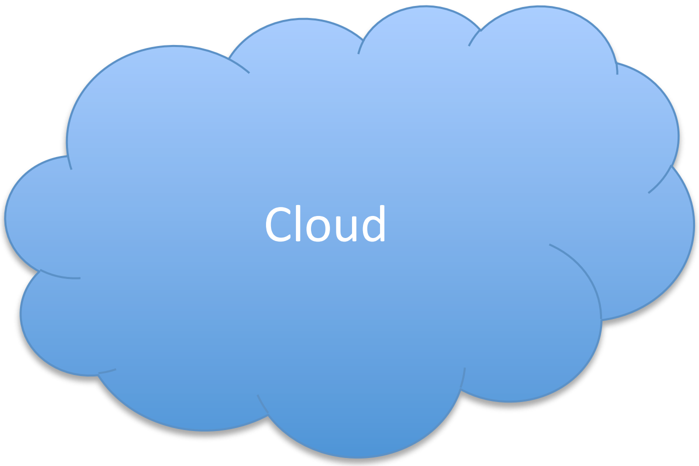
Modeling Virtualized Applications using Machine Learning Techniques

Sajib Kundu, Raju Rangaswami, Ming Zhao
Florida International University

Ajay Gulati
VMware

Kaushik Dutta
National University of Singapore

Hosting Applications in Cloud



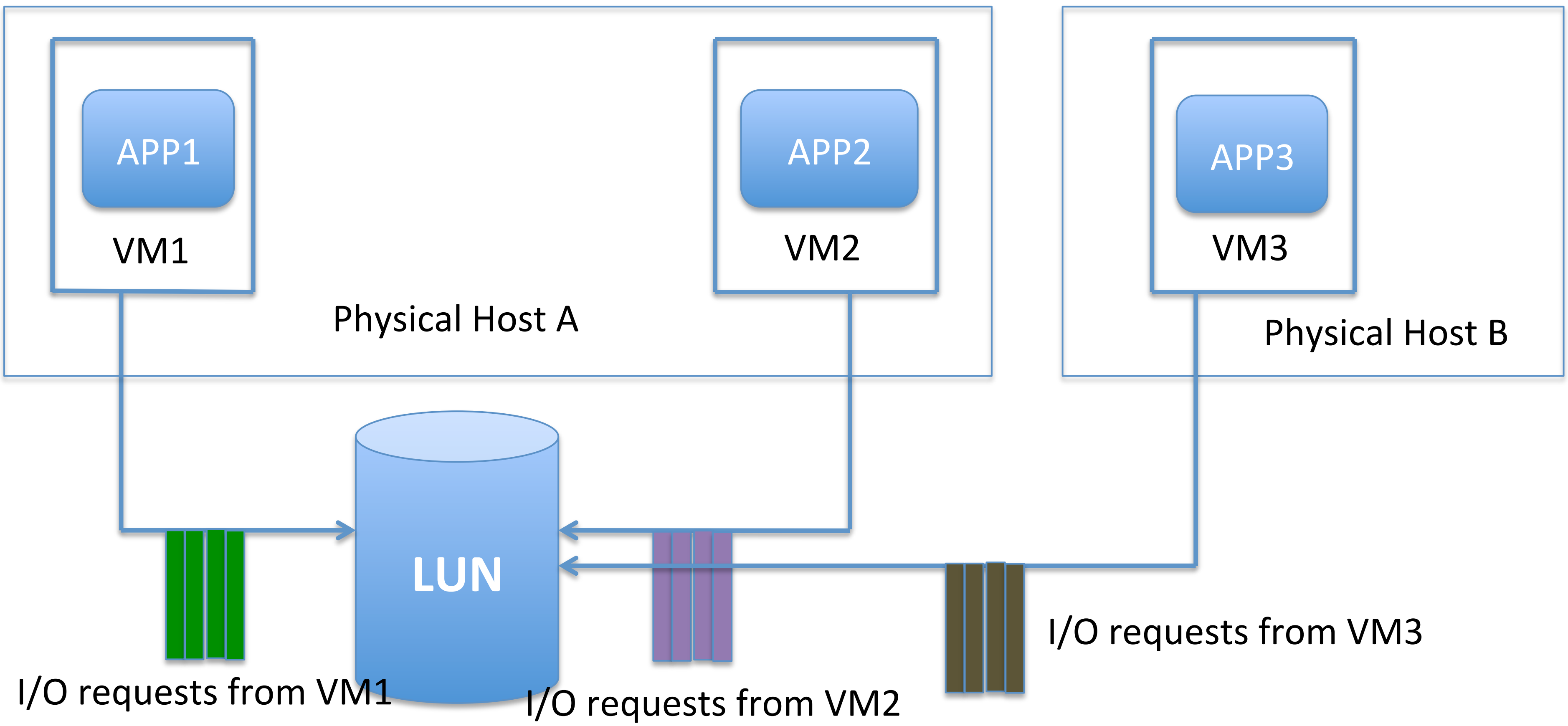
- Clients renting memory and computing capacity in the form of virtual machines (VMs) to host applications
- Customers charged based on capacity
- No performance guarantee

Client

- Over-provisioning
- Performance violations

How to size VMs?

A Key Question to Address



Performance interference on shared resources

Application Performance Modeling

- Investigating key resource parameters
- Modeling application performance
- Accounting for interference

But, modeling is challenging

Outline of the talk

- Related Work
- Model Parameters Selection
- Application Performance Model Design
- Evaluation
- VM Sizing
- Conclusion

Outline of the talk

- Related Work
- Model Parameters Selection
- Application Performance Model Design
- Evaluation
- VM Sizing
- Conclusion

Related Work

Modeling Technique	Applications

Related Work

Modeling Technique	Applications
Control Theory	Allocating CPU and memory using liner models [Padala et al.], Handling CPU interferences in cloud [Q-cloud]

Related Work

Modeling Technique	Applications
Control Theory	Allocating CPU and memory using liner models [Padala et al.], Handling CPU interferences in cloud [Q-cloud]
Regression Analysis	Modeling memory usage [CARVE], Fingerprinting datacenters [Bodik et al.]

Related Work

Modeling Technique	Applications
Control Theory	Allocating CPU and memory using liner models [Padala et al.], Handling CPU interferences in cloud [Q-cloud]
Regression Analysis	Modeling memory usage [CARVE], Fingerprinting datacenters [Bodik et al.]
Bayesian Networks	Constructing signatures of system history, Fingerprinting SLA violations [Cohen et al.]

Related Work

Modeling Technique	Applications
Control Theory	Allocating CPU and memory using liner models [Padala et al.], Handling CPU interferences in cloud [Q-cloud]
Regression Analysis	Modeling memory usage [CARVE], Fingerprinting datacenters [Bodik et al.]
Bayesian Networks	Constructing signatures of system history, Fingerprinting SLA violations [Cohen et al.]
Support Vector Machine (SVM)	Estimating power consumption [McCullough et al.]

