Portable Power/Performance Benchmarking and Analysis with WattProf

Amir Farzad, Boyana Norris
University of Oregon
Mohammad Rashti
RNET Technologies, Inc.

Motivation

- Energy efficiency is becoming increasingly important in high-performance computing.
- **US DOE Goal:** To build Exascale machine with **20MW** max power by 2020.
- With current trend on **top500*** it takes 60 years!
- Understanding the power attributes of application components.
- Performance and power/energy of HPC apps.
- Improving power/energy efficiency.

- *http://www.top500.org/

Motivation Cont.

- Hardware and software tools that enable **fine-grained** measurement of power.
- **Fine-Grain:** Synchronize power/energy measurements with application activity.

Our Contribution

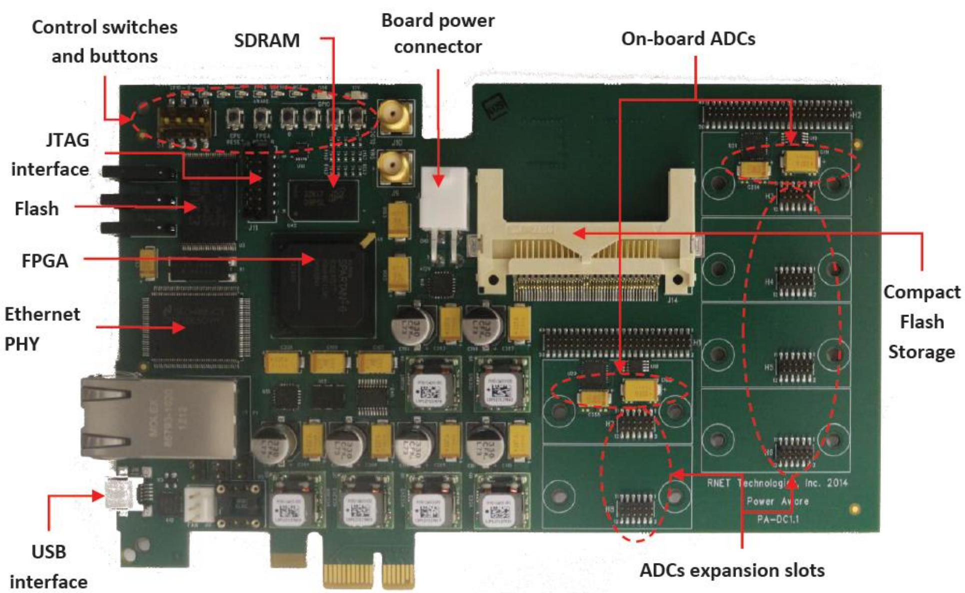
- Use of the new WattProf board [8] to collect fine-grained power and energy measurements.
- Automated source code instrumentation of C/C++ and Fortran codes for collecting function-level power and energy measurements;
- Power and energy analysis and modeling use cases based on this infrastructure.

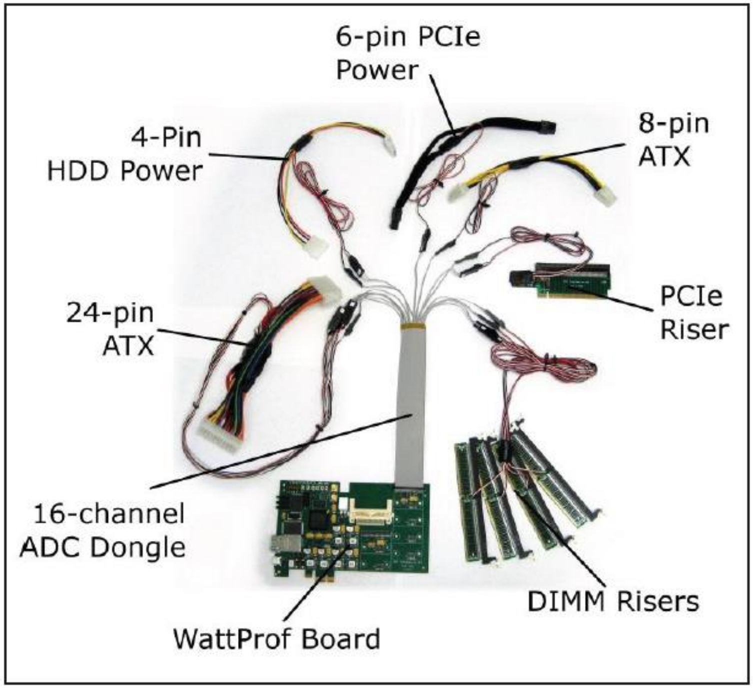
WattProf

- **WattProf (Rnet Tech. Inc.)**,
- a new power monitoring tool that enables **high frequency** (multiple kilohertz) direct power measurement
- Different components:
  - CPU, DRAM, GPU, NIC, PCIe cards, fans, hard drives, SSD

