E-cubed: End-to-End Energy Management of Virtual Desktop Infrastructure with Guaranteed Performance

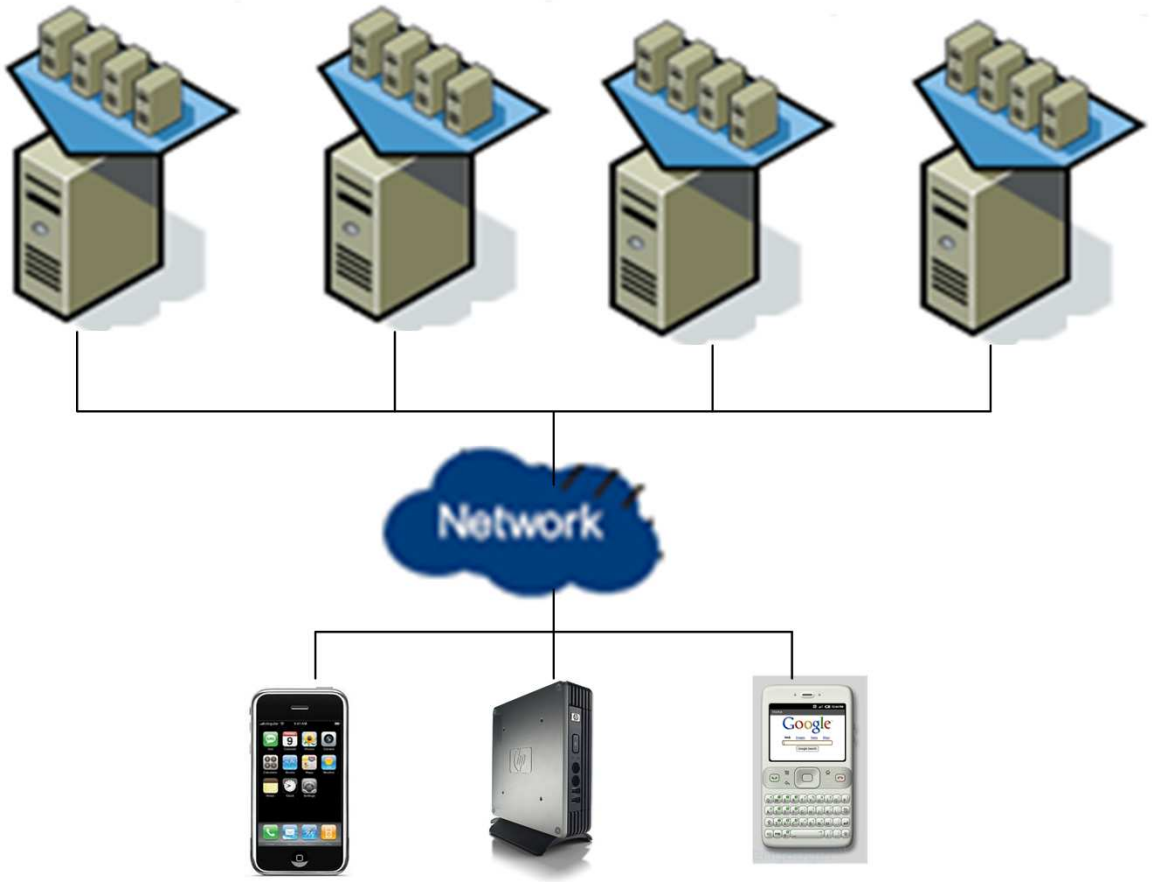
Xing Fu, Tariq Magdon-Ismail, Vikram Makhija, Rishi Bidarkar, Anne Holler, Jing Zhang



