Matrix

Achieving Predictable Virtual Machine Performance in the Clouds

Ron C. Chiang, Jinho Hwang+, H. Howie Huang, and Timothy Wood

The George Washington University

IBM T.J. Watson Research Center+

ICAC'14

June 18 2014

The Intangible VM Performance

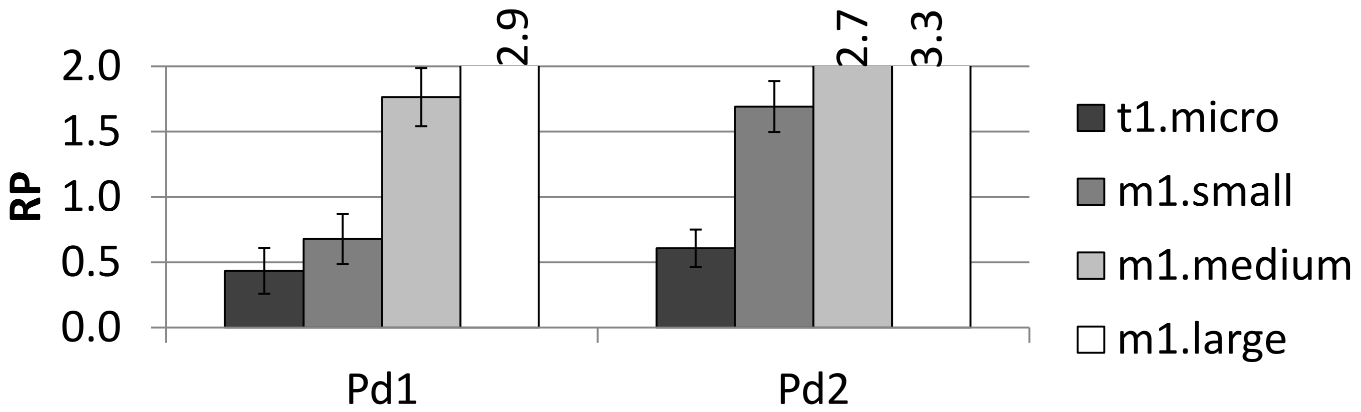
The same performance
The same look and feel

Try it in store **Buy online**

- Can I buy a VM that performs the same as my local machine?
- Not easy. Why?
  - Virtualization overhead
  - Resource contention
  - Limited control

Relative Performance (RP)

