Asociación Argentina
de Mecánica Computacional

Mecánica Computacional Vol XXXV, págs. 467-482 (artículo completo)
Martín I. Idiart, Ana E. Scarabino y Mario A. Storti (Eds.)
La Plata, 7-10 Noviembre 2017

BOOSTING MATERIALS SCIENCE SIMULATIONS BY HIGH PERFORMANCE COMPUTING

Emmanuel N. Millán[a], Carlos J. Ruestes[a], Nicolás Wolovick[b] and Eduardo M. Bringa[a]

[a]*CONICET and Facultad de Ciencias Exactas y Naturales, Universidad Nacional de Cuyo, Mendoza, Argentina*

[b]*Universidad Nacional de Córdoba, Córdoba, Argentina*

Keywords: High Performance Computing, Molecular Dynamics Simulations, performance analysis, accelerators.

Abstract. Technology development is often limited by knowledge of materials engineering and manufacturing processes. This scenario spans across scales and disciplines, from aerospace engineering to MicroElectroMechanical Systems (MEMS) and NanoElectroMechanical Systems (NEMS). The mechanical response of materials is dictated by atomic/nanometric scale processes that can be explored by molecular dynamics (MD) simulations. In this work we employ atomistic simulations to prove indentation as a prototypical deformation process showing the advantage of High Performance Computing (HPC) implementations for speeding up research. Selecting the right HPC hardware for executing simulations is a process that usually involves testing different hardware architectures and software configurations. Currently, there are several alternatives, using HPC cluster facilities shared between several researchers, as provided by Universities or Government Institutions, owning a small cluster, acquiring a local workstation with a high-end microprocessor, and using accelerators such as Graphics Processing Units (GPU), Field Programmable Gate Arrays (FPGA), or Intel Many Integrated Cores (MIC). Given this broad set of alternatives, we run several benchmarks using various University HPC clusters, a former TOP500 cluster in a foreign computing center, two high-end workstations and several accelerators. A number of different metrics are proposed to compare the performance and aid in the selection of the best hardware architecture according to the needs and budget of researchers. Amongst several results, we find that the Titan X Pascal GPU has a $\sim$3 x speedup against 64 AMD Opteron CPU cores.

1 INTRODUCTION

Molecular dynamics is a method that allows for the simulation of N-body problems, such as the physical movements of atoms and molecules, by numerical integration of the equations of motion of a set of particles in the material. The motion of the particles (atoms) is normally assumed to be classical and governed by Newton's laws, where each atom is allowed to interact with a certain number of neighboring atoms by means of an interatomic potential Allen and Tildesley (1989). The technique enabled many important contributions to the study of condensed matter and materials science Bringa et al. (2005, 2006).

High Performance Computing (HPC) allows to solve computational complex problems that without the use of parallel algorithms or clusters of computers would be impossible to solve due to time constraints. Multiple solutions exists to implement HPC software and hardware: specially developed supercomputers like the ones present in the Top 500 Supercomputers list (TOP500.org (2017)), small Beowulf-type clusters and, in the last $\sim$10 years, the use of Graphics Processing Units (GPUs) as accelerators. Generally, to access a supercomputer with thousands of CPU cores is restricted to researchers that belong to the institution that owns the supercomputer or have a research project in collaboration with local researchers of that institution. An attractive alternative is to install a small cluster of computers or use GPU as accelerators in workstations (or a combination of both). GPUs have been successfully used in multiple research works: in Molecular Dynamics Kohnke et al. (2017); Brown et al. (2011); Anderson et al. (2008), Settlement Dynamics Millán et al. (2015), N-body simulations Huang et al. (2016), statical/machine learning Gruber and West (2016); Cybenko (2017).

By taking one of the classical problems of contact mechanics, that is the spherical indentation of a metal, as a case study for a molecular dynamics simulation, this work presents a performance comparison between several HPC clusters and GPUs. The purpose of this work is to demonstrate how materials science simulations can be significantly accelerated and how this acceleration can be achieved even with desktop computers, provided the right hardware is chosen. Section 2 presents a detailed description of the indentation problem, the simulation model (Sec. 2.1) and its results (Sec. 2.2) from a materials science point of view. Section 3 provides the computational benchmarks performed with the MD simulation. The description of the metrics used for the benchmarks is described in section 3.1. Section 3.2 provides a detailed description of the software used and the hardware tested and section 3.3 summarize the results and discussion of the performance tests. The main conclusions are outlined in Section 4.

2 MOLECULAR DYNAMICS SIMULATION OF INDENTATION

Indentation is an important experimental technique, both for research and technology. It is not only used to gather information about the elastic modulus and hardness of a material, but also provides insights into cracking mechanisms, fracture toughness, strain-hardening, phase transformations, creep, and the mechanical response of superhard thin films Fischer-Cripps (2004); Armstrong et al. (2013). Since the seminal contribution by Kelchner et al. (1998), molecular dynamics (MD) simulations have been extensively applied to study plasticity mechanisms during indentation processes, and therefore it is the simulation method chosen in this work.

2.1 Model

Molecular dynamics simulations were carried out with the Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS) Plimpton (1995). The material chosen for this study

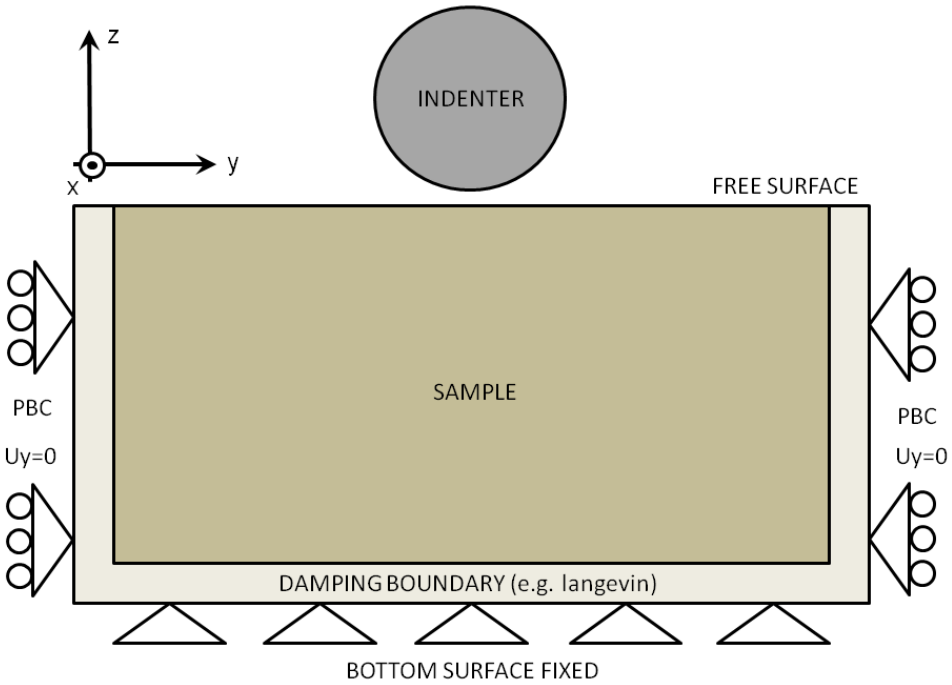

Figure 1: Simulation setup.

is tantalum, modeled by means of an interatomic potential of the extended Finnis-Sinclair type Dai et al. (2006).

