

MeterPU: A Generic Measurement Abstraction API

Enabling Energy-tuned Skeleton Backend Selection

Lu Li and Christoph Kessler
IDA, Linköping University

Objectives

- ▶ Generic measurement abstraction, simplest API, very low overhead.
- ▶ Measuring time, energy, ..., enabling optimizations.
- ▶ Easy integration with legacy tuning software using empirical models.
- ▶ Easy retargeting of optimization goals.

MeterPU

- ▶ Open source C++ template library.
- ▶ Measure energy etc..., in a similar way as time.
- ▶ Measurement abstraction on top of native measurement API.
- ▶ Supported metrics (extensible plugins):
 - ▷ Timers on CPU, GPU.
 - ▷ Energy meter on CPU, DRAM, GPU, and System.
 - ▷ ...
- ▶ Portable to different heterogeneous architectures.

MeterPU API

- ▶ `start()`: mark the start of a measurement phase/period.
- ▶ `stop()`: mark the end of a measurement phase/period.
- ▶ `calc()`: calculate the metric value between `start()` and `stop()`.
- ▶ `get_value()`: return the value by `calc()`.

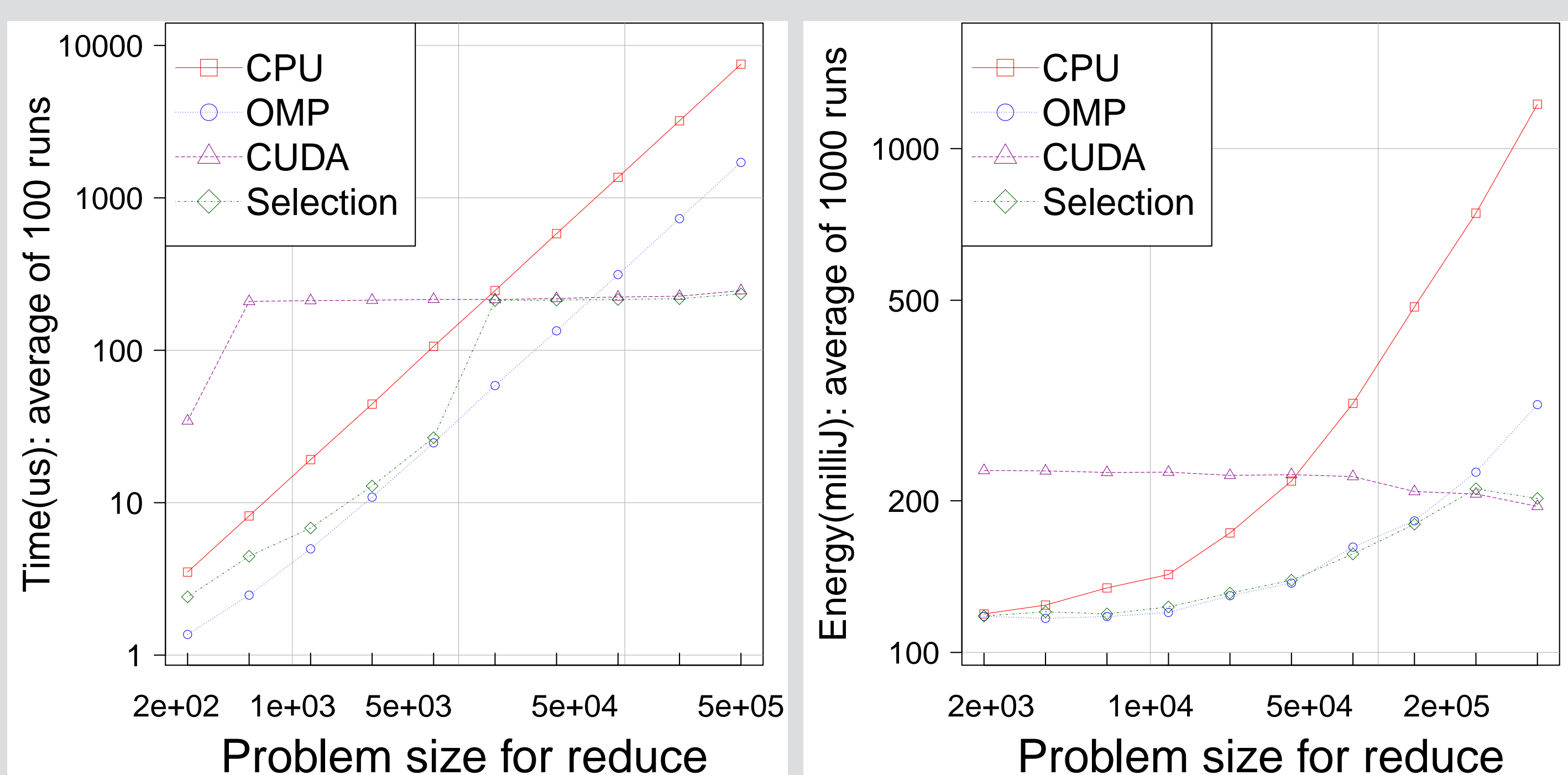
An Example MeterPU Application

```
#include <MeterPU.h>
int main()
{
    using namespace MeterPU;
    //Different meters can be defined here
    //E.g. Meter<NVML_Energy>> meter; //GPU energy meter
    Meter<CPU_Time> meter;
    meter.start();
    //Do sth here
    sleep(2);
    meter.stop();
    meter.calc();
    std::cout<<"Time consumed is: " <<meter.get_value()
              <<" microseconds."<<std::endl;
}
```