$$RP = \frac{P_{VM}}{P_d}$$



- Price chart of Amazon EC2 instances

Instances	micro	small	medium	large
Cost (cents/hour)	2	6	12	24

- Challenging to know the best tradeoff between cost and performance

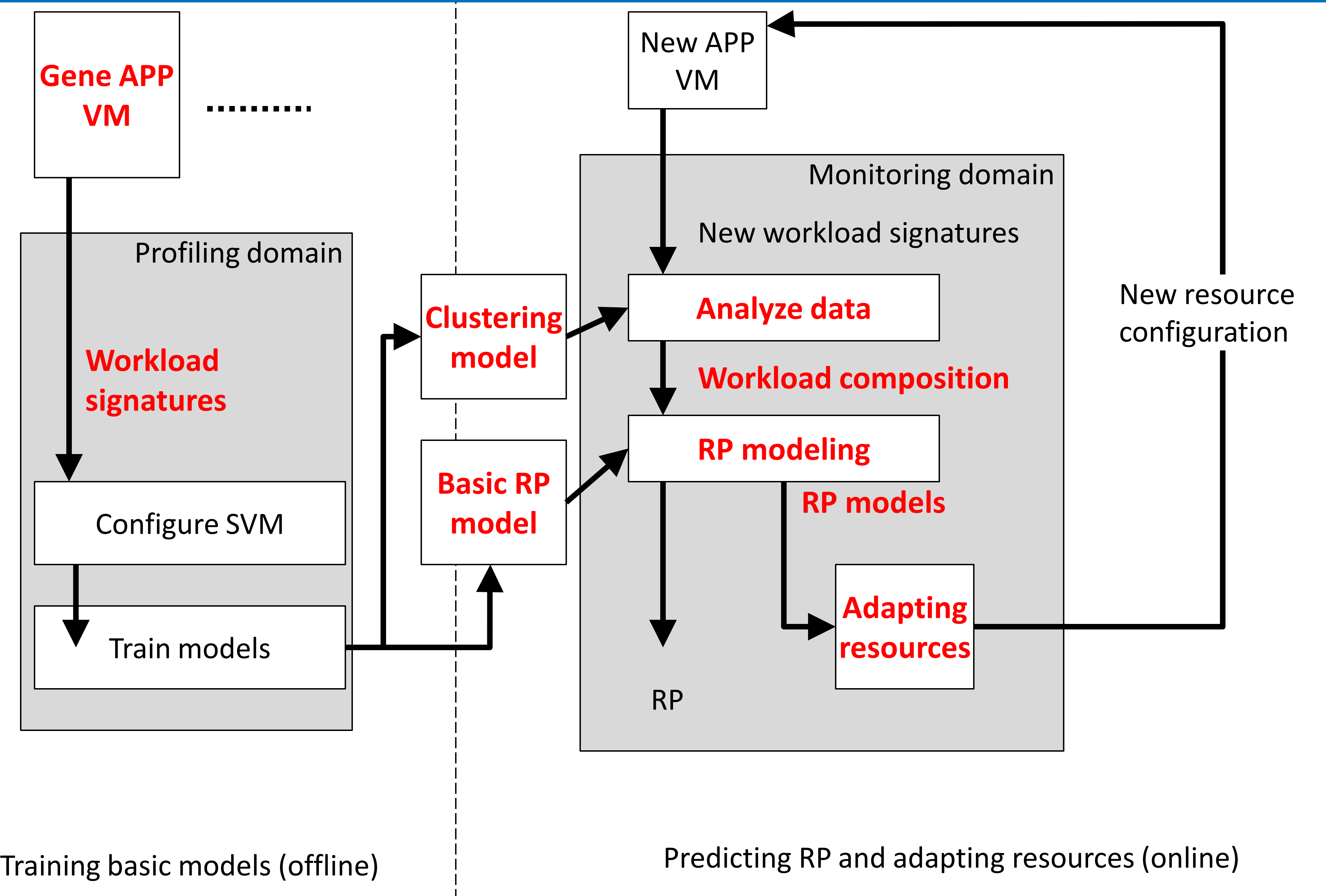
Challenges

- Keep desired performance with the best cost-efficiency?
- Characterize a new workload?
  - A set of representative workloads (gene app)
  - Soft boundary classification with probability estimates
- Handle various cloud environments?
  - Construct and verify models on a variety of clouds

Use Cases of Matrix

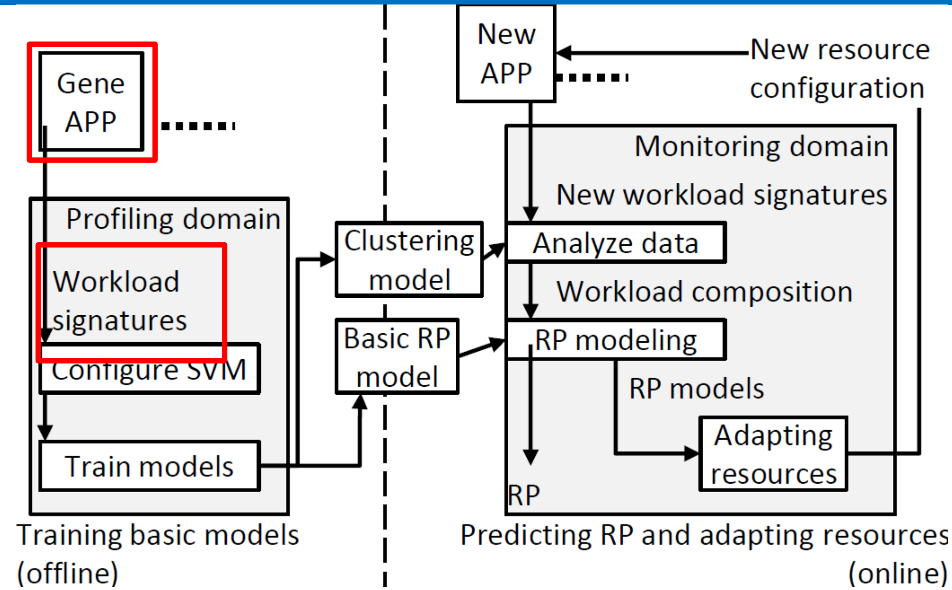
- Automatic VM configuration
  - Maintain desired performance with the best cost-efficiency
- VM instance recommendation
  - Recommend VM instance that is best suited for specific applications
- Cloud provider recommendation
  - Choose an appropriate VM from different cloud providers

