

Energy Consumption Optimization of Parallel Applications with Iterations using CPU Frequency Scaling

PhD Dissertation Defense

Ahmed Badri Muslim Fanfakh

Under the supervision of:

Raphaël COUTURIER and Jean-Claude CHARR

UBFC - FEMTO-ST - DISC Dept. - AND Team

17 October 2016

Outline



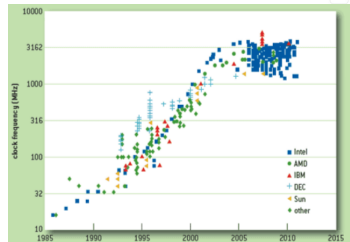
1. Introduction and Problem definition
2. Motivations
3. Energy optimization of a homogeneous platform
4. Energy optimization of a heterogeneous platform
5. Energy optimization of asynchronous applications
6. Conclusions and Perspectives

Introduction and problem definition

To get more computing power:

1) Increase the frequency of a processor.
(limited due to overheating)

2) Use more nodes.
The supercomputer Tianhe-2 has more than 3 million cores and consumes around 17.8 megawatts.



Techniques for energy consumption reduction



1) Switch-off idle nodes method

Techniques for energy consumption reduction

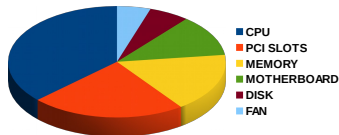


2) Dynamic voltage and frequency Scaling (DVFS)

Motivations

Why we used the DVFS method:

- The CPU is the component that consumes the highest amount of energy in a node ¹.
- DVFS reduces the energy consumption while keeping all the nodes working.
- It has a very small overhead compared to switching-off the idle nodes.



Challenge and Objective

Challenge: DVFS is used to reduce the energy consumption, **but** it degrades the performance simultaneously.

Objective: Applying the DVFS to minimize the energy consumption while maintaining the performance of the parallel application.

¹ Fan, X., Weber, W., and Barroso, L. A. 2007. Power provisioning for a warehouse-sized computer.

The first contribution



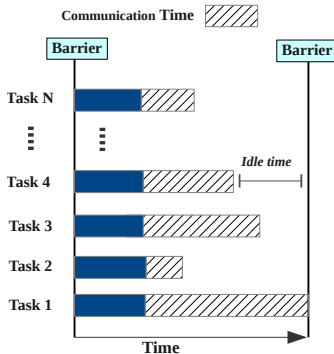
**Energy optimization of a parallel application
with iterations running over a homogeneous
platform**

Objectives

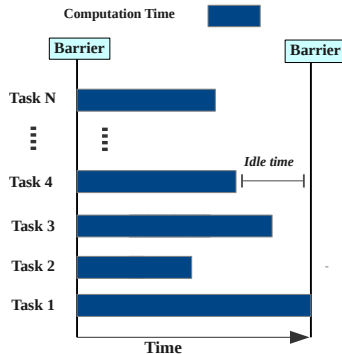


- Study the effect of the scaling factor on the **energy consumption and performance** of parallel applications with iterations.
- Discovering the **energy-performance trade-off relation** when changing the frequency of the processor.
- Proposing an algorithm for selecting the scaling factor that produces **the optimal trade-off** between the energy consumption and the performance.
- Comparing the proposed algorithm to existing methods.

Execution of synchronous parallel tasks



(a) Synchronous imbalanced communications



(b) Synchronous imbalanced computations

Energy model for a homogeneous platform

The power consumed by a processor divided into two power metrics: the dynamic (P_d) and static (P_s) power.

$$P_d = \alpha \cdot CL \cdot V^2 \cdot F \quad (1)$$

Where:

α : switching activity

V the supply voltage

CL : load capacitance

F : operational frequency

$$P_s = V \cdot N_{trans} \cdot K_{design} \cdot I_{Leak} \quad (2)$$

Where:

V : the supply voltage.

K_{design} : design dependent parameter.

N_{trans} : number of transistors.

