Reconfigurable hardware Molecular Dynamics Acceleration on Cray XD1: the system approach

B. Tsakam-Sotché

XLBiosim SA
EPFL-PSE-C
CH-1015 Lausanne, Switzerland
Brice-herve.tsakam-sotche@epfl.ch

ABSTRACT

Molecular dynamics are most realistic when they include explicitly the solvent. Standard computers need months to simulate modest timescales. More than 90% of the computing time is consumed in the evaluation of force fields. Cray XD1 systems help solving these problems by the mean of reconfigurable hardware acceleration.

The presentation contains the performance analysis of the molecular dynamics software Gromos and the acceleration of force field evaluation (coulombic, lennard-jones, reaction field interactions). The molecular dynamics acceleration is presented with a system approach including the acceleration hardware in relationship with the application level software Gromos.

1. Introduction

Popular molecular modeling packages such as Gromos [1] or Gromacs [5] are being used for atomistic level studies of entities such as peptides, sugars, membranes, DNA strands, etc. The parameterization of the software while modeling a specific molecular configuration is not trivial. Another challenge is related to running the simulation. Indeed to simulate 100ns it still take in the order of 10 weeks even for molecular systems that contain a relatively modest number of 100 thousands atoms, solvent included. It is well known that the most CPU time consuming part in these simulations is the evaluation of non-bonded interactions, most severely in the case of explicit solvent representation. Computers represent atoms with their spatial coordinates and physical properties such as coulombic charges, masses, etc. The key parameter for one nonbonded interaction is related to the relative positions or inter-atom distances and the physical properties of the involved atoms. One such interaction is represented as a pair. The process of evaluating the non bonded interaction is structured in two parts: the enumeration of the pairs or the construction of the pair list and the evaluation of the interacting forces for each pair. The evaluation of non bonded interactions is known to consume more than 90% of the computation time because of the large number of pair wise non bonded interactions and the actual function evaluation that is non linear. Thus computing systems providers and molecular dynamic software providers use different strategies to speed up the process. We will consider the cut-off approach that is a simple but proven technique to reduce the number of interactions and still obtain valuable results from the simulation.

The latest release of Gromos05 and alternatively the latest release of Gromacs3.3 share a similar approach that we will designate grid search for speeding up the building pair list which scales linearly with the cube of the cut-off distance. Other packages such as Gromos96 use an exhaustive examination of atoms to build the pair lists, which has scales squarely with the number of atoms.

Recently, we have seen the emergence of Reconfigurable hardware as a mean to accelerate scientific computing. Systems such as the Cray XD1 that have a built in FPGAs or even commercial PCs augmented with FPGA boards provide an alternative platform for running such applications. For these systems, a key parameter is the bandwidth available on the system to move data back and forth between the main processor and the FPGA accelerator.

In accelerated systems, the workload is shared between the main processor and the accelerator. The proper coordination of the system component is essential for the overall performance: the application must be properly partitioned in order to have a fast communication and a compact design on the FPGA side.

XLBiosim's MD acceleration module implements LJ6-12 interaction forces evaluation with reaction field at 85Mhz internal clock rate. The design is structured such that several acceleration modules may be contained in one FPGA, provided that there are enough I/O pins to connect external memories that feeds the design and accepts the results. The I/O pins can be the main limitation for the system's performance but in this paper, we focus on the interplay between:

- the pair list building process
- the interaction forces evaluation process
- the communication process

2. Analytical performance model

We propose a simple model to choose trade off regarding the key system components and processes. Our approach is to observe the impact of each component on the overall performance and provide a clear indication of which system architecture is more harmonious for which kind of molecular dynamics task. The goal of the model is to apprehend the issues that may appear when the molecular system's size increases. We are most interested in determining the ideally scaling architecture for molecular dynamics applications.

2.1. Model parameters

We consider atoms in a box of divided in equally sized cells thus forming a grid. We assume for the sake of simplicity that the density of atoms per cell is constant. When the number of atoms grows, we also let the cube grow. We make the assumption to be in the ideal case where the cell size is constant and the system grows by increasing the number of cells (1 full plane at a time).

The parameters of the simulation are the following.

N_a: the number of atoms

D: the atom density per unit of volume

Ncells: the number of cells

R_c: the cut-off distance

NnonBond: the number of non-bonded interacting pairs

2.2. Computation and communication performance

We are interested in the computational load that is related to two procedures:

P_{pairs}: the pair lists building procedure

Pnb_{force}: the non-bonded interaction forces evaluation procedure

When an acceleration co-processor is used, another task is the communication between the main processor and the co-processor:

C_{ac}: communication of the data from the main processor to the acceleration co-processor

C_{proc}: communication of the results from the acceleration co-processor to the main processor

2.3. System scalability in the number of atoms

As the number of atoms increases with constant density, so does, with the density as linear factor, the number of cell and the box size.

