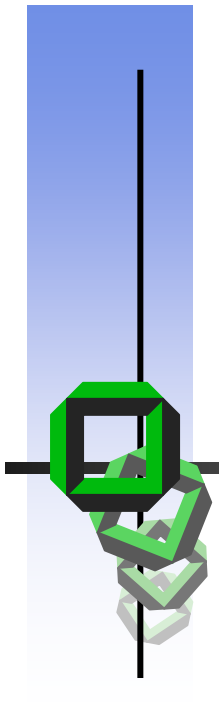


Distributed Energy Management for Virtual Machines

Gesellschaft für Informatik, Fachgruppe Betriebssysteme

Fujitsu Siemens Computers GmbH München

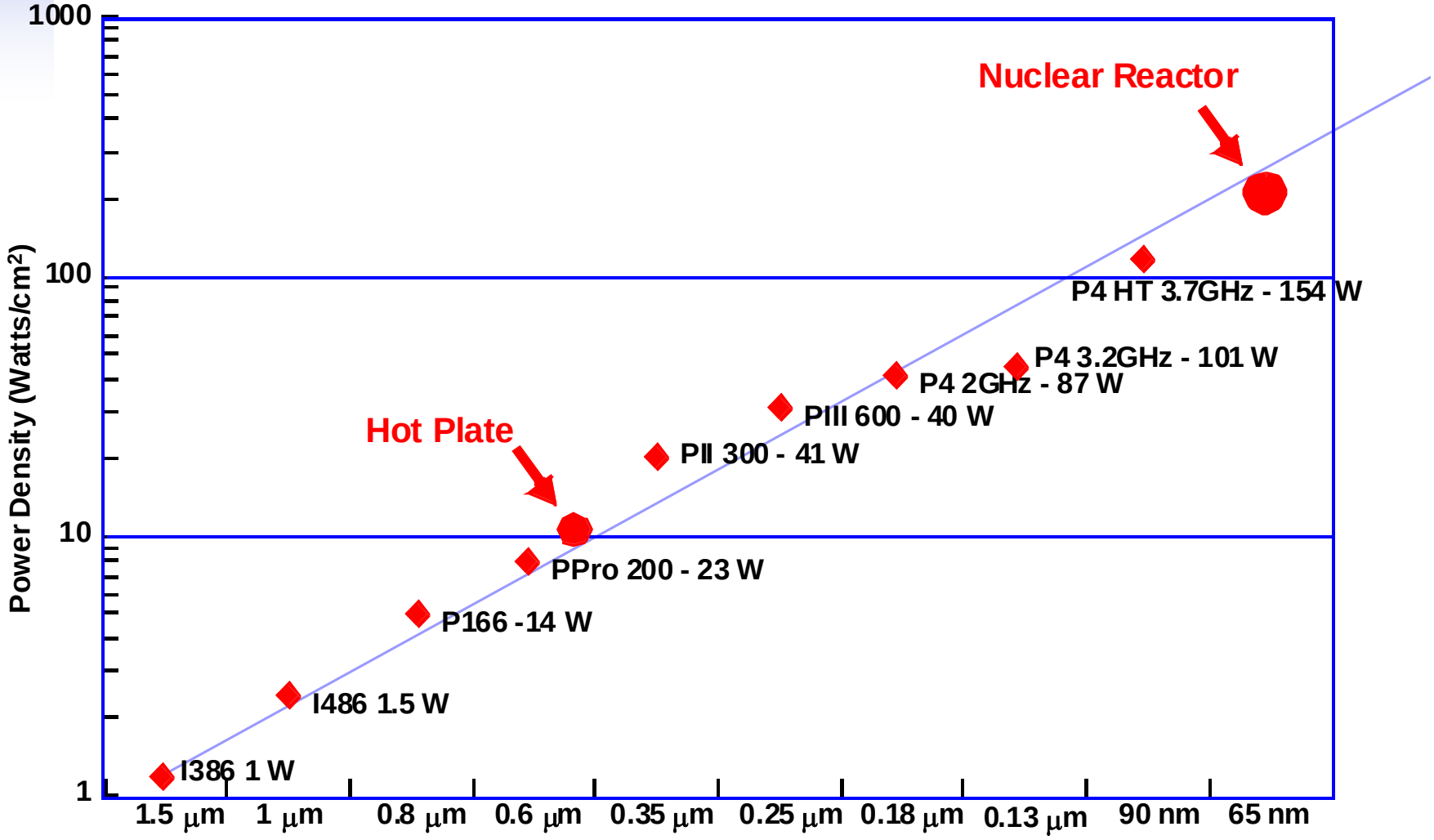
October 2006

Jan Stoess, Christian Lang, Frank Bellosa

System Architecture Group

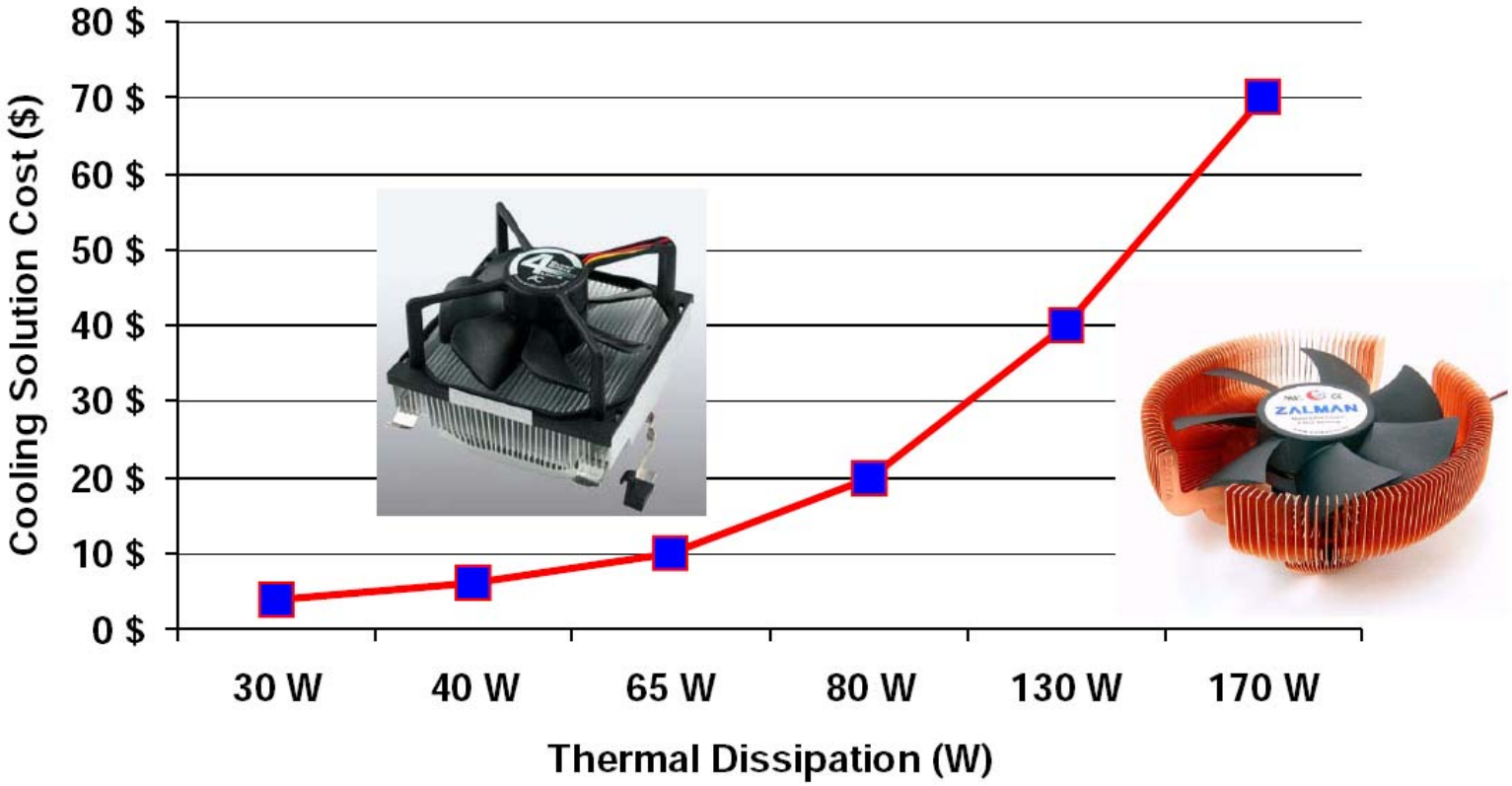
University of Karlsruhe

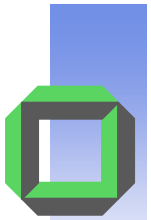
Moore's Law for Power



State of the Art in Thermal Design

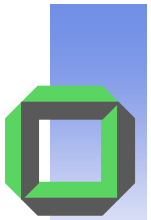
- Increasing energy dissipation in computer systems
- Increasing cooling costs





Goals of OS-Directed Energy Management

- Power management
  - Limits of battery/backup-generator
  - Peak power contracted with electric company
- Thermal management
  - Ensure safe operation in case of cooling failure
  - Design for the average case, no overprovisioning
  - Reduce energy and cooling cost
- User management
  - Control power/temperature per user
  - Ensure QoS and energy constraints at the same time
