Tools for Assessing and Optimizing the Energy Requirements of High Performance Scientific Computing Software

Kai Diethelm

GNS Gesellschaft für Numerische Simulation mbH
Braunschweig, Germany

DMV-GAMM Joint Annual Meeting
TU Braunschweig, 07–11 March 2016

Outline

Acknowledgement

The work described in the talk is part of the projects READEX (Runtime Exploitation of Application Dynamism for Energy-efficient eXascale computing) and Score-E.

The project Score-E is financially supported by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IH13001.

SPONSORED BY THE Federal Ministry of Education and Research

The project READEX has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 671657.

European Commission | Horizon 2020 European Union funding for Research & Innovation

Partners

JÜLICH FORSCHUNGSZENTRUM

TUM

RWTH AACHEN UNIVERSITY

TECHNISCHE UNIVERSITÄT DRESDEN

intel

TECHNISCHE UNIVERSITÄT DARMSTADT

OÉ Gaillimh NUI Galway

IT4Innovations national supercomputing center

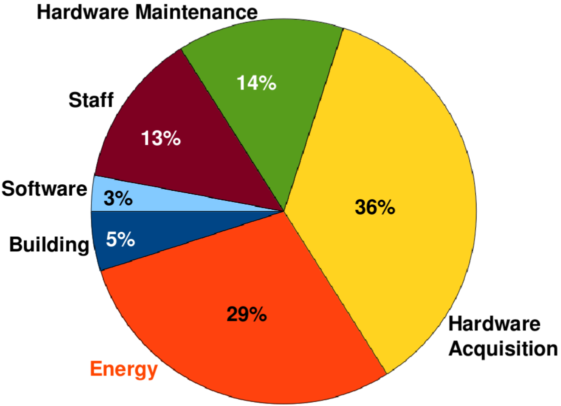
NTNU

Outline

1. Background
2. Energy and Performance Analysis
3. Outlook

Initial Observation

Distribution of Total Cost of Ownership of HPC System



Energy cost is already ∼30% of total cost, with rising tendency.

Source: Survey conducted by RWTH Aachen (2012)

Challenges

- HPC centers
  - are forced to develop strategies for energy efficient modes of operation,
  - will ask users (e.g., the scientific computing community) to run their codes in an energy efficient manner.

Challenges

- HPC centers
  - are forced to develop strategies for energy efficient modes of operation,
  - will ask users (e.g., the scientific computing community) to run their codes in an energy efficient manner.

- HPC users
  - are likely to be billed for energy instead of CPU time in the future,
  - must find out how much energy their codes need to run,
  - need to optimize their codes with respect to energy.

Possible Approach

LRZ München's Strategy for SuperMUC

Set default CPU frequency significantly lower than maximum.
Allow use of higher frequency only if users can demonstrate that it pays off.

Possible Approach

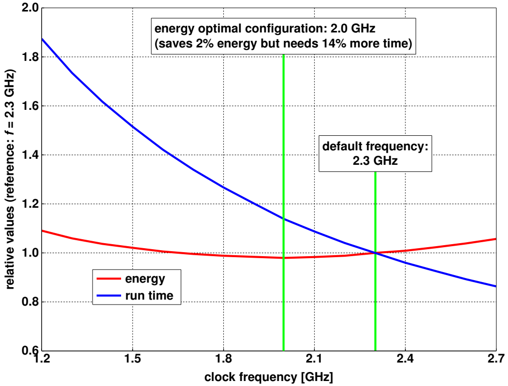
LRZ München's Strategy for SuperMUC

Set default CPU frequency significantly lower than maximum. Allow use of higher frequency only if users can demonstrate that it pays off.

- Model: $E = \int_0^T (P_{\text{static}}(t) + P_{\text{dynamic}}(t))dt$
- P_{static} is constant (idle power)
- P_{dynamic} depends on frequency f and voltage U: $P_{\text{dynamic}} \sim U^2 \cdot f$
- Run time T is non-increasing function of frequency f

Observations

- When running an application with higher frequency ...
  - static energy usually decreases (due to shorter run time)
  - dynamic energy usually increases
- Each application has its own optimal frequency!



When Does an Increased Frequency Pay Off?

Consequences of frequency reduction:

- Energy requirements are reduced
- Run time is increased
- Throughput of system is decreased
- Capabilities of hardware are not fully utilized
- More investement in hardware necessary to satisfy demand

Frequency increase pays off if run time decreases significantly

Pure focus on energy is not appropriate.

