Introduction

The tools for comprehension of High Performance Computing from theory, hardware and software give us the basis elements to go toward optimizations and benchmarking. We show that hybrid architectures seems to be the way to reach exascale in few years but many optimizations need to be done to fit the energy envelope and the ability to target all kind of applications. Our study focuses on the behavior of accelerators compared to classical processor applied on irregular applications. We showed that the downside of accelerators seems to be confronted to de-synchronization and irregular work. In this part we propose our metric putting forward accelerators behavior on irregular problems.

From the previous part we identify several bottlenecks and limitations in HPC. Those limitations are called *walls* against which nowadays architectures are confronted to reach exascale.

Memory Wall: This problem is targeted for the first time in [WM95]. The author explains that:

> We all know that the rate of improvement in microprocessor speed exceeds the rate of improvement in DRAM memory speed, each is improving exponentially, but the exponent for microprocessors is substantially larger than that for DRAMs.

The new release of accelerators also have an impact on the memory wall, the memory of the host processors and devices accelerators cannot be accessed directly and copies from one to the other are requested. The problems of coalescent access will also be address in this study.

Communication wall: The multiplicity of racks, nodes and accelerators add a communication wall. The network topology can never be perfect for all the kind of problems and even the fastest technologies are limited to the software handling. Limiting the big synchronizations steps, like in the BSP model, allows the system to be asynchronous and hide computation by communications. Unfortunately this is not applicable to all the applications and a huge care have to be taken to approach perfect scaling.

Power wall: The energy consumption of nowadays and future supercomputers is the main wall in HPC. Indeed, an exascale supercomputer could be construct using several petascale supercomputer but, with todays architectures, will require a full nuclear plant to operate. In this objective low energy consumption now and innovative architectures need to be find. Power wall can be of two kind: the energy to power the machine itself and the many nodes, processors and accelerators but also, and not the least, the energy requires to handle the heat generated by the machine. For the second part many new technologies arise with ideas like direct water cooling in the system racks.

Computational wall: The computational wall is a conjugation of the wall cited before. By increasing the memory wall, the energy consumption and the communications we can increase the overall computation power of the supercomputer. The limitation in computational power also cone from the fact that the Moore's law seems to be over. Vendors have more difficulties to shrink transistors due to physicals side effects. The frequency itself seems to reach its highest values due to the energy require to operate and the heat dissipated.

Irregular application: The irregularity in an application can have several definitions. This is defined by [JTB] as a problem which: can not be characterize a priori, is input data dependent and evolves with the computation itself. In [SL06] the author specifies that the work involve subcomputations which cannot be determined before and thus work distribution during runtime. The irregularity can then spread on all the layers of the resolution: The communications, the computation and the memory searches.

The first problem for our metric characterizes the behavior of accelerators and more specifically GPUs focusing on irregular computations. We choose the problem of Langford, known in our laboratory, and show how we can take advantage of accelerators for this kind of problem.

The second problem we choose is focused on irregular memory accesses and communications. It is based on a benchmark we presented in the previous part, the Graph500 benchmark.

Bibliography

[JTB] Christopher J Riley High-Performance Java and Gabrielle Keller Transformation-Based. Irregular parallel algorithms.

[SL06] Michael Süß and Claudia Leopold. Implementing irregular parallel algorithms with openmp. In *European Conference on Parallel Processing*, pages 635–644. Springer, 2006.

[WM95] Wm A Wulf and Sally A McKee. Hitting the memory wall: implications of the obvious. *ACM SIGARCH computer architecture news*, 23(1):20–24, 1995.