Matrix Architecture



Gene App and Workload Signatures

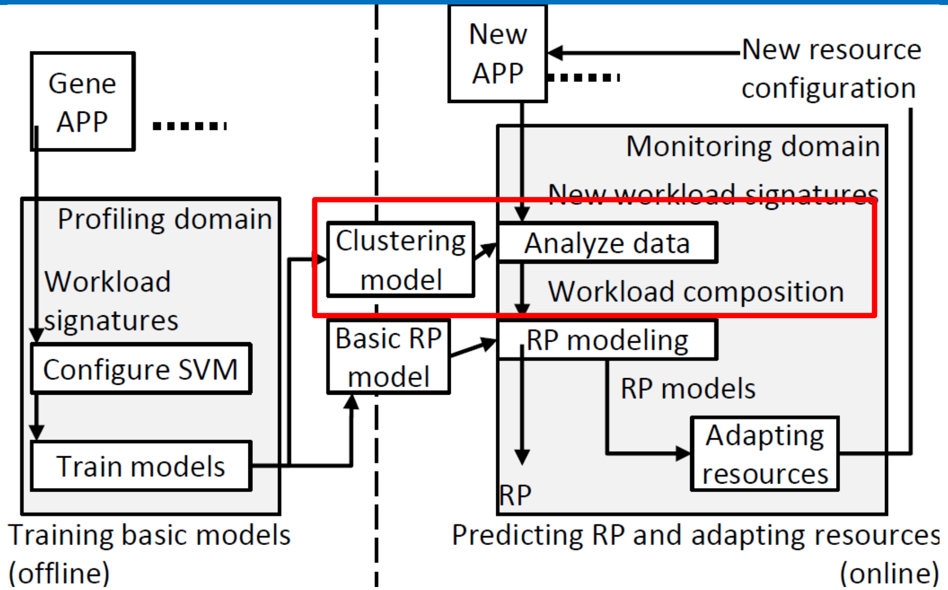
- As diverse as possible
  - from CPU intensive to data-intensive
  - problem sizes also shall vary from small to large



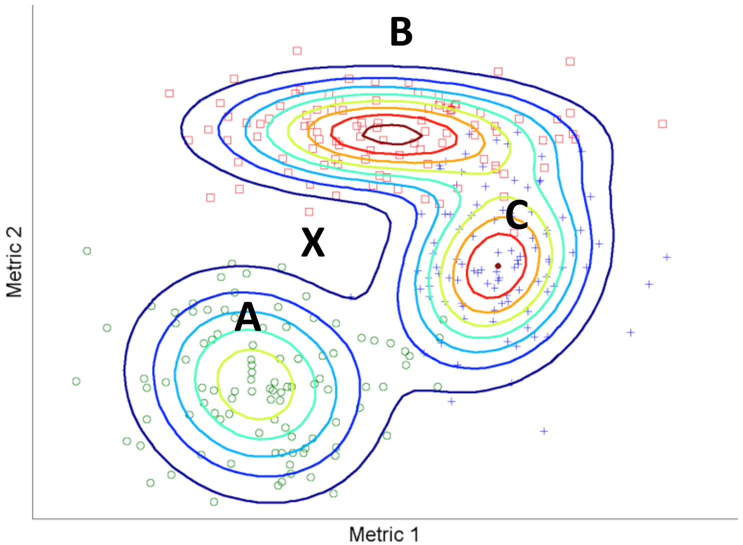
- The signature of a VM
  - *Resource allocation*: Number of VCPUs, VMs, Memory, etc.
  - *Resource demand*: Mean and C.O.V. of CPU, Memory, I/O, and Network usage
  - *Interference*: Number of background applications and their signatures