© 2009 VMware Inc. All rights reserved

Introduction

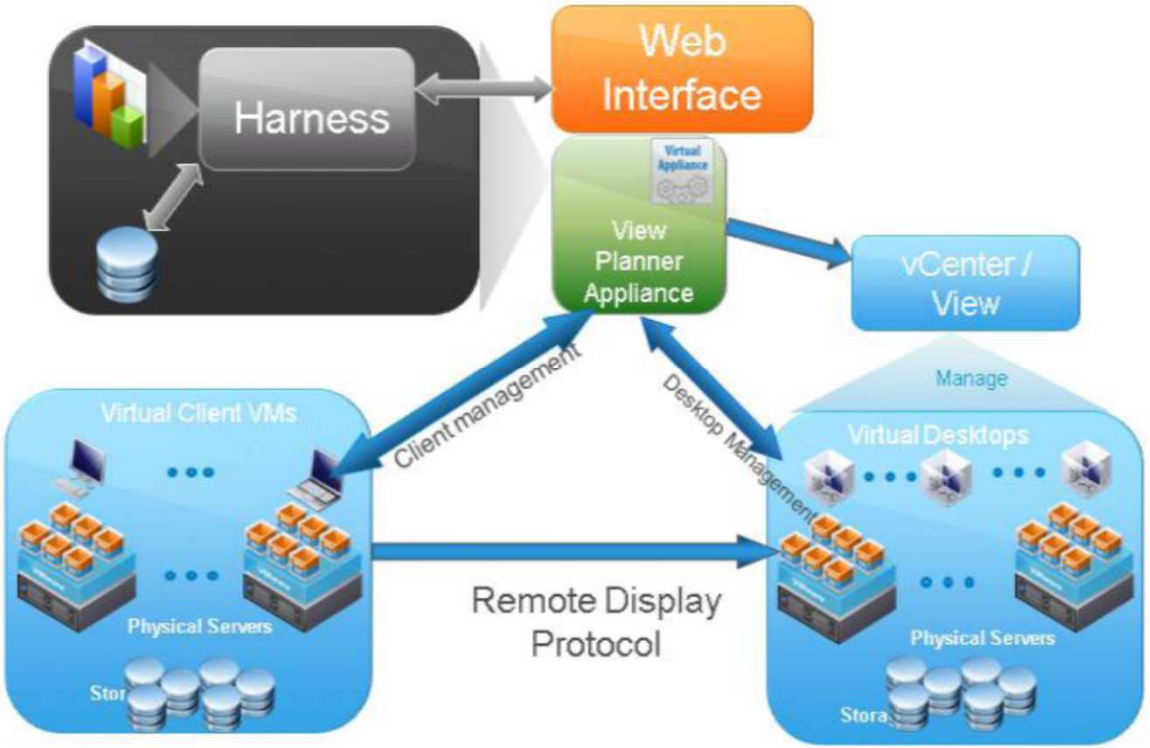
- **Virtual Desktop Infrastructure**



vmware

Introduction

- **View Planner**:
  - Realistic VDI workloads by emulating user operations of various applications



- **End-To-End Performance**
  - 95th percentile response times of all operations.

