中国科学院大学
University of Chinese Academy of Sciences

Loongson

高级计算机系统结构

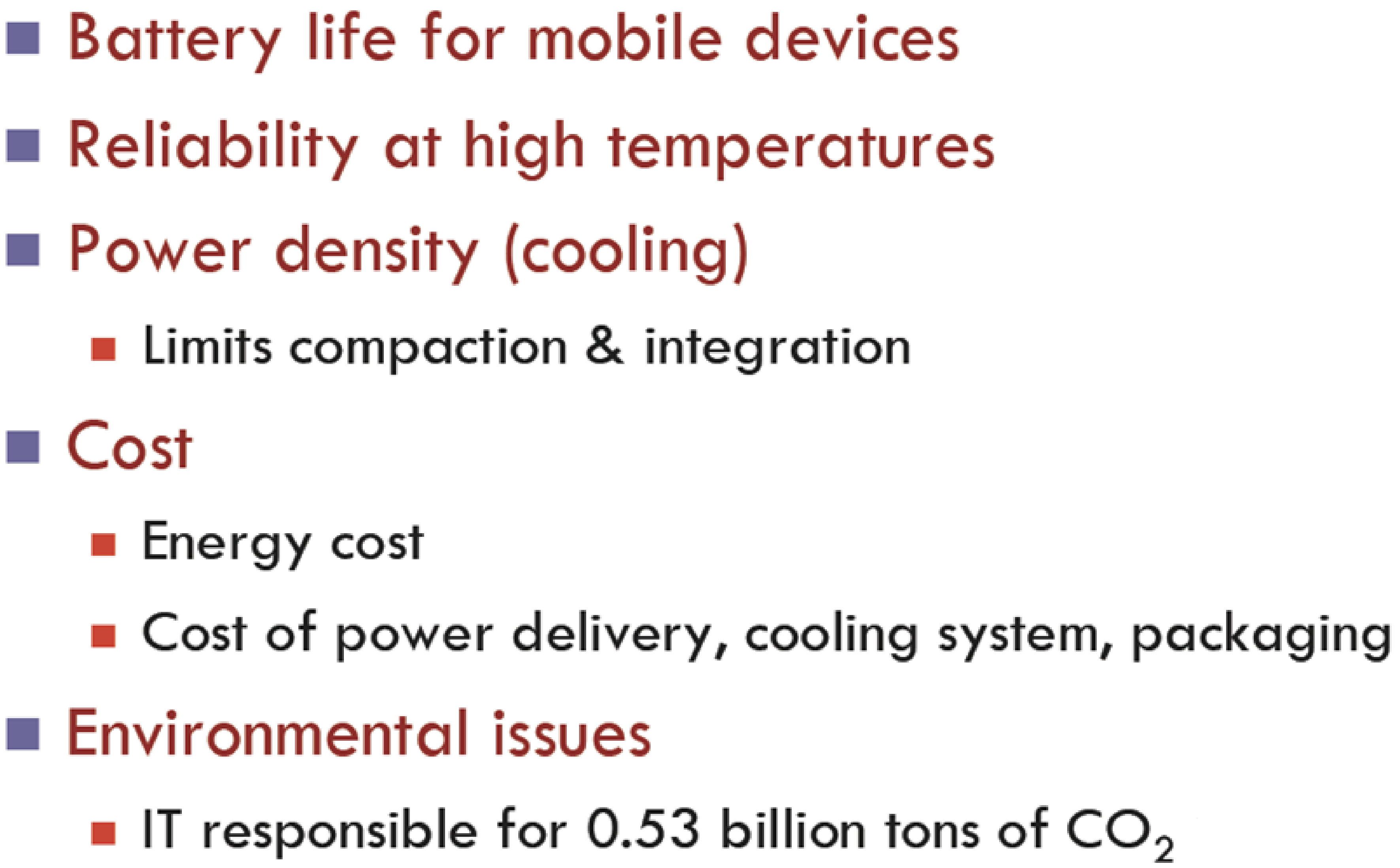
沈海华

shenhh@ucas.ac.cn

第十讲 Power Management

- Why are Power & Energy important?
- Power Metrics
- Power Consumption in ICs
- Typical Power Management
- DC: Power, Energy, & Cooling

Why Are Power & Energy Important?



- Battery life for mobile devices
- Reliability at high temperatures
- Power density (cooling)
  - Limits compaction & integration
- Cost
  - Energy cost
  - Cost of power delivery, cooling system, packaging
- Environmental issues
  - IT responsible for 0.53 billion tons of CO_2

Metrics

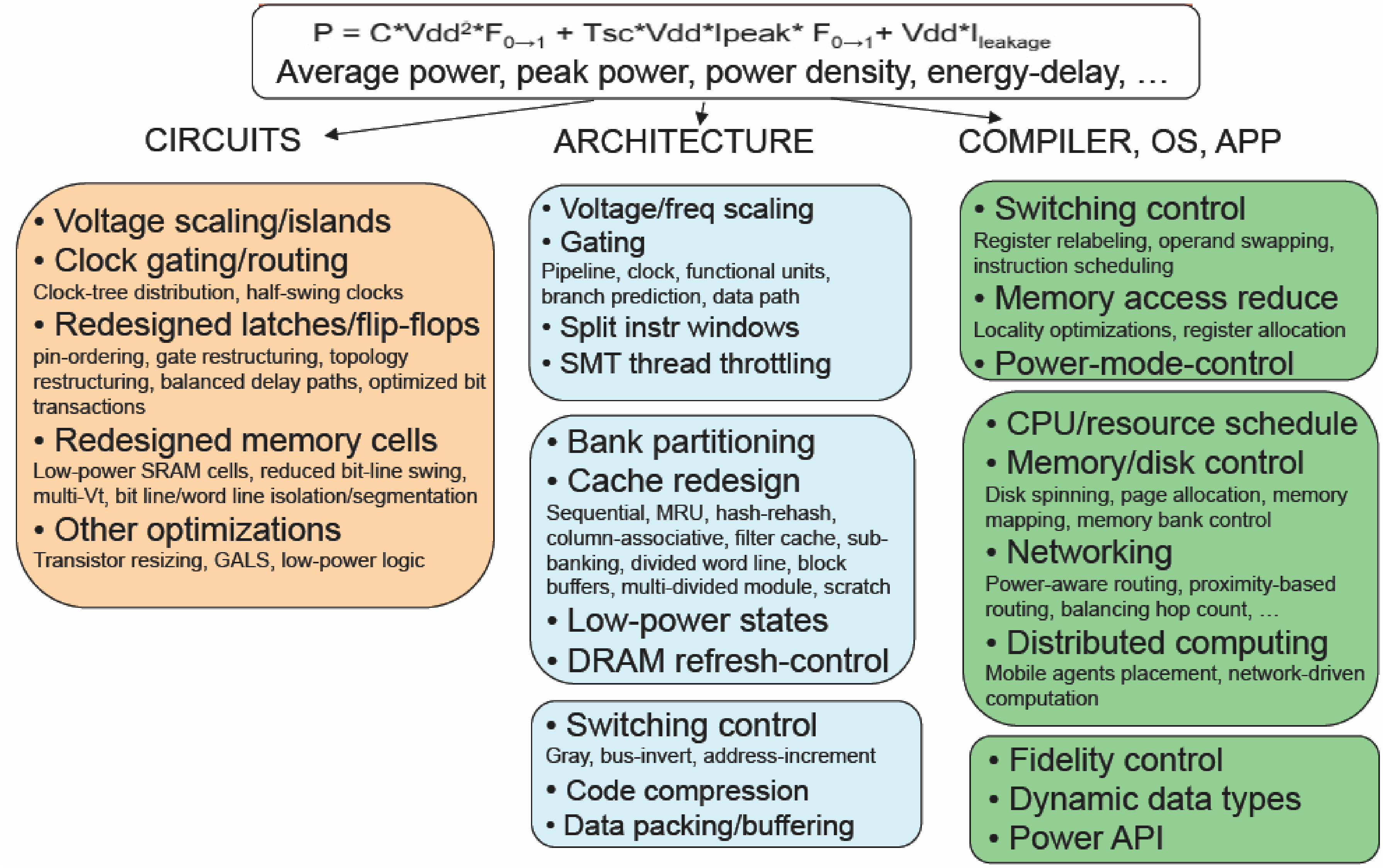
- Energy (Joules) = Power (Watts) * Time (sec)
  - Power is limited by infrastructure (e.g., power supply)
  - Energy: what the utilities charge for or battery can store
- Power density = power/area
  - The major metrics for the cooling system
- Combined metrics
  - How to tradeoff performance for power savings
  - TPS/W, energy x delay (EDP), energy x delay^2 (EDP^2), …

Power Consumption in ICs

$$P = C*Vdd^2*F_{0\rightarrow 1} + Tsc*Vdd*Ipeak*F_{0\rightarrow 1} + Vdd*I_{leakage}$$

- Dynamic or active power consumption
  - Charging and discharging capacitors
  - Depends on switching activity
- ~~Short circuit currents~~
  - ~~Short circuit path between supply rails during switching~~
  - ~~Depends on the size of the transistors~~
- Leakage current or static power consumption
  - Leaking diodes and transistors
  - Gets worse with smaller devices and lower Vdd
  - Gets worse with higher temperatures

A Sample of Power Optimizations



$P = C*Vdd^2*F_{0\rightarrow1} + Tsc*Vdd*Ipeak*F_{0\rightarrow1} + Vdd*I_{leakage}$

Average power, peak power, power density, energy-delay, …

CIRCUITS

- Voltage scaling/islands
- Clock gating/routing

Clock-tree distribution, half-swing clocks

- Redesigned latches/flip-flops

pin-ordering, gate restructuring, topology restructuring, balanced delay paths, optimized bit transactions

- Redesigned memory cells

Low-power SRAM cells, reduced bit-line swing, multi-Vt, bit line/word line isolation/segmentation

- Other optimizations

Transistor resizing, GALS, low-power logic

ARCHITECTURE

- Voltage/freq scaling
- Gating

Pipeline, clock, functional units, branch prediction, data path

- Split instr windows
- SMT thread throttling

- Bank partitioning
- Cache redesign

Sequential, MRU, hash-rehash, column-associative, filter cache, sub-banking, divided word line, block buffers, multi-divided module, scratch

- Low-power states
- DRAM refresh-control

- Switching control

Gray, bus-invert, address-increment

- Code compression
- Data packing/buffering

COMPILER, OS, APP

- Switching control

Register relabeling, operand swapping, instruction scheduling

- Memory access reduce

Locality optimizations, register allocation

- Power-mode-control

- CPU/resource schedule
- Memory/disk control

Disk spinning, page allocation, memory mapping, memory bank control

- Networking

Power-aware routing, proximity-based routing, balancing hop count, …

- Distributed computing

Mobile agents placement, network-driven computation

- Fidelity control
- Dynamic data types
- Power API

功耗问题 （I）

---------可以采取多个不同层次的工作去降低功耗。例如：

- 通过设计新的算法，减少程序执行过程中运算操作的次数；
- 通过设计新的power-efficient指令系统，如EDGE指令系统；
- 采用异构处理器结构，使得简单处理器核运行简单任务，复杂处理器核运行复杂任务，从而取得合理的能量消耗；
- 因为访问寄存器比访问存储器的功耗小，所以可以由编译优化减少访存次数来减少功耗；