I_{leak} : technology dependent parameter.

Energy model for a homogeneous platform

The frequency scaling factor is the ratio between the maximum and the new frequency, $S = \frac{F_{max}}{F_{new}}$.

Rauber and Runger's energy model

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

S_1 : the maximum scaling factor.

P_d : the dynamic power.

P_s : the static power.

T_1 : the execution time of the slower task.

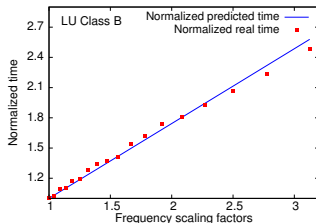
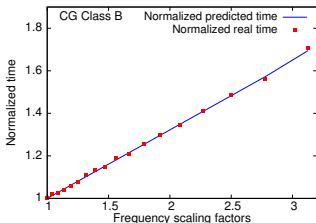
T_i : the execution time of task i .

N : the number of nodes.

Performance evaluation of MPI programs

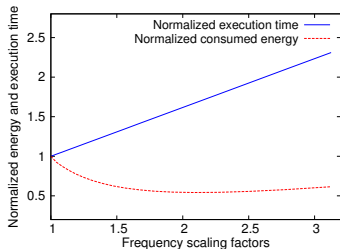
Execution time prediction model

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

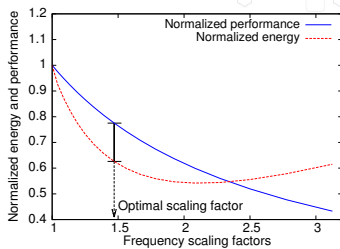


The maximum normalized error for CG=0.0073 (**the smallest**) and LU=0.031 (**the worst**).

Performance and energy reduction trade-off



(c) Real relation.



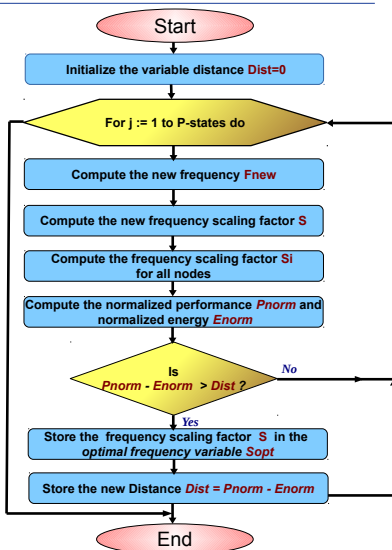
(d) Converted relation.

Where: *Performance* = *execution time*⁻¹

Our objective function

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

Scaling factor selection algorithm



Scaling algorithm example

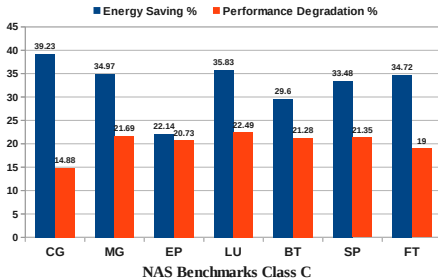
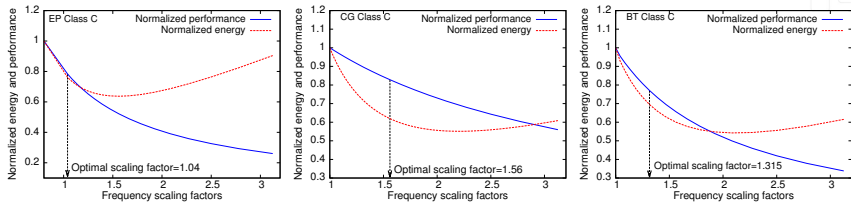


Experimental results



- The experiments were executed on the simulator SimGrid/SMPI v3.10.
- The proposed algorithm was applied to the NAS parallel benchmarks.
- Each node in the cluster has 18 frequency values from **2.5GHz** to **800MHz**.
- The proposed algorithm was evaluated over the A, B and C classes of the benchmarks using 4, 8 or 9 and 16 nodes respectively.
- $P_d = 20W$, $P_s = 4W$.

