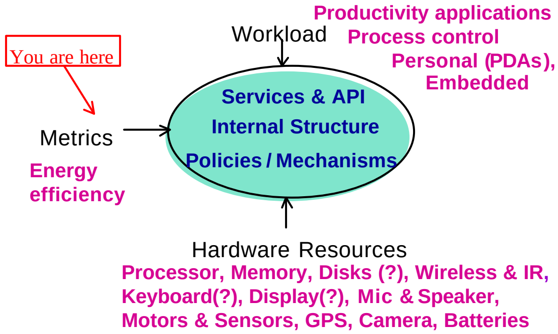
Rethinking OS Design



Energy Efficiency Metrics

- Power consumption in *watts* (mW).
- Battery lifetime in *hours* (seconds, months).
- Energy consumption in *Joules* (mJ).
- Energy * Delay
- Watt per megabyte

Physics Basics

- Voltage is amount of energy transferred from a power source (e.g., battery) by the charge flowing through the circuit.
- Power is the *rate* of energy transfer
- Current is the *rate* of flow of charge in circuit

Relationships

Energy (Joules) = Power (watts) * Time (sec)

$E = P * t$

Power (watts) = Voltage (volts) * Current (amps)

$P = V * I$

Current (amps) = Voltage (volts) / Resistance (ohms)

$I = V / R$

Battery Terminology

- Primary (non-reusable) and Secondary (rechargable)
- Voltages: V_{oc} (initial no-load)
  V (operating voltage under load)
  V_{cut} (cut-off when cell is considered discharged - 80% of V_{oc})
- Capacity expressed in amp-hours
  theoretical - based on amount of material in cell
  nominal - based on amp-hours obtained when discharged at constant current until V_{cut}

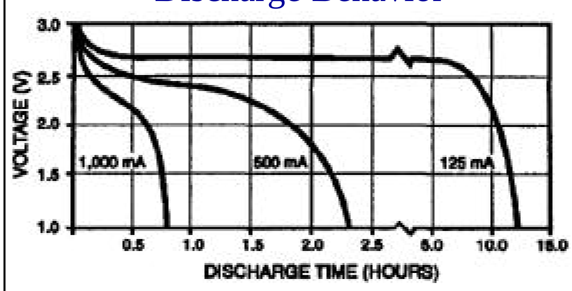
Battery Terminology

- Discharge time - elapsed time until a fully charged cell reaches V_{cut}
- C rate - discharge current expressed in amps relative to nominal capacity
  - example: for a lead acid battery with nominal capacity of 5Ah, a discharge rate of C/20 means 250mA of current.
- Specific energy - Watt-hours per kilogram delivered at constant discharge
- Energy density of cell - Watt-hours per liter

Battery Technology

Type	Volts	MAh	Rate	Wh/L	Wh/kg	cycles	loss
NiCd	1.2	1000	10C	150	60	1000	15%
NiMH	1.2	1200	2c	175	65	500	20%
PbAcid	2.0	400	C	80	40	200	2%
Li Ion	3.6	500	C	225	90	1200	8%
Li polymer	2.5	450	C/2	200	110	200	1%

Discharge Behavior

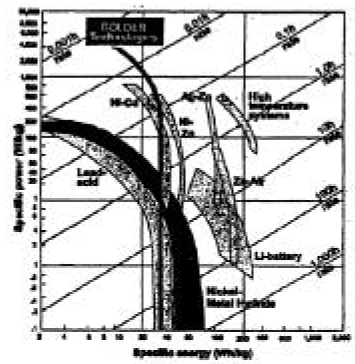


Discharge behavior of lithium-ion cell with $V_{oc} = 3V$ and $V_{cut} = 1V$

Battery Stuff

- Diffusion: At non-zero current, active material at electrode-electrolyte interface are consumed and replaced by new stuff moving in
- Polarization as current increases;
  At high enough current, diffusion is unable to compensate for depletion at electrode and cell voltage drops
- Recovery (due to diffusion) when current decreased

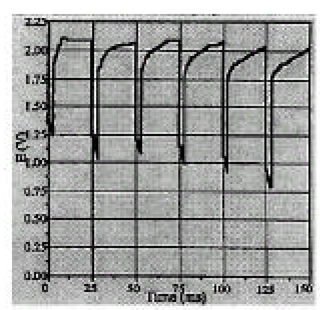
Ragone plot for different chemistries



Pulsed Discharge

- Exploiting recovery ability to get more out of a battery
- Delivered specific energy can be increased by pulsed instead of constant discharge for a fixed power level.
- [Chiasserini and Rao 99] - model & analysis
- Is bursty better for battery lifetimes?
  - Can durations of idle and busy states be optimized?

