femto-st SCIENCES & TECHNOLOGIES

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

PhD Dissertation Defense

Ahmed Badri Muslim Fanfakh

Under Supervision: Raphaël COUTURIER and Jean-Claude CHARR

University of Franche-Comté - FEMTO-ST - DISC Dept. - AND Team

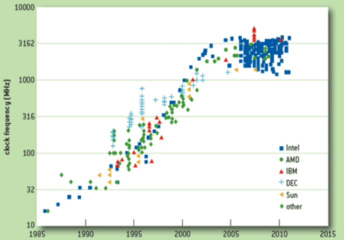
17 October 2016

Outline

Introduction and problem definition

Approaches to increase the computing power:

1) Increasing the frequency of processor

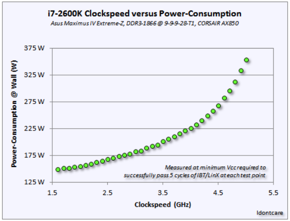


2) Increasing the number of nodes

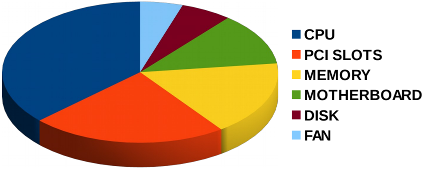
Introduction and problem definition

Processor frequency and its energy consumption

▶ The power consumption of a processor increases exponentially when its frequency is increased



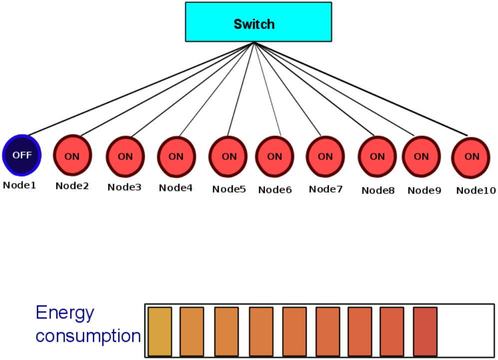
▶ The biggest power consumption is consumed by a processor in the computing node



Introduction and problem definition

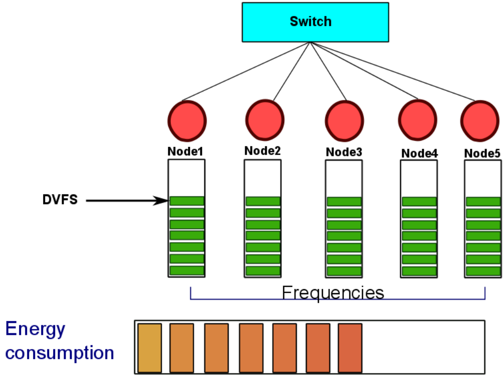
Techniques for energy consumption reduction

1) Switch-off idle nodes method



Techniques for energy consumption reduction

2) Dynamic voltage and frequency Scaling (DVFS)



Using the energy reduction method

Why we used DVFS method:

- It used to reduce the energy while keeping all node working, thus it is more conventional with parallel computing.
- It has a very small overhead compared to switch-off idle nodes method.

Challenge and Objective

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

Objective: Optimizing both energy consumption and performance of a parallel application at the same time when DVFS is used.

Contributions

First contribution

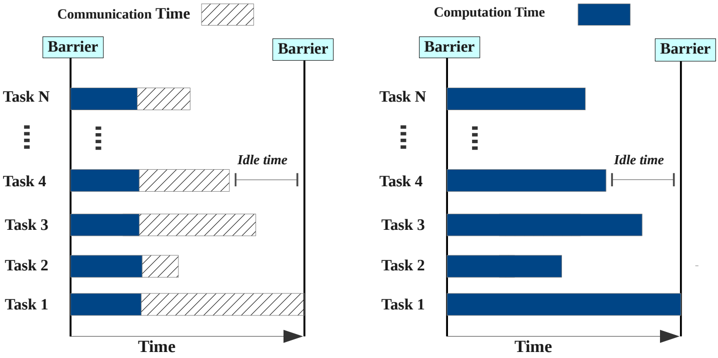
Energy optimization of homogeneous platform

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.