Classifiers with Probability Estimates

- Why probability estimates?
  - Countless workload signature combinations
  - No perfect cluster for all applications



- To analyze and represent a new application (the testing set) with genes (the training set)



- For example:

$$X = 0.5A + 0.2B + 0.3C$$

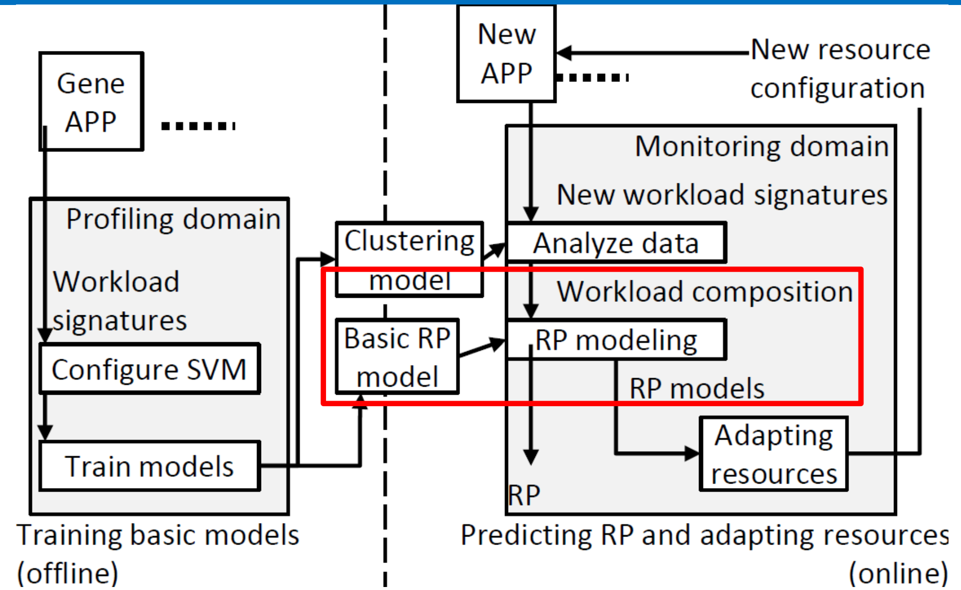
Generate New Apps' Performance Models

- Performance modeling by nu-SVR with RBF (Radial basis function) kernel

- The performance model of a gene app W_i

$$f_i(R) = \underline{\theta} \cdot \phi(r_i) + \theta_0$$

- Workload composition vector: $(p_1, \ldots, p_n)$
  - E.g., (0.5, 0.2, 0.3)



