ENERGY EFFICIENCY & OPTIMIZATION FOR SUPERCOMPUTERS

Parallel Programming course [IN2147]

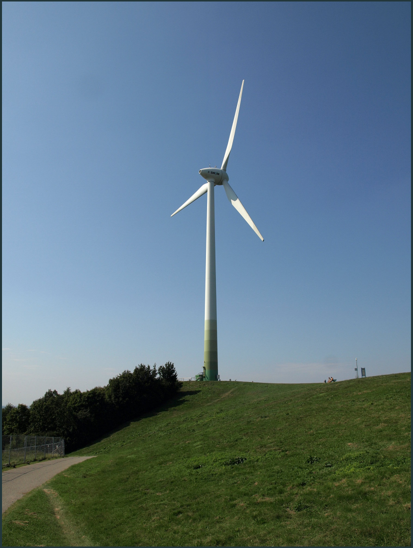
Gerasimos Chourdakis

May 27, 2015

Technische Universität München - Supervisor: Prof. Dr. Michael Gerndt

MOTIVATION

WHAT IS THIS?



FRÖTTMANING WIND POWER PLANT

Rated power: 1.5 MW

FRÖTTMANING WIND POWER PLANT

Rated power: 1.5 MW

SuperMUC power consumption: ~3.4 MW

FRÖTTMANING WIND POWER PLANT

Rated power: 1.5 MW

SuperMUC power consumption: ~3.4 MW

If this was used to power SuperMUC, it just... **wouldn’t be enough!**

Performance is important, but it comes at a **cost**!

The Green500.org list sorts the world's best supercomputers in terms of energy efficiency.

Table: Part of the Green500 list - November 2014 (rounded numbers)

Name	Top500#	Green500#	MFLOPS/W	Power (kW)
Tianhe-2	1	64	1 902	17 808
Titan	2	53	2 143	8 209
Sequoia	3	48	2 177	7 890
K Computer	4	156	830	12 660
JUQUEEN	8	40	2 177	2 301
SuperMUC	14	152	846	3 423

THE GREEN500 LIST

Performance is important, but it comes at a **cost**!

The Green500.org list sorts the world's best supercomputers in terms of energy efficiency.

Table: Part of the Green500 list - November 2014 (rounded numbers)

Name	Top500#	Green500#	MFLOPS/W	Power (kW)
JUQUEEN	8	40	2 177	2 301
Sequoia	3	48	2 177	7 890
Titan	2	53	2 143	8 209
Tianhe-2	1	64	1 902	17 808
SuperMUC	14	152	846	3 423
K Computer	4	156	830	12 660

ENERGY EFFICIENCY

A ROUGH CPU POWER MODEL

The **dominating part** of the power consumption of a CPU is its dynamic power, that is proportional to its frequency, and the square of its voltage:

$$P_{\text{CPU}} = P_{\text{dynamic}} + P_{\text{short circuit}} + P_{\text{leakage}} \tag{1}$$

$$P_{\text{dynamic}} = \text{Capacitance} \cdot \text{Voltage}^2 \cdot \text{frequency} \tag{2}$$

Modern processors try to save energy in various ways:

- Reducing frequency or voltage (DVFS)
- Turning unused areas off
- Reducing capacitance (smaller chips)
- Recycling energy stored in capacitors, and more.

ENERGY MANAGEMENT IN THE CPU

Modern processors try to save energy in various ways:

- Reducing frequency or voltage (DVFS)
- Turning unused areas off
- Reducing capacitance (smaller chips)
- Recycling energy stored in capacitors, and more.

Thinking out-of-the-box: The ARM architectures are RISC-type, in comparison to usual x86 CISC architectures. They need less transistors, that roughly means less power!

ENERGY MANAGEMENT IN THE CPU

Modern processors try to save energy in various ways:

- Reducing frequency or voltage (DVFS)
- Turning unused areas off
- Reducing capacitance (smaller chips)
- Recycling energy stored in capacitors, and more.

Thinking out-of-the-box: The ARM architectures are RISC-type, in comparison to usual x86 CISC architectures. They need less transistors, that roughly means less power!

Of course, a supercomputer is not built only with CPUs...