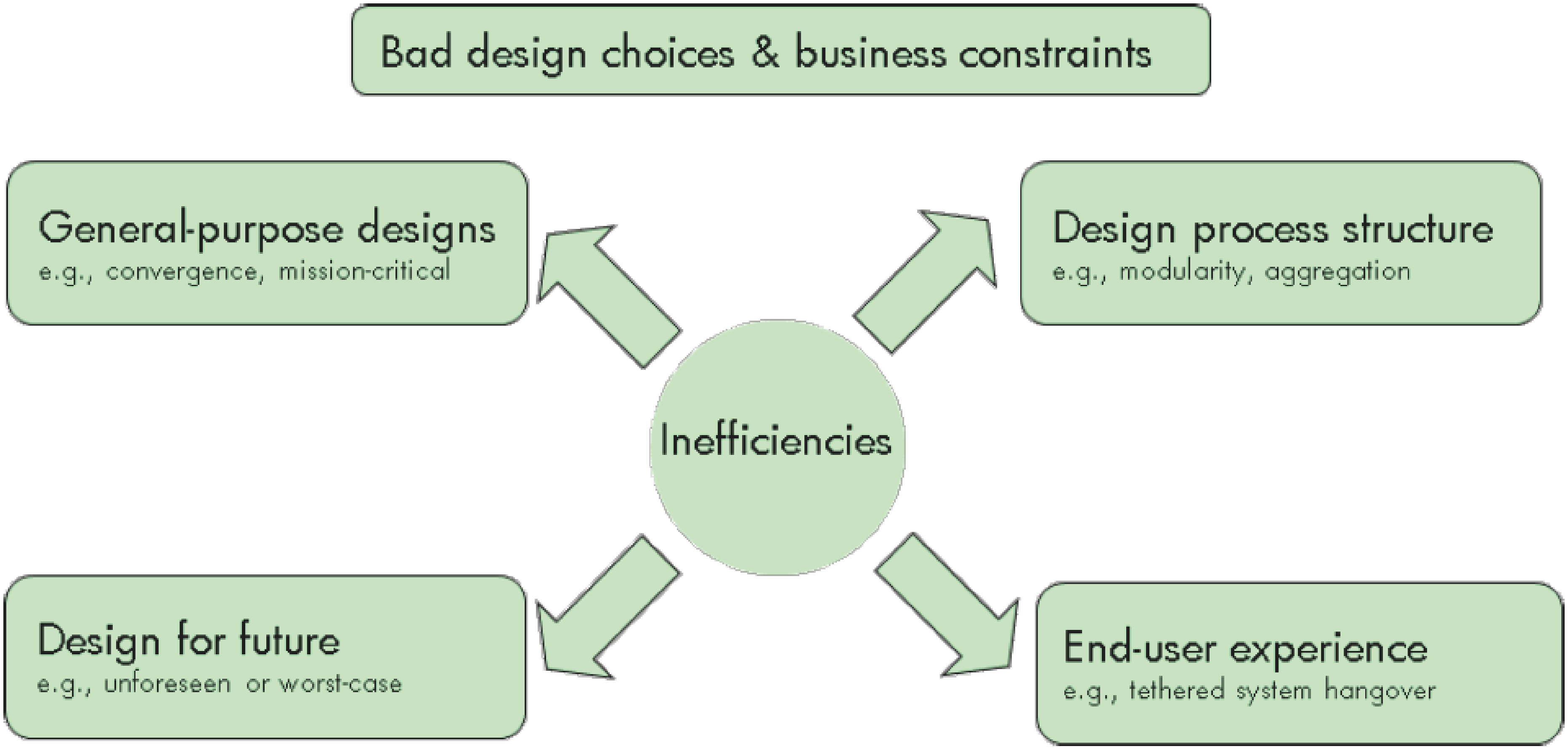
功耗问题 （II ）

- 可以通过操作系统的帮助，关闭当前不用的功能部件甚至处理器核来减少功耗；
- 可以通过操作系统动态调整处理器核的运行时钟频率或电压达到控制功耗的目的；
- 可以重新设计总线的编码方式，减少同时跳变的信号个数，来降低功耗；
- 可以通过对内存数据的压缩来减少片上存储（cache）容量，从而降低功耗；
- 在逻辑和电路层次上的低功耗设计方法更多，如门控时钟、门控电源、多阈值电压、动态电压变换、半频率时钟、异步逻辑等。
- 低功耗的片上网络设计也是多核处理器的重要研究方向。

Reducing Power/Energy

- An interdisciplinary issue
  - Circuits, architecture, software, systems
- Key high-level ideas
  - Reduce redundant work/components
  - Turn off unused components
  - Pick implementation that best matches constraints
    - E.g., don't use a 3GHz processor if 1GHz would do
- Our focus
  - A framework to reason about power management
  - A few selected examples

Sources of Inefficiency [CACM'10]



Bad design choices & business constraints

General-purpose designs
e.g., convergence, mission-critical

Design process structure
e.g., modularity, aggregation

Inefficiencies

Design for future
e.g., unforeseen or worst-case

End-user experience
e.g., tethered system hangover

Optimization Approaches [CACM'10]

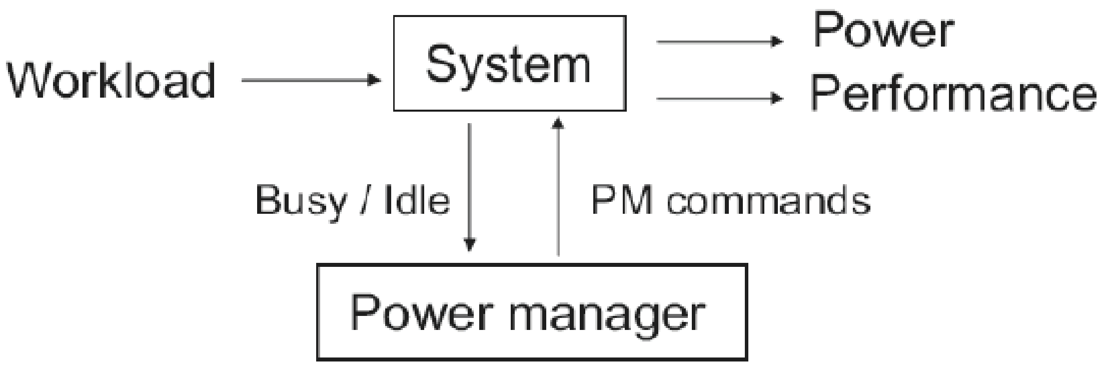
- Energy-efficient technologies
  - E.g., replace disk with flash, replace copper with optics, …
- Match power to work
  - E.g., turn-off/dial-down unused components, …
- Match work to power
  - E.g., asymmetric multicores, temp-aware scheduling, …
- Piggy-back energy events
  - E.g., shared caches, coalesced request streams, …
- Special purpose solutions
  - E.g., GPUs, ASICs, …

Optimization Approaches [CACM'10]

- Cross layers for efficiency
  - E.g., coordinate management across rack/cluster, …
- Tradeoff some other metric
  - E.g., fidelity-aware energy management
- Tradeoff the uncommon-case
  - E.g., power-supply efficiency, …
- Spend somebody's power
  - E.g., remote server offload for mobile power
- Spend power to save power
  - E.g., periodic cleanup to save energy, …

Typical Power Management

- Components have multiple power modes/states
  - Active: different levels of performance/power consumption
  - Idle: different power consumption/wake-up time
- Select power states to match constraints
  - Exploit fluctuations in use (requirements/idle times)
  - Done by the HW, OS, compiler, and/or the user
  - Tradeoffs: power saving Vs. QoS Vs. speed of resuming



Advanced Configuration and Power Interface(ACPI)

- 高级配置和电源管理接口
- 1997年由Intel、Microsoft、Toshiba 共同制定，提供操作系统应用程序管理所有电源的管理接口。先后推出ACPI 2.0（ 2000），ACPI 3.0（2004）， ACPI 4.0（ 2009）， ACPI 5.0（ 2011） 。
- ACPI可以包含下列功能：
  - 系统功耗管理（System power management）
  - 设备功耗管理（Device power management）
  - 处理器功耗管理（Processor power management）
  - 设备和处理器性能管理（Device and processor performance management）
  - 配置/即插即用（Configuration/Plug and Play）
  - 系统事件（System Event）
  - 电池管理（Battery management）
  - 温度管理（Thermal management）
  - 嵌入式控制器（Embedded Controller）
  - SMBus控制器（SMBus Controller）
  - ……
- 2013年10月，行业标准化组织UEFI（Unified Extensible Firmware Interface）论坛正式吸收ACPI规范，并承担管理ACPI规范的工作。

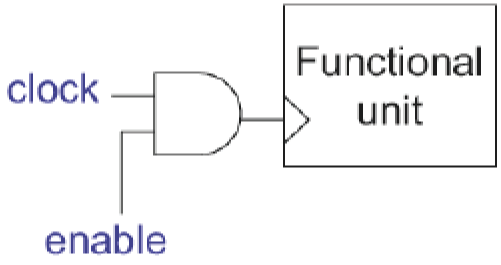
ACPI

- A standard for power management of systems
  - Describes power stages for system, devices, cores, …
  - Interface for SW to query and manage power states

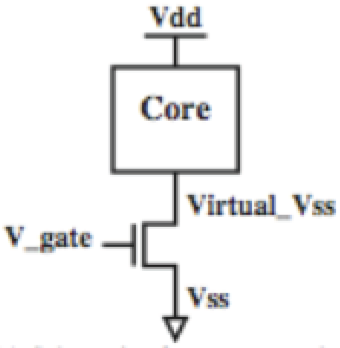
- Global system states
  - G0 working, G1 sleeping, G2 wakeup on LAN, G3 hard off
- Processor states
  - C0 is fully on
    - With P states related to DVFS stages (see following slides)
  - C1 to C3 are various idle modes
    - Clock may be stopped but state is maintained
  - C4 and beyond are various power off states
    - First the caches, then cores, and finally the whole chip

