Optimal Dynamic Frequency Scaling for Energy - Performance of Parallel MPI Programs

Jean-Claude Charr, Raphaël Couturier, Ahmed Fanfakh and Arnaud Giersch

FEMTO-ST - DISC Department - AND Team

June 3th, 2014

Outline

Definitions

- Modern processors provide **Dynamic Voltage and Frequency Scaling (DVFS)** technique.
- DVFS is used to reduce the frequency and thus to **reduce the energy consumption** by a CPU while computing.
- **But** scaling the frequency to lower level reduces the performance **(execution time)** of parallel program.
- Energy consumption for individual processor depends on two power metrics: the **static power** P_{static} and the **dynamic power** P_{dyn}.

Definitions

- $P_{dyn} = \alpha \cdot C_L \cdot V^2 \cdot F$.
- $P_{static} = V \cdot N_{trans} \cdot K_{design} \cdot I_{leak}$.
- Energy consumption by **individual processor** of a synchronous parallel program:
  $E_{ind} = P_{dyn} \cdot T_{Comp} + P_{static} \cdot (T_{Comp} + T_{Comm})$.
- The frequency scaling factor is the ratio between the maximum and the new frequency, $S = \frac{F_{max}}{F_{new}}$.

Objectives

- Study the effect of the scaling factor *S* on **energy consumption** of parallel iterative applications such as NAS Benchmarks.
- Study the effect of the scaling factor *S* on **performance** of these benchmarks.
- Discovering the **energy-performance trade-off relation** when changing the frequency.
- We propose an algorithm for selecting the scaling factor *S* producing **optimal trade-off** between the energy and performance.
- Improving Rauber and Rünger's[1] method that our method best on.

[1] Thomas Rauber and Gudula Rünger. Analytical modeling and simulation of the energy consumption of independent tasks. In Proceedings of the Winter Simulation Conference, 2012.

Energy model for homogeneous platform

The dynamic power is **exponentially** related to the scaling factor S and the static consumed energy is **linearly** related to this factor.

Rauber and Rünger's energy model

$$E = P_{dyn} \cdot S_1^{-2} \cdot \left(T_1 + \sum_{i=2}^{N} \frac{T_i^3}{T_1^2} \right) + P_{static} \cdot S_1 \cdot T_1 \cdot N$$

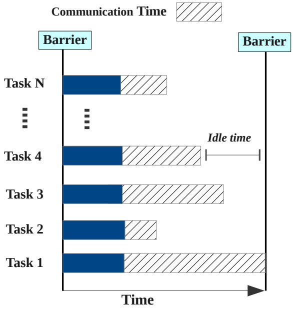
S_1: is the max. scaling factor, T_I: is the time of the slower task, T_i: is the time of the other tasks and N: is the number of nodes.

Rauber and Rünger's optimal scaling factor

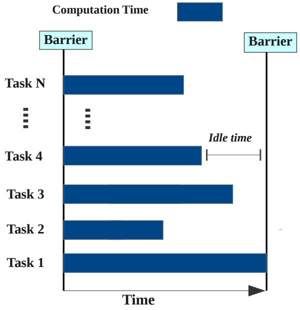
$$S_{opt} = \sqrt[3]{\frac{2}{N} \cdot \frac{P_{dyn}}{P_{static}} \cdot \left(1 + \sum_{i=2}^{N} \frac{T_i^3}{T_1^3} \right)}$$

They reduce degradation of the performance by **setting the highest frequency to the slowest task**.

Slack times of the sync. parallel program



(a) Sync. imbalanced communications



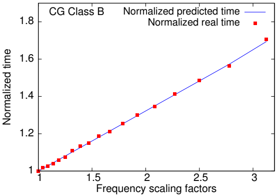
(b) Sync. imbalanced computations

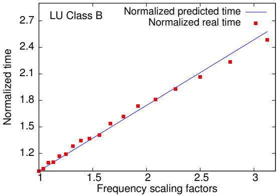
$$\boldsymbol{ProgramTime} = \max_{i=1,2,\dots,N} T_i$$

Performance evaluation of MPI programs

Execution time prediction model

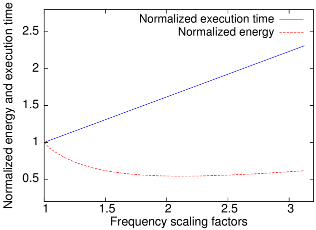
$$T_{new} = T_{MaxCompOld} \cdot S + T_{MaxCommOld}$$



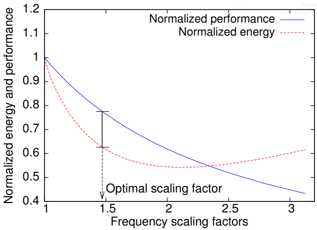


The maximum normalized error for CG=0.0073 **(the smallest)** and LU=0.031 **(the worst)**.

Performance and energy reduction trade-off



(c) Real relation.



(d) Converted relation.

$$Performance = \frac{1}{execution\ time}$$

Our objective function

$$\boldsymbol{MaxDist} = \max_{j=1,2,\dots,F}(\overbrace{P_{Norm}(S_j)}^{Maximize} - \overbrace{E_{Norm}(S_j)}^{Minimize})$$

Scaling factor selection algorithm

Enumerate the available scaling factors and find $S_{optimal}$ for which $P_{Norm} - E_{Norm}$ is maximal.

