Power-Sleuth: A Tool for Investigating your Program's Power Behavior

Vasileios Spiliopoulos, Andreas Sembrant and Stefanos Kaxiras

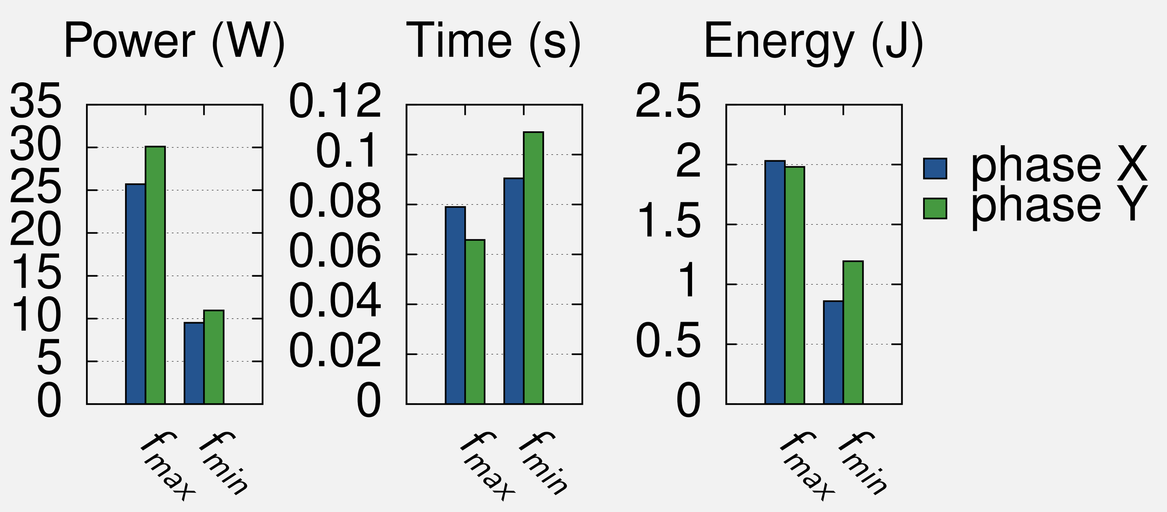
UPMARC — Uppsala Programming for Multicore Architectures Research Center

UPPSALA UNIVERSITET

1. Power Profiling in different frequencies

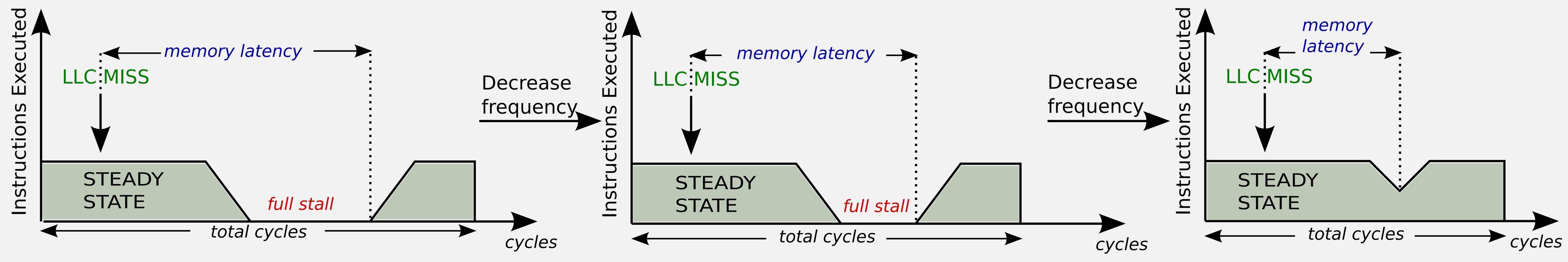
Power characterization:

- What is the execution time of each phase?
- What is the power of each phase?
- How is execution time/power affected by frequency scaling?