WattProf

- **WattProf (Rnet Tech. Inc.)**,
- more details ref. [8] in the paper
- 4KHz sampling





[8] M. Rashti, G. Sabin, and B. Norris. Power and energy analysis and modeling of high performance computing systems using WattProf. In Proceedings of the 2015 IEEE National Aerospace and Electronics Conference (NAECON), July 2015.

Source Code Instrumentation

- The **WattProf** host API can be used by application developers to measure power or energy consumption.
- The granularity of the information that **WattProf** can gather is similar to performance tools such as **PAPI**, **TAU**, and **HPC toolkit**. But for **power/energy**.
- Performance and power can be correlated for analysis and modeling.

Source Code Instrumentation

- The WattProf host API:
  - Starting and stopping a **measurement window** by calling the corresponding API functions.

- Automatic instrumentation:
  - We developed a tool that instruments the source code for power and energy measurement.
  - Available on GitHub (https://github.com/amirfarzad/opensource)

Source Code Instrumentation

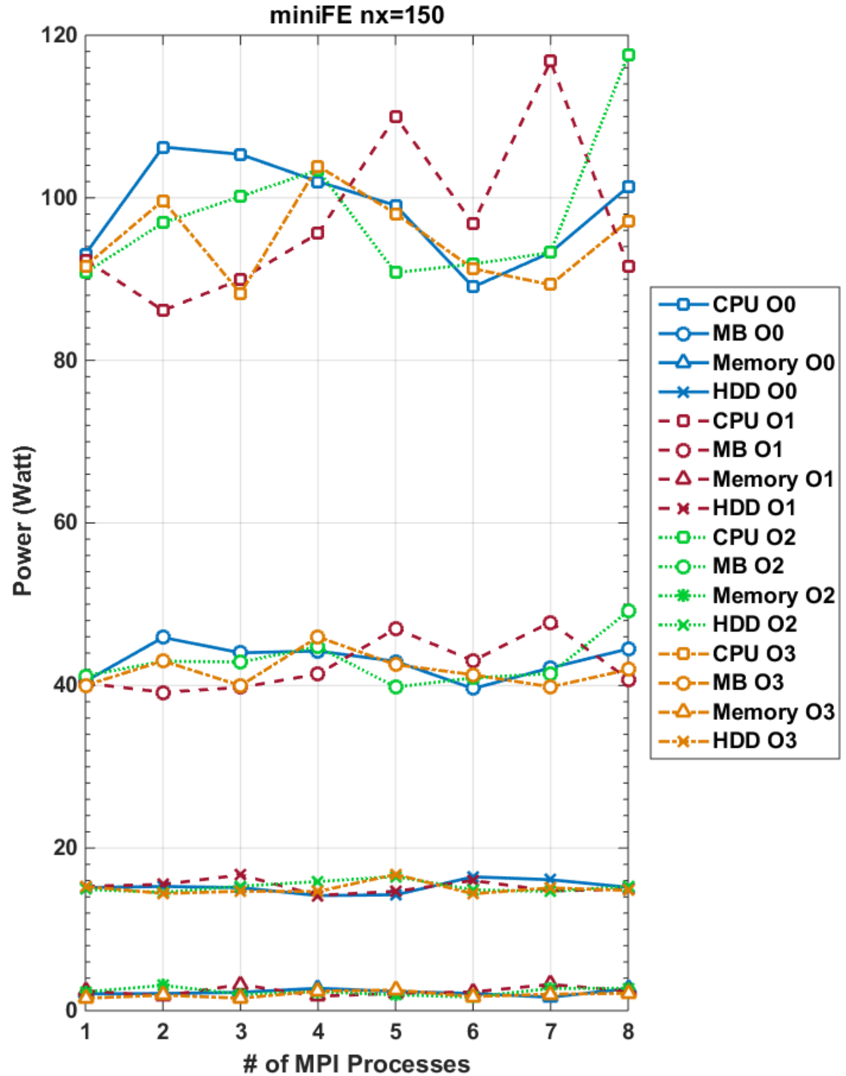
- Embeds the specific routines at the compile time in the target source code.
- works with **C**, **C++** and **Fortran** (GNU and Intel compilers).
- Note that this option does not require any manual changes in the target source code.
- Minimum overhead during measurement time:
  - Most of the post-processing is done before or after a measurement window

