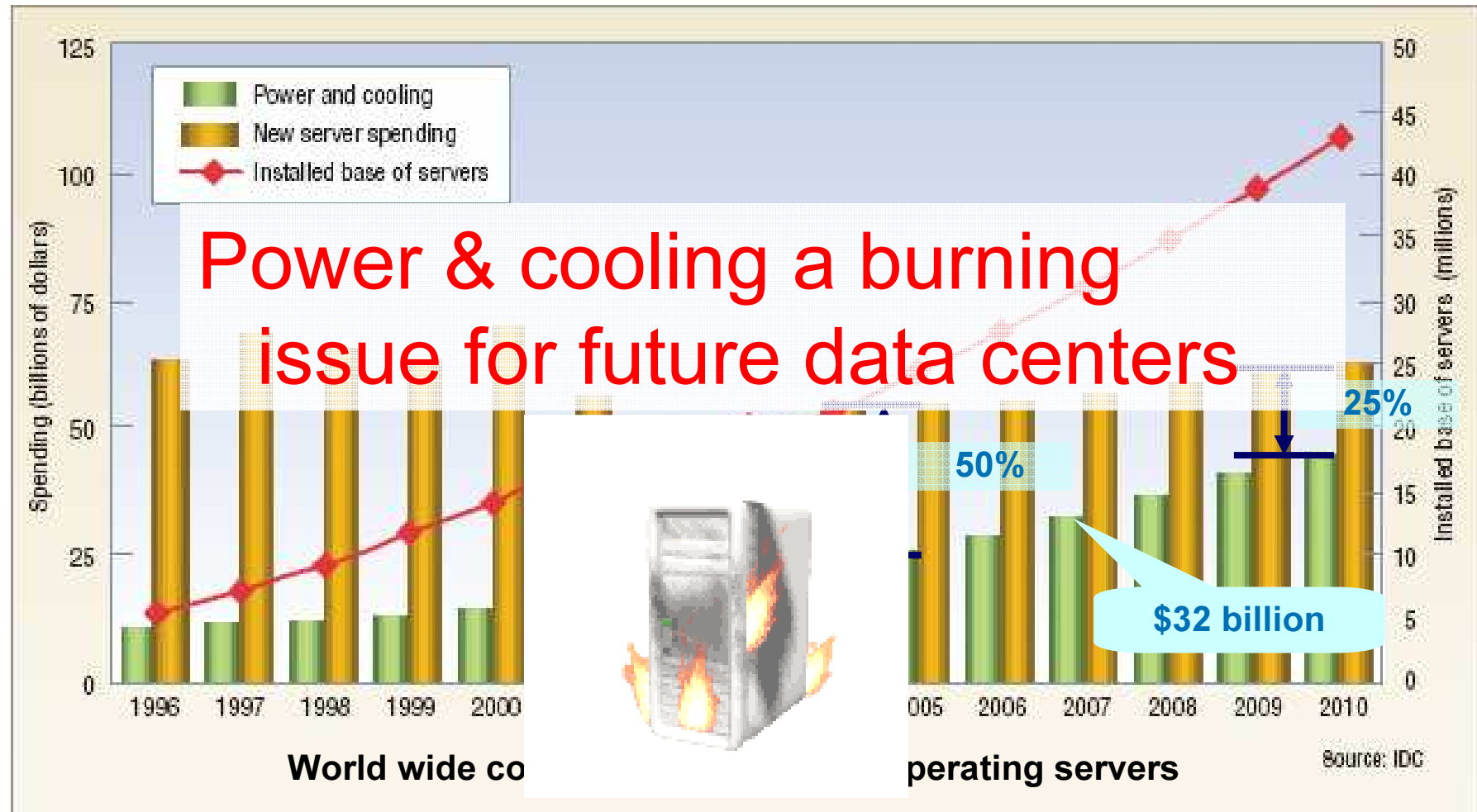


*"No Power Struggles!"*  
Co-ordinated Multi-level Power  
Management for the Data Center

Ramya Raghavendra, Partha Ranganathan, Vanish Talwar, Zhikui Wang, Xiaoyun Zhu



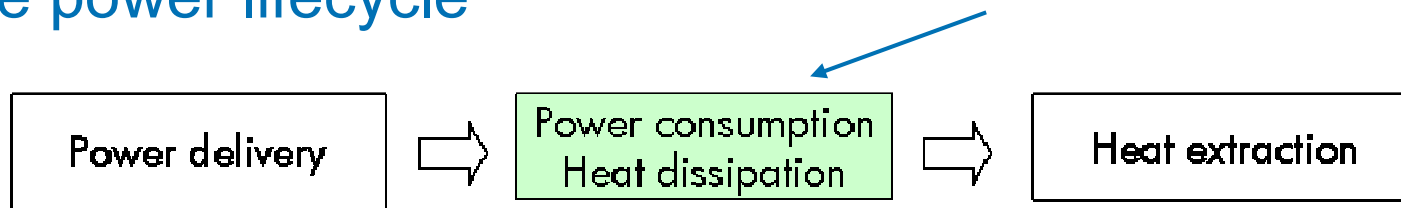
# Motivation



environmental impact, heat& density, reliability

# HP Labs SmartPower Project

- The power lifecycle



- Systems approach to reducing power

- Addressing the source means cumulative benefits

$$\text{Powercosts} = (1 + K_1 + L_1 + K_2 L_1) U_{s,grid} P_{\text{consumed hardware}}$$

- At the intersection of IT and facilities domains

- Key research streams

- Power-aware system and solution architectures
- Models and metrics for power characterization

# Some Recent Work

*(citations refer to bibliography  
at end of slide deck...)*

- Power-aware systems & solutions architecture
  - Blade power capping, multi-level power management, profit-aware scheduling, temperature-aware scheduling [3][4][7][10][13][14]
- Models and metrology for power & heat
  - JouleSort, Zesti, BladeSim, ConSil, Weatherman, Splice [1][2][5][6] [8][9][15]
- Other power management [11][12]
- More details in HotChips tutorial or SmartPower web site...

