Power Management in Ubiquitous Computing

Andreas Weißel

Department of Computer Science 4 (Operating Systems)

University of Erlangen

Martensstr. 1, 91058 Erlangen, Germany

weissel@cs.fau.de

Power Management in Ubiquitous Computing

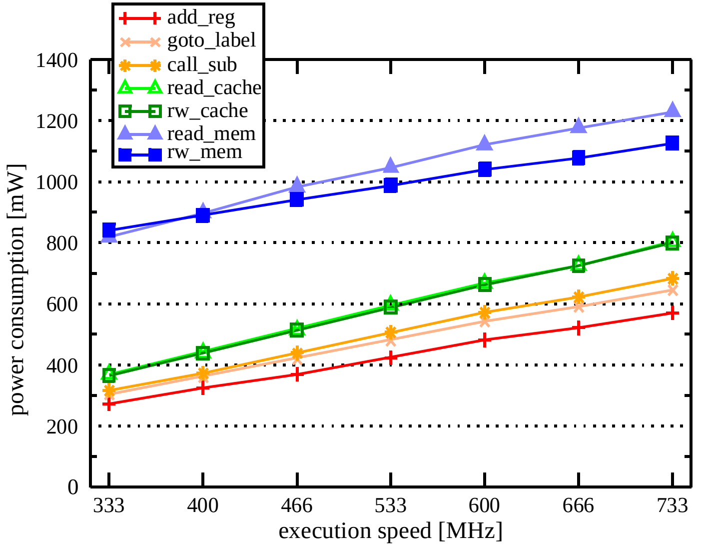
- Why worry about power consumption?
  - Battery-powered sensor nodes:
    - Battery lifetime
    - Weight and size (proportional to dimension of battery)
    - Heat (compact design, no fan available)
- Goals of power management
  - Increasing the battery lifetime or "work per battery lifetime"
  - Guarantee of a pre-defined stand-by and active time

→ The whole system architecture affects power consumption:
power source, <u>operating system</u>, applications, network design

→ "Process Cruise Control": Event-driven frequency/voltage scaling

Clock Scaling: Power Characterization

- Correlation between core clock frequency and power consumption
  Example: Intel XScale (333 MHz – 733 MHz)

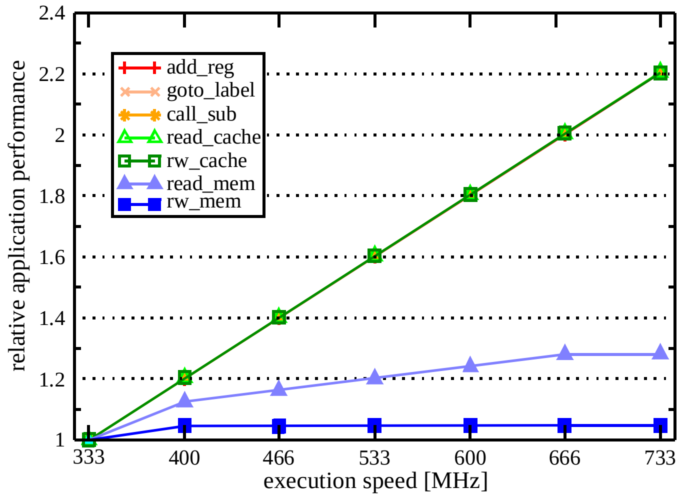


➜The higher the clock frequency, the higher the power consumption

➜Memory-intensive applications consume significantly more energy than CPU/Cache-intensive applications.

Clock Scaling: Performance Characterization

- Correlation between core clock frequency and performance



➜Memory-intensive applications don’t benefit from higher execution speed;

➜can be scheduled at lower speed without losing performance.

Process Cruise Control: Event Counters

- Embedded event counters register HW activations of different kinds (e.g. instructions, cache misses, memory operations, ...).

  ➜Let the CPU count the memory requests.

- At runtime, monitor the rate of memory requests and scale the clock frequency accordingly.

➜ Savings of up to 40% while keeping a limit on the decrease of performance.