

Energy-Efficient Virtual Screening with ARM-CPU-Based Computers

Olivia Alford¹ and David Toth²

Centre College

600 West Walnut Street

Danville, KY 40422

olivia.alford@centre.edu, david.toth@centre.edu

ABSTRACT

We attempted to find a more sustainable solution for performing virtual screening with AutoDock Vina which uses less electricity than computers using typical x64 CPUs. We tested a cluster of ODROID-XU3 Lite computers with ARM CPUs and compared its performance to a server with x64 CPUs. In order to be a viable solution, our cluster needed to perform the screen without sacrificing speed or increasing hardware costs. The cluster completed the virtual screen in a little less time than our comparison server while using just over half the electricity that the server used. Additionally, the hardware for the cluster cost about 38% less than the server, making it a viable solution.

CCS Concepts

• Applied computing—Chemistry.

Keywords

Electricity consumption; ARM CPUs; virtual screening; AutoDock Vina.

1. INTRODUCTION

The technique of virtual screening is used in the drug discovery process to reduce the time required to discover new drugs and the costs associated with that task [1]. Virtual screening is the process of using computers to simulate how well a molecule will bind to a protein or another target, using a molecular docking program, such as AutoDock Vina [2, 3]. Using virtual screening, millions of molecules can be “tested” by computers in a relatively short time, eliminating most of them from the pool of potential cures for a particular disease. This reduces the number of molecules to test in a wet lab to a very small number and thereby reduces costs of drug discovery significantly [1]. Due to the importance of virtual screening in the drug discovery process, it is critical to complete virtual screens quickly, and thus, large compute clusters and supercomputers are often used to conduct virtual screens [4].

For years, the focus of supercomputers was on increasing their computational power without concern about other aspects of supercomputing, such as electricity consumption [5]. However, the electricity consumption of supercomputers has become a large

concern over the past years [5, 6]. This concern has led to GPUs and coprocessors being used in supercomputers to achieve more processing power while using less electricity [7, 8]. Additionally, the Green500 list was created to rank supercomputers in terms of energy efficiency [9].

2. RELATED WORK

The Mont-Blanc project is exploring the potential for Advanced RISC Machines (ARM) CPUs to be used to build next-generation supercomputers [10, 11, 12]. Advantages of ARM CPUs include lower electricity consumption and cost. However, ARM CPUs are slower than x64 CPUs. Thus, the question of whether supercomputers and clusters built from x64 CPUs or ARM CPUs can provide a lower electricity consuming method of performing the same scientific computations in the same amount of time for similar cost is important. The Mont-Blanc project has built a prototype with system on chip (SoC) computers and compared its performance to the MareNostrum III supercomputer’s performance. The Mont-Blanc prototype was slower than MareNostrum, but in some cases, more energy efficient and the prototype showed potential [11].

Toth et. al. conducted performance measurements of the programs AutoDock Vina and Dock6 on various computer-on-board products with ARM CPUs [13]. The measurements were compared to two computers with x64 CPUs. In this work, one of the computers with an ARM CPU outperformed both systems with x64 CPUs, consuming less electricity for a given task. That computer also was predicted to be able to complete the same task in the same amount of time using hardware costing less money. However, the work had two shortcomings. The first issue was that instead of conducting a full screen, the number of compounds that could be screened in 24 hours was measured.

To remove the variability of time to screen compounds, which could have led to unfair results, a single compound was screened repeatedly for 24 hours. The second issue was that the performance measurements were for a single device, which ignored the extra costs and electricity consumption of a cluster and the potential issues that would occur on a cluster, rather than a single system. These issues include slowdown from network communication between nodes and using a shared file system using the network file system (NFS). It also ignored the cost and electricity consumption of a dedicated controller node for the cluster which doesn’t perform any work for the virtual screen, but just assigns tasks to each worked node and is responsible for managing the NFS shared folder.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Copyright ©JOCSE, a supported publication of the Shodor Education Foundation Inc.
DOI: <https://doi.org/10.22369/issn.2153-4136/8/2/3>

¹ Undergraduate Student

² Corresponding Author

Work by Keipert et. al. compared the performance of the computational chemistry application GAMESS on x86 and 32-bit and 64-bit ARM CPUs [14, 15]. They found that the 32-bit ARM CPUs were more energy efficient than the x86 CPUs for completing tasks.