Power-Sleuth characterizes application phases using performance and power models

Analytical DVFS performance model



Power-performance counters correlation model

$$P = f \times C_{pred} \times V^2 + P_{static}$$

$$C_{pred} = \sum_{k=0}^{5} \frac{param_k \times event_k}{cycles} + param_6$$

2. Profile applications at f_{max} and predict execution time and power at f_{max} and f_{min}

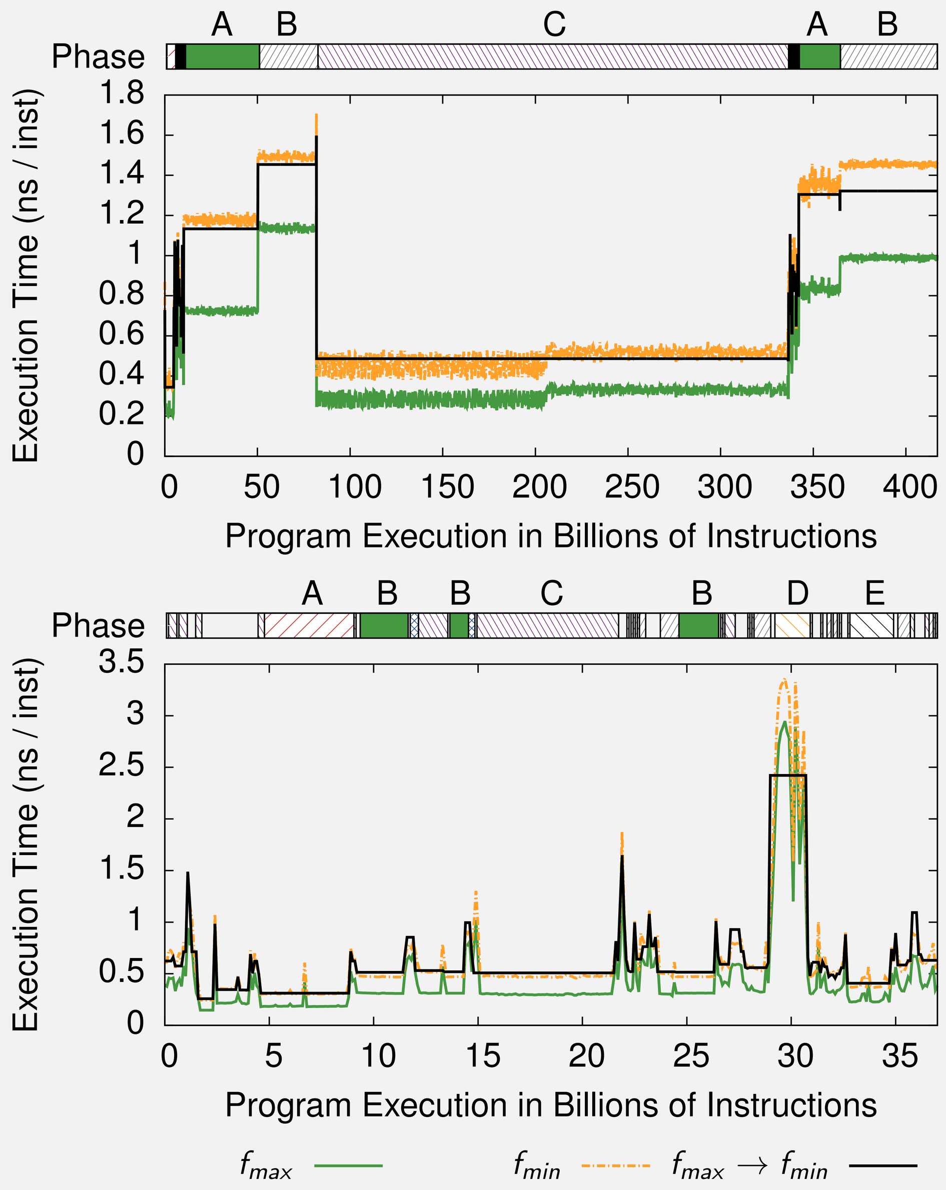
How does Power-Sleuth work?

1. **Runs** an application once
2. **Detects** phases using ScarPhase library
3. **Measures** performance counters
4. **Estimates** execution time and power consumption under any frequency without re-running the application

