International Journal of Green Computing
Volume 9 • Issue 1 • January-June 2018

Ecological Impact of Green Computing Using Graphical Processing Units in Molecular Dynamics Simulations

Izabele Marquetti, North Carolina A&T State University, USA
Jhonatam Rodrigues, North Carolina A&T State University, USA
Salil S. Desai, North Carolina A&T State University, USA

ABSTRACT

Molecular dynamics (MD) models require comprehensive computational power to simulate nanoscale phenomena. Traditionally, central processing unit (CPU) clusters have been the standard method of performing these numerically intensive computations. This article investigates the use of graphical processing units (GPUs) to implement large-scale MD models for exploring nanofluidic-substrate interactions. MD models of water nanodroplets over flat silicon substrate are tracked wherein the simulation attains a steady state computational performance. Different classes of GPU units from NVIDIA (C2050, K20, and K40) are evaluated for energy efficiency performance with respect to three green computing measures: simulation completion time, power consumption, and CO_2 emissions. The CPU+K40 configuration displayed the lowest energy consumption profile for all the measures. This research demonstrates the use of energy efficient graphical computing versus traditional CPU computing for high-performance molecular dynamics simulations.

KEYWORDS

Graphical Processing Units, Green Computing, High Performance Computation, Hybrid 3D Printing, Industry 4.0, Molecular Dynamics, Nanomanufacturing

INTRODUCTION

Green Computing

Computing technologies are usually associated with high energy consumption. Thus, the energy efficiency of hardware and operating systems has become a financial and environmental concern (Zhu, Sun, & Hu, 2012). In the United States, desktop computers can represent around 10% of the consumption of the commercial electricity. Also, high-performance computers require a powerful cooling system for heat dissipation, thereby exacerbating the energy consumption. Consequently, computers can produce carbon dioxide equivalent to millions of cars (Li, 2012). The choice of software also influences the energy consumption profile and needs to be evaluated for power optimization purposes. Therefore, green practices are necessary to be incorporated into the design and operation

DOI: 10.4018/IJGC.2018010103

of systems. Green computing is a recent approach which allows computer systems to be energy efficient by implementing newer technologies to reduce power consumption without compromising their performance (Binder & Suri, 2009; Singh, 2015; Zhou, Xiao, Yang, & Gong, 2016; Zhu et al., 2012). In addition, to improve the efficiency and economy of computer systems, green computing also considers the reduction of computer systems impact on the environment and people. These include the achievement of energy savings and environmental protection, which can impact the life cycle of computer systems (Zhang, Gong, & Li, 2011). Harmon and Auseklis (2009) define green computing as improving the efficiency of computing resources while minimizing the environmental impact, considering the entire life cycle of a product. This can be achieved by reducing the consumption of water, energy, limited resources and avoiding the use of hazardous materials, thereby minimizing all waste in the entire supply chain.

The practice of green computing involves the design, manufacture, use and proper disposal of computers and all subsystems, such as printers, monitors, and storage devices, in an efficient and effective way that causes minimal impact to the environment (Velte, Velte, & Elsenpeter, 2008). These can be achieved in our daily routine with simple actions, such as switching off the computer when not being used, choosing computer monitors that are more energy efficient (LED backlighting), keeping the computer well-maintained and optimized, disabling unnecessary programs that start automatically, recycling computer parts, batteries and printer cartridge (Appasami & Suresh Joseph, 2011; Chakraborty, Bhattacharyya, Nargiza, & Bedajna, 2009).

Green computing can be considered in terms of both software and hardware. In software, its objective is to obtain methods that increase the efficiency of programs, decreasing storage space and saving energy. Some examples include Cloud Computing, Distributed Computing, and High-Performance Computing. For hardware, green computing includes certain technologies that enable a significant reduction in the energy consumption and the emissions footprint, while increasing economic efficiency and facilitating recycling (Zhang et al., 2011).

Graphical Processing Units

Graphics processing units (GPUs) have emerged in the past few years as an alternative to Central Processor Units (CPUs) due to their powerful capacity for applications that demand high computational resources. GPUs have advanced from a fixed-function processor with great three-dimensional (3D) graphics to a powerful programmable processor with application programming interface (API), resulting in processors with high arithmetic capacity. These processor units are now designed for large and parallel computation requirements with a focus on throughput instead of latency, having different applications than a CPU architecture (Owens et al., 2008). Owens et al. (2008) emphasize the importance of GPUs for scientific computing applications to solve large and complex computational problems, including protein simulations, scalable molecular dynamics simulations, and calculations of electrostatic potential maps.

In high-performance computing, energy consumption (Jan et al., 2012) is an important factor because significant computing power is required to solve intensive calculations and execute simulations. In addition to that, it is possible to reduce the amount of time to perform an operation and increase the productivity (Vijayaraghavan, Garg, Vijayaraghavan, & Gao, 2015). Studies in the green computing field show that a large amount of energy can be saved by focusing on environmental protection and conservation of energy (NRC, 2010). Zhang et al. (2011) used three computing systems to evaluate parameters such as green-related indexes to improve the green computing systems. Singh, Naik, and Mahinthan (2015) demonstrated that the choice of energy-efficient API with optimal parameters have a significant impact on energy savings for different servers. As an example, it was possible to save up to 76% energy during a file reading routine. Cecilia (2013) presented GPU implementations of two different nature-inspired optimization methods to validate hardware enhancements using Nvidia's Fermi architecture. Picariello, Rapuano, and Villano (2013) have developed a measurement system and test-bench architecture to measure the instantaneous power consumption from the power line of a workstation motherboard. Marquetti and Desai (2016) investigated the influence of GPU units on the performance of different biomolecular molecular dynamics (MD) models benchmarking different GPUs with CPU configurations.

