

Proposal of COST IC0804 Focus Group on:

## Computing and Energy Optimisation using CPUs and GPUs

Stephane Vialle

Supelec & UMI 2958 – GT-CNRS & AlGorille INRIA Project Team

April 5, 2011

### Motivations and objectives

GPUs have become efficient SIMD/vector coprocessors to speedup adapted intensive computing applications. However a CPU and GPU disseminate more electric power than only a CPU, and a significant speedup is required to achieve computations consuming less energy.

On a GPU cluster this phenomenon can become more complex. A distributed program composed of independent computations ("*embarrassingly parallel programs*") without communications between the computing nodes can reach very high performances when using GPUs. But if the distributed application requires many communications between the computing nodes, the final speedup compared to a pure CPU cluster can be weak. Internode communications are unchanged but data transfers between CPU and GPU have to be added to most of CPU-to-CPU communications. So the communication times increase, while the computation times decrease (using GPUs). Finally, some applications can scale less on a GPU cluster than on a CPU cluster, while the electrical dissipated power increase regularly with the number of nodes and GPUs. Similar problems can appear when using the small vector units of each CPU core (ex: SSE units). CPU electrical power dissipation can change, and depending on the speedup achieved the global energy consumption of the application can decrease or increase. Moreover, on a GPU it exist different memories that lead to different speed of the computations and to different energy consumption.

So, deciding to use a cluster with CPUs and GPUs in place of a pure CPU cluster should depend on many data and objectives. It should depend at least on the computation to achieve, on the number of available and used nodes, on the problem size, and on the maximal supported electrical power dissipation (each electrical system has a limit), and it should depend on the global objective of decreasing the execution time, or the energy consumption, or both. Finally, this decision can concern the user running its application, or the scheduler running a set of applications on a set of machines.

The different models, algorithms and implementations researched aim to offer different solutions in the pareto front for the multi-objective optimization problem of energy-efficient performance.

### Previous investigations

A possible approach consists in including several computing kernels and parallelisation schemes in an applicative code, and to choose at runtime the optimal configuration, function of the problem size, the number of computing nodes, and the cluster features. The right configuration can be chosen by the user, or by the application itself function of a performance model and a global user instruction: to run fast, or to minimize the energy consumption, or to track a compromise between speed and energy consumption. Finally, the scheduler can specify the configuration to run, or the instruction to respect, in place of the user, in order to tune the application execution and to globally optimize usage of the machine or of the set of machines of a data center.

In 2009 and 2010 we have developed and compared performances of different kernels and parallelisation scheme of some distributed applications on CPU clusters and GPU clusters. We have designed some energy performance models on our CPU+GPU clusters, and we attempt to implement

the application makes an automatic choice of its best configuration (a kind of computing and energy consumption auto-tuning).

To interface our multi-configuration applications with an intelligent CPU/GPU scheduler is the next issue we aim to address.

## **Preliminary working program and schedule**

### **May 2011 – December 2011:**

- Design and implement auto-tuning in previous test applications. Make a large set of benchmarks. Analyse and optimize auto-tuning model and mechanisms.
- Investigate how a scheduler can optimize the execution of multi-configuration distributed applications on a GPU cluster and on a set of CPU and GPU cluster. Design scheduling strategies and policies.

### **January 2012 – June 2012:**

- Implement and test some scheduling policies on different testbeds.

## **Deliverables**

### **December 2011:**

- Technical report about auto-tuning model, implementation and benchmarks.
- Technical report about strategies to schedule multi-configuration applications.

### **June 2012:**

- Final report about Computing and Energy Optimisation using CPUs and GPUs