Pulsed Discharge

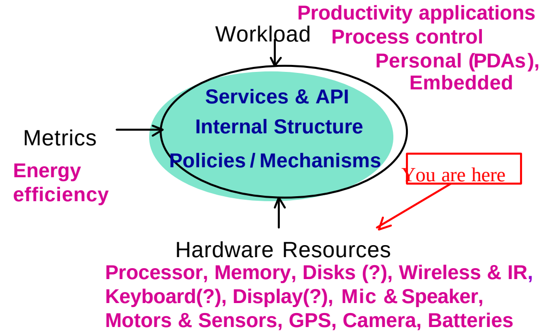


Bipolar
lead acid cell
Pulse = 3ms
Rest = 22ms

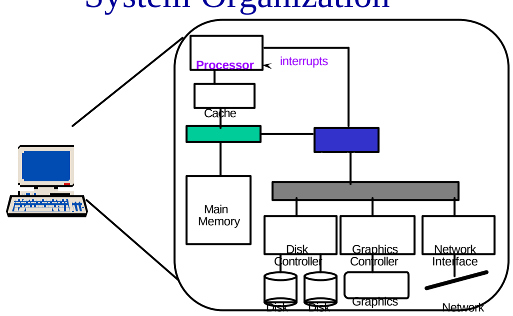
Smart Batteries

- Part of Intel Power Initiative
- Embedded battery controller that can be controlled by OS.
- Interface
  - Battery reports designed capacity, latest full charged capacity, remaining capacity.
  - Warning levels can be set. User notifications

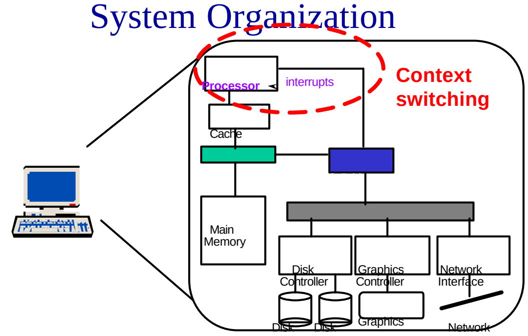
Rethinking OS Design



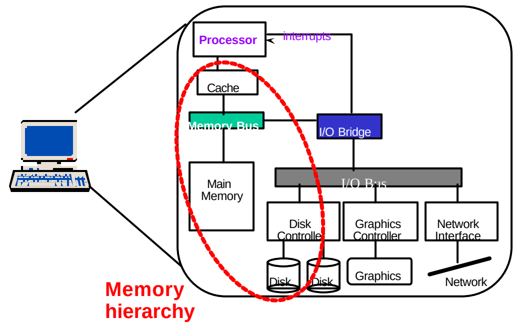
System Organization



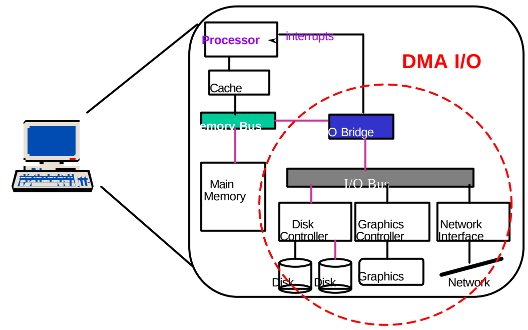
System Organization



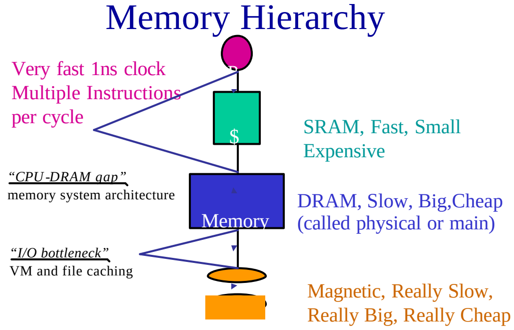
System Organization



System Organization



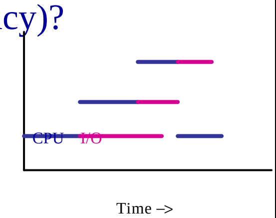
Memory Hierarchy



=> Cost Effective Memory System (Price/Performance)

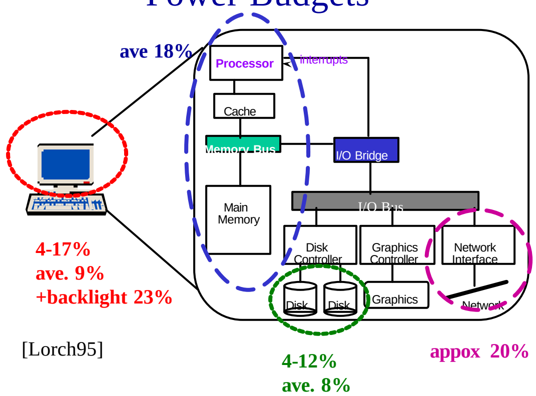
How to Improve Performance (Latency)?