## Today's talk

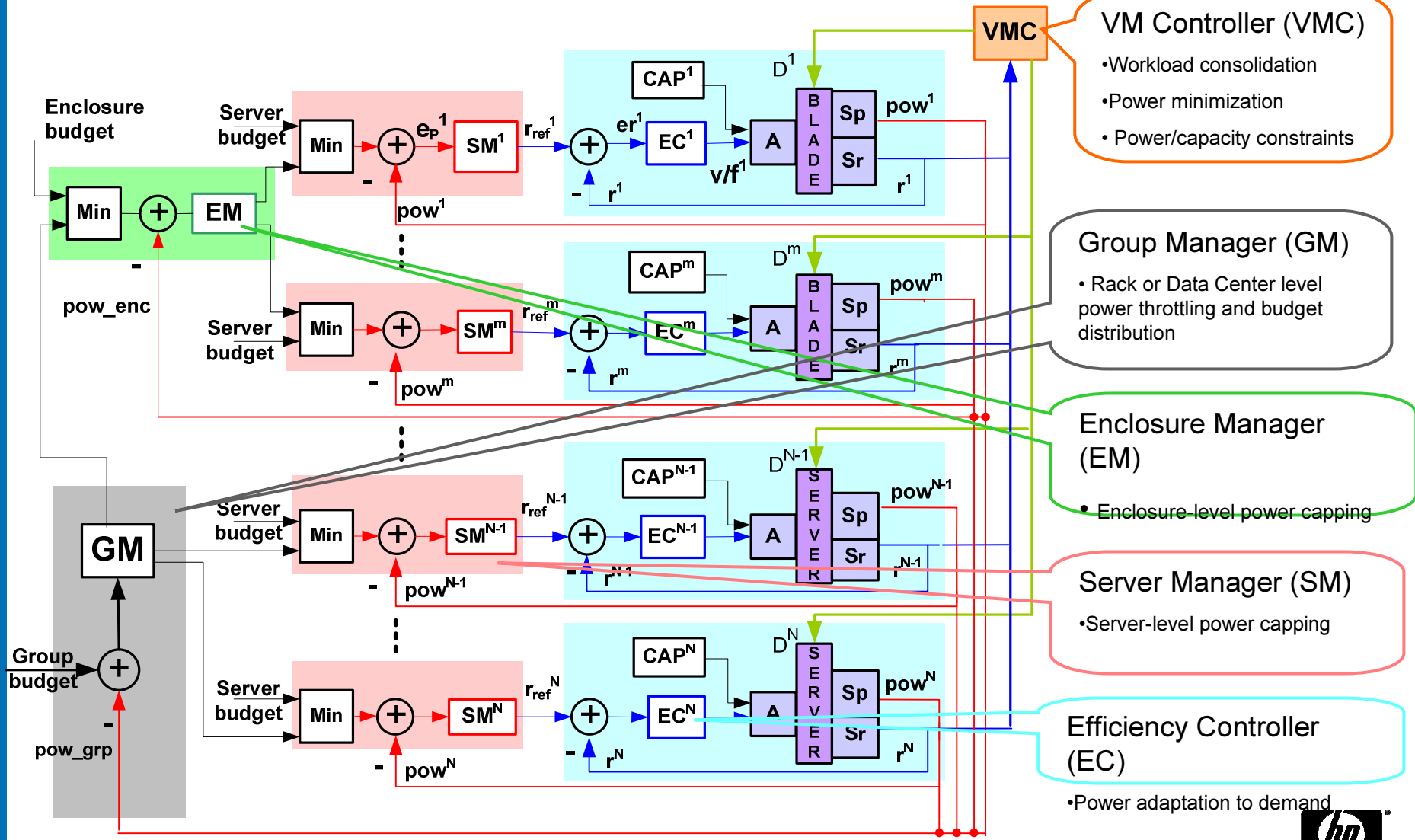
- Multi-level power management



# Contributions

- First unified architecture for data center power management
  - Minimal interfaces & information exchange between loops
  - Feedback control theory for mathematical rigor
  - Evaluation on real-world traces: correctness, stability, efficiency
- Insights on design trade-offs
  - Architectural alternatives for various objective functions
  - Implementation alternatives (time constants and hw/sw)
  - Mechanisms (p-states, VMs) & policies (pre-emptive, fair-share, ...)

# Unified and Extensible Architecture



**VM Controller (VMC)**

- Workload consolidation
- Power minimization
- Power/capacity constraints

**Group Manager (GM)**

- Rack or Data Center level power throttling and budget distribution

**Enclosure Manager (EM)**

- Enclosure-level power capping

**Server Manager (SM)**

- Server-level power capping

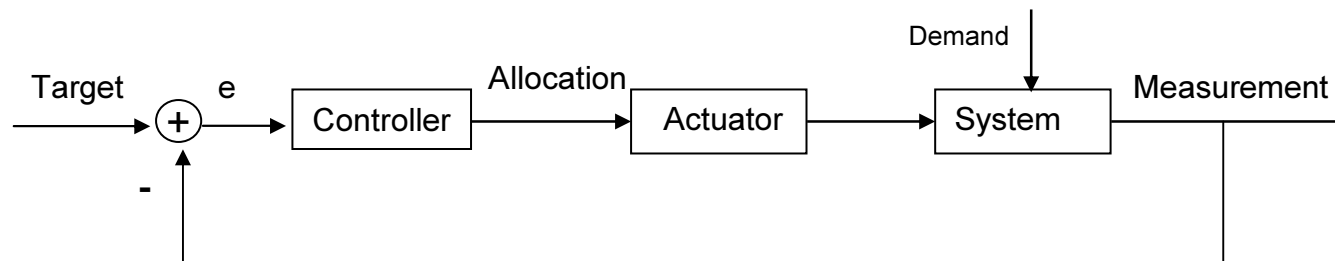
**Efficiency Controller (EC)**

- Power adaptation to demand



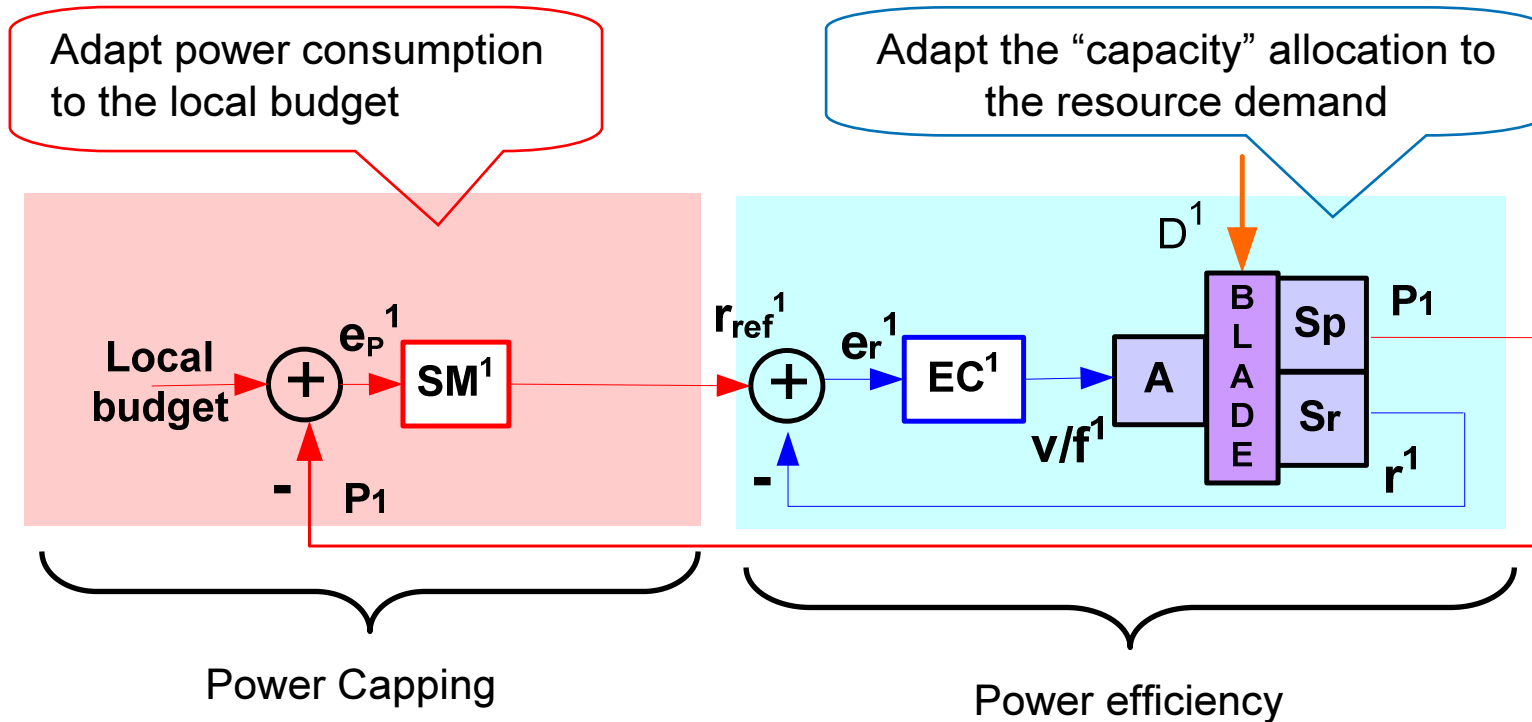
# Background

- Feedback control theory
  - Formal theoretical guarantees of stability and performance
  - Can account for inaccuracies in model
  - Adapts to the workload on the fly