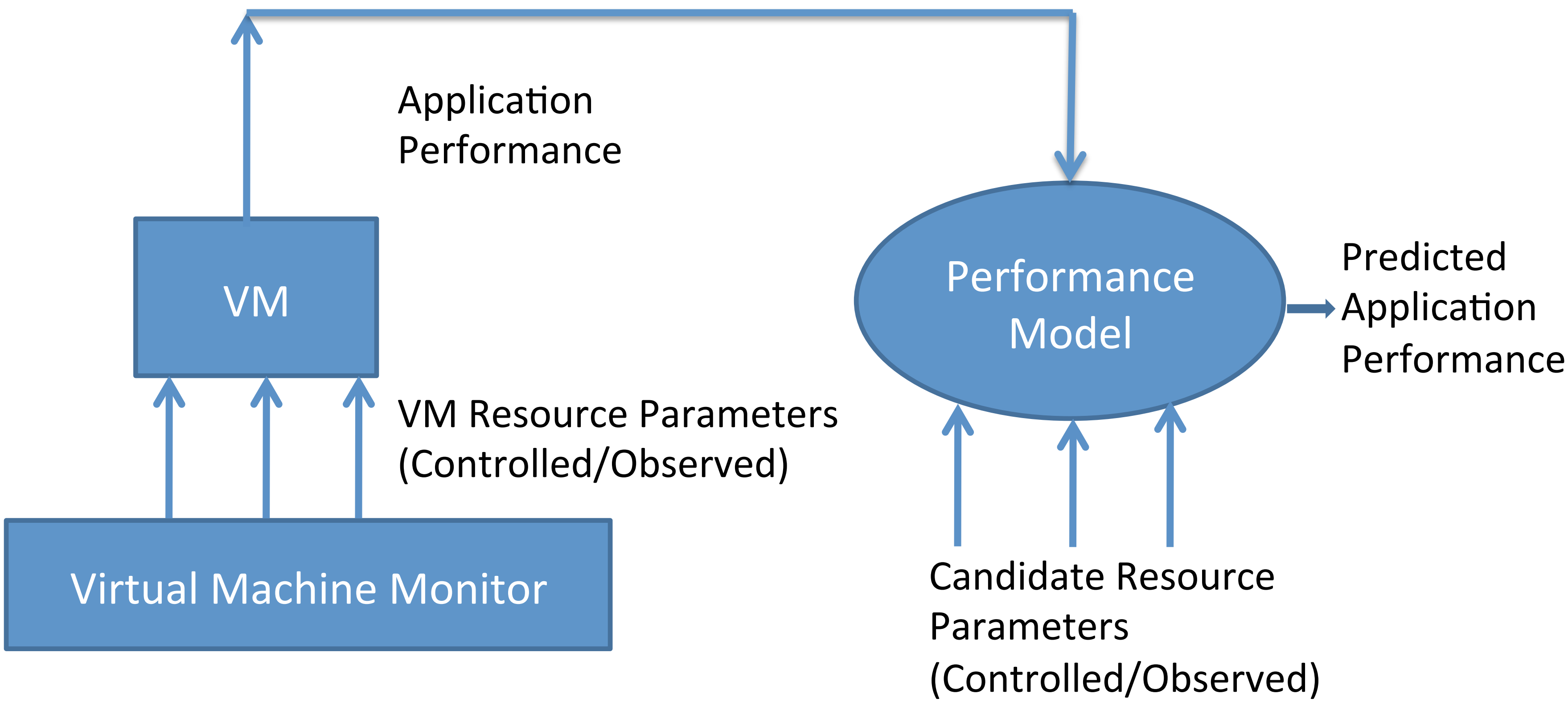
Related Work

Modeling Technique	Applications
Control Theory	Allocating CPU and memory using liner models [Padala et al.], Handling CPU interferences in cloud [Q-cloud]
Regression Analysis	Modeling memory usage [CARVE], Fingerprinting datacenters [Bodik et al.]
Bayesian Networks	Constructing signatures of system history, Fingerprinting SLA violations [Cohen et al.]
Support Vector Machine (SVM)	Estimating power consumption [McCullough et al.]
Artificial Neural Network (ANN)	Predicting performance for virtualized applications [Kundu et al.]

Outline of the talk

- Related Work ✔
- Model Parameters Selection
- Application Performance Model Design
- Evaluation
- VM Sizing
- Conclusion

Overview of Prediction Model



Parameters Selection Criteria

- Map to known resource usage behavior
- Easy to control/observe
- Account competition in shared environment
- Minimalistic set
- Application agnostic
- Widely Available

Parameter Selection

CPU

- Utilization
  vs.
  Allocation ✔
- CPU Limit chosen as the control knob

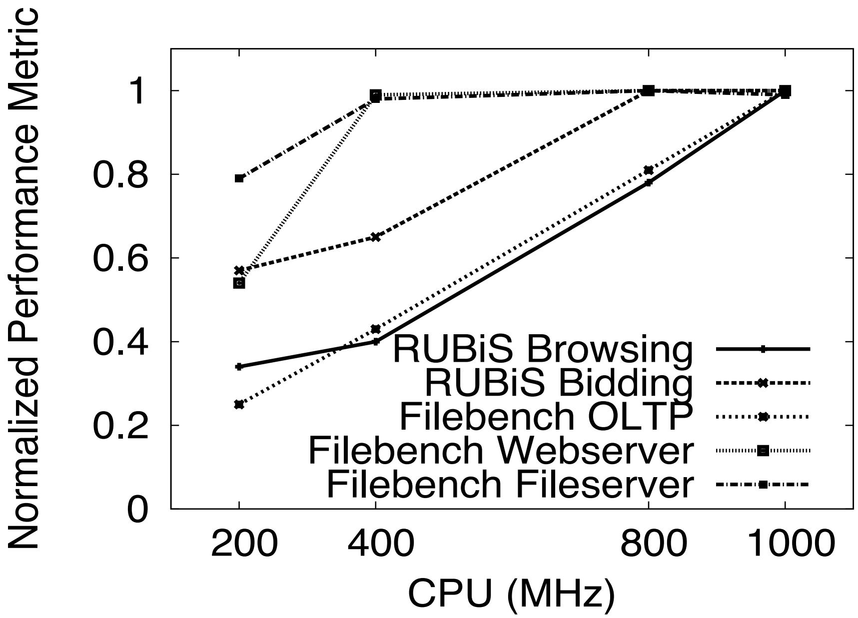
Memory

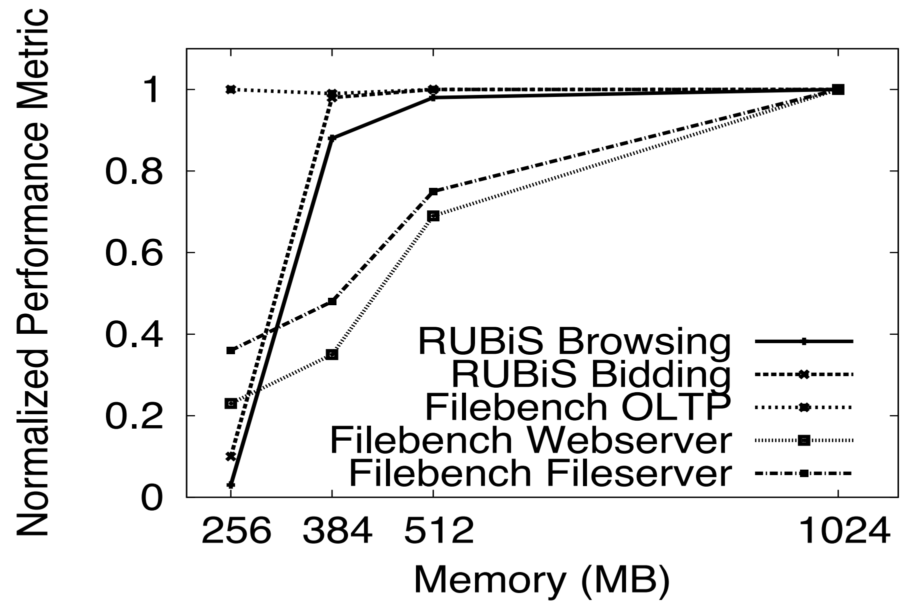
- Utilization
  vs.
  Allocation✔
- Memory Limit chosen as the control knob