The simulation setup is shown in Fig.2a. The indenter was modeled as rigid hemispherical indenter interacting with the atoms in the target with a harmonic potential Kelchner et al. (1998), $Vi = K(R - r_i)^2$, with R the indenter radius and r_i the position of atom i, and with $K = 1000$ eV/nm^2 being the specified force constant. Simulations were conducted in a displacement-controlled fashion by applying a constant penetration rate of 20 m/s, corresponding to $\sim$ 1/170 C_0, where C_0 is the directionally averaged sound velocity for Ta. Five different tip radius were tested, namely 3.2, 4.0, 5.2, 6.4, and 8.0 nm. The simulation box was varied according to indenter size: $18 \times 18 \times 16$ nm^3 ($\sim$0.3 million atoms), $23 \times 23 \times 20$ nm^3 ($\sim$0.6 million atoms), $30 \times 30 \times 25$ nm^3 ($\sim$1.3 million atoms), $36 \times 36 \times 32$ nm^3 ($\sim$2.4 million atoms), and $46 \times 46 \times 40$ nm^3 ($\sim$5 million atoms), respectively. (Note: The aforementioned simulations will be referred with numbers 1 through 5 in the computational benchmarks section (Sec. 3), simulation 1 being the biggest (8.0 nm radius) and simulation 5 being the smallest (3.2 nm radius)). Periodic boundary conditions were applied in the directions perpendicular to the indenting one, and the two bottom most layers of the sample were fixed to prevent movement of the substrate. To avoid thermally activated mechanisms, the simulations were performed at 0 K by applying a Langevin bath to the sides and bottom of the simulation domain. This boundary condition also serves as a damping boundary to prevent stress wave reflections. The entire sample was energetically minimized prior to nanoindentation. The study was restricted to the indentation of single crystals in [100] orientation.

Defective structures were filtered by Common Neighbor Analysis (CNA) Tsuzuki et al. (2007) and by means of the Dislocation Extraction Algorithm Stukowski and Albe (2010), and visualized using OVITO Stukowski (2010).

2.2 Results

Load-penetration curves for the five different tips (3.2-8.0 nm) are presented in Fig.2a. The curves are characterized by an elastic stage and, once a critical penetration is achieved, a sudden load drop is seen, corresponding to the onset of plasticity. The plastic stage of the curve then continues as that of a typical indentation experiment. The point at which the onset of plasticity occurs and the slope of the elastic-plastic stage increase with increasing radius, as revealed in fig.2a. This is something expected, and can be explained by Hertz study of the elastic interaction between a sphere of radius R and an elastic isotropic solid Hertz (1882). The force F imparted by the sphere perpendicular to the surface is proportional to the radius of the indenter and related to the displacement h into the surface through the following relationship:

$$F = \frac{4}{3}\, E^*\, R^{1/2}\, h^{3/2}. \tag{1}$$

E^* is often called the "combined indentation modulus" of the system, and is given by:

$$\frac{1}{E^*} = \frac{1-\nu^2}{E} + \frac{1-\nu'^2}{E'}, \tag{2}$$

where E, ν, E' and ν' are the elastic modulus and Poisson's ratio of the surface and indenter, respectively. The indenter used in our simulations Kelchner et al. (1998) is assumed to be rigid and, hence, the second term in Eq.2 is dropped. Also, the indenter used here is frictionless, analogous to a Hertzian indenter and, therefore, there are no forces in the direction tangential to the tip. Based on the loading penetration curves, the elastic modulus is estimated in the range of 120 - 130 GPa, in agreement with the value derived from eq. 2 for the potential used Ruestes et al. (2014).

The hardness of the material is defined as the contact pressure that, once critical indentation depth has been exceed and a possible load drop has occured, stays rather constant with increasing indentation. By relating the evolution of the load to the contact area, the attained contact pressures were determined, rendering a hardness of the order of 12 - 14 GPa, which in spite of being higher than experimental values Biener et al. (2007); Rajulapatit et al. (2010), it is in fact consistent with typical high values for MD simulations of perfect crystals Ziegenhain et al. (2010).

For all the radii studied here (3.2-8.0 nm), the onset of plasticity occurs by the homogeneous nucleation of dislocations, namely shear loops. Past the initial defect-nucleation stage, plasticity evolves by means of dislocation activity. Dislocation loops evolve in $\{110\}$, $\{112\}$ and $\{123\}$ planes with $\langle 111\rangle$ directions, consistent with the three slip systems for bcc metals Meyers and Chawla (2009). The screw segments of the dislocation loops cross-slip, leading to the generation of prismatic loops, which further move along $\langle 111\rangle$ directions. The whole process of prismatic loop formation is similar to the lasso mechanism, described in detail by Remington et al. (2014). The resulting structures at the end of the simulation for the 8.0 nm radius indenter are presented in Fig.2b, in which only atoms in defective positions are seen.

By means of the dislocation extraction algorithm Stukowski and Albe (2010), the atoms in defective positions were analyzed and dislocations were extracted from the structure, see fig. 2c also corresponding to the 8.0 nm radius indenter at the end of the simulation. Lines in green color correspond to dislocations with Burgers vector 1/2$\langle 111\rangle$, while those in violet correspond to dislocations with Burgers vector in $\langle 001\rangle$ direction that occur as a reaction of the former. Their presence in bcc metals is rare due to the high energy needed for the reaction to occur. Red arrows indicate the direction of the Burgers vector.