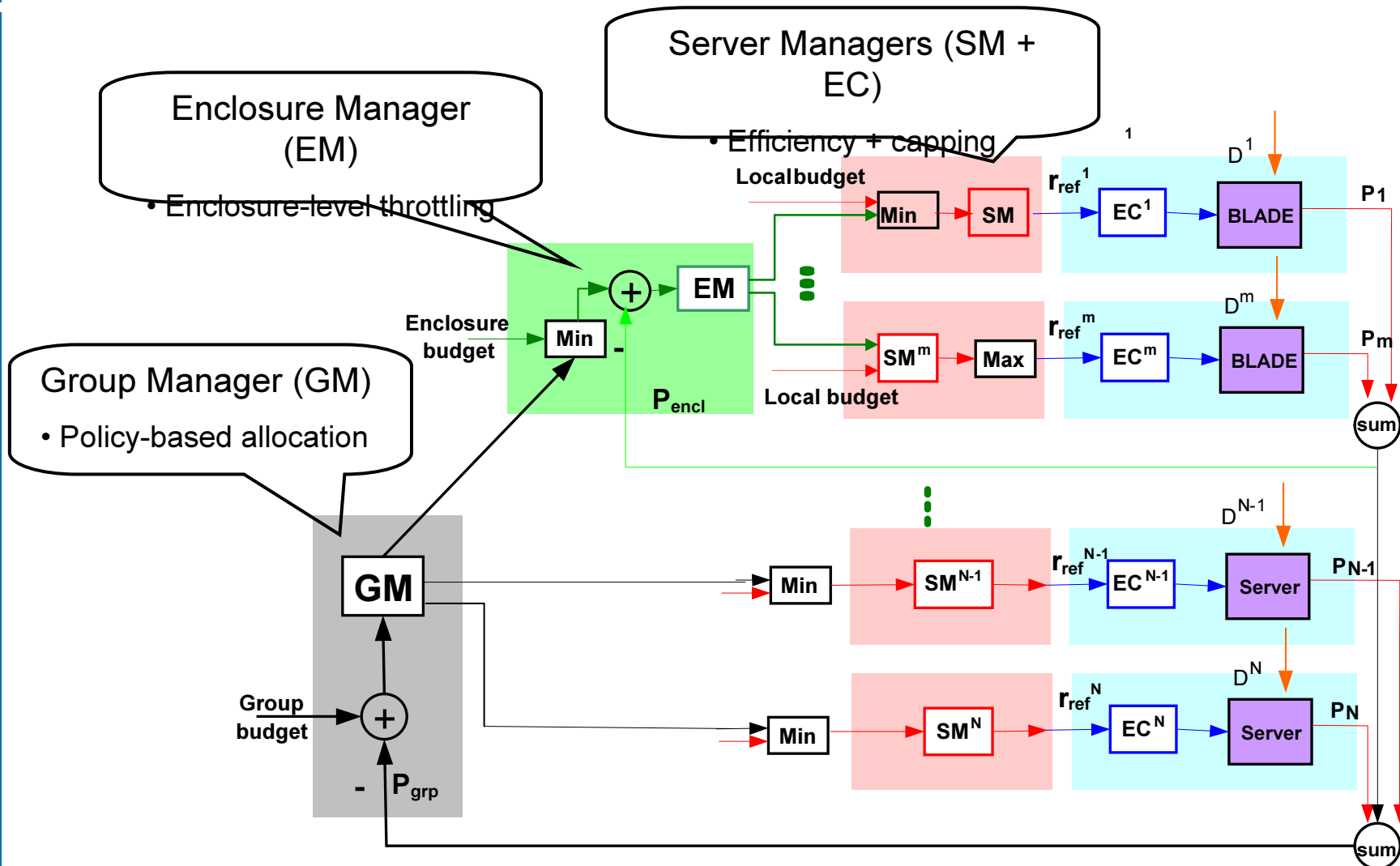
*Standard feedback control loop*

# Efficiency + capping at single server

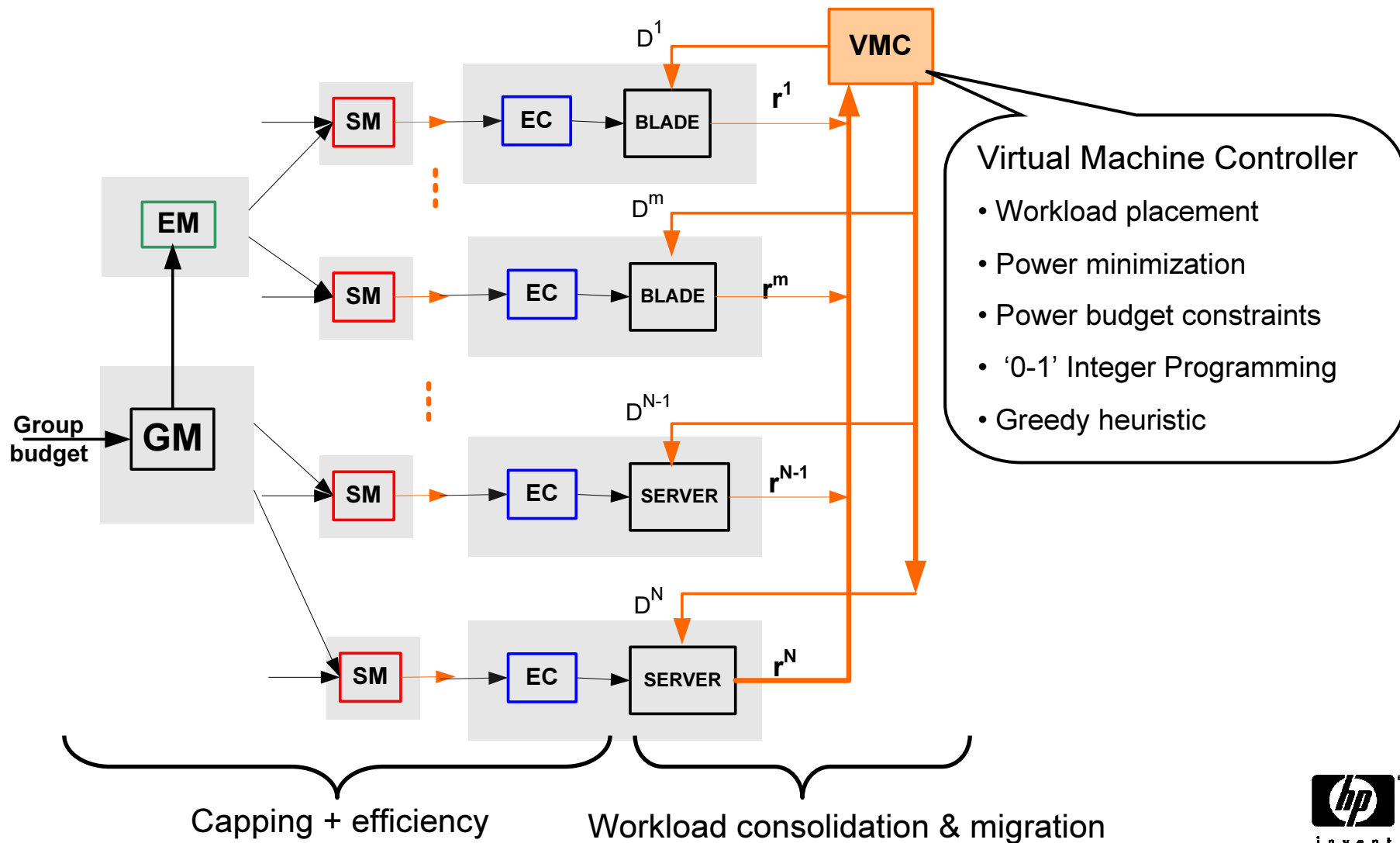


- Efficiency: tracking problem
  - Notion of “container” from ACTS work on adaptive resource control
  - Allows power consumption to track workload demand
- Capping : throttling problem
  - Change  $r_{ref}$  to adapt to power cap

