

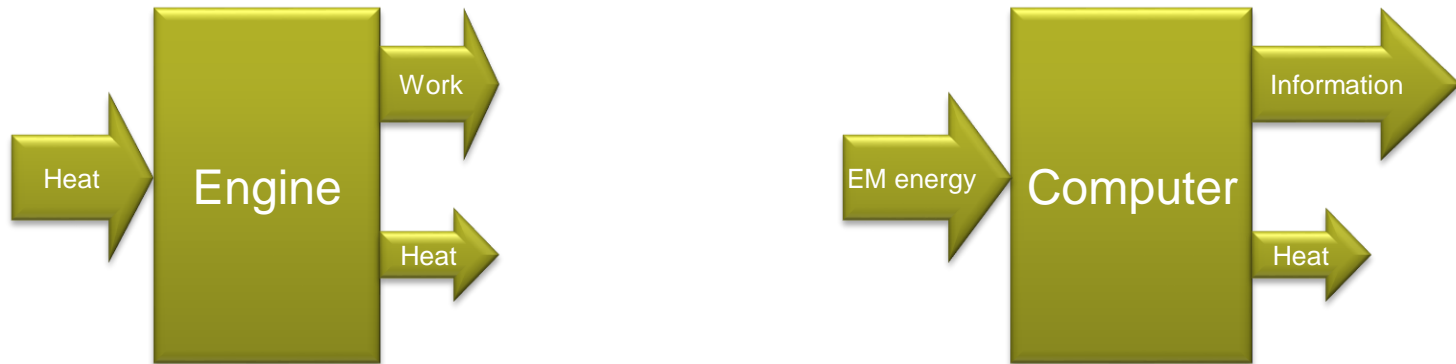
Microsoft® Research

Faculty Summit

10
YEAR ANNIVERSARY

Energy Efficient Computing:

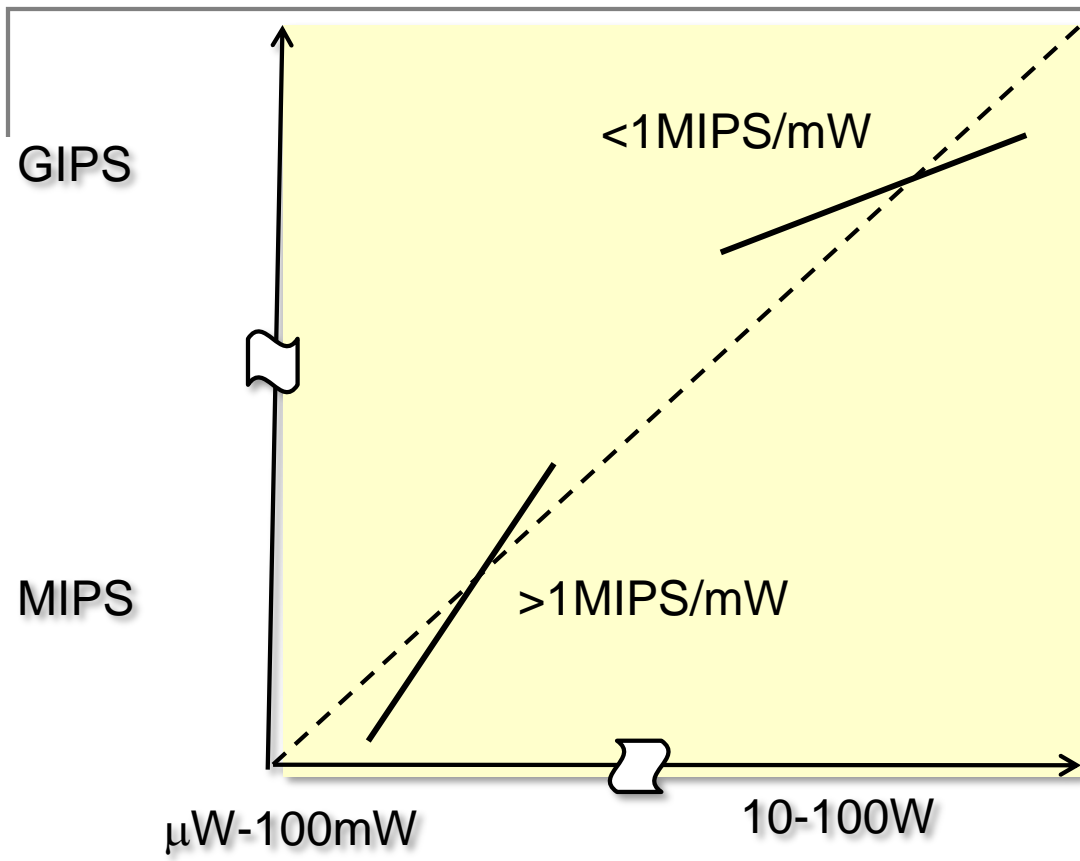
3 observations and 3 lessons from
embedded systems