Analysis

- Initial evaluation on miniFE proxy app (the Mantevo benchmark suite).
- miniFE
  - Problem size 30x30x30 to 150x150x150
  - MPI processes 1,2,…,8
  - GCC 4.8.2 with optimization levels -O0, -O1, -O2 and -O3
  - Three runs and reporting the average value
- We show how this platform can be effectively used for HPC application

Power

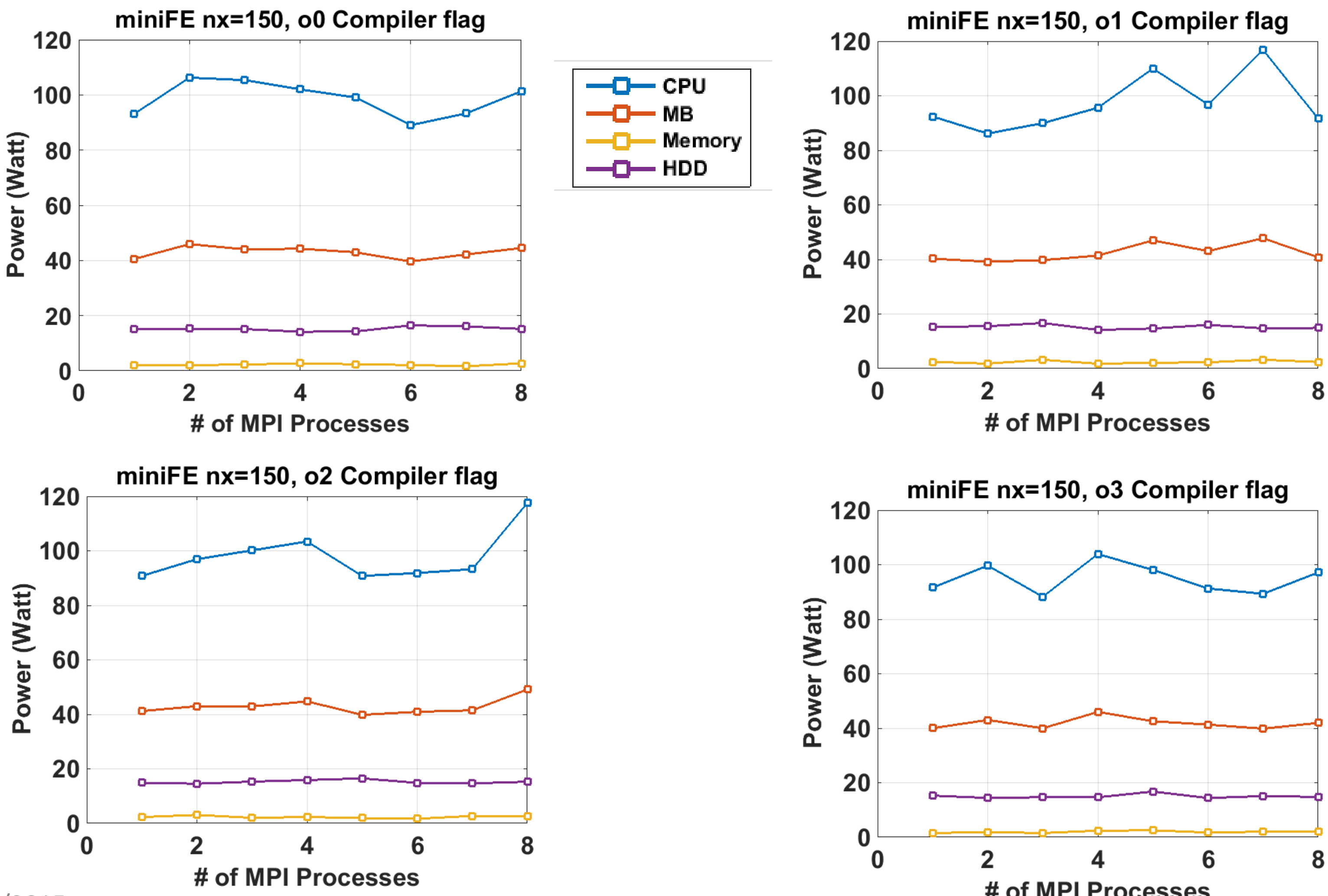
- Power for the problem size nx=150
- Prev. studies[6]:
  - the more aggressive optimization levels (-O3) may increase the power dissipation while they decrease the energy consumption due to shorter runtimes.