3. METHODS

To compare the two types of systems, we ran the same virtual screen on each type of system and recorded both the electricity and time required to conduct the screen.

3.1 Hardware

Our comparison systems were an x64 server and a cluster of ODROID-XU3 Lite SoC computers with ARM processors. The server contained four AMD Opteron 6378 processors. Each processor has 16 2.4 GHz x64 CPU cores for a total of 64 CPU cores. The server also had 64 GB of RAM, two 500 GB hard disks, and a 1000-watt power supply. The ODROID-XU3 Lite computers each have a Samsung Exynos5422 processor, which contains two quad-core processors (a CortexTM-A15 1.8 GHz quad-core processor and a CortexTM-A7 1.3 GHz quad-core processor) [16]. The ODROID-XU3 Lite computers have 10/100 Mbps Ethernet and 2 GB of RAM and are powered by a 5V4A power supply. The ODROID computers do not have persistent storage built in, and thus we used SanDisk 16 GB class 10 microSD cards for their storage. In addition, our cluster consists of a 48-port 10/100 Mbps Ethernet switch and Ethernet cables, a 240 GB SSD connected by USB 3 and an externally powered SSD enclosure, 5 power strips, and 280 1" metal standoffs. The ODROID cluster is shown in Figure 1. The cost breakdown of the cluster is shown in Table 1. To measure the electricity consumed by the virtual screens, we used P3 International P4400 Kill a Watt Electricity Usage.

3.2 Software

The server ran Ubuntu Linux 14.04.5 LTS and the computers in the cluster ran Ubuntu Linux 14.04.1 LTS. On all the computers, the graphical user interface was disabled. For the server, we ran one task per CPU core using gnu parallel [17]. On the cluster, we ran the SLURM clustering software to allow us to submit jobs so each CPU core in the cluster was always processing a molecule [18]. We used a slightly customized version of AutoDock Vina that outputs only the best score, rather than all the data that Vina outputs by default, to minimize file I/O [3]. That version is what we use to conduct virtual screens on supercomputers, servers, and clusters to improve the performance and allow for faster processing of the results of virtual screens. For the trials, we screened the full_nci_ALL_TAUTOMERS_2011 compound library from the ZINC Database, which contains 316,179 molecules [19].

4. RESULTS

We compared the data from running the virtual screen on the server with 64 x64 CPU cores to the data from running the virtual screen on the cluster of ODROID computers with ARM CPUs. The results of the virtual screens on the two platforms were identical. We compared the electricity consumed to complete the virtual screen, the time required to complete the virtual screen, and the cost of the hardware of both options. The results are summarized in Table 2. The ODROID cluster was the better solution in all three categories we measured. The electricity usage of both systems is shown in Figure 2. The cluster used only 51.8% of the electricity the server used to conduct the virtual screen. The cluster was able to complete the virtual screen in less time than the server, requiring only 94.3% of the time that the server required. The time required by each system to complete the virtual screen is shown in Figure 3. The cost of each system is shown in Figure 4. The cluster only cost 61.29% of the price of the server. Show cost parts in table here, too, to show totals.

Table 1 - Cluster Costs

Item	Cost/Unit	Units	Cost
ODROID-XU3 Lite	\$96.80	25	\$2,420.00
10/100 Mbps Ethernet cable	\$0.91	25	\$22.75
240 GB SSD	\$79.99	1	\$79.99
SSD enclosure	\$39.99	1	\$39.99
Power strips	\$24.99	5	\$124.95
280 metal standoffs	\$114.25	1	\$114.25
16 GB class 10 microSD card	\$8.25	25	\$206.25
10/100 Mbps Ethernet switch	\$139.99	1	\$139.99
Total cluster cost			\$3,148.17