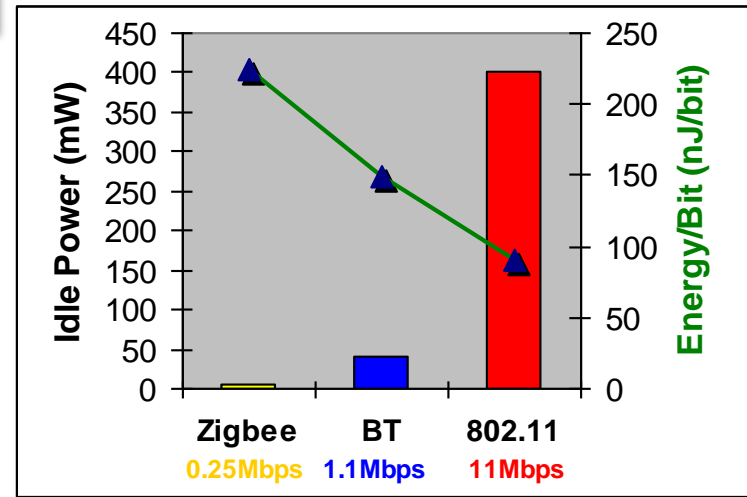
Rajesh Gupta, UC San Diego

<http://mesl.ucsd.edu>

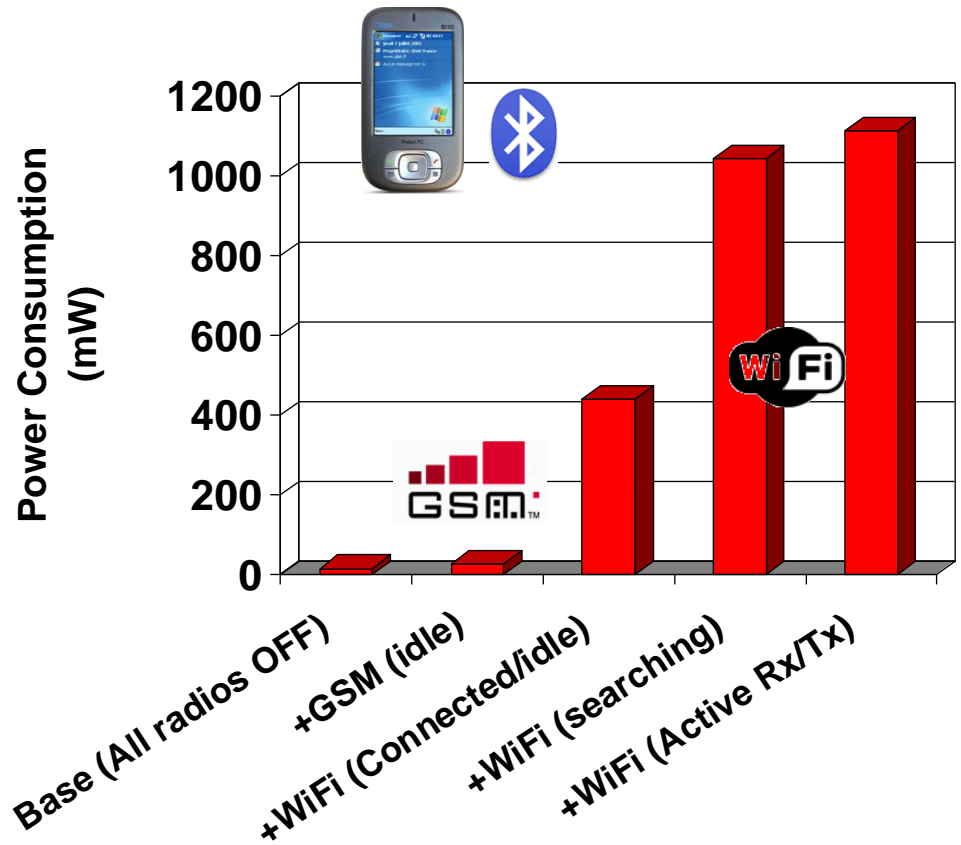
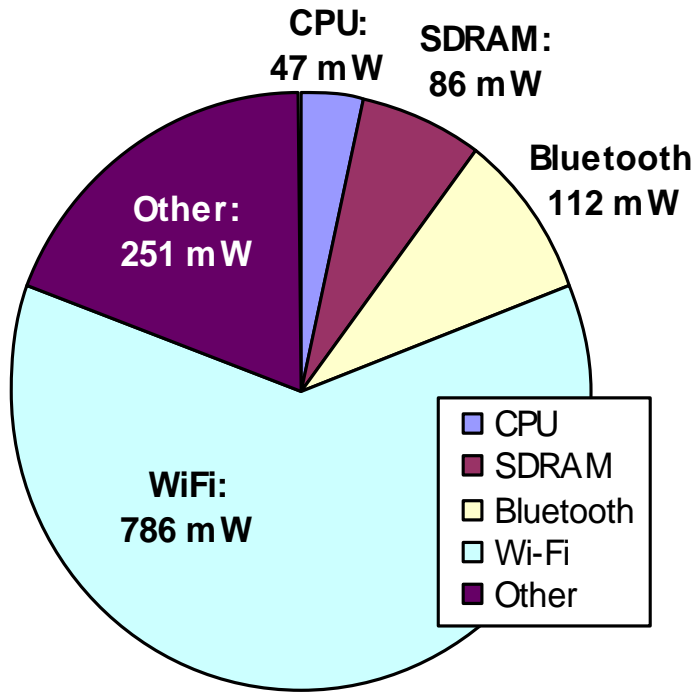
Microsoft, July 2009



