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.

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 stuff here
  sleep(2);
  meter.stop();
  meter.calc();
  std::cout << "Time consumed is: " << meter.getvalue()
             << " micro seconds." << 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

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.

Results: Tuning LU decomposition

References

Acknowledgments

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

http://www.ida.liu.se/labs/pelab/meterpu/