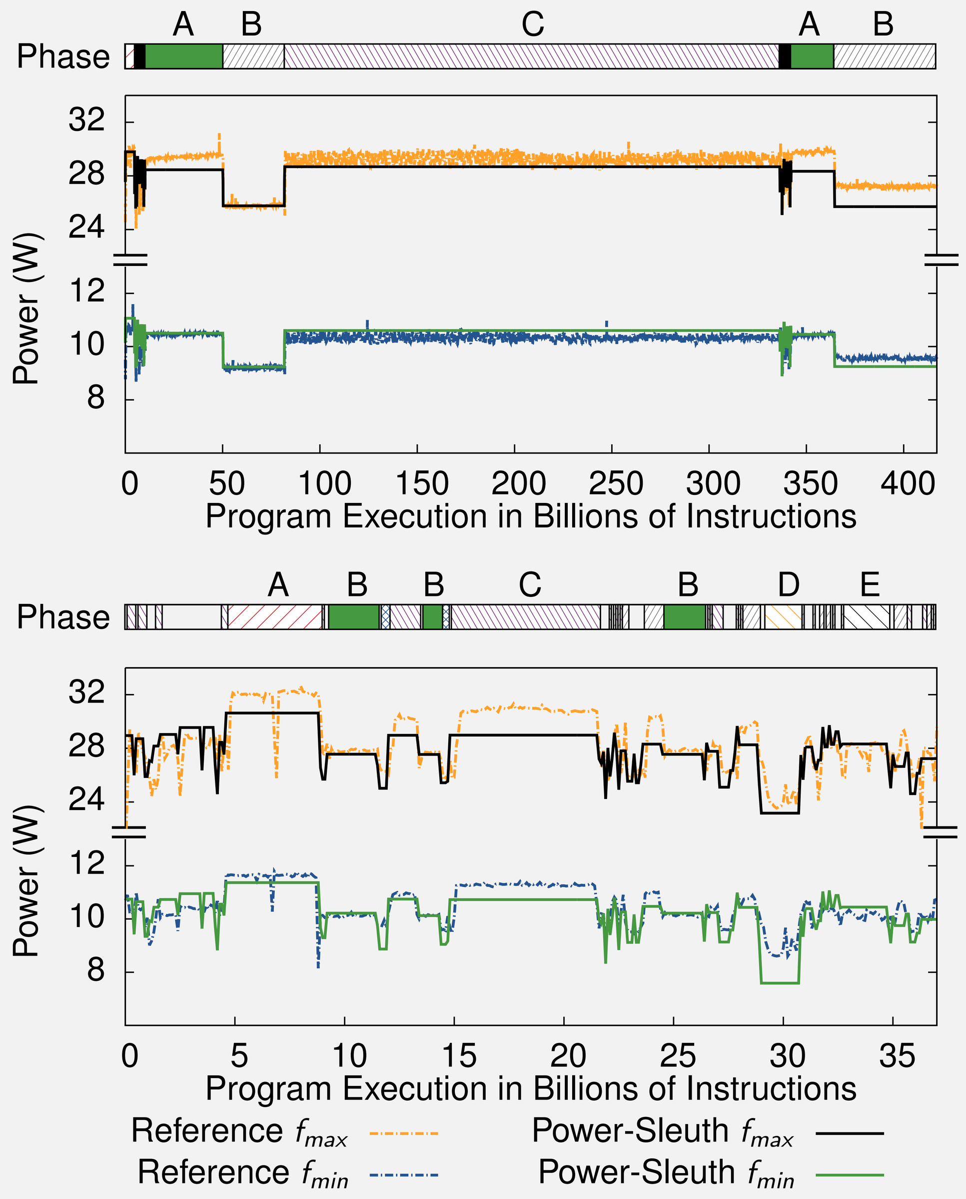
No power measurements required!

Power-Sleuth delivers

- **Accurate prediction:** less than 4% average error for both execution time and power prediction (running SPEC2006)
- **Per phase analysis:** instead of total energy and execution time, Power-Sleuth predicts for each program phase separately