Power Management in Processors

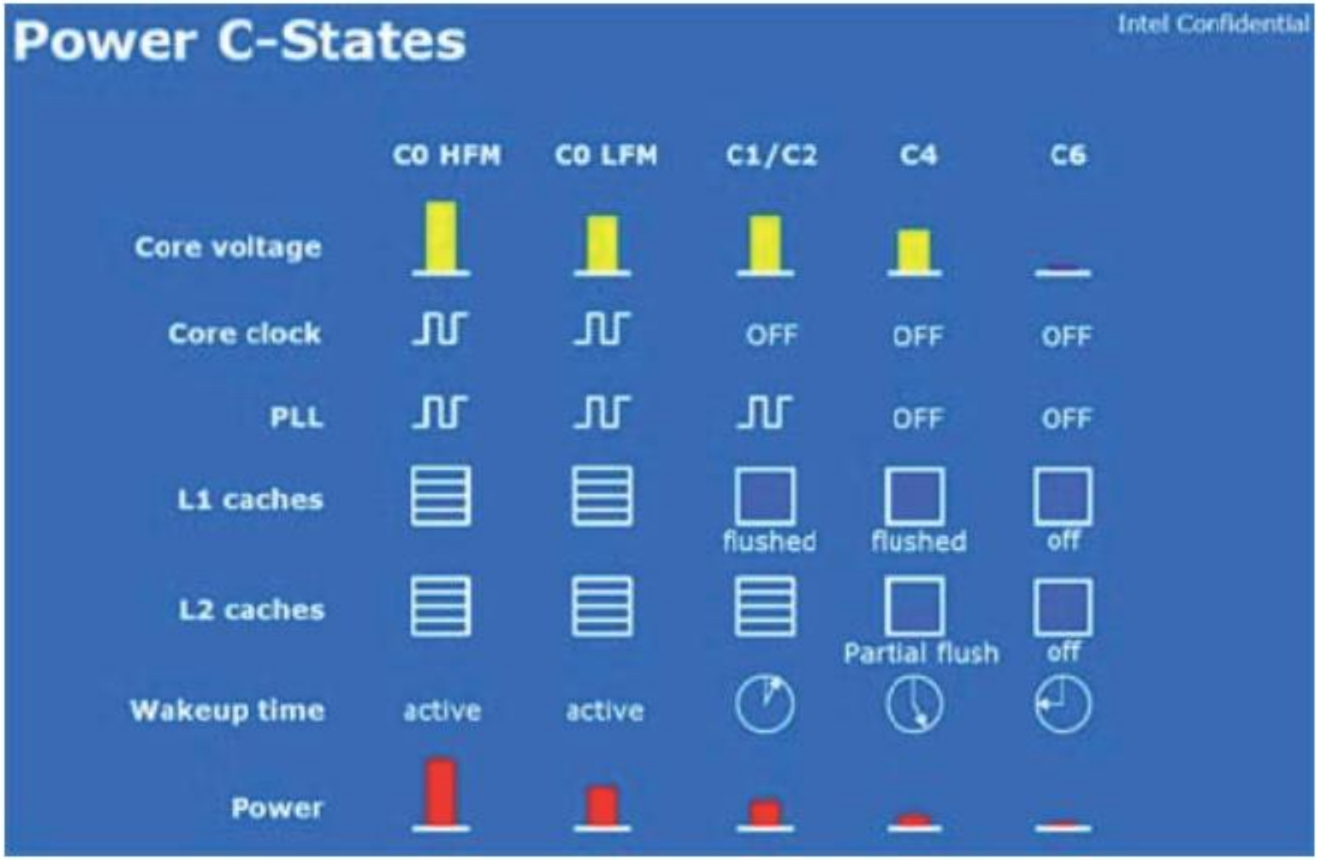
- Clock gating for idle units
  - Clock is a major power contributor
  - Done automatically in most designs
  - Near instantaneous on/off behavior



- Power gating (C4 or beyond)
  - Turn off power to unused cores/caches
  - High latency for on/off
    - Saving SW state, flushing dirty cache lines, turning off clock tree
    - Carefully done to avoid voltage spikes or memory bottlenecks
  - Area & power consumption of gate
  - Opportunity: use thermal headroom for other cores



Example C State Implementation



Power C-States

Intel Confidential

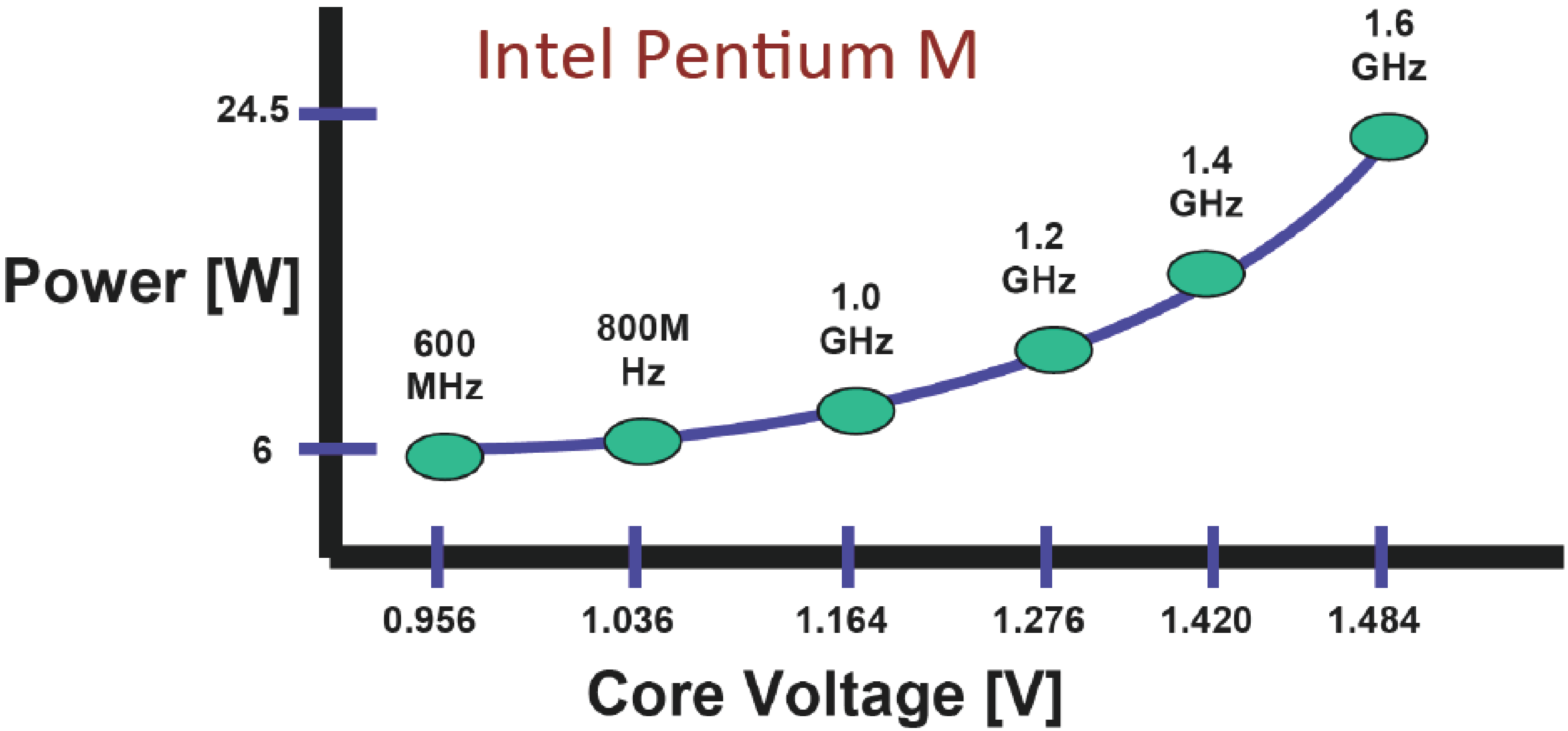
	C0 HFM	C0 LFM	C1/C2	C4	C6
Core voltage	▮	▮	▮	▮	_
Core clock	⎍	⎍	OFF	OFF	OFF
PLL	⎍	⎍	⎍	OFF	OFF
L1 caches	▤	▤	□ flushed	□ flushed	□ off
L2 caches	▤	▤	▤	□ Partial flush	□ off
Wakeup time	active	active	🕐	🕐	🕐
Power	▮	▮	▮	▬	_

- Gets more complex with multi-core and power-gating

DVFS: Dynamic Voltage/Frequency Scaling

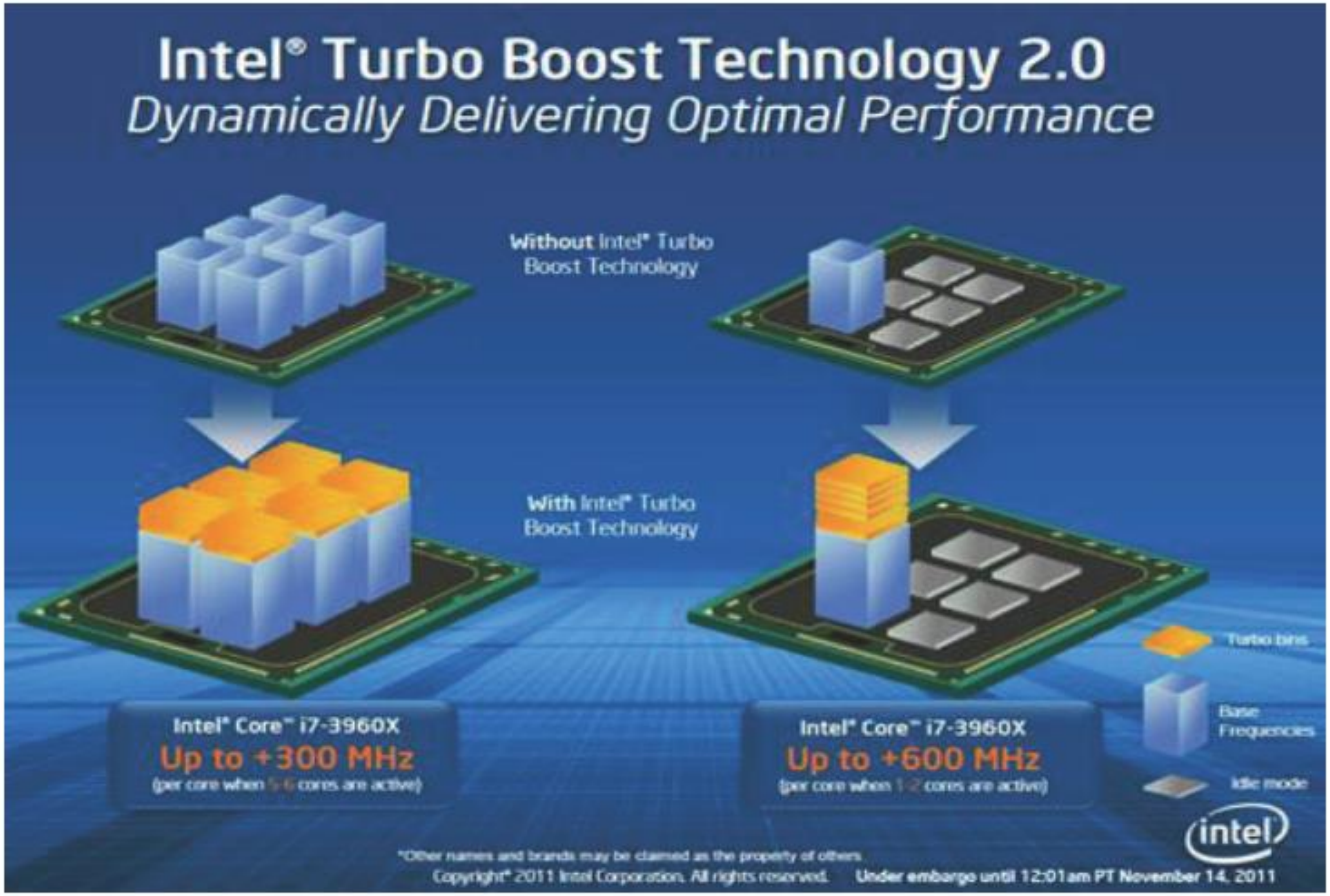
- Set frequency to the lowest needed
  - Execution time = IC * CPI * F
- Scale back Vdd to lowest for that frequency
  - Lower voltage => slower transistors
  - Power = C * Vdd^2 * F
- Provides P states for power management
  - Heavy load: frequency, voltage, power high
  - Light load: frequency, voltage, power low
  - Trade-off: power savings vs overhead of scaling
  - Effectiveness limited by voltage range