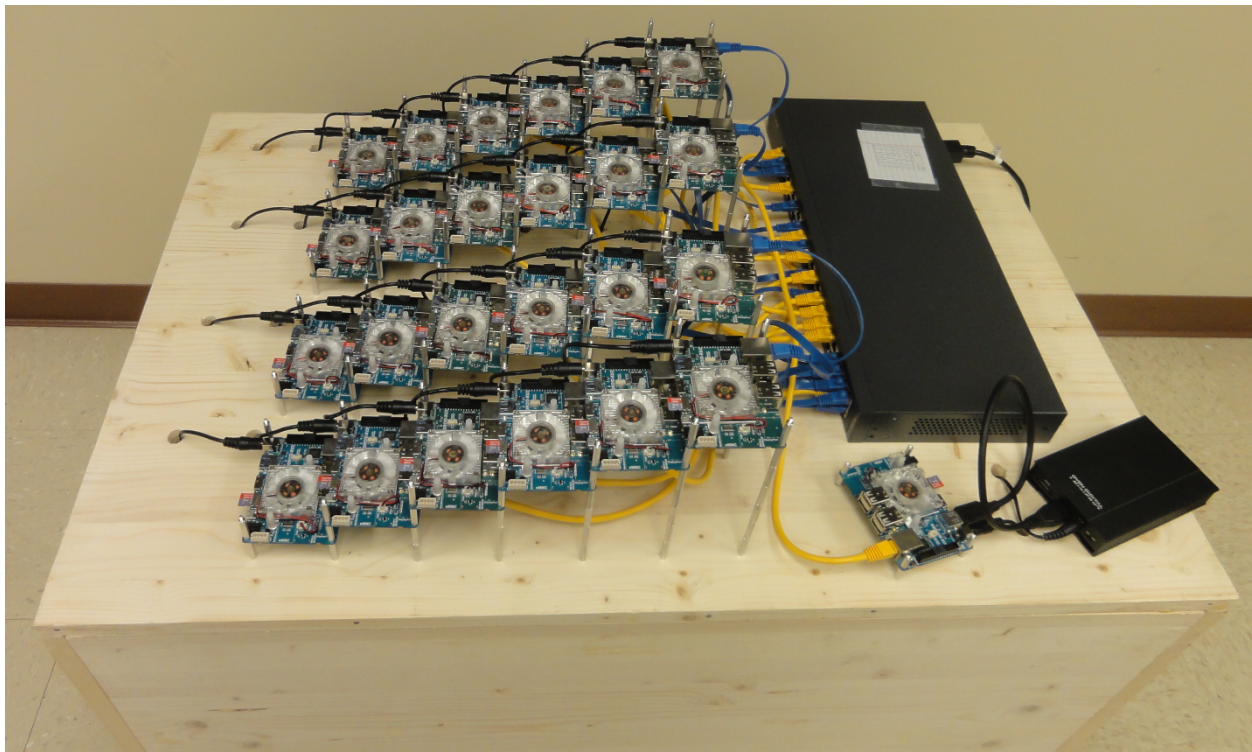


Figure 1 - The ODROID Cluster

Table 2 - Summary of Results

	Server	ODROID Cluster	Cluster's Resource Usage As Percent of Server's Resource Usage
Electricity Consumed (KWh)	86.10	44.56	51.8%
Time to Conduct Virtual Screen (sec)	513,364	484,356	94.3%
Hardware Cost	\$5136.93	\$3101.92	61.29%