- The performance model of a new workload W_{new}

$$f_{new}(R) = \sum_{i=1}^{n} p_i \cdot f_i(R),$$

$$\text{where } \sum_{i=1}^{n} p_i = 1.$$

Keep RP=1 With Min Cost

Let C_j be the cost of resource j

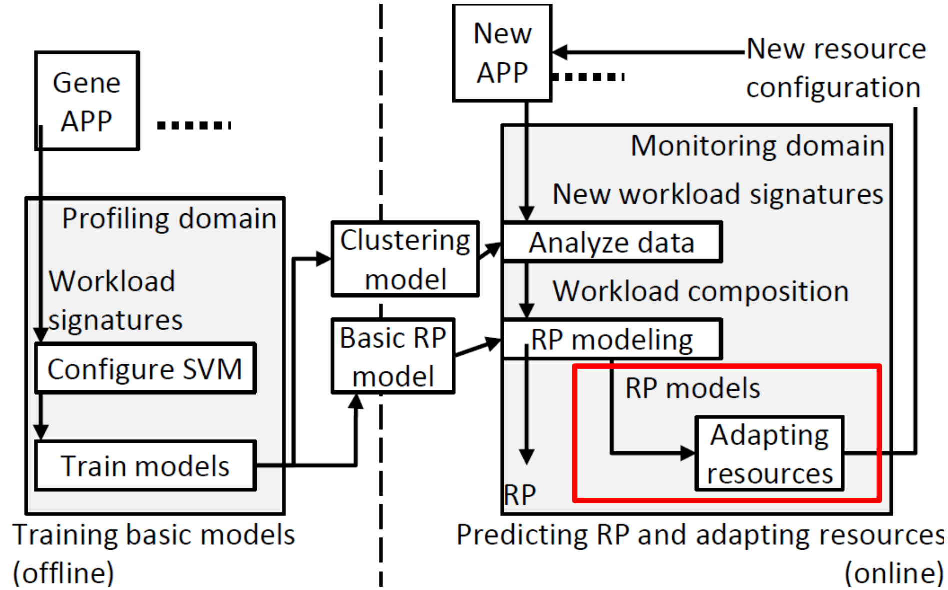
$$\underset{R}{\text{minimize}} \;\; F_c(R) = \sum_{j=1}^{m} C_j \times r_j$$

$$\text{subject to } f_{new}(R) = \sum_{i=1}^{n} p_i \cdot f_i(R) = 1,$$

$$\sum_{i=1}^{n} p_i = 1,$$

$$r_j = \{x \in \Re \mid 0 \le x \le 1\},$$

$$i \in \{1, \cdots, n\},\; j \in \{1, \cdots, m\}$$

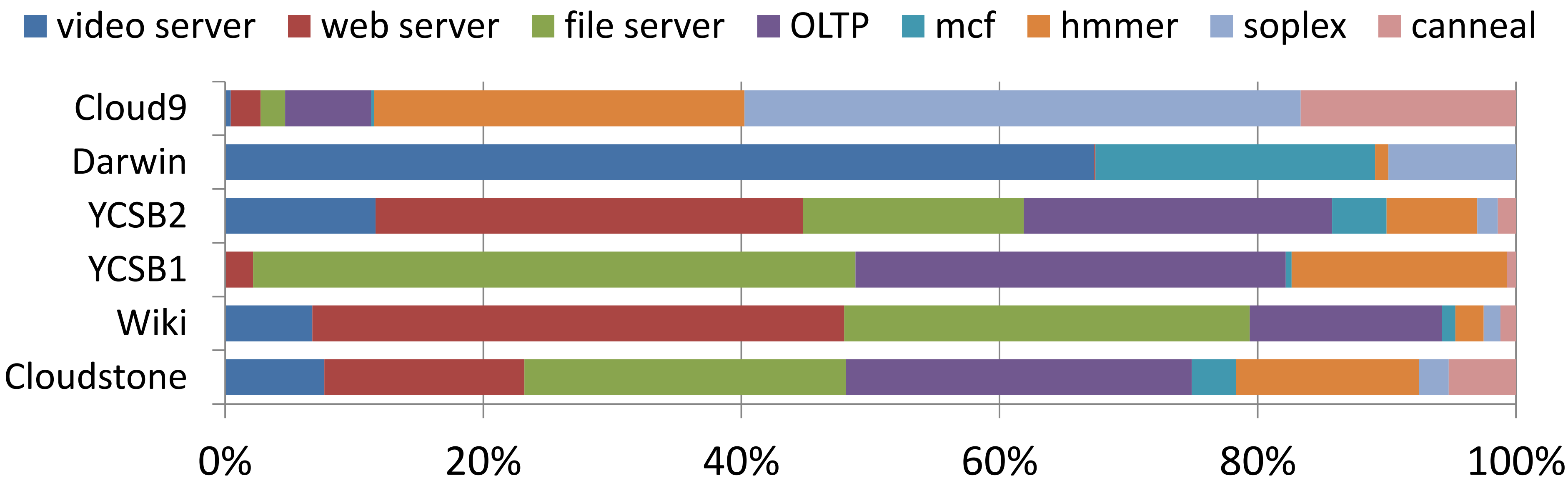


- Because of the RBF kernel
  - *f* is continuous and differentiable
  - Apply Lagrange algorithm to find *R*