- Exploit Overlap (HW parallelism)
  - Scheduling policies



- Use a lower-latency component
  - Memory Hierarchy / Caching, in general.
- Reduce overhead on the critical path.
  - Maintenance daemons

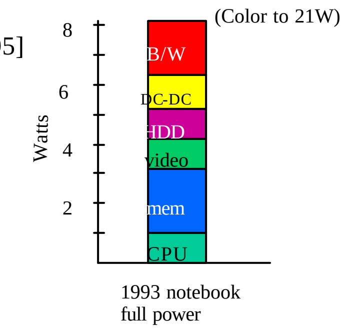
Power Budgets



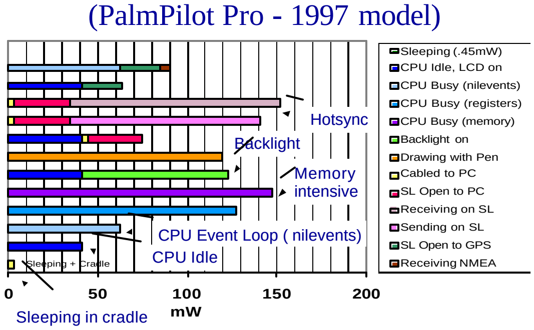
[Lorch95]

Typical Notebook Power Budgets

[Harris 95]



What are the Costs?
Measured Power Consumption
(PalmPilot Pro - 1997 model)



CPU/Memory

[Tiwari94]

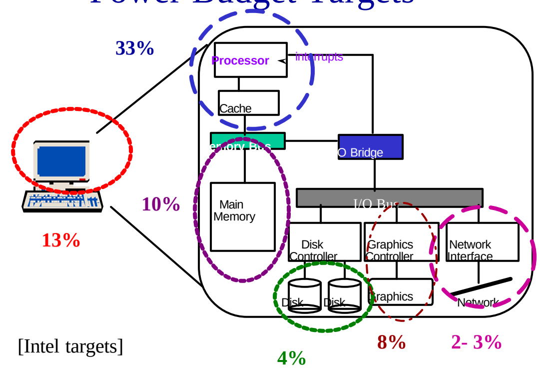
486DX2 Instr	current (mA)	Memory op	current (mA)
NOP	276	no access	5-77
Load	428	page hit	123
Store	522	page miss	248
Register add	314		
cache miss	216		

Intel Power Initiative Targets

Table 5.2 System Power Targets Comparison

	Mini Notebook	Full Size
	Average 3D Winbench* Power (W)	Average 3D Winbench* Power (W)
CPU & L2 Cache	5	9.5
Memory Controller	1.4	1.6
System Memory	.7	1.3
Graphics Subsystem	1.0	2.4
IO Subsystem	.5	.6
Audio	.7	1.8
Modem	.3	.4
Hard Drive	1.4	1.3
DVD Drive/CD	0	1.4
1394 Controller	0	0
CardBus	.2	.2
LAN	.4	.4
Power Supply	1.5	2.6
Charging	0	0
Cooling	0	0.5
Other	.8	1.0
Base Total	**13.9**	**24.8**
LCD	2.8	4.3
SYSTEM TOTAL	**16.7**	**29.1**

Power Budget Targets



[Intel targets]

Itsy Measurement Methodology

$I_{sense} = V_{sense} /.02$

Sampling rate: 5000 per second

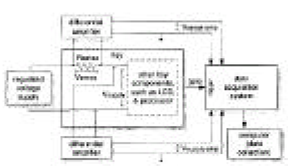


Figure 3: Block diagram showing the infrastructure used to measure the power consumed by the Itsy.