Investigating Nanoscale Processes Through Molecular Simulations

Several direct-write processes such as electrospraying and inkjet methods utilize droplets ranging from nano to microscale dimensions (Jaworek, 2007; Jaworek & Sobczyk, 2008; Kullmann et al., 2012). Atomized droplets of liquids at the nanoscale are deposited on different substrate materials to directly write three-dimensional features (Hon, Li, & Hutchings, 2008). In recent years, nanoscale droplets are being employed to build complex geometries of molecules for both biomedical, sensor and electronic applications (Buck, Xu, Brasuel, Philbert, & Kopelman, 2004; Della Rocca, Liu, & Lin, 2011; Ko et al., 2007, Marquetti & Desai, 2018a, 2018b). An important aspect of this deposition process is studying the interfacial properties between nanoscale liquids with a variety of substrate materials (Desai & Kaware, 2012; Yang, 2006). Continuum models such as computational fluid dynamics have yielded good results for microscale droplets. However, they fail to give a detailed understanding of the complex fluid-structure phenomenon at nanoscale regimes (Curtin & Miller, 2003; Schatz, 2007). Molecular dynamics modeling is a potent strategy to explore atomic-level behavior of liquids under different ambient conditions (Ritos, Dongari, Borg, Zhang, & Reese, 2013).

Computational Complexity of Molecular Models

Molecular dynamics models require enormous computational power to simulate thousands of atoms to mimic realistic phenomenon. In addition, the time steps needed for these simulations are extremely small ranging up to 1 fs ($10e^{-15}$ s) which result in longer run times. These large-scale computations demand parallel computing architectures where each time step can be completed in few milliseconds. Molecular dynamic simulations involve computing both non-bonded and bonded potential functions for pairwise interactions among atoms. The non-bonded interactions consume the higher fraction of computation power over the bonded energy calculations. Non-bonded interactions are geometry dependent which complicates the parallelization approaches for molecular dynamics. Cut-off distances are established to alleviate this problem wherein, atoms interacting within a certain cut-off distance are only considered for calculations. However, specialized MD codes such as NAMD have employed a hybrid spatial and force decomposition strategy (Nelson et al., 1996). This permits it scalability on multiple CPUs on a computing cluster (Kalé et al., 1999). In recent years, GPUs have been implemented to execute parallel architecture code including NAMD. GPUs consist of a large number of cores which are capable of parallel processing with minimal communication overhead. Thus, non-bonded computations are executed on GPUs while the CPU handles the bonded interactions. Accelerated run times as high as 12 times have been observed by running the NAMD source code on GPUs over conventional CPU systems (Phillips, Stone, & Schulten, 2008). Despite their superior computational performance, GPUs need to be evaluated for their power usage metrics over CPUs to justify their widespread implementation.

In this paper, our research focuses on investigating the green computing measures while conducting large-scale molecular dynamics models on GPUs. Our team is interested in studying the infiltration of nanodroplets within substrate matrices towards developing novel additive manufacturing (3D Printing) processes (Adarkwa & Desai, 2016; Parupelli & Desai, 2017). In this context, we explore molecular dynamics simulations as a vital tool to predict the wetting properties of different substrate materials (Cordeiro & Desai, 2016; Desai, Kaware, & Rodrigues, 2014). Large-scale molecular dynamics (MD) models are notorious for consuming substantial computing resources. High-performance parallel computing using GPUs has emerged as an alternative to traditional CPU based cluster computing (Garland, 2008). Our team investigates the energy consumption and carbon emission footprint for different computational devices which include CPU and a variety of GPUs towards green computing.

METHODOLOGY

The nanodroplet and substrate interaction was simulated using molecular dynamics modeling. Nanoscale Molecular Dynamics (NAMD) open source code was used to simulate the molecular models (Phillips et al., 2005). Virtual Molecular Dynamics (VMD) platform was used as a pre-processor for model building and visualizing the results. Both systems were chosen due to their robust performance for parallel computing using GPUs (Stone et al., 2010) and their compatibility with different force fields for the frequently used CHARMM (Brooks et al., 1983) and AMBER (Pearlman et al., 1995) packages. In this work, water nanodroplets with diameters of 4 and 10 nm were modeled with the TIP3P structure using CHARMM force field system, with 3,384 and 17,267 molecules, respectively. Figure 1 illustrates the MD model of a 10 nm water droplet placed on top of a flat silicon substrate.

MD simulations consider that each atom in the system experiences a force described by a force field model which includes all interactions of an atom with the system (Phillips et al., 2005). The atomic interactions follow the potential energy function presented in Equation 1 (Andrew, 1996). The force fields used in the simulation are compatible with the CHARMM standard:

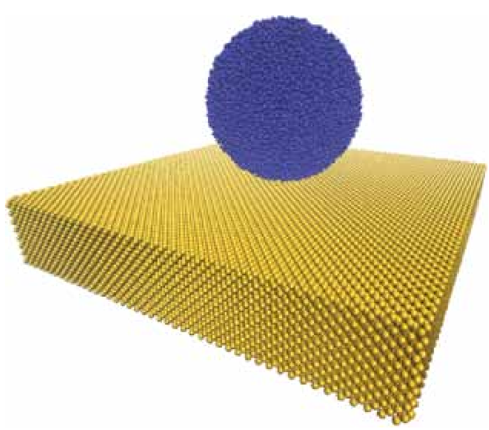
$$U_{total} = U_{bond} + U_{angle} + U_{dihedral} + U_{nonbond} \tag{1}$$

where, U_{bond}, U_{angle}, and $U_{dihedral}$ are the stretching, bending and torsion bonding interactions, respectively. $U_{nonbond}$ are the interactions between non-bonded pairs of atoms and correspond to the Van der Waals forces, expressed by a Lennard-Jones 6-12 potentials, and the electrostatic interactions between charged particles (Phillips et al., 2005).