Large Differences between components



O1: Heterogeneity is a fact of life.



Power breakdown for a *fully connected* mobile device in *idle* mode, with LCD screen and backlight turned off.

6-10X variation in power use is common.

- Cellular **voice** radio (GSM) highly optimized for low idle power
 - Cingular 2125: GSM radio consumes 38 times less power than Wi-Fi !

O2: Increasing bandwidth of power consumption.

State	Power
Normal Idle State	102.1W
Lowest CPU frequency	97.4W
Disable Multiple cores	93.1W
“Base Power”	93.1W
Suspend state (S3)	1.2W

Desktop PC

Active State : >140W

Idle State : 100W

Sleep state : 1.2W

Hibernate : 1W

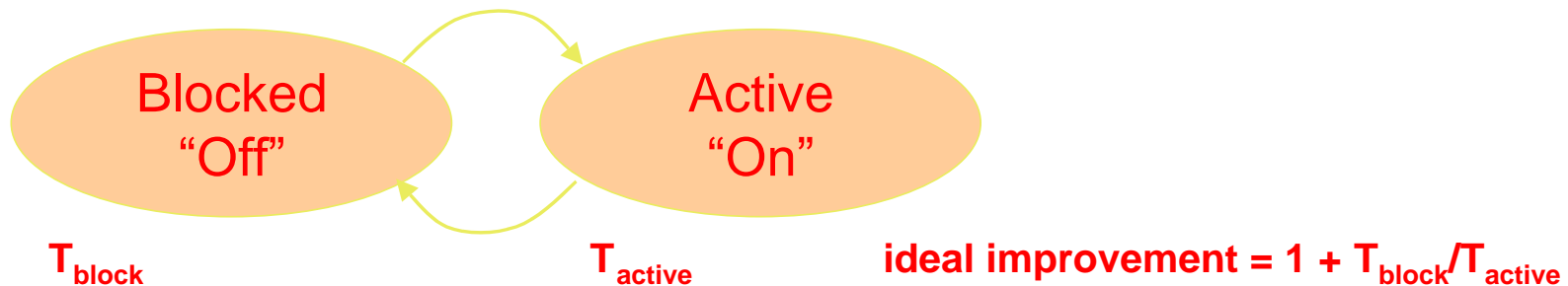


Large Differences
between ON & OFF

O3: Abstraction stacks cost power.