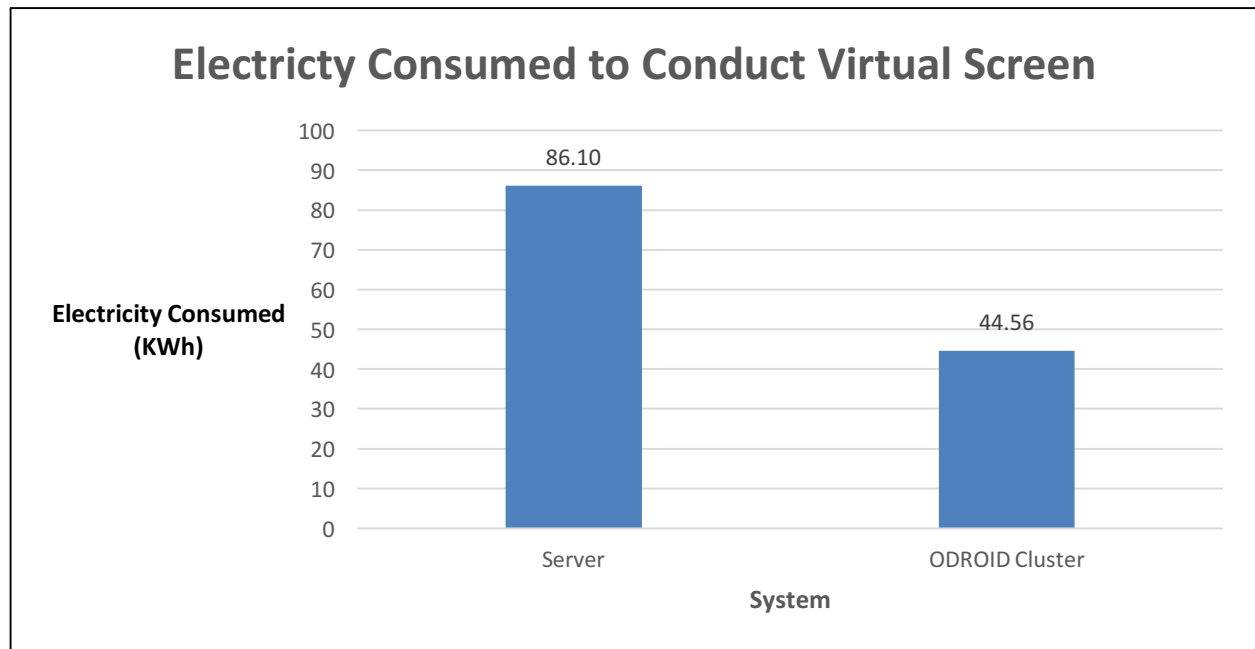


Figure 2 - Electricity Consumed by the Virtual Screen

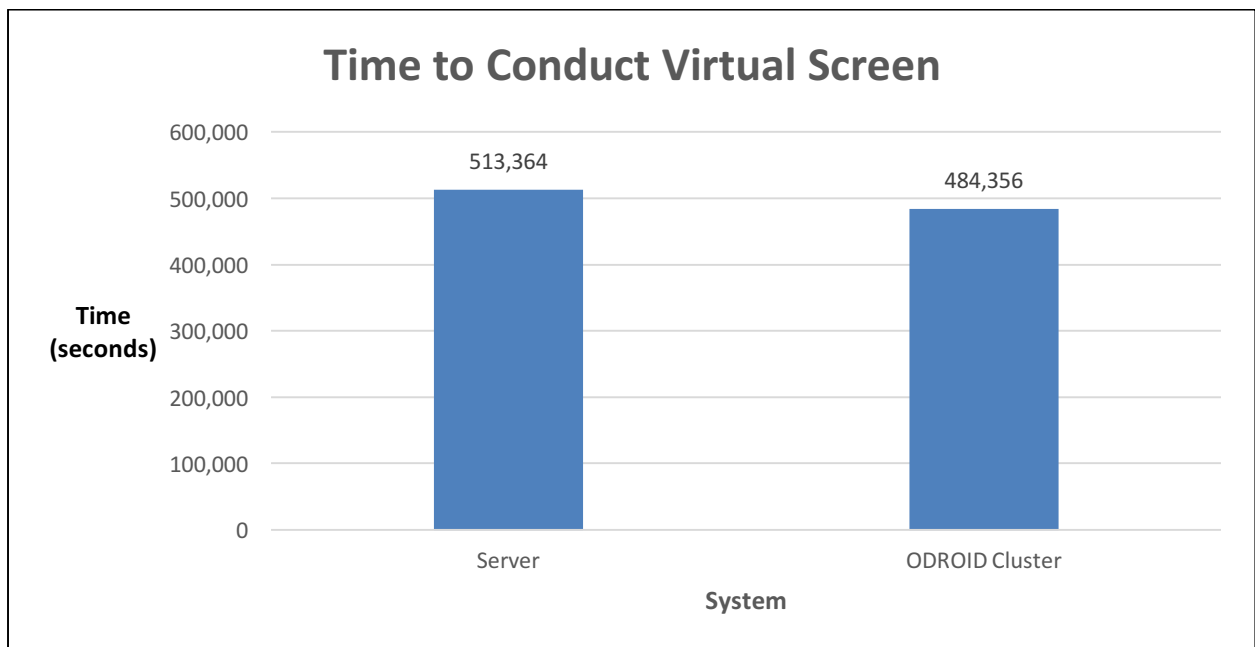


Figure 3 - Time Required to Conduct the Virtual Screen

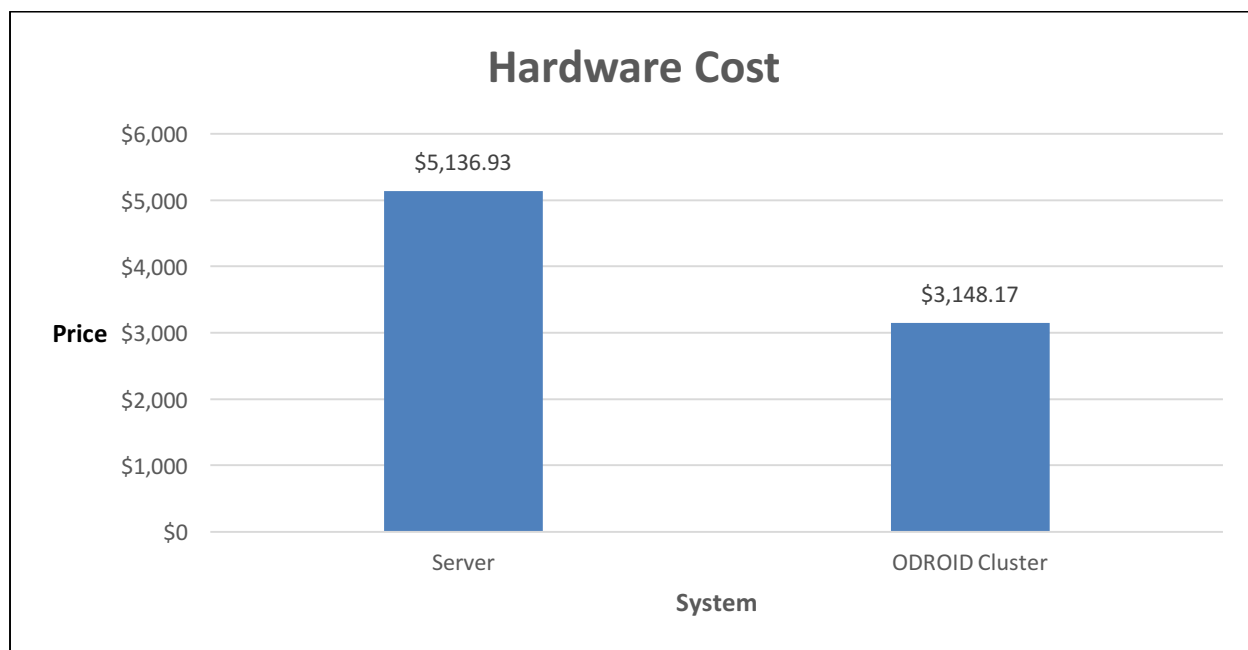


Figure 4 - The Cost of the Systems

5. CONCLUSIONS AND FUTURE WORK

We demonstrated that a cluster of ODROID-XU3 Lite computers with ARM CPUs can perform a virtual screen with AutoDock Vina using 48.2% less electricity than our x64 server. Additionally, the screen was completed using 5.6% less time and the hardware to perform the virtual screen cost about 38% less than the server. Based on our findings, clearly ARM processors have the potential to be used for scientific computing for tasks that require little memory per CPU core, like virtual screening.

However, we note that one drawback of using our cluster of ODROID-XU3 Lite computers was that we needed to flash a microSD card with the operating system installed on it for each computer. Flashing the microSD cards for an entire cluster is very time-consuming and would be a problem when producing a large cluster of SoC nodes. We are currently exploring options to avoid this process.

We also note that while the process of virtual screening worked well with our cluster, virtual screening requires only a small amount of memory per molecule being screened, so it works well on the systems with small amounts of memory per CPU core. For other scientific problems that require more memory per CPU core used, this solution may not work as well.

Additionally, virtual screening doesn't write significant amounts of output to files, so we did not run into any problems using NFS for our cluster. However, it would be interesting to see if other programs that produce more output cause issues due to the NFS share. Newer ODROID SoC computers also have Gigabit Ethernet, so that may help keep NFS from becoming a bottleneck. A hierarchical configuration where all the nodes of a particular portion of the cluster write to one NFS share while nodes in other portions of the cluster write to other NFS shares could also help prevent this from becoming a problem.