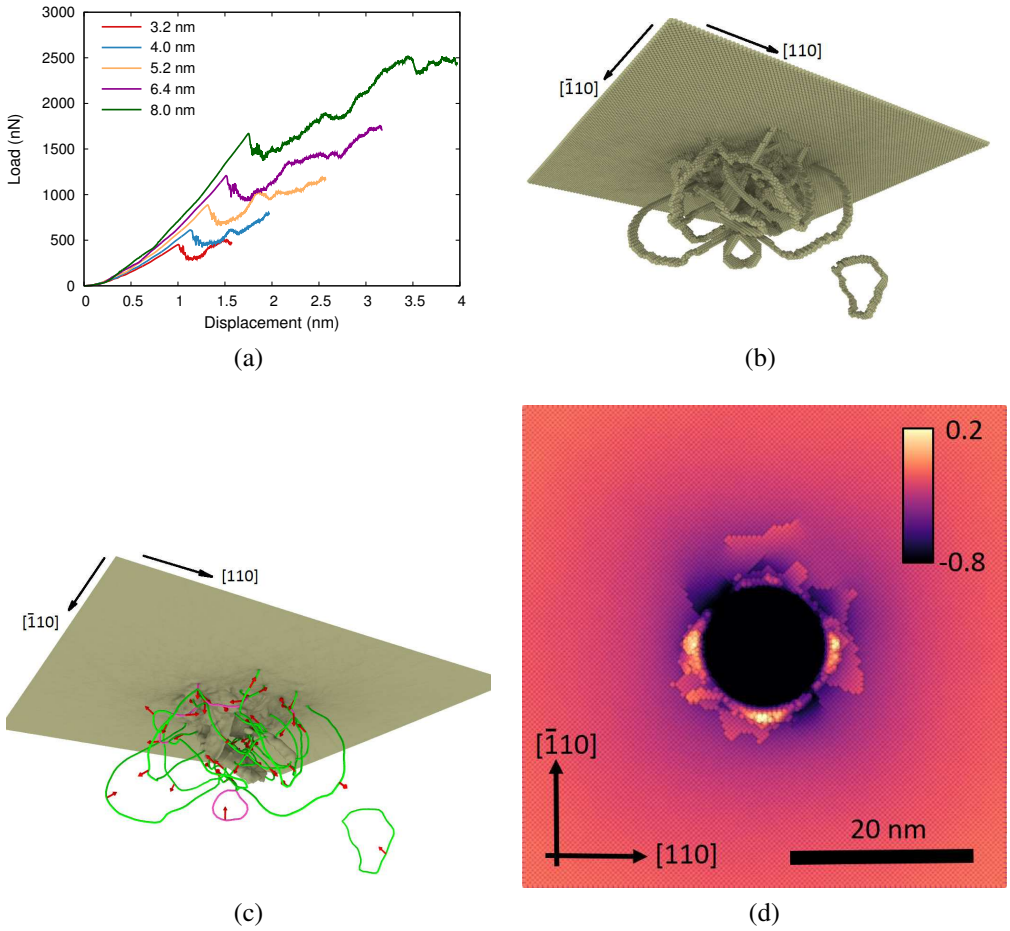

Figure 2: (a) Loading penetration curves. (b) Atomistic view of defects generated under the surface. (c) View of dislocations as identified by the dislocation extraction algorithm. (d) Typical pile-up pattern.

During the analysis of indentation experiments, pile-ups at the indenter site are often explored by Atomic Force Microscopy (AFM) Biener et al. (2007). By means of a height contrast, the pile-up pattern was extracted from our simulations for the 8 nm radius tip, shown in Fig.2d. Biener et al. (2007) observed a 4-fold symmetry and anisotropy of the pile-up pattern when studying $\{100\}$ Ta under a spherical indenter. This 4-fold symmetry is expected for indentation of a $\{100\}$ surface in a cubic crystal, because of the 4-fold symmetry of $\langle 111 \rangle$ slip in this case. The height of the pile-ups is determined by net material transport towards the surface by loop emission and glide on $\langle 111 \rangle$ directions. Crystal plasticity models implemented in Finite Element Method (FEM) calculations can predict pile-ups with the same geometry, but perfectly symmetric, i.e. equivalent in all planes Casals et al. (2007); Casals and Forest (2009). Here, local thermal and stress fluctuations lead to slightly different hillocks on the surface.

In conclusion, in spite of the fact that the simulations presented here have length scales much smaller than that of FEM simulations and experiments, the simulation strategy chosen allows for an adequate representation of the behavior of tantalum under spherical nanometer-sized indentation and opens the possibility to provide further insights into the deformation mechanisms of metals under similar loading conditions.

3 COMPUTATIONAL BENCHMARKS

One of the serious challenges in the simulation of nanoindentation is the limited time scales accessible to simulations because of limited computational resources. The restrictions in computational resources not only affect the time scale, but also the system size that can be explored with the technique. The problem of the dimensional scale can be treated by means of domain decomposition and parallelization, using multiple cores simultaneously, each one solving a subdomain of the original system. The issue of the time scale is more problematic, but computationally speaking it is linearly related to the processor speed. It is clear that parallel high performance computing is a mandatory resource to adequately study these kinds of problems Szlufarska et al. (2008), however, the election of the proper HPC architecture is not unique. In this section, simulation benchmarks are performed and discussed. First, the metrics used for the benchmarks are described. Then a description is given for the hardware infrastructure and software specifications where the simulations were executed. Finally, the discussion of the computational benchmarks are presented.

3.1 BENCHMARKS METRICS

The most common metric to measure performance is time to complete a certain task. Another common metric is *speedup*, which is defined by the time to complete a task in a specific hardware or software configuration divided by the time to complete the same task with other hardware or software configuration. The speedup is commonly used to compare the performance between CPU and GPU execution, it gives a clear understanding of how fast or slow is certain hardware over another. The performance of each hardware architecture can also be inferred by normalizing the wallclock time with the transistor count present in each microprocessor or GPU.

In HPC, two types of benchmarks are usually performed to compare performance: strong scaling and weak scaling. In strong scaling, the size or amount of data to process remains the same, and the amount of parallel processes are increased. With these conditions, wallclock time should decrease, in an ideal or perfect situation, the decrease in time should be linear. The second benchmark is a weak scaling, here, the amount of data or size to process increases accordingly to the increase in processing units. Each CPU core receives the same amount of data to process. In a perfect and ideal weak scaling, the wallclock time should maintain equal as the amount of parallel processes increases. The type of simulation that is presented in this work do not easily allows to perform a weak scaling, for this reason, only a strong scaling is performed. The strong and weak scaling allows obtaining a parallel efficiency that enables to know how a certain hardware or software configuration performs.

3.2 HARDWARE INFRASTRUCTURE AND SOFTWARE SPECIFICATIONS

The hardware and software specifications are described in this section. The simulations were executed in two workstations and three HPC clusters. The two workstations and the Opteron cluster (named “Toko”) belongs to Universidad Nacional de Cuyo, Argentina. The Mendieta cluster resides in Universidad Nacional de Córdoba, Argentina. And finally, access to the “Cab” cluster was granted by the Lawrence Livermore National Laboratory, United States of America. Table 1 shows the basic hardware description of each workstation and clusters. In table 2 it can be seen the software specifications used to compile LAMMPS Plimpton (1995) in each hardware infrastructure.

All simulations were executed using the same stable LAMMPS version, 17 November 2016,

compiled with -O3 optimizations using the GCC compiler and OpenMPI. The GPU simulations were executed using the LAMMPS "GPU" package Brown et al. (2011) compiled with double floating point precision ("-D_DOUBLE_DOUBLE" compile variable). It has been seen that in the GPUs it is possible to execute more than one parallel process of the same simulation and obtain better performance than executing only one process Millán et al. (2012); Brown et al. (2011). For this reason, the simulations in the GPUs were also executed in parallel processes using only one GPU (see section 3.3).