From previous works on molecular dynamics simulations, we know that while using the cut-off approach, the P_{pairs} workload grows like $cpairs_{EXH}$ x N_a^2 with the exhaustive search and linearly like $cpairs_{GRID}$ x D x R_c^3 with the grid search where $cpairs_{EXH}$ and $cpairs_{GRID}$ are constants.

The number of interacting pairs is averaged to NnonBond which is proportional to the density of atoms, the cut-off distance and the system size, thus following a linear function of the number of atoms: cnbond x D x R_c^3 x N_a where cnbond is a constant. With the constant density hypothesis, the workload for Pnb_{force} grows like cforce x NnonBond and with the number of atoms, resulting in a linear function of the number of atoms cforce x cnbond x D x R_c^3 x N_a. The cforce constant actually depends on the computing architecture thus we will distinguish $cforce_{cpu}$ for the main cpu and $cforce_{cop}$ for the accelerator. We assume that the exhaustive search and the grid search are both accurate and result in the same pair lists. The amount of data that is moved from the main processor to the co-processor at run time is essentially the coordinates of the atoms pairs and thus just the amount of data per pairlist multiplied by NnonBond: cmem x NnonBond, where cmem is constant. The communication from the co-processor to the main processor is required to move data back and is thus related to the number of primary atoms in the pairlists multiplied by the memory space required to store the forces applying on one atom. Again it is proportional to NnonBond : cres x NnonBond where cres is constant.

2.4. Performance of the sequential process

Our model is simplistic in the sense that it assumes totally sequential operation:

1. the main CPU builds the lists of non bonded interacting pairs of atoms or pair lists
2. the pair lists are sent through the bus to the co-processor
3. the non bonded interactions are computed
4. the atom based results are sent back to the main processor

An objective performance metric for the procedure of interest is the total time, including computation and communication time between the main CPU and the co-processor (COP):

$$\mathbf{T = t(P_{pairs}) + t(Pnb_{force}) + t(C_{ac}) + t(C_{proc})} \qquad \text{(eq 1)}$$

where :

t(P_{pairs}) is the processing time required to build pairs of non bonded interacting atoms

t(Pnb_{force}) is the processing time required to evaluate the non bonded interaction forces (or potentials)

t(C_{ac}) is the communication time required to transmit the pairs of non bonded interacting atoms to the co-processor's memory

t(C_{proc}) is the communication time required to transmit the non bonded forces that apply to the (primary) atoms which were involved in interacting pairs

The communication terms are removed if all the calculations are done on the one processing unit.

In the exhaustive search case, this formula turns into:

$$\mathbf{T_E = (cpairs_{EXH} \times N_a^2) / speed_{CPU} + (cforce_{COP} \times NnonBond) / speed_{COP} + (cmem \times NnonBond) / BW_{BUS} + (cres \times NnonBond) / BW_{BUS}} \qquad \text{(eq. 2)}$$

In the grid search case, we have

$$\mathbf{T_G = (cpairs_{GRID} \times D \times R_c^3 \times N_a) / speed_{CPU} + (cforce_{COP} \times NnonBond) / speed_{COP} + (cmem \times NnonBond) / BW_{BUS} + (cres \times NnonBond) / BW_{BUS}} \qquad \text{(eq. 3)}$$

where $speed_{COP}$ and $speed_{CPU}$ are the internal clock frequencies for the main CPU, respectively the acceleration co-processor (COP), and BW_{BUS} is the inter processor bus bandwidth.

3. Performance analysis of the system with attached co-processor

We apply the model for finding the best system compositions for and pointing the type of architecture that is most promising in terms of scalability. For the application, we also feed the model with realistic values for $speed_{COP}$, $speed_{CPU}$, BW_{BUS}, experimental values for NnonBond, $cforce_{cop}$, cmem , cres, cpairs, etc. This model reflects properties of the system that are valid only on relatively large time scale since the burst behavior of the bus or the cache performance are not considered. Our analysis explores the impact on the overall performance of variances of the co-processor (COP) and the BUS speed.

3.1. Assumptions

The performance analysis model is parametrized using bus bandwidth based on commercial buses information as shown in the following table [2], [3].