U_{bond} considers all covalent bonds in a system (Equation 2), U_{angle} are the interaction energy between angles of each covalent bonds that share a single atom in the vertex (Equation 3), $U_{dihedral}$. counts for atom pairs separated by three covalent bonds that have a torsion angle $\varnothing$ in the central bond (Equation 4):

$$U_{bond} = \sum_{bonds\, i} k_i^{bond} \left(r_i - r_{0i}\right)^2 \tag{2}$$

Figure 1. MD model configuration of 10 nm water droplet on a flat silicon substrate



$$U_{angle} = \sum_{angles\ i} k_i^{angle} \left(\theta_i - \theta_{0i}\right)^2 \tag{3}$$

$$U_{dihedral} = \sum_{dihedral\, i} \begin{cases} k_i^{dihedral} \left[1 + \cos\left(n_i \varnothing_i \text{-} \gamma_i\right)\right], & n_i \neq 0 \\ k_i^{dihedral} \left(0_i \text{-} \gamma_i\right)^2, & n = 0 \end{cases} \tag{4}$$

where r represents bonds stretching, θ, bond angle, and $\varnothing$, dihedral angle.

Table 1 presents the bonded force field parameters for the silicon substrate used in the simulations obtained from (Lippert, 1960).

The electrostatic interaction occurs due to the different distribution of charge in a molecule, and it is described by a Coulomb potential (Equation 5). Van der Waals interactions are intermolecular, where there is a weak attractive or repulsive force between molecules caused by non-covalent interactions (Equation 6) (Phillips et al., 2005):

$$U_{Coulomb} = \sum_i \sum_{j>1} \frac{q_i q_j}{4\pi\varepsilon_0 r_{ij}} \tag{5}$$

where, q is the electrostatic charges of atoms i and j, r_{ij} distance between atom i and j, and μ_0 is the dielectric constant:

$$U_{VdW} = \sum_i \sum_{j>1} 4\varepsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right] \tag{6}$$

where, ε_{ij} is a measurement of attraction between atoms/molecules i and j, and σ_{ij} is the distance in which the intermolecular potential between i and j is zero.

The nonbonded force field parameters for the silicon substrate used in the simulations are presented in Table 2. The parameters for water molecules were obtained from the CHARMM force field, while the ones for silicon atoms were obtained from (Mayo, Olafson, & Goddard, 1990).

The nanodroplets were placed on silicon substrates with dimensions of 243 × 243 × 42 Å and were composed of 129,600 atoms of silicon. During the simulation, all the substrate atoms had their atomic coordinates fixed to their initial positions, while water molecules were set to move freely.

Table 1. Bonded interaction parameters

Atom Type	k_i (kcal/mol)	r (Å)
Si Si	53.06	2.33

Table 2. Nonbonded interaction parameters

Atom Type	ε_{ij} (kcal/mol)	σ_{ij} (Å)
Si Si	-0.31	4.27

Appropriate force fields, geometry configuration, and simulation parameters were specified to initiate the MD models. Key parameters include replication cell configurations, temperature and pressure controls (Andrew, 1996).

The MD simulations were run on a 64-bit Linux platform (Fedora 21) with GPU computing units as stated in Table 3. In this research, we used graphical processing units from NVIDIA Corporation. The computing devices were housed in a workstation architecture with active cooling. A power meter (Watts up - Pro ES) was used to measure the power consumption when each simulation was run using a combination of different devices. The power meter was placed in between the wall power outlet and workstation to record different power entities such as instantaneous power and energy consumption concerning time. Data was recorded for every 10 second time interval for all the simulation runs. The performance of the CPU and each CPU+GPU configuration was evaluated according to three green computing measures: (a) simulation completion time (h), (b) power consumption (Watts), and (c) CO_2 emissions (kg). These measurements were selected to quantify the actual reduction in both power consumption of a system and impact on the environment that different GPU devices can achieve without compromising the performance of high demand computational problems.

It is important to note that in a CPU+GPU configuration, the GPUs compute the non-bonded force evaluation while the CPUs compute the energy evaluations for molecular dynamics simulations. Thus, a CPU device is mandatory to be used along with the GPU. The comparative analysis of different device configurations was conducted in two steps. In the first step, the CPU only configuration was compared to the CPU+K40-GPU configuration. The K40-GPU is the most advanced GPU among the ones being tested. Thus, we benchmarked the CPU against the K40-GPU to evaluate the largest differences among these device configurations. In the second step, the three GPUs (C2050, K20, and K40) were compared to evaluate differences among their green computing performance. One of the prerequisites for implementing this methodology is the availability of appropriate GPUs hardware for a particular molecular dynamics source code. In addition, the chosen MD source code should be configured for parallel computing architecture. This includes installation of Compute Unified Device Architecture (CUDA) libraries that are compatible with both hardware (GPUs) and software (MD source code) configurations.

For the first part of this research, models were subjected to 1,000 minimization steps, and the simulations were performed for 0.4 ns, resulting in a total of 200,000 equilibration steps. The second part of this research consisted of 0.2 ns simulations, with 1,000 minimization steps and 100,000 equilibration steps. Data was recorded every 20,000 time steps for all simulations. The simulations were executed through equilibration. MATLAB was used to code an image processing algorithm to measure the dynamic contact angles during the spread of the droplets.

