International Supercomputing (ISC) 2011
Tutorial

Sustainability and Energy Efficiency in Data Centers Design and Operation

Krishna Kant, George Mason University
David Du, University of Minnesota

Outline

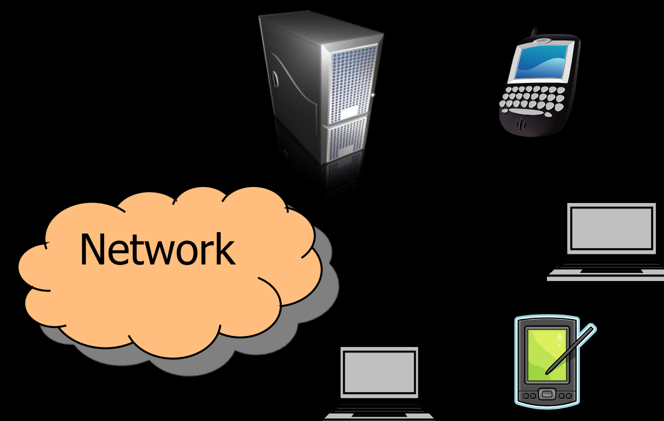
- Data Centers Energy & Sustainability Problem
- Sustainability in Data Centers
- Energy Adaptation in Data Centers
- Power States and Management
- Power Management Methods
- Network Power Management
- Storage Power Management
- Data Center Cooling
- Coordinated Power Management
- Conclusions & Future Challenges

ICT Power Growth until 2020

- Increase in spite of power efficient designs
  - Clients: 8x in number, 3X in power
  - Data Centers: > 2X increase
  - Network: 3X increase

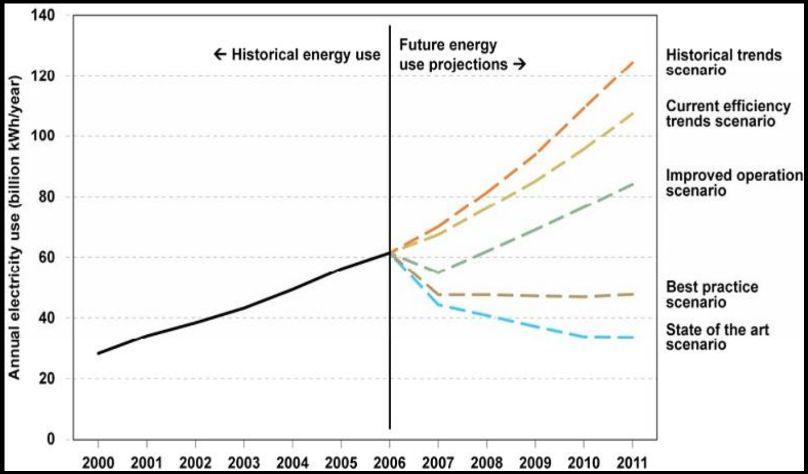
Transmission, conversion & distribution

Data Center



Need for Data Center Energy Efficiency

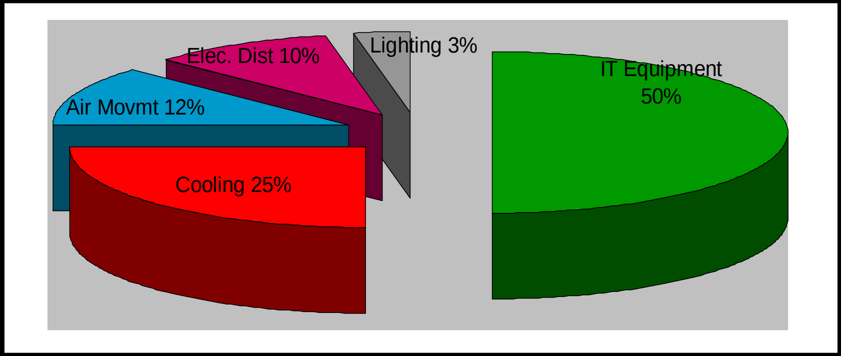
- Substantial energy consumption
  - 2007: ~1.5% of US total electricity consumption, $5.0B annual cost, 20-40% of operational cost
  - 2020: Up to 10% of total, much higher fraction of operational cost.
- Issues:
  - Concentrated demand on power grids
  - Environment impact.
  - Sustainability issue s – use of resources



EPA DC power projections in 2007

Energy Use in Data Centers

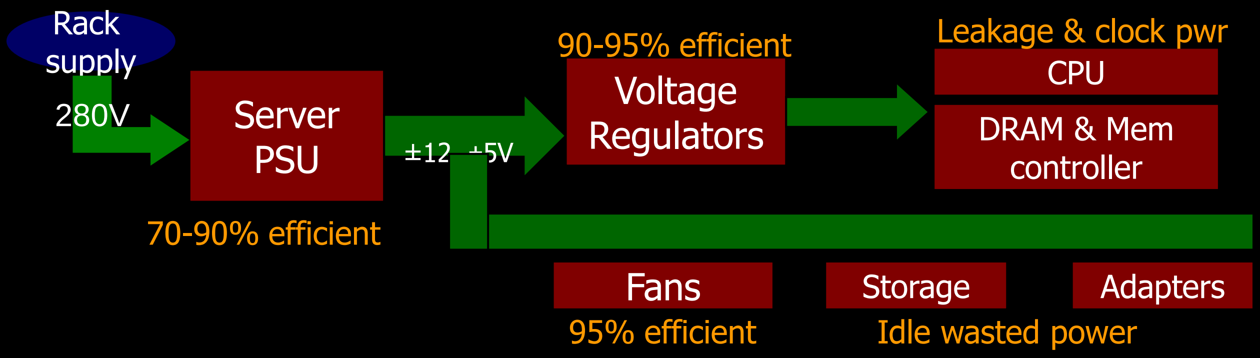
- Data Center Power Consumption
  - 50% HVAC
  - 20-35% Servers
  - 10-25% Storage
  - 5% Networking



- Different Types of data centers
  - Compute Centric (Ex: HPC)
    - 35% Servers, 10% Storage, 5% Networking
  - Data Centric (Ex: Enterprise)
    - 20% Servers, 25% Storage, 5% Networking
  - Average Case
    - 25% Servers, 20% Storage, 5% Networking

IT Equipment Efficiency

50% power wasted!



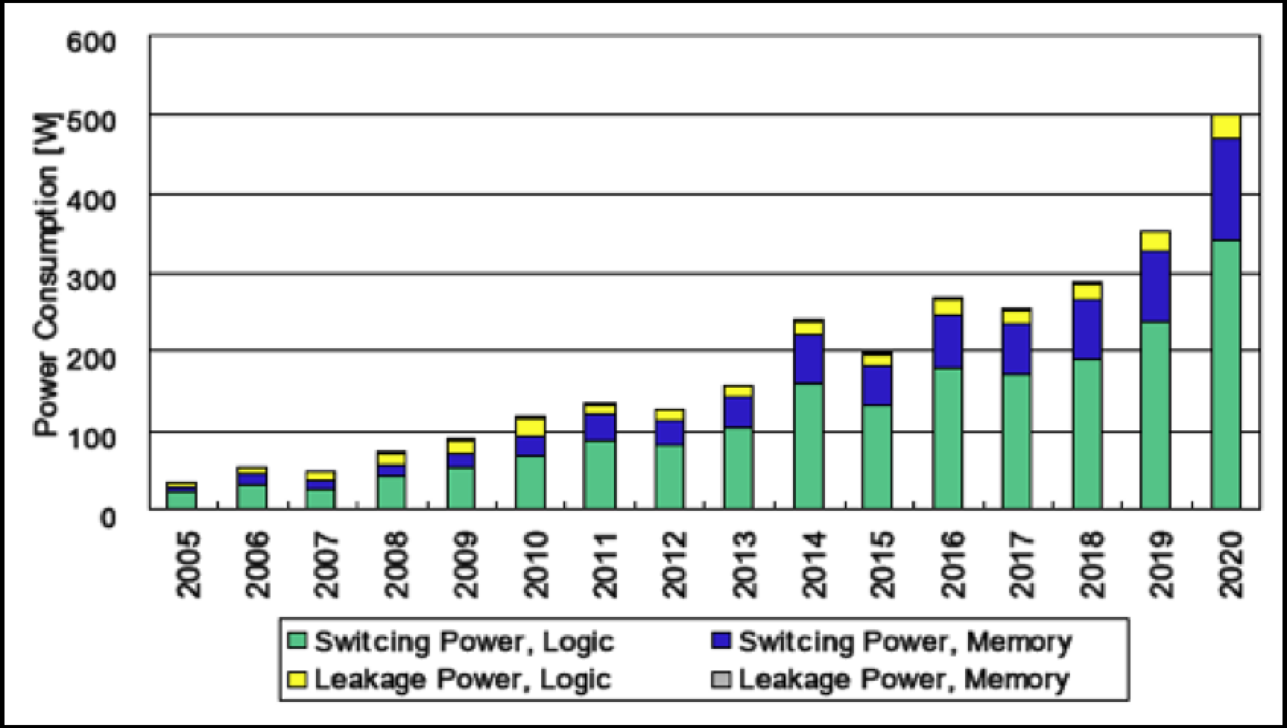
Component	Total	Used	Comments
CPU	80	60	Operating at 100% utilization
Fans	50	25	Temp. directed fan at 100% util
Memory (32 GB)	88	24	2GB DIMMS, 4W idle, 19W active
Hard drives	40	10	6 SATA drives, 25% busy
I/O adapters	20	4	25% disk, 15% network
Motherboard	22	12	N/S bridges & devices, VR's, …
Total DC power	**300**	135	
Power supply loss	50	7	14% → 5% loss of AC input pwr
AC input power	**350**	142	> 50% of power is wasted

Does Moore's Law Solve the Problem?

- No!
  - Per transistor power goes down as the feature size shrinks, but
    - Increasing number of transistors per chip
    - Increasing operational speeds → More power
  - Voltage margins already very small →
    - Voltage downshift to lower power is disappearing!
- It's even worse …
  - Wires don't scale: nonlinear increase in power
  - Increasing leakage current: present even when idle

Technology Trends

- Power increase in-spite of feature size reduction
  - More transistors, Leakage, wire power, switching rate, …

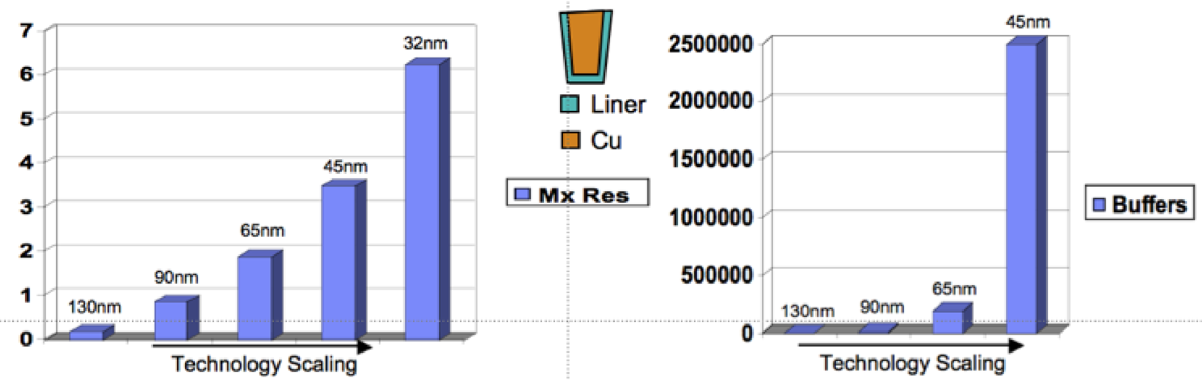


Technology Trends
Wires don't Scale

Number of Repeaters is Exploding as a Power of 10 per 33% Shrink

IBM

Chip Integration – Technology Challenges



- **A fundamental Shift in technology has occurred in terms of interconnects**
  - Mx resistance is increasing at an alarming rate
  - High Resistance drives repeater challenges
    - 130nm-2000, 90nm-20K, 65nm-193K, 45nm ~2-3M
  - Costs us lots of power with buffers being the leakiest and accounting for > 50% of logic leakage.

Sematech/ACM Thermal & Design in 3D IC's, 2007

Smart Energy Mgmt is Essential

- Hardware Level
  - Clock gating & other circuit mechanisms
  - Aggressive power mgmt at each level
    - CPU cores, caches, interconnect, …
    - Subsystems: CPU, DRAM, mem controller, links, adapters, …
  - Coordination within and across level levels
- Server Level
  - Fans, power supplies, system power states, ...
  - OS, SW, VM & app level power mgmt
- Data Center Level
  - Cooling & airflow management
  - Cooling/thermal aware placement/scheduling, …

Is Energy Efficiency Enough?

- Operational energy a substantial target to reduce, but …
- Energy efficiency less important, its carbon footprint really matters
- Data Centers are very infrastructure heavy
  - Use a lot of materials (metals, water, …)
  - A substantial carbon & energy footprint
- Energy efficiency does not reduce energy usage!
  - Rebound effect, Jevons paradox

Cooling Infrastructure

- Cooling is very resource intensive
  - Lot of materials
  - Water, much of which evaporates

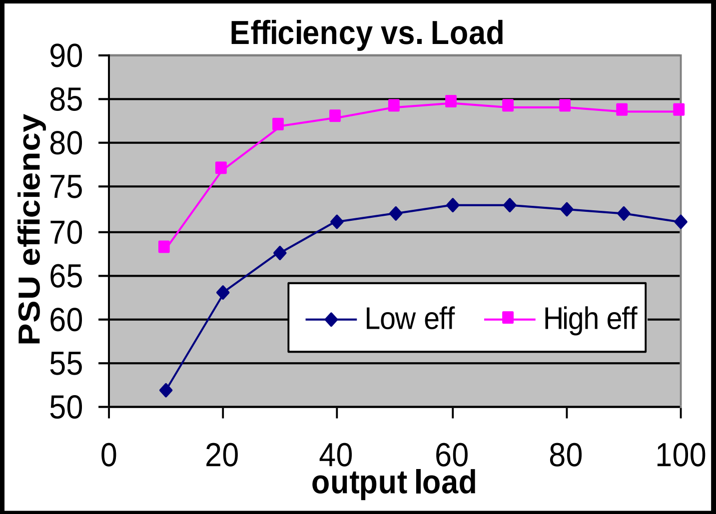
Power Distribution Infrastructure

- 9-10% distribution loss at power source
- Lots of earth's resources used (metals, rare earths, …)

Overdesign

- Overdesign is the norm
  - Data Center Level: Huge UPS, Generators, dist. frames, …
  - Server Level: Large power supplies, fans, heat sinks, …
  - Others: All resource much larger than needed

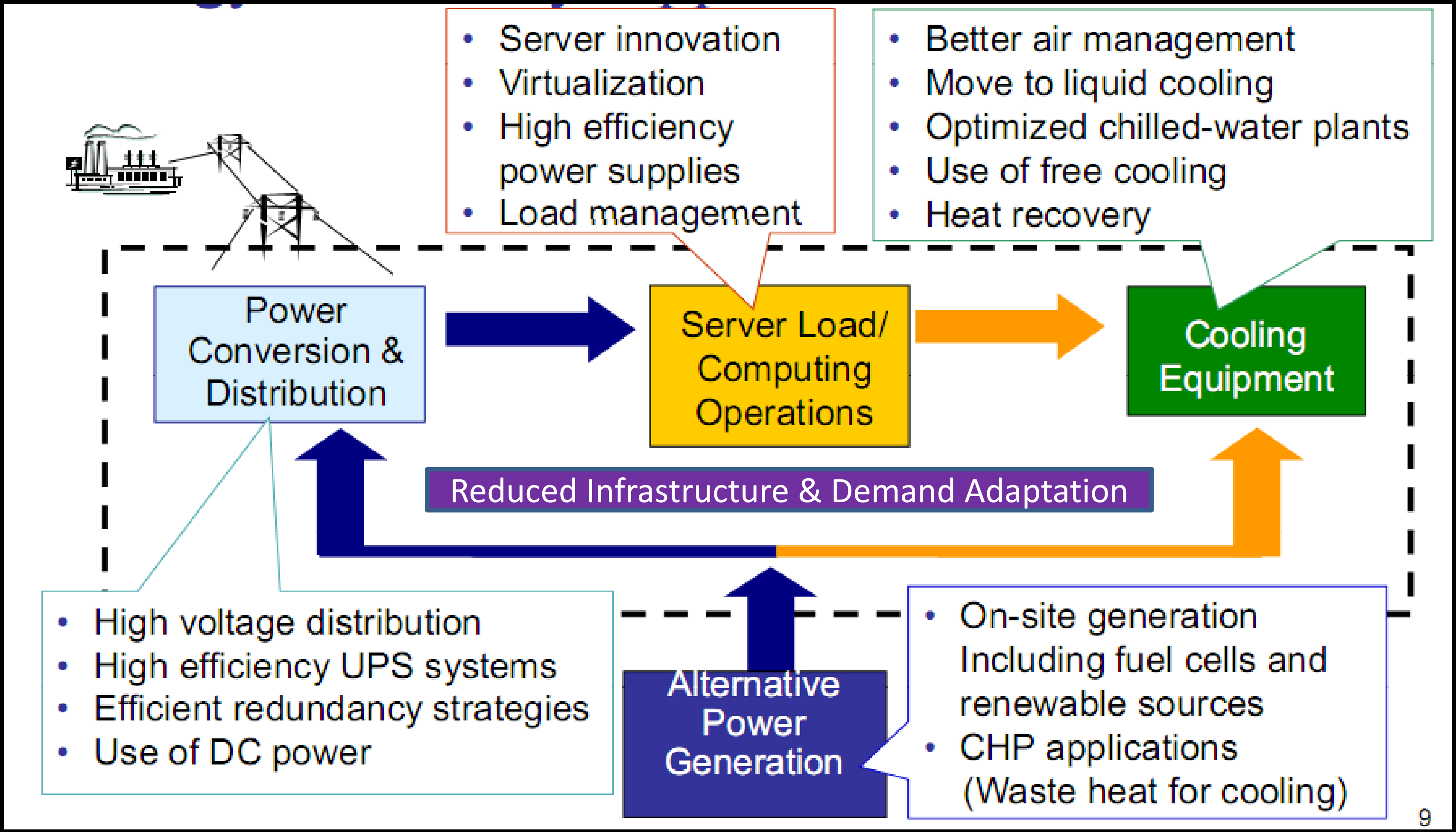
- Engineered for worst case
  - Huge waste of power, materials, …
- Example: Power Supply
  - Most PS run at very low utilizations, especially for dual redundant PSUs
  - Low utilization ➔ Low efficiency
- Voltage regulators: Similar issues



Sustainability Considerations in Data Centers

- Facilitate use of renewable energy
  - Must deal with variability in energy availability
  - Available energy may be inadequate.
- Thrifty use of energy & materials in all stages
  - Free Cooling instead of CRAC
  - Reduce size of UPS, generators, …
  - Reduce capacities of power supplies, heat sinks, fans, …
- Smart adaptation to deal with undercapacity