One Lesson in Three Parts

- Exploit Heterogeneity: large differences
 - Between components, power states
- Deliver power when and where needed
 - JIT power delivery
- Match application needs to power availability
 - Differentiated quality of computation

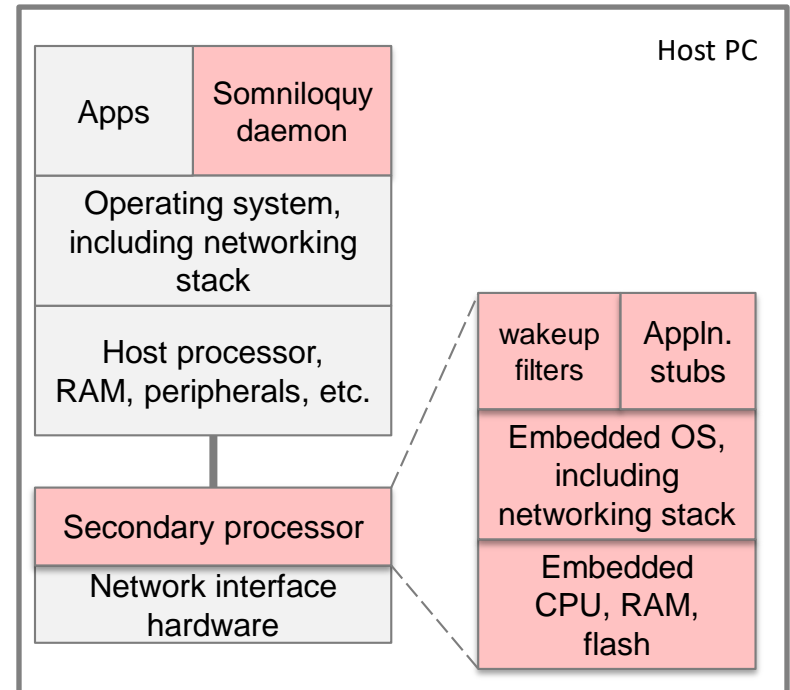
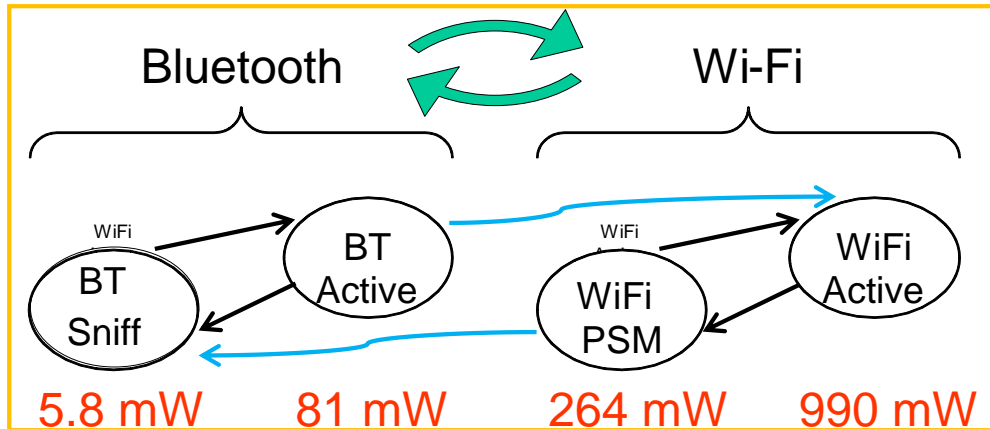


Beyond all low power tricks, duty-cycle keeps on giving

Easier Said Than Done

- Reliability? Availability? Usability?

Paging Radios



Sleep Talking Processors

Exploit heterogeneity to lower power by duty cycling.