[6] J. H. Laros, P. Pokorny, and D. DeBonis. PowerInsight{a Conference (IGCC), 2013 International, pages 1-6, 2013.

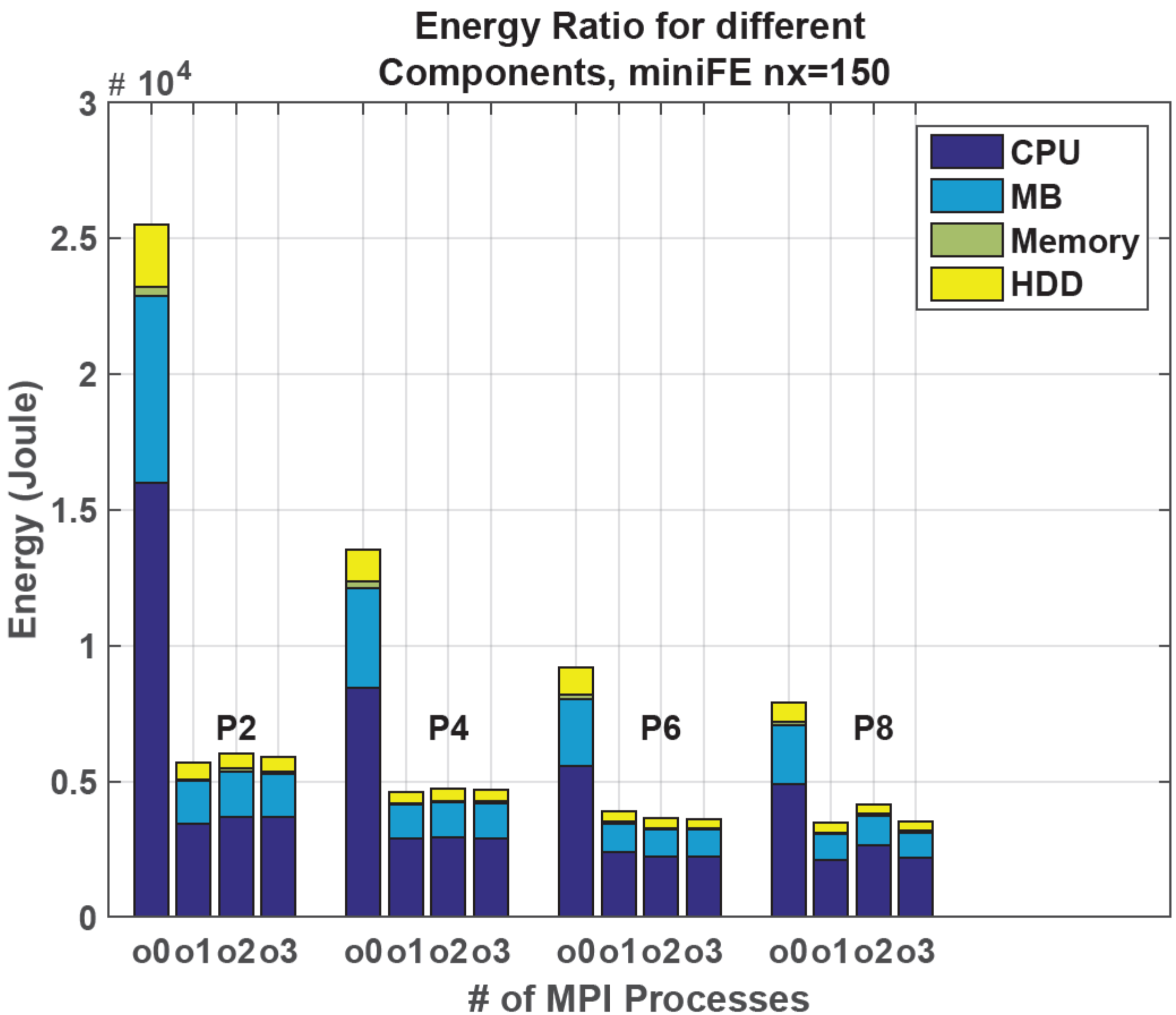
Power, Cont.

