Information Services for Energy-Aware Policies

Andreas Weissel
weissel@cs.fau.de

Department of Computer Sciences IV
Distributed Systems and Operating Systems
Friedrich-Alexander University of Erlangen-Nuremberg

http://www4.cs.fau.de

Motivation

- Mobile, battery-powered devices
  - energy consumption determines service time, size and weight of batteries
- Power consumption of system components
  - low-power operating modes (e.g., stop spindle motor of hard disk)
    - additional delays when device is accessed
  - reduced speed of operation (e.g., CPU frequency/voltage scaling)

➔ Energy savings at the cost of delays, reduced speed

Motivation

- Impact of power management on performance?
  - response time, speed, quality-of-service ...
  - e.g. low-power mode of wireless network card: delays when receiving network packets
  - playback of audio stream: no difference in quality
  - list entries of NFS-mounted directory: 16-32x slowdown
- Application-specific performance demands
  - e.g., text editor: process single keystrokes within the perception threshold (50–100ms)
  - web browser: delays of hundreds of milliseconds o.k.

➔ Application-specific trade-off between energy savings and performance

PhD Thesis

- System Services for energy-aware policies
  - infrastructure for adaptive, application-specific power management
- Three approaches
  1. allow energy-aware applications to attribute **system calls** with information on performance demands
  2. run-time classification of applications and their **user-specified** performance demands
  3. automatic **quantification** of the performance of interactive applications (work-in-progress)
- Operating System as mediator
  - knows of applications and there characteristics
  - controls system components (low-power modes)

1. Cooperative I/O

- Extended interface to the OS
  - energy-aware applications can attribute I/O operations with information on performance demands

```
read_coop(int fd, void *buf, size_t count,
          int time-out, int abort);
```

  - ➔ flexible timing of I/O operations

- The power management policy decides when to serve these requests
  - e.g., cluster accesses to the hard disk
  - ➔ reduce overhead of switching between active & low-power modes

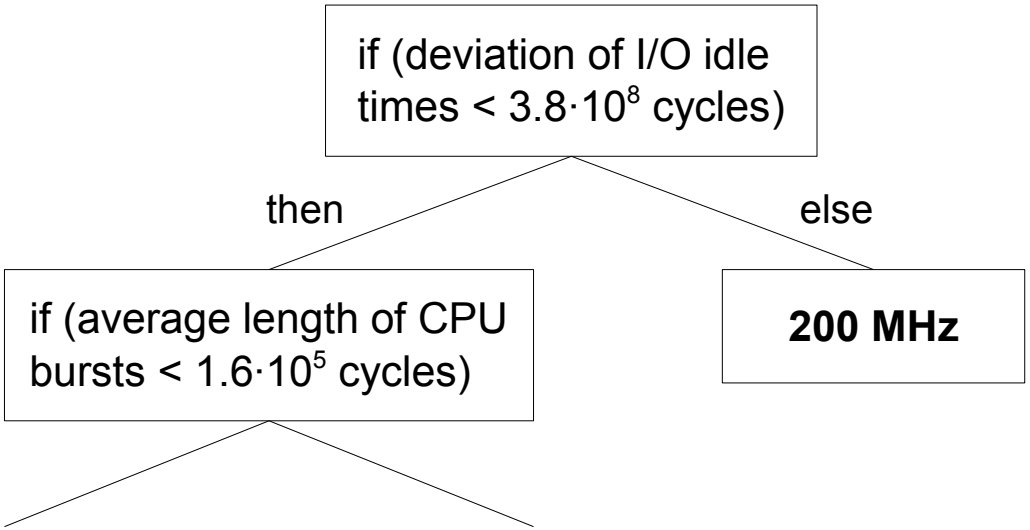
- But: applications have to be rewritten

2. Classifying Performance Demands

- Allow the user to specify performance requirements for application runs
  - ➔ optimal power management settings for each application (operating modes of hard disk & network card, CPU speed)
- During run-time, the OS monitors performance-related events and data on resource consumption
  - e.g., CPU bursts, idle times, periodicity of computation, time between I/O operations, ...
  - averages & deviations

2. Classifying Performance Demands

- Supervised learning
  - training algorithm determines characteristic properties of each application, generates **classification tree**