Exampled DVFS Implementation



- Transitions typically take a few usec

Turbo Mode（Intel i3/i5/i7）



Intel® Turbo Boost Technology 2.0

Dynamically Delivering Optimal Performance

Without Intel® Turbo Boost Technology
With Intel® Turbo Boost Technology
Turbo bins
Base Frequencies
Idle mode
intel

Intel® Core™ i7-3960X
Up to +300 MHz
(per core when 5-6 cores are active)

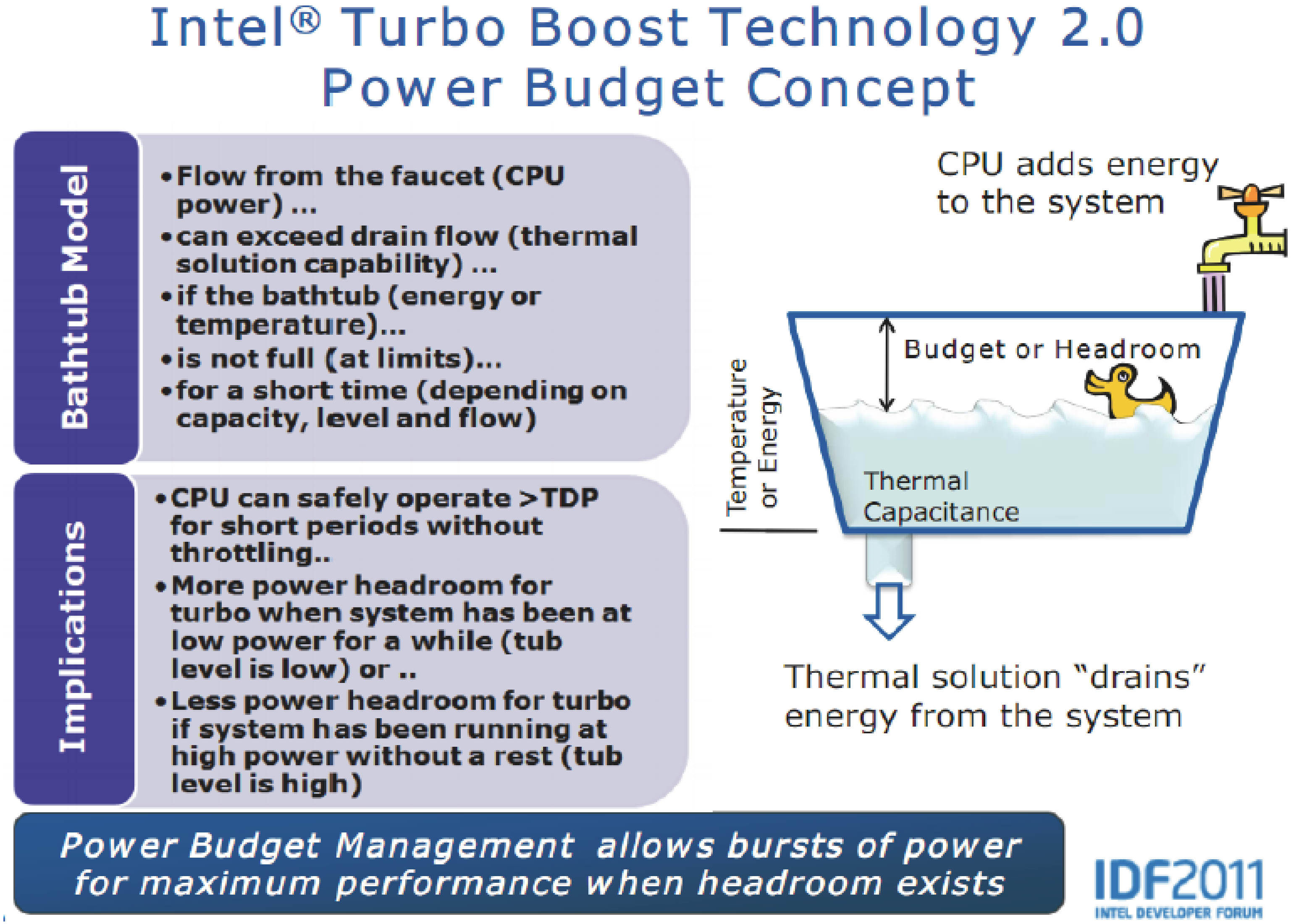
Intel® Core™ i7-3960X
Up to +600 MHz
(per core when 1-2 cores are active)

*Other names and brands may be claimed as the property of others.

Under embargo until 12:01am PT November 14, 2011

- Use power budget of idle cores to boost single thread perf

Power/Thermal Budgeting



Intel® Turbo Boost Technology 2.0 Power Budget Concept

Bathtub Model

- Flow from the faucet (CPU power) …
- can exceed drain flow (thermal solution capability) …
- if the bathtub (energy or temperature)…
- is not full (at limits)…
- for a short time (depending on capacity, level and flow)

Implications

- CPU can safely operate >TDP for short periods without throttling..
- More power headroom for turbo when system has been at low power for a while (tub level is low) or ..
- Less power headroom for turbo if system has been running at high power without a rest (tub level is high)

CPU adds energy to the system

Budget or Headroom

Temperature or Energy

Thermal Capacitance

Thermal solution "drains" energy from the system

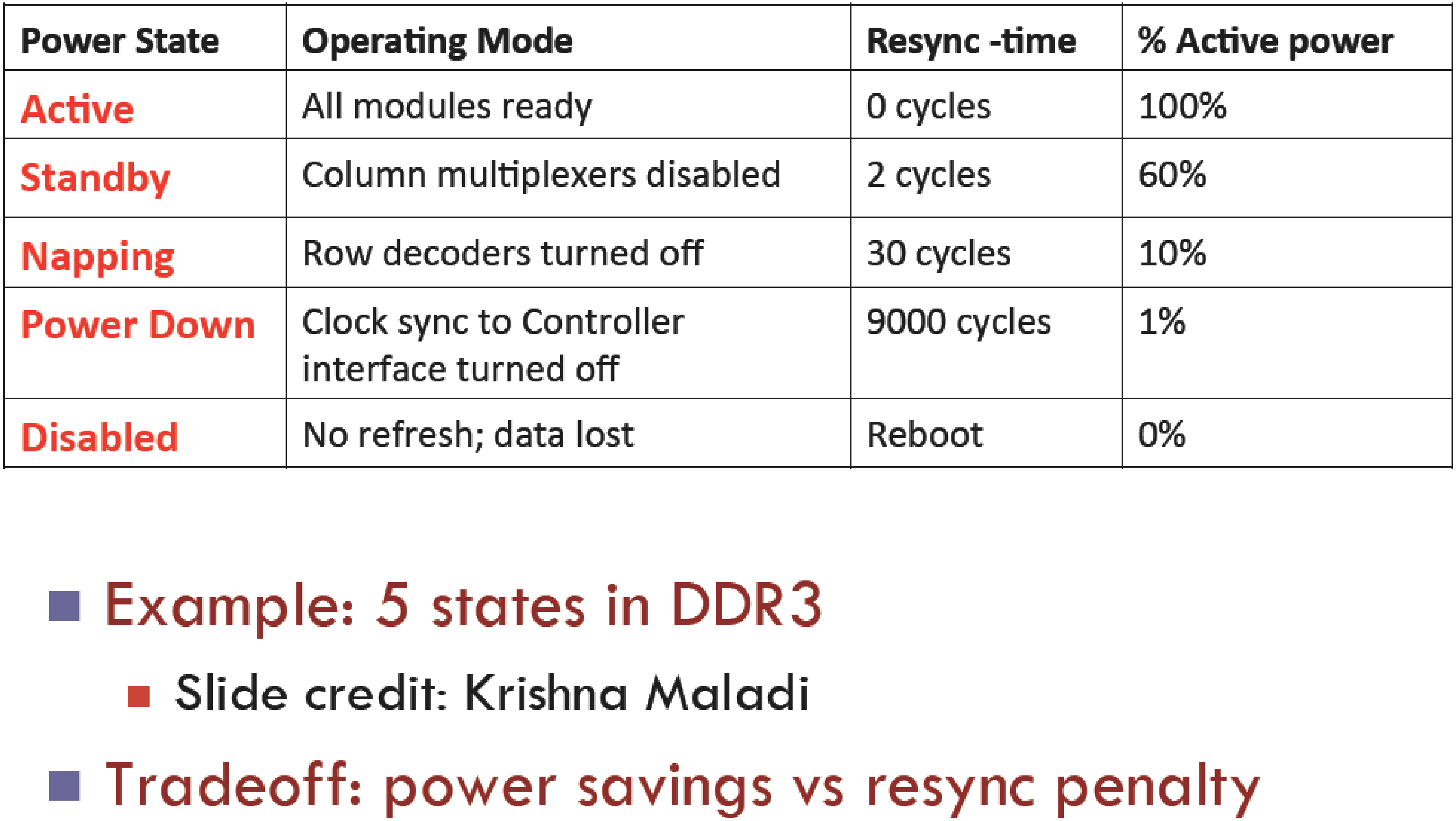
Power Budget Management allows bursts of power for maximum performance when headroom exists

IDF2011
INTEL DEVELOPER FORUM

DDR3 Power Management

- Idea: When not accessing a chip, power it down
- Power states
  - Active (highest power)
  - All banks idle (precharged)
  - Power-down
    - Fast: cut power to row buffers, Slow: stop clock distribution
    - Long latency to exit (i.e. ~100ns, hundreds of clock cycles)
  - Self-refresh (lowest power)
    - DRAM chip manages refresh independently
    - Very long latency to exit (i.e. 100s of ns to usecs)
- Chip cannot be accessed during state transitions!

DDR3 Power Management



Power State	Operating Mode	Resync -time	% Active power
Active	All modules ready	0 cycles	100%
Standby	Column multiplexers disabled	2 cycles	60%
Napping	Row decoders turned off	30 cycles	10%
Power Down	Clock sync to Controller interface turned off	9000 cycles	1%
Disabled	No refresh; data lost	Reboot	0%

- Example: 5 states in DDR3
  - Slide credit: Krishna Maladi
- Tradeoff: power savings vs resync penalty

Food for Thought

- Does the interleaving scheme affect DRAM power management?
- Would DVFS work for DRAM?