RESULTS AND DISCUSSION

Figure 2 presents the simulations progressing over time for droplets spreading on the silicon substrate for 4 and 10 nm. Figure 3 shows the variation in dynamic contact angle for 4 and 10 nm droplets, respectively. As can be seen in Figure 3, the 4 nm droplet shows wide fluctuations in contact angle

Table 3. CPU and GPU details used to run molecular dynamics models

Device	Cores	Memory	Clock
K40-GPU	2880	12 GB	475 MHz
K20-GPU	2496	5 GB	706 MHz
C2050-GPU	448	3 GB	1150 MHz
CPU	16	15.6 GB	3500 MHz

Figure 2. Droplet spreading progression for 4 and 10 nm droplets on a flat silicon substrate over time

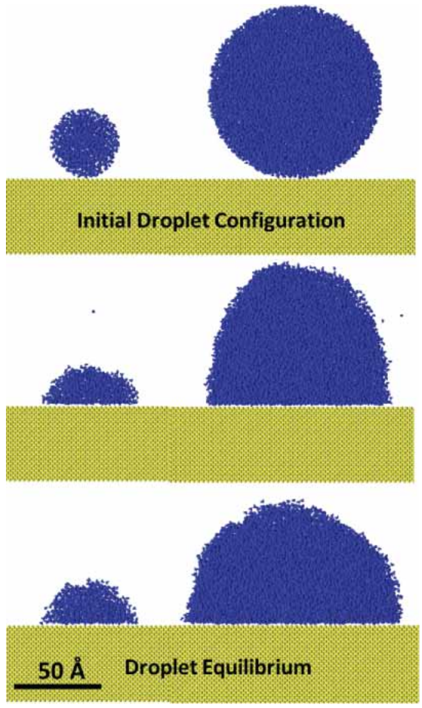
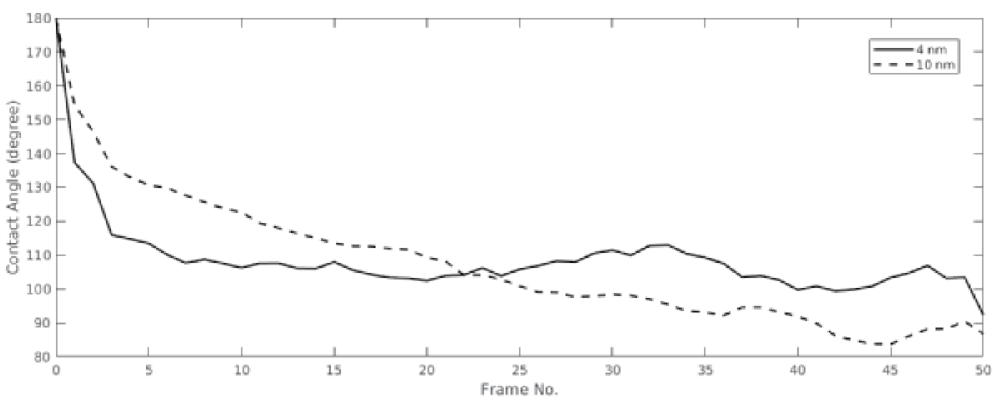


Figure 3. Dynamic contact angle for 4 and 10 nm droplets



measurements as compared to the 10 nm droplet. This can be attributed to the presence of higher number of surface atoms for 4 nm droplet which result in rapid evaporation of the water molecules. Thus, the 4 nm droplet is continuously adapting to a change in shape which results in this type of a profile. In addition, the 4 nm droplet has a lower number of total molecules as compared to the 10 nm droplet resulting in lower contact angle during the initial stages of the simulation. The 10 nm droplet shows a consistent reduction in contact angle from the initial droplet configuration (180°) to the final contact angle position (95°) concerning time. Both the droplets reach a contact angle of around 95° during equilibration. Since the simulations for 10 nm droplets require substantial computational resources, the measurements of GPU performance were done exclusively for this larger system.

The nanodroplet-substrate interaction molecular model was simulated through equilibration. For the first phase of the research, the CPU only configuration was compared with the CPU+K40-GPU for T = 0.4 ns of simulation period. The atomic interactions are evaluated for time steps of $\Delta t = 2$ fs, resulting in a total of 200,000 time steps for the simulation to be completed. Figure 4 shows the instantaneous power usage and time to completion of simulation with the CPU+K40-GPU and CPU only configuration. As can be seen from the figure, the CPU+K40-GPU consumes higher instantaneous power (208 W) as compared to the CPU only (109 W) configuration. However, the CPU+K40-GPU setup has a substantial lower time to completion (3.65 h) as compared to the CPU only (21.6 h). The CPU+K40-GPU configuration shows a spiked power consumption profile. This is because, at the end of every non-bonded energy calculation with the GPU, the CPU computes the energy parameters for the next time stepping cycle.

The total energy consumption for the CPU+K40-GPU (574 Wh) was approximately 4 times lower as compared to the CPU only (2311 Wh) configuration, as illustrated in Figure 5. We further calculated the CO_2 emissions based on the United States Environmental Protection Agency (US EPA) standards. As per the US EPA, every 1000 kWh of energy consumption generates 347 kilograms of CO_2 (EPA). Typically, high-performance computation servers and workstations operate 24/7 for 365 days a year. We considered a maintenance downtime of 15 days which, results in 350 server/workstation operation days. Thus, a carbon footprint that each simulation would produce if run daily was calculated for both the CPU+K40-GPU and CPU only configurations using the above described information. Figure 6 shows that CPU only configuration (280.7 kg) produces substantial CO_2 emissions as compared to the CPU+K40-GPU (69.7 kg) setup. Thus, though the CPU only setup had a lower instantaneous power consumption, it took at least 6 times longer to perform the same computation as compared to

Figure 4. Instantaneous power usage and time to completion of simulation for K40-GPU and CPU