Data Center Energy Opportunities

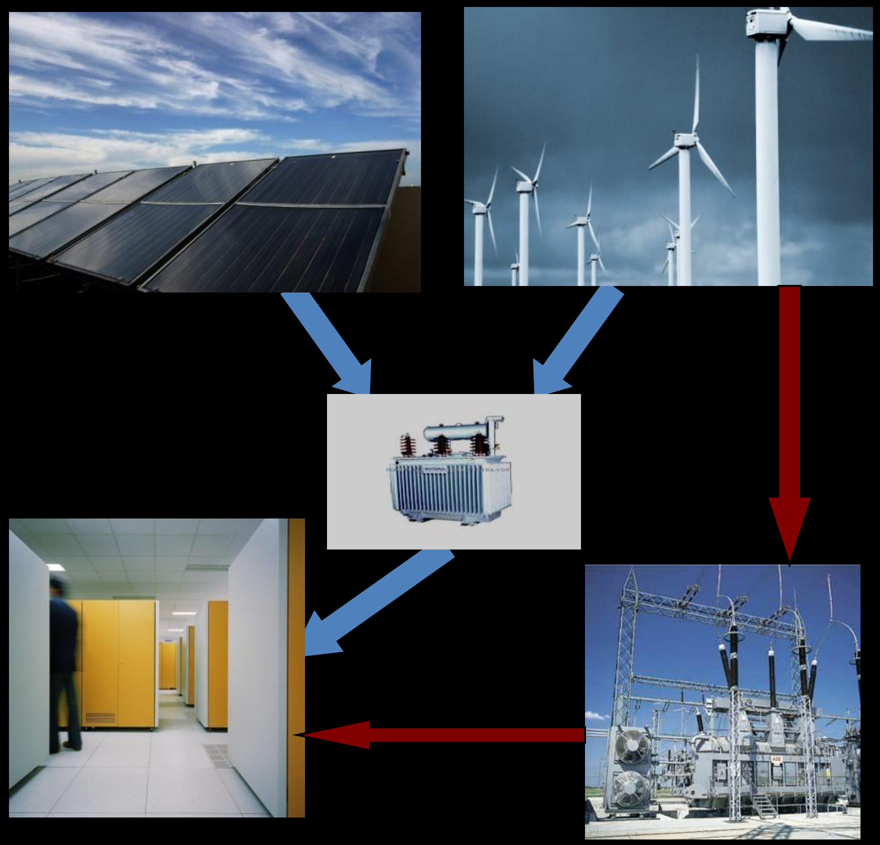


Source: US DOE: Data Center Energy Efficiency Program

Sustainability in Data Centers

Powered by Renewable Energy

- Limit or eliminate energy draw from grid
  - Less infrastructure & losses, but variable supply
  - Need to consider impact on both computing & communications
- Similar issues with unreliable grid supply



Need better power adaptability

High Temperature Operation

- Chiller-less data centers
  - Less energy/materials, but space inefficient

- High temperature operation of comm./computing equipment
  - Smaller $T_{outlet} - T_{inlet}$
  - Deal with occasionally hitting temp. limits.

Need smarter thermal adaptability

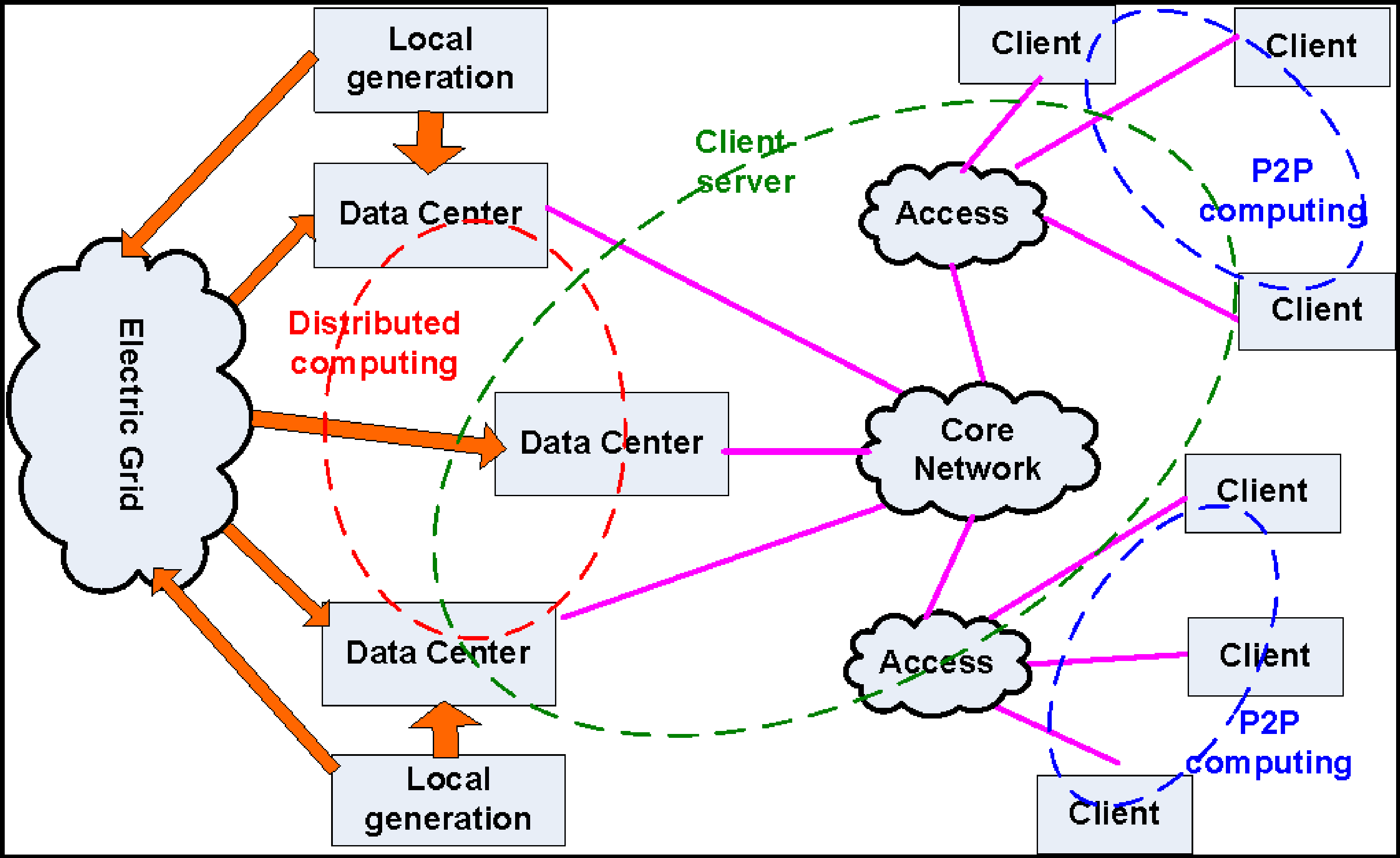
Energy Adaptive Computing

- Dynamic end to end adjustment to
  - Workload adaptation
    - What to run, at what precision, granularity, …
  - Infrastructure adaptation
    - Where to run, when to run, and how well
- What's new?
  - Mandatory, rather than opportunistic power and thermal mgmt.
  - Coordination across compute, network & storage.
  - Integration of workload/infra adaptation

Adaptation Methods

- Workload Adaptation
  - Shut down low priority tasks
  - Degraded service
    - Lower resolution, precision, partial service, …
- Infrastructure Adaptation
  - Load consolidation & migration
  - QoS degradation
    - Higher delay (Batched service, mandatory sleep mode use)
    - Lower tput (lower freq/voltage, “width” control, …)
- Workload adaptation always done first (this paper)

EAC Instances



Client-server EAC

- Transparently adapt to client energy states
  - State = {on-AC, normal, low-battery, …}
  - Service contract Ci = {setup QoS, operational QoS}
- Adaptation Challenges
  - Communicating & enforcing contracts.
  - Group adaptation of clients forced by network/servers ?

Cluster EAC

- Adaptation to intra & inter-DC limits
  - Multi-level: Server, rack & DC levels
- Adaptation Challenges
  - Estimate & collect power deficits/surplus at multiple levels
  - Coordination across large range of devices
    - Location based services
    - Coordination across levels
  - Simultaneously handle client-server loop

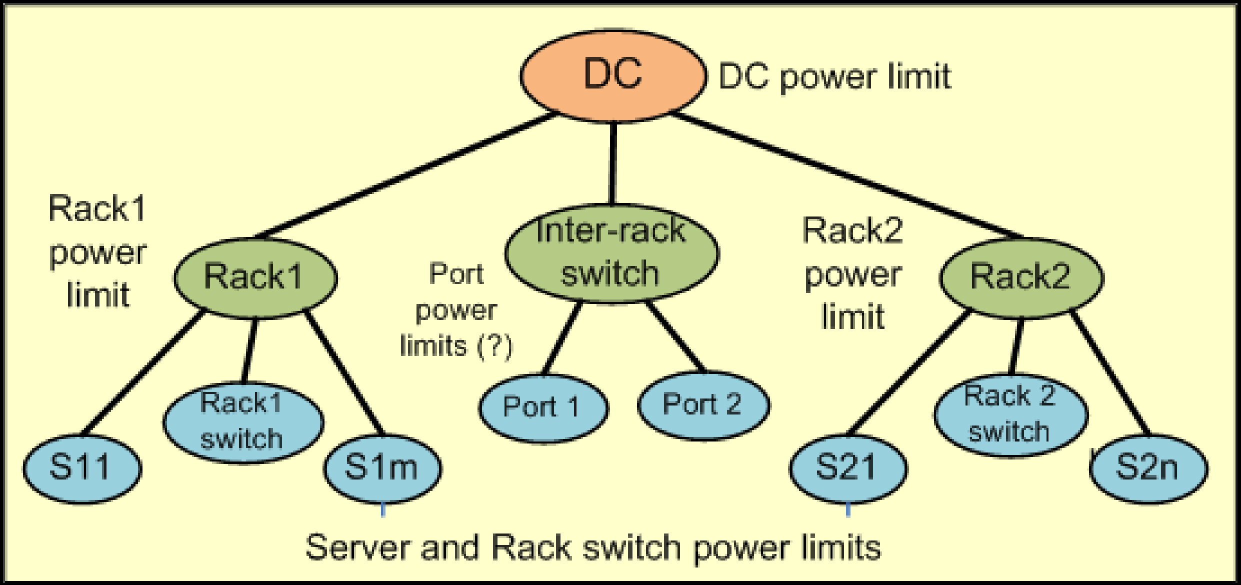
P2P EAC

- Adaptation based on “available energy”
    - Content: video resolution, audio coding, …
    - Network: modulate wireless radio usage (?)
    - Energy proportional use of peer resources
    - Energy driven content replication & reorganization
- Adaptation Challenges
    - – Satisfying QoS ?
    - – Balancing src/dest usage vs. relay node energy usage ?

Energy Adaptation in Data Centers

Infrastructure Adaptation

- Need a multilevel scheme –
  – Individual "assets" up to entire data center
- Need both supply & demand side adaptations



Supply Side Adaptation

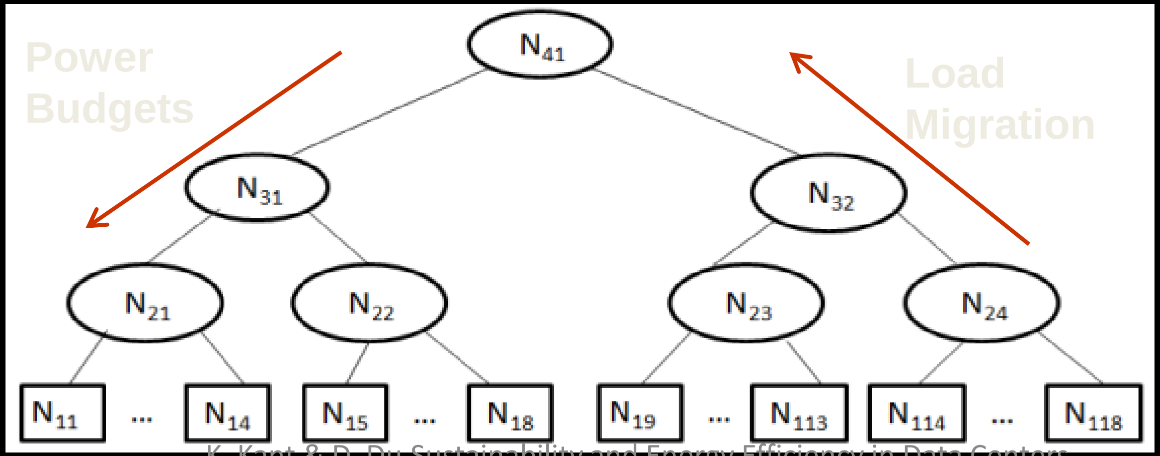
- “Hard” vs. “Soft” (artificial) limits.
  - Time const. depending on energy storage.
- Hard limits
  - Energy availability limits (at DC level) or lower levels (e.g., Power supply circuit limits)
  - Thermal/cooling related consumption limits
- Soft limits
  - Rationing at each level (servers & switches)
    - Allow independent adaptation further down
  - Load consolidation
    - Essential part of energy efficient operation, but needs to work with soft capping

Demand Side Adaptation

- Needs to deal with fluctuating demand
  - Dynamic migration & consolidation
  - Use of low power modes
    - For idled nodes (S3/S5) vs. active nodes (C, P, L, …)
- Combined supply & demand side adaptation
  - Imbalance: One node squeezed while other has surplus power
  - Ping-pong Control: Oscillatory migration of workload
  - Error accumulation down the hierarchy.

A Proposed Algorithm

- Systematic control
  - Power budgets changes move downwards
  - Load migration moves up the hierarchy, from local to global.
    - Local migrations are temporary & do not trigger changes to “soft” caps on supply.

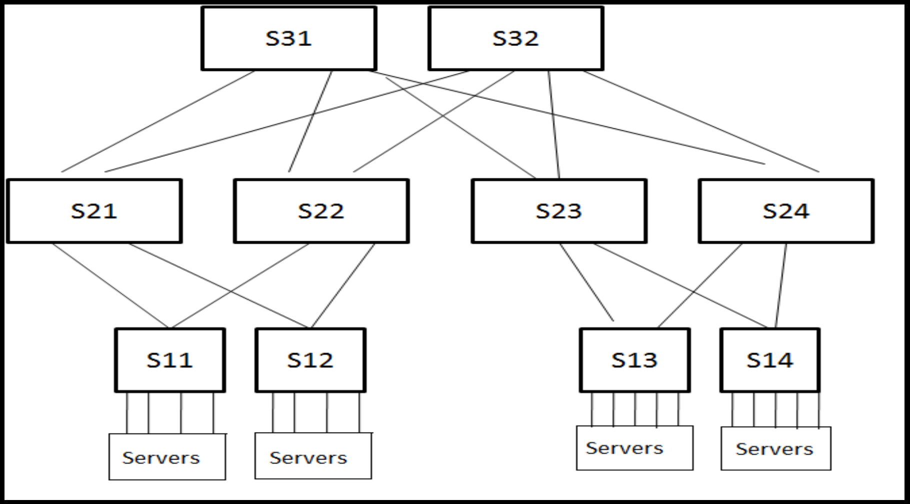


Proposed Algorithm

- Target Node selection
  - Based on bin packing (best-fit decreasing)
  - Allows for more imbalance, which can be exploited for workload consolidation
- Properties
  - Minimizes nonlocal migrations & ntwk traffic.
  - Avoids ping-pong, attempts to minimize imbalance
  - But, constraints limit certain adaptations.

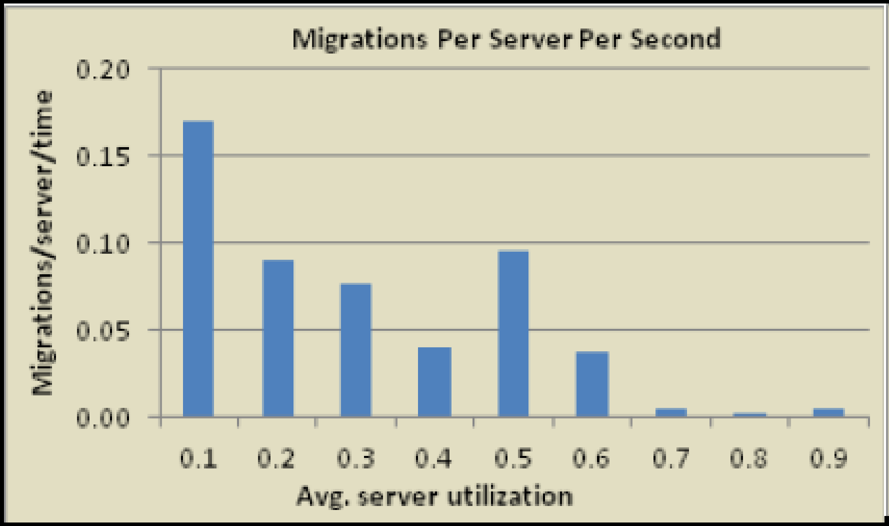
Experimental Results

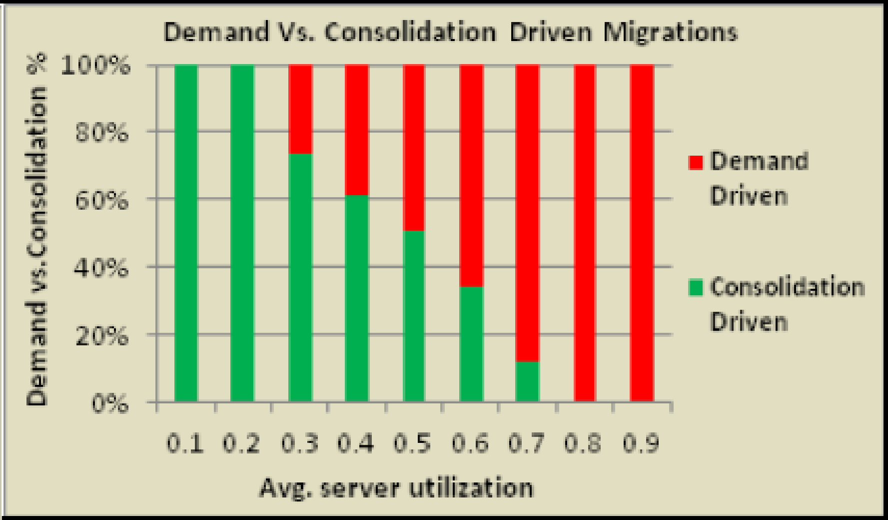
- Scenario
  - 3 levels, 18 identical servers (4+4 + 5+5)
  - Switch hierarchy identical to server hierarchy
  - 3 applications, total of 25 app instances
  - Any app can run on any server
  - Demand Poisson (active power ∞ utilization)