  - Pay-per-use model

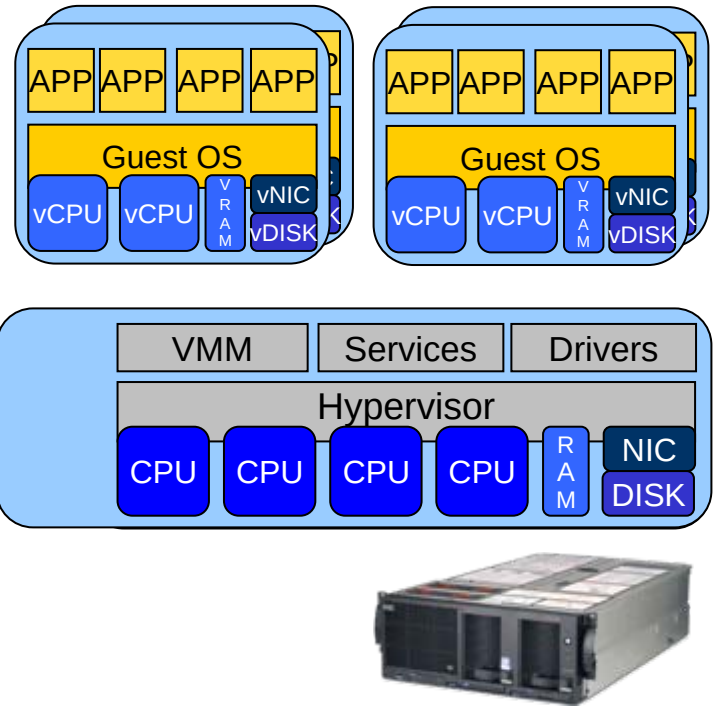


State-of-the-Art in Energy Management

- Hardware mechanisms for reduction of active power
  - Basic idea: multiple device power states
  - Processor halt cycles
  - Processor frequency/voltage scaling
  - Multi-speed disks
- Energy-management in current OSes
  - Widely used: system-wide throttling
    - Low power state during inactivity
    - Low power state on thermal/power emergencies
  - More recently: user-centric energy management
    - Throttling of energy-intensive low-priority tasks
    - Balancing hot and cold tasks among devices

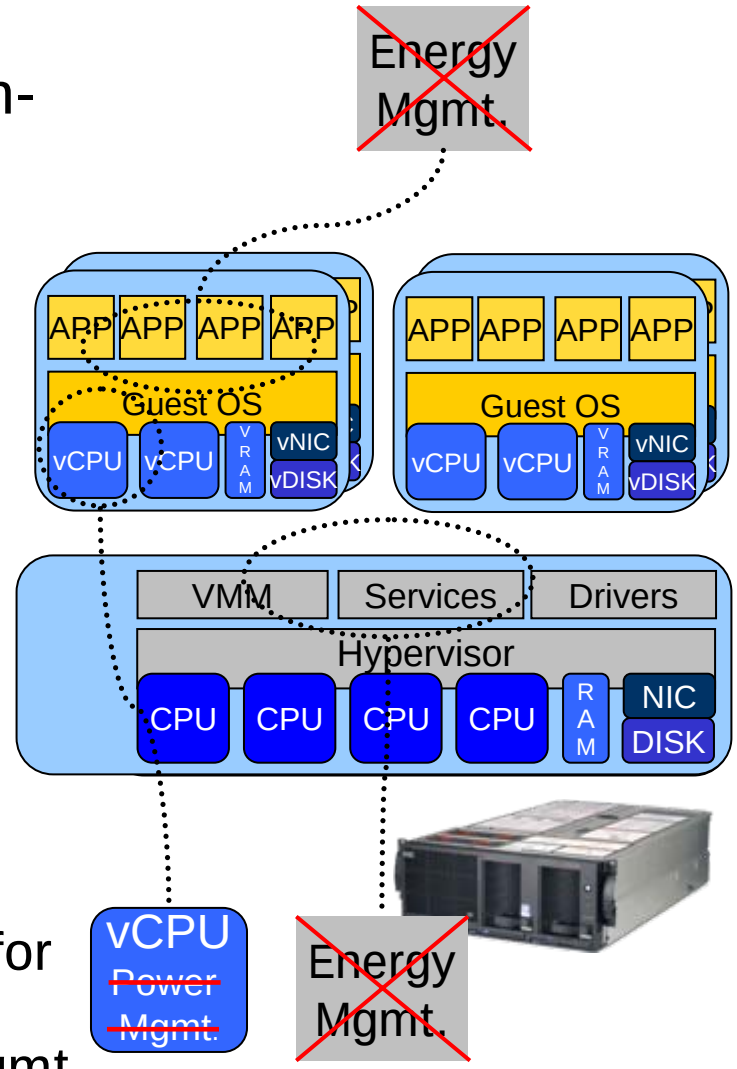
The Trend to OS Virtualization

- Virtual infrastructure
  - Multiple guest OSes
  - Hypervisor
  - Drivers and service containers
- Improve/innovate OS functionality
  - Server consolidation
  - Live migration
  - Improved reliability
- Retain legacy compatibility
  - Convenient migration path
  - Simultaneously use multiple OS personalities