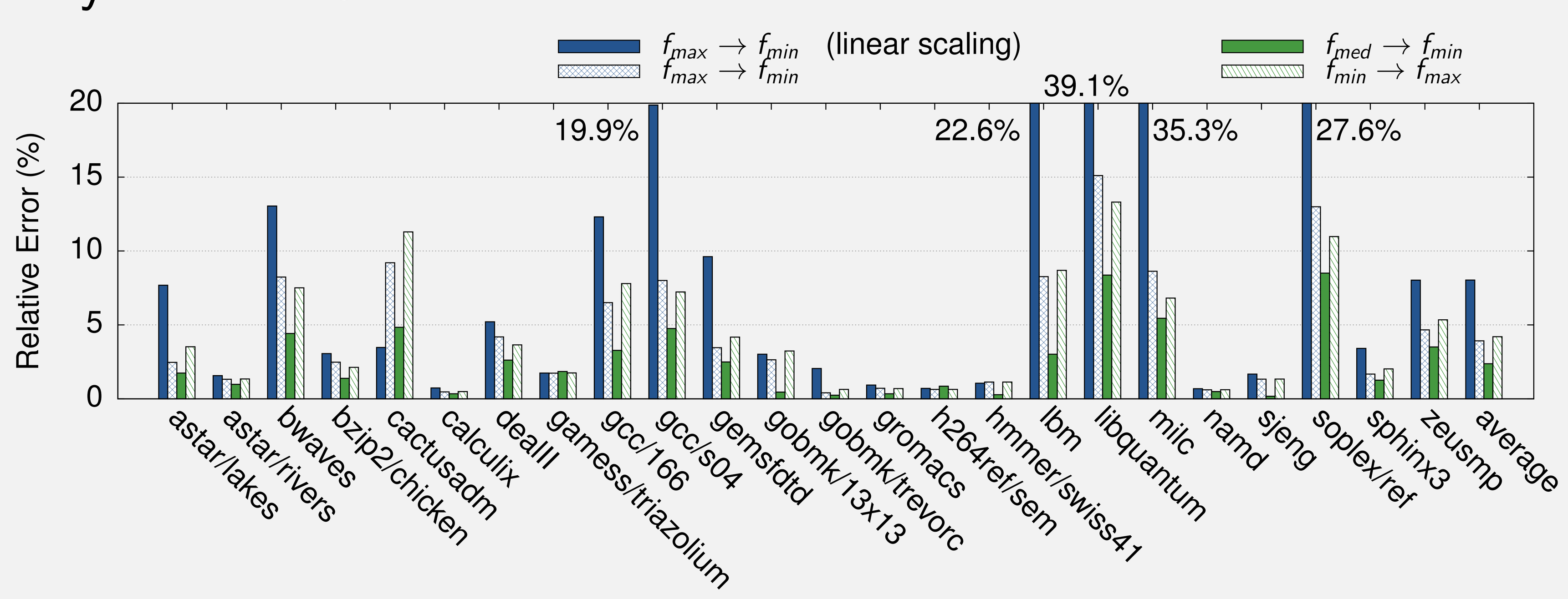
Execution time (measured and predicted) for astar/lakes (top graph) and gcc/166(bottom graph)



Power estimation in f_{max} and f_{min} for astar/lakes (top graph) and gcc/166(bottom graph)

3. Power-Sleuth accuracy

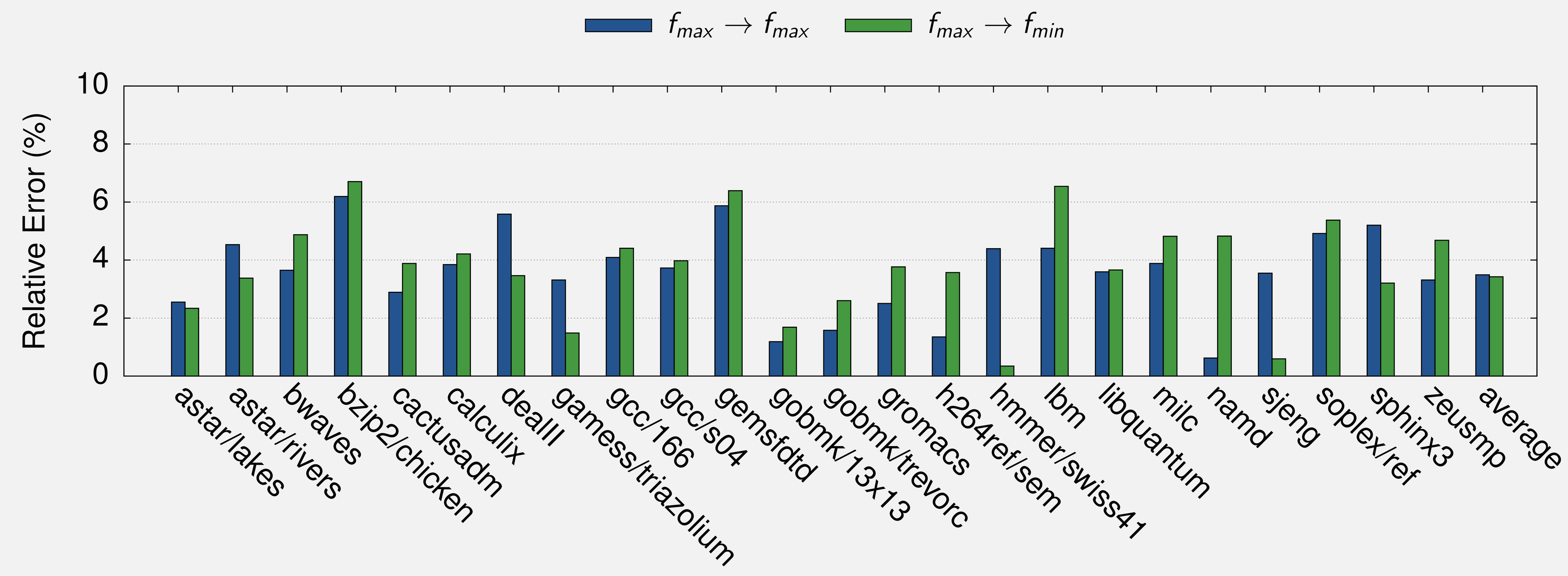
Execution time **does not** scale linearly with frequency: need the analytical DVFS model



Execution time prediction error for different profiling and target frequencies.