Experimental results

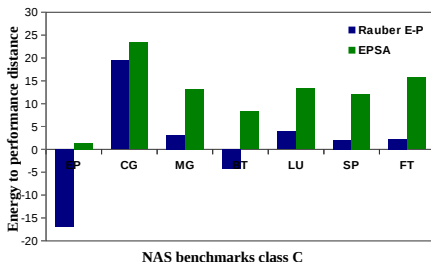


Results comparison



Rauber and Runger's optimal scaling factor

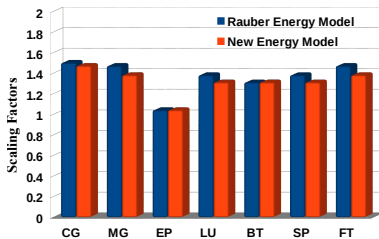
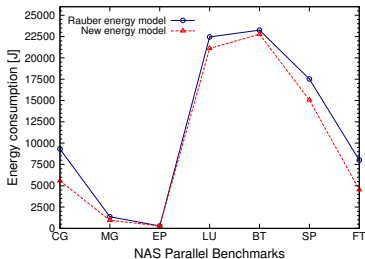
$$S_{opt} = \sqrt[3]{\frac{2}{N} \cdot \frac{P_{dyn}}{P_{static}} \cdot \left(1 + \sum_{i=2}^N \frac{T_i^3}{T_1^3}\right)}$$



The proposed new energy model



Comparing the new model with Rauber's model



The second contribution



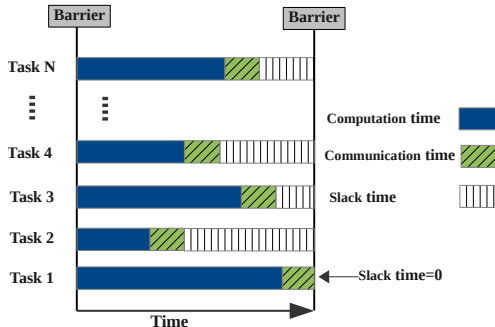
Energy optimization of a parallel application with iterations running over a Heterogeneous platform

Objectives



- Proposing **new energy and performance models** for message passing applications with iterations running over a heterogeneous platform (cluster or Grid).
- Studying the effect of the scaling factor S on both the **energy consumption and the performance** of message passing iterative applications.
- Computing the vector of scaling factors (S_1, S_2, \dots, S_n) producing **the optimal trade-off** between the energy consumption and the performance.

The execution time model



The execution time prediction model

$$T_{new} = \max_{i=1,2,\dots,N} (T_{cpOld_i} \cdot S_i) + \min_{i=1,2,\dots,N} (T_{cm_i}) \quad (3)$$

Where: T_{cm} = *communication times* + *slack times*

The energy consumption model



The overall energy consumption of a message passing synchronous application executed over a heterogeneous platform can be computed as follows:

$$E = \sum_{i=1}^N (S_i^{-2} \cdot Pd_i \cdot T_{cp_i}) + \sum_{i=1}^N (Ps_i \cdot (\max_{i=1,2,\dots,N} (T_{cp_i} \cdot S_i) + \min_{i=1,2,\dots,N} (T_{cm_i}))) \quad (4)$$

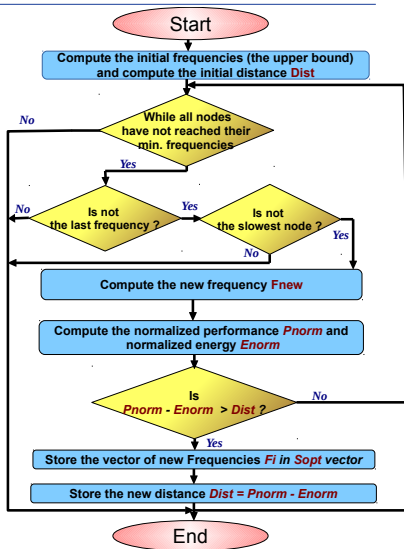
where:

N : is the number of nodes.