Power Management vs. OS Virtualization

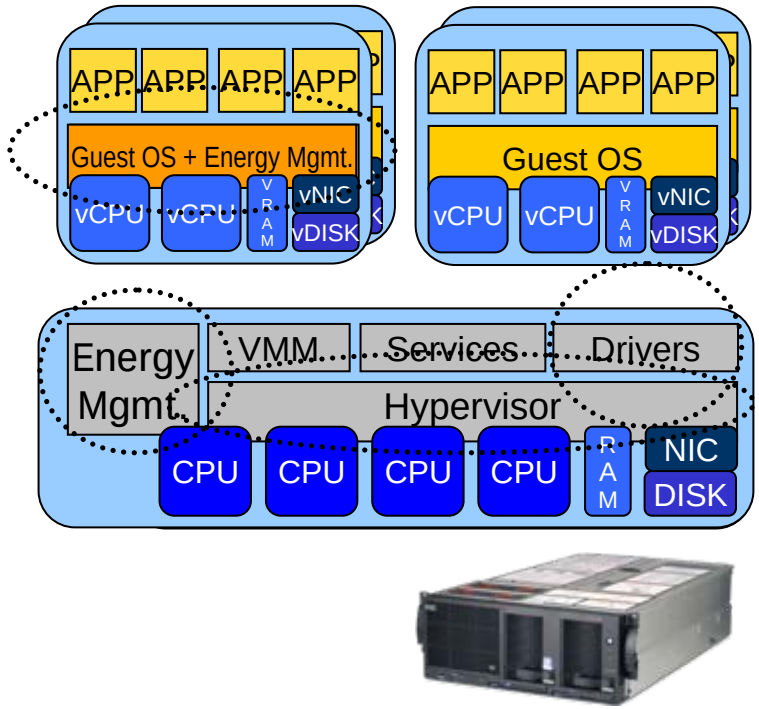
- Energy-management virtualization-unaware
  - Assumes full control over devices
  - Assumes full control over apps
  - Guest OSes have none
- Virtual infrastructure energy-unaware
  - Distributed control over devices
  - Distributed control flow and applications
  - Physical effects hard to virtualize
- Required
  - Distributed energy management for VM environments
  - Virtual energy for user-centric mgmt.



Distributed Energy Management

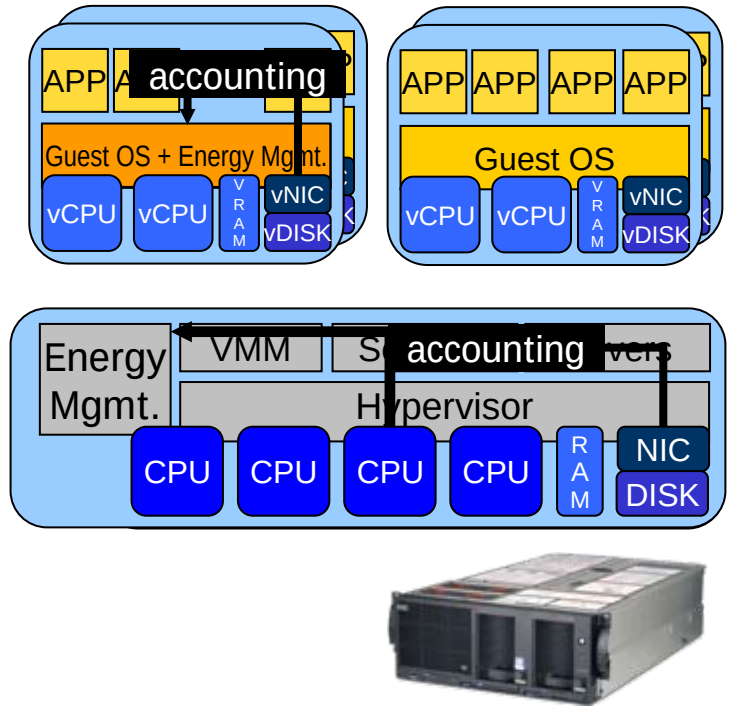
- Novel framework for distributed, multi-layered OS environments
- Use energy as basic notion
  - Unified, partitionable, distributable
  - Express thermal constraints via thermal model
- Assume parties to be distributed
  - Energy management substeps

1. Energy accounting
2. Analysis and allocation decision
3. Power control and allocation



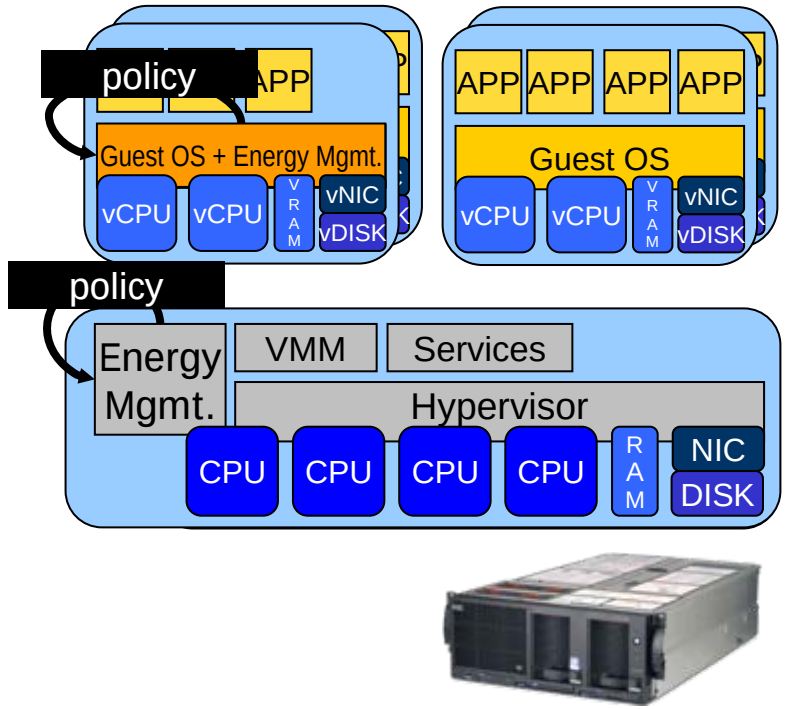
Distributed Energy Management

1. Distributed energy accounting
   - Measure/estimate energy per activity and resource
   - Incorporate layers and dependencies