<u>Where:</u>

$$E_{Norm} = \frac{E_{Reduced}}{E_{Original}} = \frac{P_{dyn} \cdot S_1^{-2} \cdot \left(T_1 + \sum_{i=2}^{N} \frac{T_i^3}{T_1^2} \right) + P_{static} \cdot T_1 \cdot S_1 \cdot N}{P_{dyn} \cdot \left(T_1 + \sum_{i=2}^{N} \frac{T_i^3}{T_1^2} \right) + P_{static} \cdot T_1 \cdot N}$$

$$P_{Norm} = \frac{T_{old}}{T_{new}} = \frac{T_{MaxCompOld} + T_{MaxCommOld}}{T_{MaxCompOld} \cdot S + T_{MaxCommOld}}$$

Scaling factor selection algorithm

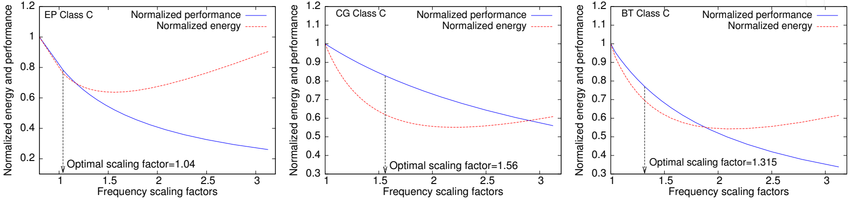
Algorithm characteristics

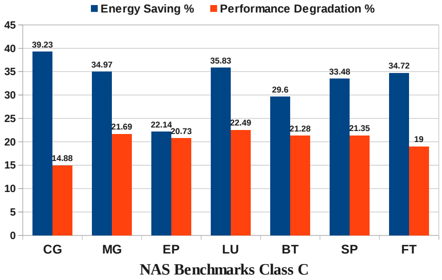
- It works online.
- It predicts both the energy consumption and performance.
- It is simultaneously reduces the energy consumption and maintaining performance of iterative algorithm.
- It takes into account the communication time.
- It is well adapted to imbalanced tasks. $F_i = \frac{F_{max} \cdot T_i}{S_{optimal} \cdot T_{max}}$
- It has a very small overhead. It takes **6.65** μs for 32 nodes.

Experimental results

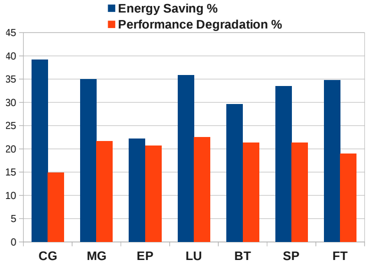
- Our experiments are executed on the simulator SimGrid/SMPI v3.10.
- Our algorithm is applied to NAS parallel benchmarks.
- Each node in the cluster has 18 frequency values from **2.5**GHz to **800**MHz.
- We run the classes A, B and C on 4, 8 or 9 and 16 nodes respectively.
- The dynamic power with the highest frequency is equal to **20** W and the power static is equal to **4** W.

Experimental results

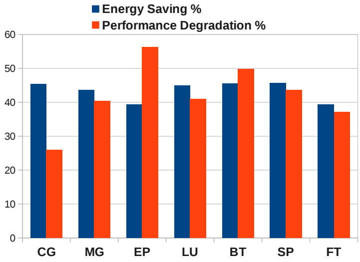




Results comparison

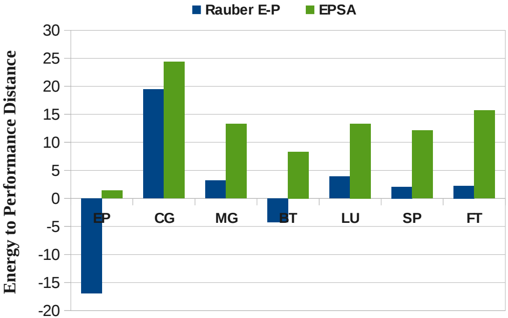


Our Method for NAS benchmarks Class C



Rauber and Rünger for NAS benchmarks Class C

Results comparison



Comparing our method with Rauber and Rünger method for NAS benchmarks class C

Conclusions

- We have presented a new online scaling factor selection method that **optimizes simultaneously the energy and performance**.
- It predicts **the energy consumption and the performance** of the parallel applications.
- Our algorithm **saves more energy** when the communication and the other slacks times are big.
- It gives the **best trade-off between energy reduction and performance**.
- Our method **outperforms Rauber and Rünger's method** in terms of energy-performance ratio.

Future works

- We will apply the proposed algorithm to a heterogeneous platform.
- While the nodes of a heterogeneous platform are different in:

  - Dynamic and static power.
  - Individual energy consumption.
  - The available frequencies.
  - Performance capabilities.

- We will apply the proposed algorithm to a real cluster.
- We will apply the proposed algorithm to a real applications.

Thanks for Listening

To appear

This work will be appear in ISPA conference proceedings, August 2014

Questions?