The energy model example for heter. cluster



The scaling algorithm for heter. cluster



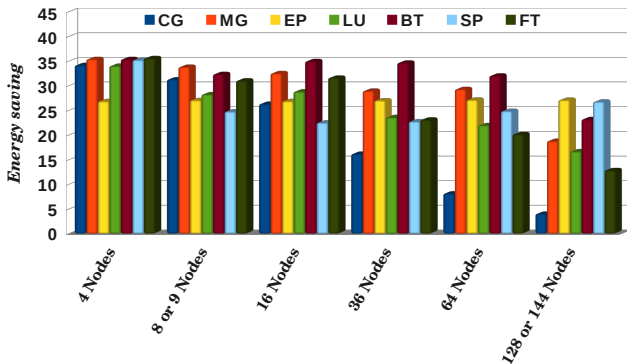
The scaling algorithm example



Experiments over a heterogeneous cluster

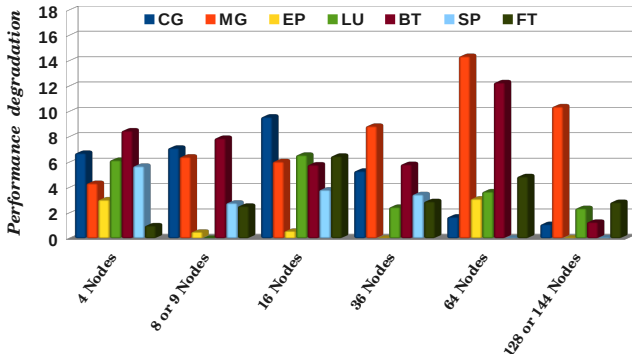
- The experiments were executed on the simulator SimGrid/SMPI v3.10.
- The scaling algorithm was applied to the NAS parallel benchmarks class C.
- Four types of processors with different computing powers were used.
- The benchmarks were executed with different number of nodes ranging from 4 to 144 nodes.
- It was assumed that the total power consumption of the CPU consist of 80% dynamic power and 20% static power.

The experimental results



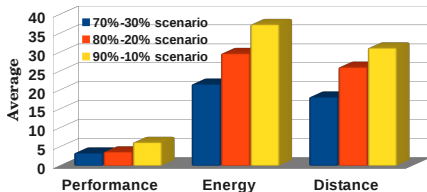
On average, it reduces the energy consumption by **29%** for the class C of the NAS Benchmarks executed over 8 nodes

The experimental results



On average, it degrades by **3.8%** the performance of NAS Benchmarks class C executed over 8 nodes

The results of the three power scenarios



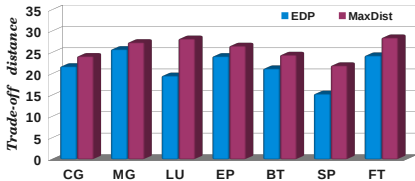
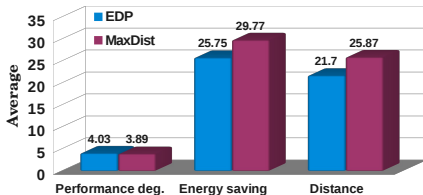
Selected frequency scaling factors for 8 nodes

70%-30% Scenario	1.04	1.33	1.61	1.88	1.04	1.24	1.61	1.88
80%-20% Scenario	1.04	1.33	1.61	1.88	1.04	1.33	1.61	1.88
90%-10% Scenario	1.08	1.42	1.61	2.03	1.04	1.33	1.61	1.88