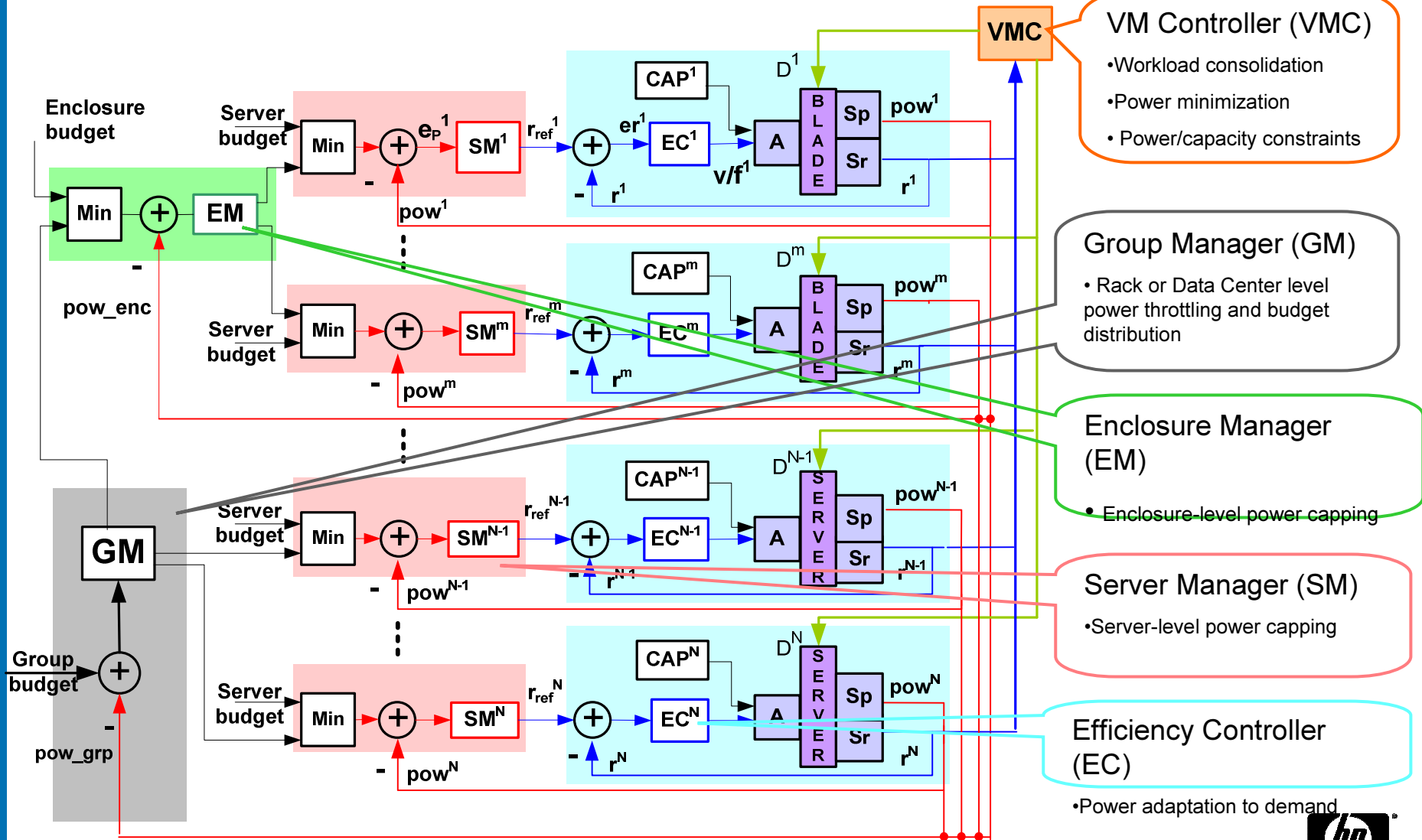
# Multi-level Power Capping



# Virtual Machine Workload Distributor



# Unified and Extensible Architecture



# Evaluation - Methodology

- Challenges with full-scale deployment study
  - Anyone have a live data center I can borrow?
  - Evaluating next-gen hardware with current systems?
  - Prototyping scale and work-benefit tradeoffs
- Our approach: multi-pronged
  - Simulation-based evaluation of design space (BladeSim)
    - Real world traces from 9 enterprises (180 servers)
    - Power/perf models from prototype calibration
  - Smaller-scale prototype evaluation of specific solutions
  - Model-based analysis for stability

# Evaluation – Design Choices

- Combinatorial explosion in design space
  - Controllers, hardware, frequencies, overheads, policies, levels, ...
- Our approach: representative subset
  - simplifying but not simplistic assumptions

$$(Models) : pow = g_p(r) = c_p r + d_p, \quad p = 0, 1, 2, \dots$$

$$perf = h_p(r) = a_p r, \quad p = 0, 1, 2, \dots$$

$$(EC) : f(k) = f(k-1) - \lambda f_Q(k-1)r(k-1) / r_{ref}(r_{ref} - r(k-1)).$$

$$(SM) : r_{ref}(\hat{k}) = r_{ref}(\hat{k}-1) - \beta_{loc}(cap\_loc - pow(\hat{k}-1)).$$

$$(EM) : cap\_loc = \min(CAP\_LOC, \frac{pow\_loc}{pow\_enc} \times cap\_enc).$$

$$(GMS) : cap\_enc = \min(CAP\_ENC, \frac{pow\_enc}{pow\_grp} \times CAP\_GRP).$$

$$cap\_loc = \min(CAP\_LOC, \frac{pow\_loc}{pow\_grp} \times CAP\_GRP).$$

(VMCs)

$$(1) \min \sum_{i=1}^m pow_i + \sum_{i=1}^m \sum_{j=1}^n \alpha_M |X_{ij} - X_{ij}^0|$$

$$(2) s.t. \sum_{j=1}^n X_{ij} r_j (1 + \alpha_V) \leq \bar{r}$$

$$(3) pow_i \leq (1 - b_{loc}) CAP\_LOC_i, \quad i = 1, 2, \dots, m$$

$$(4) \sum_{i=1}^m M_{iq} pow_i \leq (1 - b_{enc}) CAP\_ENC_q, \quad q = 1, 2, \dots, l$$