Application Of A Legacy Autotuning Software: SkePU

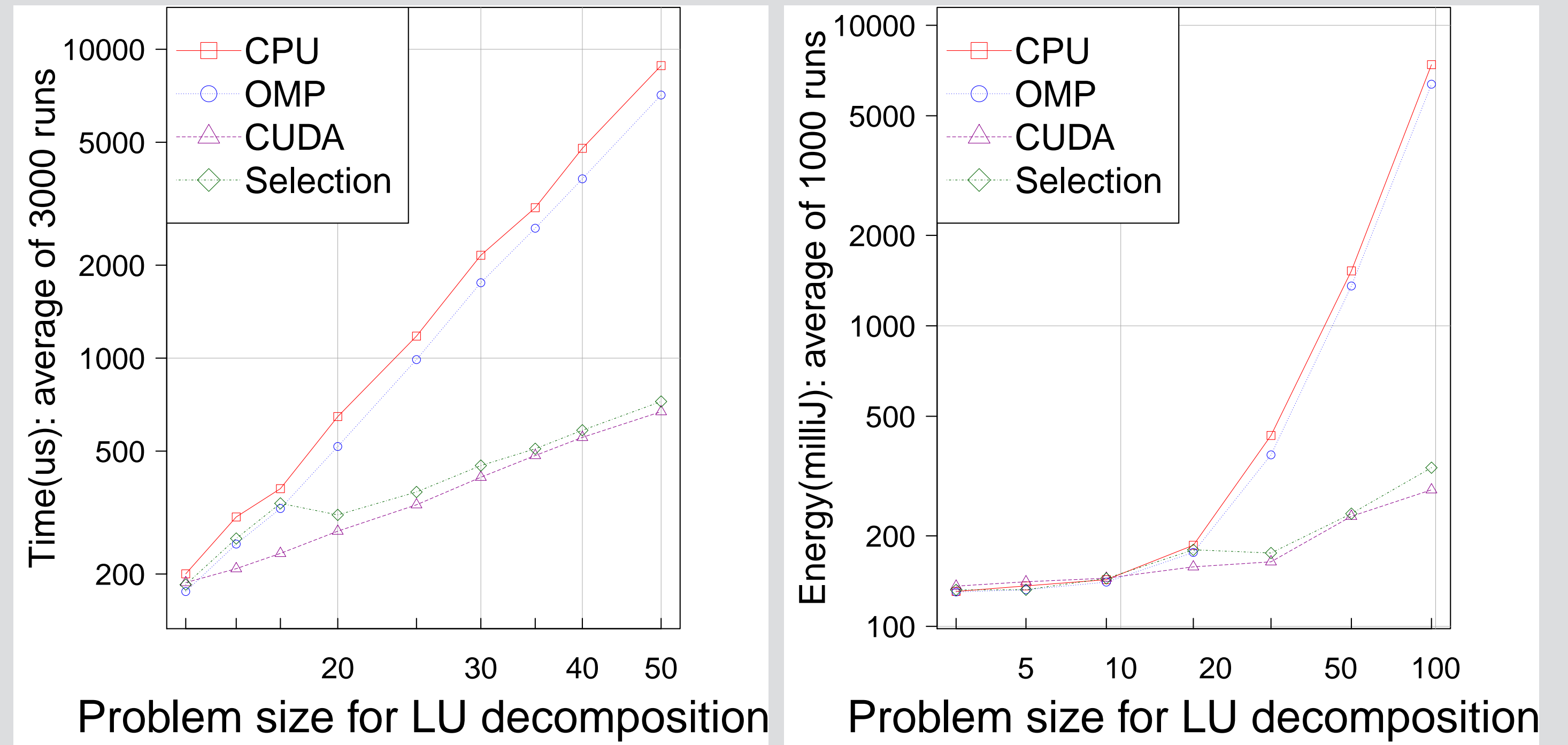
- ▶ SkePU: state-of-art skeleton programming framework on heterogeneous systems, with automatically tunable backend selection.
- ▶ Originally tuned for time.
- ▶ Easily targeted to energy optimization by integration with MeterPU.