Table 1: Hardware infrastructure.

Name	CPU	RAM	GPUs
FX-8350	AMD FX-8350 8 cores at 4 GHz	32 GB	GeForce GTX Titan X Maxwell (GM200) 12 GB Tesla C2050 Fermi (GF100) 3 GB
Ryzen	AMD Ryzen 1700X 8 cores at 3.4 GHz with 16 threads	16 GB	GeForce Titan X Pascal (GP102) 12 GB
Opteron	Four AMD Opteron 6373 16 cores each, at 2.3 GHz	128 GB	None
Mendieta	Two Intel Xeon E5-2680 v2 10 cores each (20 cores) at 2.8 GHz (HT disabled), 14 nodes (used only 4)	64 GB	Tesla K20x Kepler (GK110) 6 GB
Cab	Two Intel Xeon 8-core E5-2670 8 cores each (16 cores), at 2.6 GHz (HT disabled), 1296 nodes	32 GB per node 41.5 TB total	None

Table 2: Software used to execute the simulations.

Name	Linux distribution	Kernel version	OpenMPI	GCC	CUDA	NVIDIA Driver
FX-8350	Slackware 14.1 64bit	3.10.17	1.10.0	4.8.2	7.0	349.16
Ryzen	Ubuntu 16.04 64bit	4.10.1	1.10.2	5.4	8.0	375.39
Opteron	Slackware 14.1 64bit	3.10.17	1.10.0	4.8.2	NA	NA
Mendieta	CentOS 6.5 64bit	2.6.32-504	2.0.2	4.8.4	8.0	340.29
CAB	TOSS (RedHat) 64bit	2.6.32-696	1.4.3	4.4.7	NA	NA

3.3 BENCHMARKS RESULTS

In this sections several benchmarks are shown for the simulations previously described. The performance of the GPUs accelerators and the different CPU clusters can be seen in Figure 3. This figure shows the simulation number 3 executed with different number of parallel processes (commonly called a strong scaling). The strong scaling parallel efficiency is shown in Figure 4. Figure 5 shows the performance of CPU clusters and GPU accelerators for the five simulations running in eight parallel processes. In Figure 6 the results for the simulation 3 are shown normalized with the transistor count of each processing unit (CPU and GPU). Figure 7 shows the relative performance of all the GPUs compared with some CPU configurations and the

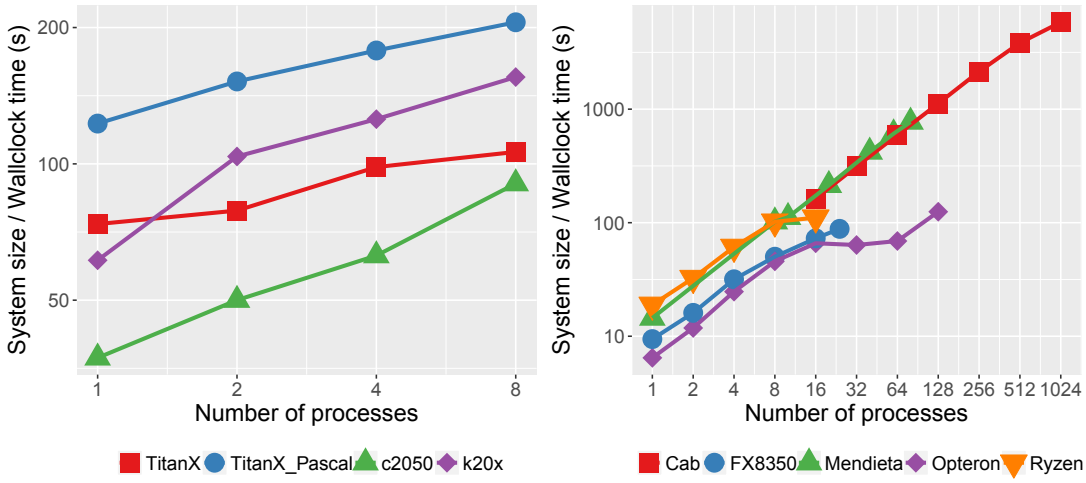

Figure 3: Results for simulation number 3 executed in the GPUs using up to 8 parallel processes in each GPU (left), and the CPUs clusters and workstations in up to 1024 parallel processes (right). Higher is better.

newly available AMD Ryzen CPU (Zen microarchitecture) is compared against CPU clusters and GPUs.

The performance of the GPUs for one of the simulations (simulation number 3) can be seen in figure 3 (left). This simulation was executed in one process in each GPU and in parallel using also one GPU (see Millán et al. (2012); Brown et al. (2011) for more information). Four generations of NVIDIA GPUs were tested: Fermi (c2050), Kepler (k20x), Maxwell (Titan X) and Pascal (Titan X Pascal). The Pascal GPU results in the best performance and the k20x follows. The k20x is one generation older than the Titan X, but the latter is less prepared to execute double precision computations than the k20x that was specially designed to GPGPU. The k20x has more double precision units than the Titan X which is more prepared for gaming rather than HPC (see technical details in NVIDIA (2012) and NVIDIA (2016)). From the Fermi architecture and onward, switching between kernels of different applications running in the same GPU works 20 times faster than in previous generations Glaskowsky (2009), this is one of the reasons why it is advisable (if the application allows it) to execute multiple processes in the same GPU.

The CPU clusters and workstations performance for the same simulation (number 3) can be seen in figure 3 (right) and the parallel efficiency in figure 4. The Cab and Mendieta clusters shows a good scaling up to 1024 cores (Cab) and 80 cores (Mendieta). The Ryzen workstation has a good performance compared with Mendieta, FX8350 and Opteron up to 8 cores, when executing in 16 threads the efficiency decays but still performs better than 24 FX8350 CPU cores and 64 Opteron cores. This drastic decrease in performance and efficiency (see figure 4) in the Ryzen CPU using 16 threads is due to the fact that this CPU only has 8 real CPU cores, each of them capable of executing two threads with an approach called SMT (simultaneous multithreading) similar to Intel HyperThreading. The Opteron cluster shows a good scaling up to 16 cores, then, when using 32 and 64 cores the efficiency decays considerably. The results for the FX8350 CPU shows a good linear scaling up to 24 cores. The strong scaling parallel efficiency can be observed in figure 4, the Cab and Mendieta clusters show a good parallel efficiency, but the Opteron cluster displays a poor parallel efficiency.