Parallel tasks execution over Homo. Platform



(a) Sync. imbalanced communications

(b) Sync. imbalanced computations

Figure: Parallel tasks on homogeneous platform

Energy model for homogeneous platform

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

$$P_d = \alpha \cdot CL \cdot V^2 \cdot F \tag{1}$$

Where:

α: switching activity

CL: load capacitance

V the supply voltage

F: operational frequency

$$P_s = V \cdot N_{trans} \cdot K_{design} \cdot I_{Leak} \tag{2}$$

Where:

V: the supply voltage.

N_{trans}: number of transistors.

K_{design}: design dependent parameter.

I_{leak}: technology dependent parameter.

Energy model for homogeneous platform

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

Rauber and Rünger's energy model

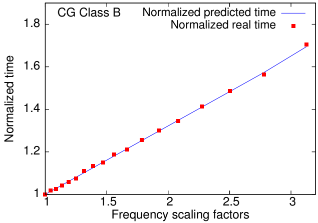
$$E = P_d \cdot S_1^{-2} \cdot \left(T_1 + \sum_{i=2}^{N} \frac{T_i^3}{T_1^2} \right) + P_s \cdot S_1 \cdot T_1 \cdot N$$

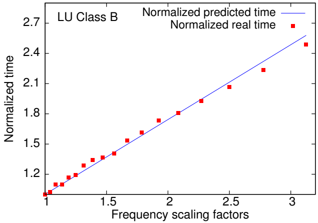
S_1: the max. scaling factor
P_d: the dynamic power
P_s: the static power
T_l: the time of the slower task
T_i: the time of the other tasks
N: the number of nodes

Performance evaluation of MPI programs

Execution time prediction model

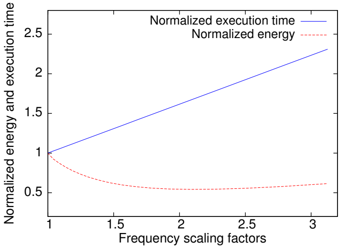
$$T_{new} = T_{MaxCompOld} \cdot S + T_{MinCommOld}$$



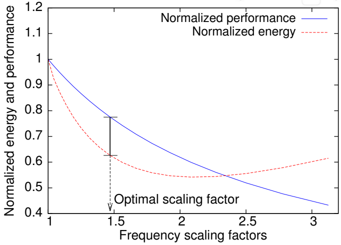


The maximum normalized error for CG=0.0073 **(the smallest)** and LU=0.031 **(the worst)**.

Performance and energy reduction trade-off



(a) Real relation.



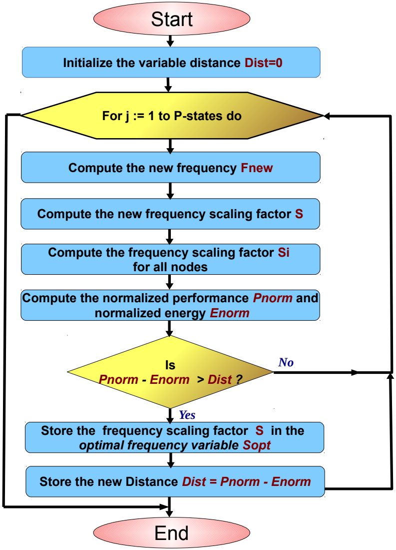
(b) Converted relation.