Bus	**Frequency / Bit width**	**Bandwidth [MB/s]**
PCI	33MHz / 32	128
PCI-X*	133MHz / 64	1024
XD1-Hypertransport	200** (400) / 64	1600* (3200)

Table 1: Bus bandwidth

*we use the PCI-X specifications for our base case

**the hypertransport bus is bi-directional thus the effective bus bandwidth is 3.2GB/s however, our model assumes sequential operation and does not exploit the bi-directional feature.

The relationship between the number of atom and the number of interactions (NnonBond), the average pair list size are taken from an experimental case of water simulation with gromos05, with a cut-off distance of 1nm.

Number of atoms	**Average number of interacting pairs**	**Average pair list size**
90,000	1,500,000	75

Table 2: number of atoms and pairs

We assume a perfectly linear relationship among the 2 first columns, the pair list size being proportional to the cut-off distance to the power 3.

The remaining constants are evaluated based on a simulation run with 90,000 atoms.

Constant	Value
cforceCOP*	2
Cmem	24
Rc (nm)	1
Cpairs	2
Cforce	24
D	1
cnbond	15
cres	0.16

Table 3: Model constant parameters

*This constant is related to one acceleration module.

Actually each XLBiosim acceleration module at 85MHz, optionally featuring pair lists support, is capable of handling as many pairs as a general purpose processor (with standard math libraries on linux) with a frequency of 2.2 GHz in the same time frame. A single FPGA may contain several such modules and achieve 10 times speedups or more (depending on CPU cache performance) versus a generic processor of the same generation, as measured with virtex2 standard densities and XST synthesis in ISE6 flow. However, the number of modules inside the FPGA may not be the most important parameter in terms of overall system performance.

3.2. Exhaustive search

We first exploit the model to compare the impact of the pairlist building algorithm, assuming a PCI-133Mhz bus.

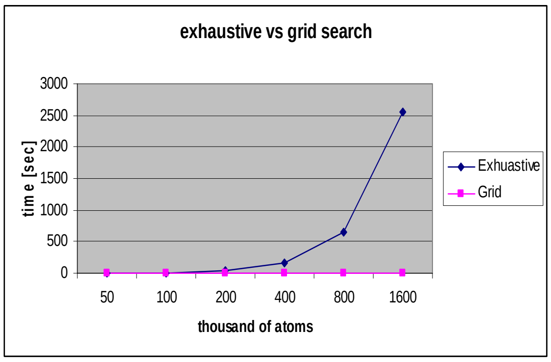

Figure 1: scalability of exhaustive search vs grid search

As expected [4], the squarely complexity make the exhaustive search unpractical for large configuration from 150 thousand atoms or more in our model. If the actual number of atoms where the exhaustive search becomes an issue depends also on other parameter in this model, the trend will occur in any case when the number of atoms increases. This indicates that for large configurations grid search is required. The grid search is also very favorable for parallel implementation since it is relatively easy to conceive allocating regions of the grid to different computing nodes. Indeed this feature was implemented in the latest Gromos05.

The model shows that accelerating the non bonded interactions evaluation with a co-processor may be useless when:

- an exhaustive search is used to build pair lists and,
- the number of atom is relatively large (beyond 100 thousand atoms)

Drawing conclusions from this projection, we will now look closer at the variations in the other parameter, in the grid search case only.

3.3. Grid search

Co-processor speed

The first parameter under consideration with our simple model is the number speed of the co-processor (speed$_{COP}$) or equivalently, the number of acceleration modules in the FPGA co-processor. This parameter directly impacts the co-processor's raw acceleration capability and thus very often drains a lot of efforts from developers and a lot of attention from users. However, the model shows that just having a faster co-processor is not necessarily sufficient for significantly improving the system's performance. In the following graph, we have a 4 fold performance increase of the co-processor and we benefit only a 35% performance improvement that is relatively stable with respect to the number of atoms.