Results: Tuning Individual Skeleton



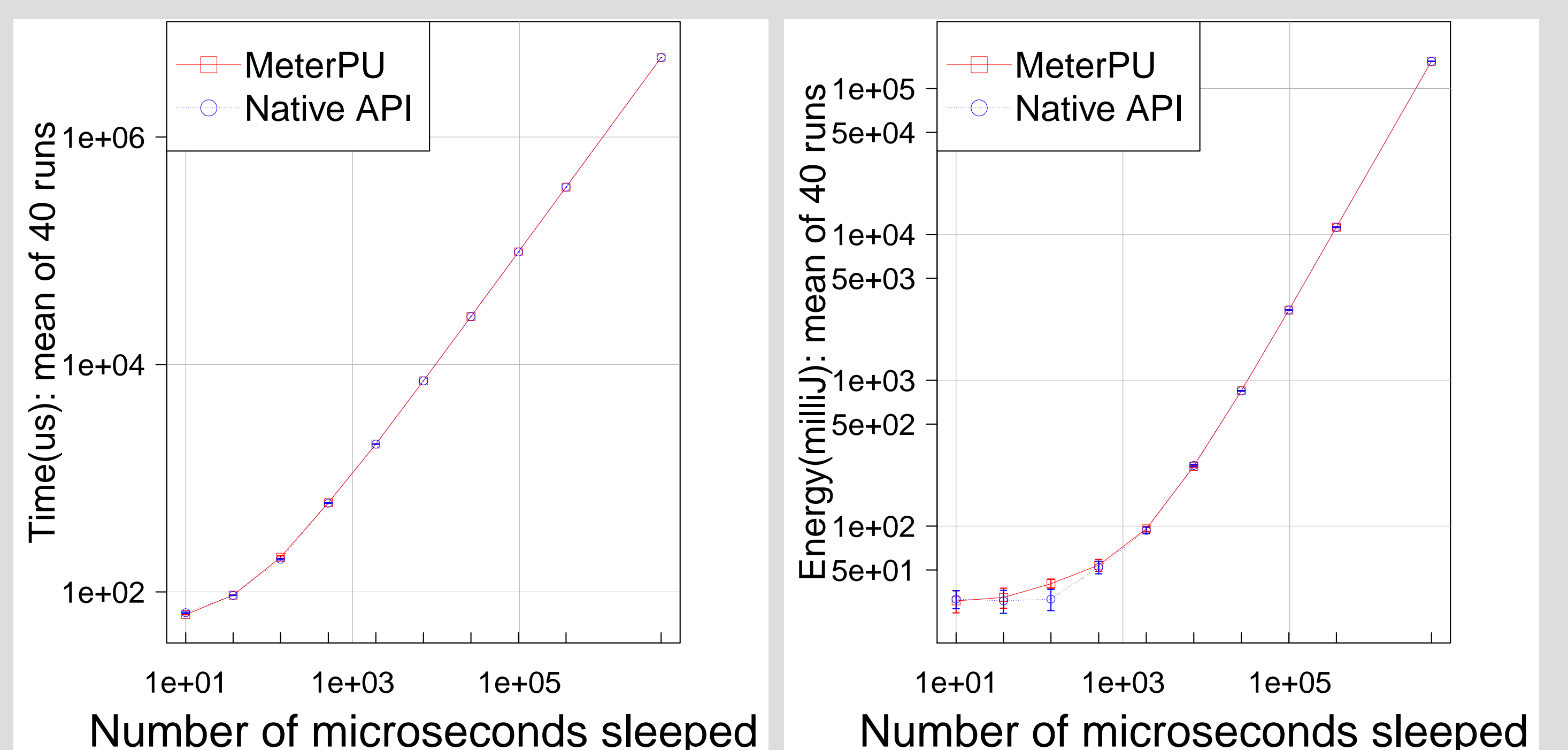
(a) Time tuning. (b) Energy tuning.
Figure 1: Results obtained on SkePU Reduce Skeletons with MeterPU

Results: Tuning LU decomposition



(a) Time tuning (b) Energy tuning
Figure 2: Time and energy tuning of LU decomposition by SkePU and MeterPU

Results: Overhead



(a) MeterPU time overhead (b) MeterPU energy overhead
Figure 3: Nonobservable MeterPU overhead (only one function call)

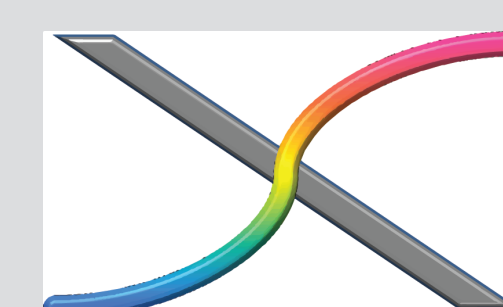
Conclusion

- ▶ MeterPU facilitates easy measurement, switching among different optimization goals, and reuse of legacy empirical tuning frameworks.
- ▶ With MeterPU, SkePU provides the first energy-tunable skeletons for heterogeneous systems.

References

- [1] Lu Li, Christoph Kessler, "MeterPU: A Generic Measurement Abstraction API Enabling Energy-tuned Skeleton Backend Selection.," *Journal of Supercomputing*, pp. 1–16, 2016. doi: 10.1007/s11227-016-1792-x.
- [2] U. Dastgeer, L. Li, and C. Kessler, "Adaptive implementation selection in a skeleton programming library," in *Proc. of the 2013 Biennial Conference on Advanced Parallel Processing Technology (APPT-2013)*, vol. LNCS 8299, pp. 170–183, Springer, Aug. 2013.

Acknowledgments



Contact Information

- ▶ Web: <http://www.ida.liu.se/labs/pelab/meterpu/>
- ▶ Email: lu.li@liu.se, christoph.kessler@liu.se

