# CROSS- HPC System Bayesian Optimization with Adaptive Transfer

Abrar Hossain, Kishwar Ahmed
*Department of Electrical Engineering and Computer Science*
*The University of Toledo, USA*

***Abstract*—This paper introduces CROSS − BOAT (Cross- HPC System Bayesian Optimization with Adaptive Transfer), a novel approach for efficient parameter tuning in High-Performance Computing (HPC) systems. Optimizing the numerous configurable parameters in HPC environments typically requires expensive evaluations on each target system. We propose a transfer learning approach that leverages knowledge from a well-understood source system to accelerate optimization on new target systems. CROSS − BOAT employs an adaptive knowledge transfer mechanism that combines the expected improvement from the target system with a source knowledge term, weighted progressively to balance exploration and exploitation. Our experiments on simulated HPC systems demonstrate that CROSS − BOAT significantly outperforms standard Bayesian optimization when target systems differ substantially from the source, achieving up to 24.5% better performance with fewer system evaluations. For similar systems, standard methods remain competitive, highlighting the context-dependent value of transfer learning. These results suggest that cross-system knowledge transfer can dramatically reduce optimization time and improve configurations for new or upgraded HPC infrastructure, particularly when architectural differences are substantial.**

***Index Terms*—HPC Parameter Autotuning, Transfer Learning, Bayesian Optimization, HPC Applications, Performance Modeling.**

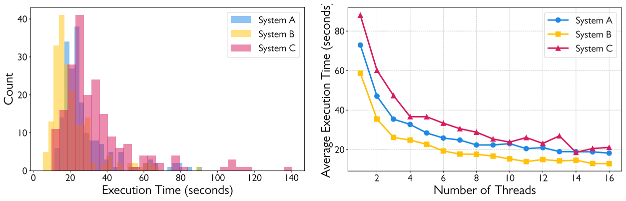
## I. INTRODUCTION

High-Performance Computing (HPC) systems require careful parameter tuning to achieve optimal performance [1]. Modern HPC applications involve numerous configurable parameters including thread count, block size, communication protocols, memory layout, and scheduling policies. Finding optimal configurations manually is impractical due to the vast search space and complex parameter interactions. Bayesian Optimization (BO) has emerged as an effective approach for automatic parameter tuning. However, traditional BO requires many expensive evaluations on target systems. This paper introduces a transfer learning [2] approach that leverages knowledge from a source HPC system to accelerate optimization on new target systems, reducing the number of required evaluations and potentially finding better configurations.

## II. PROBLEM FORMULATION

We formulate HPC parameter tuning as finding configuration $\mathbf{x}^* \in \mathcal{X} = \mathcal{X}_{\text{cont}} \times \mathcal{X}_{\text{cat}}$ that minimizes execution time across a mixed continuous-categorical parameter space.

For source system $\mathcal{S}_s$ and target system $\mathcal{S}_t$, we model performance functions as Gaussian Processes: $f_s(\mathbf{x}) \sim \mathcal{GP}(m_s, k_s)$ and $f_t(\mathbf{x}) \sim \mathcal{GP}(m_t, k_t)$ with Matérn kernels.



(a) Time Optimized.

(b) Power Optimized.

Fig. 1. Performance analysis with synthetic error in measurement data

The key challenge is efficiently optimizing $f_t$ with minimal evaluations by leveraging knowledge from $f_s$, where the relationship $f_t(\mathbf{x}) = h(f_s(\mathbf{x}), \mathbf{x}) + \epsilon$ involves an unknown transfer function $h$ that varies with parameter importance.