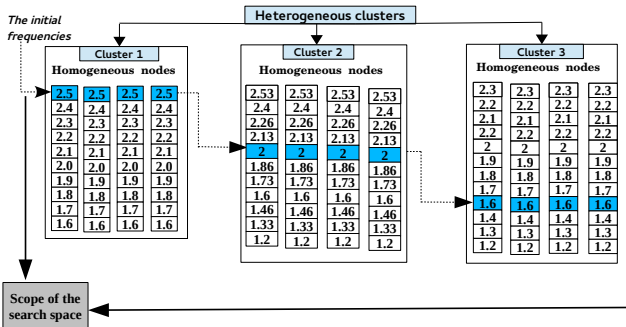
Arrows indicate the following adjustments from the 70%-30% scenario to the 90%-10% scenario:
- Node 6: 1.24 to 1.33
- Node 7: 1.61 to 1.61 (no change)
- Node 8: 1.88 to 1.88 (no change)
- Node 1: 1.04 to 1.08
- Node 2: 1.33 to 1.42
- Node 3: 1.61 to 1.61 (no change)
- Node 4: 1.88 to 2.03
- Node 5: 1.04 to 1.04 (no change)

Comparing the objective function to EDP

EDP is the products between the energy consumption and the delay.

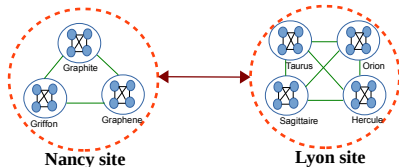


The grid architecture

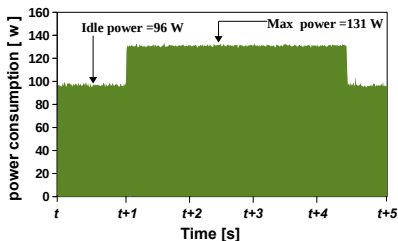


Experiments over Grid'5000

The experiments were conducted using three clusters distributed over one or two sites.



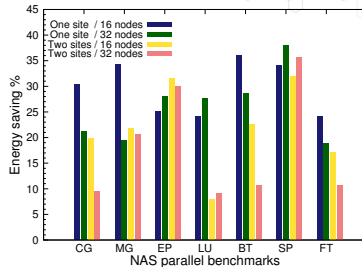
Grid'5000 power measurement tools were used.



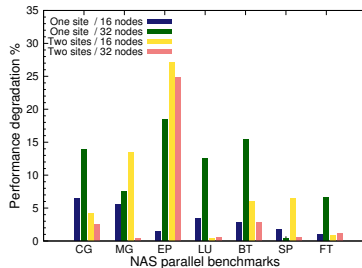
Experiments over Grid'5000



The average energy saving
= 30%



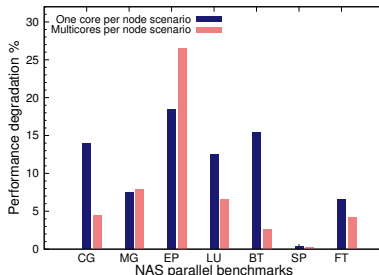
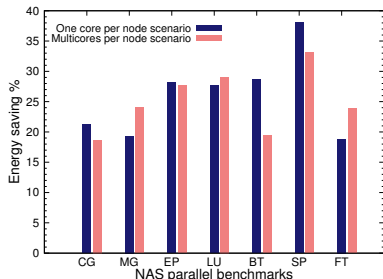
The average performance
degradation = 3.2%



Experiments over Grid'5000



One core and Multi-cores per node results:



Using multi-cores per node scenario decreases the computations to communications ratio.

The third contribution



Energy optimization of asynchronous iterative message passing applications

Problem definition

The execution of a synchronous parallel iterative application over a grid



Problem definition

The execution of an asynchronous parallel iterative application over a grid



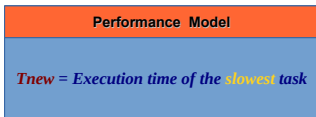
Solution

Using asynchronous communications with DVFS

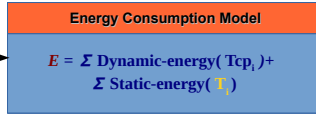
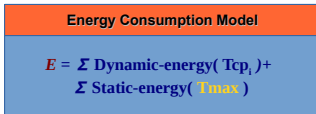
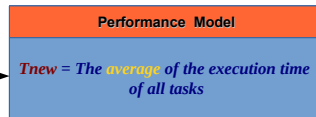