Where: $\color{blue}{Performance} = execution\ time^{-1}$

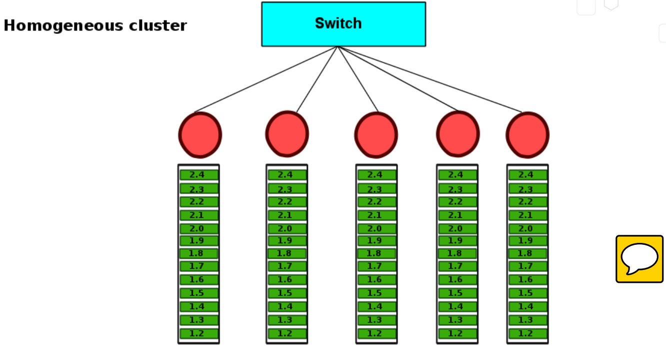
Our objective function

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

Scaling factor selection algorithm



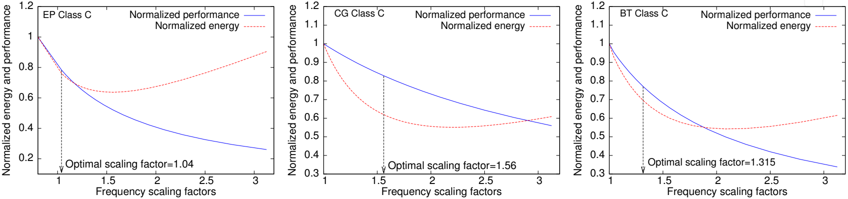
Scaling algorithm example

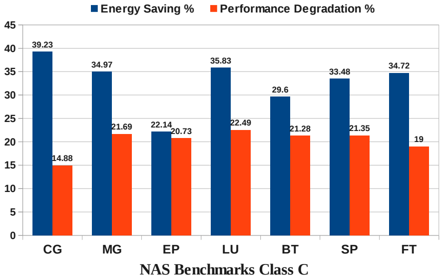


Experimental results

- 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

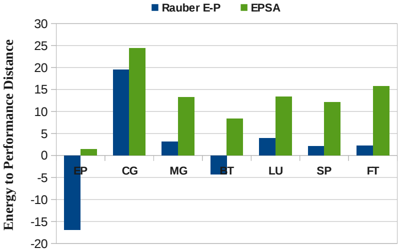




Results comparison

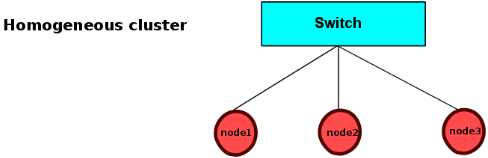
Rauber and Rünger's optimal scaling factor

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

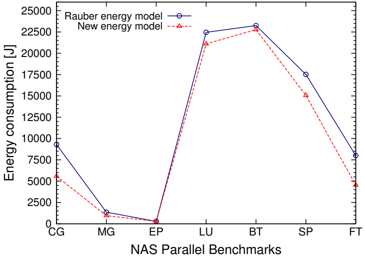


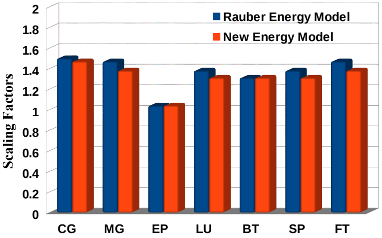
Comparing our method with Rauber and Rünger method for NAS benchmarks class C

The proposed new energy model



Comparing the new model with Rauber model





Contribution

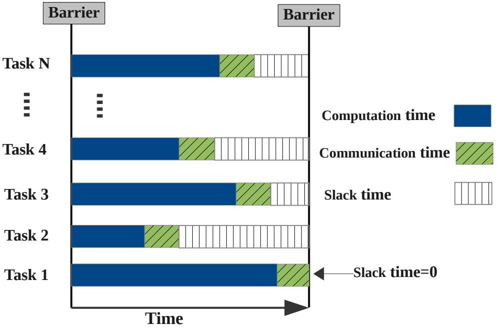
Second contribution

Energy optimization of Heterogeneous platform

Objectives

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

The execution time model



The execution time prediction model

$$T_{new} = \max_{i=1,2,\ldots,N} (TcpOld_i \cdot S_i) + \min_{i=1,2,\ldots,N} (Tcm_i) \qquad (3)$$

Where: $Tcm = communication\ times + slack\ times$

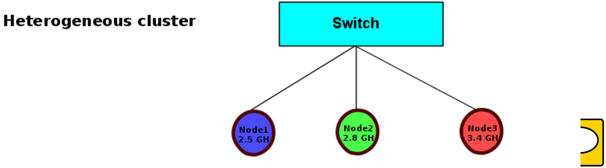
The energy consumption model

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

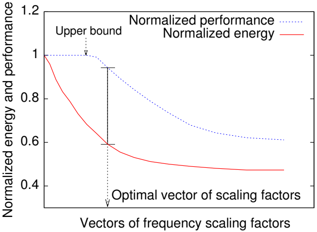
$$E = \sum_{i=1}^{N} (S_i^{-2} \cdot Pd_i \cdot Tcp_i) + \sum_{i=1}^{N} (Ps_i \cdot (\max_{i=1,2,\dots,N} (Tcp_i \cdot S_i) + \min_{i=1,2,\dots,N} (Tcm_i))) \tag{4}$$

<u>where</u>:
N : is the number of nodes.