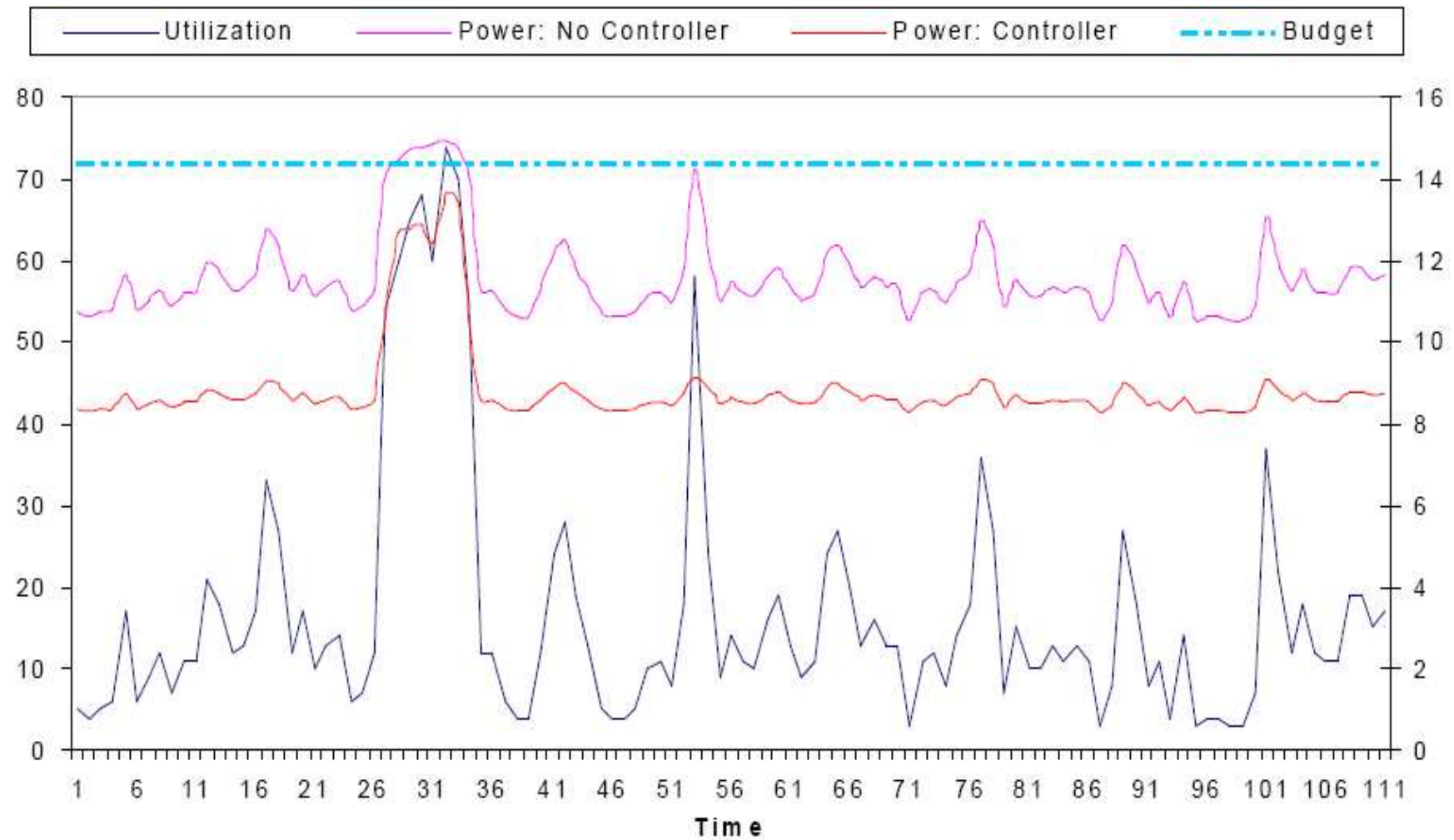
$$(5) \sum_{i=1}^m pow_i \leq (1 - b_{grp}) CAP\_GRP$$

$$(6) \sum_{i=1}^m X_{ij} = 1, \quad j = 1, 2, \dots, n$$

$$(7) X_{ij} \in \{0, 1\}, \quad i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n$$

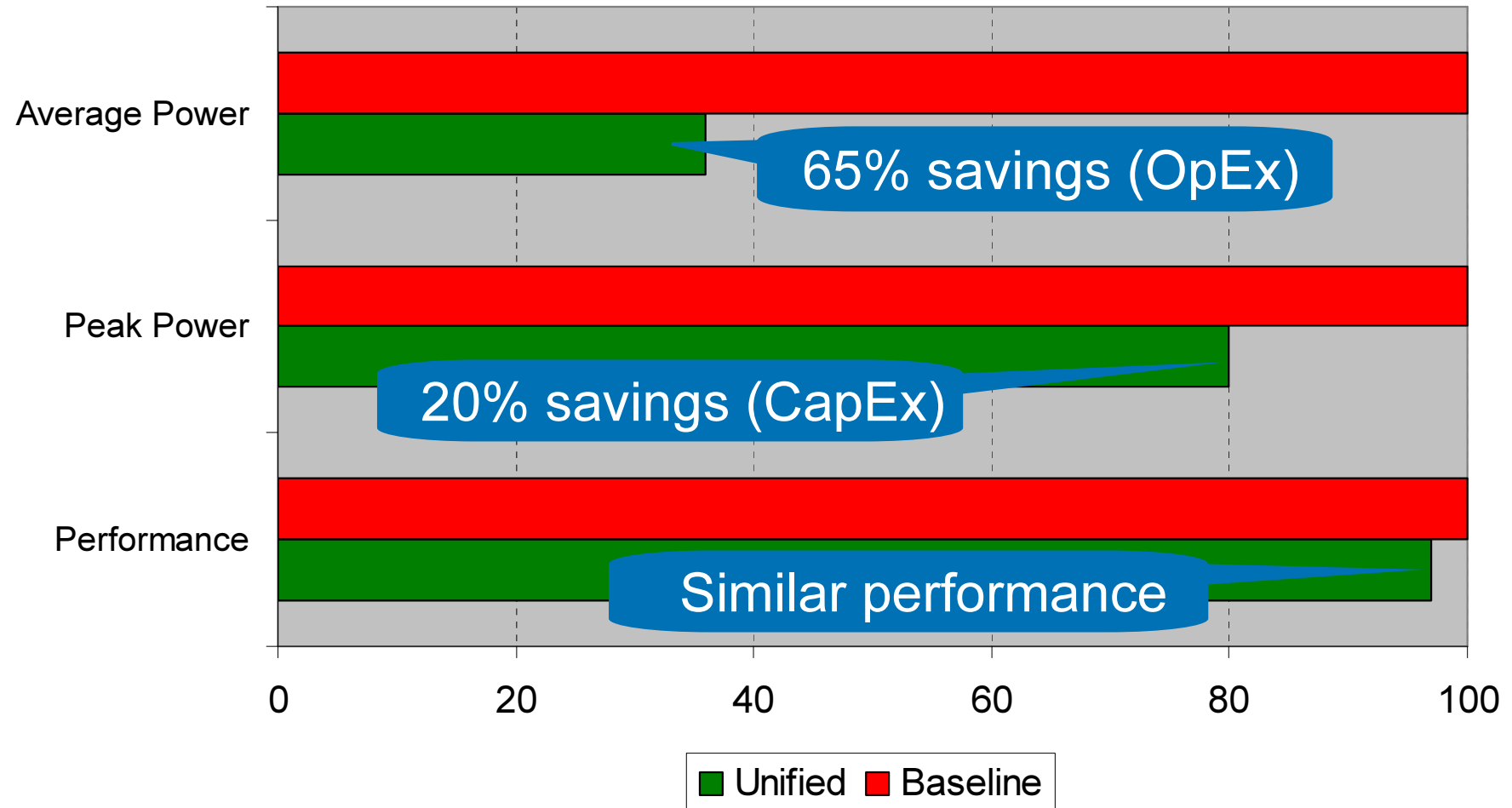
	Metrics and knobs	Notation	Base value
Server	static power budget	CAP_LOC	10% off server max
	dynamic power budget	cap_loc	tuned by EM or GM
	power consumption	pow	measured for SM/EM/GM
	target utilization	r_ref	tuned by SM
	measured utilization	r	measured for EC
	P-states	p0, p1, ...	p0, ..., p4, tuned by EC
	desired clock frequency	f	Hz
	quantized frequency	f_Q	[1G, 833M, 700M, 600M, 533M] Hz
	performance	perf	work done
	static power budget	CAP_ENC	15% off enclosure max
Enclosure	dynamic power budget	cap_enc	tuned by GM
	power consumption	pow_enc	measured for EM and GM
Group	power budget	CAP_GRP	20% off group max
	power consumption	pow_grp	measured for GM
Virtual Machine	virtualization overhead	$\alpha_V$	10% of VM utilization
	migration overhead	$\alpha_M$	10% of VM utilization
	constraints buffers	b_loc, b_enc, b_grp	tuned based on budget violations
Workload	number of workloads	n	180 enterprise traces
	demand for capacity	D	in utilization
	placement on servers	X	matrix with 0/1 elements
System Property	number of servers	m	180
	number of enclosures	l	20
	relationship between servers & enclosures	M	matrix with 0/1 elements
Control Interval	efficiency control (EC)	T_ec	1
	server manager (SM)	T_sm	5
	enclosure manager (EM)	T_em	25
	group manager (GM)	T_grp	50
	VM Controller (VMC)	T_vmc	500
Controller Gain	efficiency control (EC)	$\lambda$	0.8
	server manager (SM)	$\beta_{loc}$	1