Migration Frequency

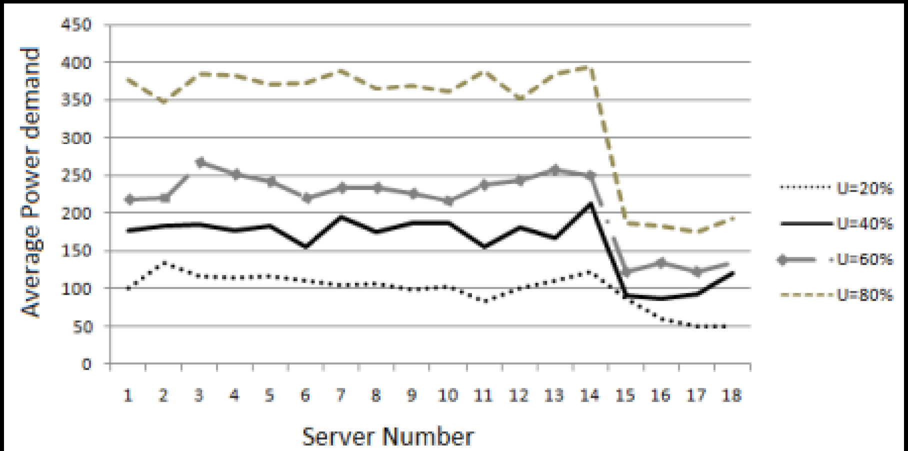
- Migration drivers: consolidation vs. energy deficiency
  - Low util ➔ Consolidation, High util ➔ Energy deficiency
- Other characteristics
  - Migration frequency low in all cases
  - No ping-pong observed

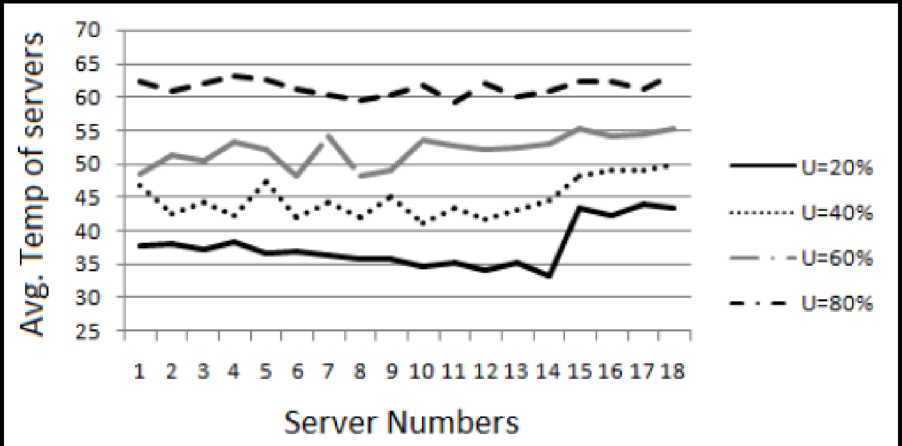




Results w/ Thermal Effects

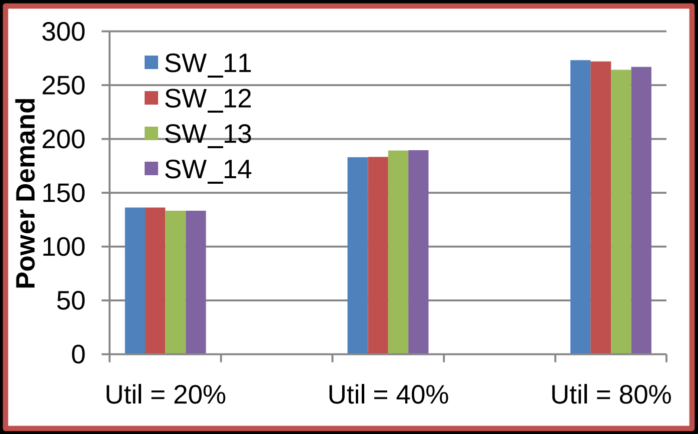
- Imbalanced cooling
  - Servers 1-14: T_a=25^o C, Servers 15-18: T_a=40^{o}C
  - Temperature limit: 65^{o}C
- Power demand is adjusted by the alg. to account for higher temperature

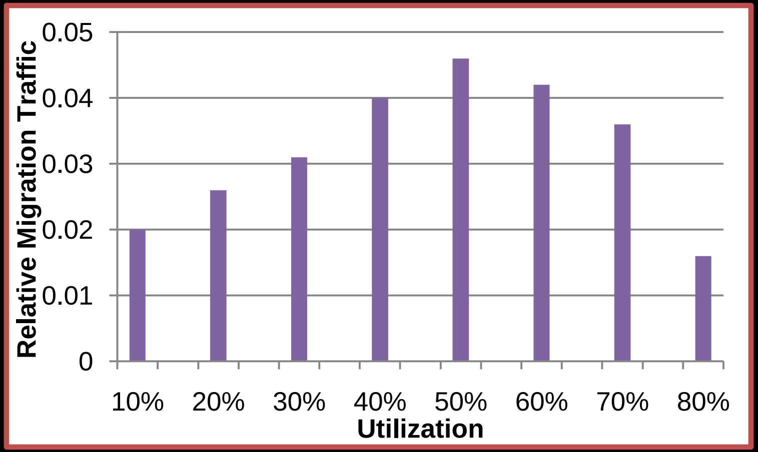




Results for Switch Power

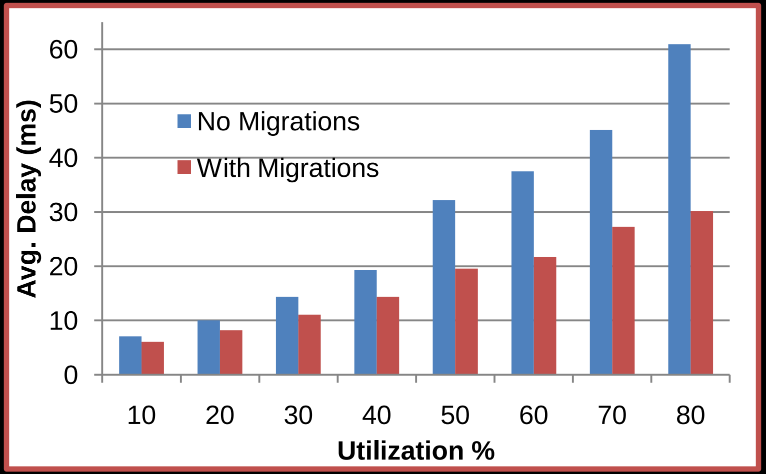
- Local migration also limits network traffic across multiple switch hops.
- Power budget allocated to switch and considered in the migration.

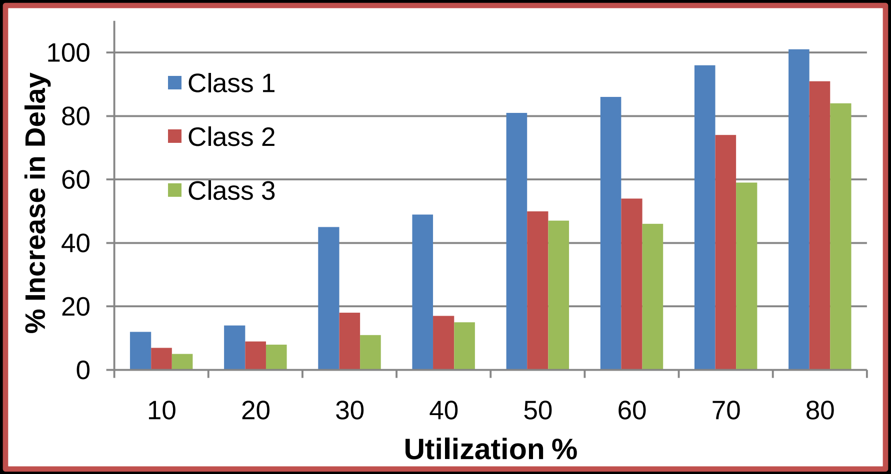




Results with QoS

- 3 classes of apps, w/ priority treatment
  - Class 1 most important, class 3 least
  - Under energy constraints, drop class 3 first, and then class 2
  - Although delay increases w/ util, migrations protect higher priority classes.





Mandatory Sleep

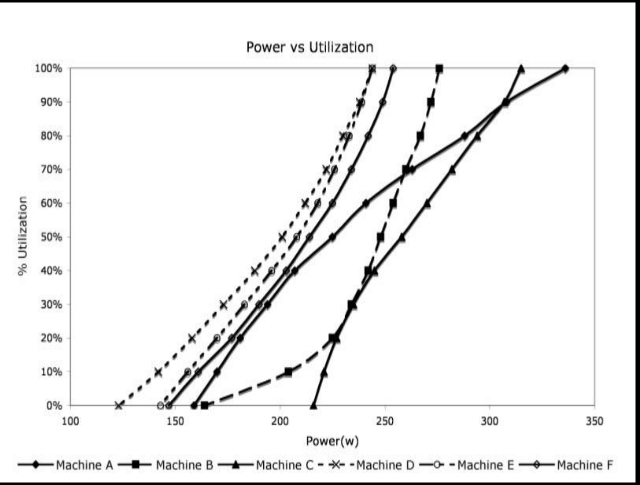
- Blink architecture [ASPLOS'11]
  – Define a duty cycle for each server
  – Adjust sleep durations based on current power availability.
  – Proactive workload mgmt to deal with sleep
    - Migrate tasks away before the sleep begins.
    - Migrate tasks in just in time for wakeup
- Characteristics
  – Another form of energy adaptive computing
  – Mandatory sleep for all servers, instead of keeping some servers down ➔ More overhead

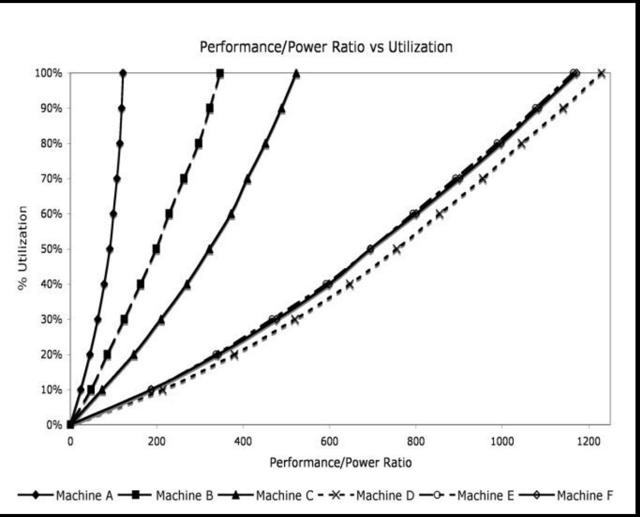
Power States and Management

Background: Server Power Modeling

- Power Components
  - Idle power (primarily leakage power)
  - Active power (utilization dependent)
- Idle power reduction
  - Low power modes (if available)
- Active power reduction
  - Voltage (α V^2) and Frequency (α f)

- SPEC Power 2008
  - Captures Power Characteristics at different load/utilization points for entire server
  - **Static Idle Power** + Utilization based dynamic power





Background: Storage Power Modeling

Disk Spindle Power (60-80%)

$$P_{spindle} \alpha \omega^2$$

+

Disk Head Assembly Power (10-30%)

(Access Pattern)

+

Disk Buffer/electronics Power (5-10%)

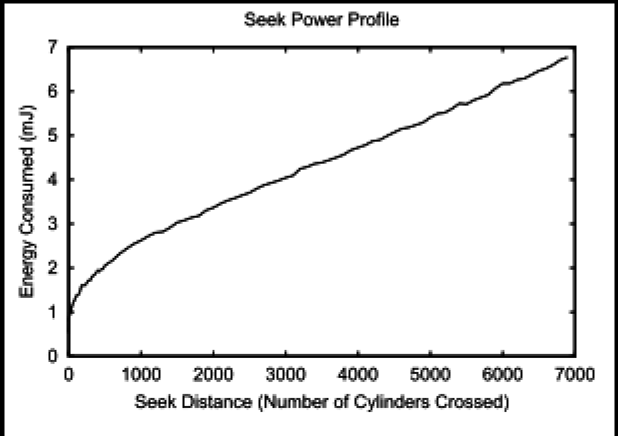
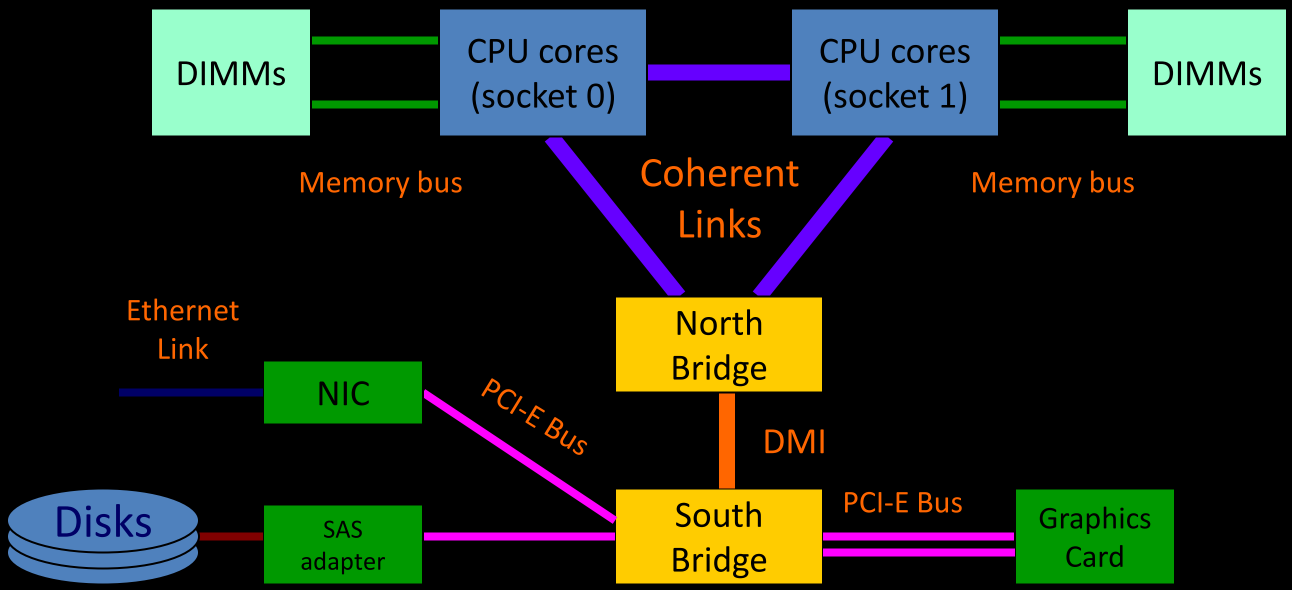


Figure 3: Seek-Power Profile for the IBM Microdrive.

Typical Models

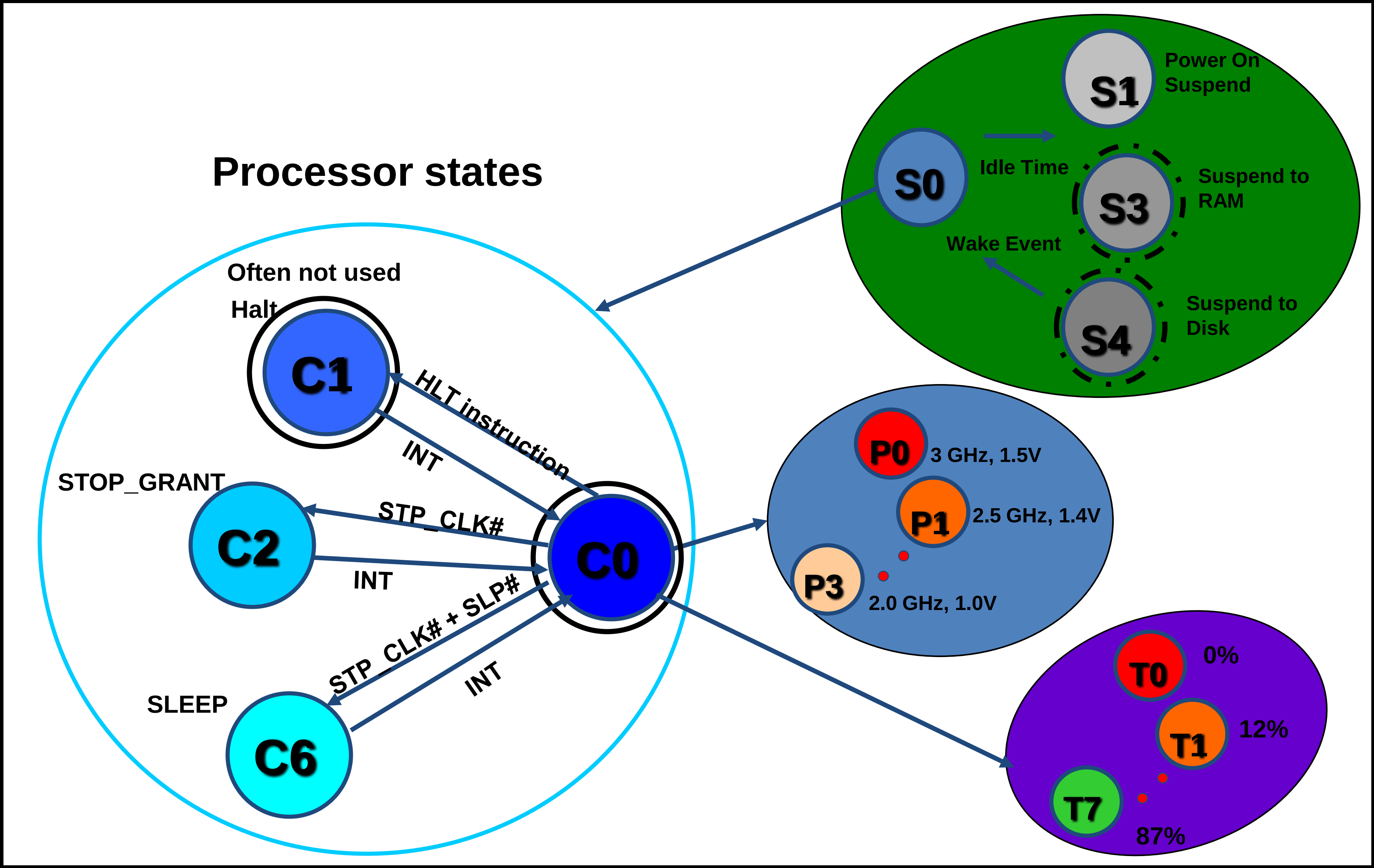
- **Static/Idle** Power + Utilization/Access Pattern based dynamic Power