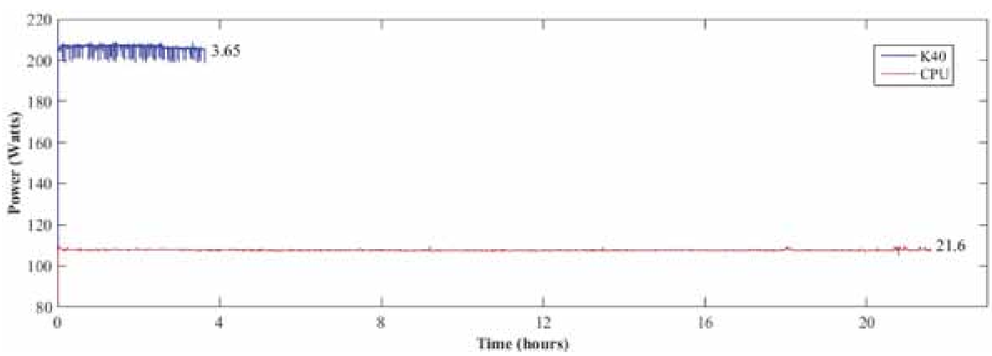


Figure 5. Total energy consumption for K40-GPU and CPU

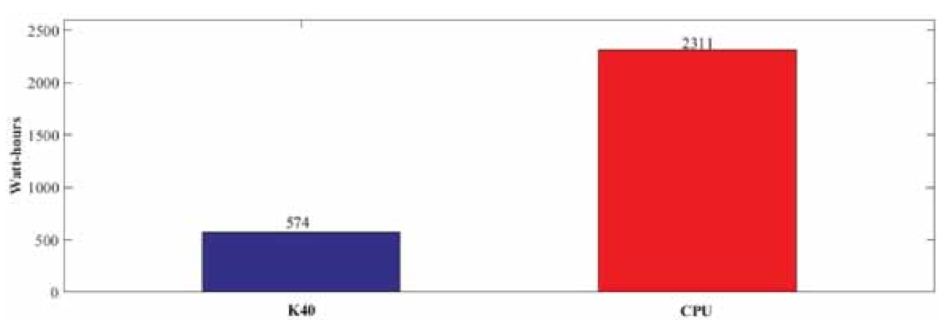
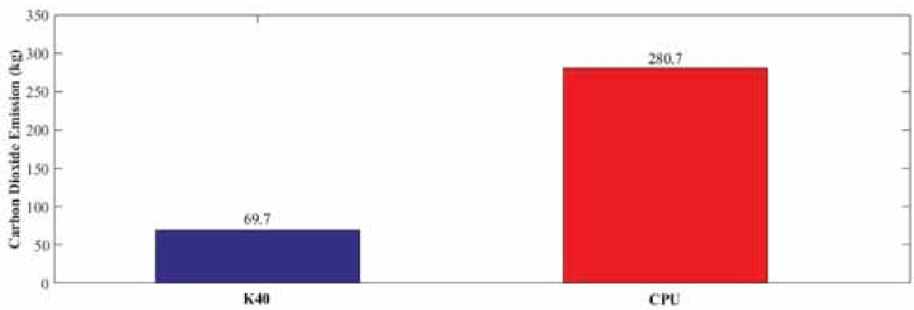


Figure 6. CO_2 emissions for K40-GPU and CPU



the CPU+K40-GPU. Moreover, it has far higher total energy consumption and carbon footprint. These facts indicate that CPU+K40-GPU configuration is both time and energy efficient for simulating large-scale molecular dynamics models.

In the second phase of the research, three GPU configurations were compared for their green computing performance. As stated earlier, GPUs must be used in combination with a CPU for performing molecular dynamics simulations based on non-bonded force and bonded energy calculations. Each of the three GPUs (C2050, K20, and K40) from NVIDIA ® Corporation have unique parallel computing architectures (Table 1). Thus, they had to be simulated at their peak operating levels to evaluate their parallel computing performance. To attain this requirement, we simultaneously simulated five identical molecular dynamic models presented in the methodology section. This procedure ensured that the K40 (highest performing GPU) was loaded to its 100% computational capacity. Thereby, the K20 and C2050 GPUs with lower performance ratings would be subsequently loaded to their 100% computational capacity. All the GPUs were evaluated for T = 0.2 ns of the simulation period. Each time step was $\Delta t = 2$ fs, resulting in a total of 100,000 time steps for the simulation to be completed. Figure 7 shows the instantaneous power usage and time to completion of simulation for the three GPUs. The C2050 consumes the highest instantaneous power (358 W) followed by the K40 (280 W) and K20 (258 W). However, the K40 has the lowest time to completion (3.55 h) followed by C2050 (4.36 h) and K20 (4.45 h), respectively. Also, the graph shows that K40 and K20 GPUs have a relatively steep powering down profile as compared to C2050. This is indicative of the fact that all 5 simulations are completed simultaneously for the K40 and K20 GPUs, while in the case of the C2050 GPU the simulations are completed in a staggered fashion.

Figure 8 shows the total power consumption for the GPUs. The K40 GPU has the lowest total energy consumption (993 Wh) followed by followed by K20 (1153.7 Wh) and C2050 (1543 Wh),

Figure 7. Instantaneous power usage and time to completion of simulation for GPUs

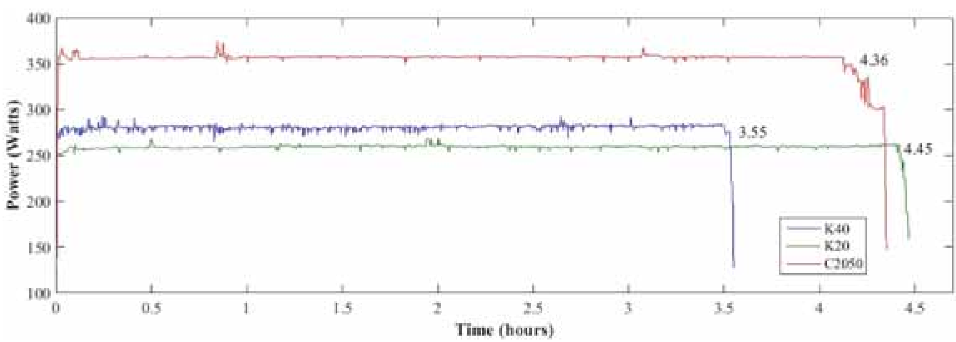
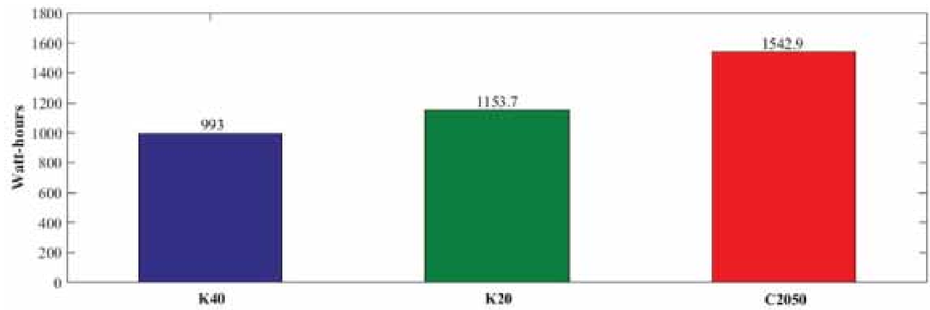