Somniloquy

USB Interface (Wake up Host + Status + Debug)

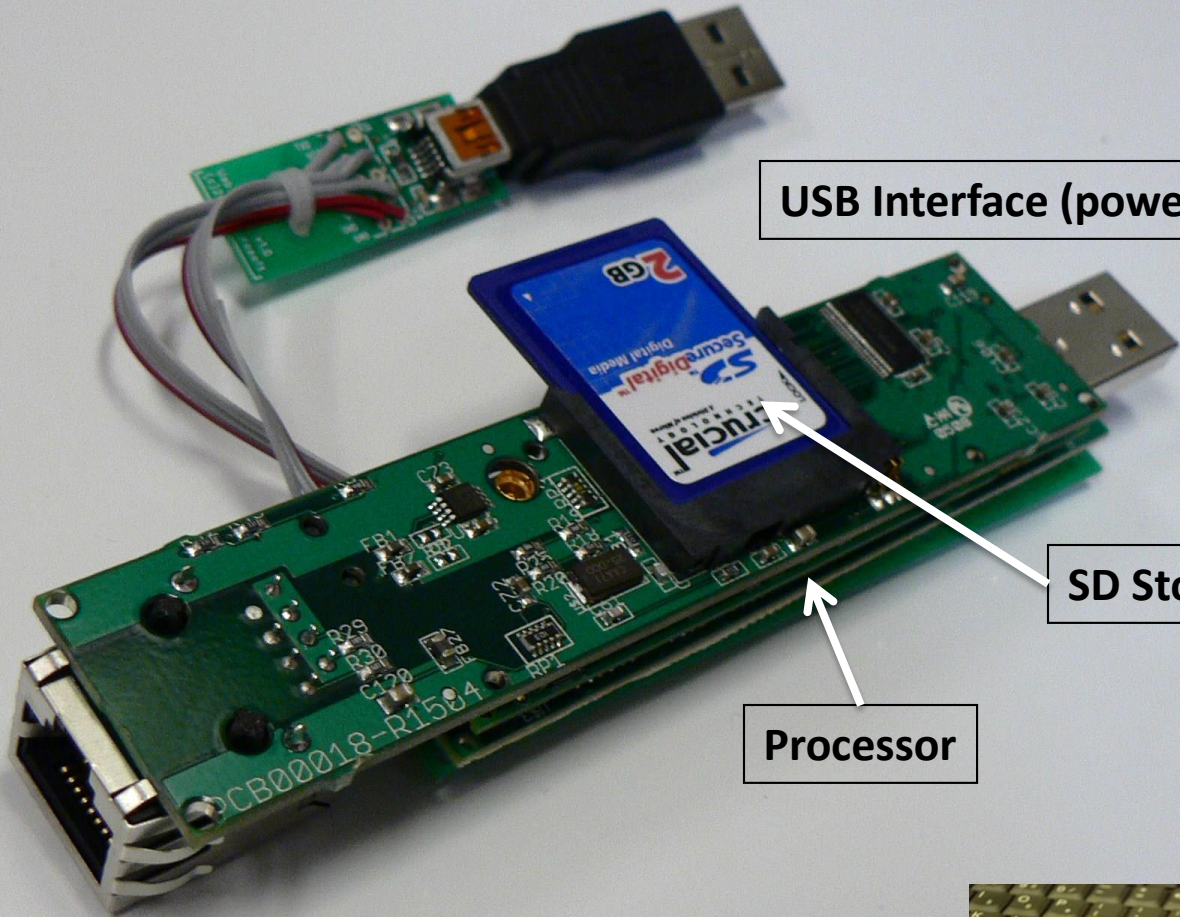
USB Interface (power + USBNet)