Distributed Energy Management

1. Distributed energy accounting
   - Measure/estimate energy per activity and resource
   - Incorporate layers and dependencies
2. Separate policy management
   - Multiple subsystems
   - Each responsible for distinct devices and activities

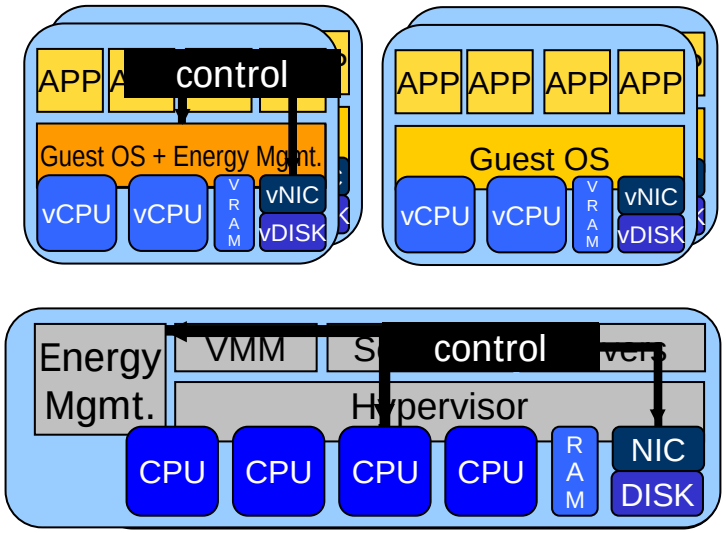


Distributed Energy Management

1. Distributed energy accounting
   - Measure/estimate energy per activity and resource
   - Incorporate layers and dependencies

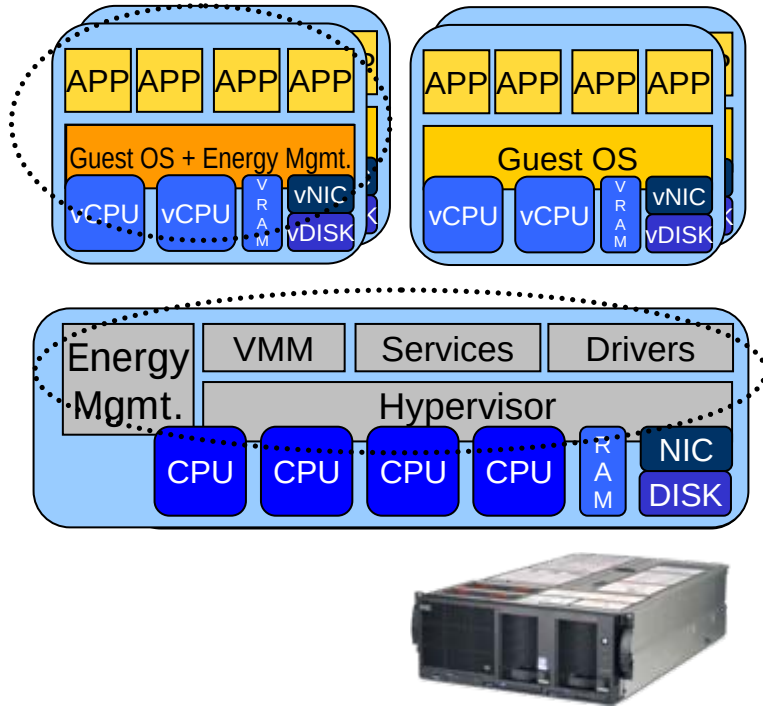
2. Separate policy management
   - Multiple subsystems
   - Each responsible for distinct devices and activities

3. Exposed control of active power
   - HW: halt cycles, DVFS
   - SW: Activity throttling, migration



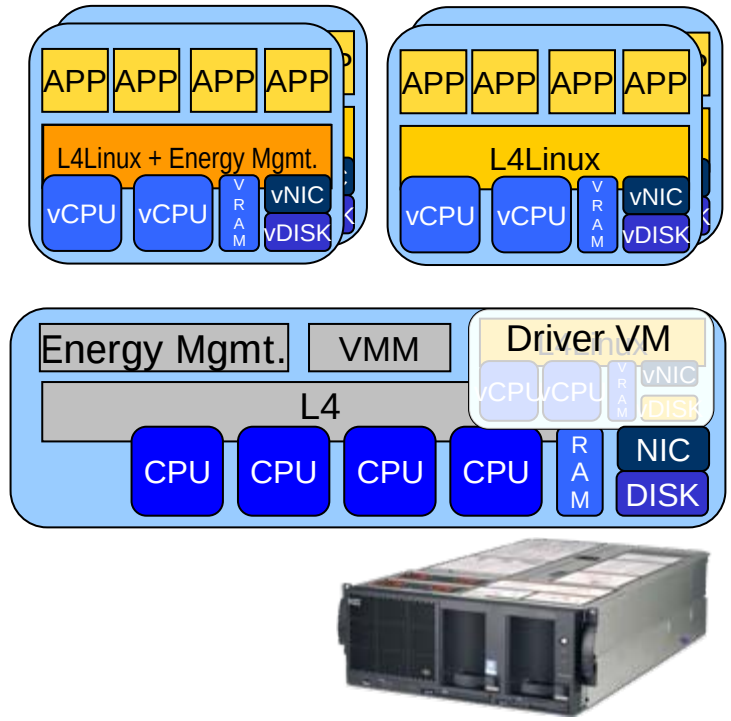
A Prototype for Hypervisor-Based Systems

- Two-level system, based on the previous design principles
- Host-level energy management
  - Constrains energy across VMs
  - Enforces constraints for energy-unaware or malicious VMs
  - “Virtualizes” energy
- Optional guest-level energy management
  - Fine-grain, application-level energy management
  - Linux-compatible