# Results: It works!

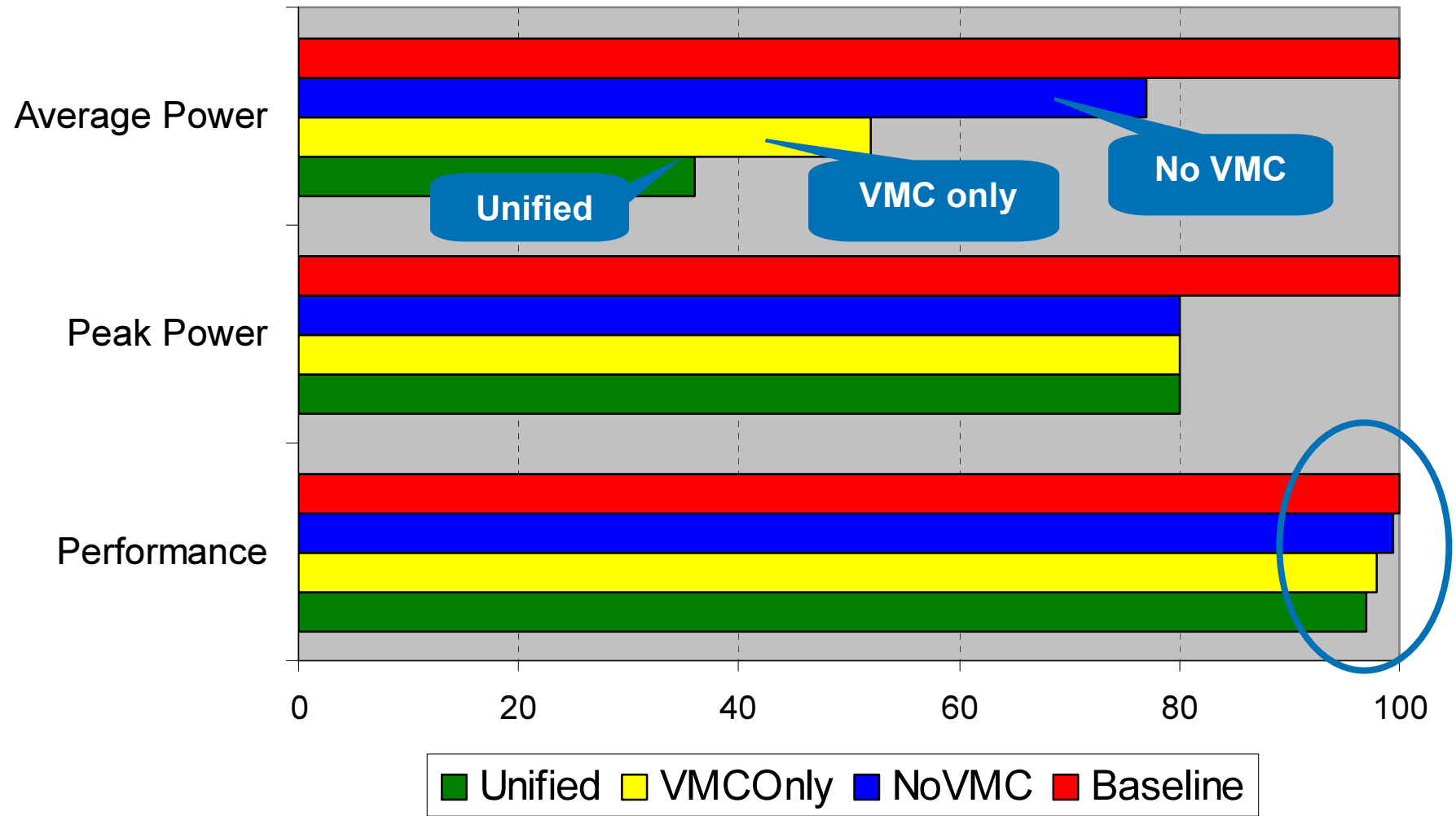


- Workload-aware adaptation & correctness

# Results: It works well!



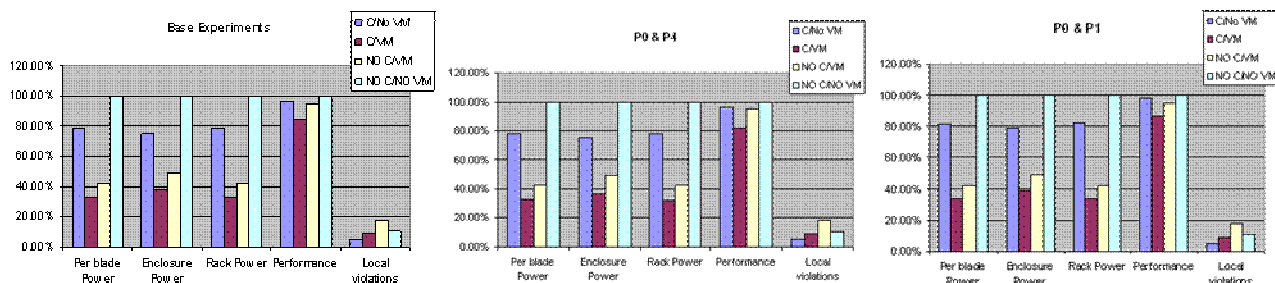
# Design Insights: An example



# Other insights on design tradeoffs

- Architectural choices
  - VM vs EC, coordination alternatives
- System design choices
  - Power budgets, P-states
- Implementation aspects
  - Knobs, overheads, time-constants, policies

Coordination enables flexibility and simplicity



See paper for more details

# Ongoing and Future Work

- Detailed validation and prototyping
  - Deploy on HP Labs Smart Data Center
- Extension to other controllers
  - Performance, cooling, ... SLA...

# Summary

- Power an important challenge for future data centers
  - Data center power “hot” area
    - \$30B market in 2007, compaction, environmental benefits
  - Uncoordinated solutions inefficient & unstable
    - Varying time constants, overlapping objective functions & actuators, ...
- First unified architecture for data center power management
  - Unified architecture based on formalism
    - Adaptive controller core + well-defined interfaces and policies
  - correctness + stability + performance
    - 65% electricity reduction, 20% provisioning reduction, similar perf
  - Insights on design tradeoffs
    - Architectural and implementation tradeoffs; mechanisms and policies

# Questions?



"TWO WORDS? ONE WORD!  
STARTS WITH... SOUNDS LIKE..."