Experiment Settings and Applications

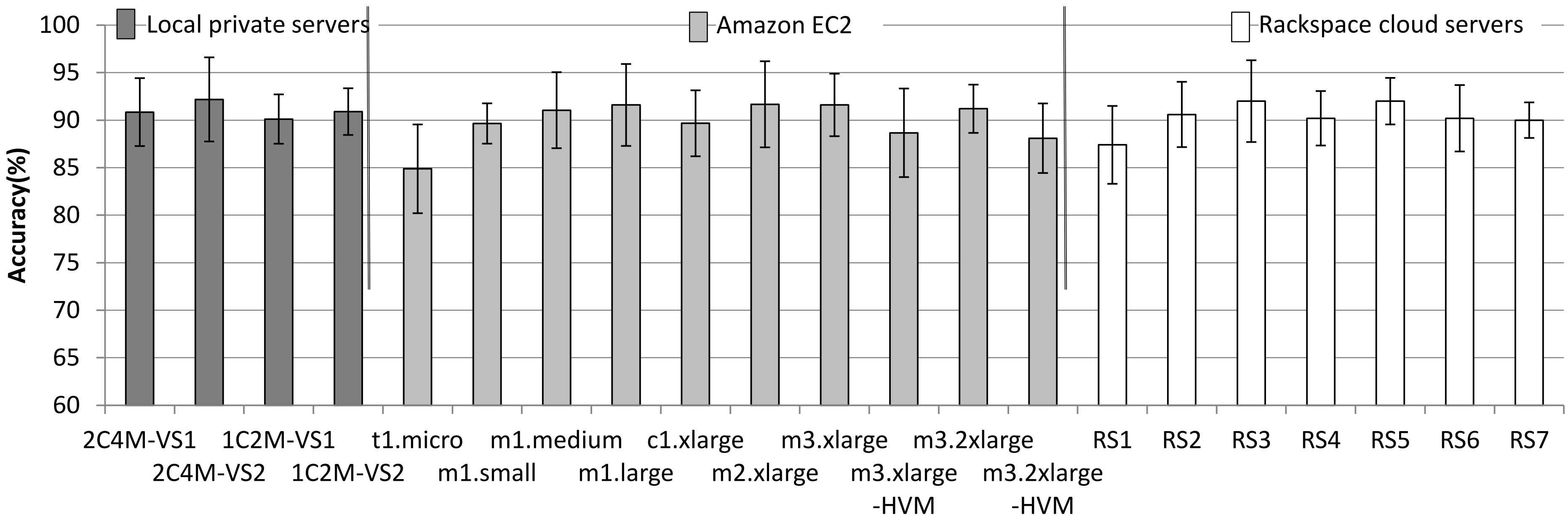
- Testing Environments
  - Each local host has two six-core Intel Xeon CPUs, Linux 2.6, Xen 4.0
  - For tests on public clouds, we use Amazon EC2 and Rackspace
- Testing Applications
  - *CloudStone*, a performance measurement framework for Web 2.0
  - *Wikipedia* with Database dumps from Wikimedia foundation and real request traces from the Wikibench
  - *Darwin*, an open source version of Apple's QuickTime video streaming server
  - *Cloud9* makes use of cloud resources to provide a high quality on-demand software testing service
  - *YCSB* (Yahoo! Coud Serving Benchmark), a performance measurement framework for cloud serving systems
    - Five workload characteristics YCSB1~YCSB5

