Automating HPC Model Selection on Edge Devices

Abrar Hossain
Electrical Engineering and Computer Science
The University of Toledo

Kishwar Ahmed
Electrical Engineering and Computer Science
The University of Toledo

ABSTRACT

The increasing demand for processing power on resource-constrained edge devices necessitates efficient techniques for optimizing high-performance computing (HPC) applications. In this paper, we propose HPEE (HPC Parameter Exploration on Edge), a novel approach that formulates the parameter search space problem as a pure exploration multi-armed bandit (MAB) technique. By efficiently exploring the search space using the MAB framework, we achieve significant performance improvements, while respecting limited computational resources of edge devices. Experimental results, based on HPC application, demonstrate the effectiveness of our approach in optimizing parameter search on edge devices, offering a promising solution for enhancing HPC performance in resource-constrained environments.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability.

KEYWORDS

HPC parameter autotuning, Multi-Armed Bandit, Edge devices

ACM Reference Format:
Abrar Hossain and Kishwar Ahmed. 2023. Automating HPC Model Selection on Edge Devices. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (SC '23).* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The rapid proliferation of edge devices [1], such as smartphones and internet of things (IoT) devices, has created a demand for high-performance computing (HPC) capabilities at the edge. However, resource constraints and energy limitations pose significant challenges for achieving optimal performance on these devices. Edge devices are becoming crucial for HPC due to their ubiquity and ability to support computationally demanding applications. Processing data locally on edge devices reduces latency and ensures privacy and security. Additionally, HPC on edge devices enables offline execution, enhancing user experiences in resource-constrained environments.

Parameter search optimization is essential for achieving optimal performance in HPC applications on edge devices. However, exhaustively exploring the vast search space is impractical, necessitating efficient techniques for automatic configuration selection. The multi-armed bandit (MAB) model emerges as a suitable choice [2], dynamically allocating resources to different parameter configurations based on their performance. MAB models are well-suited for edge environments, where resource constraints and real-time performance are critical factors. Traditional approaches have limitations [3] in terms of adaptability and computational cost, making them impractical for edge devices. To address these challenges, we propose HPEE (**H**PC **P**arameter **E**xploration on **E**dge), which leverages the MAB technique to optimize the parameter search space. HPEE employs random sampling and resource allocation to identify the best-performing configurations while discarding underperforming ones. We present the design and methodology of HPEE and evaluate its performance on the NVIDIA Jetson Nano, demonstrating its effectiveness in optimizing the parameter search space for HPC applications on edge devices.

2 PROBLEM FORMULATION

HPC parameters can be continuous, discrete, or categorical with potential dependencies. We sample configurations uniformly and evaluate using performance metric $t(k, x)$. k is resource allocation and x is the parameter configuration from set X, measuring performance at time step k.

We define a resource parameter R, distributed among the configuration spaces, to search for the optimal configuration choice. We set the limit of the performance metric as $t^* = \inf_{k \to R} t_k$, which can be defined as the most optimal performance metric given a particular configuration $x \in X$ as k approaches R. We define $v^* = \inf_x t^*(x)$, which can be defined as the smallest lower bound of the performance metric for all configurations considered within the scope of our evaluation, i.e., the best configuration.

Given the uncertainty regarding the rate of variation of $t(k, x)$ for fixed k and $t(k, x)$ approaching $t^*(x)$ for fixed x, we frame the problem statement as an MAB problem. The algorithm samples configurations x from the probability distribution, and the performance metric $t(x)$ becomes a random variable with an unknown distribution due to the unknown value of $t^*(x)$.

Since the algorithm does not have prior knowledge about the speed of convergence of $t(k, x)$ to $t^*(k, x)$ or the distribution of $t(x)$, the objective is to identify a parameter configuration x that minimizes $(t^*(x) - v^*)$. We formalize our parameter search as the following optimization problem OPS (**O**ptimum **P**arameter **S**earch)

$$\begin{aligned} \text{OPS} : \underset{x}{\text{minimize}} \quad & t^*(x) - v^* \\ \text{subject to} \quad & p(k, x) < p^*, \end{aligned}$$

SC '23, November 12-17, 2023, Denver, CO

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Algorithm 1 HPEE

```
Input: Maximum computational iteration ($R$), prune factor ($\eta$)

Initialization: Initialize maximum stage number, $s_{\max} = \log_\eta R$ and budget, $B = (s_{\max} + 1) \cdot R$;
for $s \in \{s_{\max}, s_{\max} - 1, \ldots, 0\}$ do
    $n = \frac{B}{R} \frac{\eta^s}{s+1}$, $r = R\eta^{-s}$;
    for $i \in \{0, \ldots, s\}$ do
        $n_i = n\eta^{-i}$, $r_i = r\eta^i$;
        $L = \{\text{find the execution times}\}$;
        $T = \text{top\_}k(T, L, \frac{n_i}{\eta})$;
    end for
end for
return configuration with the smallest intermediate loss seen so far
```