- Figs. Separate for O0, O1, O2, O3.



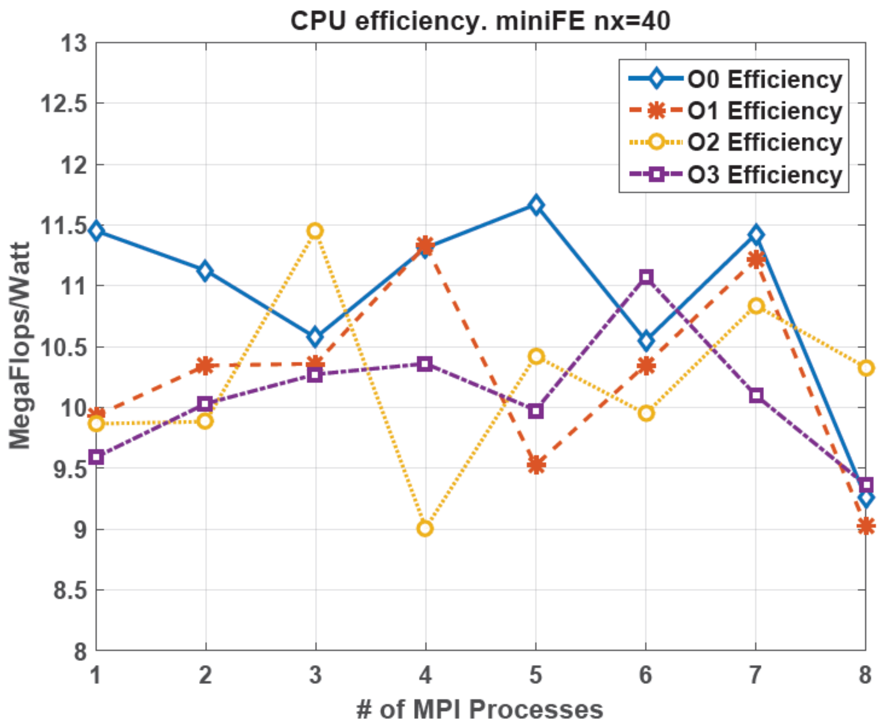
Energy Measurement

- **Compiler Flags:**
- **O0>>**
- **O3<O2**
- **O1?**

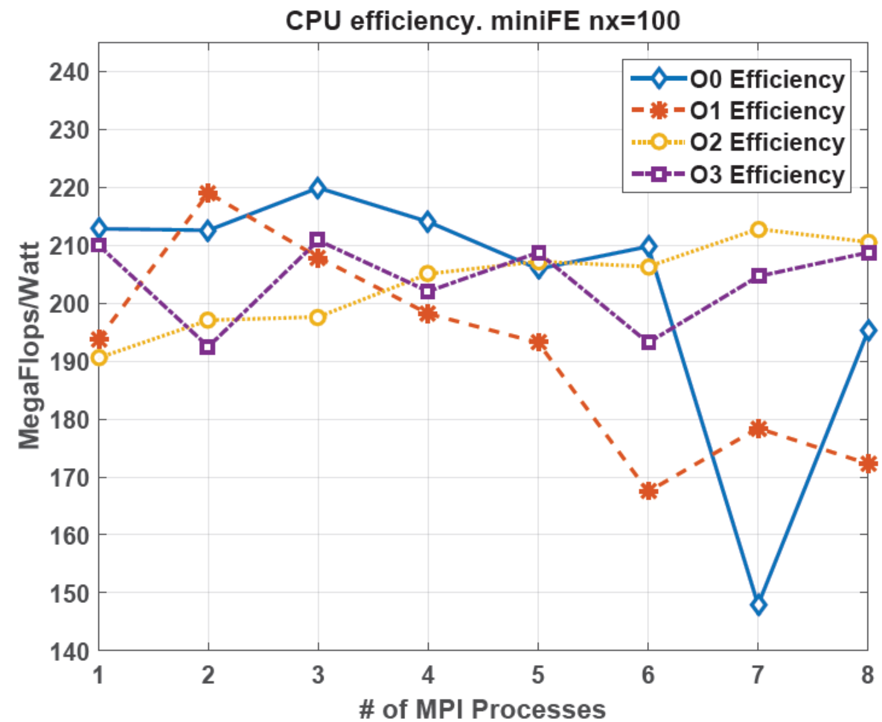


CPU efficiency

- floating-point operations per Watt.
- desirable to maximize the CPU efficiency.