The energy model example for heter. cluster



The trade-off between energy and performance



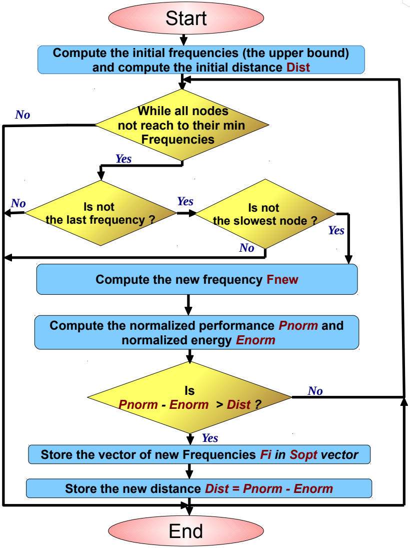
Step1: computing the normalized energy $E_{norm} = \frac{E_{reduced}}{E_{Max}}$.

Step2: computing the normalized performance $P_{norm} = \frac{T_{Max}}{T_{new}}$.

The tradeoff model

$$MaxDist = \max_{\substack{i=1,\dots,F \\ j=1,\dots,N}} (\overbrace{P_{norm}(S_{ij})}^{\text{Maximize}} - \overbrace{E_{norm}(S_{ij})}^{\text{Minimize}}) \quad (5)$$

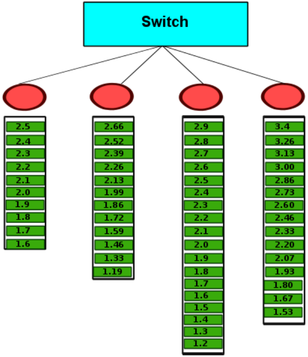
The scaling algorithm for heter. cluster



femto-st SCIENCES & TECHNOLOGIES

The scaling algorithm example

Heterogeneous cluster

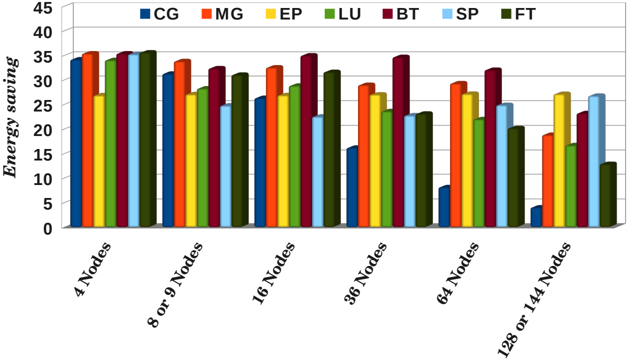


Experiments over heterogeneous cluster

- The experiments 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.
- We ran the benchmarks on different number of nodes ranging from 4 to 144 nodes.
- The total power consumption of the chosen CPUs is composed of 80% for dynamic power and 20% for static power.

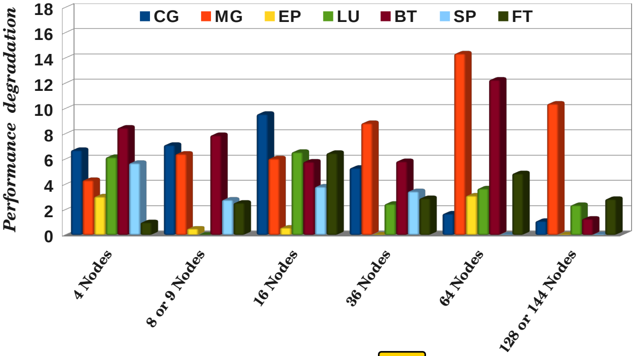
femto-st SCIENCES & TECHNOLOGIES

The experimental results



On average, it saves the energy consumption by 29% of NAS benchmarks class C executed over 8 nodes

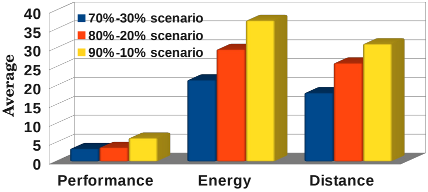
The experimental results



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

femto-st SCIENCES & TECHNOLOGIES

The results of the three powers scenarios



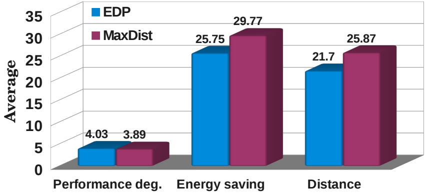
Selected frequency scaling factors for 8 nodes