Outline

Elementary Approach

Energy-delay product:

$$\mathrm{EDP} = E \cdot T^w$$

- E: Energy used by the application run
- T: Run time used by the application run
- Parameter w is used to prioritize run time
- Usually $w \in \{1, 2, 3\}$ (EDP1, EDP2 and EDP3)
- A *single* metric to optimize both energy and hardware cost
  $\Rightarrow$ useful for our purpose in spite of physical implausibility

Static Tuning

- Run code with a few typical example data sets and different frequencies
- Measure run times and energy requirements
- Select frequency with best EDP for production runs

Static Tuning

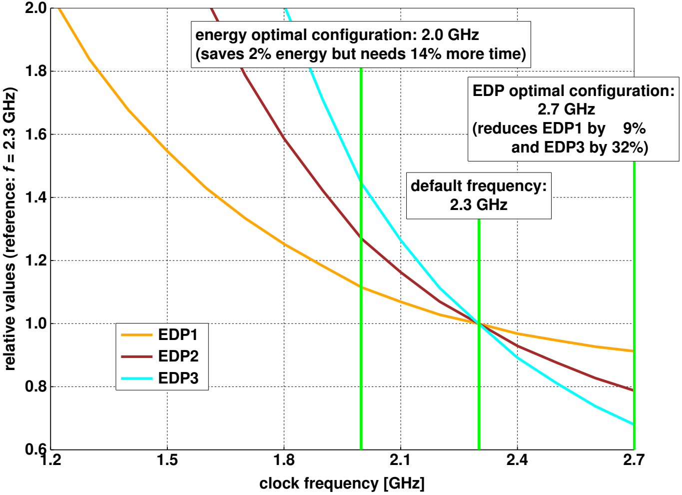
- Run code with a few typical example data sets and different frequencies
- Measure run times and energy requirements
- Select frequency with best EDP for production runs

Same approach can be followed for other tuning parameters:

- # of MPI tasks
- # of OpenMP threads
- ...

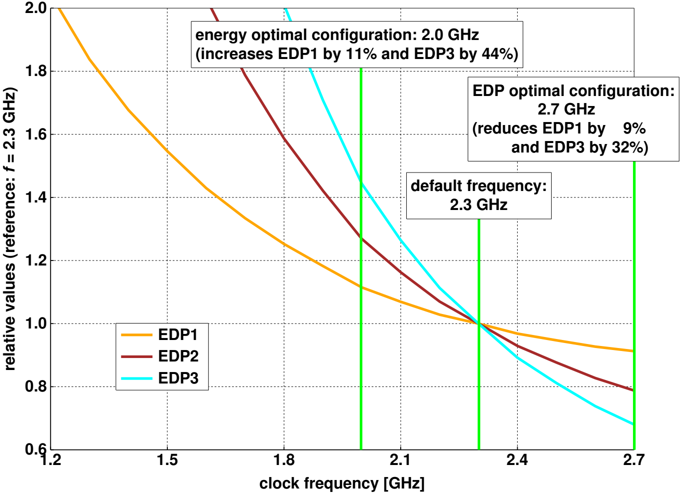
Static Tuning

Example: Indeed (FE code for sheet metal forming simulation)



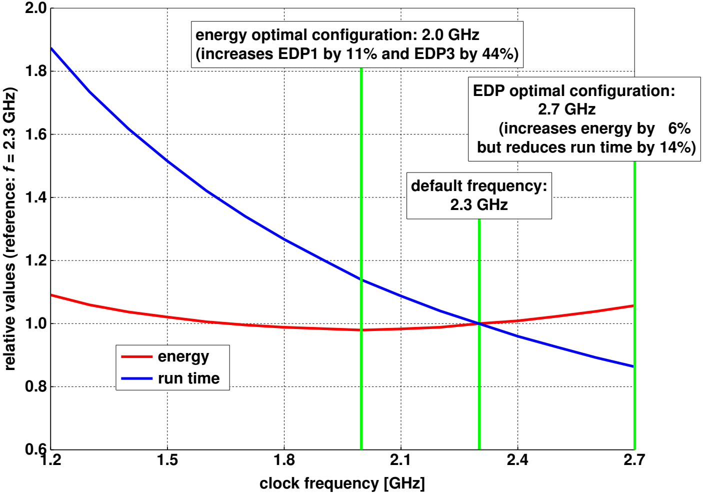
Static Tuning

Example: Indeed (FE code for sheet metal forming simulation)



Static Tuning

Example: Indeed (FE code for sheet metal forming simulation)

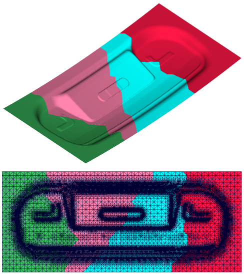


Dynamic Tuning

Fundamental Property

Many HPC codes exhibit dynamic behaviour.