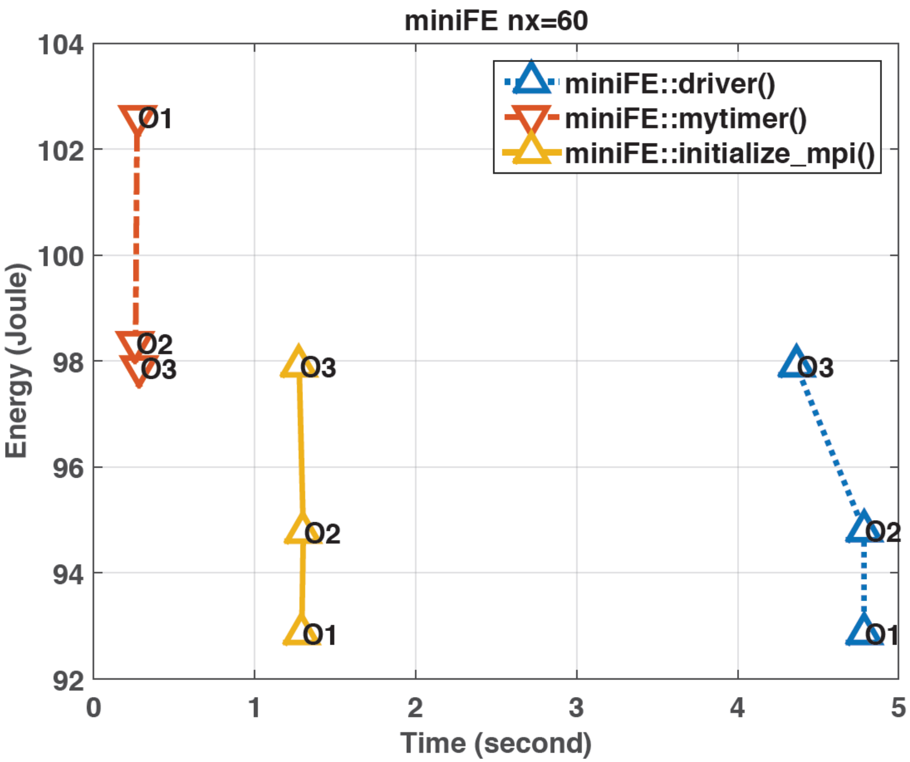
(a) CPU efficiency for problem size 40x40x40.



(b) CPU efficiency for problem size 100x100x100.

Profiling and Optimization

- To demonstrate the ability of WattProf to profile the power of individual functions.
- Fine grain resolution. Can be correlated with hardware performance counters for the same functions