No power measurement infrastructure required

- We estimate power for both max and min frequency

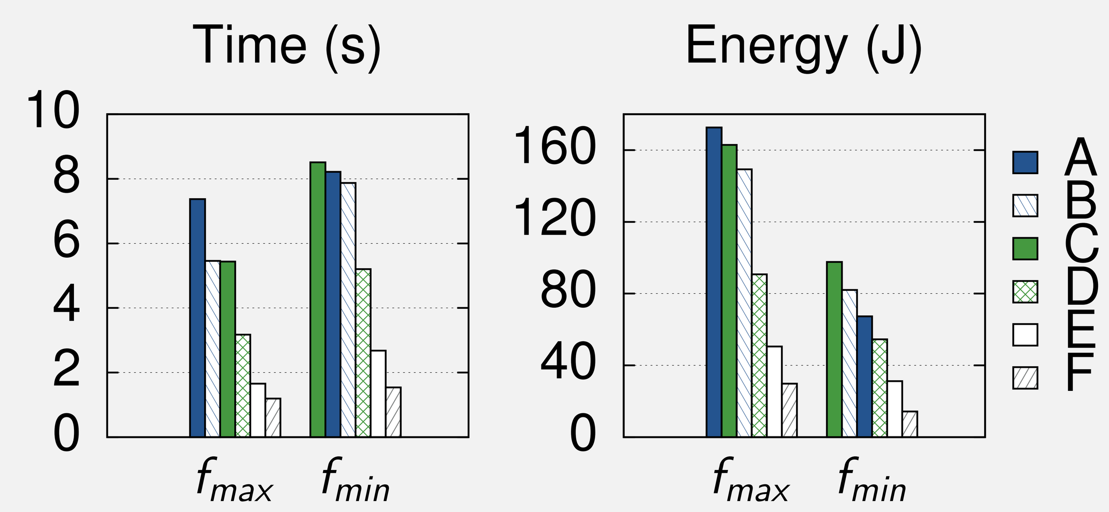


Power prediction error. Profiling is performed under maximum frequency.

4. Using Power-Sleuth

Optimizing for power

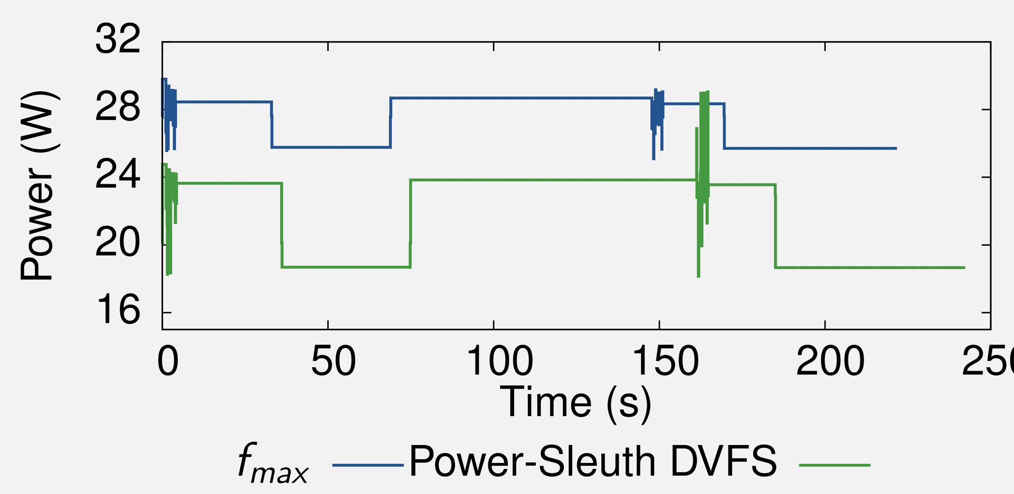
- Accurately estimating execution time and power of every program phase under any voltage-frequency setting
- Linking phases back to the source code



Execution time and energy consumed under max/min frequency for the phases of gcc/166.

Optimal DVFS scheduling

- Performance constraints for specific phases
- Optimizing power-efficiency (e.g. minimum EDP)



Minimizing EDP under the constraint that performance is not penalized more than 10%. The appropriate DVFS schedule is provided by Power-Sleuth.