## III. CROSS − BOAT: CROSS-SYSTEM BAYESIAN OPTIMIZATION WITH ADAPTIVE TRANSFER

CROSS − BOAT employs an adaptive Bayesian framework using two Gaussian Process models: one for the source system trained on extensive data ($\mathcal{GP}_s$) and another for the target system with minimal initial data ($\mathcal{GP}_t$). The key innovation is our composite acquisition function:

$$a(\mathbf{x}) = (1 - w(t)) \cdot \mathrm{EI}_t^{\text{norm}}(\mathbf{x}) + w(t) \cdot \mathrm{SK}_t^{\text{norm}}(\mathbf{x})$$

Here, $\mathrm{EI}_t^{\text{norm}}$ is the normalized Expected Improvement, $\mathrm{SK}_t^{\text{norm}}$ is the normalized source knowledge term $\mathrm{SK}_t(\mathbf{x}) = 1 - \frac{\mu_s(\mathbf{x}) - \mu_s^{\min}}{\mu_s^{\max} - \mu_s^{\min}}$, and $w(t)$ is an adaptive weight decreasing from 0.5 to 0.1 as iterations progress. This balances source knowledge with target exploration, ensuring effective cross-system optimization.

## IV. EVALUATION

We evaluated CSTO on three simulated HPC systems with different parameter-performance relationships.**System A (Source)** is the base system with standard performance characteristics. **System B (Similar Target)** is similar to System A but differs in scaling factors. **System C (Dissimilar Target)** has fundamentally different parameter preferences. Each system has 7 tunable parameters. The `num_threads` parameter can range from 1 to 16 threads. The `block_size` parameter takes values of 64, 128, 256, 512, and 1024. The `storage_format` can be either row-major or column-major. The `comm_protocol` specifies the communication protocol, which can be MPI, OpenMP, or a hybrid approach. The `precision` parameter allows computations in either

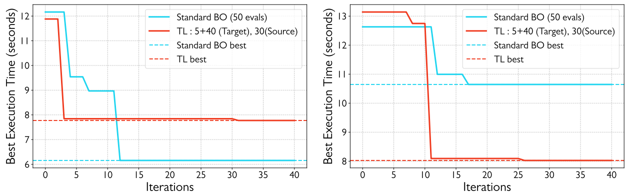
**Algorithm 1** CROSS – BOAT – CROSS-System Bayesian Optimization with Adaptive Tuning

```
Require: Source system $\mathcal{S}$, Target system $\mathcal{T}$, Parameter space $\mathcal{X}$, $n_s$, $n_t$, $T$
Ensure: Optimal configuration $\mathbf{x}^*$ for target system
// Source knowledge acquisition
$\mathcal{D}_s \leftarrow \{(\mathbf{x}_i, f_s(\mathbf{x}_i))\}_{i=1}^{n_s}$ where $\mathbf{x}_i \sim \mathrm{LHS}(\mathcal{X})$
$\mu_s, \sigma_s \leftarrow \mathrm{GP}(\mathcal{D}_s)$
// Target initialization
$\mathcal{D}_t \leftarrow \{(\mathbf{x}_j, f_t(\mathbf{x}_j))\}_{j=1}^{n_t}$ where $\mathbf{x}_j \sim \mathrm{LHS}(\mathcal{X})$
for $i = 1$ to $T$ do
    $\mu_t, \sigma_t \leftarrow \mathrm{GP}(\mathcal{D}_t)$
    $\mathbf{X}_c \leftarrow$ candidate points from $\mathcal{X}$
    $f_t^* \leftarrow \min_{(\mathbf{x},y)\in\mathcal{D}_t} y$
    // Compute acquisition components
    for $\mathbf{x} \in \mathbf{X}_c$ do
        $\mathrm{EI}_t(\mathbf{x}) \leftarrow \mathbb{E}[\max(f_t^* - f_t(\mathbf{x}), 0)]$
        $\mathrm{SK}_t(\mathbf{x}) \leftarrow 1 - \frac{\mu_s(\mathbf{x}) - \min_{\mathbf{x}'} \mu_s(\mathbf{x}')}{\max_{\mathbf{x}'} \mu_s(\mathbf{x}') - \min_{\mathbf{x}'} \mu_s(\mathbf{x}')}$
    end for
    // Normalize and combine
    $w_i \leftarrow \max\{0.5 \cdot (1 - i/T), 0.1\}$
    $\mathbf{x}_{next} \leftarrow \arg\max_{\mathbf{x}\in\mathbf{X}_c} \{(1 - w_i) \cdot \hat{\mathrm{EI}}_t(\mathbf{x}) + w_i \cdot \hat{\mathrm{SK}}_t(\mathbf{x})\}$

    $\mathcal{D}_t \leftarrow \mathcal{D}_t \cup \{(\mathbf{x}_{next}, f_t(\mathbf{x}_{next}))\}$
end for
return $\arg\min_{(\mathbf{x},y)\in\mathcal{D}_t} y$ =0
```

