Estimating the Potential for Reduced Energy Consumption for Scientific Workflows by using DVFS in Nextflow

Kevin Trogant

January 27, 2025

1 Introduction

Many scientific research questions, for example analysis of genomic variations [10], genome sequence processing [8] or earthquake hazard characterization [2] require processing of very large datasets. These processes are often expressed as *scientific workflows* [4], which are comprised of a set of computational tasks and the dependencies between them. Scientific workflows are executed by specialized *scientific workflow systems*, which allows the user to specify the workflow and manages the execution of the tasks. As part of this, the systems scheduler makes sure that a tasks dependencies are satisfied before it runs. For analyzing large datasets, it is desirable to execute these tasks in a distributed environment on multiple nodes, for example on a cluster or in the cloud. A scientific workflow system can take care of assigning tasks to nodes on which they should be executed, which includes ensuring that all required data is accessible by the task. It is also possible that this functionality is implemented by a dedicated scheduler.

Nextflow [24] is a scientific workflow system that allows the user to specify their workflow in a domain specific language. The specification is independent from the execution environment: The same workflow can be executed on the users local machine or in a cluster, without changing the specification. Nextflow has seen some use in scientific projects (for examples see: [18, 20, 17]). For example, it has been used to detect emerging variants of the SARS-Cov2 virus [16], which causes the COVID-19 disease.

Execution on clusters or clouds involves large numbers of physical machines and infrastructure, often in dedicated data centers. Their operation consumes great amounts of energy, which has motivated research focused on ways to reduce the energy consumption of the data centers themselves [12] and the compute tasks that run on them [22, 9]. *Dynamic voltage and frequency scaling* (DVFS) is an optimization technique used to reduce energy consumption. The basic idea is to reduce the frequency or voltage at which the processor operates. This offers a trade-off: By decreasing the processor performance, therefore increasing task runtime, it reduces the energy consumption of the processor. Previous studies have shown that power consumption is approximately proportional to the square of the voltage and to the frequency at which the processor operates. Generally speaking, voltage is linearly related to frequency [3], which means that power is proportional to the cube of the frequency. This has lead to a common [11, 6, 7] way to estimate power consumption as a function of CPU frequency as follows:

$$P(f) = \alpha \cdot f^3 + P_{static}$$

where P_{static} is basic energy consumption due to the processor running, f is the operating frequency in MHz and α is a CPU dependent coefficient. This model implies that a frequency reduction can significantly reduce the power consumption of the CPU.

We aim to combine dynamic voltage and frequency scaling with a scientific workflow system on real hardware. The goal is to investigate the potential for energy savings by executing scientific workflows with different CPU frequencies and measuring the consumed energy. The gained insights can be used to determine if integrating dynamic voltage and frequency scaling into a scientific workflow system is a viable approach for reducing the energy consumption of scientific workflows.

2 Related Work

Wang et al. [5] proposed a scheduling algorithm for task graphs that uses DVFS to reduce energy consumption of CPUs. A task graph has a *critical path* that provides a lower bound for its runtime [1]. The critical path is formed by the sequence of dependent tasks (which have to be executed serially) from an entry task to an exit task with the longest execution time. It is easy to see that the total execution time of the task graph cannot be lower than the execution time of its critical path. Wang et al. observe that tasks not on the critical path have some *slack time*: An amount of time by which their execution might be delayed or extended without impacting overall execution of the whole set of tasks. When executing a non-critical task their system uses the slack time of the task to determine a reduced operating frequency for the node that executes it. In simulations, their scheduling algorithm reduced energy consumption.

Silva et.al. [14] analyzed two production scientific workflows on a distributed platform instrumented with power meters. According to their work the two workflows, *SoyKB* [10] and *Epigenomics* [8] have many I/O intensive tasks and diverse CPU usages. Their goal was to compare the measured energy consumption of these workflows to a common assumption that the power used by a task is linearly related to its CPU utilization. By analyzing their measurements, the authors found that energy consumption depends not only on CPU utilization, but also on I/O operations. To improve the commonly used model for energy consumption, they propose a model that considers both CPU utilization and I/O operations. It assumes that power consumption can be expressed as the sum of power consumption caused by CPU utilization and I/O operations.

Choudhary et.al. [15] combine DVFS and task clustering to develop an energy-aware scheduling algorithm for scientific workflows. Task clustering is a technique that combines fine-grained tasks and executes them as one coarser task to reduce scheduling overhead. Their work focuses on cloud environments, where virtual machines (VMs) are provisioned by the cloud provider. Virtual machines are usually billed based on usage time and their performance. This introduces an additional problem dimension for a scheduling algorithm, because some tasks may require more expensive and higher performance VMs, while others can be executed on cheaper low performance machines. The algorithm proposed by Choudhary et.al. attempts to assign tasks to virtual machines where they will consume the least amount of energy while still satisfying other constraints, such as resource availabilty. The algorithm also dynamically provisions new VMs when necessary. The authors report that their algorithm reduces energy consumption an cost of execution in cloud environments in simulations.