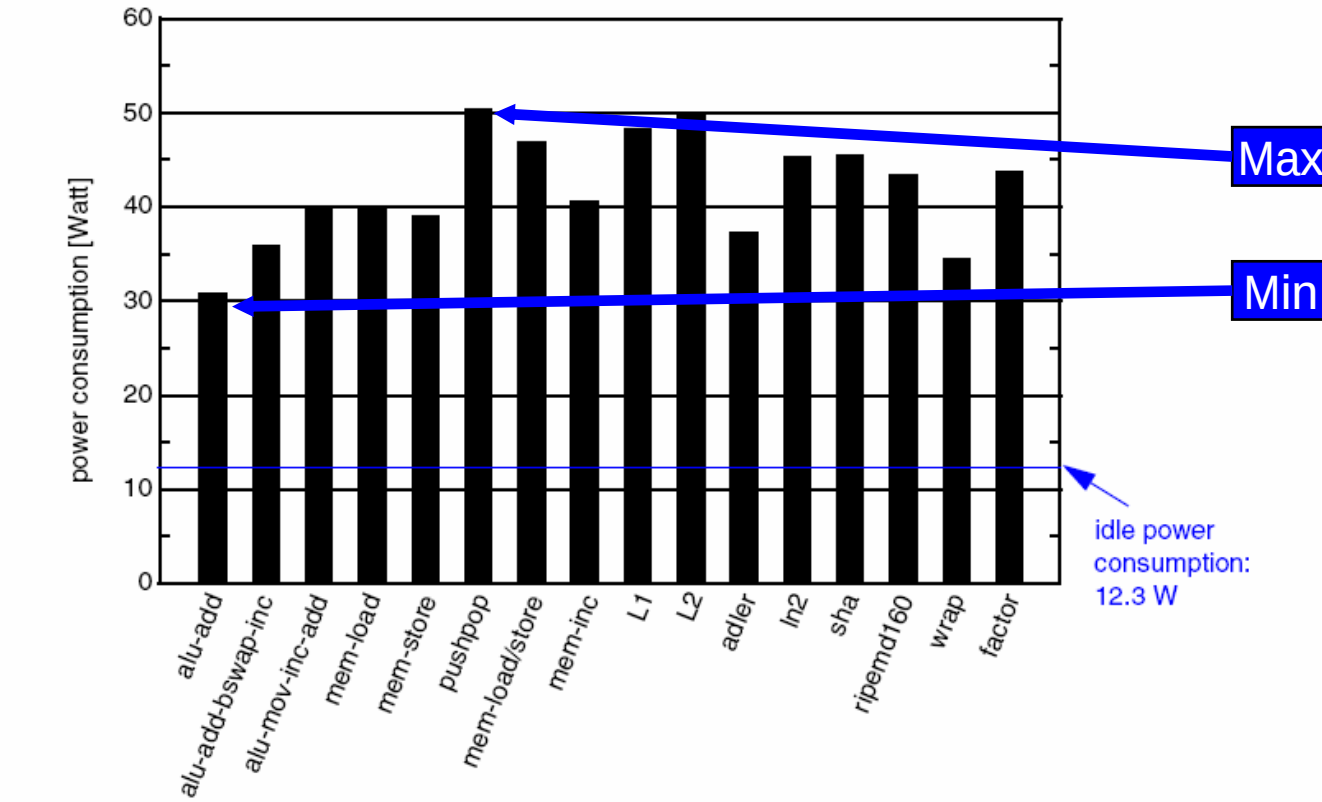
A Prototype for Hypervisor-Based Systems

- Based on L4 hypervisor/microkernel
- Virtualization technology
  - Para-virtualized L4Linux instances
  - Legacy application workload
  - Energy-aware/unaware guests
  - User-level device drivers
- Microkernel paradigms
  - Host-level energy manager app.
  - Lightweight abstractions
- Currently supported devices
  - CPU
  - Hard disk



CPU Energy Accounting

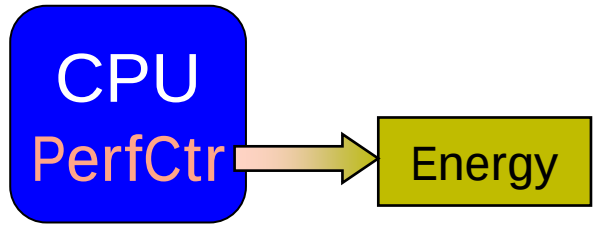
- P4 2GHz with compute intensive tasks (100% CPU load)



- Required: high resolution energy accounting

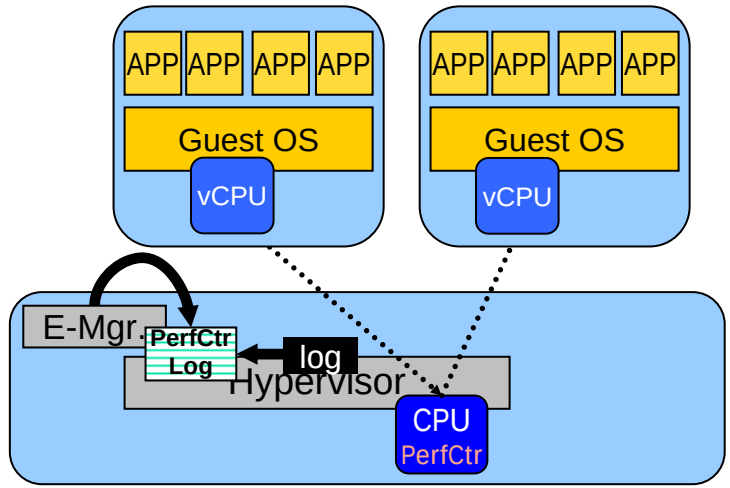
CPU Energy Accounting

- Event-driven energy estimation
  - Leverage performance monitoring counters
  - Energy = Σ_i #event$_i$ x weight$_i$



- Report to energy manager
  - Hypervisor traces PerfCtrs on vCPU context switches
  - Stores them in user-accessible PerfCtr logs
  - Energy manager derives active and idle energy from logs

vCPU1	*vCPU2*	*...*
8_i+12_a	8_i+20_a	

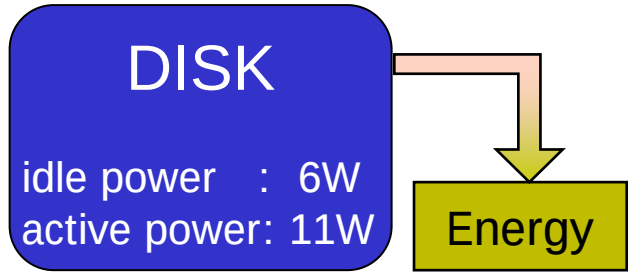


$$idle_{VCPU} : \frac{1}{\#vCPU} \Sigma_{vCPU} \#event_{tsc,vCPU} \times weight_{tsc}$$

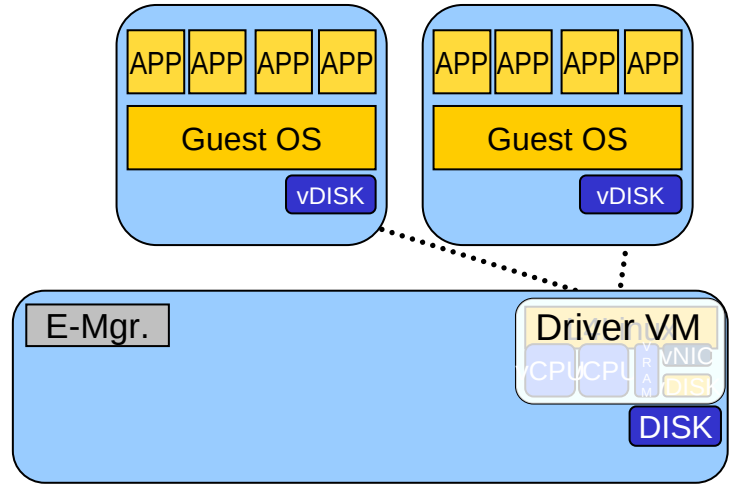
$$active_{vCPU}: \Sigma_{i \setminus tsc} \#event_{i,vCPU} \times weight_i$$