Figure 8. Total energy consumption for GPUs

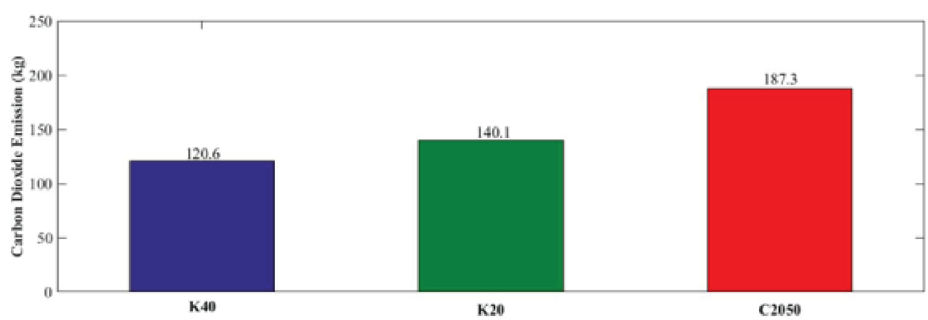


respectively. Thus, the C2050 consumes 1.55 times more energy as compared to the K40 and 1.33 times of the K20, respectively. Figure 9 shows that C2050 produces higher CO_2 emissions (187.3 kg) as compared to the K20 (140.1 kg) and K40 (120.6 kg) GPUs. Thus, though the C2050 completes the MD simulations faster than the K20, its instantaneous power consumption is much higher. The comparative analysis of GPUs based on green computing measures indicates the preference order of K40 over K20 over C2050 for simulating large-scale molecular dynamics models. It is important to note that the GPU units were executed using single threads and multithreading of the molecular dynamics code on multiple GPUs can increase the computational performance even further.

CONCLUSION

This paper benchmarks the use of graphical processing units (GPUs) over traditional central processing units (CPU) for high-performance computing applications. Four and 10 nm water nanodroplets were simulated on a flat silicon substrate using molecular dynamics (MD) modeling. Three GPUs from NVIDIA ® Corporation, C2050, K20, and K40, were implemented in combination with an Intel CPU (Intel Xeon E5-2637 v3). The performance of the CPU only configuration and each CPU+GPU configuration was evaluated concerning three green computing measures: (1) simulation completion time, (2) power consumption, and (3) CO_2 emissions. Results of this research revealed that the CPU+K40-GPU configuration had the fastest time to completion, lowest total power consumption and CO_2 emissions as compared to the CPU only setup. Moreover, the CPU only setup took 6 times longer to complete the simulation and generated 4 times the carbon footprint. A comparative analysis between the three GPUs indicated that the K40 outperformed both the K20 and C2050 concerning all the green computing measures. This research lays the foundation for the implementation of graphical

Figure 9. CO_2 emissions for GPUs



processing units (GPUs) as energy efficient alternatives to traditional CPU based cluster computing. Also, GPUs are well-suited to solve large-scale high-performance computing applications that can assist in the development of novel additive (3D Printing) nano/micro manufacturing processes.

ACKNOWLEDGMENT

The authors extend their gratitude to the US National Science Foundation (NSF CMMI Awards: 1663128 and 1435649) for support towards this research. Also, we would like to thank CNPq (National Council for Scientific and Technological Development - Brazil) for the "Science without Borders" fellowship.

REFERENCES

Adarkwa, E., & Desai, S. (2016). Scalable Droplet Based Manufacturing Using In-Flight Laser Evaporation. *Journal of Nanomanufacturing and Nanoengineering*, *6*(2), 1–6. doi:10.1166/jnan.2016.1265

Andrew, R. L. (1996). *Molecular modeling principles and applications*. Longman.

Appasami, G., & Suresh Joseph, K. (2011). Optimization of Operating Systems towards Green Computing. *International Journal of Combinatorial Optimization Problems and Informatics*, *2*(3), 39–51.

Binder, W., & Suri, N. (2009). Green computing: Energy consumption optimized service hosting. In SOFSEM 2009, LNCS (Vol. 5404, pp. 117–128). Springer.

Brooks, B. R., Bruccoleri, R. E., Olafson, B. D., States, D. J., Swaminathan, S., & Karplus, M. (1983). CHARMM: A program for macromolecular energy, minimization, and dynamics calculations. *Journal of Computational Chemistry*, *4*(2), 187–217. doi:10.1002/jcc.540040211

Buck, S. M., Xu, H., Brasuel, M., Philbert, M. A., & Kopelman, R. (2004). Nanoscale probes encapsulated by biologically localized embedding (PEBBLEs) for ion sensing and imaging in live cells. *Talanta*, *63*(1), 41–59. doi:10.1016/j.talanta.2003.12.048 PMID:18969403

Cecilia, J. M., Nisbet, A., Amos, M., García, J. M., & Ujaldón, M. (2013). Enhancing GPU parallelism in nature-inspired algorithms. *The Journal of Supercomputing*, *63*(3), 773–789. doi:10.1007/s11227-012-0770-1

Chakraborty, P., Bhattacharyya, D., Nargiza, S., & Bedajna, S. (2009). Green computing: Practice of efficient and eco-friendly computing resources. *International Journal of Grid and Distributed Computing*, *2*(3).