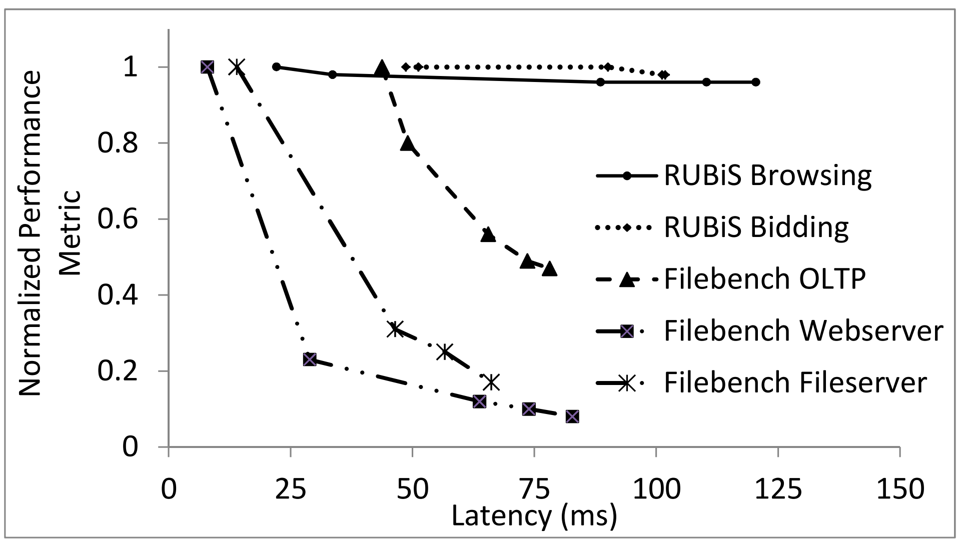
Storage

- No strong isolation
- Modeling interference
- Any appropriate metric?
- VM I/O Latency

Effects of Model Parameters on Performance





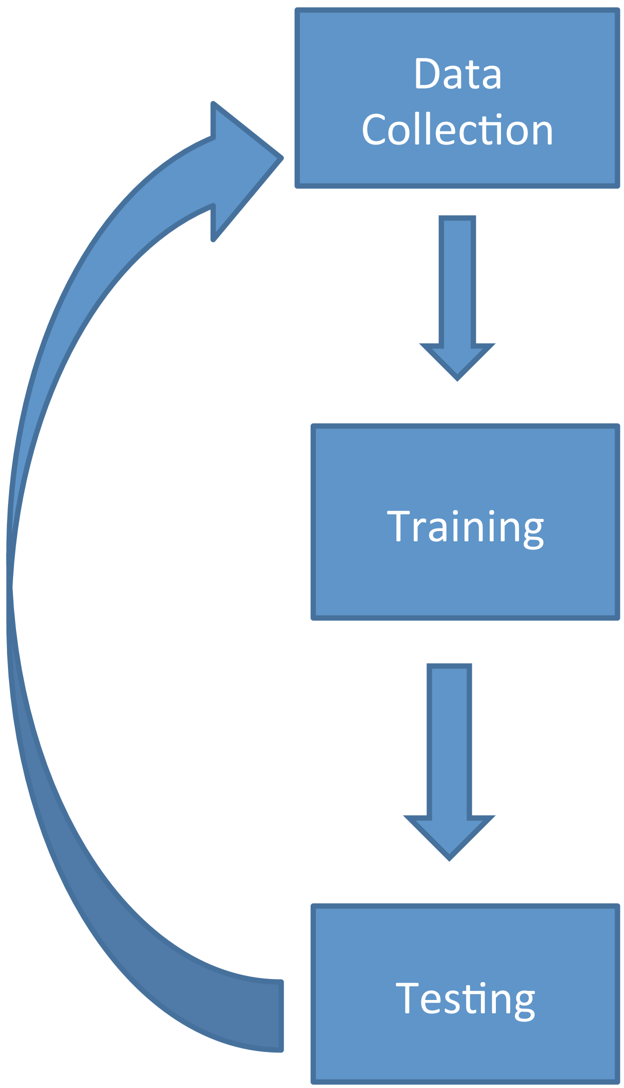


- Widely Varying across application types
- Non-linearity

Outline of the talk

- Related Work ✔
- Model Parameters Selection ✔
- Application Performance Model Design
- Evaluation
- VM Sizing
- Conclusion

Modeling: Steps



Modeling Techniques

Regression Analysis

- Linear Regression (LR)
- LR with quadratic terms
- LR with pairwise interactive terms

Limited power of modeling complex non-linear trends

Evolutionary Tools

- Artificial Neural Network (ANN)
- Support Vector Machine (SVM) Regression

Capable of modeling complex characteristics

Naïve Application of Machine Learning Models

Benchmark	%Avg.	90p.
RUBiS Browsing	68.57	340.00
RUBiS Bidding	19.30	60.18
Filebench OLTP	11.59	21.08
Filebench Webserver	19.85	38.60
Filebench Fileserver	12.89	28.78