Disk Energy Accounting

- Time-based energy estimation
  - Assume constant disk power
  - Idle energy accounted to client VMs
  - Active energy accounted to requests
    - based on their transfer time
    - $time_{req} = size_{req} / transfer\ rate$



- Report to energy manager
  - Driver VM periodically obtains active and idle energy per vDISK



vDISK1	*vDISK2*	*...*
3_i+2_a	3_i+3_a	

$idle_{VDISK} : \frac{1}{\#VDISKS}$ idle power x period

$active_{vDISK} = \Sigma_{req(vDISK)}$ active $power_{req}$ x $time_{req(vDISK)}$

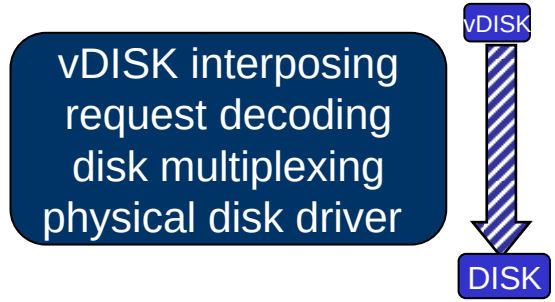
Recursive Energy Accounting

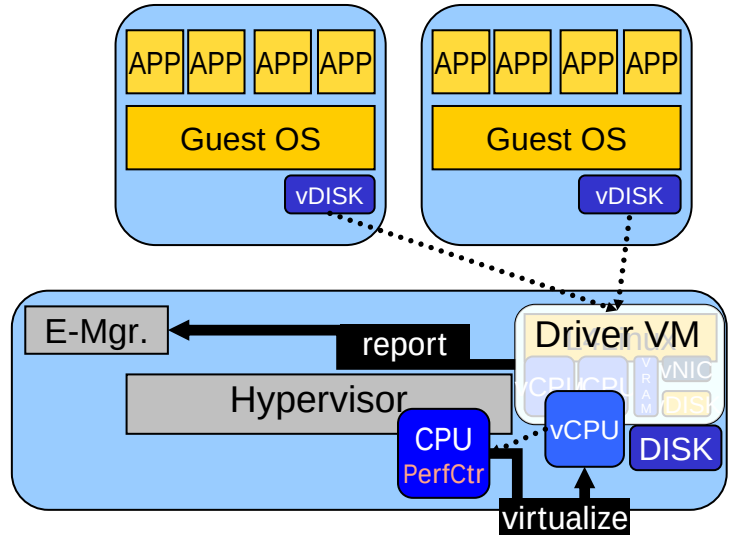
- Disk virtualization requires CPU processing
- Recursively account CPU energy
  - Disk driver VM
    - Accounts per-VM disk utilization
    - Accounts its own vCPU utilization

	vDISK1	*vDISK2*	...
DISK	3_i+2_a	3_i+3_a	
vCPU	4_i+4_a	4_i+6_a	

  - Hypervisor
    - Accounts and virtualizes driver CPU energy consumption

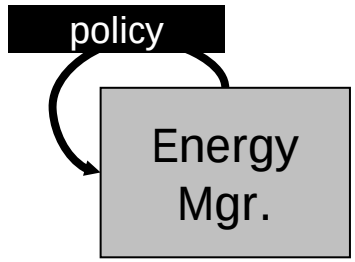
	vCPU
CPU	6_i+12_a



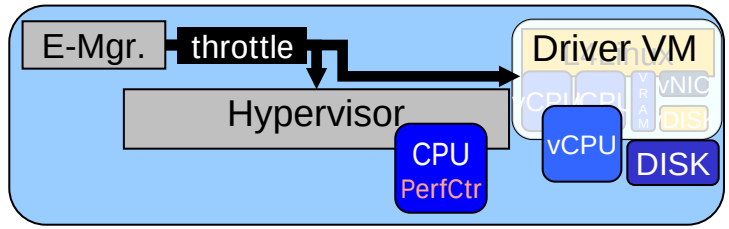


Exposed Power Control

- Energy manager
  - Obtains current CPU & disk energy
  - Limits it via per-VM energy reallocation



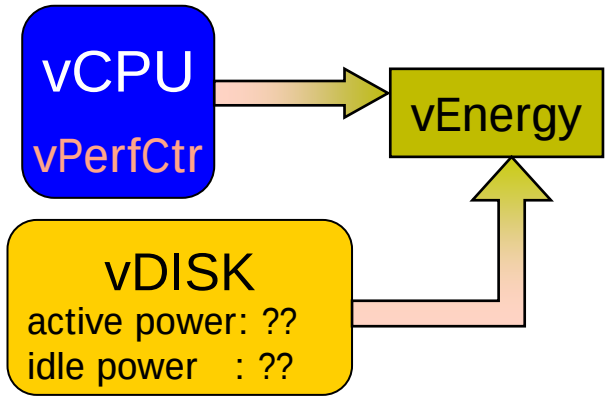
- CPU allocation
  - Hypervisor uses stride scheduler
  - Strides exported to energy manager
  - Unalloted cycles translated into halt cycles

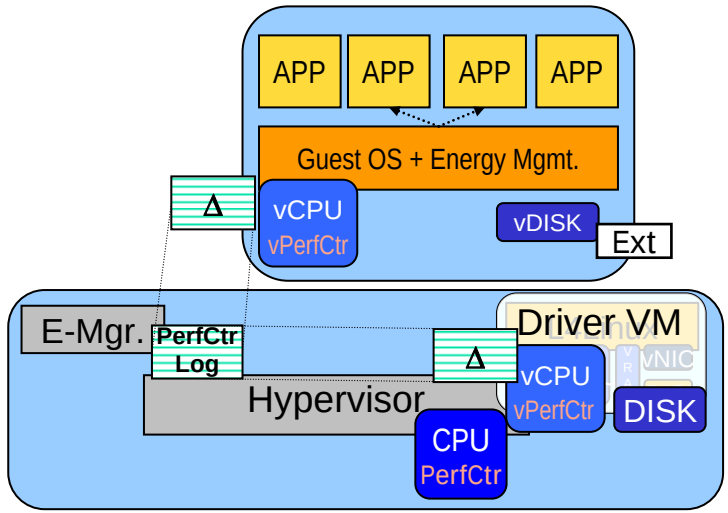


- Disk (and driver vCPU) allocation
  - Driver VM uses a throttle factor
  - Delays the throttled requests until the next timer tick