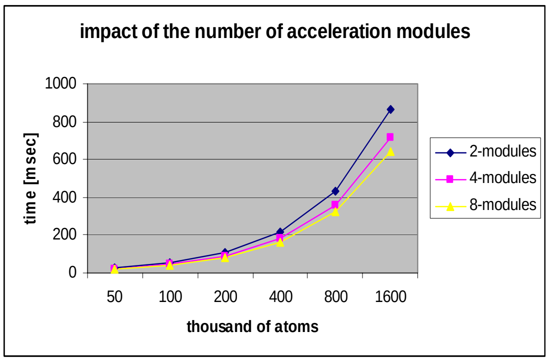

Figure 2: scalability vs co-processor speed

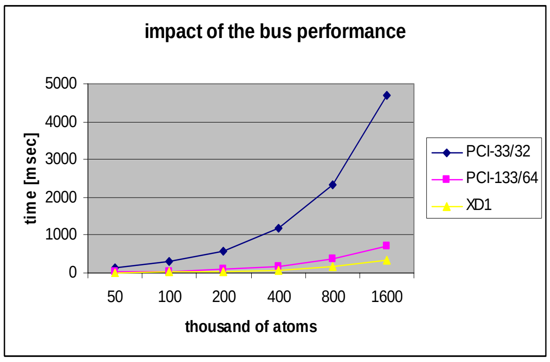

Figure 3: scalability vs bus bandwidth

This observation shows that in some systems, the performance of the acceleration co-processor is not leveraged because the bottleneck is not anymore in the non bonded interaction evaluation but elsewhere in the system. Actually, provided that the evaluation of those interaction takes about 90% of the computation time, we should fully benefit from any acceleration from the co-processor up to a 10 times speedup. Only at that point, the initial 90% work load due to the non bonded interactions would require just as much time as the initial 10% work load due to all the other procedures. Thus the extraction of the non bonded interactions evaluation and its porting on another device created a new bottleneck in the system!

Communication bus bandwidth

In the computing system that includes the main CPU and a specialized co-processor (COP), the new element that will affect the performance is the CPU to co-processor (COP) bus. We considered the case of a 4 modules instantiation in the FPGA, and let the bus' bandwidth vary. Our model is very optimistic in that it assumes that we are able to harness extremely efficiently the bandwidth of all the buses. We observe that the bus bandwidth has a great impact not only on the computing system's performance but also on its scalability, while using grid search. Indeed, the system with a PCI-33Mhz bus performs poorly. This type of bus should not be used for our target application. We observe a modest improvement, for large systems, while using XD1's hypertransport in comparison with PCI-133Mhz.

This figure calls for faster bus architectures. This part of the system is also critical for parallel implementations: the data must be transmitted quickly from the main CPU to the co-processor and also from main CPU to remote main CPU or from main CPU to remote co-processor.

On the developer side, some significant improvements may also be obtained by having more compact pair lists representations. While this is of modest interest in pure CPU based systems, it is most relevant for co-processor accelerated systems. Indeed, having 8 times more compact pair lists has the same impact as from switching from PCI-33MHz to PCI-133MHz!

4. Discussion

While tackling FPGA based acceleration, the focus shall not be only on the speed and density of the acceleration module, but also on the communication between the main CPU and the co-processor. Under certain bus conditions, improving the co-processor's performance is ineffective. Indeed, one may systematically consider implementing dedicated software drivers and hardware blocks that compress, respectively decompress the transmitted data. This means devoting main CPU cycles and FPGA area for improving communication performance. Practical libraries and hardware blocks implemented for this purpose are not known to the author at the time of writing.

In the case of parallel implementations fast communication are required not only between the main processor and the co-processor, but also between the main CPU and a remote CPU. In this respect, with a sizable number of AMD processor, each attached to an assigned high density FPGA and the whole connected on a hypertransport fabric, the XD1 (with FPGA) systems provides a unique combination of critical features for accelerated computation systems.

Acknowledgements

The Author acknowledges the Gromos group at ETHZ, led by Pr van Gunstered and Pr Hünenberger for enlightening discussions and key insights regarding Gromos and grid search methods.

References

[1] Christen, M., **Hünenberger**, P.H., Bakowies, D., Baron, R., Bürgi, R., Geerke, D.P., Heinz, T.N., Kastenholz, M.A., Kräutler, V., Oostenbrink, C., Peter, C., Trzesniak, D. & van Gunsteren, W.F. The GROMOS software for biomolecular simulation: GROMOS05 *J. Comput. Chem.*, **26**, 1719-1751 (2005).
[2] www.cray.com
[3] D. Abbott, „PCI Bus Demystified“, LLH Technology publishing, 2000
[4] Heinz, T.N. & **Hünenberger**, P.H. A fast pairlist-construction algorithm for molecular simulations under periodic boundary conditions. *J. Comput. Chem.*, **25**, 1474-1486 (2004).
[5] www.gromacs.org

About the author(s)

Brice Tsakam-Sotché graduated from EPFL SCC in 1999. After a stay with Panasonic Research in Santa Barbara he joined Acterna, a leading provider to telecom testing equipment then Transwitch a fabless semiconductor company. His technical background was completed with the Excecutive Master in Management of Technology from HEC Lausanne. Then he joined XLBiosim, a leading startup in the field of acceleration systems for molecular modeling hosted by the scientific parc at Federal Institute of Technology, Lausanne.
Contact: Brice-herve.tsakam-sotché@epfl.ch