Cordeiro, J., & Desai, S. (2016). The Leidenfrost Effect at the Nanoscale. *Journal of Micro and Nano-Manufacturing*, *4*(4), 041001. doi:10.1115/1.4034607

Curtin, W. A., & Miller, R. E. (2003). Atomistic/continuum coupling in computational materials science. *Modelling and Simulation in Materials Science and Engineering*, *11*(3), R33–R68. doi:10.1088/0965-0393/11/3/201

Della Rocca, J., Liu, D., & Lin, W. (2011). Nanoscale metal–organic frameworks for biomedical imaging and drug delivery. *Accounts of Chemical Research*, *44*(10), 957–968. doi:10.1021/ar200028a PMID:21648429

Desai, S., & Kaware, R. (2012). Computational modeling of nanodroplet evaporation for scalable micro-nano-manufacturing. *IIE Transactions*, *44*(7), 568–579. doi:10.1080/0740817X.2011.635181

Desai, S., Kaware, R. D., & Rodrigues, J. (2014). Temperature-Dependent Wettability on Silicon Dioxide and Silicon Nitride Substrates. *Journal of Nanoengineering and Nanomanufacturing*, *4*(3), 237–246. doi:10.1166/jnan.2014.1198

EPA. (n.d.). Carbon Emission Standards. Retrieved from http://www2.epa.gov/energy/egrid

Garland, M., Le Grand, S., Nickolls, J., Anderson, J., Hardwick, J., Morton, S., & Volkov, V. et al. (2008). Parallel computing experiences with CUDA. *IEEE Micro*, *28*(4), 13–27. doi:10.1109/MM.2008.57

Harmon, R. R., & Auseklis, N. (2009, August 2-6). Sustainable IT services: Assessing the impact of green computing practices. *Paper presented at the PICMET '09 - 2009 Portland International Conference on Management of Engineering & Technology*.

Hon, K., Li, L., & Hutchings, I. (2008). Direct writing technology—Advances and developments. *CIRP Annals-Manufacturing Technology*, *57*(2), 601–620. doi:10.1016/j.cirp.2008.09.006

Jan, C. H., Bhattacharya, U., Brain, R., Choi, S. J., Curello, G., Gupta, G., & Bai, P. (2012). A 22nm SoC platform technology featuring 3-D tri-gate and high-k/metal gate, optimized for ultra low power, high performance and high density SoC applications. In *2012 IEEE International Electron Devices Meeting (IEDM)* (pp. 1-3).

Jaworek, A. (2007). Micro-and nanoparticle production by electrospraying. *Powder Technology*, *176*(1), 18–35. doi:10.1016/j.powtec.2007.01.035

Jaworek, A., & Sobczyk, A. (2008). Electrospraying route to nanotechnology: An overview. *Journal of Electrostatics*, *66*(3), 197–219. doi:10.1016/j.elstat.2007.10.001

Kalé, L., Skeel, R., Bhandarkar, M., Brunner, R., Gursoy, A., Krawetz, N., & Schulten, K. et al. (1999). NAMD2: Greater scalability for parallel molecular dynamics. *Journal of Computational Physics*, *151*(1), 283–312. doi:10.1006/jcph.1999.6201

Ko, S. H., Park, I., Pan, H., Grigoropoulos, C. P., Pisano, A. P., Luscombe, C. K., & Fréchet, J. M. (2007). Direct nanoimprinting of metal nanoparticles for nanoscale electronics fabrication. *Nano Letters*, *7*(7), 1869–1877. doi:10.1021/nl070333v PMID:17547465

Kullmann, C., Schirmer, N. C., Lee, M.-T., Ko, S. H., Hotz, N., Grigoropoulos, C. P., & Poulikakos, D. (2012). 3D micro-structures by piezoelectric inkjet printing of gold nanofluids. *Journal of Micromechanics and Microengineering*, *22*(5), 055022. doi:10.1088/0960-1317/22/5/055022

Li, K. (2012). Power allocation and task scheduling on multiprocessor computers with energy and time constraints. In A. Y. Zomaya & Y.-C. Lee (Eds.), *Energy efficient distributed computing systems*. Hoboken, NJ: Wiley. doi:10.1002/9781118342015.ch1

Lippert, E. (1960). The Strengths of Chemical Bonds, von T. L. Cottrell. Butterworths Publications Ltd., London 1958. 2. Aufl., X, 317 S., geb.t—/32/—. *Angewandte Chemie*, *72*(16), 602–602. doi:10.1002/ange.19600721618

Marquetti, I., & Desai, S. (2016). Molecular Modeling of Biochemical Cues using High Performance GPU. *Paper presented at the International Interdisciplinary Conference On Engineering Science & Management*, Goa, India.

Marquetti, I., & Desai, S. (2018a). Adsorption Behavior of Bone Morphogenetic Protein-2 on a Graphite Substrate for Biomedical Applications. *American Journal of Engineering and Applied Sciences*, *11*(2), 1037–1044. doi:10.3844/ajeassp.2018.1037.1044

Marquetti, I., & Desai, S. (2018b). Molecular modeling the adsorption behavior of bone morphogenetic protein-2 on hydrophobic and hydrophilic substrates. *Chemical Physics Letters*, *706*, 285–294. doi:10.1016/j.cplett.2018.06.015

Mayo, S. L., Olafson, B. D., & Goddard, W. A. (1990). DREIDING: A generic force field for molecular simulations. *Journal of Physical Chemistry*, *94*(26), 8897–8909. doi:10.1021/j100389a010

Nelson, M. T., Humphrey, W., Gursoy, A., Dalke, A., Kalé, L. V., Skeel, R. D., & Schulten, K. (1996). NAMD: A parallel, object-oriented molecular dynamics program. *The International Journal of Supercomputer Applications and High Performance Computing*, *10*(4), 251–268. doi:10.1177/109434209601000401

NRC. (2010). *Advancing the Science of Climate Change*. Washington, DC: The National Academies Press.