System Architecture



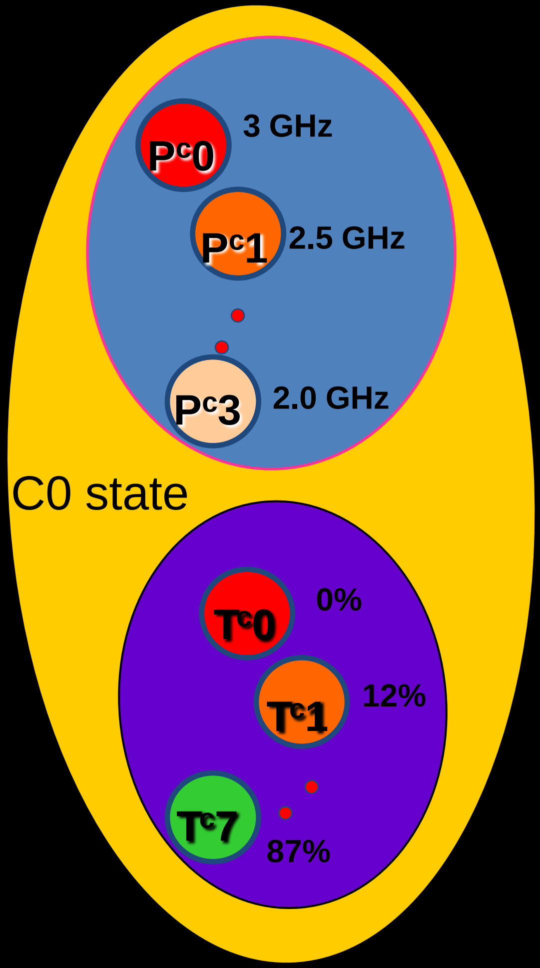
- Need effective power control of all components in a coordinated fashion

System & CPU Power States



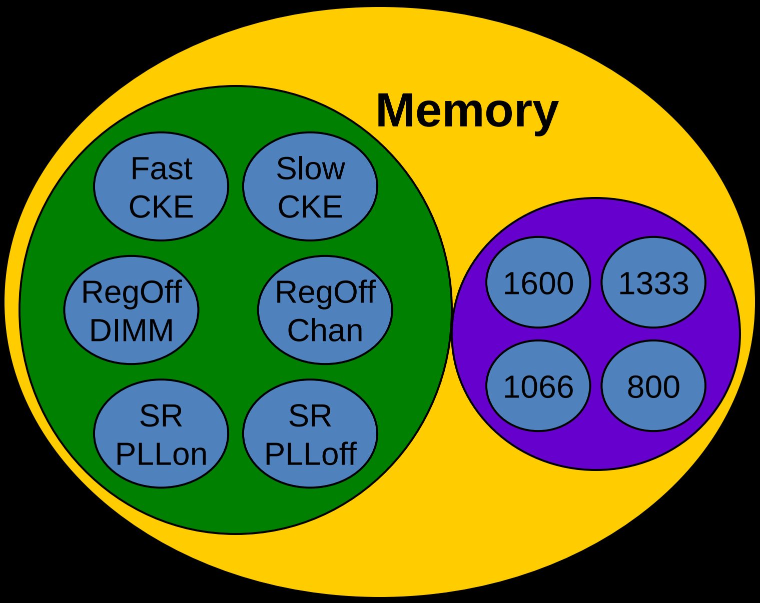
More States …

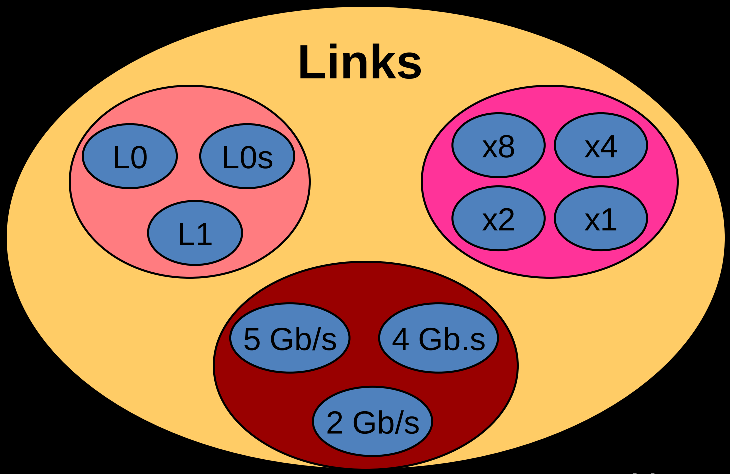
- Multi-core CPUs
  - Core-specific C states (C^c).
  - Core specific P^c and T^c states.
- Relationship between CPU states and core states
  - Core transition to low power OS controlled (e.g., MWAIT instruction)
  - CPU in state C_x iff All cores in state C_x or higher?
  - Cores may be limited in P states.



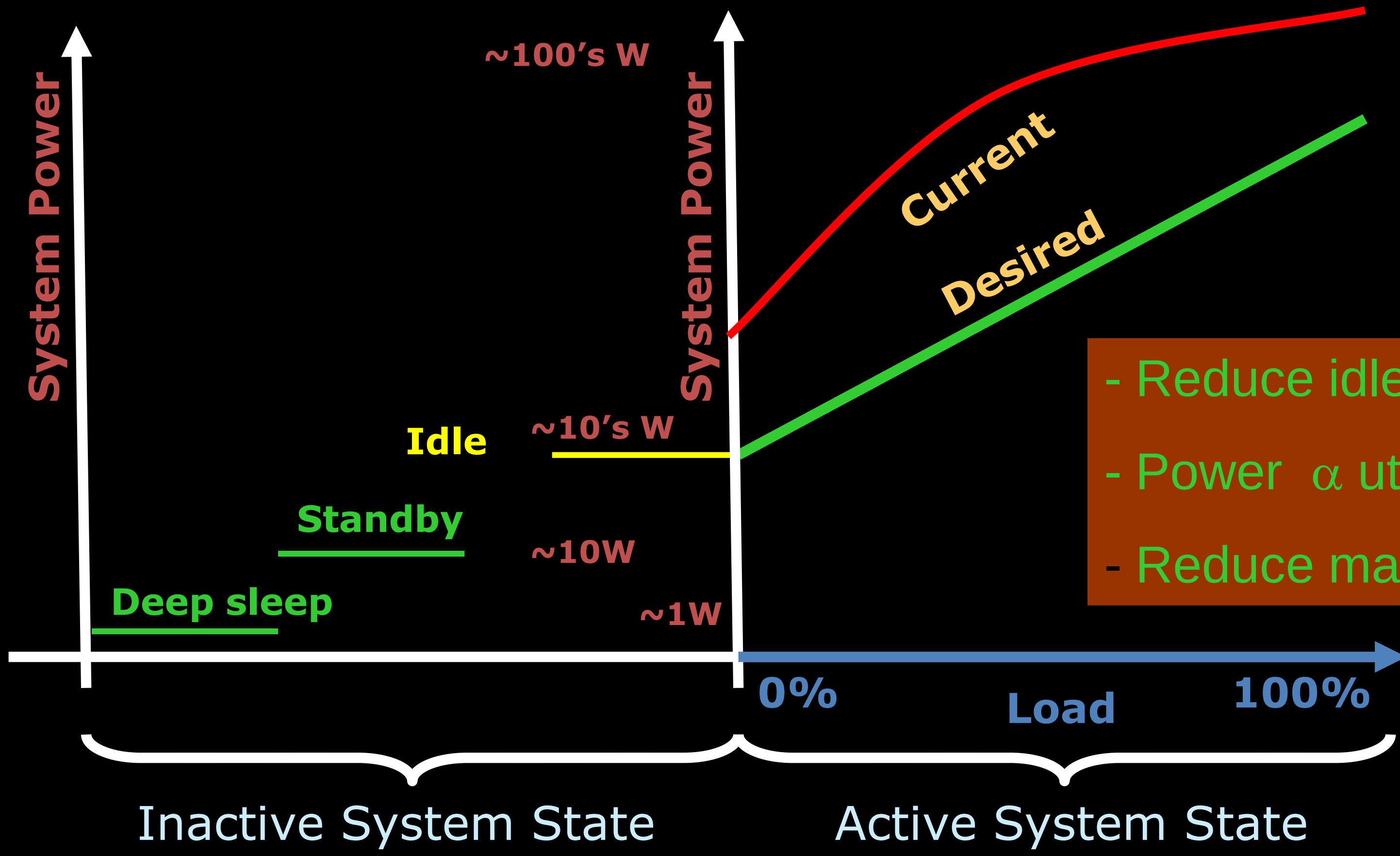
Even More States …

- Memory
  - Multiple frequencies
    - Per channel?
  - Range of idle states
- Links (PCIE, ENet, …)
  - Signaling rate
  - Sleep states
  - Widths (for serial)
- Storage
  - RPMs (for disks)
  - Ready vs. spun-down





What do we want?



- Reduce idle power
- Power α utilization
- Reduce max power

What do we want?

- Power ≤ Idle + Slope x U   (U = 0..1)
- Use available active/idle power states to
  - Minimize Idle power and Slope
  - Subject to Perf_loss (U) < threshold L
  - L may be a function of U
- Turn it around
  - What power states do we need?
  - How do we handle the Cartesian product problem?

Power Management Methods

Isolated Power Management

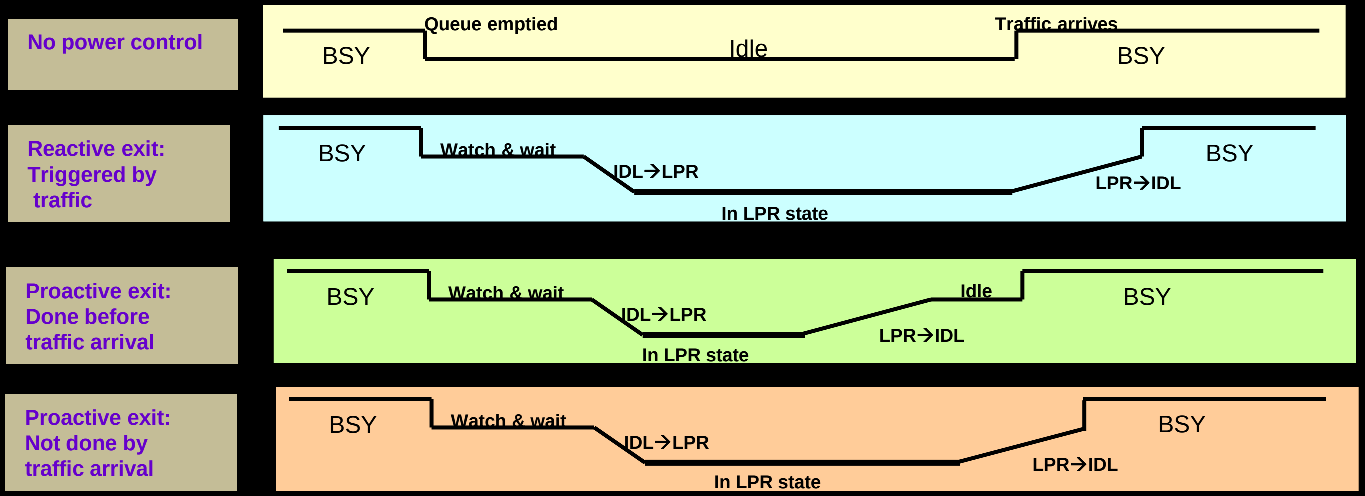
- Three major controls
  - Active states
    - Frequency, voltage, etc. (cpu, mem, link)
  - Inactive states:
    - C, core-C, CKE, L0s, …
  - Width control
    - Bit-serial links (all links going bit-serial)
    - #active CPU cores (others in deep sleep)
    - #active memory ranks
- These controls may be applied together

Active State Control

- Major Issues
  - Voltage levels approaching limits ($P \alpha V^2$)
  - Frequency change ($P \alpha f$)
    - PLL re-synchronization (latency!)
    - Very difficult for individual memory ranks
    - Very slow for links (needs handshake)
  - T state control: Can be a performance killer
  - Race to sleep vs. walk
    - Running slower is not always better

Inactive State Control

- Entry into inactive state
  - Triggered by idled resource -- involuntary sleep
  - Preplanned (move away workload before sleeping)
  - Forced by energy availability – involuntary sleep
- Exit from inactive state
  - Reactive (driven by traffic arrival or energy availability)
  - Proactive (Based on prediction/planning)
    - Prediction accuracy is crucial



Width Control

- Enable only a subset of identical instances
  - Most frequent use – multi-lane bit serial links
  - E.g., 40 Gb/s – 4 lanes @ 10 Gb/s (Gen 3) technology
  - Other instances: #cpu cores, #copies of resources.
- Why Width Control?
  - Power proportional to number of active instances.
  - Can allow for larger transition latencies.
- Width Control Issues
  - Only certain widths may be allowed, e.g., x1, x2, x4
  - Width increase/decrease -- gradual or drastic?

Granularity of Power Mgmt

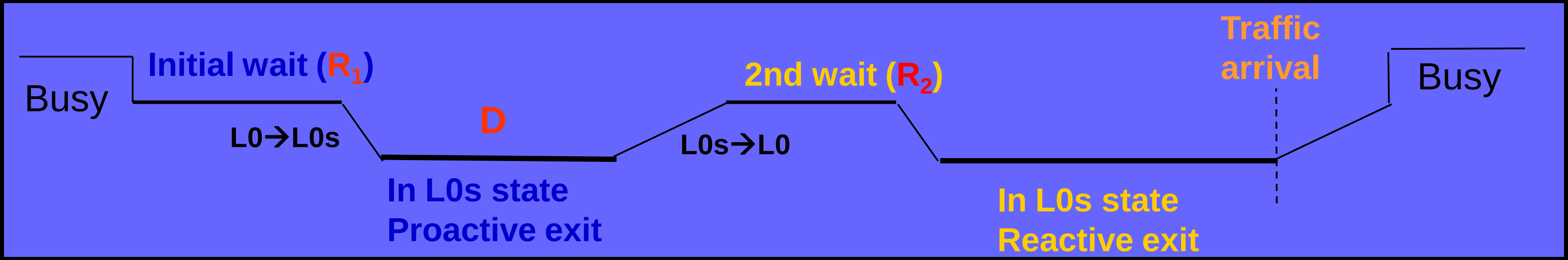
- Coarse: Low util. over ~10 mins ➔
  - Workload consolidation to change traffic paths
  - Shutoff of unneeded switches, interfaces, …
- Medium: Low util. over ~10 sec ➔
  - "Slow Controls", e.g., speed change
  - Dynamic consolidation of ports, e.g., shadow port
- Fine: Low util. over μs to sec
  - Lot of opportunities to save power, but
  - Solutions must be simple & HW implementable

Speed/Frequency Control

- Generally utilization driven
  - Change frequency to keep utilization close to a target (e.g., 80%).
- Lots of techniques for CPU's
  - Increase to max freq, decrease in steps (speed-step)
  - Others (including those based on perf counters)
- Issues
  - Need to be combined with others (e.g., T & C state control for CPUs)
  - Memory & links: Only coarse granularity control feasible.

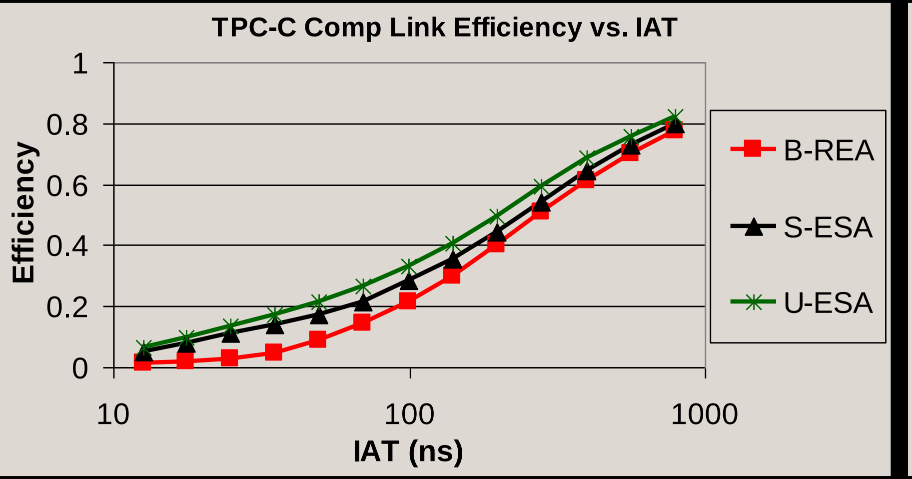
ESA: A Hardware Algorithm

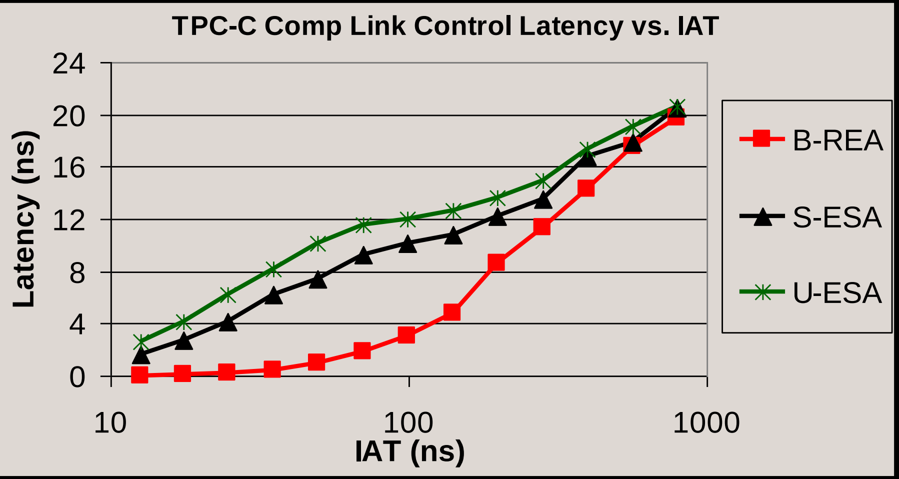
- Characteristics
  - A two phase algorithm w/ proactive & reactive exits
  - Proactive duration (D)
    - Uses biased exponential smoothing
    - Bias makes the algorithm more sensitive to gap decrease.
  - Very easy to implement at high speeds: (~4000 gates w/o stats)
- Can work as a combined algorithm
  - Measure R_2 starting from beginning
  - Small R_2 → Reactive only; Large R_2 → Proactive only