Percentage Prediction Error Statistics Using a Single ANN Model

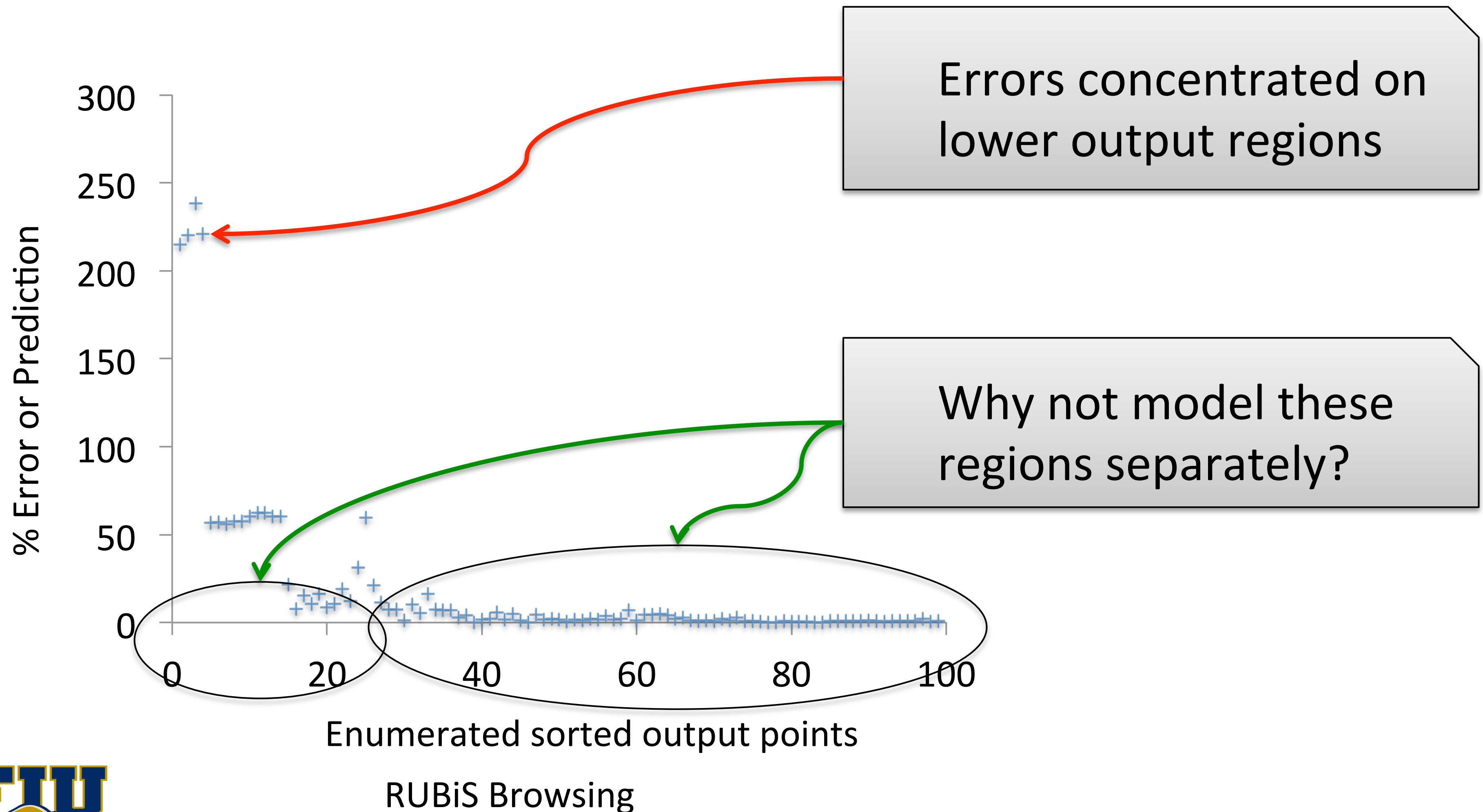
Naïve Application of Machine Learning Models

Benchmark	%Avg.	90p.
RUBiS Browsing	**68.57**	**340.00**
RUBiS Bidding	**19.30**	**60.18**
Filebench OLTP	11.59	21.08
Filebench Webserver	19.85	38.60
Filebench Fileserver	12.89	28.78

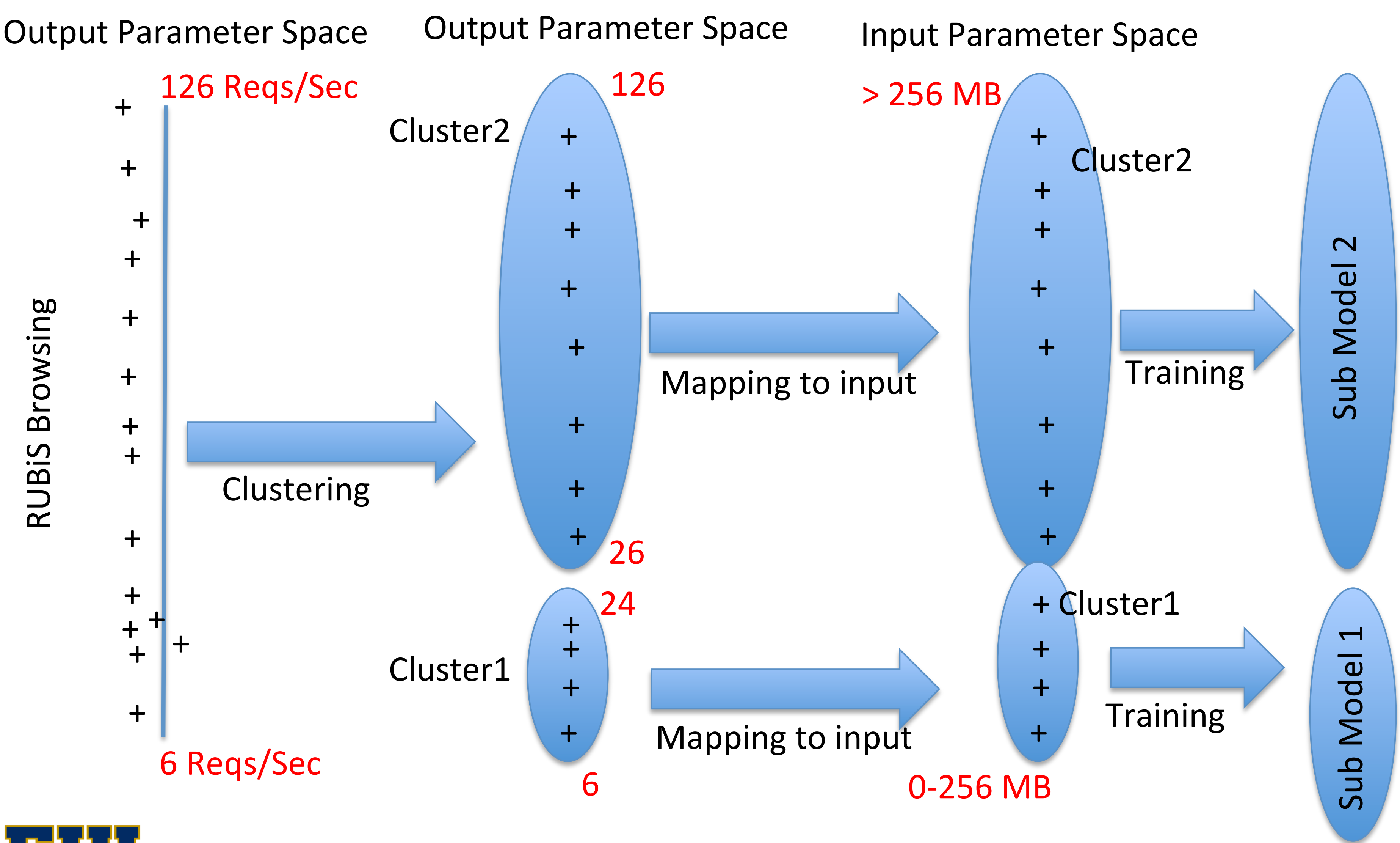
Percentage Prediction Error Statistics Using a Single ANN Model