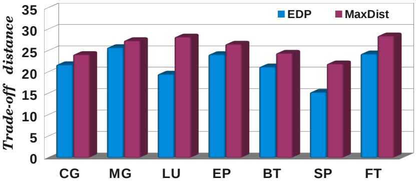
70%-30% Scenario	1.04	1.33	1.61	1.88	1.04	1.24	1.61	1.88
80%-20% Scenario	1.04	1.33	1.61	1.88	1.04	1.33	1.61	1.88
90%-10% Scenario	1.08	1.42	1.61	2.03	1.04	1.33	1.61	1.88

femto-st SCIENCES & TECHNOLOGIES

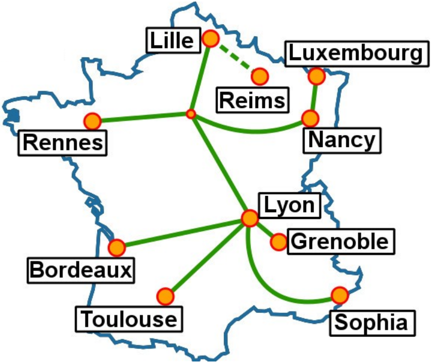
The comparing our method

The proposed method (MaxDist) was compared to the EDP algorithm that minimizes the *energy* $\times$ *delay* value.





Energy optimization of grid platform



10 sites distributed over France and Luxembourg

Performance, Energy and trade-off models

The performance model of grid

$$T_{New} = \max_{\substack{i=1,\ldots N \\ j=1,\ldots,M_i}} (T_{cpOld\,ij} \cdot S_{ij}) + \min_{j=1,\ldots,M_h} (T_{cm\,hj}) \tag{6}$$

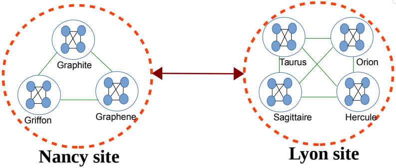
The energy model of grid

$$E = \sum_{i=1}^{N} \sum_{i=1}^{M_i} (S_{ij}^{-2} \cdot P_{d\,ij} \cdot T_{cp\,ij}) + \sum_{i=1}^{N} \sum_{j=1}^{M_i} (P_{s\,ij} \cdot T_{New}) \tag{7}$$

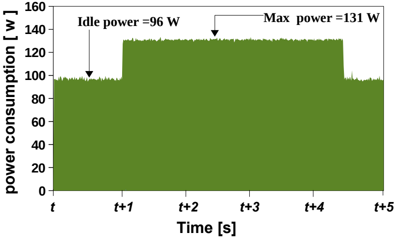
The trade-off model of grid

$$MaxDist = \max_{\substack{i=1,\ldots N \\ j=1,\ldots,M_i \\ k=1,\ldots,F_j}} (\overbrace{P_{Norm}(S_{ijk})}^{\text{Maximize}} - \overbrace{E_{Norm}(S_{ijk})}^{\text{Minimize}}) \tag{8}$$

Experiments over Grid’5000



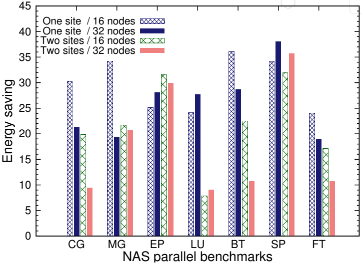
The experiments executed over one site and two sites scenarios



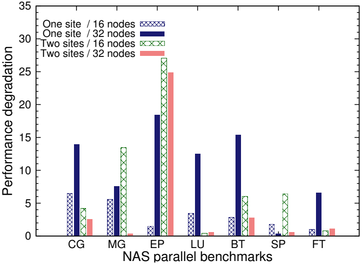
We used Grid’5000 power measurement tools