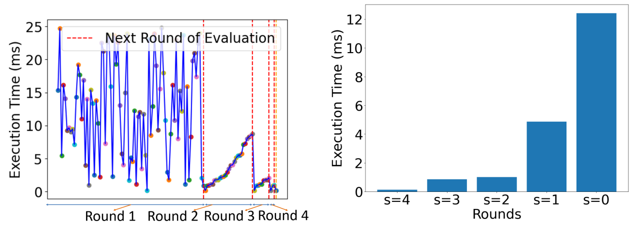
where $p(k, x)$ denotes the power consumption by the edge device during the evaluation of the configuration space, and p^* is the maximum allowable power range for the evaluation on the edge device, considering its limited power capability.

3 HPEE

HPEE presented in Algortihm 1 shows the iterative configurations refining process based on resource allocation (R) and the number of configurations (η). The algorithm operates in stages, adjusting R and η within each stage. The pruning process retains the top n_i/η configurations at each stage. Within each inner loop iteration, the algorithm dynamically adjusts the number of configurations n_i and the resource allocation r_i based on the current stage index i. It then evaluates the configurations in T by running them and calculating the corresponding execution time. The algorithm retains the top $\frac{n_i}{\eta}$ configurations based on their evaluated performances. This process continues until all stages have been iterated over. Finally, the algorithm returns the set of parameter configurations T that performed the best across all stages. The choice of R determines convergence and efficiency-accuracy balance, while η controls exploration-exploitation tradeoff. Tuning R and η allows customization of the algorithm to meet specific requirements and constraints, making it suitable for resource-constrained edge devices.

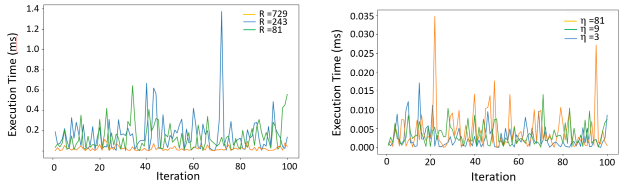
4 EVALUATION

We performed analysis using simulated data collected from AMG HPC application run on the Nvidia Jetson Nano device. For this particular setting, we evaluated performance in terms of execution time. To account for real-world variations, we introduced uniform noise. The evaluation of our optimization model for HPC parameter configuration exploration followed a systematic approach. Random sampling was conducted across the entire parameter configuration space, allocating resources to measure execution time for each sampled configuration. Iterative pruning was employed with a predetermined pruning rate, discarding configurations with extended execution times until the best-performing configuration was identified. Multiple rounds of sampling were executed, including a focused sampling phase denoted as $s = 4$ to explore the majority of the configuration space and make informed decisions on optimal configurations. The evaluation results, as depicted in Figure 2, highlight the sampling approaches and the identification of the most optimal configuration through iterative pruning. In contrast, $s = 0$ involved random sampling, resulting in poor execution times. The model's behavior was further assessed by tuning the hyperparameters R and η. Increased resource allocation ($R = 243$) yielded improved performance, and different combinations of R and η showcased variations in performance.



(a) Finding optimal configuration through successive rounds of evaluation (b) Lowest execution time across multiple rounds

Figure 1: Evaluation of the model across different parameter configuration space with R = 243 and η = 3.



(a) Performance of HPEE by customizing R. (b) Performance of HPEE by customizing η

Figure 2: Performance of HPEE by customizing R and η

5 CONCLUSION

In this paper, we proposed HPEE, an approach for optimizing parameter search space in HPC applications on edge devices. By leveraging the MAB framework, random sampling, and resource allocation, HPEE efficiently explores the configuration space. Customization options allow users to optimize performance based on specific requirements. Results demonstrate the effectiveness of HPEE, and future research can focus on broader device validation and advanced MAB strategies for improved optimization.

6 ACKNOWLEDGEMENT

This work is supported in parts by the US National Science Foundation under grant CNS-2300124.

REFERENCES

[1] Richard Neill, Alexander Shabarshin, and Luca P Carloni. 2010. A heterogeneous parallel system running open mpi on a broadband network of embedded set-top devices. In *Proceedings of the 7th ACM international conference on Computing frontiers*. Association for Computing Machinery, New York, NY, USA, 187–196.

[2] Steven L Scott. 2010. A modern Bayesian look at the multi-armed bandit. *Applied Stochastic Models in Business and Industry* 26, 6 (2010), 639–658.

[3] Xingfu Wu, Michael Kruse, Prasanna Balaprakash, Hal Finkel, Paul Hovland, Valerie Taylor, and Mary Hall. 2022. Autotuning PolyBench benchmarks with LLVM Clang/Polly loop optimization pragmas using Bayesian optimization. *Concurrency and Computation: Practice and Experience* 34, 20 (2022), e6683.