The performance and the energy models

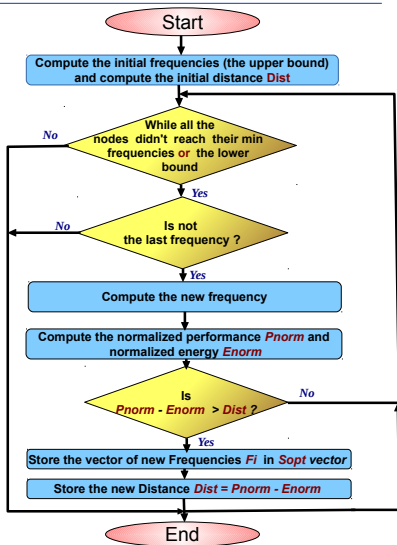
Synchronous Applications



Asynchronous Applications



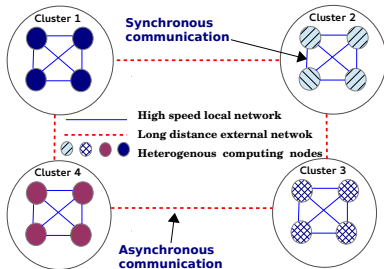
The scaling algorithm for Asynch. applications



The experiments



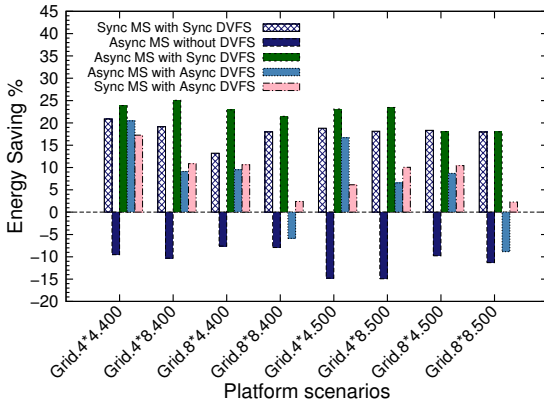
- The architecture of the grid:



- Applying the proposed algorithm to the asynchronous iterative message passing multi-splitting method.
- Evaluating the application over the simulator and Grid'5000.

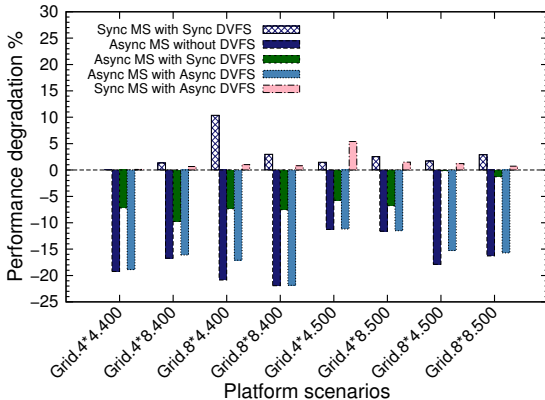
The simulation results

The best scenario in terms of energy and performance is the Async.
MS with Sync. DVFS



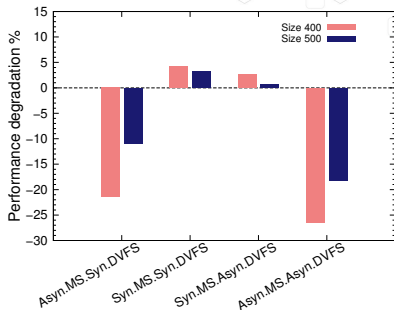
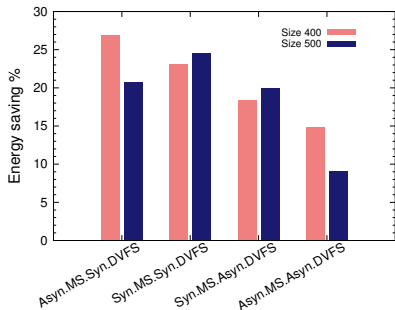
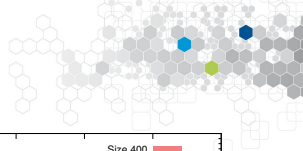
The average energy saving = **22%**