Reactive vs. Proactive Perforamance

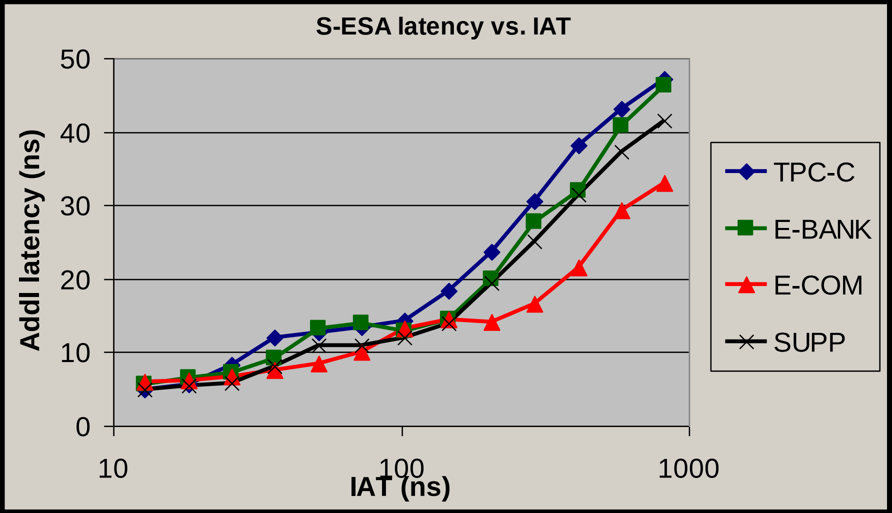
- Showing 3 algorithms
  - B-REA – basic reactive
  - S-ESA (Simple ESA) – Bang bang control of runway
  - U-ESA (Utilization based ESA) – Runway duration a Resource utilization
- Observations
  - Proactive: Higher efficiency but higher latency.
  - Simple algorithm works almost as well as the complex one

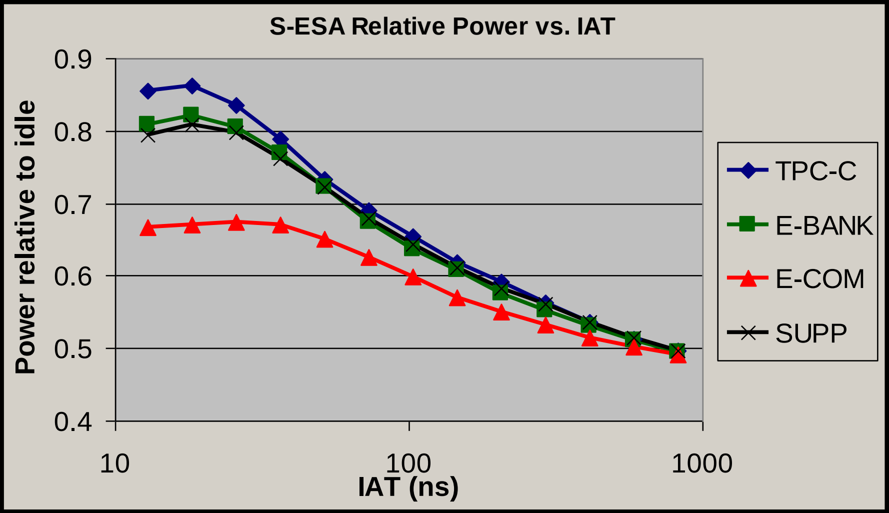




Effect of Workload

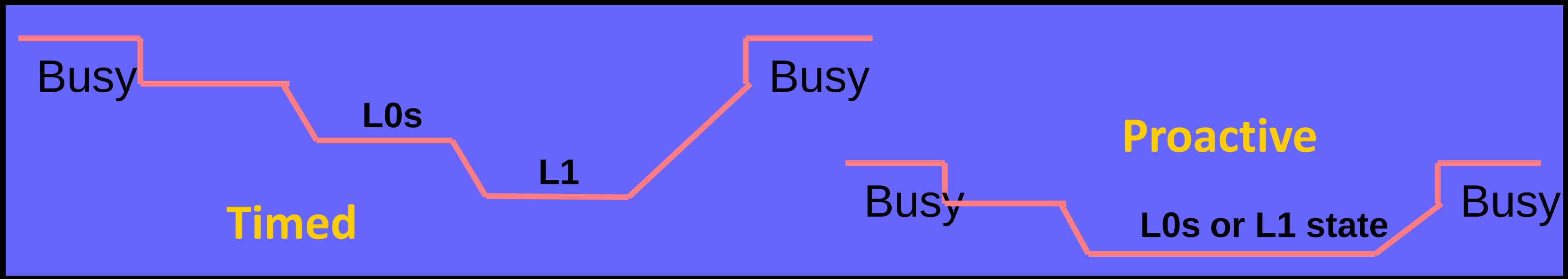
- Better predictability ➔ Higher efficiency & Lower latency





Multi-State Control

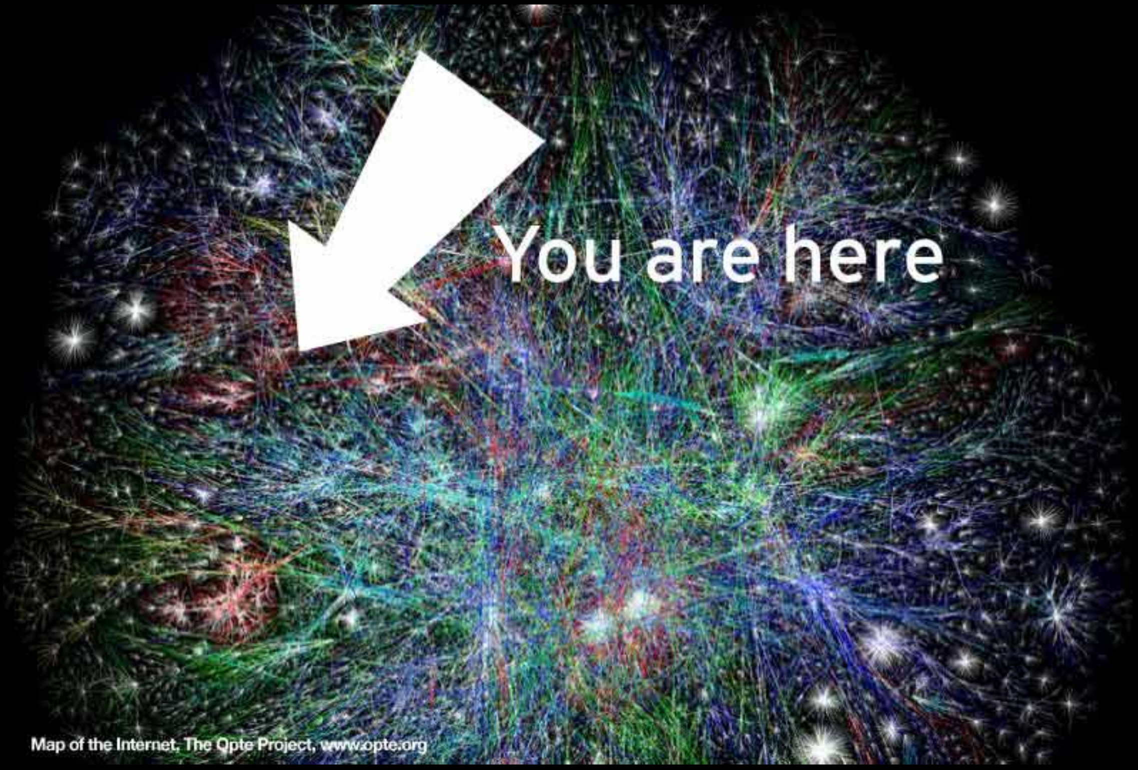
- Progressively lower-power & but slower transition states.
- Two basic methods
  - Timed promotion to deeper state
  - Proactive selection of sleep state based on recent activity
    - Timed promotion is still required
  - Proactive demotion possible, but usually not sensible
- Complications
  - Usually transitions via active state – frequent switch a bad idea!
  - May have minimum residence requirements



Width Control Algorithm

- Down-shift – At beginning of gap
  - No change in progress & $W > W_{min}$
  - Recent link utilization < Thres1
- Up-shift -- At end of every pkt
  - No change in progress & $W < W_{max}$
  - Current $QL > Q_{HT} \times W$, or
  - $QL > Q_{LT} \times W$ & recent link utilization > Thres2
- Notes:
  - Link util estimate: from busy periods & gaps
  - Thres1 & Thres2 related to provide hysteresis

Network Power Management



Network Energy Consumption

- Increasing network power consumption
  - Storage networks, e.g., SAN switches & links (mostly FC)
  - Large numbers of Ethernet switches in DCs (& homes, offices, …)
  - Numerous links inside the server
- Substantial power waste
  - Rapidly increasing data rates (e.g., 10 Gb/s) ➔ High power consumption
  - But, average utilization rapidly decreasing
    - Upgrades driven by latency & peak BW needs, not avg BW.
  - Large data centers may have 1000s of fabric ports

Power Consumption of Ethernet Switch

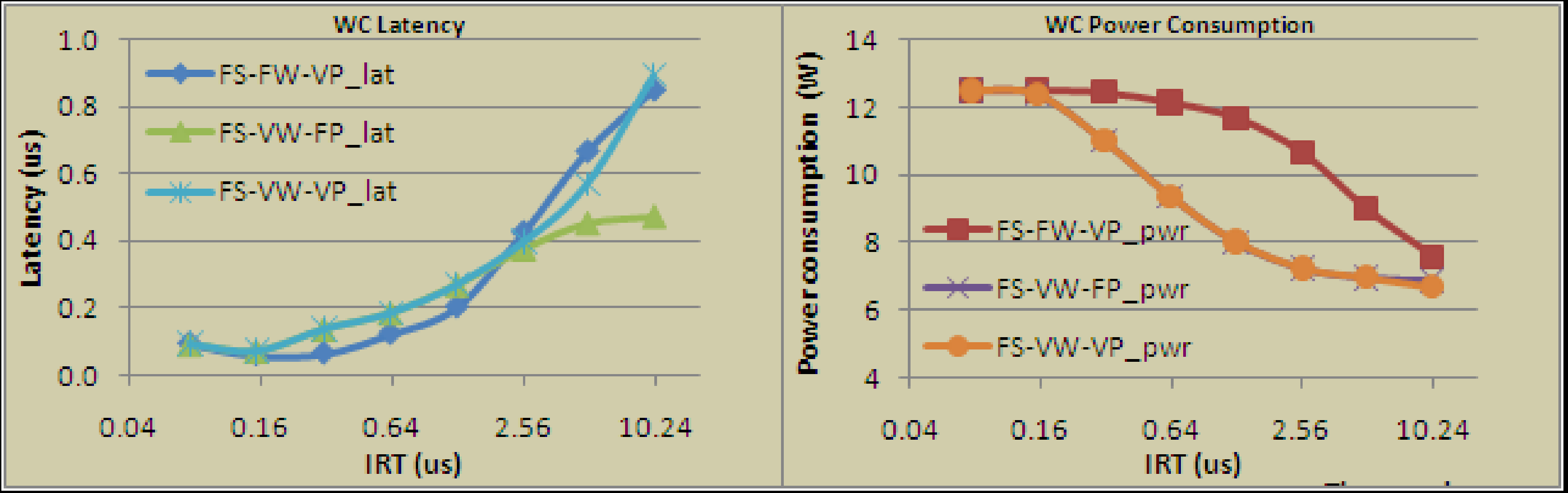
Parameter	Value
Power fixed	60W
Power Fabric	315W
Power Line Card (first card)	315W
Power Line Card (subsequent card)	49W
Power Port	3W
Power Port Idle	0.1W
Port Transition Power	2W
Port Transition Time	1-10 ms

Network Energy Management

- Fine grain
  - Use link low power modes: speed control, width control, power state control
- Coarse grain
  - Shadow ports – collects traffic while the associated link is unavailable
  - Coordinated end-to-end power state management.
- Semi-static
  - Periodically redirect flows to allow certain ports/switches to stay in low power mode.
  - Intelligent data placement (and dynamic reshuffling) to minimize active ports.

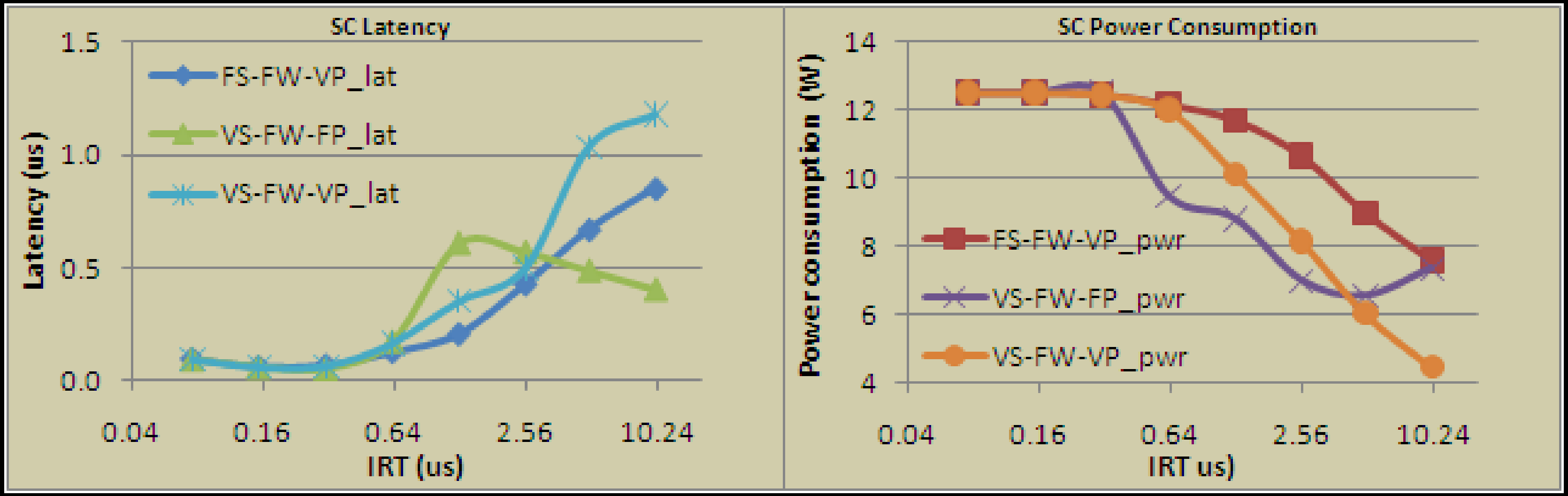
Width vs. State Control

- Key to graphs: [F/V] [S | W | P]
- Similar latency but much higher power savings.
- Power state Control helps width control marginally at very low utilizations



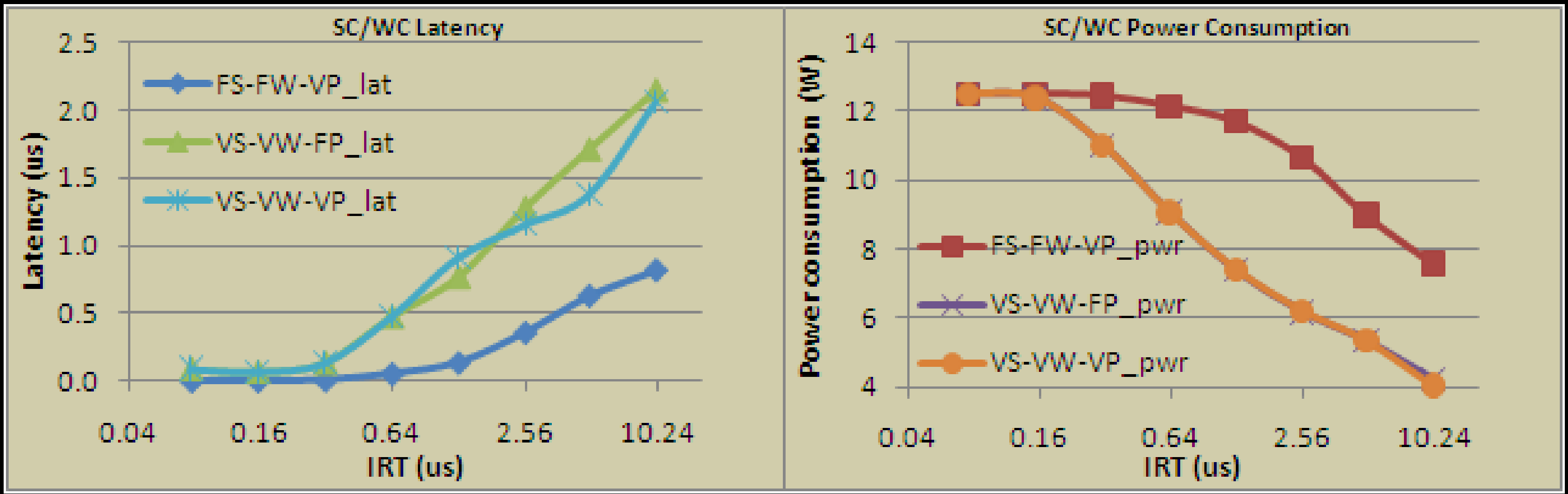
Speed and State Control

- Power state control better than speed change control.
  - Depends on low entry/exit latencies & idle power
- Speed control has erratic behavior because of large transition latencies
- Combination can yield provide even more savings



Speed and Width Controls

- Width Control effect dominates.
- No real advantage of adding speed control
  - Running the link slower only extends busy periods and hurts power management

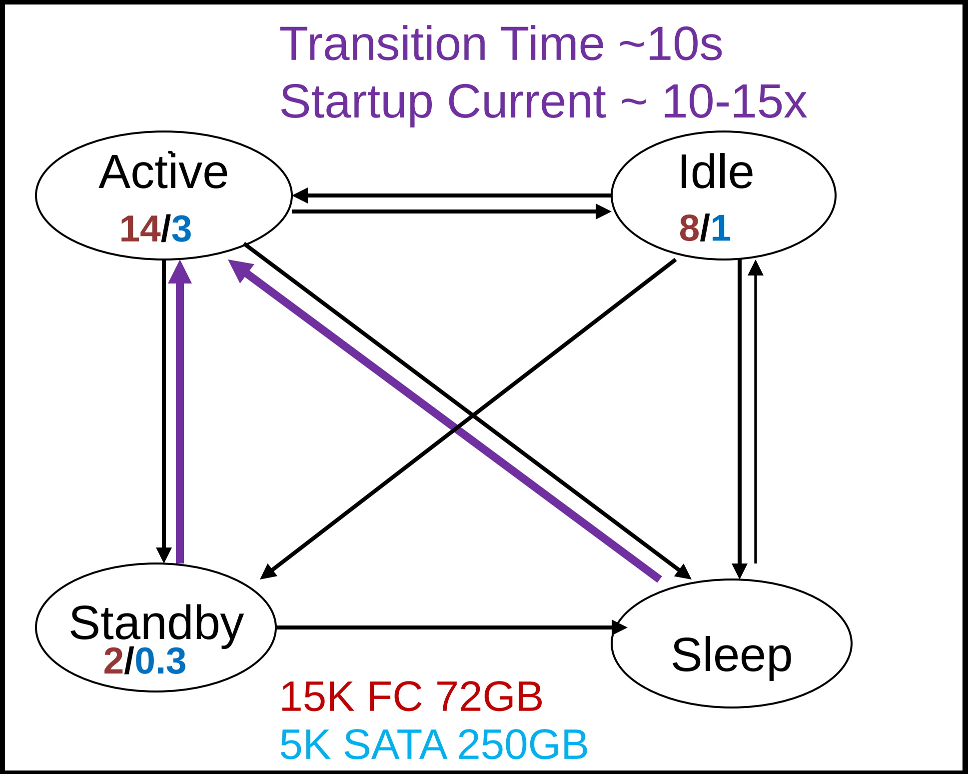


Storage Power Management

Storage Power Consumption

- Storage demands growing 60%/yr due to
  - Growth in content richness of data
  - Compliance issues requiring stricter retention policies
- Archival & Nearline storage footprints growing faster
  - Outpacing online storage footprint
  - Could potentially overtake server power consumption with increased use of disks instead of tapes
- Data access rate increase << Data volume increase
  - Potential for energy efficient storage systems.
  - Reliability an important component for energy efficient systems.