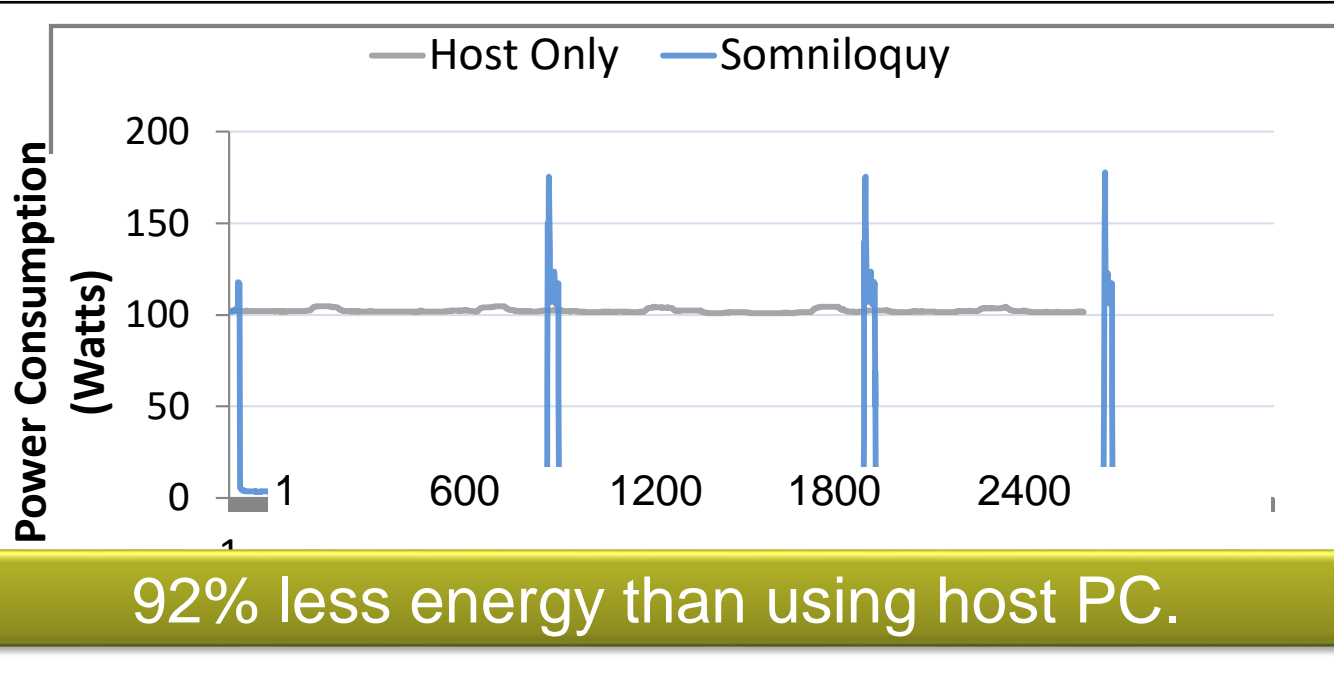
SD Storage

Processor

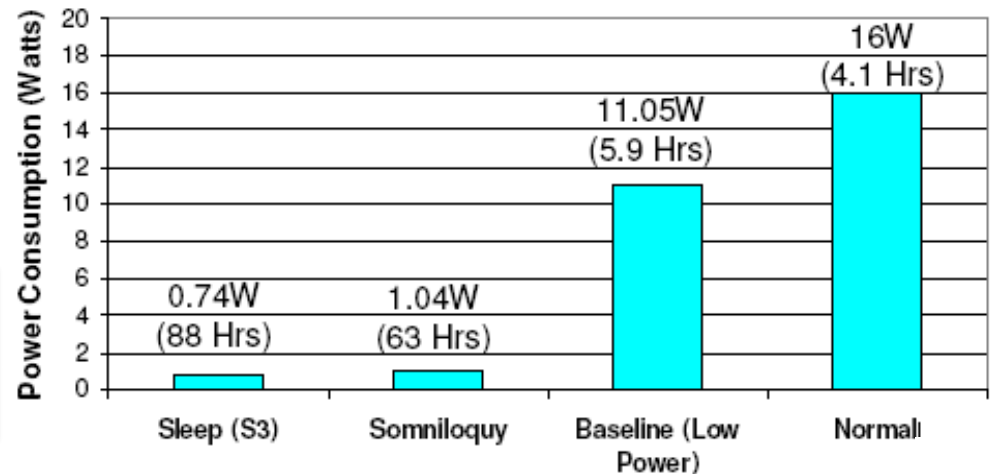
100Mbps Ethernet Interface

In collaboration with Microsoft Research





IBM X60 Power Consumption



Increase battery life from <6 hrs to >60 hrs

From Embedded Systems to HPC: **We KNOW duty cycling is useful.**

Takeaways

- Slowdown or low power design ultimately reaches a limit
- Duty-cycling keeps on giving
 - But causes non-trivial problems in availability, usability, reliability
- Challenge for the community
 - Algorithms: what are the right combination of slowdown and shutdown strategies?
 - Architectures: what is the right organization of components for maximal duty cycling?

“Future lies in system architectures built for aggressive duty-cycling”

“Koala Class” Computing



**“Size Your Brain Power,
Storage, and Sleep Cycles to
Your Problem,” Tom DeFanti**