3 Approach

The goal of this project is to investigate the potential for energy savings with a real (i.e., not simulated) workflow scheduling system that employs dynamic voltage and frequency scaling techniques. To achieve this goal, we will measure the energy consumption of example workflows using a range of CPU frequencies.

As a first step, we will set up Nextflow [24], a workflow system capable of executing tasks on a single local machine, clusters of machines and in the cloud. It is important to make sure that the measurements are representative of workflows that would be used in real-life scenarios. To ensure this, we will use workflows from nf-core [13], a curated collection of open-source data analysis workflows for nextflow. Because DVFS impacts the energy consumption of CPU and file I/O bound workflows in different ways, we will use workflows with different characteristics regarding their I/O and CPU loads. Additionally, the size of the input data set likely impacts the ratio of time spent on I/O versus on computation. Intuitively, this should affect the amount of saved energy via DVFS. A goal of this project will be to determine if this effect actually occurs and how large its impact is. To achieve this, we will use multiple different input data sets with different sizes.

The next step would be to set up a system for measuring energy consumption during workflow execution. We plan to use Intel RAPL [19, Chapter 16.10] to measure the power consumed by the CPU during execution. RAPL is an interface (realized through a set of registers) that reports the energy consumption of various parts of the system, most notably of the CPU package and system RAM. There are various ways of accessing RAPL information from user space on Linux systems. As part of this project, we will have to determine the best way to achieve this. One straightforward way to implement this could be to program a script that runs concurrently to the workflow execution, samples the energy consumption reported by RAPL at a fixed frequency and writes the information into a file suitable for evaluation. One challenge as a result of using RAPL is that it always measures the whole CPU, not individual processes. Because a real system has many concurrently running processes, not all the energy consumption will be caused by the workflow task. As a simplifying assumption, it seems plausible that energy consumption is related to the CPU utilization of the task, which can be determined via standard tools such as *top* [21].

After a system has been prepared to take measurements, workflows can be executed with CPU frequency limited to different values. This step will likely take a long time, but it will hopefully require only minimal direct interaction. The simplest approach to this would be to execute all workflows on a single physical machine. Nextflow supports this approach with its *local* executor. A more realistic execution environment would use multiple nodes, for example via the *kubernetes* executor, which executes tasks in containerized environments.

After measurements have been taken, evaluation can begin. We want to answer the following questions:

- What is the relationship between CPU frequency and consumed energy?
- How large is the impact of the reduced CPU frequency on the workflow execution time?
- How does the size of the input data influence the energy consumption?

If time permits, there might be an opportunity to take more fine grained measurements. Nextflow permits the user to change which program is used to execute tasks [23, Process Reference Section], by default the bash shell is used. It may be possible to write a wrapper program that launches bash and takes energy consumption measurements during the process execution. This would allow us to associate measurements with specific tasks. However, there are several complicating factors: Tasks can be executed concurrently, which means that that energy measurements for one tasks would be influenced by concurrent tasks. Furthermore, it is unclear (from reading the manual), which requirements such a program would need to satifsy in order to work with nextflow. The tool would likely need to imitate the behavior of bash in all scenarios that could occur during execution. It is hard to predict how much time the development of such a tool would require.

References

[1] Yu-Kwong Kwok and I. Ahmad. “Dynamic critical-path scheduling: an effective technique for allocating task graphs to multiprocessors”. In: *IEEE Transactions on Parallel and Distributed Systems* 7.5 (1996), pp. 506–521. DOI: 10.1109/71.503776.

[2] Scott Callaghan et al. “Reducing Time-to-Solution Using Distributed High-Throughput Mega-Workflows - Experiences from SCEC CyberShake”. In: *2008 IEEE Fourth International Conference on eScience* (2008), pp. 151–158. URL: https://api.semanticscholar.org/CorpusID:6284075.

[3] Elmootazbellah Elnozahy, Michael Kistler, and Ramakrishnan Rajamony. “Energy-Efficient Server Clusters”. In: vol. 2325. Jan. 2008, pp. 179–197. ISBN: 978-3-540-01028-9. DOI: 10.1007/3-540-36612-1_12.

[4] Bertram Ludäscher, Shawn Bowers, and Timothy McPhillips. “Scientific Workflows”. In: *Encyclopedia of Database Systems*. Ed. by LING LIU and M. TAMER ÖZSU. Boston, MA: Springer US, 2009, pp. 2507–2511. ISBN: 978-0-387-39940-9. DOI: 10.1007/978-0-387-39940-9_1471. URL: https://doi.org/10.1007/978-0-387-39940-9_1471.

[5] Lizhe Wang et al. “Towards Energy Aware Scheduling for Precedence Constrained Parallel Tasks in a Cluster with DVFS”. In: *Proceedings of the 10th IEEE/ACM International Conference on Cluster and Cloud and Grid Computing*. CCGRID ’10. Melbourne, Victoria, Australia: IEEE Computer Society, Washington, DC, USA, May 2010, pp. 368–377. ISBN: 978-0-7695-4039-9. DOI: 10.1109/CCGRID.2010.19. URL: https://laszewski.github.io/papers/vonLaszewski-10-ccgrid.pdf.