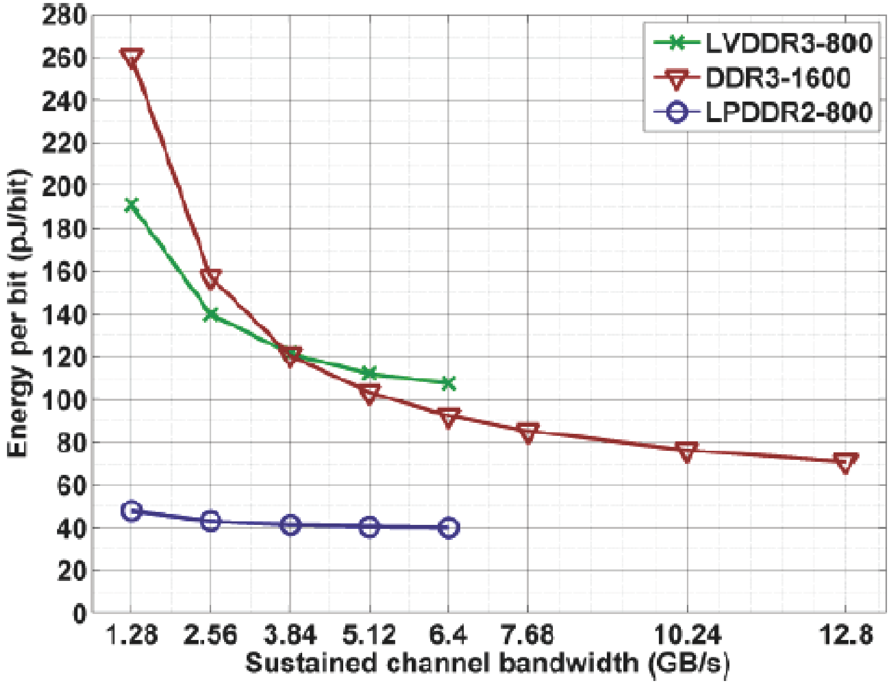
Broad View on DRAM power

- Problem: many chips involved with DRAM access
  - i.e. x4 chips on 64-bit bus means 16 chips per access!
- Problem: emphasis on high capacity and bandwidth
  - Fast clock means high voltage
    - DDR3 is 1.5V @ 800MHz
  - Multiple ranks means high-current drivers and termination needed, very high energy per bit

Mobile DRAM: LPDDR2

- Similar internal structure
  - Same manufacturing technology, latency, array structure
- Low voltage, low frequency: 1.2V @ 400MHz
  - $P = C * f * V^2$
- x16 or x32 interface! (often, <64 bit channel)
- No termination, no local clock regeneration
  - 1 rank maximum, so hard to get high capacity.
  - Fast exit from power-down mode (10s of ns)

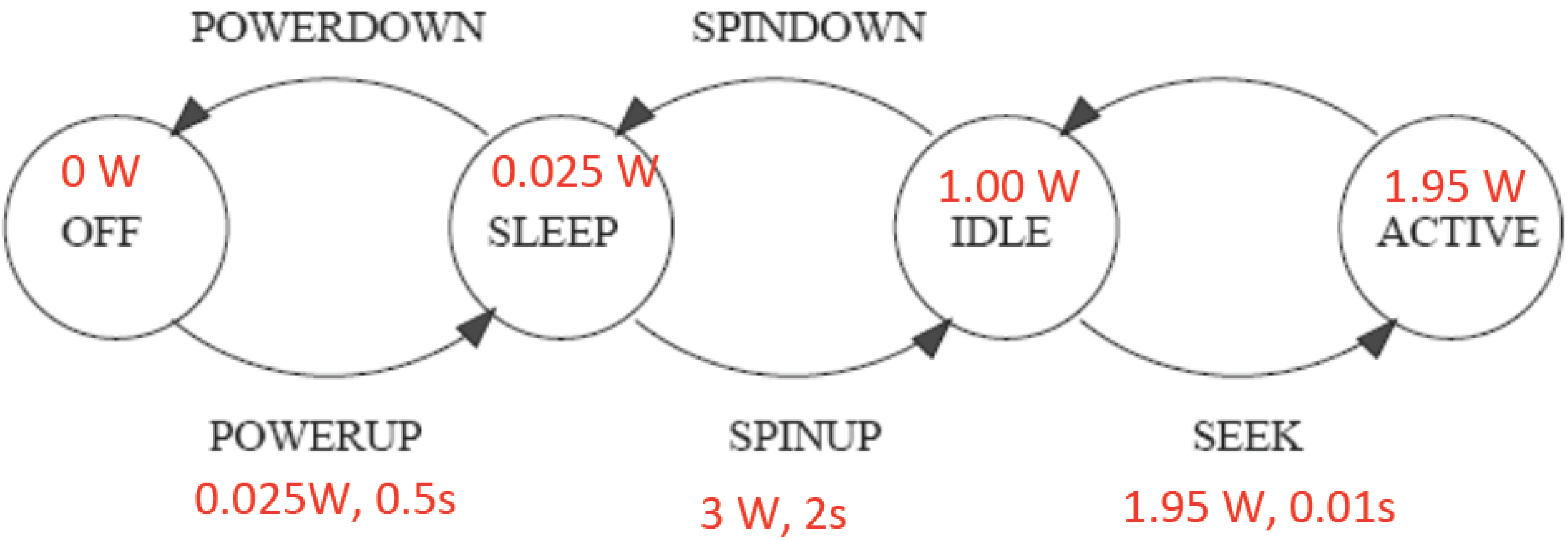
DDR3 vs LPDDR2



- ½ the pin bandwidth
- 1/5 the energy consumption
- Better proportionality
  - why is this important?

Disk Drive Power Modes

- Common optimization
  - Stop spinning disk when it is unused for a certain period of time



State diagram from Li, 1993.

Per-server Power Management

- Monitor and manage each server in a data-center
  - Q: how do you monitor power consumption?
- Various policies may be desirable
  - Power capping: limit power consumption to available budget
    - Why? Optimizations?
  - Conservation policy: switch to low power modes in the evenings
    - Why does this make sense?
  - Emergency issues: drop some server in lower power state on emergencies
- Other?

Cluster-level Power Management

- Power-aware load distribution to a server cluster
  - Try to create idle resources to send to low-power/off
  - Try to stay within "good" operating point of components
    - Coarser-grain power capping or temperature control
  - Interactions DVFS and inter-server load balance
  - Watch out for heterogeneity
  - Interactions with performance and more broadly SLAs
- Lots of policies
  - Predictions, economy-based, batching
  - What is the tradeoff?

Energy Proportionality

- Systems often underutilized
  - Diurnal traffic patterns, spikes, design for future growth, imbalance
- Need to have energy scale with work done
  - “Energy scale-down” or “do nothing well”
- Unfortunately most computers are not proportional
  - Why?

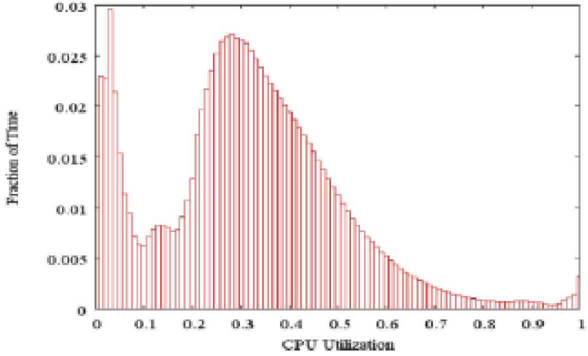


FIGURE 5.5: Activity profile of a sample of 5,000 Google servers over a period of 6 months

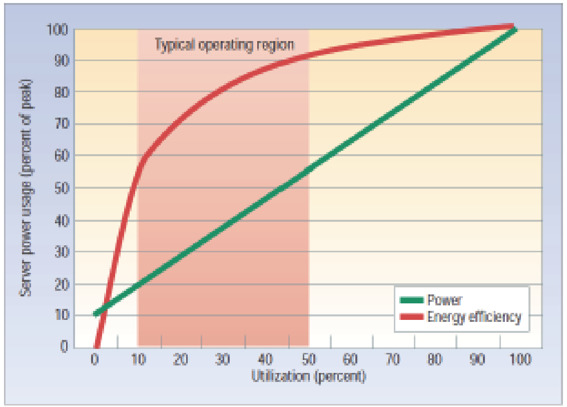


Figure 4. Power usage and energy efficiency in a more energy-proportional server. This server has a power efficiency of more than 80 percent of its peak value for utilizations of 30 percent and above, with efficiency remaining above 50 percent for utilization levels as low as 10 percent.

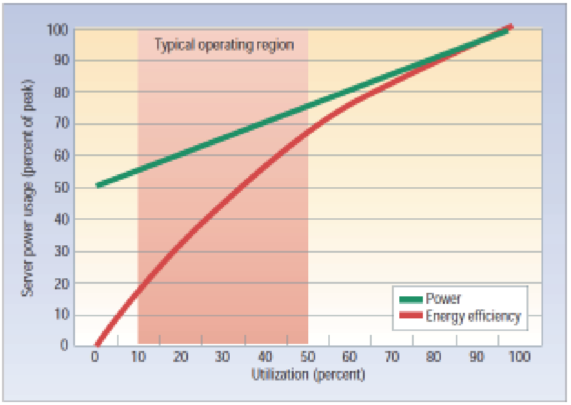


Figure 2. Server power usage and energy efficiency at varying utilization levels, from idle to peak performance. Even an energy-efficient server still consumes about half its full power when doing virtually no work.

In case you are wonder:Humans are More Energy Proportional

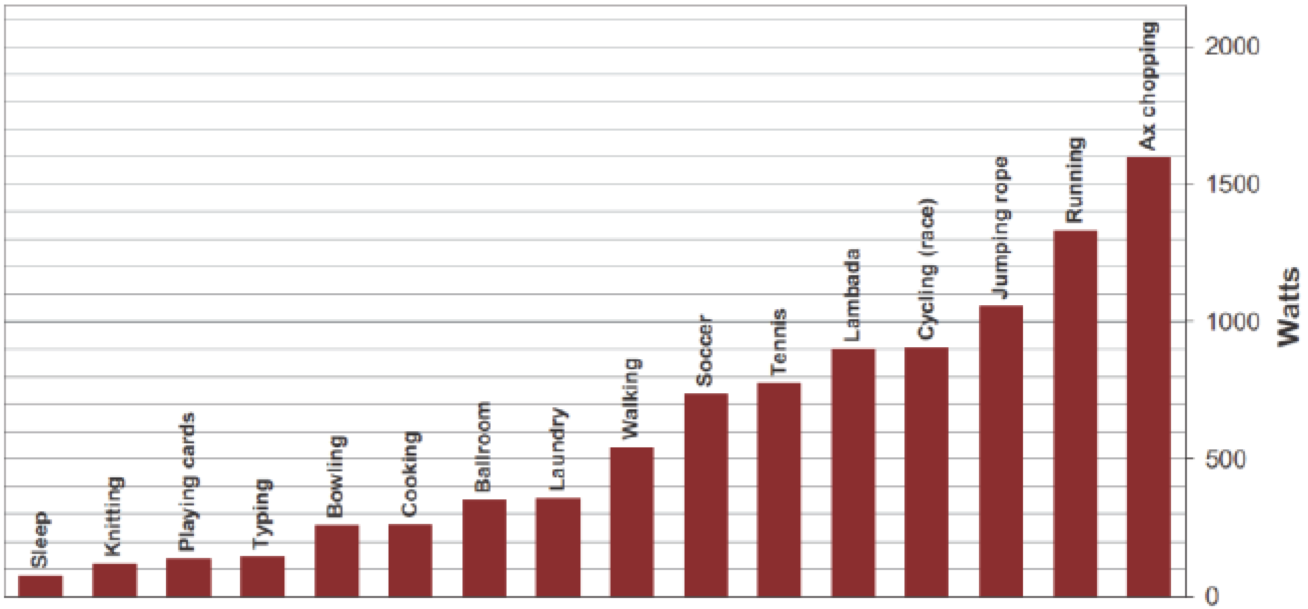


FIGURE 5.7: Human energy usage vs. activity levels (adult male) [52].