- ❖ Simple application of ANN not enough
- ❖ Errors too high for some workloads

Distribution of Errors using Single Model



Creation of Sub-Models



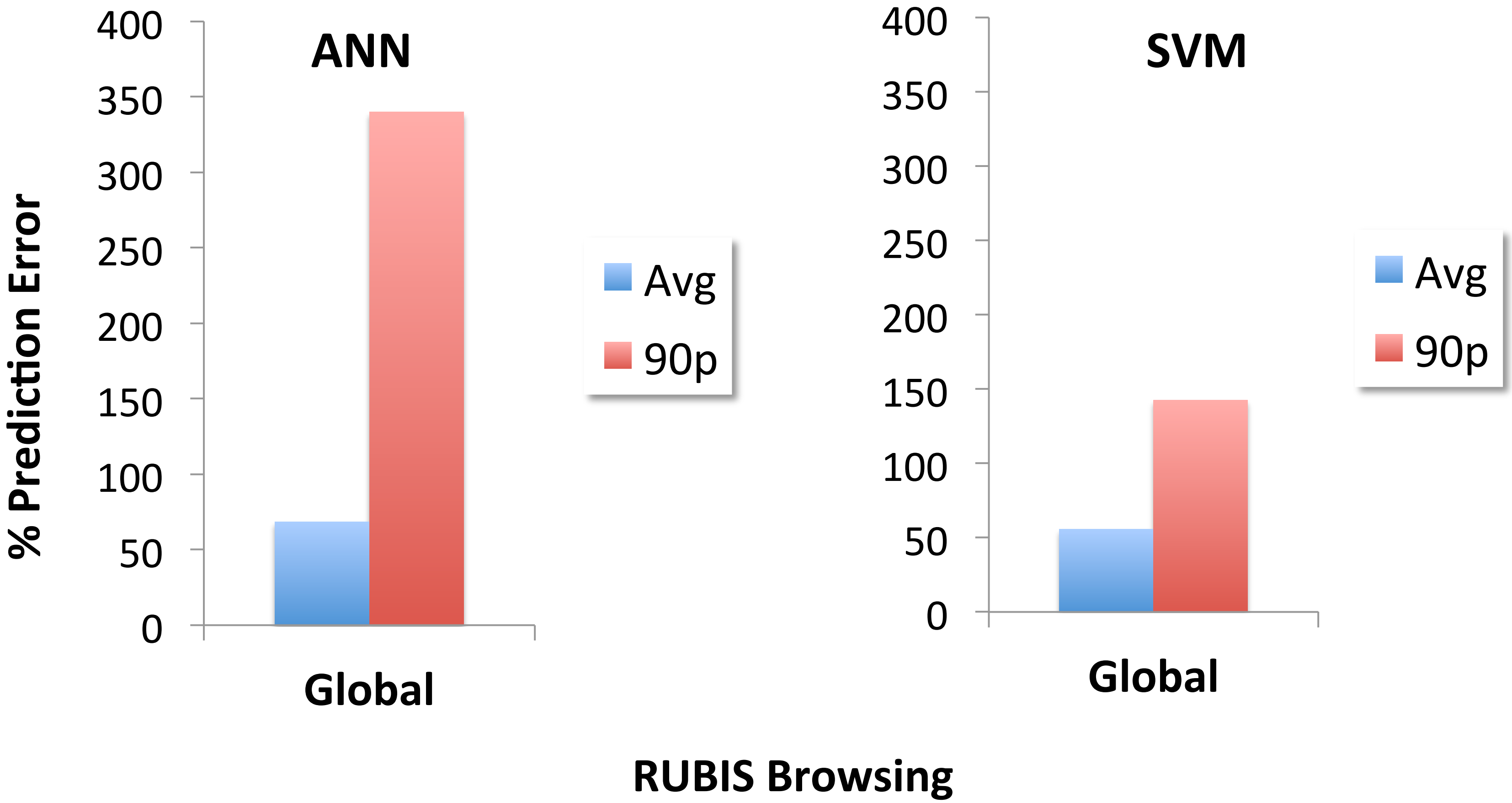
Outline of the Talk

- Related Work ✔
- Model Parameters Selection ✔
- Application Performance Model Design ✔
- Evaluation
- VM Sizing
- Conclusion

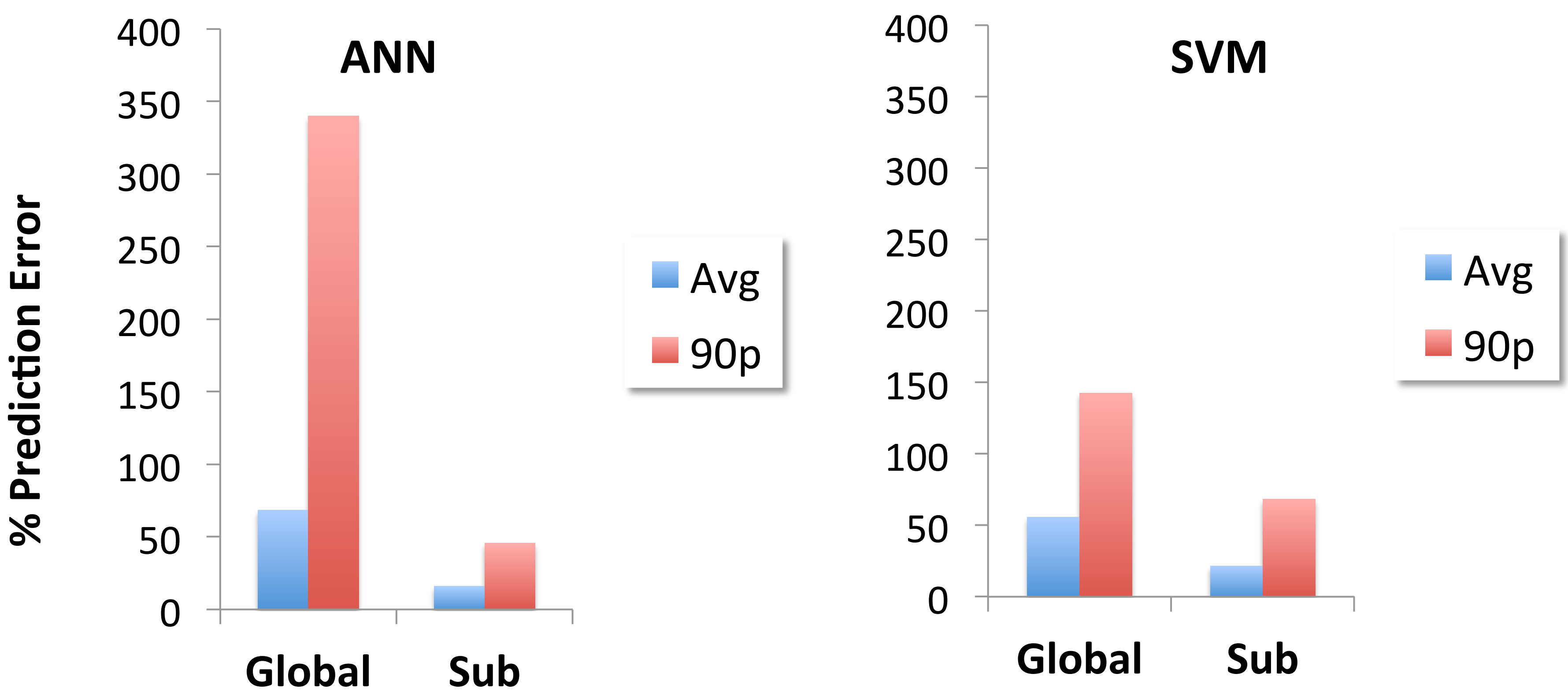
Experimental Set Up

Hardware	AMD-based Dell PowerEdge, 12×2.4GHz CPU, 32 GB Memory
Hypervisor	VMware ESX 4.1
Guest OS	Ubuntu-Linux 10.04
LUN	Local 7200 RPM SAS drive
I/O Contention Generator	Running *fio* in a large VM

Does Sub-Modeling Help?