The results for all 5 simulations running in 8 parallel processes can be seen in figure 5. The

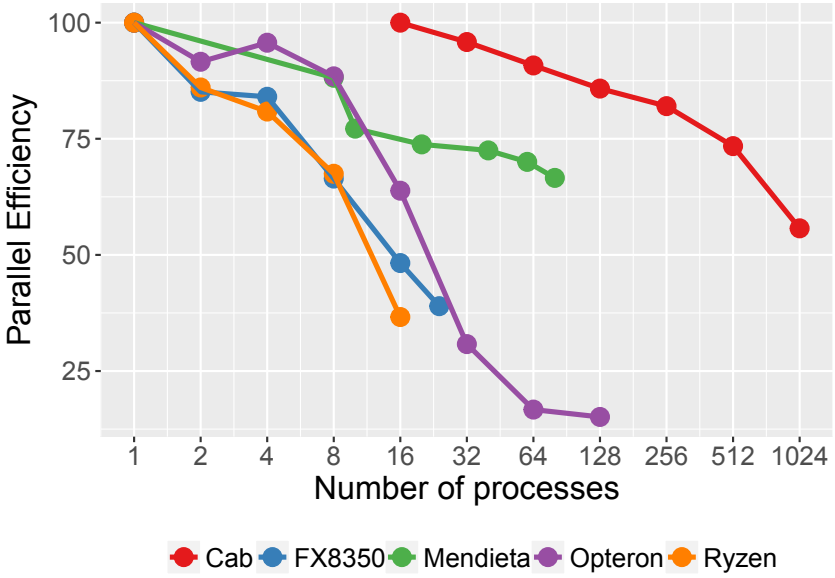

Figure 4: CPU strong scaling parallel efficiency for the same simulations as figure 3 (right)

best performance is obtained with the Titan X Pascal GPU, and the k20x follows. For the bigger cases the Titan X Maxwell and the Ryzen CPU have a similar performance, for the smaller simulations the Titan X performs better. The oldest GPU, the c2050, has a better performance than the FX8350 CPU and the Opteron cluster (all running with 8 parallel processes) and a similar performance than the Ryzen CPU and Mendieta cluster. The decrease in performance as the size of the system increases is due to the increment in communication time and the increase of computations of LAMMPS fixes and modify commands as the number of particles increases.

Figure 6 show performance (measured in system size normalized with wallclock time in seconds) normalized with the transistor count present in each microprocessor or GPU (simulation number 3). The FX8350 and the Opteron processors show a performance per transistor very similar (with 16 cores using two FX-8350 and one Opteron 6376), this is an expected result since both processors share the same microarchitecture, Bulldozer for the Opteron and Piledriver (small improvement over Bulldozer) for the FX8350 (AMD (2013)). The Mendieta and Cab cluster also show an equivalent performance per transistor, both clusters have a similar microprocessor, the Xeon E5-2680 v2 in Mendieta (released in late 2013) and Xeon E5-2670 v1 (released in early 2012) in Cab. The results for the same benchmark performed in the GPUs (figure 6 (left)), shows that the performance per transistor for the k20x is better than the Titan X Maxwell performance, a result already observed in figure 3. The oldest GPU, the c2050, shows the best performance per transistor, although it does not have the best performance as shown previously, this is due to improvements in each CUDA core (shaders) and the increase in core count in newer GPUs.

A commonly used metric to directly compare the performance between CPUs and GPUs is the speedup, the difference between wallclock time of two simulations executed in different hardware. Figure 7 (left) shows the speedups between GPUs and different CPU hardware configurations. As seen in the previous benchmarks, the Titan X Pascal shows the best performance in all the hardware configurations shown in the figure. This GPU has better performance than 16 CPU cores of the Cab cluster, has the double of performance than 24 cores FX8350, $\sim$4

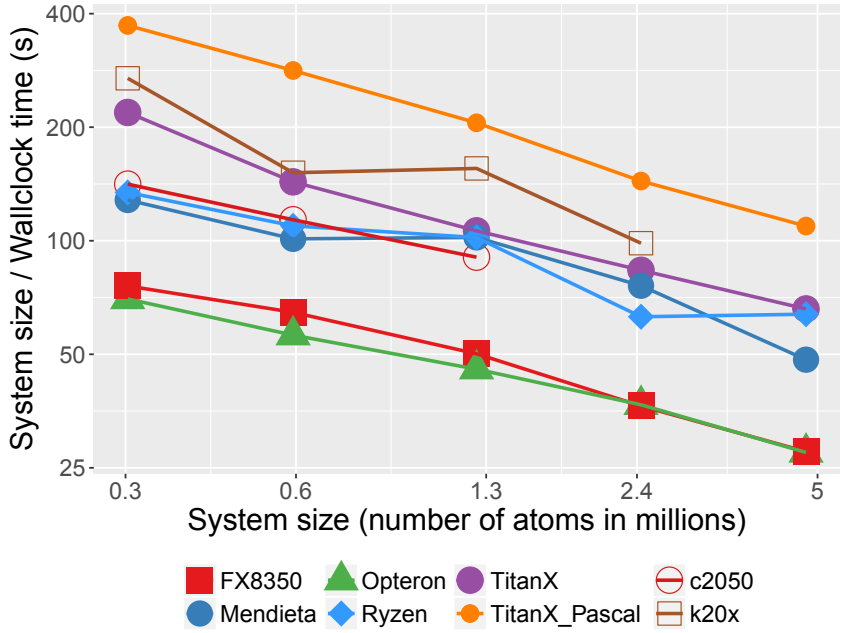

Figure 5: All simulations executed in all the GPUs and all clusters (except Cab) for 8 parallel processes.

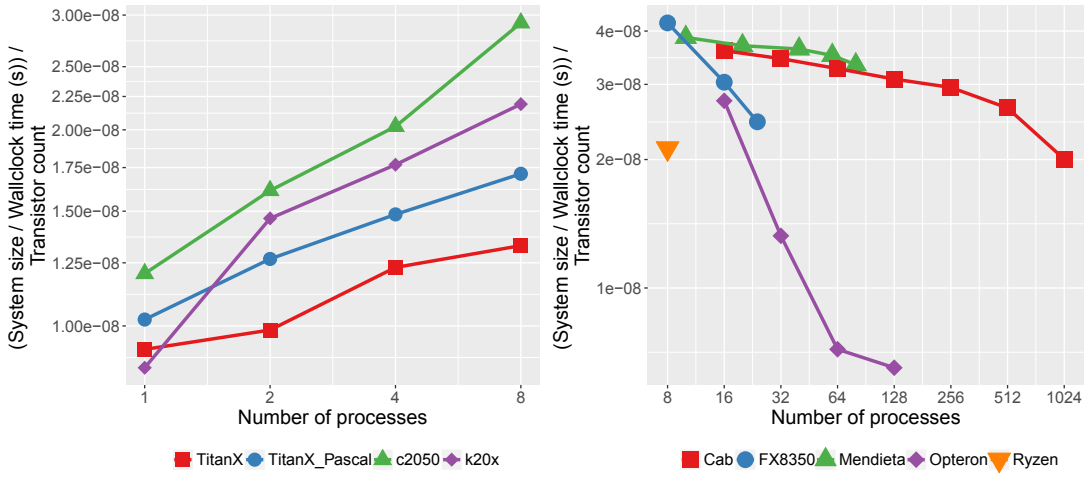

Figure 6: Results for simulation number 3 executed in the GPUs (left) and CPU clusters and workstations (right), normalized with the transistor count present in each microprocessor.