Disk States & Power Usage



- Active: Spindle, Head &Buffer On
- Idle: Spindle, Head &Buffer On
- Standby: Spindle &Head Off, Buffer On
- Sleep: Spindle, Head &Buffer Off

- Spindle Motor (60 – 80%)

$$P_{spindle} \alpha \omega^2$$

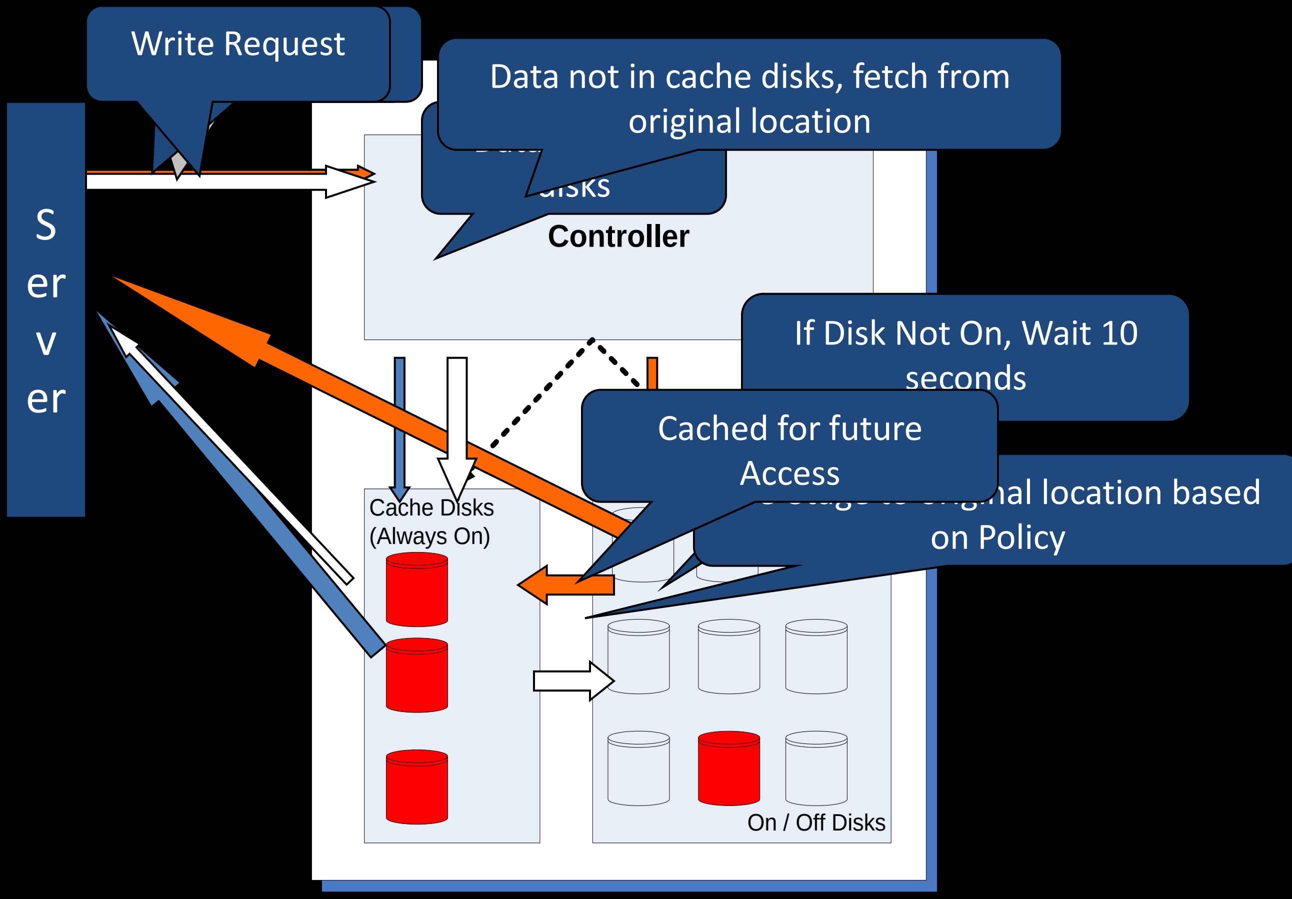
- Head Assembly (10-30%)
- Buffers/Electronics(5-10%)

Typical Specs
(15K enterprise drives)
- Idle Mode: 8-10W
- Active: 12-14W
- Standby: 2W

Storage Power Mgmt Approaches

	Pros	Cons
MAID[ICS02]	Passive disks –saving power	Two-group
PDC[ICS04]	Multi-Group	No redundancy
DIV[Sigmetrics06] (Diverted Access)	Multi-Group, for WAN storage,	No flash, Only Redundant disks off, no cache
GreenStor[MSST07]	app hint, cache disk	Reliability, No Flash
Pergamum[FAST08]	Reliable, using NVRAM	No data migration, Not SSD
New Design	using SSD, High-speed Disk, automatic way, app hint, performance, reliable, saving power	Cost? Write?

Background: Massive Array of Idle Disks (MAID)



Background: MAID Characteristics

- Majority of Disks are turned Off
  - 5-25% of the disks are used as Cache Disks (always On),
  - Remaining disks are turned-on on cache miss
- Significant power savings in large disk farms
  - No need for any hardware/engineering change to disk drives
  - Temporal Locality based caching
  - Performance highly dependent on #cache drives
- Average Worst Case response time quite large

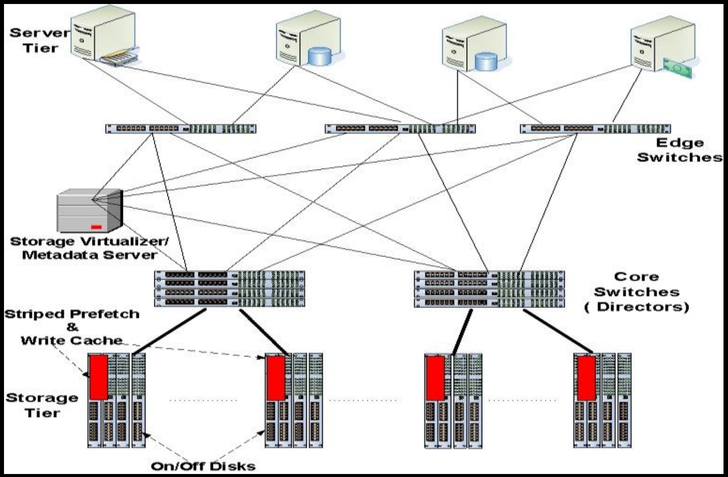
Copan Systems MAID Commercial Implementation

Performance State	Drive Performance	Power On/ Mount Time	Total Response Time
Case 1 MAID - Request is on a powered on, spinning drive	Average device service time - typically 20-40ms.	None	~20-40ms
Case 2 MAID – Request is on a powered off drive	Average device service time - typically 20-40ms.	10 seconds	~14-15 seconds
Case 3 ATL – Request for data in tape library and a tape drive is available	Average access time- typically 40 seconds	10 seconds average mount time	~50 seconds
Case 4 ATL – request for data in tape library and no tape drives are available	Average access time - typically 40 seconds	10 seconds average mount time	~50 seconds plus wait time for tape drive availability

Note: Stated access times are from vendor's published specifications.

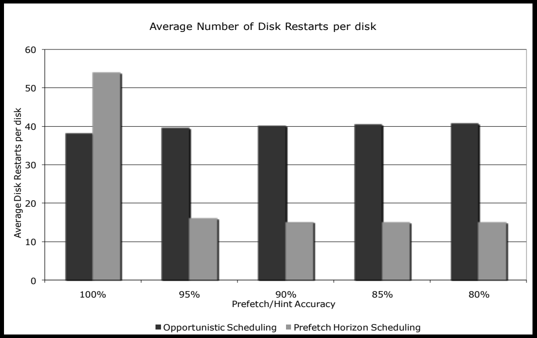
GreenStor

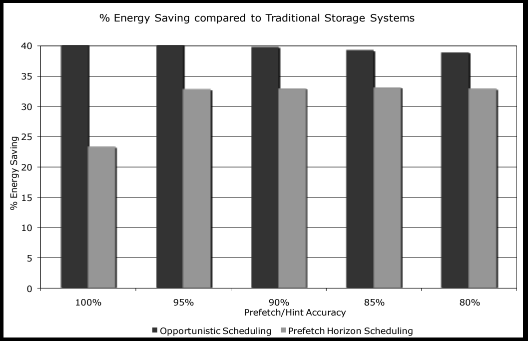
- Distributed Virtualized Read-Prefetch / Write cache
  – Minimize Cache hotspots
  – Maximize Data Hotspots (Facilitate longer idle periods)
- Opportunistic prefetch
  – System monitoring information combined with current system state is used for predicting expected state
- Scheduling
  – Maintain deadline based fairness
  – Scheduling for Power Optimality
- Maximize batch execution at the disk

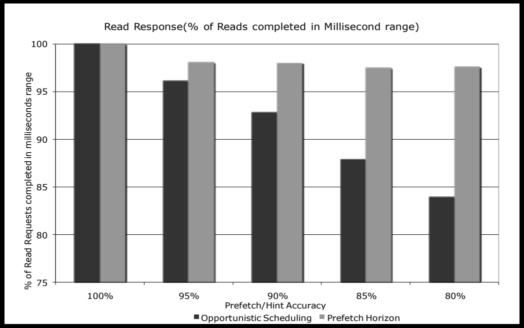


GreenStor Performance

- Performance
  - Opportunistic scheduling consistently outperforms prefetch horizon (wait until absolutely necessary) based schemes
  - Saving close to 40% even with decrease in prediction accuracy
  - Disk Restart penalties have a larger impact on Opportunistic scheduling -- more restarts (as a result of lazy batch behavior)
- Read Response Time
  - Relatively better in case of Prefetch Horizon when prediction accuracy is low
  - (Disks are more likely to be On with prefetch horizon)



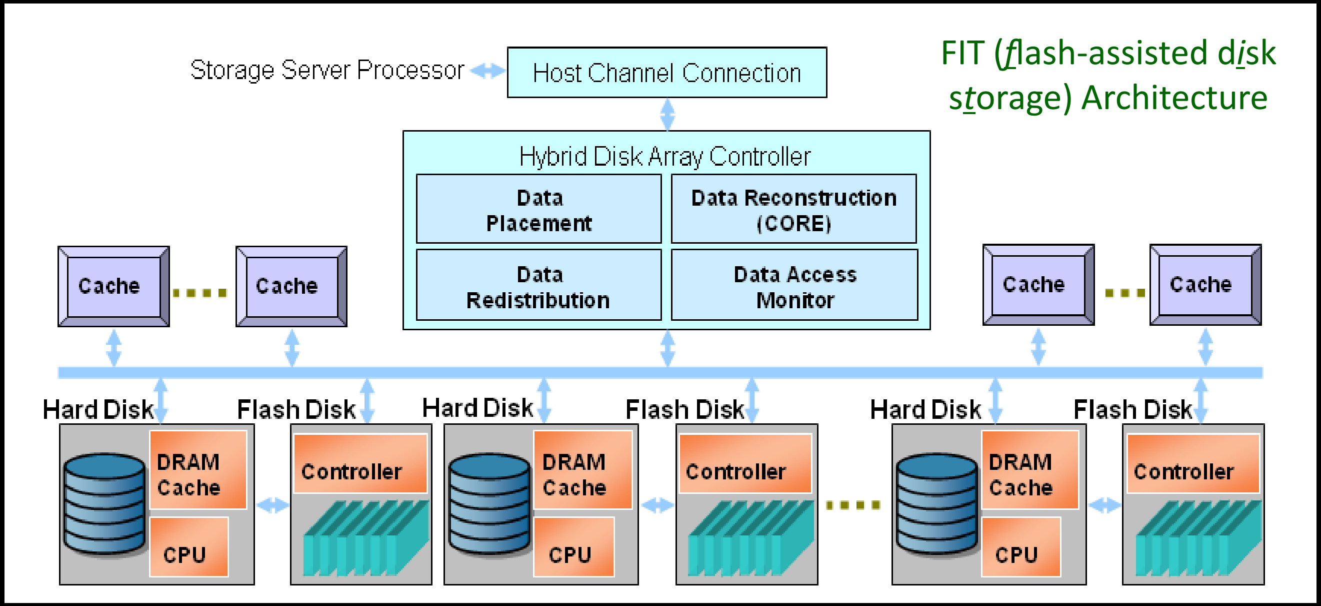




Solid State Drives (SSD)

- Much more energy efficient. Useful as a cache in storage hierarchy for active data

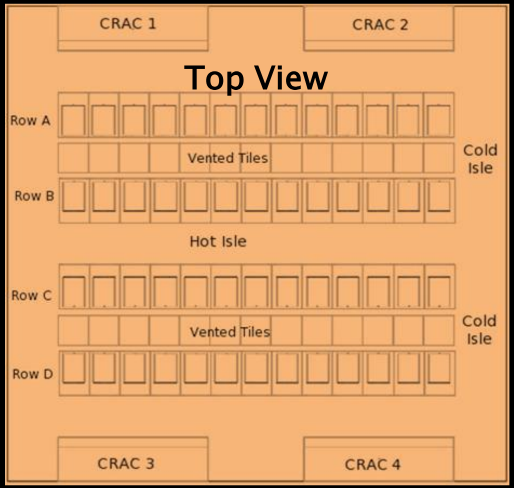
Technology	Power cons.	mW/GB
DRAM (1 GB DIMM)	5W	5000
15K RPM 300 GB HD	17.2 W	57.33
7.2K RPM 750 GB HD	12.6W	16.8
128 GB SSD	2.0W	15.6

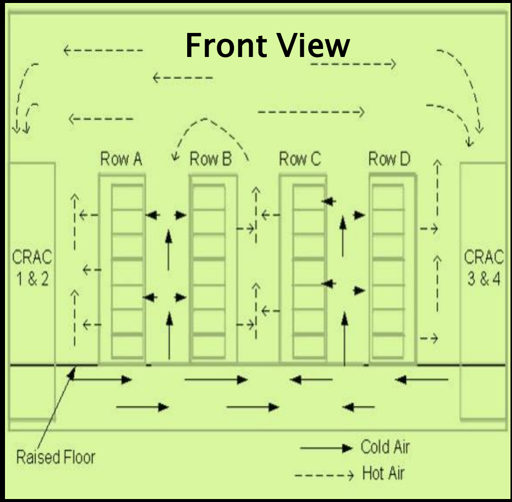


Data Center Cooling

Typical Data Center

- Fans suck in Cold Air from the vents at front of servers (inlets)
- Keep Inlet temp. below 25^0 C for safe operation (Thermal Redlining)
- Efficient Cooling
  - Q: Heat generated is a function of System Load $= (T_{outlet} – T_{inlet})/C_p\ f_\rho$
  - W: Work done in removing/extracting Q units of heat
  - COP (Coeff of perf.): Heat removed per unit work = Q/W

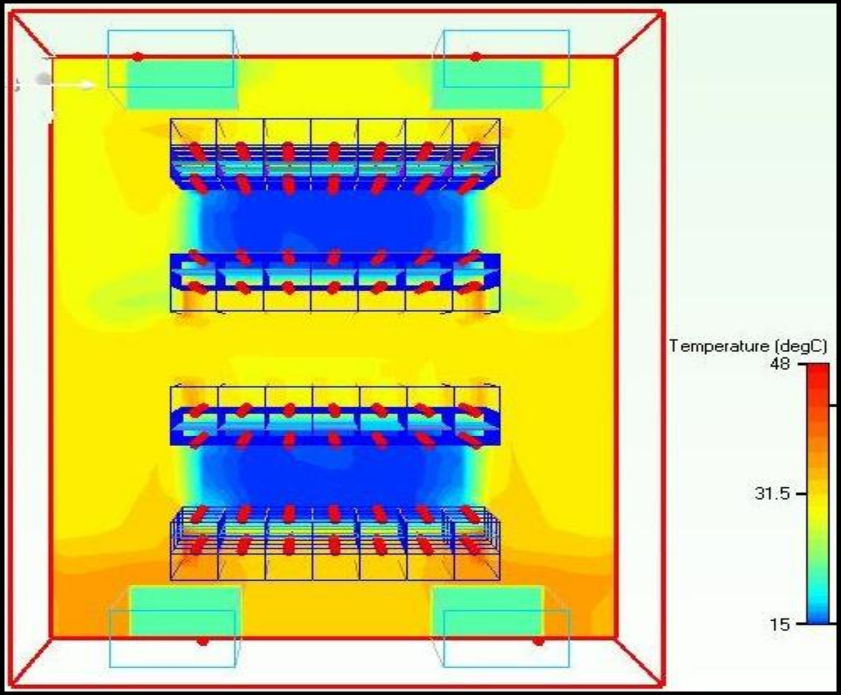




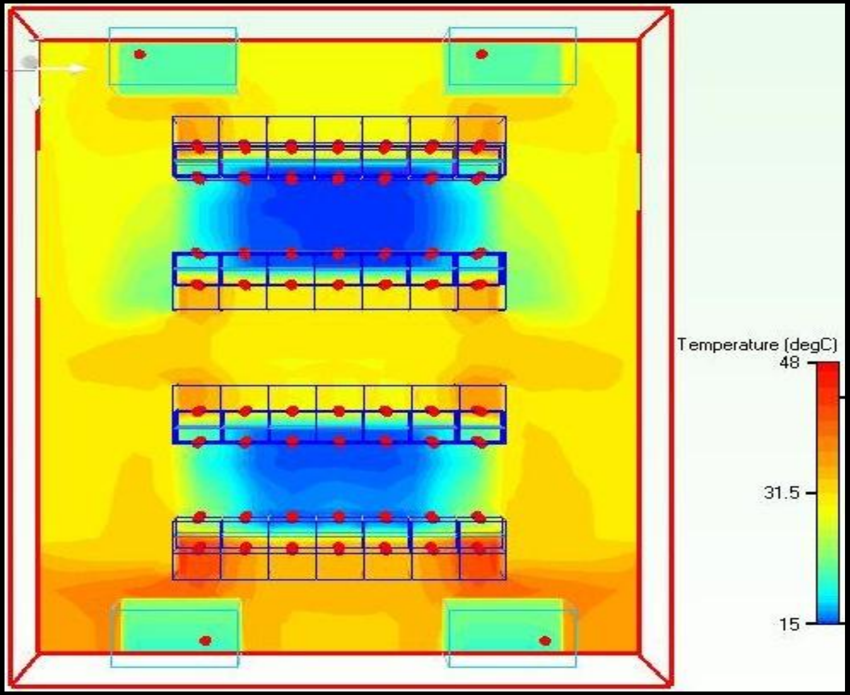
Inefficiency in Cooling

- Heat Recirculation or Hot gas bypass
  - Hot air does not completely reach CRAC for extraction
    - A portion recirculates into the cold isle & mixes with cold air.
  - Natural recirculation around end of isles, top of racks, & unused slots.
- Effect
  - Inlet temperature at various servers higher than the supply temperature
- Factors that affect heat recirculation
  - Data Center Layout/dimensions
  - Workload distribution