Example: Distribution of workload in Indeed run (4 MPI tasks)



Dynamic Tuning

Exploit dynamic behaviour at run time:

- reduce CPU frequency for I/O bound or memory bound parts of code
- increase CPU frequency for compute bound parts of code
- reduce # of OpenMP threads in case of lock contention
- reduce # of MPI tasks if problem size is small or if algorithm scales poorly
- …

Tools Landscape

Unified measurement infrastructure

Score-P (`http://www.score-p.org`)

Tools Landscape

Analysis tools for performance

- CUBE: Profiling
- Scalasca: Automatic trace analysis
- Vampir: Interactive trace analysis
- TAU: Profiling and tracing
- Periscope Tuning Framework:
  On-line analysis and tuning

Unified measurement infrastructure

Score-P (`http://www.score-p.org`)

Tools Landscape

Analysis tools for performance and energy requirements

- CUBE: Profiling
- Scalasca: Automatic trace analysis
- Vampir: Interactive trace analysis
- TAU: Profiling and tracing

new: visualization & analysis capabilities for energy related metrics

- Periscope Tuning Framework: On-line analysis and tuning

new: energy tuning plugins (PCAP, DVFS, MPIProcs, . . .)

Unified measurement infrastructure

Score-P (`http://www.score-p.org`)

new: interface to energy measurement hardware

Outline

Extensions (Work in Progress)

(Semi)-automatic energy tuning I: At design time

Extensions (Work in Progress)

(Semi)-automatic energy tuning I: At design time

- Select parameters to be used for tuning, e.g.
  - # of OpenMP threads
  - # of MPI processes
  - CPU frequency
  - different code paths
  - . . .

Extensions (Work in Progress)

(Semi)-automatic energy tuning I: At design time

- Select parameters to be used for tuning, e.g.
  - # of OpenMP threads
  - # of MPI processes
  - CPU frequency
  - different code paths
  - …
- Identify points in code where change of tuning parameters is reasonable

Extensions (Work in Progress)

(Semi)-automatic energy tuning I: At design time

- Select parameters to be used for tuning, e.g.
  - # of OpenMP threads
  - # of MPI processes
  - CPU frequency
  - different code paths
  - ...
- Identify points in code where change of tuning parameters is reasonable
- Identify certain scenarios at design time

Extensions (Work in Progress)

(Semi)-automatic energy tuning I: At design time

- Select parameters to be used for tuning, e.g.
  - # of OpenMP threads
  - # of MPI processes
  - CPU frequency
  - different code paths
  - ...
- Identify points in code where change of tuning parameters is reasonable
- Identify certain scenarios at design time
- Find energy-optimal configuration for continuation of program run

Extensions (Work in Progress)

(Semi)-automatic energy tuning II: At run time

Extensions (Work in Progress)

(Semi)-automatic energy tuning II: At run time

- Automatic switching between configurations at run time according to current scenario (via READEX runtime library)

Extensions (Work in Progress)

(Semi)-automatic energy tuning II: At run time

- Automatic switching between configurations at run time according to current scenario (via READEX runtime library)

Advantages:

- Platform-independent software development process
- User friendliness

Extensions (Work in Progress)

(Semi)-automatic energy tuning II: At run time

- Automatic switching between configurations at run time according to current scenario (via READEX runtime library)

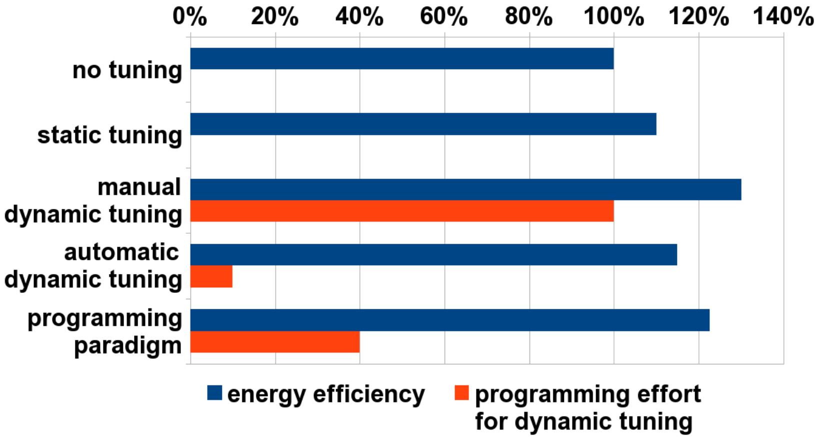
Advantages:

- Platform-independent software development process
- User friendliness

Future work:

- Development of programming paradigm for expressing dynamism
  Goal: further improvement of automatic dynamic tuning

Typical Outcome



Thank you for your attention!

Further information: `http://www.vi-hps.org/projects/score-e`
`http://www.readex.eu`

Contact: **diethelm@gns-mbh.com**