layer resource adjustment at fine time granularity (e.g., every 10s)
- Guest-layer adaptation at coarse time granularity (e.g., every minute)
- Performance model can be quickly updated (e.g., using fuzzy modeling)

Case Study: Virtualized Databases

- Databases represent applications with sophisticated internal optimizations
  - Query optimizer automatically evaluates the cost of different query execution plans and chooses the most efficient one
- Cross-layer optimization for virtualized databases
  - Adapts the query cost estimation and find the optimal query plan for its current resource availability
  - E.g., Adapts *sequential_page_cost* to *random_page_cost* ratio

Database $C_i \rightarrow P_i$ Model

- Run a simple query that reads a large table with either sequential- or index-scan methods
- Iterate with different I/O allocations
- Measure performance impact for each scanning method
- Build mapping between I/O allocations and the *rand:seq*



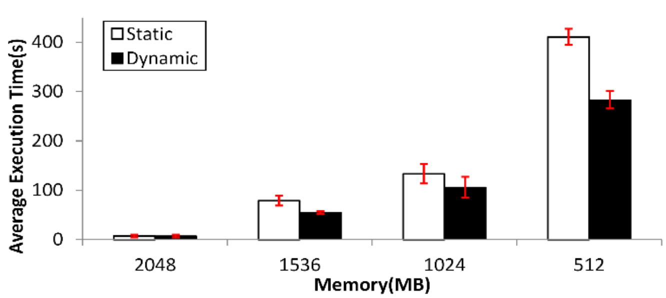
Evaluation

- Hardware: a server with 2 six-core Xeon processors, 32GB RAM, and 500GB SAS disk
- VM environment:
  - o Xen 3.3.1 with Ubuntu Linux
- Benchmark:
  - o TPC-H queries

VISA Ming Zhao, PhD 15

TPC-H

- *Dynamic*: dynamically adjust *seq:rand* based on VM's current resource availability
- *Static*: *seq:rand* fixed to 1:4

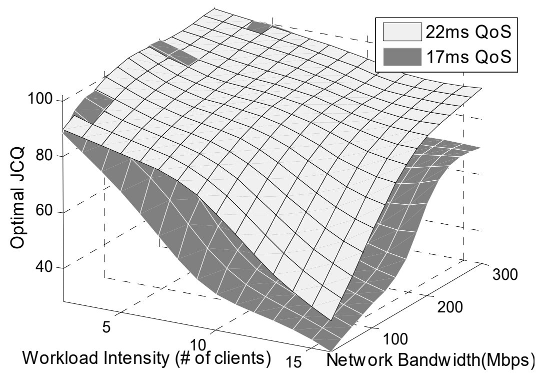


- *Dynamic* improves performance by 33.5%

VISA 16

Case Study: Virtualized Map Services

- Map services represent applications that can tune their QoS based on resource availability
  - E.g., JCQ affects response time and image quality
  - Higher JCQ → better map resolution, but more data transfer



VISA

17

Evaluation

- Hardware: a server with 2 six-core Xeon processors, 32GB RAM, and 500GB SAS disk
- VM environment:
  - Microsoft Hyper-V 6.2 with Windows Server
- Benchmark:
  - Terrafly: a production web-based map system

VISA

Ming Zhao, PhD

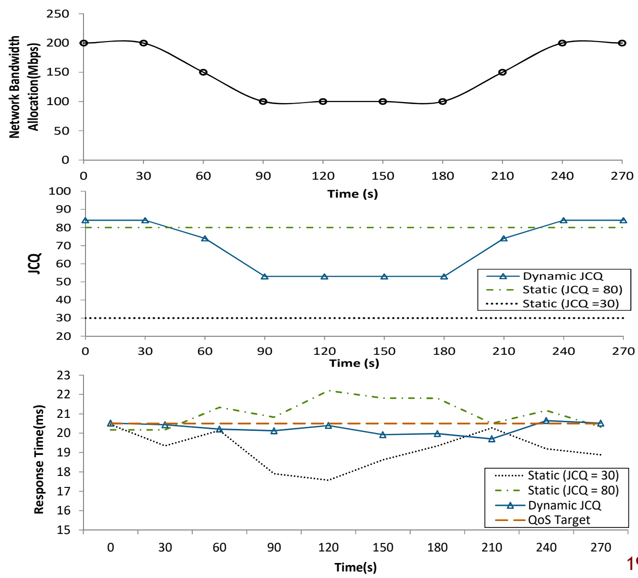
18

TerraFly with a Competing VM

Network bandwidth variation due to a competing FTP workload

Dynamic vs. static JCQ settings

Dynamic always meets SLO; improves image quality by 40%



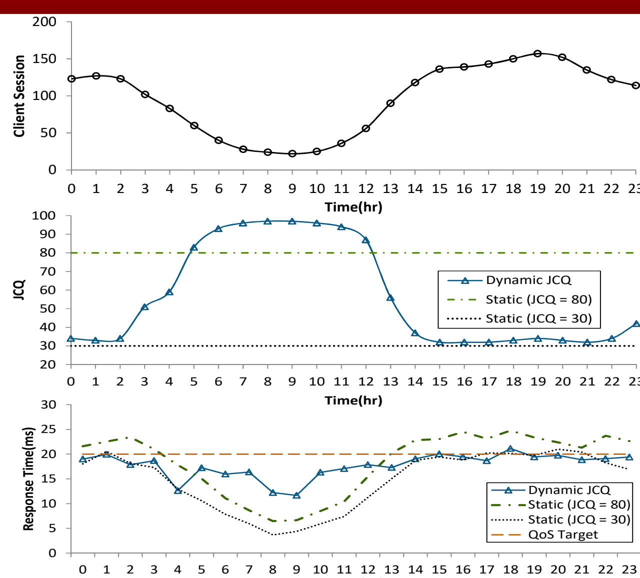
19

TerraFly with Varying Workload

Real workload variation

Dynamic vs. static JCQ settings

Dynamic always meets SLO; improves image quality by 26.3%



20

Conclusions

- Cloud is dynamic
  - o It is important to enable applications to adapt to changing resource availability in the cloud
- A systematic solution with cross-layer optimization
  - o 33.5% improvement in performance for TPC-H
  - o 40% in image quality for TerraFly
- Future work
  - o Distributed/parallel applications
  - o Integration with VM migrations

VISA Ming Zhao, PhD 21

Acknowledgement

- Sponsors
  - o National Science Foundation: CAREER award CNS-1619653, CNS-1629888, IIS-1633381, and CMMI-1610282
- VISA Research Lab
  - o http://visa.lab.asu.edu
- *Thanks! Questions?*

VISA