Thesis proposal: Energy-efficient execution on mustard

Relevant grades

Operating systems - EDA093
Project in parallel computing - DAT630
Introduction to data science and AI - DAT565
Sustainable computing - DAT278
Interconnection networks - DAT575
High-performance parallel programming - DAT400
Data communication - LEU061
Sustainable development - critical perspectives and possible solutions - FFR102
Algorithms and data structures - LET375
Introduction to real time systems - LET627
Computer programming - LEU483
Linear algebra - LMA212
Multivariable analysis - MVE035
Real analysis - TMA976

1.Introduction

The number of GPUs in use worldwide is increasing as companies invest heavily in AI. When accelerators are utilised for both training and inference, the computations they perform impose significant energy demands and performance requirements. Data centres accounted for 4.4% of energy consumption in the USA in 2023, and this is estimated to increase to between 6.7% and 12% by 2028. Progress in energy-efficient computing can make a sizeable impact on global energy consumption.[2]

This thesis project shall investigate how energy consumption can be reduced during computations on GPU clusters for statically schedulable computation problems through dynamic frequency and voltage scaling (DVFS).

2.Context

Turimbetov et al. have recently published a paper in which they moved scheduling and load-balancing computations to GPUs. Seeing considerable performance improvements as a consequence of removing CPU-GPU communication. The resulting software from their research is called Mustard[3].

In their paper, Turimbetov et al. make no statements regarding whether any performance predictions are made in conjunction with scheduling tasks. The ability to predict computation time across different levels of the program, combined with the task dependency graph itself, can indicate which parts of the program are resilient to performance degradation, while minimising the increase in the program's overall computation time. Furthermore, both tuning for core and memory clock frequencies is available. Depending on how compute- or data-intensive a task is, selecting to lower either or both should have a different impact on performance[3].

Managing GPU power can be done programmatically using the NVIDIA Management Library [4]. The API supports features such as power management and active processes. It should be possible to implement DVFS decisions utilising NVML. However, NVML api calls involve the CPU.

Since this thesis proposal aims to minimise adverse impacts on computational time, the research results have the potential to be applied across settings where Mustard is used.

3. Goals & challenges

1) At what level should DVFS decisions be made? More granular decisions risk incurring higher compute overhead, whereas more coarse-grained decisions risk overlooking potential locations where DVFS could be applied at low cost.

2) How can performance and energy predictions be made accurately, while minimising overhead?
3) Is it possible to implement DVFS using NVML (or some alternative method) without involving the CPU in such a way that the GPU-CPU calls do not become the bottleneck?
4) How do the different solutions compare against offline DVFS tuning on different granularity levels?
5) Is it possible to predict how memory/compute-intensive a subsection of the program is and apply DVFS accordingly in a cluster of GPUs?

4. Approach

During the literature review, the forefront performance and energy prediction algorithms will be identified and studied. The goal is to find and compare algorithms with different qualities, such as low overhead, high accuracy, and decision granularity.

Furthermore, research into implementing the algorithms on the GPU side will be conducted. As the Mustard framework is based on making decisions entirely on the GPU side, any extensions made to the framework would preferably be GPU-side computations as well.

For the implementation stage, a benchmark suite will be devised. Different benchmarks will target different qualities. The goal is to develop or find a suite of benchmarks which can measure how the different predictive algorithms perform in various scenarios, including but not limited to:

- Compute-intensive scenario
- Data-intensive scenario
- Course/fine-grained tasks

Microbenchmarks will also be devised to measure:
- The execution time for execution-time estimation algorithm decisions
- The execution time for different energy prediction algorithm decisions

The final goal is to determine the best way to apply DVFS to GPUs when running computations with the mustard framework, given the desired characteristics.

5. References:

[1] E. Agostini, D. Rossetti, and S. Potluri, “Offloading communication control logic in GPU accelerated applications,” in *Proc. 2017 17th IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput. (CCGRID)*, Madrid, Spain, 2017, pp. 248–257, doi: 10.1109/CCGRID.2017.29.

[2] A. Shehabi *et al.*, “2024 United States Data Center Energy Usage Report,” Lawrence Berkeley National Lab, Berkeley, CA, 2024, doi: 10.71468/P1WC7Q.

[3] I. Turimbetov, M. Wahib, and D. Unat, “A device-side execution model for multi-GPU task graphs,” in *Proc. 39th ACM Int. Conf. Supercomput. (ICS ’25)*, 2025, pp. 384–396, doi: 10.1145/3721145.3730426.

[4] NVIDIA Corporation, "Management Library (NVML)," *NVIDIA Developer*. [Online]. Available: https://developer.nvidia.com/management-library-nvml. [Accessed: Dec. 12, 2025].