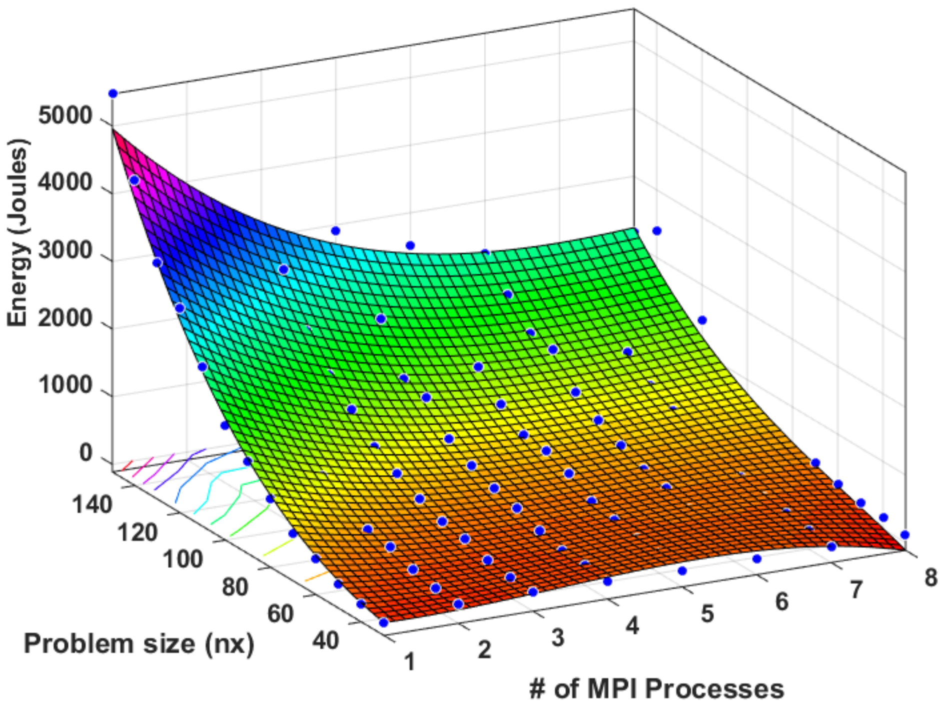


- `miniFE::mytimer()` → **(O1 > O2 > O3),**
- `miniFE::driver()` → **(O1 < O2 < O3),**

Modeling CPU energy

- Modeling for -O3
- MPI p=1,2,…,8.
- Nx=30,40,…,150.

Modeling CPU energy



$$f(x,y) = -141.2 + 68.68x + 6.387y + 31.31x^2 - 5.443xy + 0.07479y^2 - 5.877x^3 + 0.9003x^2y - 0.03153xy^2 + 0.001114y^3,$$

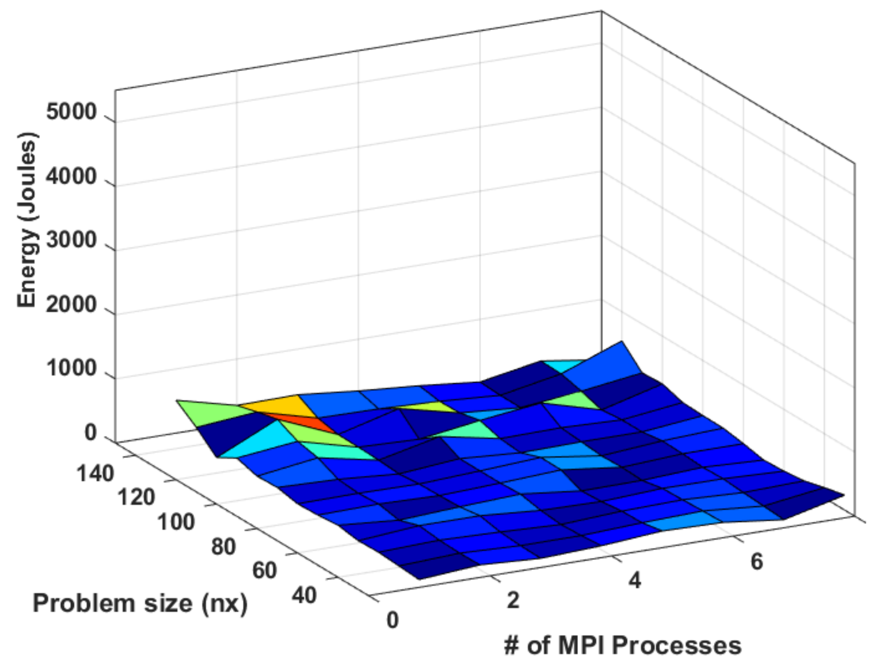


Figure 6: Absolute error value between the model and the experimental energy data. $R^2 = 97.85\%$ **and** $MSE = 105.1477$.

Conclusion and Future Work

- Fine-grained portable measurement infrastructure (WattProf card) can be used successfully for accurate measurement and analysis of realistic applications.
- Modeling for CPU energy
- new infrastructure aims to automate the data gathering, analysis and model-generation process for power and energy.
- integrating power measurement and modeling in the Orio (http://brnorris03.github.io/Orio/)auto-tuning framework.

?

(Extra Slides)

Top 500

RANK	SITE	SYSTEM	CORES	RMAX (TFLOP/S)	RPEAK (TFLOP/S)	POWER (KW)
1	National Super Computer Center in Guangzhou China	**Tianhe-2 (MilkyWay-2)** - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P NUDT	3,120,000	33,862.7	54,902.4	17,808
2	DOE/SC/Oak Ridge National Laboratory United States	**Titan** - Cray XK7 , Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x Cray Inc.	560,640	17,590.0	27,112.5	8,209
3	DOE/NNSA/LLNL United States	**Sequoia** - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom IBM	1,572,864	17,173.2	20,132.7	7,890
4	RIKEN Advanced Institute for Computational Science (AICS) Japan	K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect Fujitsu	705,024	10,510.0	11,280.4	12,660
5	DOE/SC/Argonne National Laboratory United States	**Mira** - BlueGene/Q, Power BQC 16C 1.60GHz, Custom IBM	786,432	8,586.6	10,066.3	3,945

WattProf

- The board can collect data for up to 128 sensors at up to 12KHz.
- We set it to 4KHz to be safe for call stack (Software bottleneck)
- Intel RAPL (Intel is just CPU and RAM). Model Based. Closed source.