EFFICIENT INFRASTRUCTURE

Remember: we cannot ignore the energy balance! The electric power that we consume is transformed to heat.

CPUs have specific operating temperature specifications, so we need to remove the produced heat (cooling).

Also, a supercomputing center does not consume energy only for the supercomputer. A good example: LRZ!

EXAMPLE: WARM-WATER COOLING

SuperMUC is water-cooled (IBM ”Aquasar”). More than this, it is cooled with warm water!?!?

EXAMPLE: WARM-WATER COOLING

SuperMUC is water-cooled (IBM ”Aquasar”). More than this, it is cooled with warm water!?!?

Do we really need a fridge-cold liquid (cost for chilling)? Typical temperature limit is 85°C and it is not exceeded even with 60°C ”cooling” water.

SuperMUC is water-cooled (IBM ”Aquasar”). More than this, it is cooled with warm water!?!?

Do we really need a fridge-cold liquid (cost for chilling)? Typical temperature limit is 85°C and it is not exceeded even with 60°C ”cooling” water.

Plus: the water at the output contains energy in high temperature, i.e. useful energy! All the LRZ buildings can be heated with this ”free” energy.

We use more and more **accelerators** in place of more CPU cores. But accelerators are power-greedy!

We use more and more **accelerators** in place of more CPU cores. But accelerators are power-greedy!

What about **hybrid systems**? For example, ARM CPUs connected with GPUs.

We use more and more **accelerators** in place of more CPU cores. But accelerators are power-greedy!

What about **hybrid systems**? For example, ARM CPUs connected with GPUs.

Exa-scale challenges: could we reliably produce and manage e.g. 20MW of electricity just for a supercomputer? Could we also efficiently remove this (condensed) heat?

Keep in mind: electric energy can be cheaper if it is predictably consumed at a constant rate. This changes our viewpoint: variate the number of nodes to keep the energy constant. (scheduler)

ENERGY OPTIMIZATION

Remember:

$P_{\text{dynamic}} = \text{Capacitance} \cdot \text{Voltage}^2 \cdot \text{frequency}$

So, we could just (easily) decrease the frequency, right?

Hmm...

We are not actually paying for power, but for energy, i.e. power $\cdot$ time.

For which frequency do we get the desired minimum Energy to Solution?

Other metrics: Energy Delay Product (EDP) or even ED^nP.

$$E = \frac{W_0 + (W_1 \cdot f + W_2 \cdot f^2)t}{\min(t \cdot P_0 \cdot f/f_0, P_{roof})} \tag{3}$$

$$f_{optimal} = \sqrt{\frac{W_0}{w_2 \cdot t}} \tag{4}$$

Assumptions:

1. Certain baseline power W_0 is used for the whole (multicore) chip.
2. An active core consumes a dynamic power of $W_1 \cdot f + W_2 \cdot f^2$.
3. At baseline frequency f_0 we have a baseline performance P_0. A bottleneck may restrict the performance to P_{roof}.

HOW TO GET DATA?

We are trying to optimize the ”consumed energy”, but at which level?

HOW TO GET DATA?

We are trying to optimize the ”consumed energy”, but at which level?

We could measure the energy consumed at **hardware level**, by using the internal CPU event counters. This would give us high **measurement frequency**, and more **information** but also an **overhead** due to in-band measurement and model inaccuracies.

HOW TO GET DATA?

We are trying to optimize the ”consumed energy”, but at which level?

We could measure the energy consumed at **hardware level**, by using the internal CPU event counters. This would give us high **measurement frequency**, and more **information** but also an **overhead** due to in-band measurement and model inaccuracies.

We could directly measure the energy consumed at **node or higher level**, by using ”smart” power supplies. These give us low frequencies but they can provide useful statistics for **long-run** applications.

...or ”energy optimization made easy”!

Optimize performace first! For this, we need a profiling tool to locate the ”hot spots” of our code. Already wide-spread tools: gprof, Vampir, Scalasca and other.

…or ”energy optimization made easy”!

Optimize performace first! For this, we need a profiling tool to locate the ”hot spots” of our code. Already wide-spread tools: gprof, Vampir, Scalasca and other.

After we optimize performance, what about manipulating **compiler flags and cpu frequency**?