Impact of Heat Recirculation



Height:3ft

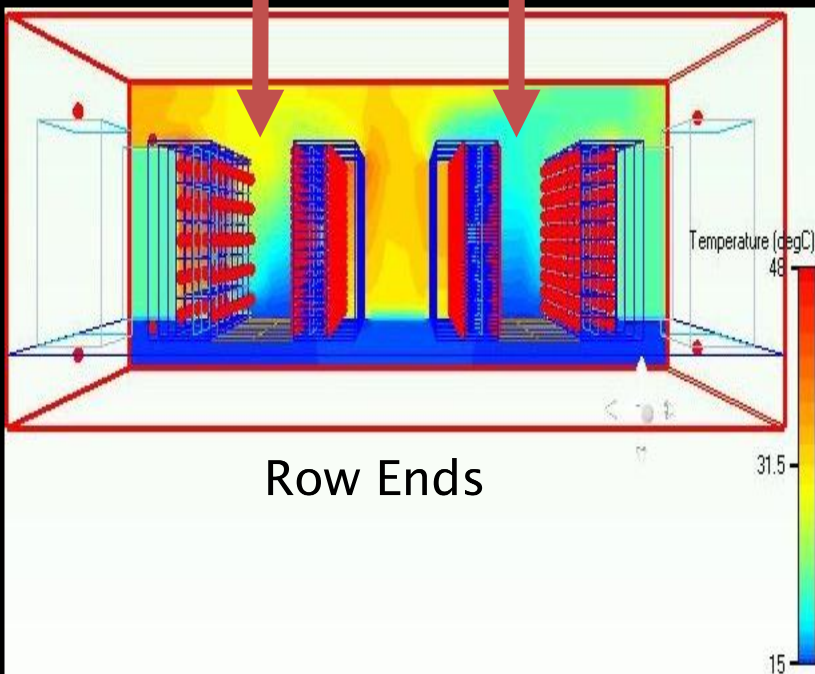


Height:6ft

- Recirculation increases with height
    - Temperatures at rack tops are higher than at rack bottom

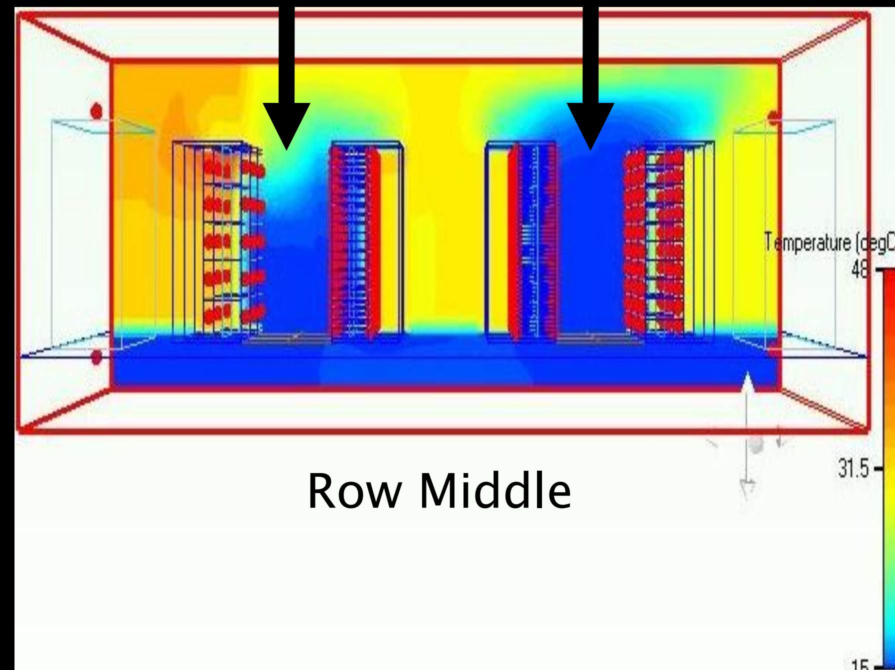
Impact of Heat Recirculation

Difference ????



Row Ends

Difference ????



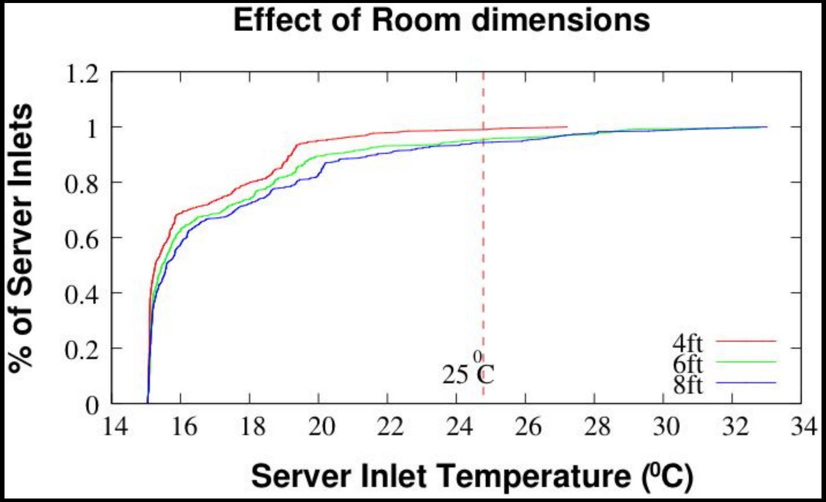
Row Middle

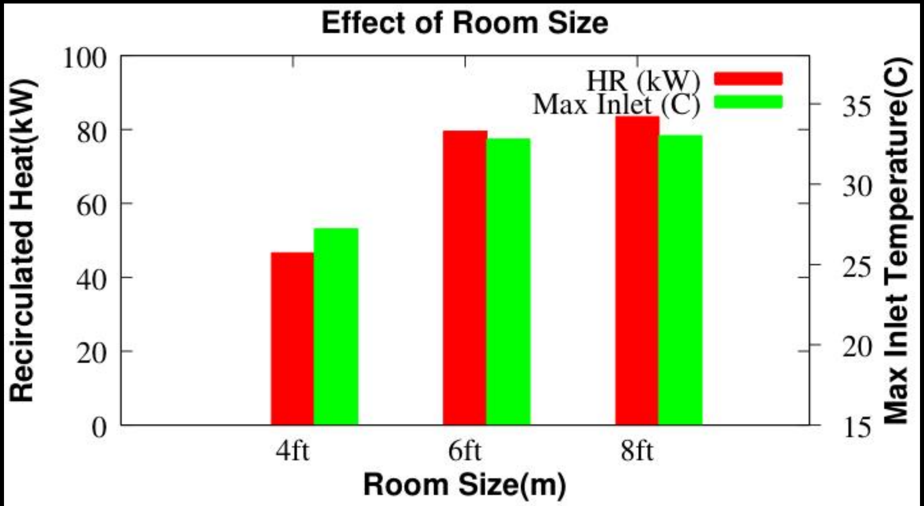
- Lesser at middle of rows/isles
- Increases towards row/isle ends

Floor Layout Planning

- Objective
  - Derive floor planning best practices using system models
  - Temperature Profile as function of
    - Data Center Dimensions (Room Size)
    - CRAC placement
    - Raised Floor Depth
    - Ceiling Height
- Constraints
  - Prevent thermal redlining
- Given
  - Thermal Characteristics of devices
  - Performance characteristics of devices

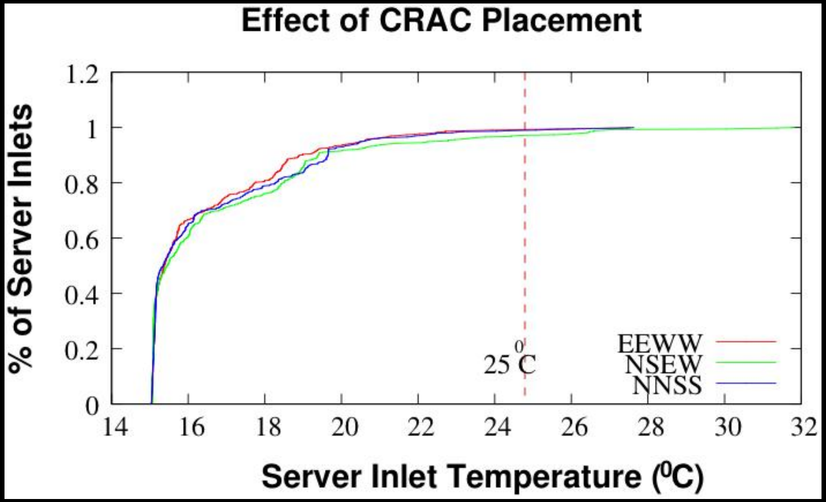
Effect of Room Size

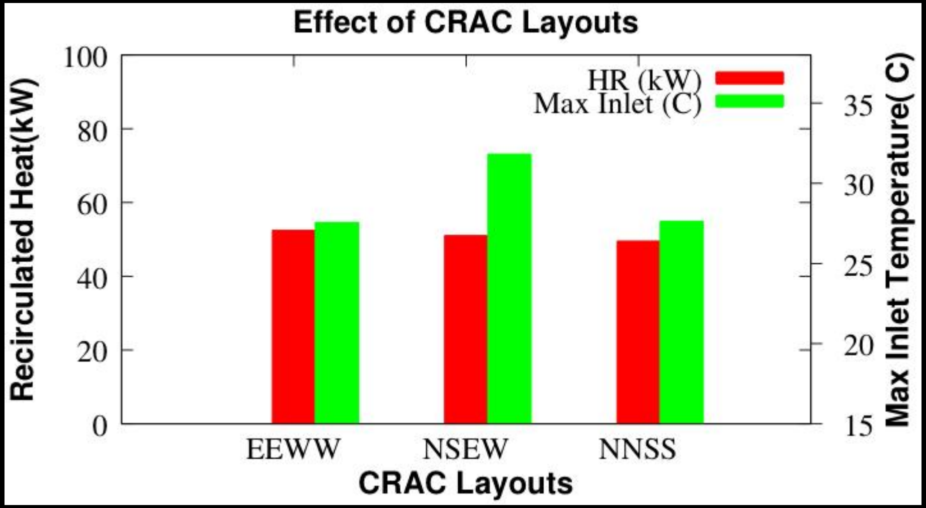




Size	4ft	6ft	8ft
# of Servers > 25 F	4	23	30

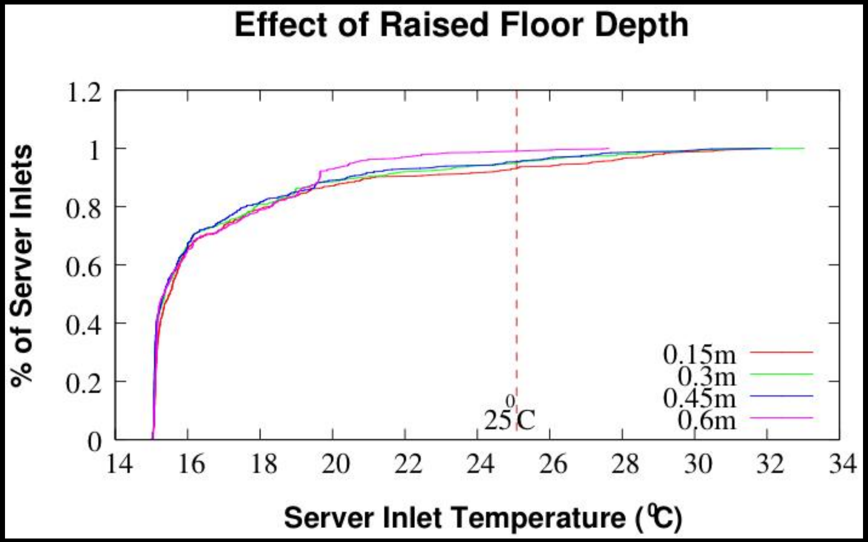
Effect of CRAC Placement

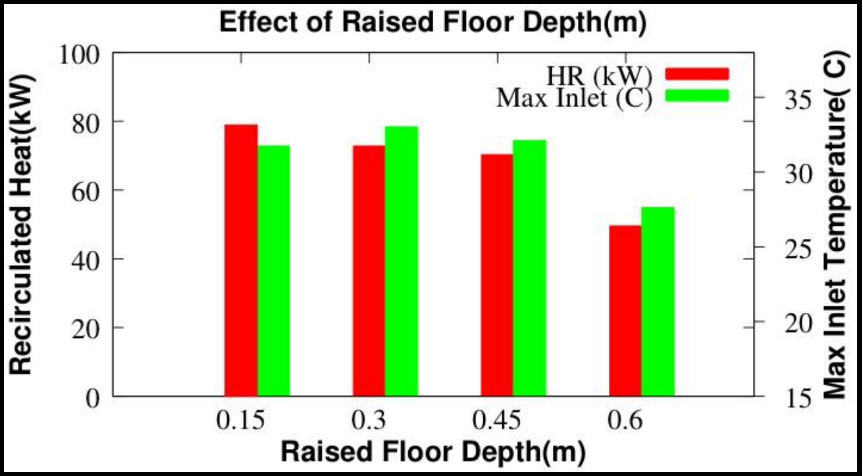




Layout	EEWW	NSEW	NNSS
# of Servers > 25 F	4	15	6

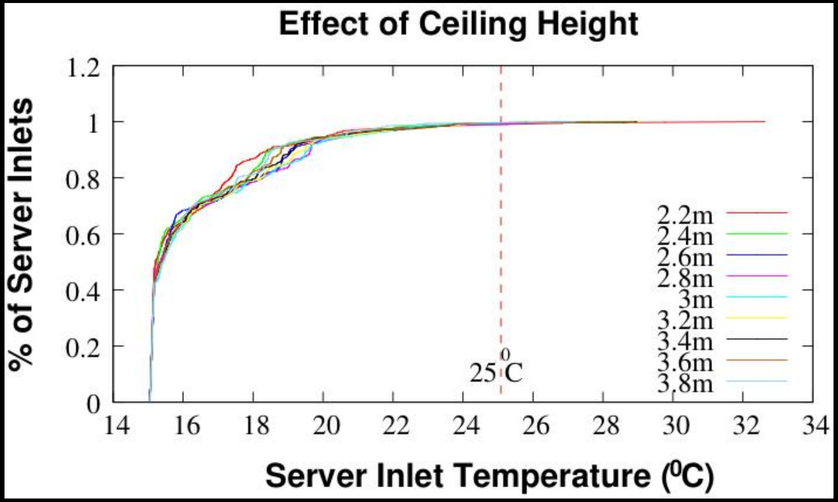
Effect of Raised Floor

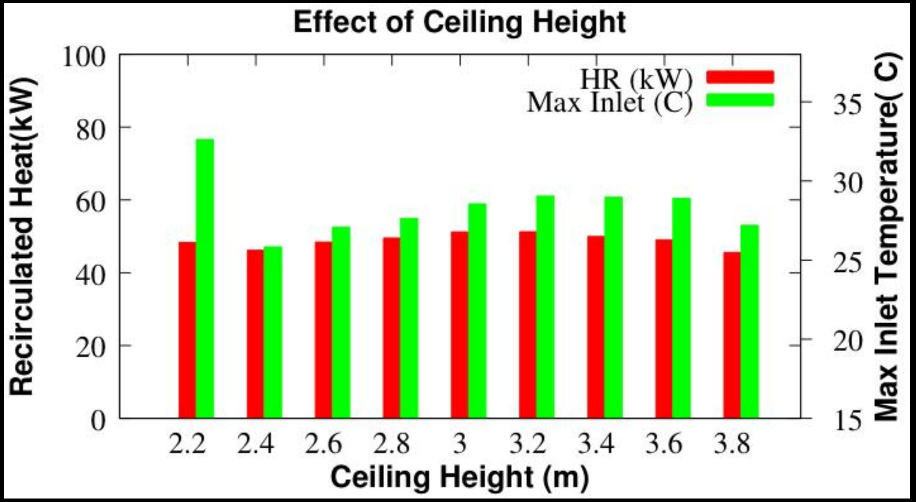




Raised Floor Depth	0.15m	0.3m	0.45m	0.6m
# of Servers > 25C	37	28	25	6

Effect of Ceiling Height





Ceiling Height	2.2	2.4	2.6	2.8	3	3.2	3.4	3.6	3.8
# of Servers >25F	6	3	4	6	4	2	2	3	2

New Data Center Designs

- Container-Based Data Center
- Google Container Based Data Center
  http://www.youtube.com/watch?v=zRwPSFpLX8I
- Microsoft built a container based data center in Chicago area for 220 containers with 1000 to 2000 server support in each container
- Goal is to reduce the area to be cooled down
- Power delivering systems within data centers
  – Making each component power efficient

Coordinated Power Management

Coordinated Power Management

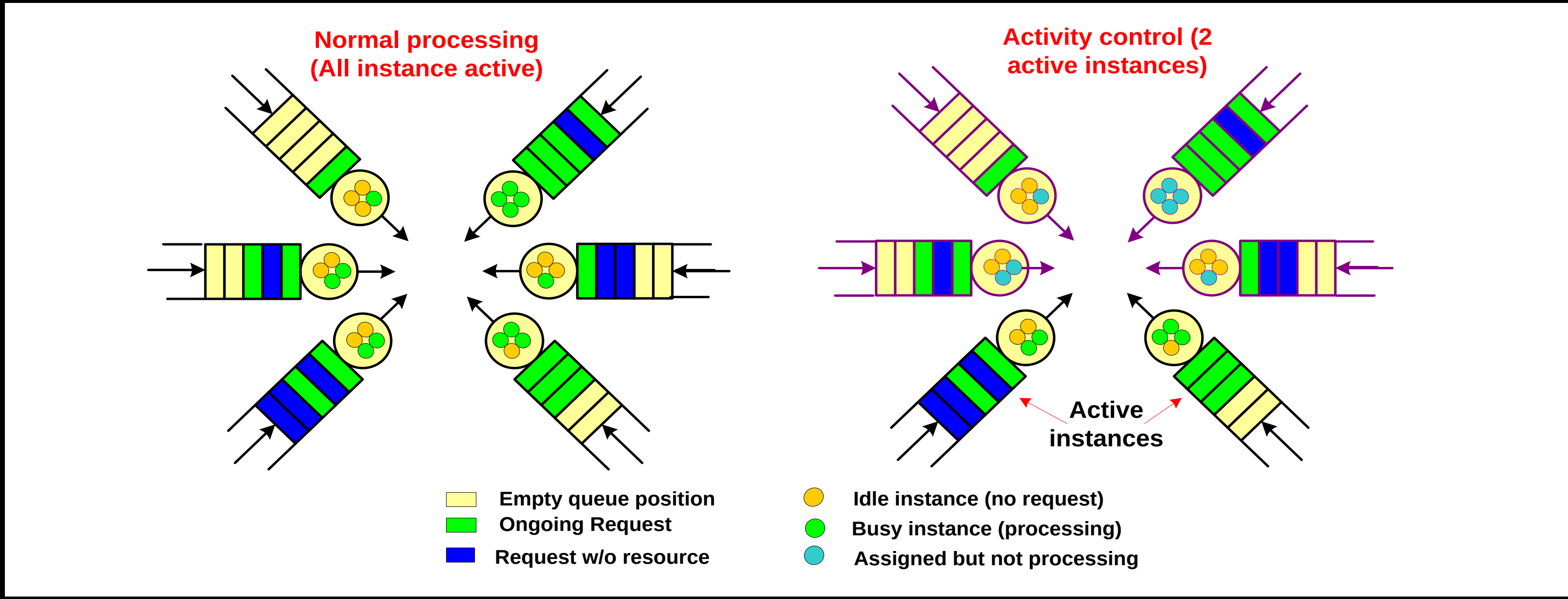
- Multiple identical instances
  - Memory ranks across a channel or socket
  - Multiple cores in a CPU or socket
- Multiple devices in a socket
  - When CPU in C6, put links in L1 & memory in SR
  - As more CPU cores go into C6, be more aggressive in placing memory ranks in CKE.
- Coordination across sockets & systems
  - Control of links based on activity in end-points
  - Shut-down & migration (well researched)
- Coordination across multiple levels
  - HW, firmware (BMC) and OS – policies and interfaces

Coordination Across Cores

- Socket level
  - When all cores in state ≥ C^c1, put socket in C1E
  - Additional opportunity to reduce voltage & freq
- System level: light sleep
  - When all cores in all sockets ≥ C^c3, put system in C3
  - Allows putting link in L1 & memory in SR
- System level: deep sleep
  - When all cores in all sockets ≥ C^c6, put system in C6
  - Further allows turning off PLLs & most of socket HW
- What are other smart control policies, e.g.,
  - Use P states in the equation?