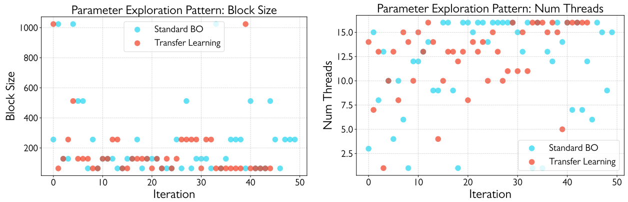
single or double precision. The `scheduling` method can be static, dynamic, or guided. Lastly, `cache_blocking` can be either enabled or disabled. These tunable parameters influence the performance and efficiency of HPC applications by optimizing resource utilization and computational throughput.

Our experiments evaluated CROSS – BOAT comparing against standard Bayesian Optimization (BO). Figure 2 shows the convergence behavior for both similar and dissimilar target systems. For System B (Fig. 2(a)), which shares architectural similarities with the source system, standard BO ultimately achieved better performance (6.15s vs. 7.77s). However, for System C (Fig. 2(b)), which has fundamentally different parameter-performance relationships, CROSS – BOAT significantly outperformed standard BO, finding configurations that were 24.5% faster (8.03s vs. 10.64s). The parameter exploration patterns in Figure 3 reveal how CROSS – BOAT navigates the parameter space differently from standard BO. For block size (Fig. 3(a)), transfer learning initially explores smaller values (128-256) more intensively before converging on optimal settings. Similarly, for thread count (Fig. 3(b)), CROSS – BOAT demonstrates more structured exploration, focusing on higher thread counts that performed well on the source system. Figure 4 illustrates how CROSS – BOAT efficiently discovers optimal configurations across key parameters. For block size (Fig. 4(a)), CROSS – BOAT quickly identifies that the smallest (64) and largest (1024) values yield the best performance on System C, avoiding the suboptimal mid-range values. For thread count (Fig. 4(b)), CROSS – BOAT consistently evaluates configurations with higher thread counts (12-16), which ultimately proved optimal for System C. These results demonstrate that CROSS – BOAT is particularly valuable when optimizing dissimilar systems, where knowledge transfer helps avoid local optima by guiding exploration toward promising regions that might otherwise be overlooked by standard optimization approaches.



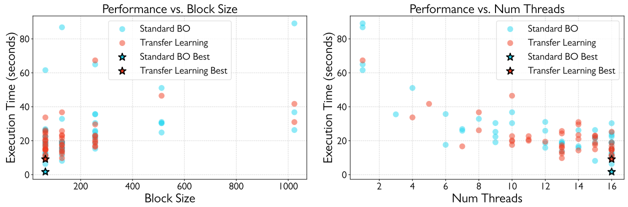
(a) System B (b) System C

Fig. 2. Comparison of Transfer Learning and Standard Bayesian Optimization



(a) Block Size (b) Number of threads

Fig. 3. Exploration pattern of CROSS – BOAT



(a) Block Size (b) Number of threads

Fig. 4. CROSS – BOAT efficiently finds the optimal configuration

## V. CONCLUSION

This paper introduced CROSS – BOAT, a transfer learning approach for HPC parameter optimization across systems, with an adaptive mechanism to balance source knowledge and target exploration. CROSS – BOAT excelled in optimizing dissimilar systems, outperforming Bayesian Optimization by avoiding local optima. Future work will extend CROSS – BOAT for multi-objective optimization, advanced transfer mechanisms, and real-world validation.

## REFERENCES

[1] Y. Hu, G. Huang, and P. Huang, "Automated reasoning and detection of specious configuration in large systems with symbolic execution," in *(OSDI'20)*.

[2] E. Saule, "Experiences on teaching parallel and distributed computing for undergraduates," in *2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pp. 361–368, IEEE, 2018.