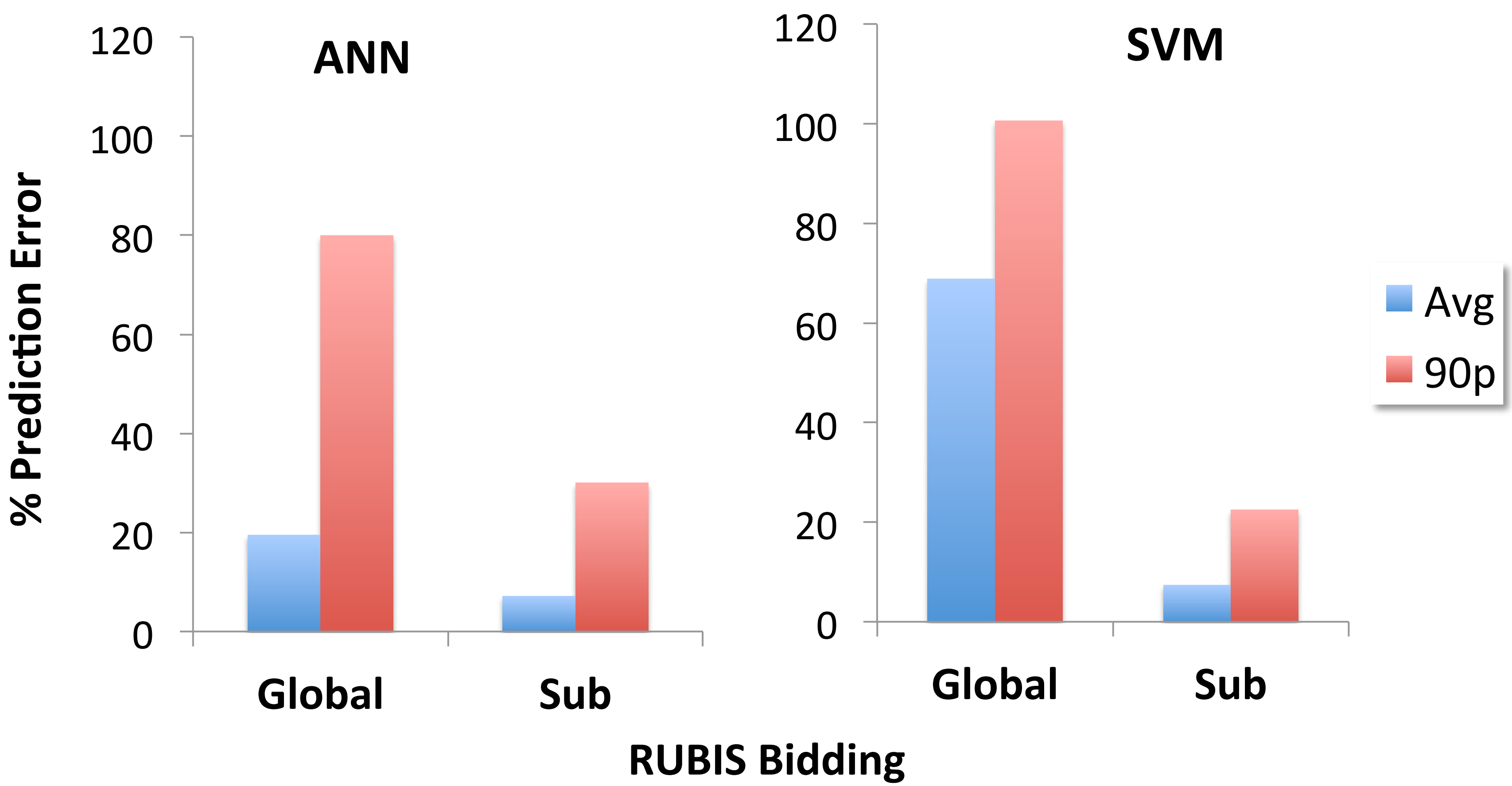
Does Sub-Modeling Help?



RUBIS Browsing

Using sub-modeling with ANN
- Avg. error reduces from 69% to 16%
- 90p error reduces from 340% to 46%

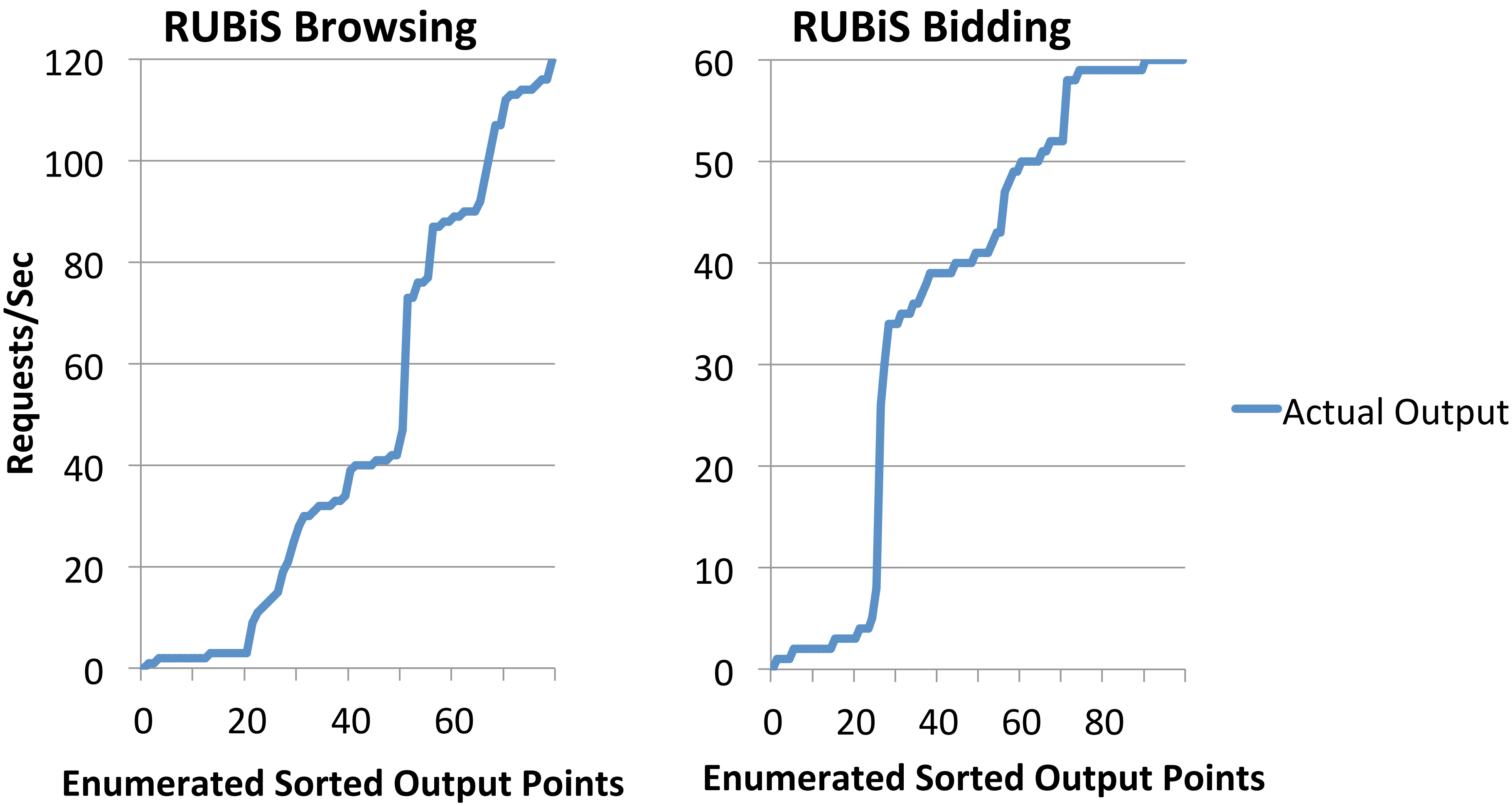
What about another Workload?