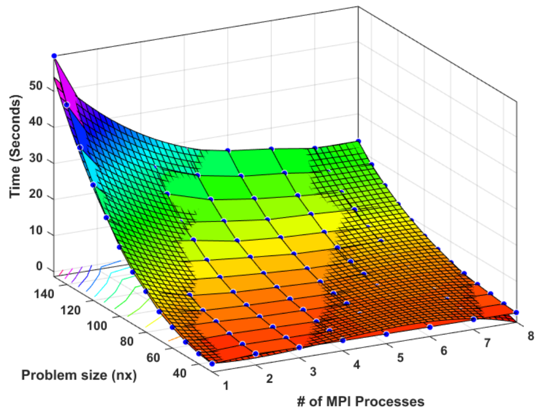
Machine Specs

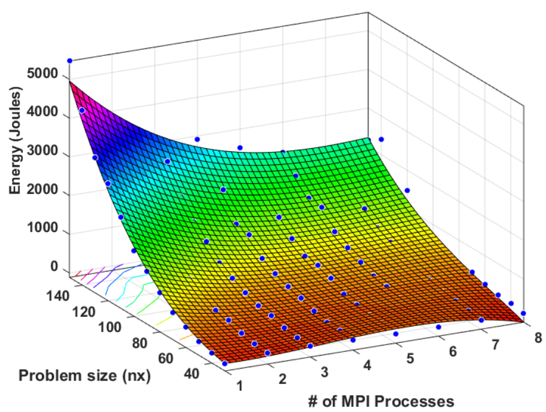
- We used the **WattProf** card on a machine with two Intel **Xeon CPUs E5620** with **24GB** memory

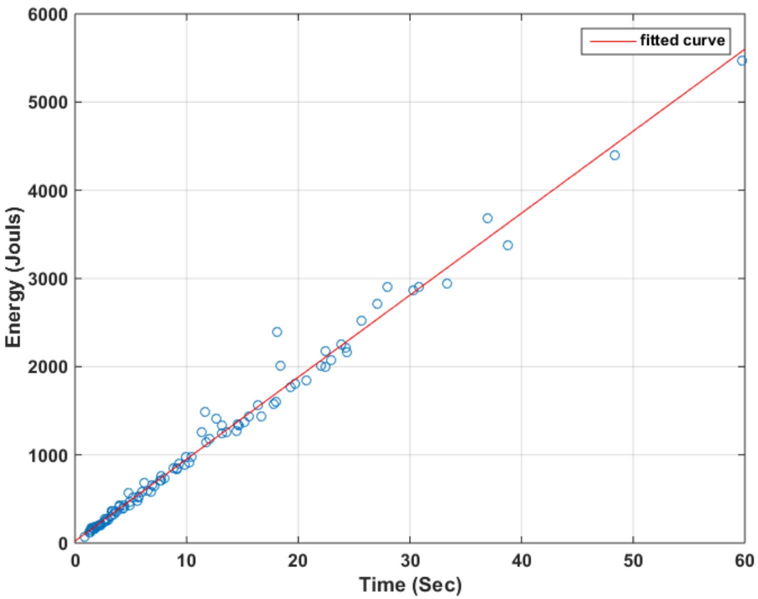
- and running **Ubuntu 14.04.2** with Linux kernel **3.13.** We
- considered problem sizes ranging from **30x30x30 to 150x150x150**
- and different numbers of MPI processes ranging from 1 to
- 8. We compiled the MPI-based miniFE with **GCC 4.8.2**
- with optimization **levels -O0, -O1, -O2 and -O3** in order to study optimization on power and energy consumption.

Energy Model and Time

- Time and CPU energy are highly correlated (~97%)
- Time is more predictable. Smoother curve.







Energy Model and Time

- Time and CPU energy are highly correlated
- Time is more predictable. Smoother curve.