TUM, LRZ and other European partners develop the Periscope Tuning Framework and the AutoTune plugin: periscope.in.tum.de.

It is an **online** (no tracing files) analysis and tuning framework for performance and energy, built to work with Eclipse.

It can help to decide which compiler flags and MPI parameters to use, as well as to find the optimal CPU frequency for our code.

How does it work?

Through a graphical user interface, the user sets up the different **scenarios** to be checked and provides a test dataset.

The user also selects a **search algorithm** that compiles and runs the program for the different scenarios in order to find the combination of parameters that gives the optimal execution time or energy metrics.

SUMMARY

SUMMARY

- Supercomputers are energy-hungry and there is a good reason in spending a lot of effort to optimize their energy consumption.

SUMMARY

- Supercomputers are energy-hungry and there is a good reason in spending a lot of effort to optimize their energy consumption.
- CPUs consume less energy at lower frequency, but the energy to solution is what matters. Sometimes, the time to solution is even more important than energy.

SUMMARY

- Supercomputers are energy-hungry and there is a good reason in spending a lot of effort to optimize their energy consumption.
- CPUs consume less energy at lower frequency, but the energy to solution is what matters. Sometimes, the time to solution is even more important than energy.
- The energy consumed by the CPU also produces energy that needs to be removed. Warm-water cooling can decrease the cooling cost and could be paired with other heating processes.

SUMMARY

- Supercomputers are energy-hungry and there is a good reason in spending a lot of effort to optimize their energy consumption.
- CPUs consume less energy at lower frequency, but the energy to solution is what matters. Sometimes, the time to solution is even more important than energy.
- The energy consumed by the CPU also produces energy that needs to be removed. Warm-water cooling can decrease the cooling cost and could be paired with other heating processes.
- We can optimize based on measurements provided from the CPU or from other equipment, with varying measurement frequency and overhead.

SUMMARY

- Supercomputers are energy-hungry and there is a good reason in spending a lot of effort to optimize their energy consumption.
- CPUs consume less energy at lower frequency, but the energy to solution is what matters. Sometimes, the time to solution is even more important than energy.
- The energy consumed by the CPU also produces energy that needs to be removed. Warm-water cooling can decrease the cooling cost and could be paired with other heating processes.
- We can optimize based on measurements provided from the CPU or from other equipment, with varying measurement frequency and overhead.
- Before we optimize energy consumption, we must have an efficient code. Profiling tools can make our life easier.

SUMMARY

- Supercomputers are energy-hungry and there is a good reason in spending a lot of effort to optimize their energy consumption.
- CPUs consume less energy at lower frequency, but the energy to solution is what matters. Sometimes, the time to solution is even more important than energy.
- The energy consumed by the CPU also produces energy that needs to be removed. Warm-water cooling can decrease the cooling cost and could be paired with other heating processes.
- We can optimize based on measurements provided from the CPU or from other equipment, with varying measurement frequency and overhead.
- Before we optimize energy consumption, we must have an efficient code. Profiling tools can make our life easier.
- The Periscope Tuning Framework is a nice tools that can analyze and fine-tune our code for the specific system that we are using.

1. ”Tools and methods for measuring and tuning the energy efficiency of HPC systems”, R. Schöne et al., Scientific Programming 22 (2014) 273–283. DOI 10.3233/SPR-140393.
2. ”A Case Study of Energy Aware Scheduling on SuperMUC”, A. Auweter & L. Brochard, LRZ, 30/06/2014. (presentation)
3. ”Automatic Tuning of HPC Applications with Periscope”, M. Gerndt, M. Firbach and I. Compres, LRZ, 23/04/2015. (presentation)

LICENSING

You can find this presentation on Github:

`https://github.com/MakisH/`

This work is provided with a Creative Commons Attribution-ShareAlike 4.0 International License.

Picture on slide #3 belongs to Hochgeladen von Hihiman and is covered by a CC BY-SA 3.0 license. Downloaded from `https://de.wikipedia.org/`.

The slides are built with the ”m” beamer theme by Matthias Vogelgesang that is provided with a CC BY-SA 4.0 license and can be found at `github.com/matze/mtheme`.

Questions?