6. REFLECTIONS

I have been taking computer science courses since I was in high school, so I've known that I have a knack for the type of analytical thinking required in computer science for a while. Since I was a sophomore in high school, I have loved using computers to build things and solve problems. By the time I got to college, I was pretty set on furthering my education in computer science and eventually pursuing a job in the field, though at the time I assumed that school was the immediate concern and making a difference in the field would come post-graduation. I had never imagined that my opportunity to advance and explore the field would come while I was still an undergraduate.

Through a mentor who is very dedicated to the education of his students both in and out of the classroom, I received the opportunity to participate in the Blue Waters internship program. Once we had discussed the details, I knew that I would be spending my summer doing research, and I couldn't have been more excited. I was excited to have the opportunity to explore problems that were more applicable and important to the "real world" than those I had been working on for my classes. In addition to this, I looked forward to expanding my knowledge base beyond what the traditional class room had to offer.

The summer that I spent doing the research was memorable for many reasons. It began with my trip to University of Illinois at Urbana-Champaign for the Blue Waters Petascale Institute. This two-week workshop provided me with the unique opportunity to meet other students who excel in the field of computer science, learn new skills to bring back to my own project, and see, as well as work with, the Blue Waters supercomputer. These opportunities were unique to the Petascale Institute and something that I could never have hope to experience without this internship.

When I returned to school to begin my project I had a new expanse of knowledge with which to work, and couldn't have been more excited. The project was engaging and gave me the

opportunity to test some of my new skills in an independent setting.

Overall, I could not have asked for more from a summer experience. I was given the chance to learn new skills and put them to the test on a project that was manageable, yet challenging. Through this experience I discovered that I have an interest in hardware which I would not have otherwise discovered in my undergraduate studies. This new-found interest led me to pursue a more hardware-based internship this last summer. I found that I thoroughly enjoy writing embedded software and have chosen to pursue a master's degree in computer engineering at the University of Louisville while continuing work in embedded software after I graduate in the spring. Without this internship, it is quite possible that I would have simply stuck to software for the remainder of my college career and never discovered my interest in hardware.

7. ACKNOWLEDGMENTS

This research is part of the Blue Waters sustained-petascale computing project, which is supported by the National Science Foundation (awards OCI-0725070 and ACI-1238993) and the state of Illinois. Blue Waters is a joint effort of the University of Illinois at Urbana-Champaign and its National Center for Supercomputing Applications. We thank the Blue Waters Student Internship Program for providing Olivia with this opportunity. Finally, we thank Centre College, which provided Olivia with housing for the summer and provided us with the equipment we used for this work.