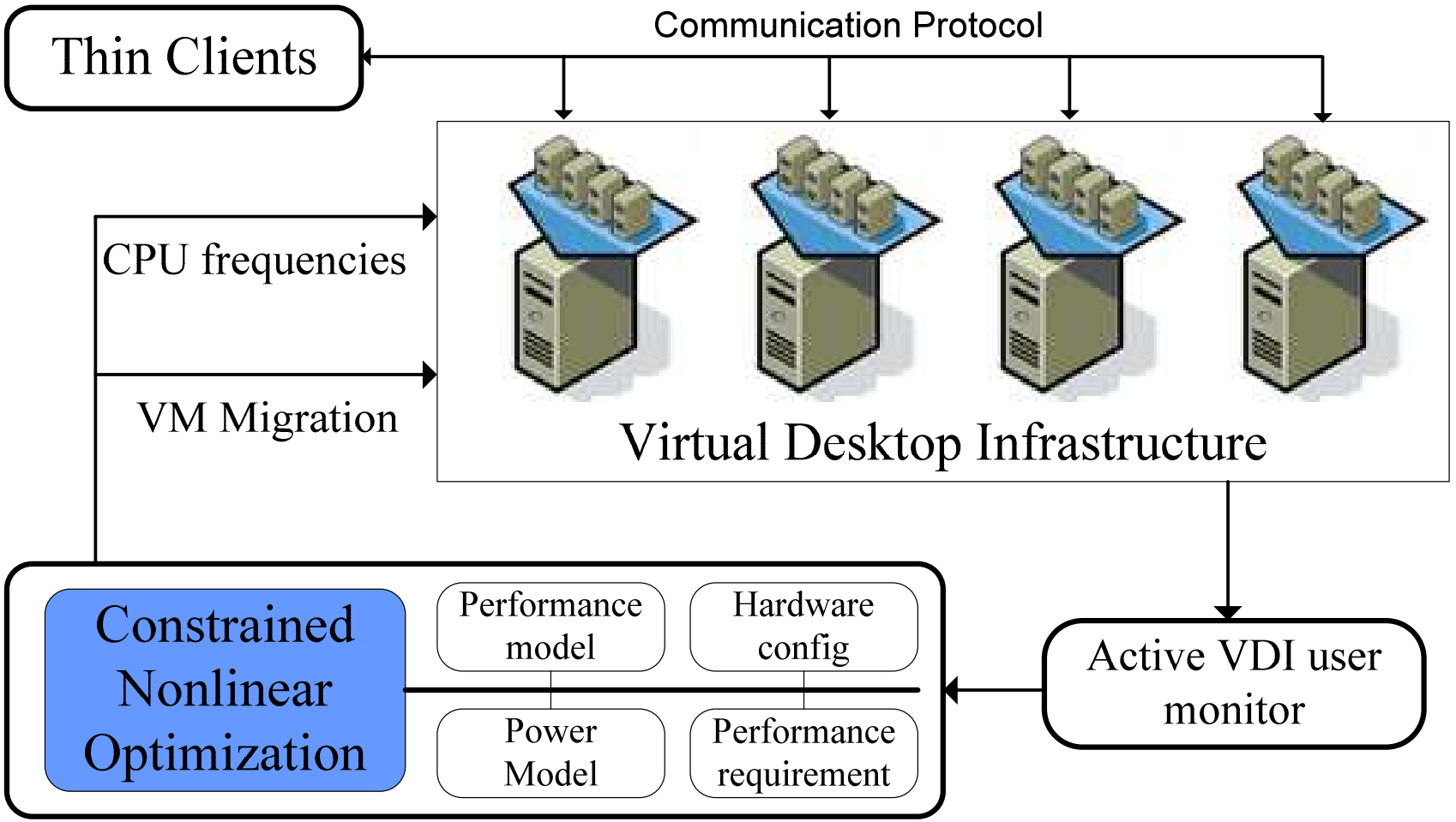
Introduction

- **Related work**

  - **Embedded and mobile devices**
    - Power-aware CPU utilization control for distributed real-time systems, WANG, X., FU, X., LIU, X., AND GU, Z., *RTAS* (2009).
  - **Data center**
    - How much power oversubscription is safe and allowed in data centers? FU, X., WANG, X., AND LEFURGY, C. *ICAC* (2011).
  - **Networking**
    - Traffic aware power optimization for network applications on multicore servers, KUANG, J., BHUYAN, L., AND KLEFSTAD, R., *DAC* (2012).
    - From myth to methodology: cross-layer design for energy efficient wireless communication, EBERLE, W., BOUGARD, B., POLLIN, S., AND CATTHOOR,F., *DAC* (2005).

vmware

System Architecture



Performance Model

- **Empirical modeling by surface fitting**

$$Q_i(f_i, R_i) = K_i\left(\frac{a_1}{f_i} + a_2\right)\left(b_1 R_i^{\,2} + b_2 R_i + b_3\right)$$

Q_i	End-To-End Performance	f_i	Chip-wide CPU frequency
R_i	Number of desktop VMs	K_i	Response time when CPU runs at the maximum frequency