Owens, J. D., Houston, M., Luebke, D., Green, S., Stone, J. E., & Phillips, J. C. (2008). GPU computing. *Proceedings of the IEEE*, *96*(5), 879–899. doi:10.1109/JPROC.2008.917757

Parupelli, S., & Desai, S. (2017). Understanding Hybrid Additive Manufacturing of Functional Devices. *American Journal of Engineering and Applied Sciences*, *10*(1), 264–271. doi:10.3844/ajeassp.2017.264.271

Pearlman, D. A., Case, D. A., Caldwell, J. W., Ross, W. S., Cheatham, T. E. III, DeBolt, S., & Kollman, P. et al. (1995). AMBER, a package of computer programs for applying molecular mechanics, normal mode analysis, molecular dynamics and free energy calculations to simulate the structural and energetic properties of molecules. *Computer Physics Communications*, *91*(1-3), 1–41. doi:10.1016/0010-4655(95)00041-D

Phillips, J. C., Braun, R., Wang, W., Gumbart, J., Tajkhorshid, E., Villa, E., & Schulten, K. et al. (2005). Scalable molecular dynamics with NAMD. *Journal of Computational Chemistry*, *26*(16), 1781–1802. doi:10.1002/jcc.20289 PMID:16222654

Phillips, J. C., Stone, J. E., & Schulten, K. (2008). Adapting a message-driven parallel application to GPU-accelerated clusters. *Paper presented at the International Conference for High Performance Computing, Networking, Storage and Analysis SC 2008*.

Picariello, F., Rapuano, S., & Villano, U. (2013). Evaluation of power consumption of workstation computers using benchmarking. In *12th IMEKO TC10 Workshop on Technical Diagnostics New Perspectives in Measurements, Tools and Techniques for Industrial Applications*, Florence, Italy (pp. 6-7).

Ritos, K., Dongari, N., Borg, M. K., Zhang, Y., & Reese, J. M. (2013). Dynamics of nanoscale droplets on moving surfaces. *Langmuir*, *29*(23), 6936–6943. doi:10.1021/la401131x PMID:23683083

Schatz, G. C. (2007). Using theory and computation to model nanoscale properties. *Proceedings of the National Academy of Sciences of the United States of America*, *104*(17), 6885–6892. doi:10.1073/pnas.0702187104 PMID:17438274

Singh, J., Naik, K., & Mahinthan, V. (2015). Impact of Developer Choices on Energy Consumption of Software on Servers. *Procedia Computer Science*, *62*, 385–394. doi:10.1016/j.procs.2015.08.423

Velte, T., Velte, A., & Elsenpeter, R. (2008). *Green IT: Reduce your information system's environmental impact while adding to the bottom line*. McGraw-Hill Osborne Media.

Vijayaraghavan, R., Garg, A., Vijayaraghavan, V., & Gao, L. (2015). Development of energy consumption model of abrasive machining process by a combined evolutionary computing approach. *Measurement*, *75*, 171–179. doi:10.1016/j.measurement.2015.07.055

Yang, S.-C. (2006). Effects of surface roughness and interface wettability on nanoscale flow in a nanochannel. *Microfluidics and Nanofluidics*, *2*(6), 501–511. doi:10.1007/s10404-006-0096-5

Zhang, X., Gong, L., & Li, J. (2011). Research on green computing evaluation system and method. In *7th IEEE Conference on Industrial Electronics and Applications* (pp. 1177-1182).

Zhou, Z. B., Xiao, W., Yang, J., & Gong, F. (2016). Special issue on green networking, computing, and software systems. *Journal of Network and Computer Applications*, *59*, 1–3. doi:10.1016/j.jnca.2015.11.008

Zhu, R., Sun, Z., & Hu, J. (2012). Special section: Green computing. *Future Generation Computer Systems*, *28*(2), 368–370. doi:10.1016/j.future.2011.06.011

Izabele Marquetti is a Ph.D. candidate at North Carolina Agricultural and Technical State University in the Department of Industrial and Systems Engineering. She earned her Master's degree at the Universidade Tecnologica Federal do Parana, Brazil, in Food Technology with a specialty in Process in the Food Industry, and her bachelor's degree at Pontificia Universidade Catolica do Parana, Brazil, in Chemical Engineering. Her current research includes the use of molecular dynamics simulations for the studying protein-biomaterial interactions. She implements high-performance green computing for evaluating adsorption behavior of different biological growth factors on hydrophobic and hydrophilic substrates for biomedical applications.

Jhonatam Cordeiro obtained his Ph.D. in Industrial and Systems Engineering from North Carolina A&T State University. His research is focused on the nanoscale properties of materials and molecular modeling using high-performance green computing. The current emphasis of his research is the influence of substrate topography, temperature allocation, and surface energy potentials on nanoscopic fluid-substrate interactions using graphical processing units. Outside academia, Dr. Cordeiro works on developing enterprise computing systems by assisting companies to deploy a hyper-converged compute infrastructure.

Salil Desai, PhD, is a Professor in the Department of Industrial & Systems Engineering at NC A & T State University and holds adjunct faculty appointment at the Wake Forest Institute for Regenerative Medicine. He directs the Integrated Nano & Bio Manufacturing Laboratory and is an affiliate faculty of the Joint School of Nanoscience and Nanoengineering (NCA&TSU/UNCG). His research focuses on high performance green computing, hybrid nano/ micro & bio manufacturing combining direct-write fabrication and nanoimprint lithography with applications in the areas of semiconductor electronics, energy devices, drug delivery, and regenerative tissue engineering.