8. REFERENCES

- [1] Zhu, G (ed.). 2012. *NMR of Proteins and Small Biomolecules*. Springer Science & Business Media, Heidelberg. DOI: <https://doi.org/10.1007/978-3-642-28917-0>.
- [2] Shoichet BK. 2004. Virtual screening of chemical libraries. *Nature*. 432(7019) (Dec. 2004), 862-865. DOI: <https://doi.org/10.1038/nature03197>.
- [3] Trott, O. and Olson, A. J. AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization and multithreading. *Journal of Computational Chemistry* 31 (2010) 455-461. DOI: <https://doi.org/10.1002/jcc.21334>.
- [4] Toth, D., Franco, J., and Berkes, C. 2013. Attacking HIV, tuberculosis and histoplasmosis with XSEDE resources. In *Proceedings of the Conference on Extreme Science and Engineering Discovery Environment: Gateway to Discovery*. (San Diego, CA, July 22-25, 2013). XSEDE '13. ACM, New York, NY. DOI: <https://doi.org/10.1145/2484762.2484766>.
- [5] Sharma, S., Hsu, C., and Feng, W. Making a case for a green500 list. In *Proceedings of the 20th international conference on Parallel and distributed processing*. (Rhodes Island, Greece, Apr. 25-29, 2006). IPDPS'06. IEEE Computer Society, Washington, DC, USA. DOI: <https://doi.org/10.1109/ipdps.2006.1639600>.
- [6] Hsu, J. Supercomputer 'Titans' Face Huge Energy Costs. <http://www.livescience.com/18072-rise-titans-exascale-supercomputers-leap-power-hurdle.html>.
- [7] New top supercomputer dumps cores and increases power efficiency. <http://royal.pingdom.com/2012/11/13/new-top-supercomputer-dumps-cores-and-increases-power-efficiency/>.
- [8] Power Optimization in HPC, Enterprise and Mobile Computing. <http://www.scientificcomputing.com/article/2013/02/power-optimization-hpc-enterprise-and-mobile-computing>.
- [9] Green500 | Top500 Supercomputer Sites: <https://www.top500.org/green500/>.
- [10] Francesquinia, E., Castroc, M., Pennae, P., Duprosf, F., Freitase, H., Navauxc, P., and Méhautg, J. On the Energy Efficiency and Performance of Irregular Application Executions on Multicore, NUMA and Manycore Platforms. *Journal of Parallel and Distributed Computing* 76 (2015) 32-48. DOI: <https://doi.org/10.1016/j.jpdc.2014.11.002>.
- [11] Rajovic, N., Rico, A., Puzovic, N., Adeniyi-Jones, C., and Ramirez, A. Tibidabo: Making the case for an ARM-based HPC system. *Future Generation Computer Systems* 36 (2014) 322-334. DOI: <https://doi.org/10.1016/j.future.2013.07.013>.
- [12] Rajovic, N., Rico, A., Mantovani, F., Ruiz, D., Vilarrubi, J., Gomez, C., Backes, L., Nieto, D., Servat, H., Martorell, X., Labarta, J., Ayguadé, E., Adeniyi-Jones, C., Derradji, S., Gloaguen, H., Lanucara, P., Sanna, N., Mehaut, J-F., Pouget, K., Videau, B., Boyer, E., Allalen, M., Auweter, A., TAFANI, D., Brayford, D., Brömmel, D., Halver, R., Meinke, J., Beivide, R., Benito, M., Vallejo, E., Valero, M., Ramirez, A. The Mont-Blanc prototype: An Alternative Approach for HPC Systems. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. (Salt Lake City, UT, Nov. 13 - 18, 2016). SC '16. DOI: <https://doi.org/10.1109/sc.2016.37>.
- [13] Toth, D., Kamperman, F., and Paige, E. Making Drug Discovery More Sustainable by Reducing the Electricity Consumption and Cost of Virtual Screening. In *Proceedings of the Second International Conference on Computer Science, Computer Engineering, and Education Technologies*. (Kuala Lumpur, Malaysia, Sept. 8-10, 2015). CSCEET 2015.
- [14] Keipert, K., Mitra, G., Sunriyal, V., Leang, S., Sosonkina, M., Rendell, A., and Gordon, M. Energy-Efficient Computational Chemistry: Comparison of x86 and ARM Systems. *Journal of Chemical Theory and Computation* 11 (2015) 5055-5061. DOI: <https://doi.org/10.1021/acs.jctc.5b00713>.
- [15] Tiwari, A., Keipert, K., Jundt, A., Peraza, J., Leang, S., Laurenzano, M., Gordon, M., and Carrington, L. Performance and energy efficiency analysis of 64-bit ARM using GAMESS. In *Proceedings of the 2nd International Workshop on Hardware-Software Co-Design for High Performance Computing*. (Austin, TX, Nov. 15, 2015). Co-HPC '15. ACM, New York, NY. DOI: <https://doi.org/10.1145/2834899.2834905>.
- [16] ODROID | Hardkernel: http://www.hardkernel.com/main/products/prdt_info.php?g_code=G141351880955&tab_idx=2.
- [17] Gnu parallel: <https://www.gnu.org/software/parallel/>.
- [18] Slurm Workload Manager: <http://slurm.schedmd.com/>.
- [19] Irwin, J. J., Sterling, T., Mysinger, M. M., Bolstad, E. S., and Coleman, R. G., ZINC: A Free Tool to Discover Chemistry for Biology. *Journal of Chemical Information and Modeling*

57 (7) (2012) 1757-1768. DOI:
<https://doi.org/10.1021/ci3001277>.