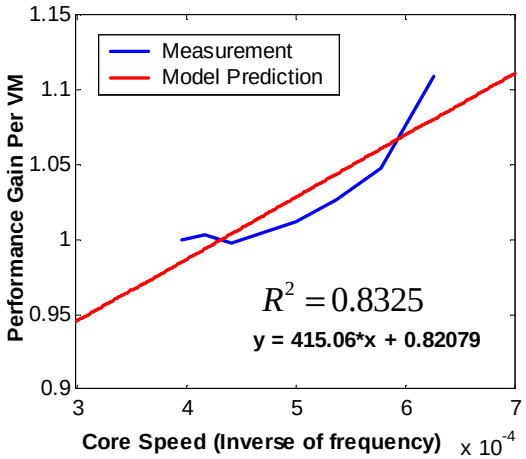
$\left(\frac{a_1}{f_i} + a_2\right)$	Response time inflation due to CPU frequency throttling	$\left(b_1 R_i^{\,2} + b_2 R_i + b_3\right)$	Response time inflation due to VM consolidation

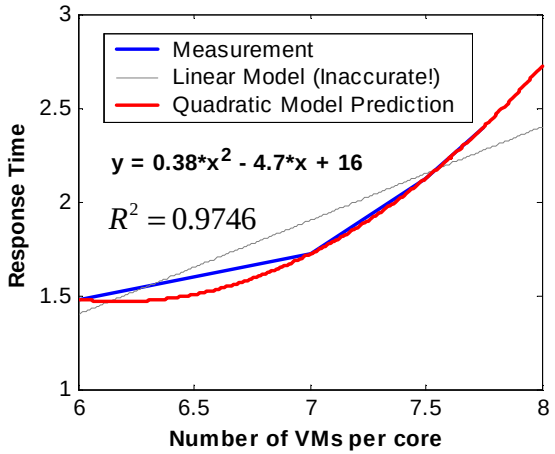
$$R^2 = 0.85$$

vmware

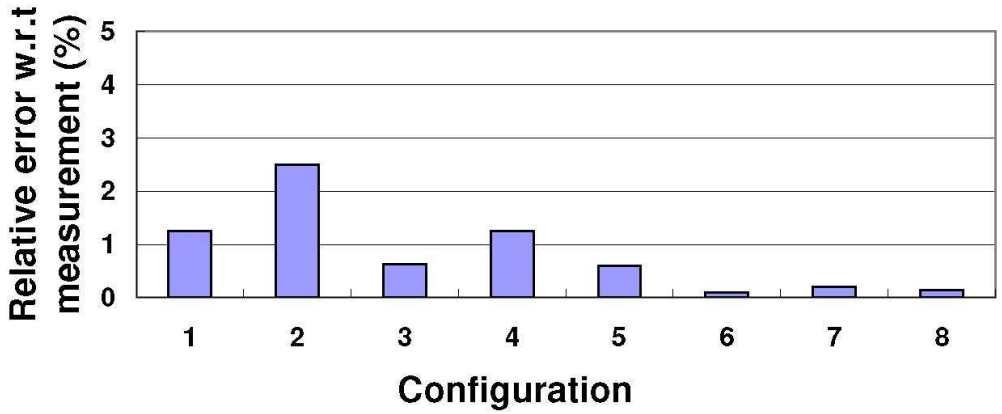
Performance Model

- **Empirical Modeling**





- **Model Validation**



vmware

Optimal Algorithm

- **Cost function**

$$P_{VDI} = \sum_{i=1}^{N_{host}} S_i \left(\alpha_i f_i^{\beta_i} + M_i R_i + L_i \right)$$

S_i	If the ith server hosts dekstop VMs, $S_i = 1$ If the server is idle and powered off, $S_i = 0$
$\alpha_i, \beta_i, M_i, L_i$	server-specific parameters