[6] Jean-Marc Pierson and Henri Casanova. “On the utility of DVFS for power-aware job placement in clusters”. In: *Euro-Par 2011 Parallel Processing: 17th International Conference, Euro-Par 2011, Bordeaux, France, August 29-September 2, 2011, Proceedings, Part I 17*. Springer. 2011, pp. 255–266.

[7] Tom Guérout et al. “Energy-aware simulation with DVFS”. In: *Simulation Modelling Practice and Theory* 39 (2013), pp. 76–91.

[8] Gideon Juve et al. “Characterizing and profiling scientific workflows”. In: *Future Generation Computer Systems* 29.3 (2013). Special Section: Recent Developments in High Performance Computing and Security, pp. 682–692. ISSN: 0167-739X. DOI: https://doi.org/10.1016/j.future.2012.08.015. URL: https://www.sciencedirect.com/science/article/pii/S0167739X12001732.

[9] Anne-Cécile Orgerie, Marcos Dias de Assunção, and Laurent Lefèvre. “A survey on techniques for improving the energy efficiency of large-scale distributed systems”. In: *ACM Comput. Surv.* 46.4 (2013), 47:1–47:31. DOI: 10.1145/2532637. URL: https://doi.org/10.1145/2532637.

[10] Trupti Joshi et al. *Next generation resequencing of soybean germplasm for trait discovery on xsede using pegasus workflows and iplant infrastructure*. 2014.

[11] Songyun Wang et al. “A DVFS Based Energy-Efficient Tasks Scheduling in a Data Center”. In: *IEEE Access* 5 (2017), pp. 13090–13102. DOI: 10.1109/ACCESS.2017.2724598.

[12] Torsten Wilde. “Assessing the Energy Efficiency of High Performance Computing (HPC) Data Centers”. en. PhD thesis. Technische Universität München, 2018, p. 107. URL: https://mediatum.ub.tum.de/1399734.

[13] Philip A Ewels et al. “nf-core: Community curated bioinformatics pipelines”. In: *bioRxiv* (2019). DOI: 10.1101/610741. eprint: https://www.biorxiv.org/content/early/2019/04/16/610741.full.pdf. URL: https://www.biorxiv.org/content/early/2019/04/16/610741.

[14] Rafael Ferreira da Silva et al. “Characterizing, Modeling, and Accurately Simulating Power and Energy Consumption of I/O-intensive Scientific Workflows”. In: *J. Comput. Sci.* 44 (2020), p. 101157. DOI: 10.1016/J.JOCS.2020.101157. URL: https://doi.org/10.1016/j.jocs.2020.101157.

[15] Anita Choudhary et al. “Energy-aware scientific workflow scheduling in cloud environment”. In: *Clust. Comput.* 25.6 (2022), pp. 3845–3874. DOI: 10.1007/S10586-022-03613-3. URL: https://doi.org/10.1007/s10586-022-03613-3.

[16] Valentin Tilloy et al. “ASPICov: An automated pipeline for identification of SARS-Cov2 nucleotidic variants”. In: *PLOS ONE* 17.1 (Jan. 2022), pp. 1–5. DOI: 10.1371/journal.pone.0262953. URL: https://doi.org/10.1371/journal.pone.0262953.

[17] Juliette Hayer et al. “Baargin: a Nextflow workflow for the automatic analysis of bacterial genomics data with a focus on Antimicrobial Resistance”. In: *J. Open Source Softw.* 8.90 (2023), p. 5397. DOI: 10.21105/JOSS.05397. URL: https://doi.org/10.21105/joss.05397.

[18] Viktória Spisaková, Lukás Hejtmánek, and Jakub Hynst. “Nextflow in Bioinformatics: Executors Performance Comparison Using Genomics Data”. In: *Future Gener. Comput. Syst.* 142 (2023), pp. 328–339. DOI: 10.1016/J.FUTURE.2023.01.009. URL: https://doi.org/10.1016/j.future.2023.01.009.

[19] *Intel® 64 and IA-32 Architectures Software Developer’s Manual.* Available at https://www.intel.com/content/www/us/en/developer/articles/technical/intel-sdm.html. Intel Corporation. 2024.

[20] Ryan P. Seaman et al. “A cloud-based training module for efficient *de novo* transcriptome assembly using Nextflow and Google cloud”. In: *Briefings Bioinform.* 25.4 (2024). DOI: 10.1093/BIB/BBAE313. URL: https://doi.org/10.1093/bib/bbae313.

[21] *top(1) - Linux manual page.* Online version of the linux manual page hosted by Michael Kerrisk at https://man7.org/linux/man-pages/man1/top.1.html. 2024.

[22] Mehul Warade et al. “Energy and Scientific Workflows: Smart Scheduling and Execution”. In: *J. Inf. Sci. Eng.* 40.5 (2024), pp. 957–977. URL: http://jise.iis.sinica.edu.tw/JISESearch/pages/View/PaperView.jsf?keyId=200%5C_2713.

[23] *Nextflow Documentation*. Available at https://www.nextflow.io/docs/latest/index.html. Seqera Labs, S.L. 2025.

[24] *Nextflow: Data-driven computational pipelines*. https://www.nextflow.io/. Accessed: 7th January 2025.