Basic Approach

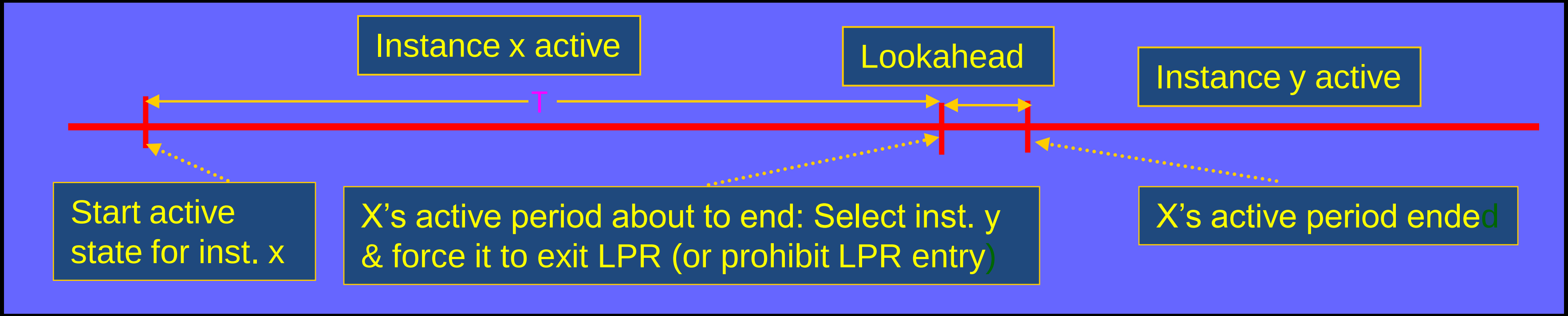
- A set of instances with a separate queue.
  - Instances of cpu cores, memory ranks, disk spindles, …
- Each queue has multiple servers (or resources)
- Keep only some instances active (or enabled)
  - Others inactive, but continue to accumulate traffic



Characteristics

- Enabled fraction (Rf)
  - Fraction of instance that are kept active
- Active Instances
  - New requests that can get a token are scheduled immediately.
  - If no ongoing requests, go into LPR mode.
    - May use reactive or proactive algorithm
- Inactive Instances
  - No scheduling of any new requests
  - When all ongoing requests finish, put it in LPR mode immediately
  - Starvation guard (via a timer)
    - Immediately substitute starved instances with an active instance.
    - Rotate victim instance to avoid any preferences

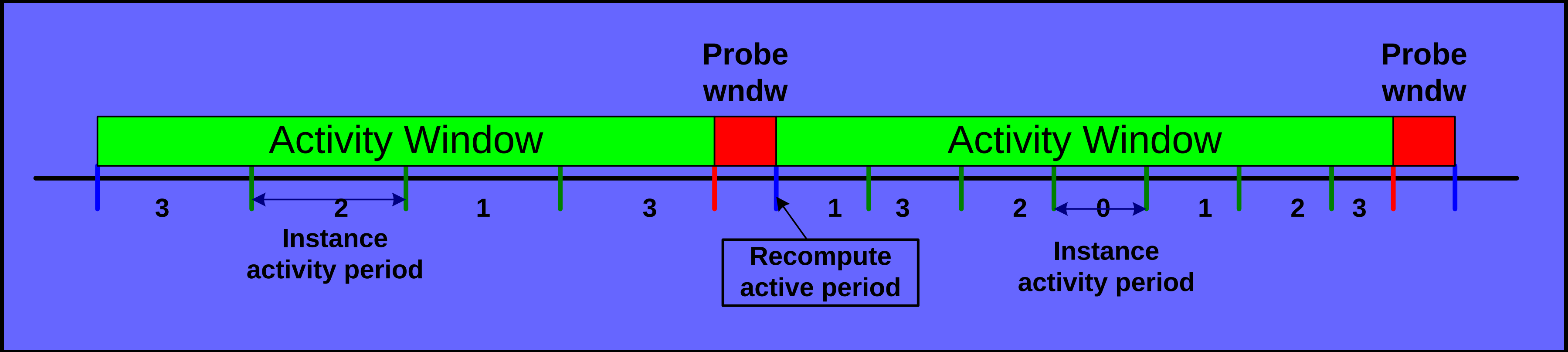
Instance Switching



- Look-ahead
  - Overlaps LPR exit of y with active state of x
  - Look-ahead time: LPR exit time
- Next instance selection
  - Several policies possible
    - Round robin: Usually bad
    - Instance w/ most waiting requests: Works well
    - Instance w/ most schedulable requests: Doesn't help much!

Activity Control

- Keep instance active for some time ("active window")
  - Gives throughput under activity control (λ_d)
- Remove activity control for "probe period"
  - Ideally, gives unperturbed throughput (λ_0)
- Estimate throughput degradation & adjust activity to keep degradation below a target



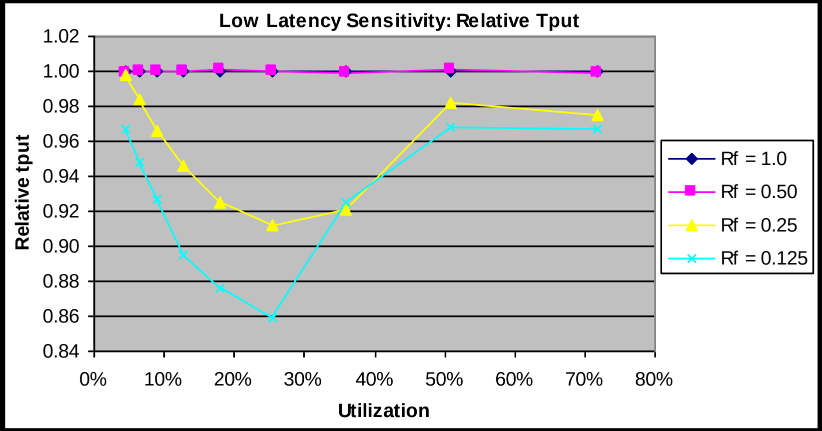
Activity Adjustment

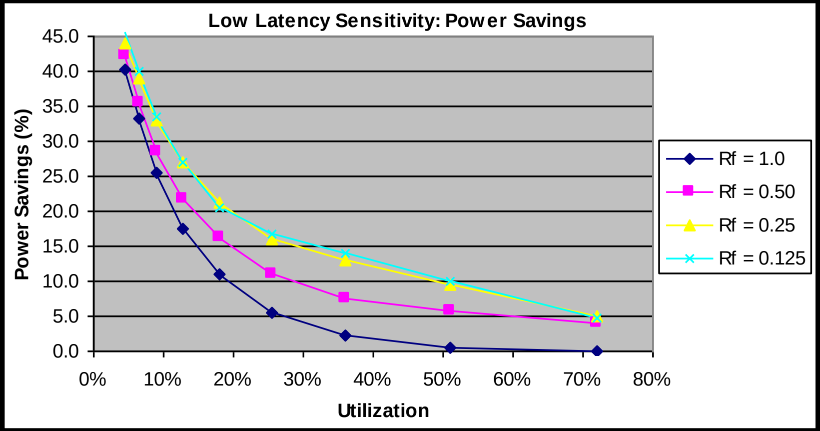
- Target throughput degradation (D), e.g., D = 5%
- Adjust active period T to ensure degradation ε [0.8D, D]

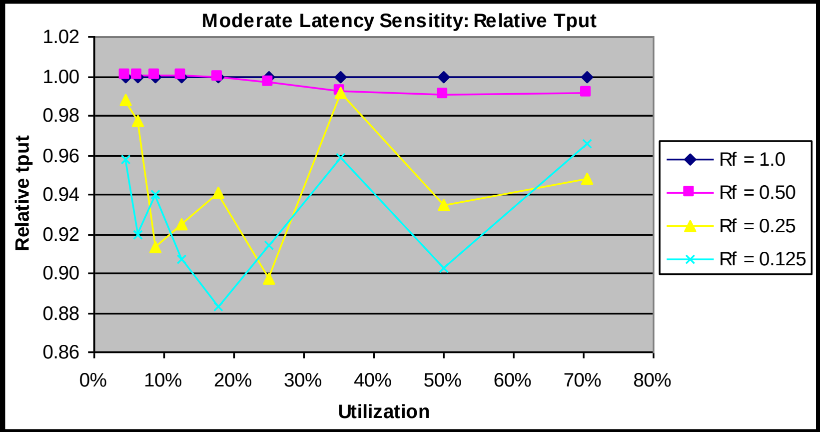
Condition	Action
Degradation < 0.8 D	Increase active period by Δ_1
D < Degradation <= 2D	Decrease active period by Δ_1
Degradation > 2D	Decrease active period by Δ_2
Degradation > D for N activity windows	Disable activity control until degradation < 0.8 D for N activity windows

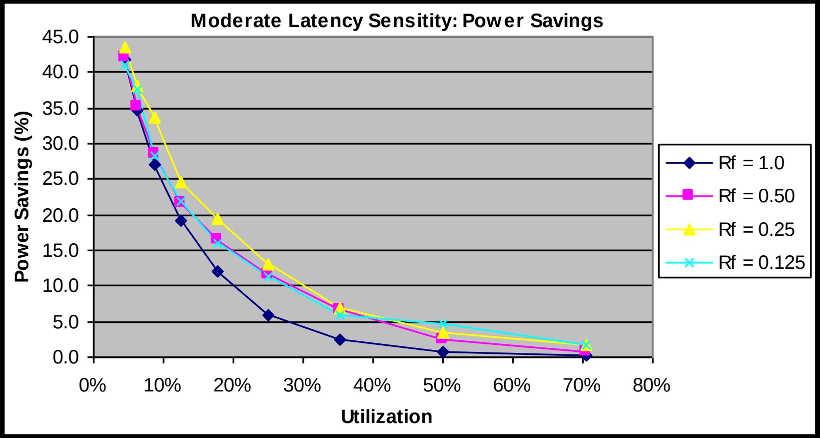
- **Explicit control on degradation**
  - **Activity control adds latency → Mechanism estimates tolerable latency & converts it to power savings!**

Sample Results









- **Works well for Rf = 0.5**
  - **For Rf < 0.5, throughput drop exceeds target (probing inadequate!)**

Observations & Issues

- Observations
  - Can provide additional power savings at high utilizations (isolated control will be useless here).
  - Latency insensitivity is key, else no savings!
- Issues
  - Probing period must be large enough to enable recovery.
  - Dependencies are a problem
    - Holding off a request may choke others
- Enhancements
  - Avoid requests to some instances altogether
    - E.g., by reorganizing data

Multi-level Coordination

- Data Center Level
  - Intelligent cooling controls (CRACs air volume & temperature, airflow direction, …)
  - Global workload placement/migration to alleviate impact of inefficient room level cooling (recirculation, hot-spot).
    - VM placement/migration to balance temperature (not load!)
    - Cooling/temperature aware scheduling of tasks
  - Coordination between servers, network (switches/routers) & storage systems
- Application Level
  - Management of various app components to meet QoS needs
  - App management to adapt to energy availability constraints

Multi-Level Coordination

- Management with each rack having independent cooling
  - Workload consolidation or some racks to minimizing cooling
  - Co-optimization of workload placement & cooling across racks
- Rack/Chassis Level with global cooling
  - Local workload placement/scheduling considering local controls (chassis or server fan speeds) and airflow issues
  - Temperature balancing & power consumption tradeoffs within rack/chassis
- Server Level
  - Coordination between CPU, MC/DRAM, adapters, etc.
- Potential conflicts between various control loops
  - Need to coordinate these control loops (game theoretic solutions?)

Future Challenges

Conclusions

- Numerous issues in data center energy management
  - Cooling, workload placement, migration, scheduling, adaptation, …
  - Power mgmt of servers, network, and storage
  - Varying levels of granularity (temporal and spatial)
  - Sustainability considerations bring in additional control actions (adaptation to available or consumable energy)
- Coordination is key to effective power mgmt
  - Coordination across components at a given level
  - Coordination across levels
  - Coordination among various control loops

Sustainability in Data Center Design

- Need to go beyond energy efficiency
  - Design devices/systems to minimize life-cycle energy and environmental footprint
  - Adapt to available energy & operate “at the edge”
  - Operation over variable/harvested energy sources.
- Future Directions
  - Coordinated server, network & storage adaptation to available/usable energy.
  - New mechanisms for workload adaptation & its coordination with power mgmt
  - Graceful QoS relaxation under energy constraints.

Thermal & Cooling Challenges

- Data Center Management
  - Optimization for total cost of ownership across different layers
  - Tools to visualize and understand power, thermal and performance issues and take appropriate actions.
- Thermal and Cooling Challenges
  - Feedback Loops between IT Equipment and Cooling System
  - Holistic cross-layer heat management
  - New load balancing algorithms that account for performance, thermal & power angles.

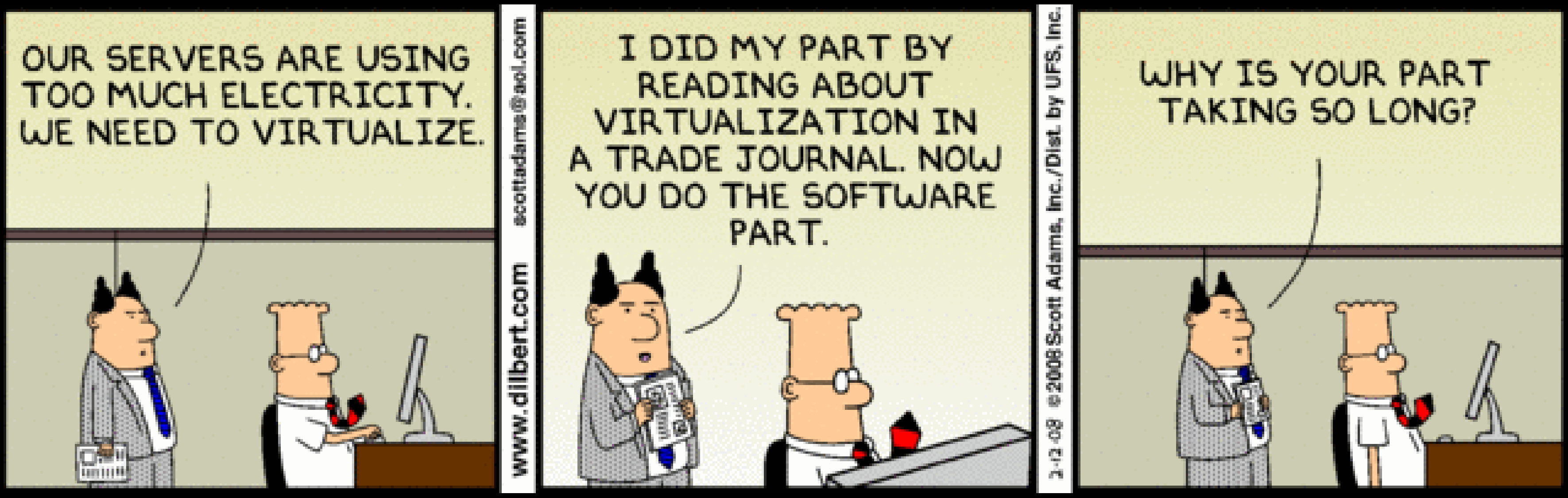
Modeling and Design Challenges

- Benchmarks, tools, and models
  - Measure and predict energy usage & availability.
  - Evaluation of multi-level of energy efficiency schemes
- Design of power mgmt features
  - How many power states do we need? What should be their characteristics?
  - How do we design effective controls?
- Theory for Tradeoffs between Energy, Performance and Reliability
  - Models to assist in obtaining bounds on performance under energy constraints (or vice versa)
  - Models to study dynamic power allocation among components to optimize performance.

Storage Energy Challenges

- Storage & storage energy will continue to grow.
- Technological challenges
  - Integration of (SSDs) into existing storage hierarchy to save energy.
  - Best mechanisms to use evolving NVRAM technologies.
- Storage Algorithms
  - Prediction & pre-fetching of required data for energy efficient reads & writes
  - Data de-duplication & exploiting data redundancies.
- Energy mgmt of storage devices and storage network.

Thank You!


OUR SERVERS ARE USING TOO MUCH ELECTRICITY. WE NEED TO VIRTUALIZE.
scottadams@aol.com
www.dilbert.com
I DID MY PART BY READING ABOUT VIRTUALIZATION IN A TRADE JOURNAL. NOW YOU DO THE SOFTWARE PART.
3-12-08 © 2008 Scott Adams, Inc./Dist. by UFS, Inc.
WHY IS YOUR PART TAKING SO LONG?