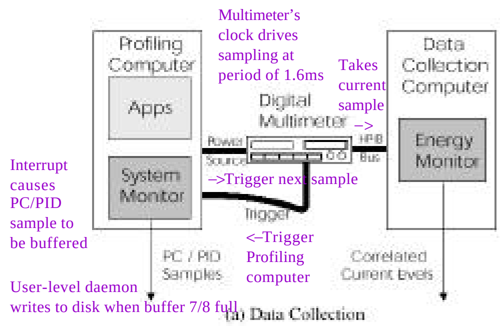
Itsy Results

# (1)	Micro-Benchmark (2)	Processor Voltage (3)	Itsy Power (Watts) at Specified MHz 59.0 (4)	132.7 (5)	206.4 (6)	Itsy Energy (Joules) at Specified MHz 59.0 (7)	132.7 (8)	206.4 (9)	ThinkPad Power (Watts) (10)
1	Sleep mode	normal	0.010	0.010	0.011	—	—	—	0.232
2	Idle mode	normal	0.094	0.128	0.162	—	—	—	3.30
3	Idle mode, LCD enabled	normal	0.134	0.164	0.198	—	—	—	5.15-7.24
4	Busy wait	normal	0.276	0.412	0.606	—	—	—	6.30
5		reduced	0.177	0.324	0.482	—	—	—	
6	Busy wait, LCD enabled	normal	0.265	0.447	0.637	—	—	—	10.4-13.1
7		reduced	0.217	0.363	—	—	—	—	
8	Addition loop	normal	0.314	0.612	0.889	6.388	5.534	5.228	7.42
9		reduced	0.247	0.490	0.729	5.020	4.437	4.181	
	Memory Test with instruction cache, MMU, write buffer and data cache enabled								
10	In-cache read test	normal	0.385	0.763	1.123	0.228	0.201	0.191	7.81
11	Out-of-cache read test	normal	0.458	0.719	0.777	8.423	6.429	5.880	8.74
12	In-cache write test	normal	0.385	0.763	1.130	0.227	0.200	0.189	7.43
13	Out-of-cache write test	normal	0.731	1.290	1.573	3.884	3.555	3.877	7.20
	Memory Test with only instruction cache enabled								
14	In-cache read-test	normal	0.504	0.800	1.023	3.717	3.291	3.197	
15	Out-of-cache read-test	normal	0.523	0.840	1.063	3.853	3.415	3.320	
16	In-cache write test	normal	0.506	1.076	1.183	3.838	3.018	3.617	
17	Out-of-cache write test	normal	0.506	1.076	1.183	3.574	3.013	3.617	

PowerScope [Flinn]

- Statistical sampling approach
  - Program counter/process (PC/PID) + correlated current readings.
  - Off-line analysis to generate profile
- Causality
  - Goal is to assign energy costs to specific application events / program structure
  - Mapped down to procedure level
  - System-wide. Includes all processes, including kernel

Experimental Setup Data Gathering



System Monitor Kernel Mods

- NetBSD
- recording of PC and PID
- fork(), exec(), exit() instrumented to record pathname associated with process
- new system calls to control profiling
- pscope_init(), pscope_start(), pscope_stop(), pscope_read() (user-level daemon, to disk)

Energy Analyzer

- Voltage essentially constant, only current recorded.
- Each sample is binned into process bucket and procedure within process bucket.
- Energy calculated by summing each bucket

$$E = V_{meas} \sum_{t=0}^{n} I_t \, \Delta t$$

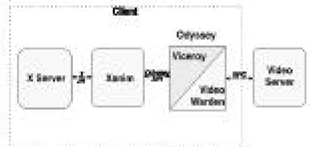
Process	Elapsed Time (s)	Total Energy (J)	Average Power (W)
/usr/odyssey/bin/xanim	66.57	643.17	9.66
/usr/X11R6/bin/X	35.72	331.58	9.28
[illegible] (kernel)	50.88	328.71	6.46
Interrupts-WaveLAN	18.62	165.88	8.91
/usr/odyssey/bin/odyssey	12.19	123.40	10.12
Total	183.99	1592.75	8.66

Energy Usage Detail for process Interrupts-WaveLAN

Kernel-level procedures:

Procedure	Elapsed Time (s)	Total Energy (J)	Average Power (W)
_xferDMAbuffer	16.66	147.38	8.85
_pwlread	0.30	2.90	9.65
_pwlget	0.30	2.68	8.93
_pwlintr	0.24	2.31	9.62

Case Study



Video application
original 12.1MB

- Step 1
  lossy compression
  B: 7MB, C: 2.8MB
- Step 2: display size reduced from 320x240 to 160x120
  A_{small}: 4.9MB, C_{small}: 1MB
- Step 3: WaveLAN put into standby mode when not used
- Step 4: Disk powered off