- **Subject to**

$$Q_i(f_i, R_i) \le QoS$$

$$f_{i,\min} \le f_i \le f_{i,\max}$$

$$\sum_{i=1}^{N_{host}} R_i = N_{VM}$$

vmware

Optimal Algorithm

- **Transform to Mixed-Integer Nonlinear Programming**
  - The relationship between ${S_i}$ and ${R_i}$ can be established as

$$S_i = sgn\left(R_i\right)$$

  - MINLP requires the cost function to be a continuous function
    - signum can be approximated using a continuous function such as tanh

- **Algorithm Complexity**
  - Curse of dimension
  - Require heuristic for large-scale VDI deployments.

vmware

Scalable Algorithm

- **Observation**
  - Idle physical server shutdown can lead to significantly more energy savings than throttling DVFS levels

Algorithm 1 Heuristic for large-scale VDI deployments

begin

1: The 1st step:
2: Calculate the number of desktop VMs per server according to the performance model, and denote the number by R_0.
3: Turn on $\left\lfloor \frac{N_{VM}}{R_0} \right\rfloor$ physical servers which runs at the highest DVFS level;
4: **if** $Q\left(f_{\max}, R_0\right) < QoS$ **then**
5: The 2nd step: Invoke a host-level optimization algorithm
6: **end if**
7: Actuators: shut-down idle physical servers, enforce f_i.

end

vmware

Implementation

Hardware	
CPU	One-socket AMD Opteron 6128 12-core CPU 1.9GHz Two-socket AMD Opteron 254 dual-core CPU 2.8GHz
Memory	16GB and 4GB
Network	1 Gb/s Ethernet local switch
Shared Storage	OpenNAS

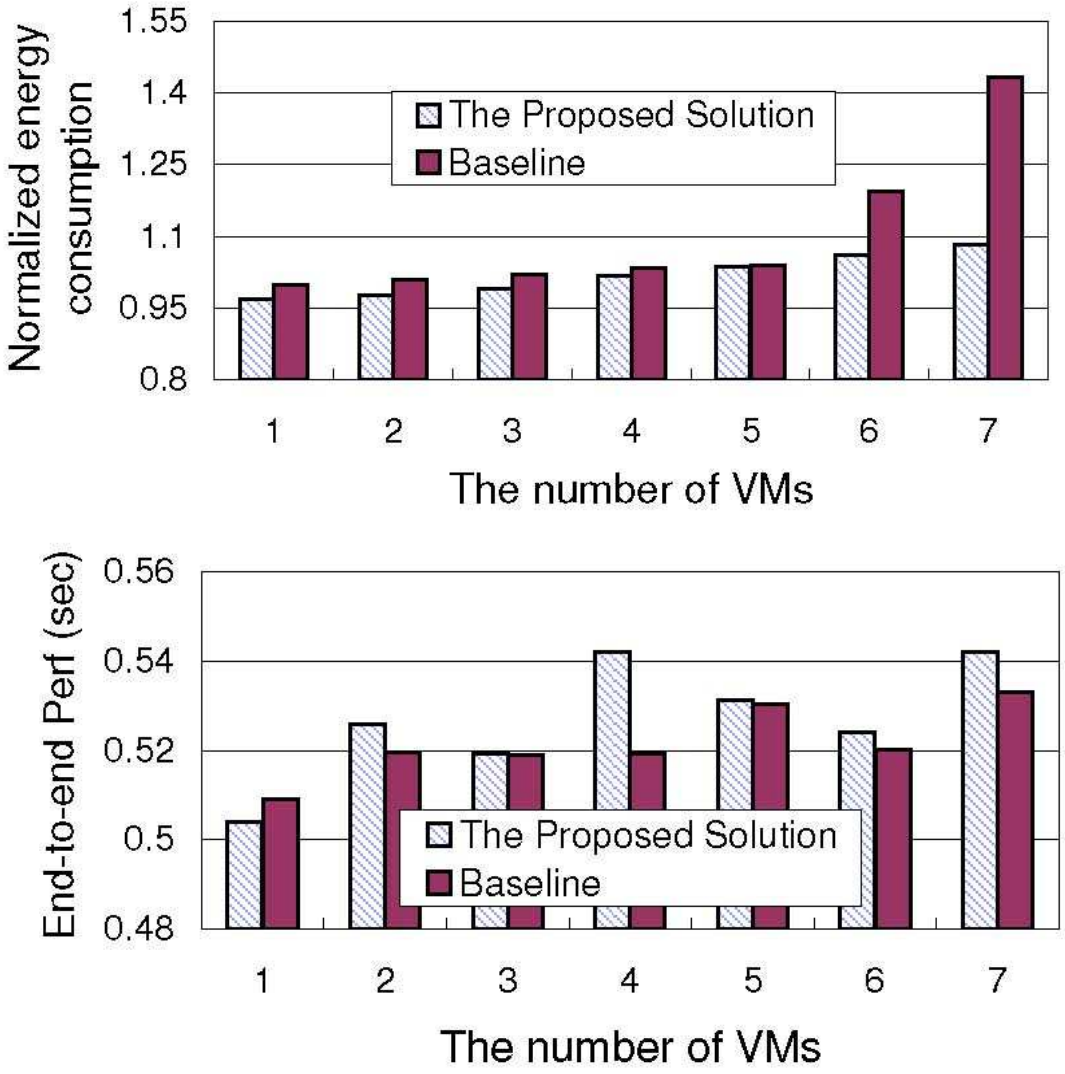
Software	
VMWare vSphere	5.0
VMWare vCenter Server	5.0
View Planner	2.0