times better than 8 cores of the same CPU, is ∼2.5 faster than 64 Opteron cores, more than 10 times faster than one Ryzen CPU core and ∼2 times faster than 8 Ryzen CPU cores. The Titan X Maxwell has a similar performance than 24 FX8350 cores and 8 Ryzen cores. The k20x GPU has a performance equivalent to 16 Cab cores and is faster in all cases to the Titan X Maxwell, is almost twice as fast as 24 FX8350 cores and ∼1.5 faster than 8 Ryzen cores. The oldest GPU, the c2050, is almost equivalent to 32 Opteron cores, faster than 8 FX8350 cores and cannot achieve the performance of the rest of the CPU configurations. In table 3 and in the figure 8 it can be seen the difference in performance between executing the simulation in one process in each GPU and in 2, 4 and 8 parallel processes in the same GPU. As previously stated, in general, executing more than one process in a GPU results in a much better performance than executing only one. This is, separating the same simulation in equal parts and executing them in the same GPU. In the case of MD simulations, the Domain Decomposition is applied, and each subdomain is executed as a separated process in the same GPU. For this type of MD simulation, changing a GPU from one CPU hardware configuration to another can have an impact on the performance of the simulation. Performing the simulation number 3 and 5 with the TitanX (Maxwell) GPU in the Ryzen CPU improves the performance by ∼15 % against the same simulation executed in the FX-8350 CPU with the same GPU.

Table 3: Speedups of each GPU comparing wallclock time for one processes between 2, 4 and 8 parallel processes in the same GPU

Parallel processes	c2050	k20x	TitanX	TitanX_Pascal
1	1	1	1	1
2	1.26	1.55	1.09	1.23
4	1.53	1.77	1.41	1.53
8	2.21	2.64	1.83	1.71

Also in Figure 7 (right) it can be seen the performance comparison of the 16 threads of the Ryzen CPU against all GPUs and CPUs clusters up to 64 parallel processes. Values beneath 1 indicates that the Ryzen CPU cannot match the performance with that hardware configuration. Values above 1 indicates that the Ryzen CPU in 16 threads perform better than the selected configuration. The interesting result to highlight is that the 16 Ryzen threads are ∼1.4 faster than 24 FX8350 cores, ∼1.7 faster than 32 Opteron cores and ∼1.6 faster than 64 Opteron cores. The improvement in performance with the new AMD Zen microarchitecture over the older Bulldozer microarchitecture is significant. Comparing against Xeon CPUs (without HyperThreading), the 16 Ryzen threads run ∼1.3 faster than 8 Xeon E5-2680 v2 cores and slower (∼0.6 x) than 16 Xeon E5-2670 v1 cores (Cab cluster). The Ryzen 16 threads match the performance of the Titan X Maxwell GPU (in 8 parallel processes) and has almost half the performance of the Titan X Pascal GPU (also in 8 parallel processes).

A common metric to compare the performance of MD simulations is the wallclock time divided by the total number of atoms in the system, divided by the number of steps that the simulation was executed and finally divided by the number of parallel processes the simulation was executed on. A typical value for an EAM simulation obtained by the LAMMPS community is ∼1.85 μs for a Intel Xeon 3.47 GHz processor using one CPU core (see LAMMPS benchmark page at LAMMPS (2017)). Table 4 shows the performance obtained in this work for simulation number 3 using one CPU core in all hardware configurations except the Cab cluster.

The memory consumption in some cases can determine if the simulation can be executed

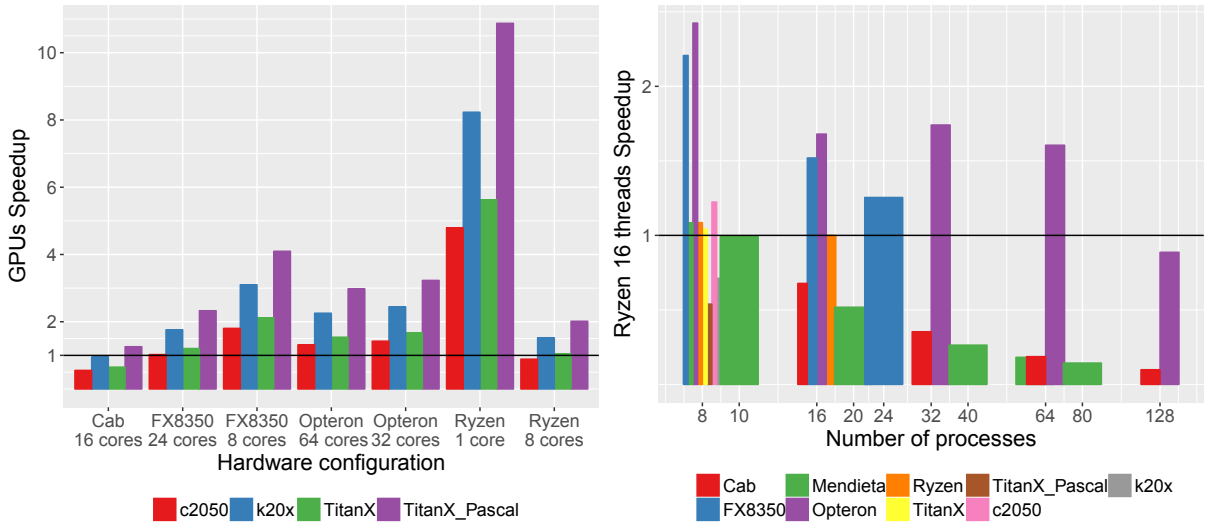

Figure 7: Performance comparison for simulation number 3 between each of the GPUs and a selected number of CPU configurations (left). In the plot on the right, speedup between the Ryzen CPU using 16 threads and all CPU and GPU hardware configurations (from 8 parallel processes up to 128). Speedup represents wallclock time from CPU divided by GPU time for the same simulation.