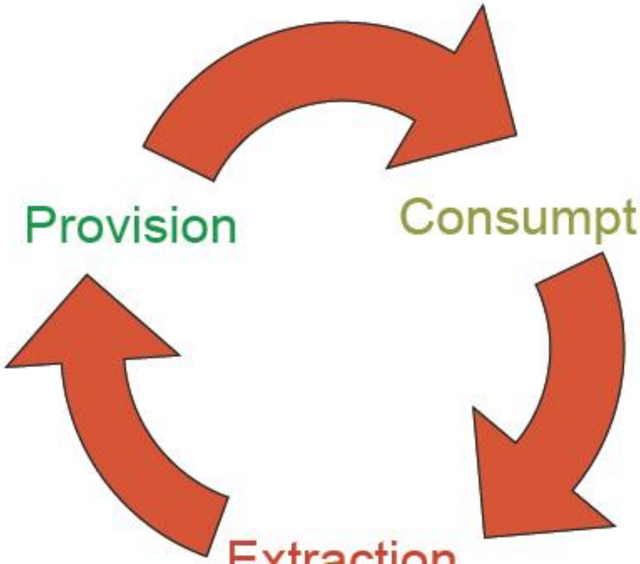
Optimizations for Energy Proportionality

- Component-based approach
  - Improve proportionality of *all* components
    - Processors: use DVFS or power-gating
    - Missing similar capabilities for memory/disks
  - Improve efficiency range of power distribution
- System-based approach
  - Resource aggregation
  - Scale-down: consolidate work & power-down idle servers
    - Issue: latency QoS robustness
  - Difficult if servers are part of the global file system
    - Energy proportionality Vs availability

DC: Power, Energy, & Cooling

The Power Lifecycle

(Feeds, PDUs, UPS, etc.)

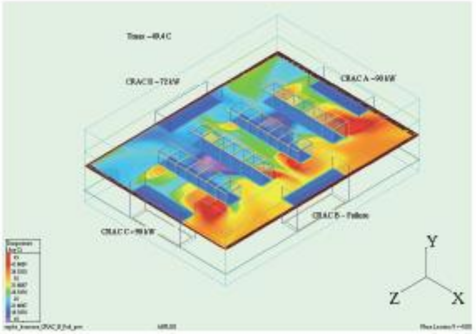


Provision

Consumption

Extraction

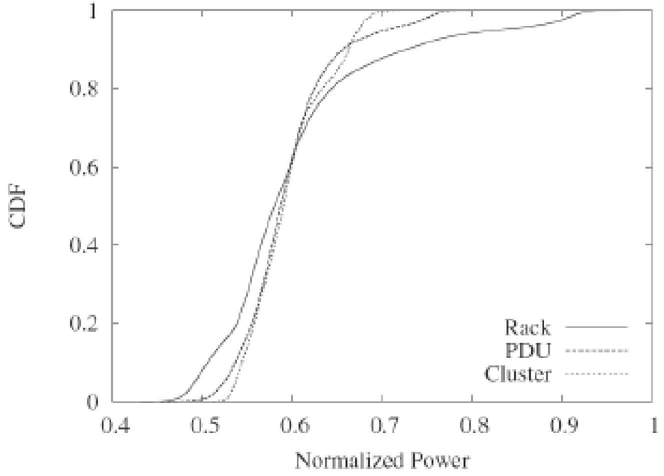
(Compute nodes)



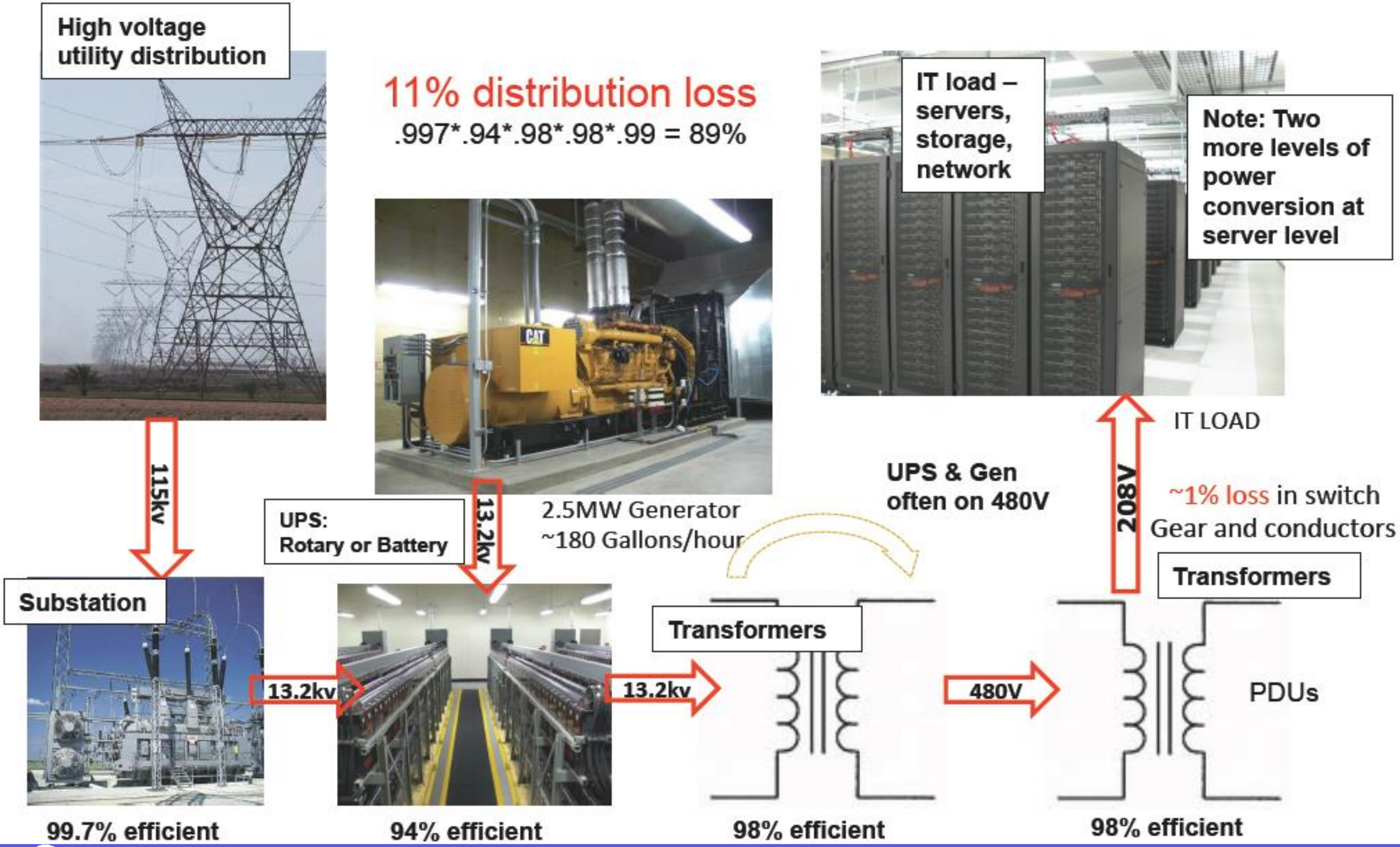
(CRAC units, floor tiles, etc.)

Energy Provisioning

- How much power does the DC need?
  - ~$15/Watt for building costs
- Conservative: max server power x # of servers
  - But for most apps, max power never reached
- Alternative: oversubscribe facility power
  - Using typical/max power consumption for apps
  - Requires careful management of unexpected peaks



Power Distribution (J. Hamilton)