Experiments over Grid’5000

Execution the NAS class D on 16 nodes saves the energy by 30%

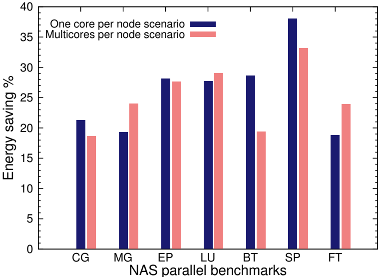


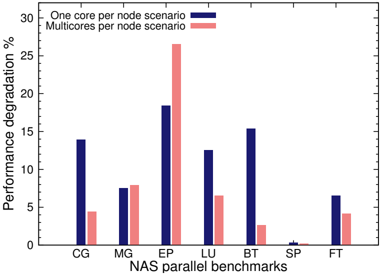
Execution the NAS class D on 16 nodes degrades the performance by 3.2%



Experiments over Grid'5000

One core and Multi-cores per node results:





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

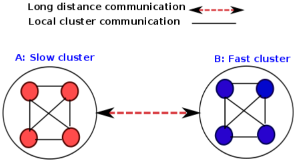
Continuation

Third contribution

Energy optimization of asynchronous applications

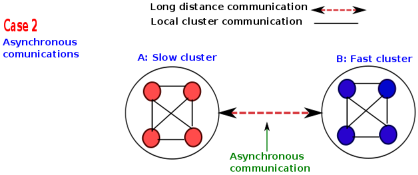
Problem definition

Execution the parallel iterative application with synchronous communications



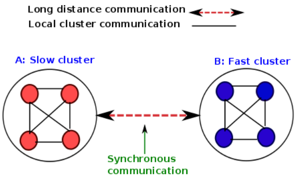
Problem definition

Execution the parallel iterative application with synchronous communications



Solution

Using asynchronous communications with DVFS



The performance models

The performance model of Asynch. Applications

$$T_{New} = \frac{\sum_{i=1}^{N} \sum_{j=1}^{M_i} (T_{cpOld_{ij}} \cdot S_{ij})}{N \cdot M_i} \tag{9}$$

The performance model of Hybrid Applications

$$T_{New} = \frac{\sum_{i=1}^{N} (\max_{j=1,\ldots,M_i} (T_{cpOld_{ij}} \cdot S_{ij}) + \min_{j=1,\ldots,M_i} (L_{tcm_{ij}}))}{N} \tag{10}$$

The energy consumption models

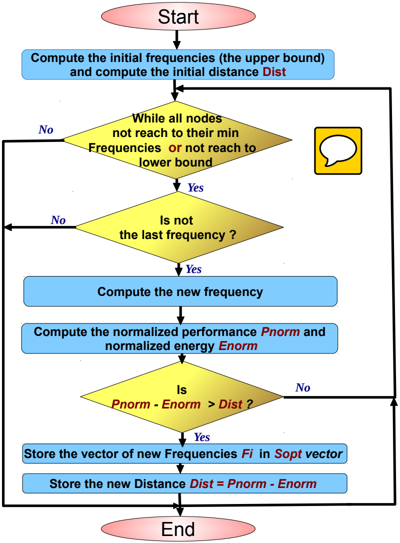
The energy model of Asynch. Applications

$$E = \sum_{i=1}^{N} \sum_{j=1}^{M_i} (S_{ij}^{-2} \cdot T_{cp_{ij}} \cdot (P_{d_{ij}} + P_{s_{ij}})) \tag{11}$$

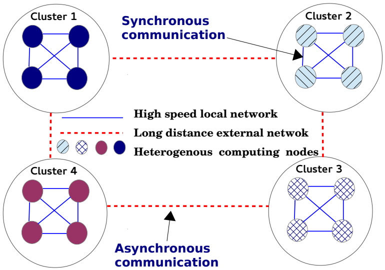
The energy model of Hybrid Applications

$$E = \sum_{i=1}^{N} \sum_{j=1}^{M_i} (S_{ij}^{-2} \cdot P_{d_{ij}} \cdot T_{cp_{ij}}) + \sum_{i=1}^{N} \sum_{j=1}^{M_i} (P_{s_{ij}} \cdot \\ (\max_{j=1,\dots,M_i} (T_{cp_{ij}} \cdot S_{ij}) + \min_{j=1,\dots,M_i} (L_{tcm_{ij}}))) \tag{12}$$