# Bibliography

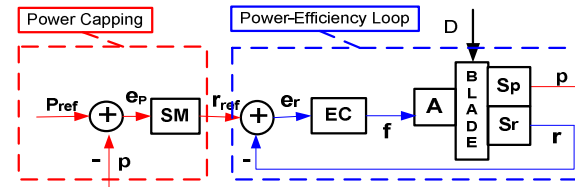
1. "JouleSort: A Balanced Energy-Efficiency Benchmark," Suzanne Rivoire, Mehul Shah, Christos Kozyrakis, Parthasarathy Ranganathan, November 2006 [pdf]
2. "Zesti: Full-System Power Modeling and Estimation," Dimitris Economou, Suzanne Rivoire, Christos Kozyrakis, and Parthasarathy Ranganathan, June 2006 [pdf]
3. "Cost-aware Scheduling for Heterogeneous Enterprise Machines (CASH'EM)," Jennifer Burge, Parthasarathy Ranganathan, Janet L. Wiener, September 2006 [pdf]
4. "No Power Struggles: A Unified Power Management Architecture for the Data Center," Ramya Raghavendra, Parthasarathy Ranganathan, Vanish Talwar, Xiaoyun Zhu, and Zhikui Wang, *ASPLOS*, March 2008 [hp-pdf]
5. "Simulating Complex Enterprise Workloads using Utilization Traces," Parthasarathy Ranganathan and Philip Leech, *Tenth Workshop on Computer Architecture Evaluation using Commercial Workloads (CAECW)*, held with HPCA-13, February 2007 [pdf]
6. "Full-system Power Analysis and Modeling for Server Environments," Dimitris Economou, Suzanne Rivoire, Christos Kozyrakis, and Parthasarathy Ranganathan, *Workshop on Modeling, Benchmarking, and Simulation (MoBS)*, June 2006 [pdf]
7. "Ensemble-level Power Management for Dense Blade Servers," Parthasarathy Ranganathan, Phil Leech, David Irwin, and Jeff Chase, *Proceedings of the International Symposium on Computer Architecture (ISCA)*, June 2006 [pdf] An earlier version appeared in Techcon 2005.
8. "ConSil: Low-cost Thermal Mapping of Data Centers," Justin Moore, Jeff Chase, and Parthasarathy Ranganathan. *First Workshop on Tackling Computer Systems Problems with Machine Learning Techniques (SysML)*, June, 2006 [pdf]
9. "Weatherman: Automated, Online, and Predictive Thermal Mapping and Management for Data Centers," Justin Moore, Jeff Chase, and Parthasarathy Ranganathan, *Proceedings of the Third International Conference on Autonomic Computing (ICAC)*, June 2006 [pdf]
10. "PowerBalancing for Future-generation Blades," Hernan Laffitte, Phil Leech, Parthasarathy Ranganathan, Charlie Shaver, Khaldoun Alzien, *Proceedings of HP Techcon*, April 2006 [hp-pdf]
11. "Energy-aware user interfaces and energy-adaptive displays," Parthasarathy Ranganathan, Erik Geelhoed, Meera Manahan, and Ken Nicholas, *IEEE Computer*, March 2006 (cover feature) [pdf]
12. "Heterogeneous chip multiprocessors," Rakesh Kumar, Dean Tullsen, Norman Jouppi, Parthasarathy Ranganathan, *IEEE Computer*, November 2005 (cover feature) [pdf]
13. "Dense and smart: Hardware-software co-ordination for blade server power reduction," Parthasarathy Ranganathan et al, *Proceedings of HP TechCon*, March 2005. [hp-pdf]
14. "Making Scheduling Cool: Temperature-aware Resource Scheduling," Justin Moore, Jeff Chase, Parthasarathy Ranganathan, Ratnesh Sharma. *Proceedings of the 2005 Annual Usenix Conference*, April 2005. [pdf] A shorter version appears as a poster in HP TechCon, March 2005 [hp-pdf]
15. "Data Center Workload Monitoring, Analysis, and Emulation," Justin Moore, Jeff Chase, Keith Farkas, and Parthasarathy Ranganathan. In the *Eighth Workshop on Computer Architecture Evaluation using Commercial Workloads (CAECW)*, February, 2005. (Invited paper) [pdf]



# Bonus Slides

# Guaranteeing Stability (Appendix A)

- Control theory provides formal analysis and synthesis for desirable properties of the system under control, e.g., stability and zero tracking error.
- Stability guarantees predictable behavior when the system experiences changes, e.g., workload changes, different budget configuration.
- A quantitative proof for both stability and zero tracking error in one example scenario is sketched.
  - The EC controller can make the CPU utilization track a specified utilization target by dynamically tuning the clock frequency, in spite of changes in the workload demand;
  - The SM controller can make the server power consumption track a given local power cap, possibly set by the upper layer controllers such as the EM or the GM, by dynamically tuning the utilization target fed into the EC controller.
- The hierarchical architecture design and use of different time scales in different control loops make it possible to provide at least qualitative arguments for stability.



$$(Models) \quad pow = g_p(r) = c_p r + d_p, \quad p = 0, 1, 2, \dots$$

$$perf = h_p(r) = a_p r, \quad p = 0, 1, 2, \dots$$

$$(EC) \quad f(k) = f(k-1) - \lambda \frac{f_Q(k-1)r(k-1)}{r_{ref}} (r_{ref} - r(k-1)).$$

$$(SM) \quad r_{ref}(\hat{k}) = r_{ref}(\hat{k}-1) - \beta_{loc} (cap_{loc} - pow(\hat{k}-1)).$$

- I: For a given utilization target  $r_{ref}(\hat{k})$  the CPU utilization of the server,  $r(k)$ , converges globally and asymptotically, using the efficiency controller (EC), under the condition

$$0 < \lambda < 1/r_{ref}.$$

- II: For a given local power cap  $cap_{loc}$  the server power consumption  $pow(k)$  converges globally, using the server manager (EC), under the condition