High voltage utility distribution

11% distribution loss

.997*.94*.98*.98*.99 = 89%

IT load – servers, storage, network

Note: Two more levels of power conversion at server level

UPS: Rotary or Battery

2.5MW Generator
~180 Gallons/hour

UPS & Gen often on 480V

~1% loss in switch Gear and conductors

Substation

Transformers

Transformers

PDUs

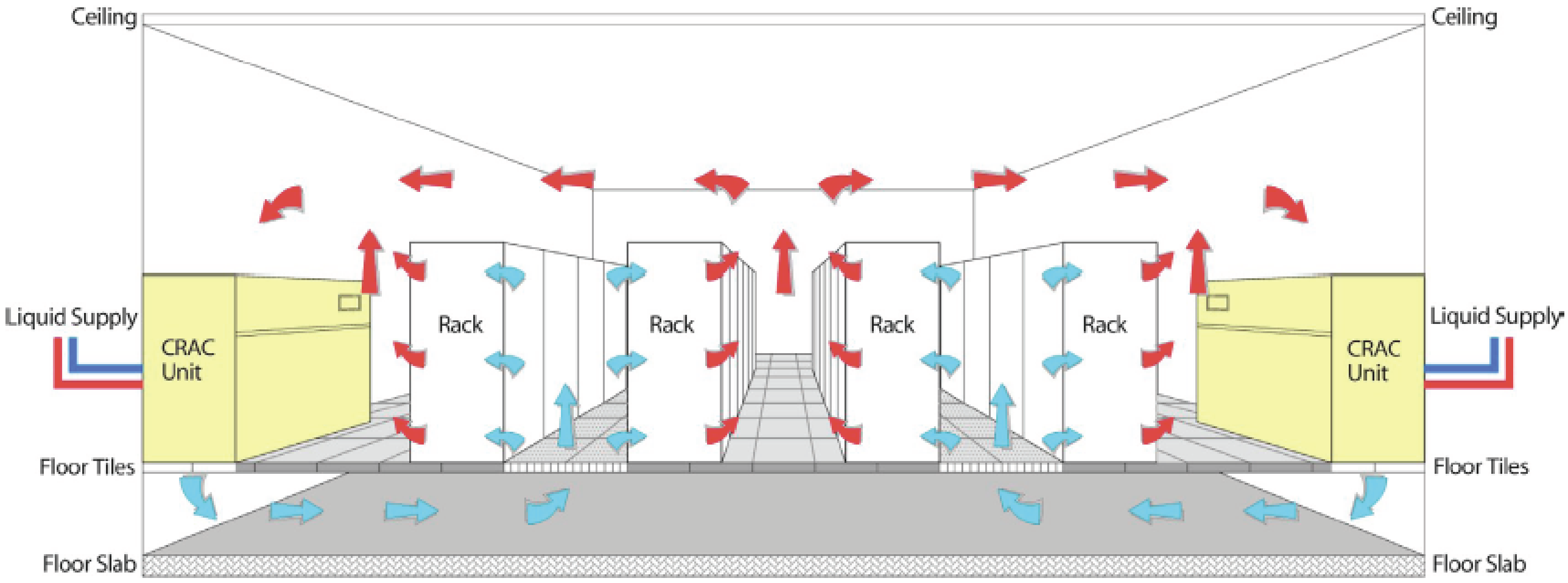
99.7% efficient

94% efficient

98% efficient

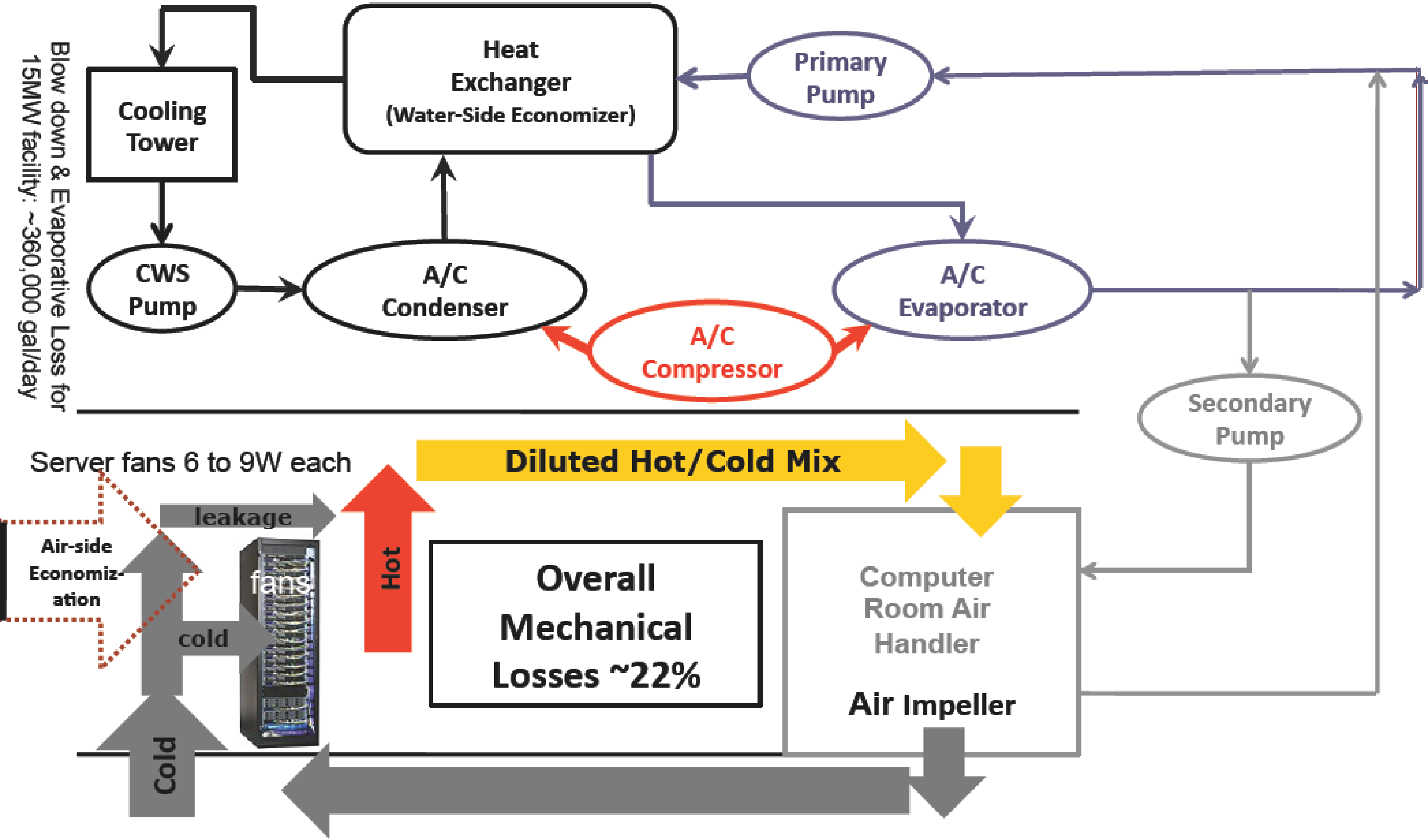
98% efficient

Cooling: Cold/Hot Aisles



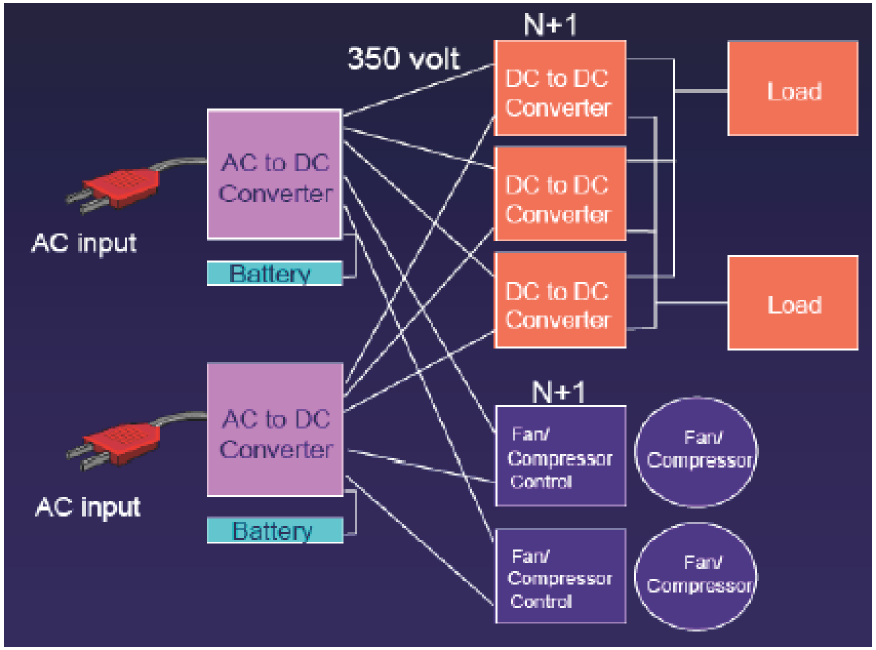
- CRAC = computer room air conditioning
  - Cold airs goes through servers and exits in hot aisle
  - Cold aisles ~18-22C, hot aisles ~35C
  - CRAC units consume significant amount of energy!

Cooling System Design (J. Hamilton)



Power/Cooling & Reliability

- Tiers of data centers
  - Tier I: no power/cooling redundancy – 99%
  - Tier II: N+1 redundancy for availability – 99.7%
  - Tier III: N+2 redundancy for availability – 99.98%
  - Tier IV: multiple active/redundant paths (2N) – 99.995%

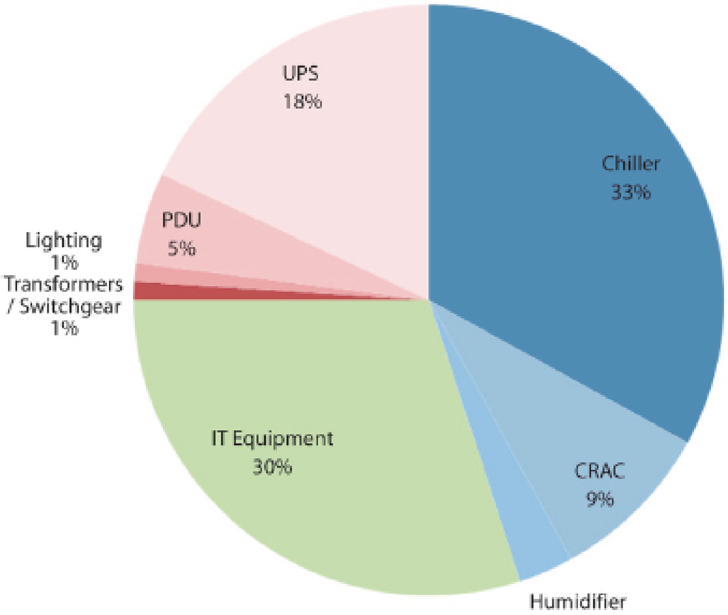


DC Energy Efficiency

$$\text{Efficiency} = \frac{\text{Computation}}{\text{Total Energy}} = \left(\frac{1}{\text{PUE}}\right) \times \left(\frac{1}{\text{SPUE}}\right) \times \left(\frac{\text{Computation}}{\text{Total Energy to Electronic Components}}\right)$$

- PUE = power usage effectiveness
  - Building power/power of IT (servers, switches etc)
  - Some DCs as bad as PUE = 3
  - Current state of the art PUE = ~1.2
- SPUE = server power usage effectiveness
  - Server power/power for CPUs, DRAM, disk, etc
  - Most servers have SPUE = 1.6
  - State of the art SPUE = 1.2
- If PUE=SPUE=1.2 => 30% of energy is “wasted”

Energy Use in a DC



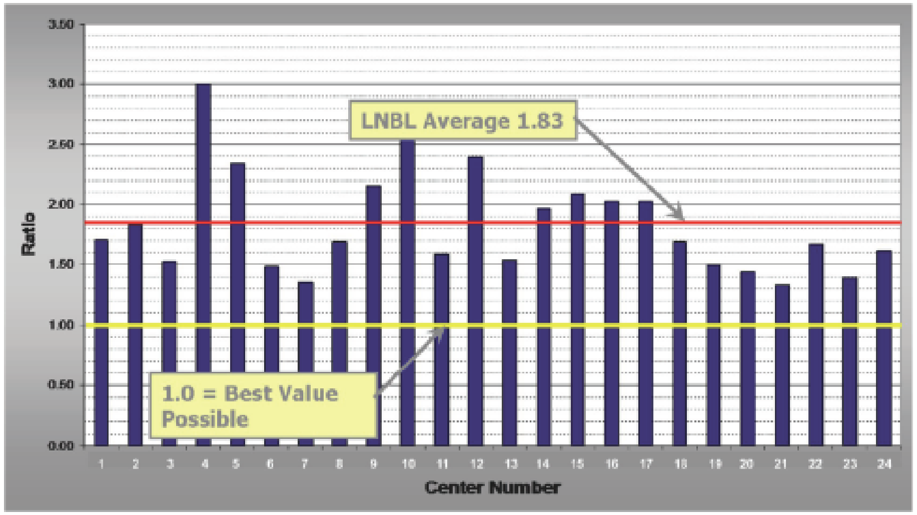


FIGURE 5.1: LBNL survey of the power usage efficiency of 24 datacenters, 2007 (Greenberg et al.)