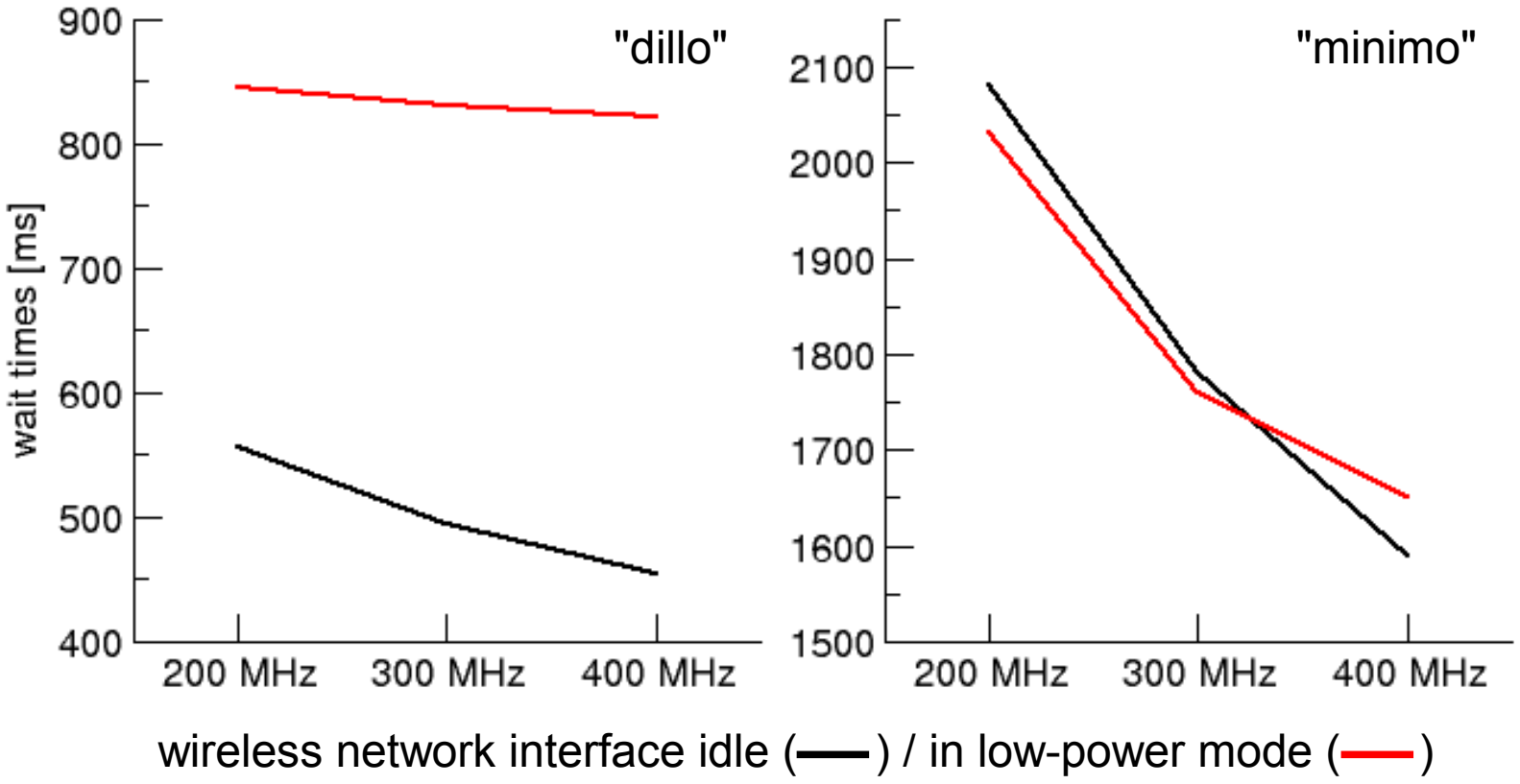
- Run-time classification of applications
  - derive performance requirements, i.e. appropriate operating modes

3. Quantifying Application Performance

- Performance of interactive applications
  - **wait time**: time to respond to a request from the user
  - monitor scheduler, I/O operations, inter-process communication
- Is wait time acceptable for a specific application?
  - e.g., compare to perception threshold (< 100ms)
  - but: applications (on iPAQ) with much higher wait times
- Quantitative analysis
  - monitor wait times and their changes over different power management settings (e.g. CPU clock frequencies)
  - determine dependencies between operating modes of different system components

3. Quantifying Application Performance

- Wait times of two web browsers on iPAQ (Linux)



wireless network interface idle (——) / in low-power mode (——)

➔ Feedback on impact of power management settings

Conclusion

- Trade-off between energy savings and performance

- System services for application-specific, adaptive power management policies
  1. attribute **system calls** with information on performance demands
  2. let the **user** specify application-specific performance requirements
  3. determine **impact of power management** on application performance