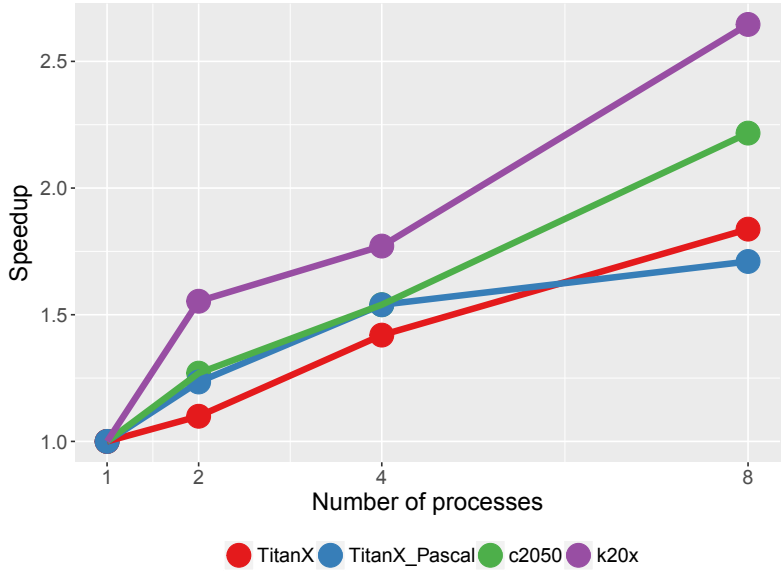

Figure 8: GPU speedups from parallel processes against one GPU process in each of the four tested GPUs. See data in table 3. This figure shows how much improvement in performance was obtained in each GPU when executing more than one parallel process in each of them.

Table 4: Performance measured in *wallclock time / natoms / nsteps in* μs running in one CPU core for simulation number 3.

Hardware	Opteron	FX-8350	Mendieta	Ryzen	TitanX	TitanX_Pascal	k20x	c2050
Performance in μs	1.19	0.81	0.53	0.4	0.1	0.06	0.12	0.2

in a GPU or not. For the simulations performed in this work, simulations 1 and 2 cannot be executed in the c2050 GPU due to memory constraints, a problem that was not present in the CPU clusters and workstation. Simulation number 3 uses up to ∼1800 MB of GPU memory (reported by the *nvidia-smi* utility) and the same simulation running in the Ryzen CPU uses ∼920 MB of RAM (reported by LAMMPS).

From the performed benchmarks one could highlight a number of results. The use of GPUs to perform MD simulations is an attractive alternative to CPU only supercomputers or clusters, due to the cost of implementation, the obtained performance and the availability of software. The k20x GPU (released in 2012) has a greater performance than 8 Ryzen CPU cores (2017), and matches the performance of 16 Xeon CPU cores from the Cab cluster (2012). The most recent Titan X Pascal doubles the performance of 8 Ryzen CPU cores and performs better than 64 Opteron 6376 cores (∼3 x of speedup). It can be interesting to evaluate the cost to acquire a GPU workstation and one node of an Opteron cluster. For example, the Titan X Pascal can cost up to US$1600 (July 2017) and a workstation in Argentina for that GPU costs approx. US$1200. A 64 cores Opteron 6376 node like the one used in this work costs up to US$9000 (with 64 GB of RAM and in the year 2016). If the simulation can be executed in the GPU (the software is GPU-ready and the 12 GB of GPU memory are enough), it can be much cheaper to acquire a GPU workstation than invest in a Cluster Node.

The newly released AMD Ryzen CPU shows a good performance compared with the older Xeon and AMD Opteron processors. Running in 16 threads in the Ryzen CPU results in better performance than in three AMD FX-8350 CPUs (24 cores), comparing with 32 Opteron cores gives a speedup of ∼1.7 x. The Intel Xeon CPUs present in the Cab and Mendieta clusters show perfect scaling up to 1024 cores (in Cab) and 80 cores (in Mendieta). The AMD Opteron 6376 shows a poor scaling in almost every simulation performed here, except for the smaller simulation number 5, the scaling problems of this microprocessor are discussed in detail in Millán et al. (2015) for an HPC Cellular Automata implementation.

4 CONCLUSIONS AND FUTURE WORK

One of the serious challenges that atomistic simulations face has to do with the limitation on the scales which can be accessed, which in turn is dictated by limitations on computational resources. This problem can often be overcome by means of high performance computing, provided adequate equipment is chosen. In this work, several CPU and GPU architectures were tested using local resources and an international supercomputer.

The multiple benchmarks performed in this work show that using a workstation with a powerful GPU can match or exceed the performance of a CPU cluster. The Titan X Pascal match the performance of 16 Xeon cores from the Cab cluster, and it has a ∼3 x speedup against 64 Opteron CPU cores. The newly released AMD Ryzen 1700X CPU performs well compared with the GPUs and against other CPU processors. The 16 threads of the Ryzen CPU outperforms 24 FX-8350 CPU cores and match the performance of 10 Xeon E5-2680 v2 CPU cores from the Mendieta cluster. The Cab and Mendieta clusters with Xeon processors show a good

scaling up to 1024 cores (Cab) and 80 cores (Mendieta). The Opteron cluster shows a poor scaling and performance compared with other CPU configurations.

In the future, it will be of interest to test Xeon Phi processors and compare the performance of theses devices with the GPUs presented in this work, as in recent work Parks et al. (2017); Pennycook et al. (2013). The new AMD processors for Servers, named EPYC, based on the Zen microarchitecture present an attractive platform to test, since the Ryzen CPU performed well in these benchmarks. This new processor has 32 cores (64 threads) per socket (with the possibility of using two sockets per cluster node) and 8 memory channels.

5 ACKNOWLEDGEMENTS

The support of ANPCyT PICT-2014-0696 and PICT-2015-0342 and SeCTyP-UNCuyo grants is gratefully acknowledged. The Titan X Pascal used for this research was donated by the NVIDIA Corporation. This work used Mendieta Cluster from CCAD-UNC, which is part of SNCAD-MinCyT, Argentina, and the Toko cluster from FCEN-UNCuyo, Mendoza, Argentina. Finally, we thank Livermore Computing for supercomputer resources (CAB) used for this work.

REFERENCES

Allen M.P. and Tildesley D.J. *Computer simulation of liquids*. Oxford university press, 1989.

AMD A.M.D. Bios and kernel developers guide (bkdg) for amd family 15h models 00h-0fh processors. 42301(3.14), 2013.

Anderson J.A., Lorenz C.D., and Travesset A. General purpose molecular dynamics simulations fully implemented on graphics processing units. *Journal of Computational Physics*, 227(10):5342–5359, 2008. ISSN 0021-9991. doi:10.1016/j.jcp.2008.01.047.

Armstrong R., Elban W., and Walley S. Elastic, plastic, cracking aspects of the hardness of materials. *International Journal of Modern Physics B*, 27(08), 2013.

Biener M., Biener J., Hodge A., and Hamza A. Dislocation nucleation in bcc ta single crystals studied by nanoindentation. *Physical Review B*, 76(16), 2007. ISSN 1098-0121, 1550-235X.

Bringa E., Rosolankova K., Rudd R., Remington B., Wark J., Duchaineau M., Kalantar D., Hawreliak J., and Belak J. Shock deformation of face-centred-cubic metals on subnanosecond timescales. *Nature materials*, 5(10):805–809, 2006.