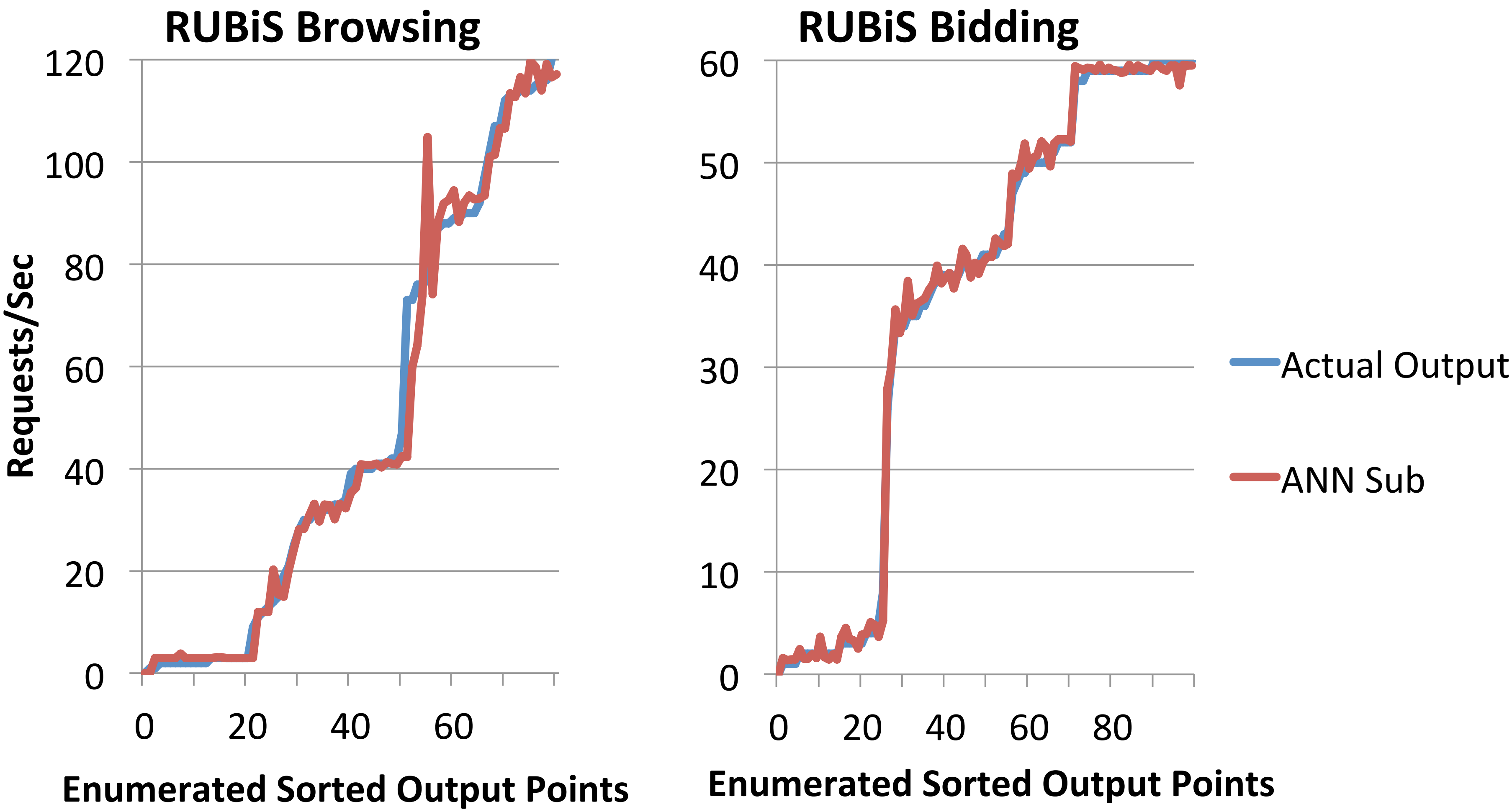
Using sub-modeling with SVM

- Avg. error reduces from 69% to 7%
- 90p error reduces from 101% to 23%

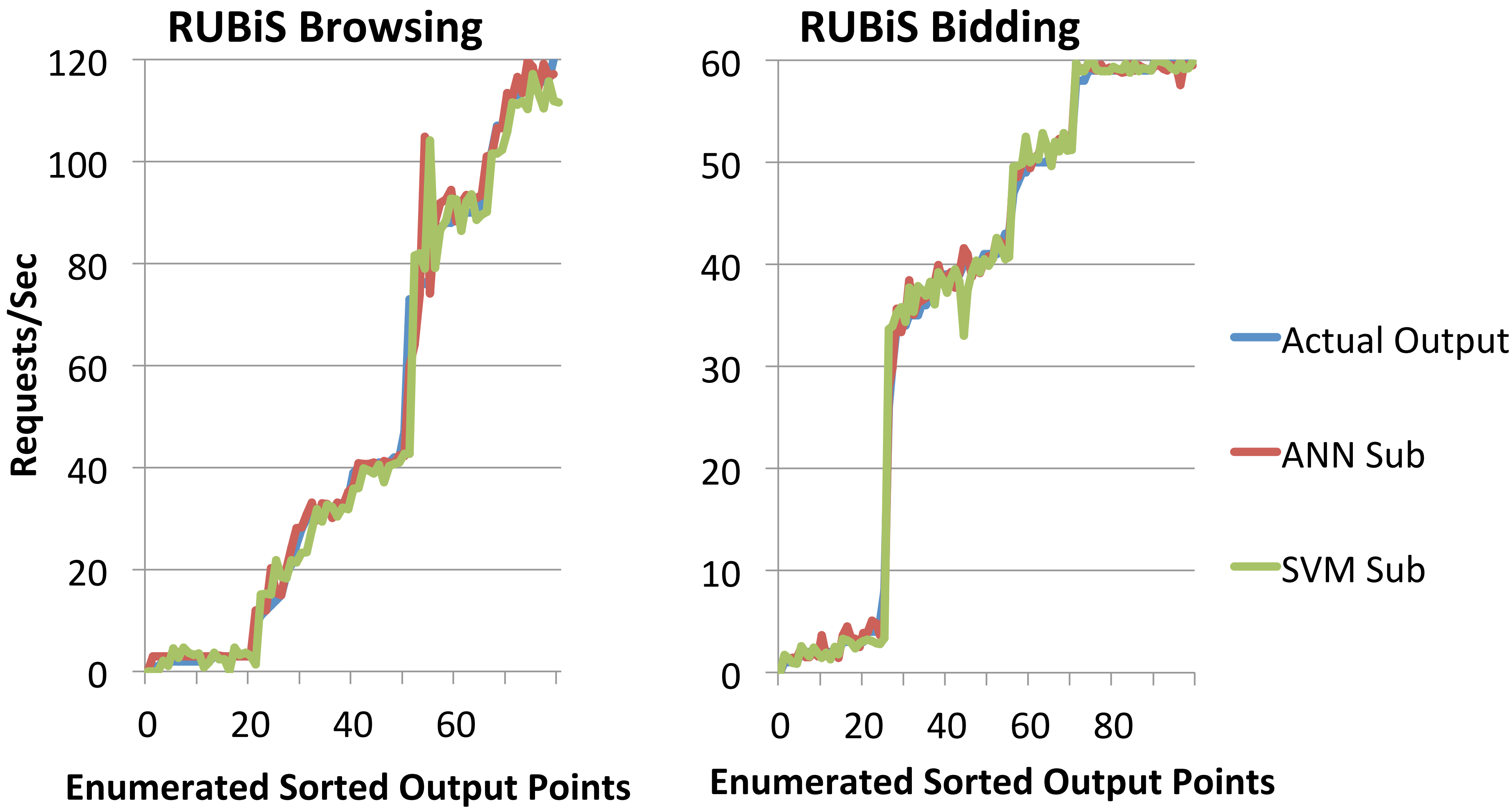
High Prediction Accuracy with Sub-Modeling



High Prediction Accuracy with Sub-Modeling



High Prediction Accuracy with Sub-Modeling



Predicted output values closely follow actual output

Outline of the talk

- Related Work ✔
- Model Parameters Selection ✔
- Application Performance Model Design ✔
- Evaluation ✔
- VM Sizing
- Conclusion

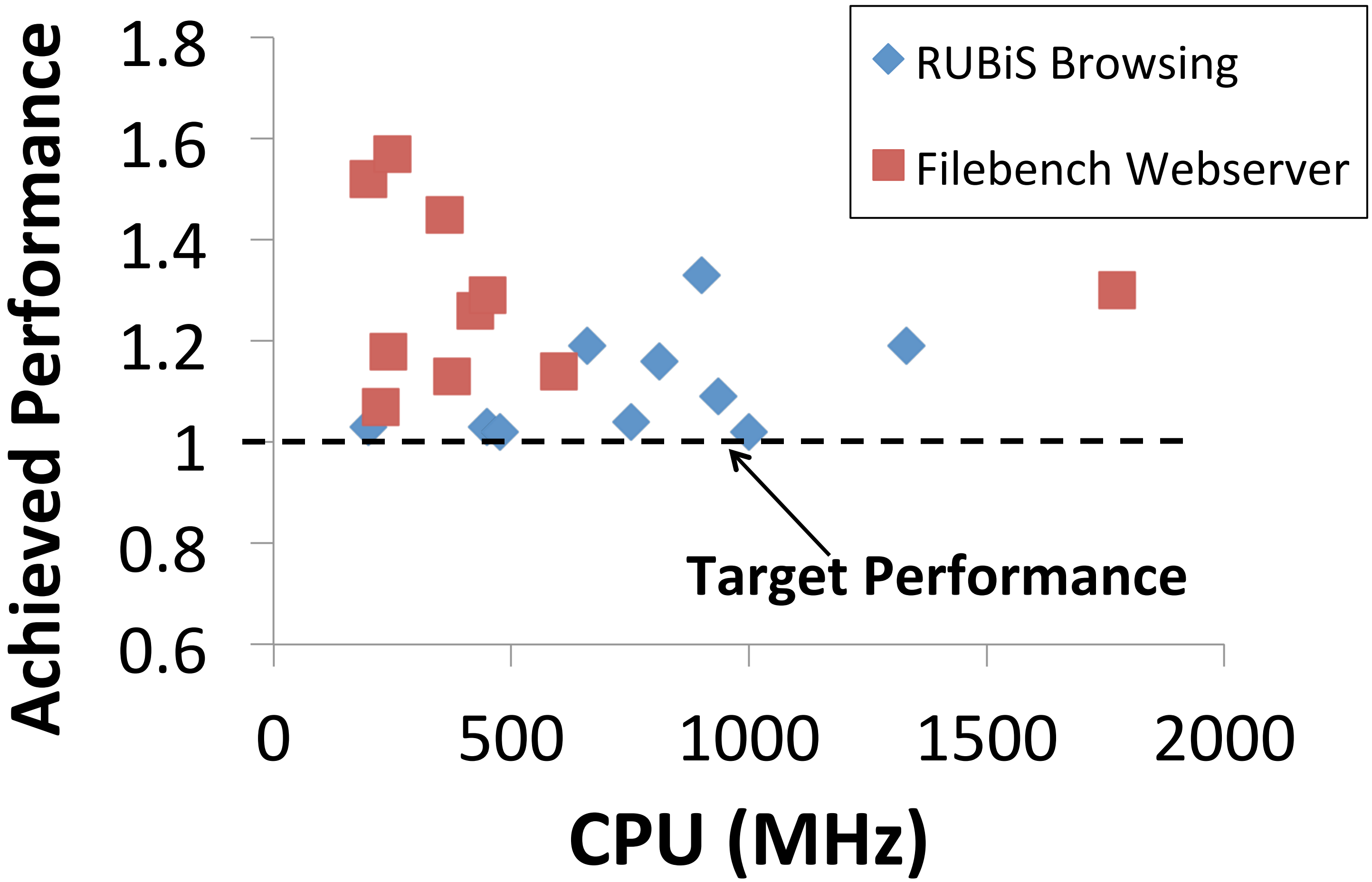
VM Sizing

Objectives:

- Estimating required CPU and Memory to achieve target performance given a VM I/O latency level
- Optimal calculation of resources

Performance models used to achieve both the goals

VM Sizing: Target Performance Achieved



- Target levels achieved in all cases
- 65% sizes optimal and the rest near-optimal

Conclusion

- Identified resource parameters for creating performance models
- ANN and SVM useful for characterizing virtualized applications
- Enhancements needed to improve prediction accuracy
- Performance models effective for VM sizing

QUESTIONS?