Workload Composition Example



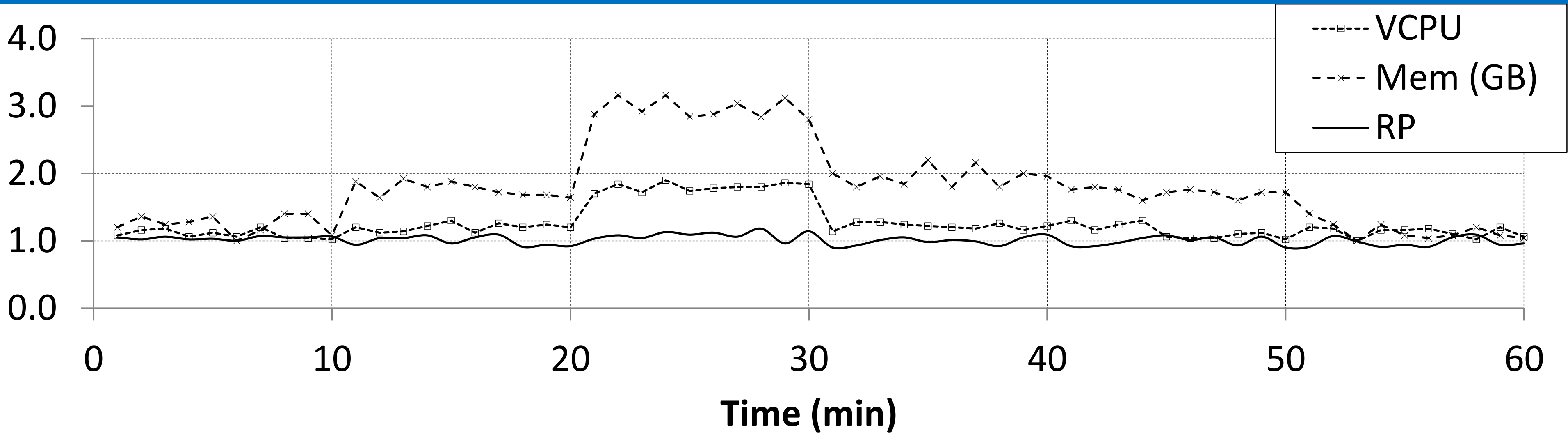
- The legend shows the training set
- The Y-axis shows the testing applications
- *Darwin*, a video streaming server, is not 100% like the *video server* from FileBench because the *video server* only emulates I/O operations
- The missing CPU activities are represented by other CPU-intensive apps.

Performance Prediction



- Accuracy = 1 - |predicted value – actual value|/ (actual value)
- X-axis shows VM configurations
- Most of the results are higher than 80%
- The overall average is 90.15%

Keep Desired Performance



- Running YCSB2 for one hour and change workload intensities every ten minutes
- Matrix adapts resource allocation to accommodate the changes
- The average RP over the whole testing time is 1.06

Choose VMs and Providers

Applications	Workload Intensities		
	Light	Medium	Heavy
CloudStone	RS3	RS3	m1.large
YCSB1	m1.small	m1.medium	m1.medium
YCSB2	RS2	m1.medium	m1.medium

- Light CloudStone uses RS3 because it gives higher RP with the same price
- Light YCSB1 chooses m1.small against RS2 because its memory space is larger and YCSB1 is sensitive to the heap size
- Medium YCSB1 and YCSB2 choose m1.medium against RS4 because it costs less although RS4 performs better
- Heavy CloudStone chooses m1.large against RS4 because of the higher performance with the same price

Cost Efficiency

- RP-Cost product (RPC)
  - Defined as |RP-1|X (VM cost)
  - The dollars waste on extra or insufficient performance
  - The smaller the better
- Performance Per Cost (PPC)
  - Defined as RP / (VM cost)
  - How much RP can you buy per dollar
  - The bigger the better
- Normalized to Matrix's values

	Matrix	4Xm1.small	4Xm1.medium	4Xm1.large
RPC	1	24.00	20.41	143.02
PPC	1	0.84	0.47	0.33

Related Work

- DejaVu and JustRunIt
  - Adapt resources to suit new demands
  - Require dedicated sandbox machines to clone and profile VMs

- DeepDive
  - Utilize mathematical models and clustering methods to detect interference
  - Require comparing the performance from VM clones in dedicated machines

Conclusion and Future Work

- Clustering methods with probability estimates can be used to classify new cloud workloads and recommend instance types
- Better cost-efficiency is achieved with an approximate optimization algorithm
- Future work
  - Including the price of data usage in the cost model
  - Heterogeneous machines and clusters

- E-mail: cchiang@stthomas.edu
- Web: https://sites.google.com/site/chilungchiang/