Guest-Level Support – Virtual Energy

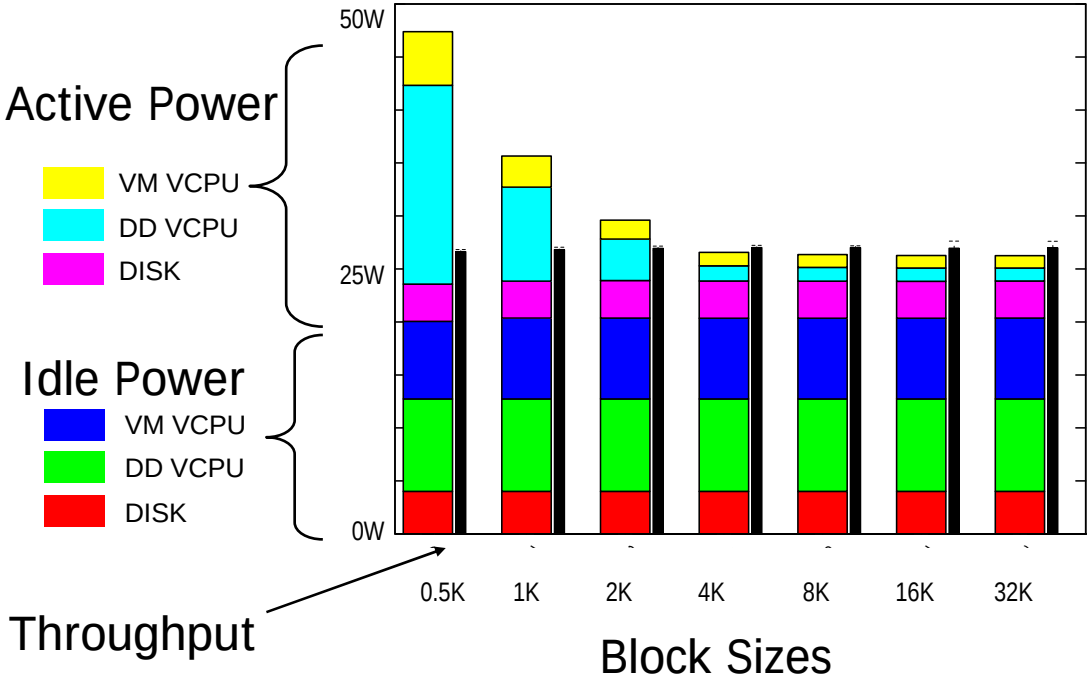
- Virtual CPU energy
  - Virtualize performance counters
- Virtual disk energy
  - Power depends on level of sharing
  - Use para-virtual disk extension
- Virtualizing Performance Counters
  - $PerfCtr_{virt} = PerfCtr_{phys} - \Delta PerfCtr_{OtherVMs}$
  - Get $\Delta PerfCtr_{OtherVMs}$ from logs
- Virtual disk extension
  - Idle energy exported as meter
  - Active energy exported with normal request data

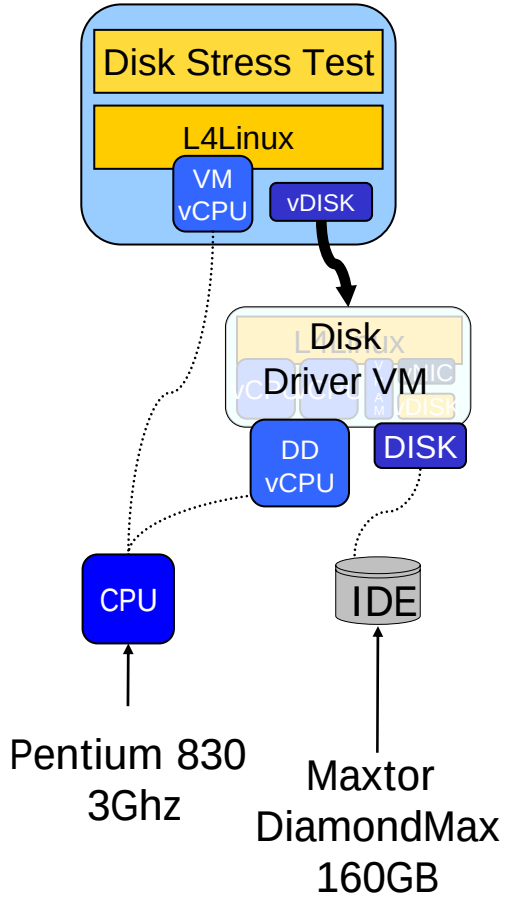




Experiments and Results

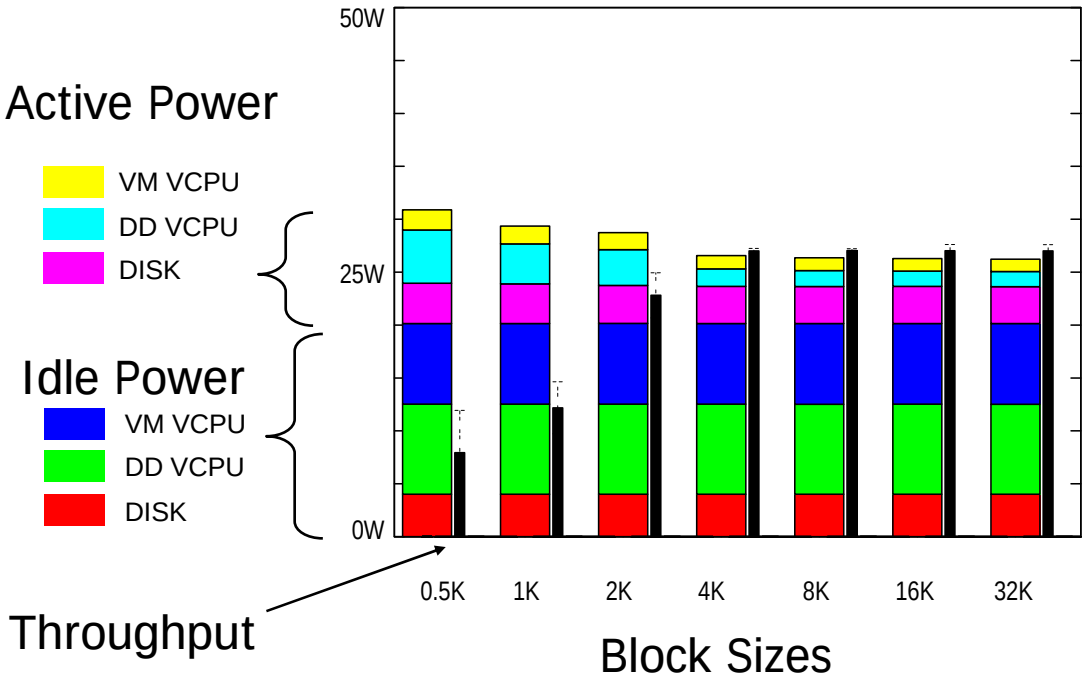
- Energy distribution of disk processing

