Commun. Comput. Phys.
doi: 10.4208/cicp.OA-2024-0233

Vol. **37**, No. 5, pp. 1358-1382
May 2025

Application of Computational Modelling to Particle Physics

Marco Barbone[1,*], Alexander Howard[1], Mihaly Novak[2], Wayne Luk[1], Georgi Gaydadjiev[3] and Alex Tapper[1,*]

[1] *Imperial College London, London, United Kingdom.*
[2] *European Laboratory for Particle Physics (CERN), Geneva, Switzerland.*
[3] *Delft University of Technology, Delft, Netherlands.*

Received 27 September 2024; Accepted (in revised version) 11 February 2025

Abstract. This study introduces a methodology for forecasting accelerator performance in Particle Physics algorithms. Accelerating applications can require significant engineering effort, prototyping and measuring the speedup that might finally result in disappointing accelerator performance. The proposed methodology involves performance modelling and forecasting, enabling the prediction of potential speedup, identification of promising acceleration candidates, prior to any significant programming investment. By predicting worst-case scenarios, the methodology assists developers in deciding whether an application can benefit from acceleration, thus optimising effort. A Monte Carlo simulation example demonstrates the effectiveness of the proposed methodology. The result shows that the methodology provides a reasonable estimate for GPUs and, in the context of FPGAs, the predictions are extremely accurate, within 2% of the realised execution time.

AMS subject classifications: 68U01

Key words: High performance computing, Monte Carlo, FPGA acceleration, GPU Acceleration, performance modelling.

1 Introduction

Programming accelerators such as Graphics Processing Units (GPUs), Field-Programmable Gate Arrays (FPGAs), etc., presents many different challenges that developers need to overcome to successfully accelerate application. To avoid investing

*Corresponding author. *Email addresses:* m.barbone19@imperial.ac.uk (M. Barbone), a.tapper@imperial.ac.uk (A. Tapper), alexander.howard@cern.ch (A. Howard), mihaly.novak@cern.ch (M. Novak), w.luk@imperial.ac.uk (W. Luk), g.n.gaydadjiev@tudelft.nl (G. Gaydadjiev)

 ©2025 Global-Science Press

needless development time and resources into accelerating applications that may not benefit from acceleration, it is important to model and forecast the performance. This is most evident in the case of FPGAs where compilation time can take up to several days. Hence, it is crucial to accelerate only applications that can achieve a significant speedup whilst keeping the number of compilation iterations to a minimum.

This paper proposes methodology that aims to accelerate only the relevant hotspots while minimising changes to the original code-base. Minimising the number of changes has an additional benefit of reducing the likelihood of introducing bugs, which can be extremely difficult to fix due to the asynchronous execution and extreme threading of code on accelerators compared to the host CPU. Performance modelling provides an estimate of the potential speedup that can be achieved by off-loading portions of an application to an accelerator [3]. In addition, performance modelling helps to identify the most promising acceleration candidates and provides insight into the design parameters that need to be tuned to achieve optimal performance [19]. Performance forecasting involves predicting the performance of an application on an accelerator before any hardware implementation is made. This helps in the design space exploration and enables developers to identify the most promising hardware configurations and application partitioning schemes [15]. Our methodology also predicts the worst-case scenario, enabling developers to make informed decisions before investing significant engineering efforts in accelerating applications.

Notable works include PPT-GPU [2], a scalable and accurate simulation framework for predicting GPU performance; Choi et al. [7], who propose a methodology for distributed GPU performance modelling; Da Silva et al. [8], which extends the roofline model for FPGA optimisation in HLS tools; Goswami et al. [10], which introduces a machine learning-based estimator for FPGA-based CNN accelerator design; GCoM [11], which models GPU core-side stalls and predicts performance; and Voss et al. [19], who develop a methodology for reconfigurable hardware design and performance estimation.

In the context of particle physics, the two largest general-purpose experiments at the LHC, namely ATLAS and CMS, are responsible for generating $\sim$10 billion events per year with Monte Carlo (MC) detector simulation and event reconstruction. The execution of extensive event generation campaigns incurs a substantial computational cost [17]. ATLAS forecasts that by the year 2028, approximately 75% of its computational resources will be allocated for simulating various aspects of particle generation in collisions. This includes event generation, modelling interactions with detectors, converting signals into digital data, and ultimately reconstructing this data for analysis. Similar projections have been made by other experiments operating at the LHC. This significant increase in computational demand motivates the adoption of GPUs and FPGAs for MC simulations [6].

GPUs show promising results, achieving a speedup of over 7x for Leading Order calculations compared to CPUs using the MadFlow framework [5]. FPGAs also show promising results. Voss et al. in [18] implemented a simplified electromagnetic shower (EM) simulation on an FPGA that compared to CPU achieves a speedup of over 4x. Additionally, Barbone et al. in [4] demonstrated a massive speedup of 270x for Coulomb