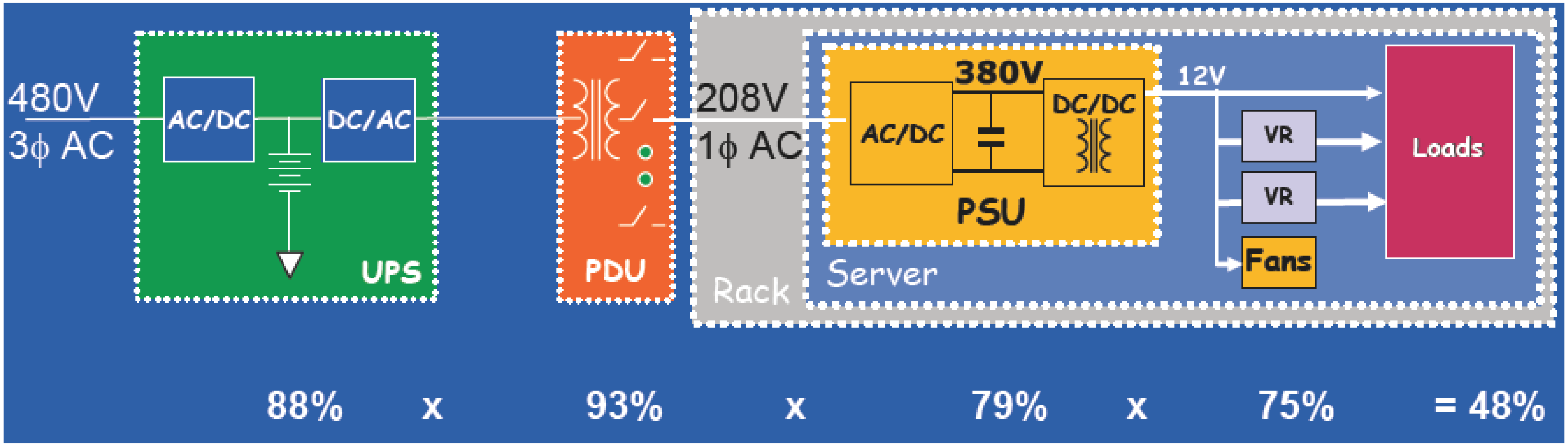
- Cooling infrastructure is a major contributor
  - Picture from a PUE=3 data center
  - Current datacenters: PUE: 1.1 to 2

PUE Optimizations: Power Distribution

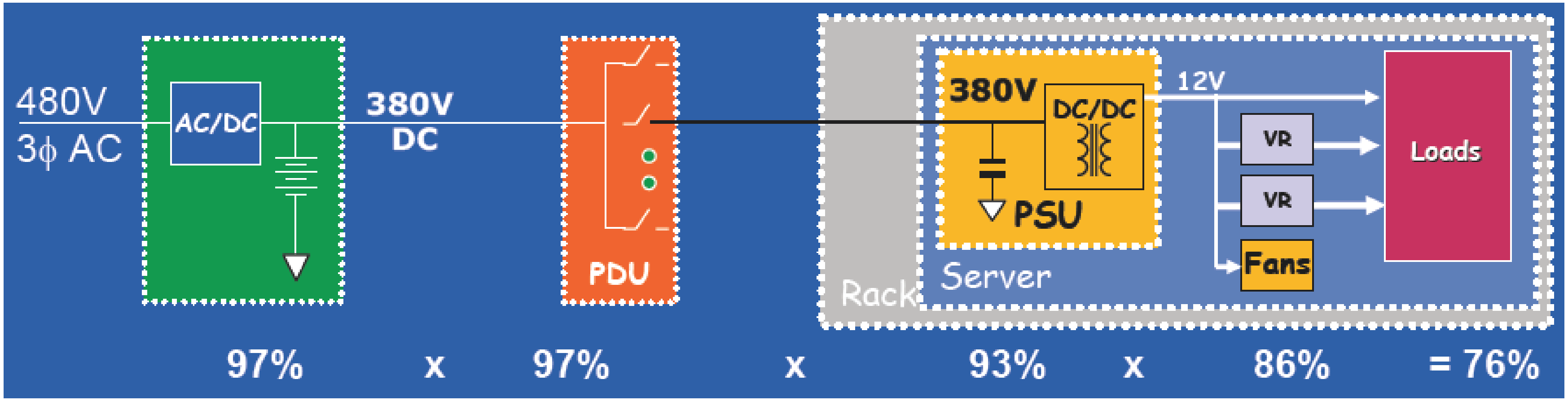
- Reduce number/increase efficiency of conversions
- High V close to load
- DC distribution within rack
- Per-rack or per-server 12-V battery
  - No UPS
  - Can also use "UPS" capability to deal with power spikes
    - But watch out for impact on battery lifetime

Example: Power Delivery Options

Conventional power distribution scheme (AC)



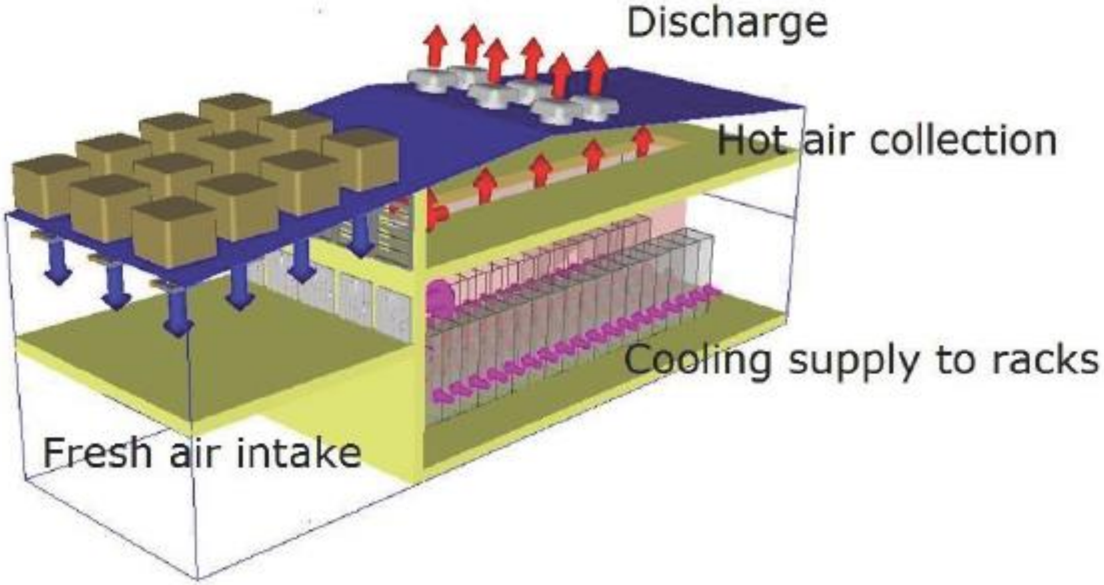
DC power distribution



PUE Optimizations: Cooling

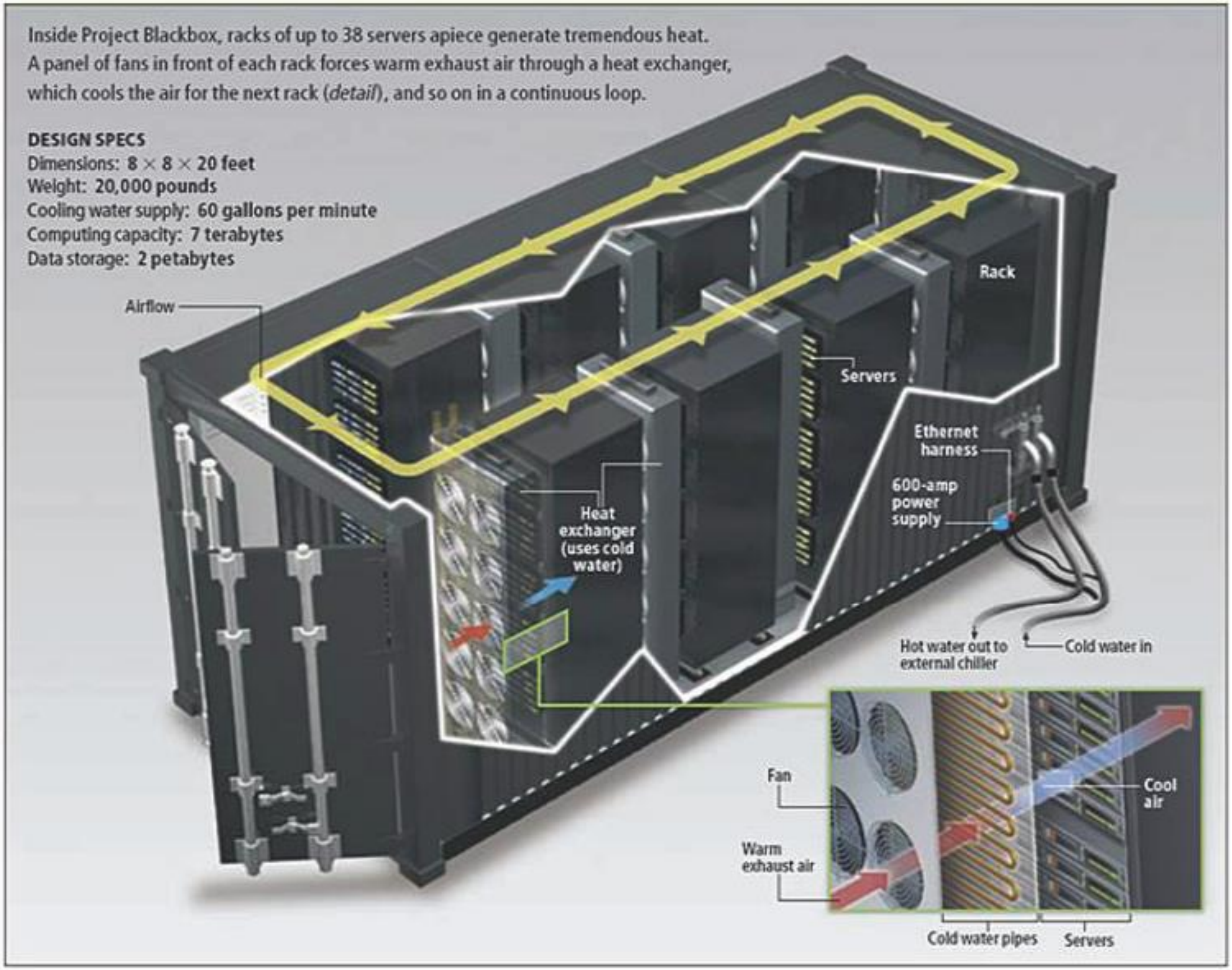
- Higher DC temperatures
  - E.g., 27C cold aisles
- Better airflow handling
  - Separate hot/cold aisles (e.g., using vinyl curtains)
  - Shorter airflow paths
- Air-side economization
  - Open the window ☺
- Water-side economization
  - Use cold water to avoid running A/C
- Waste-heat energy reclamation

Example: Airflow with Air-economization



- Open-loop system with mostly free cooling
  - Need to filter and push air around though

Container-based DCs



Inside Project Blackbox, racks of up to 38 servers apiece generate tremendous heat. A panel of fans in front of each rack forces warm exhaust air through a heat exchanger, which cools the air for the next rack (*detail*), and so on in a continuous loop.

DESIGN SPECS
Dimensions: 8 × 8 × 20 feet
Weight: 20,000 pounds
Cooling water supply: 60 gallons per minute
Computing capacity: 7 terabytes
Data storage: 2 petabytes

Container-based DCs

- Container = servers + heat exchange + power distribution
- Allows for various optimizations
  - In-rack cooling (water distribution through racks)
    - Allows higher power densities
  - Simplify server design (less metal) & wiring
  - Independent fire suppression (reduces insurance)
  - Allows for faster infrastructure innovation, easier incremental DC growth
- DC = a parking lot for containers (not a building)
  - + generator, UPS, chiller, …

致谢：

本讲内容参考了M.I.T. Daniel Sanchez教授的课程讲义，特此感谢。