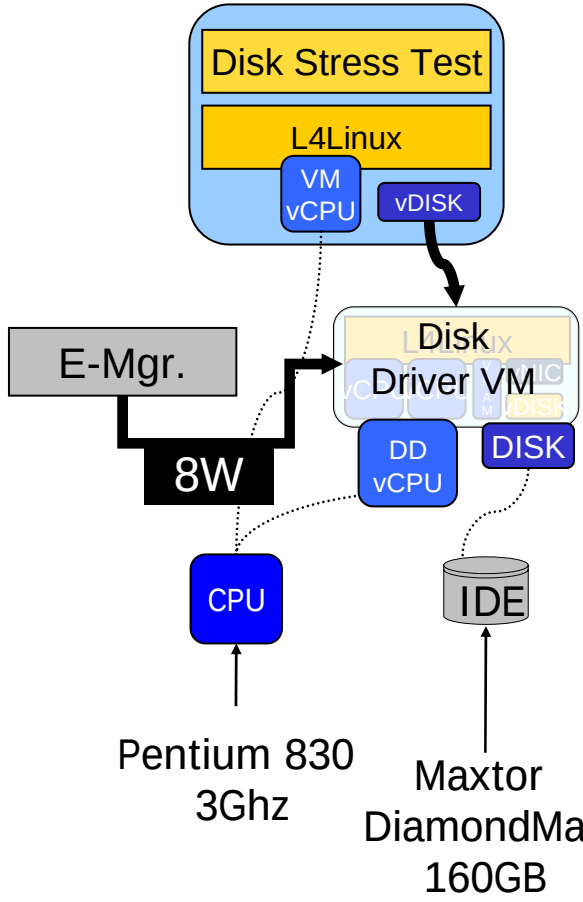




Experiments and Results

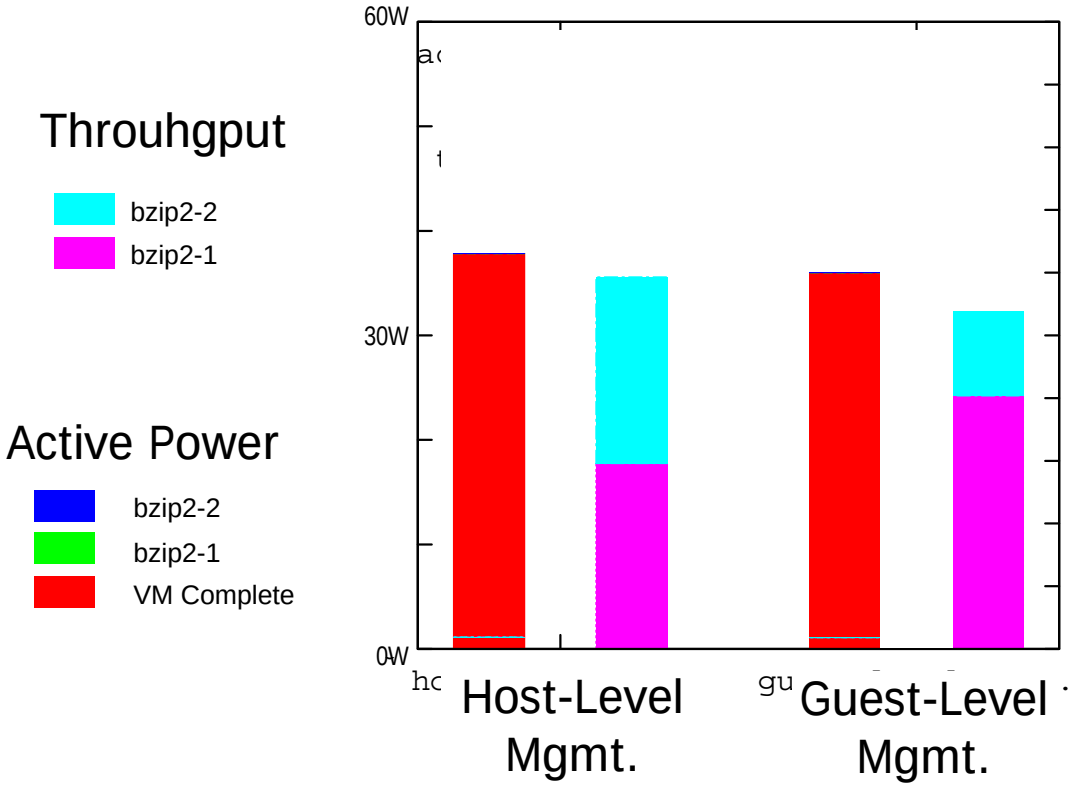
- Energy distribution of disk processing

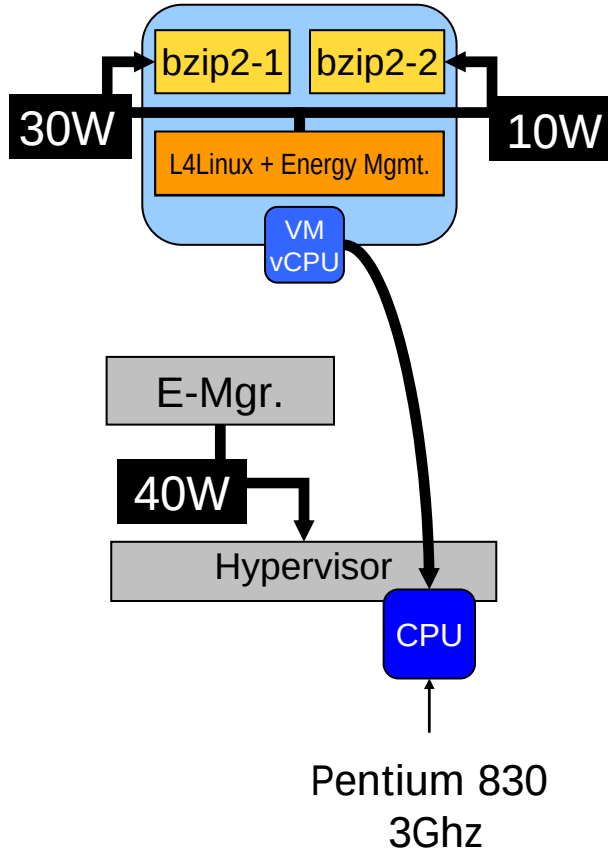


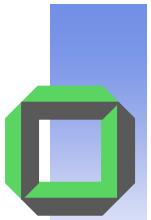


Experiments and Results

- Two-layer CPU energy management







Conclusion & Future Work

- The Problem
  - Current energy-management virtualization-unaware
  - Virtualization solutions disregard energy aspects
- Approach: Distributed energy management
  - Distributed, recursive energy accounting
  - Exposed control over active device power
  - Prototype supports control of CPUs and disks
  - Fine-grain, enforceable management at two layers
- Future Work
  - Other device types and management policies
  - Full virtualization, many-core architectures