The scaling algorithm for Asynch. applications



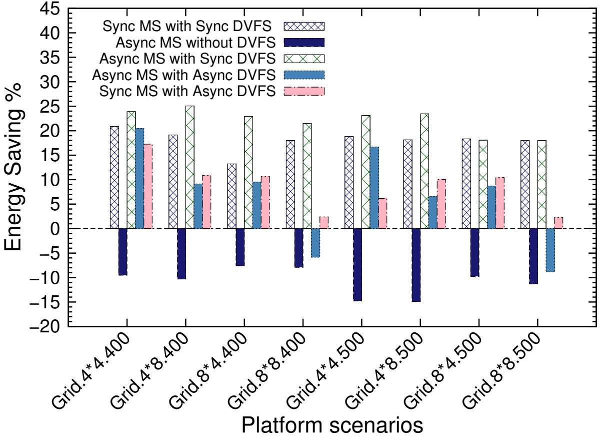
The experimental results



- Execution the iterative multi-splitting method over simulated Grid.
- Execution the iterative multi-splitting method over Grid'5000 test-bed.

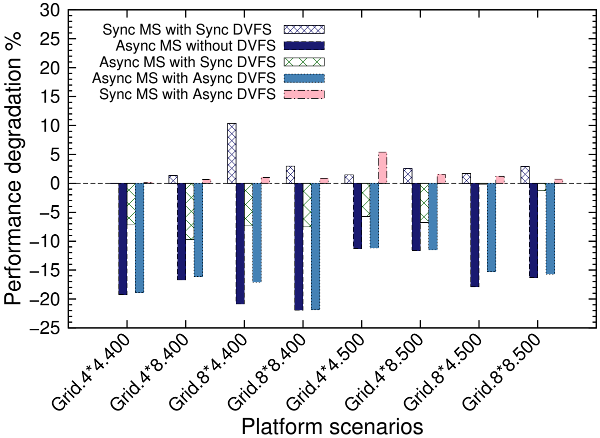
The simulation results

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



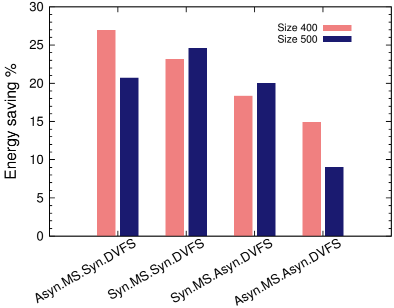
The average of energy saving = 22%

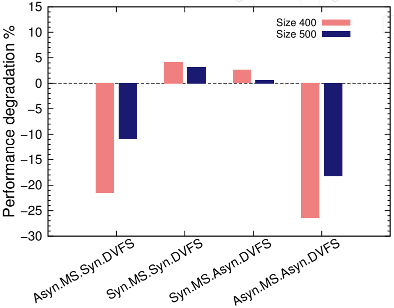
The simulation results



The average of speed-up = 5.72%

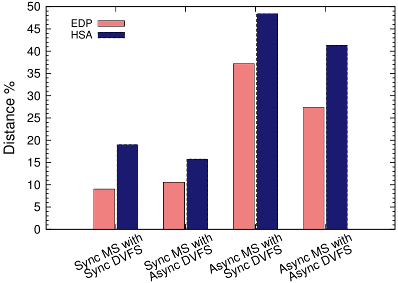
The Grid'5000 results

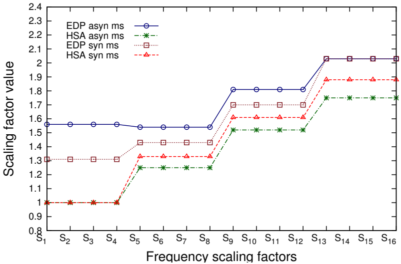




The energy saving = 26.93%, speeds up = 21.48%

The comparison results





Conclusions

- We have proposed a new energy consumption and performance models for synchronous and asynchronous parallel applications with iterations.
- The parallel applications with iterations were executed over different parallel architectures such as: homogeneous cluster, heterogeneous cluster and grid.
- We have proposed new objective function to optimize both the energy consumption and the performance.
- 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 Grid'5000 testbed.
- All the proposed methods were compared with either Rauber and Rünger method or EDP objective function.

Publication

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).

Perspectives

- We will adapt the proposed algorithms to 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

Thanks for Your Listening

Questions?