Baseline

- **State-of-the-art power management in a virtualized environment**
  - Controls the utilization of CPU and main memory within a fixed range
  - Works well for CPU and main memory intensive workloads
    - No knowledge of end-to-end performance

- **Conservative**
  - Power management decision based solely on CPU utilization.
  - A fixed safe margin exists

Experimental Results



vmware

Conclusion

- Propose **end-to-end** energy management of virtual desktop infrastructure with guaranteed performance
- Establish end-to-end **performance model** of View Planner workload
- Designed mixed-integer nonlinear programming **optimal** algorithm and **scalable** heuristic.
- Hardware experiments show that for high consolidation ratios energy reduction ranges from **11% to 25%**.

vmware

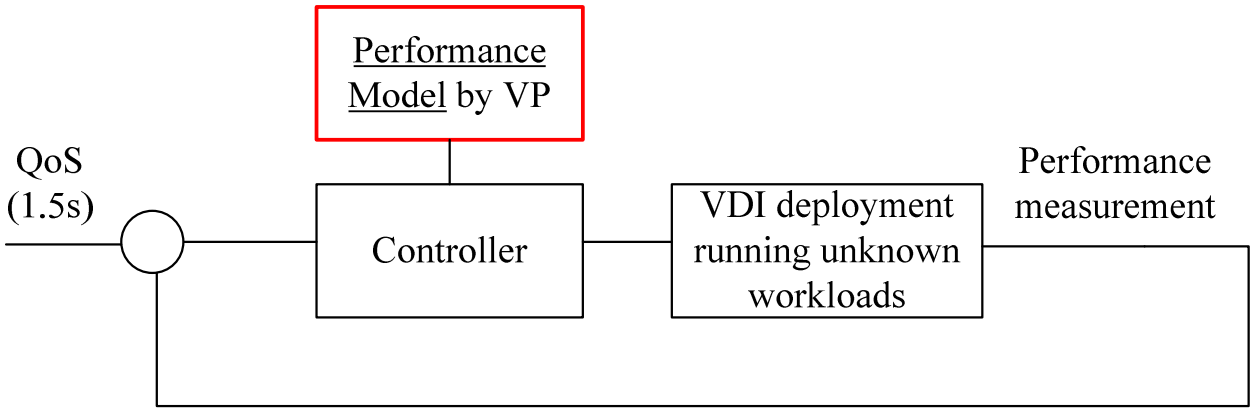
Questions?

Thanks for your feedback

vmware

Future work

- **GPU-intensive VDI workload**
  - Manage power-hungry GPU
- **VDI Performance guarantee under unknown workload**



- **Performance counters from CPU processors**