Bringa E.M., Caro A., Wang Y., Victoria M., McNaney J.M., Remington B.A., Smith R.F., Torralva B.R., and Van Swygenhoven H. Ultrahigh strength in nanocrystalline materials under shock loading. *Science*, 309(5742):1838–1841, 2005.

Brown W.M., Wang P., Plimpton S.J., and Tharrington A.N. Implementing molecular dynamics on hybrid high performance computers - short range forces. *Computer Physics Communications*, 182(4):898–911, 2011. ISSN 0010-4655. doi:10.1016/j.cpc.2010.12.021.

Casals O. and Forest S. Finite element crystal plasticity analysis of spherical indentation in bulk single crystals and coatings. *Computational Materials Science*, 45(3):774–782, 2009.

Casals O., Očenášek J., and Alcala J. Crystal plasticity finite element simulations of pyramidal indentation in copper single crystals. *Acta materialia*, 55(1):55–68, 2007.

Cybenko G. Parallel computing for machine learning in social network analysis. In *2017 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. IEEE, 2017. doi:10.1109/ipdpsw.2017.178.

Dai X.D., Kong Y., Li J.H., and Liu B.X. Extended Finnis-Sinclair potential for bcc and fcc metals and alloys. *Journal of Physics: Condensed Matter*, 18(19):4527–4542, 2006.

Fischer-Cripps A.C. *Nanoindentation*. Mechanical engineering series. Springer, New York, 2nd

ed edition, 2004. ISBN 0387220453.

Glaskowsky P.N. NVIDIA's Fermi: the first complete GPU computing architecture. *NVIDIA*, 2009.

Gruber L. and West M. GPU-accelerated bayesian learning and forecasting in simultaneous graphical dynamic linear models. *Bayesian Analysis*, 11(1):125–149, 2016. doi:10.1214/15-ba946.

Hertz H. Ueber die berührung fester elastischer körper. *Journal für die reine und angewandte Mathematik (Crelle's Journal)*, 1882(92), 1882.

Huang S.Y., Spurzem R., and Berczik P. Performance analysis of parallel gravitationalN-body codes on large GPU clusters. *Research in Astronomy and Astrophysics*, 16(1):011, 2016. doi:10.1088/1674-4527/16/1/011.

Kelchner C., Plimpton S., and Hamilton J. Dislocation nucleation and defect structure during surface indentation. *Physical Review B*, 58(17):11085–11088, 1998.

Kohnke B., Ullmann R.T., Kutzner C., Beckmann A., Haensel D., Kabadshow I., Dachsel H., Hess B., and Grubmüller H. A flexible, GPU - powered fast multipole method for realistic biomolecular simulations in gromacs. *Biophysical Journal*, 112(3):448a, 2017. doi:10.1016/j.bpj.2016.11.2402.

LAMMPS. Lennard Jones liquid benchmark. http://lammps.sandia.gov/bench.html#lj, 2017.

Meyers M.A. and Chawla K.K. *Mechanical behavior of materials*. Cambridge University Press, Cambridge; New York, 2009.

Millán E., Garcıa Garino C., and Bringa E. Parallel execution of a parameter sweep for molecular dynamics simulations in a hybrid gpu/cpu environment. In *XVIII Congreso Argentino de Ciencias de la Computación 2012 (CACIC)*. Bahia Blanca, Buenos Aires, 2012.

Millán E.N., Bederian C., Piccoli M.F., García Garino C., and Bringa E.M. Performance analysis of cellular automata HPC implementations. *Computers & Electrical Engineering*, 48:12–24, 2015. ISSN 0045-7906. doi:10.1016/j.compeleceng.2015.09.015.

Millán E.N., Goirán S.B., Piccoli M.F., García Garino C., Aranibar J.N., and Bringa E.M. Monte Carlo simulations of settlement dynamics in GPUs. *Cluster Computing* , 2015. ISSN 1573-7543. doi:10.1007/s10586-015-0501-5.

NVIDIA. Whitepaper NVIDIA Next Generation CUDA Compute Architecture: Kepler GK110. v1.0, 2012.

NVIDIA. Nvidia tesla p100. 2016.

Parks C., Huang L., Wang Y., and Ramkrishna D. Accelerating multiple replica molecular dynamics simulations using the intel xeon phi coprocessor. *Molecular Simulation*, 43(9):714–723, 2017. doi:10.1080/08927022.2017.1301666.

Pennycook S.J., Hughes C.J., Smelyanskiy M., and Jarvis S. Exploring SIMD for molecular dynamics, using intel xeon processors and intel xeon phi coprocessors. In *2013 IEEE 27th International Symposium on Parallel and Distributed Processing*. IEEE, 2013. doi:10.1109/ipdps.2013.44.

Plimpton S. Fast parallel algorithms for short-range molecular dynamics. *Journal of computational physics*, 117(1):1–19, 1995.

Rajulapatit K., Biener M., Biener J., and Hodge A. Temperature dependence of the plastic flow behavior of tantalum. *Philosophical Magazine Letters*, 90(1):35–42, 2010.

Remington T., Ruestes C., Bringa E., Remington B., Lu C., Kad B., and Meyers M. Plastic deformation in nanoindentation of tantalum: A new mechanism for prismatic loop formation. *Acta Materialia*, 78:378–393, 2014.

Ruestes C., Stukowski A., Tang Y., Tramontina D., Erhart P., Remington B., Urbassek H.,

Meyers M., and Bringa E. Atomistic simulation of tantalum nanoindentation: Effects of indenter diameter, penetration velocity, and interatomic potentials on defect mechanisms and evolution. *Materials Science and Engineering: A*, 613:390–403, 2014.

Stukowski A. Visualization and analysis of atomistic simulation data with OVITO-the open visualization tool. *Modelling and Simulation in Materials Science and Engineering*, 18(1):015012, 2010.

Stukowski A. and Albe K. Extracting dislocations and non-dislocation crystal defects from atomistic simulation data. *Modelling and Simulation in Materials Science and Engineering*, 18(8):085001, 2010.

Szlufarska I., Chandross M., and Carpick R.W. Recent advances in single-asperity nanotribology. *Journal of Physics D: Applied Physics*, 41(12):123001, 2008.

TOP500.org. Top 500 supercomputers, list of june 2017. 2017.

Tsuzuki H., Branicio P.S., and Rino J.P. Structural characterization of deformed crystals by analysis of common atomic neighborhood. *Computer physics communications*, 177(6):518–523, 2007.

Ziegenhain G., Urbassek H.M., and Hartmaier A. Influence of crystal anisotropy on elastic deformation and onset of plasticity in nanoindentation: A simulational study. *Journal of Applied Physics*, 107(6):061807, 2010. ISSN 00218979.