The simulation results



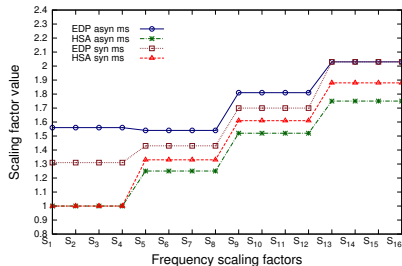
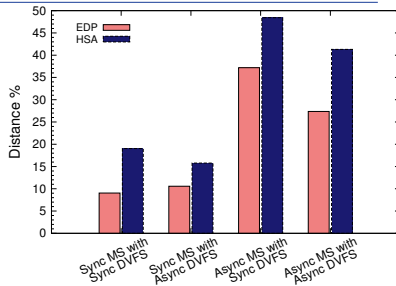
The average speed-up = **5.72%**

The Grid'5000 results



The average energy saving = **26.93%**, the average speed-up = **21.48%**

The comparison results



Conclusions



- Three **new energy consumption and performance** models were proposed for synchronous or asynchronous parallel applications with iterations running over **homogeneous and heterogeneous clusters or grids**.
- A **new objective function** to optimize both the energy consumption and the performance was proposed.
- **New online frequency selecting algorithms** for clusters and grids were developed.
- The proposed algorithms were applied to the **NAS parallel benchmarks** and **the Multi-splitting** method.
- The proposed algorithms were evaluated over the **SimGrid simulator** and over the **Grid'5000 testbed**.
- All the proposed methods were compared to either **Rauber and Runger's method** or to the **EDP objective function**.

Publications

Journal Articles

- [1] Ahmed Fanfakh, Jean-Claude Charr, Raphaël Couturier, Arnaud Giersch. Optimizing the energy consumption of message passing applications with iterations executed over grids. *Journal of Computational Science*, 2016.
- [2] Ahmed Fanfakh, Jean-Claude Charr, Raphaël Couturier, Arnaud Giersch. Energy Consumption Reduction for Asynchronous Message Passing Applications. *Journal of Supercomputing*, 2016, (Submitted)

Conference Articles

- [1] Jean-Claude Charr, Raphaël Couturier, Ahmed Fanfakh, Arnaud Giersch. Dynamic Frequency Scaling for Energy Consumption Reduction in Distributed MPI Programs. *ISPA 2014*, pp. 225-230. IEEE Computer Society, Milan, Italy (2014).
- [2] Jean-Claude Charr, Raphaël Couturier, Ahmed Fanfakh, Arnaud Giersch. Energy Consumption Reduction with DVFS for Message Passing Iterative Applications on Heterogeneous Architectures. *The 16th PDSEC*. pp. 922-931. IEEE Computer Society, INDIA (2015).
- [3] Ahmed Fanfakh, Jean-Claude Charr, Raphaël Couturier, Arnaud Giersch. CPUs Energy Consumption Reduction for Asynchronous Parallel Methods Running over Grids. *The 19th CSE conference*. IEEE Computer Society, Paris (2016).



- ▶ The proposed algorithms should take into consideration the **variability between some iterations**.
- ▶ The proposed algorithms should be applied to **other message passing methods with iterations** in order to see how they adapt to the characteristics of these methods.
- ▶ The proposed algorithms for heterogeneous platforms should be applied to heterogeneous platforms composed of **CPUs and GPUs**.
- ▶ Comparing the results returned by the energy models to the values given by **real instruments that measure the energy consumptions** of CPUs during the execution time.

Fin



Thank you for your listening

Questions?