$$0 < \lambda < 1/r_{ref} \quad \text{and} \quad 0 < \beta < 2/c_{max}.$$

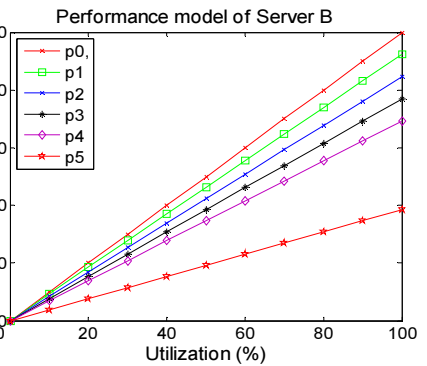
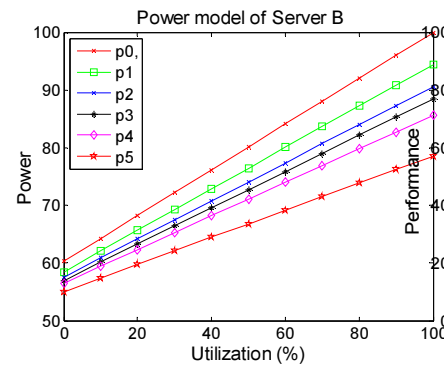
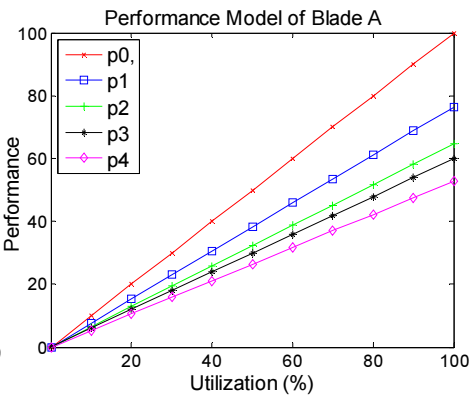
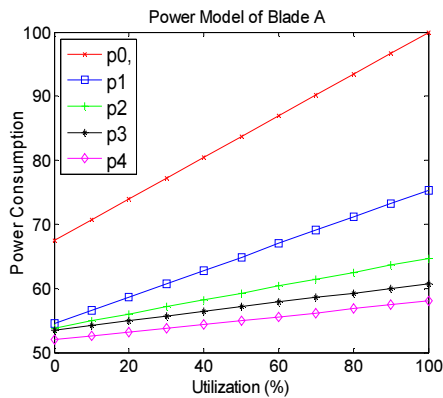
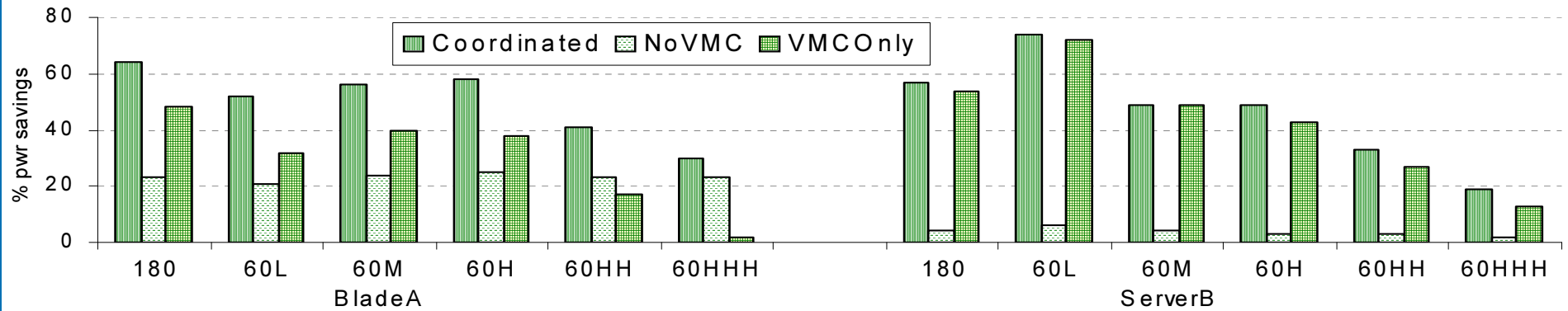
# Coordination

- Federation vs centralization
  - Multi-vendor, isolation, abstraction, local information
- Solving one aspect doesn't work
  - Figure 9 in paper
- Example application
  - VMC + group capper vicious cycle
- Extensions
  - Component/platform coord
  - Electrical power capper
  - Multiple actuators at level
  - VM-platform coord
  - Heterogeneity
  - Energy-delay
  - Implementations in hardware/sw
- Interfaces API
  - DMTF, CIM

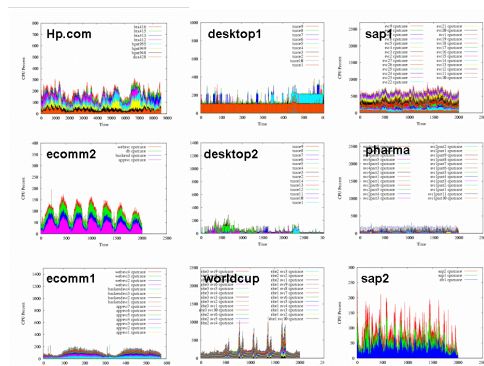
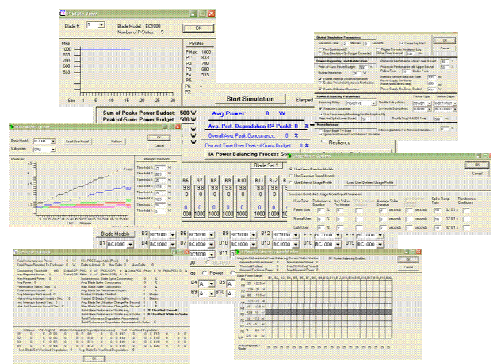
System under control	Power violations			perf loss	pwr save
	GM	EM	SM		
Blade A					
Coordinated	0	0	-5	-3	64
Uncoordinated	0	0	-8	-12	72
Coordinated, appr util	0	0	-3	-2	56
Coordinated, no feedback	0	0	-7	-4	69
Coordinated, no budget limits	0	-5	-23	-8	76
Uncoordinated, min Pstates	0	0	0	-13	71
Server B					
Coordinated	0	0	-7	-6	57
Uncoordinated	0	0	-1	-19	63
Coordinated, appr util	0	0	-3	-3	44
Coordinated, no feedback	0	0	-13	-7	66
Coordinated, no budget limits	0	-15	-18	-12	72
Uncoordinated, min Pstates	0	0	0	-19	50

Level	Changes to enable coordination
EC	Expose API to SM to change r_ref
SM	Expose API to EM and GM to change power budget
EM	Expose API to GM to change power budget
GM	Expose power budget violations to VMC
VMC	Use "real utilization"; use power budgets as constraints; explicit feedback to violations

# Models and sensitivity



# Simulation methodology

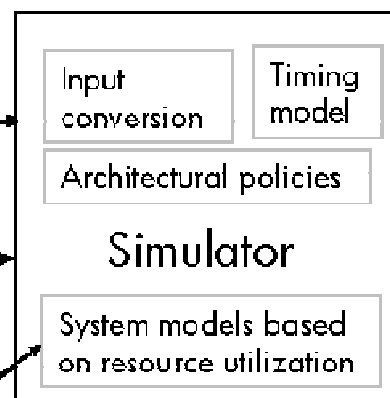


System monitoring & trace collection

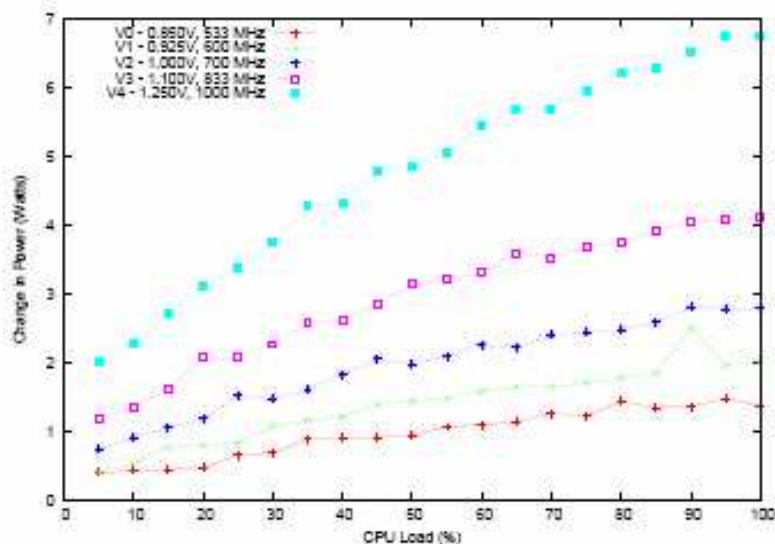
Application resource utilization traces

System configuration

System exerciser and model calibration



System simulated behavior & metrics



$\delta$

