Predictive Resource Usage Characterization for Extreme Edge Computing

Ruslan Kain, Yuanzhu Chen, and Hossam S. Hassanein
School of Computing, Queen’s University, Kingston, ON, Canada
{r.kain, yuanzhu.chen}@queensu.ca and {hossam}@cs.queensu.ca

Abstract—This paper introduces a novel Extreme Edge Computing (EEC) framework that addresses the growing computational demands of Internet of Things (IoT) devices. Our framework utilizes a distributed system of heterogeneous Extreme Edge Devices (EEDs) to enhance computational capabilities at the network edge. We present a predictive worker resource characterization scheme that integrates these devices seamlessly, overcoming challenges such as variability in task execution times and resource contention. Empirical studies on heterogeneous EEDs aim to bring enhancements in distributed computing scheduler efficiency and task allocation. Operational results illustrate dynamic task scheduling and efficient resource utilization.

Index Terms—Edge Computing, Internet of Things, Dynamic Resource Usage, Extreme Edge Devices

I. INTRODUCTION

The proliferation of Internet of Things (IoT) devices alongside advancements in generative models has escalated the demand for computational power. Despite exponential growth in compute demands—doubling every 5.8 months since 2010—GPU development has not kept pace, with a doubling only every 2.3 years [1]. Edge computing has traditionally offered a solution by enabling data processing closer to its source. Taking this concept further, Extreme Edge Computing (EEC) leverages a distributed system of diverse and dynamic Extreme Edge Devices (EEDs) to decentralize and amplify computational capabilities. Our research addresses the integration challenges of such heterogeneous devices into a cohesive EEC framework, overcoming issues like variability in task execution times and contention for computational resources due to user engagement.

We introduce a predictive worker resource characterization scheme to assess and optimize the performance of EEDs. Empirical studies conducted on a wide range of edge devices reveal that our framework has the potential to boost the efficiency of distributed computing schedulers and task allocation processes. The system’s ability to dynamically allocate benchmark tasks provides a robust mechanism for evaluating the performance of EEDs in active use. This paper showcases our framework and how it operates. Our work not only enhances the feasibility of EEC by providing a scalable and predictive approach to managing edge device resources effectively.

II. FRAMEWORK

The proposed framework, illustrated in Figure 1, is structured around three core modules that function in tandem to characterize and optimize the resource utilization of Extreme Edge Devices (EEDs). The Data Generation module [2] automates application execution across EEDs, capturing resource usage data under varied operational scenarios. This includes applications like video streaming, gaming, and cryptocurrency mining, which influence the adjustments in CPU frequencies, network conditions, and application durations.

Subsequently, the captured data is processed by the Resource Usage Prediction module [3], which employs advanced predictive models like the Hierarchical Dirichlet Process-Hidden Semi-Markov Model (HDP-HSMM) and Hybrid Bidirectional-LSTM based Encoder Decoder (HBLED) model, to forecast future resource states and usage values, for multiple step-ahead sizes. The predictions may be utilized to enable the proactive allocation of tasks.

Lastly, the Worker Resource Characterization [4] module, leverages these predictions to perform real-time benchmark tasks, composed of cpu-intensive, memory-intensive, network-intensive, and mixed types, effectively assessing and characterizing each worker’s performance and Quality of Service (QoS). By dynamically assigning these benchmark tasks based on predicted resource states, the U-WORC module ensures optimal task allocation to the most suitable devices.

III. PRELIMINARY RESULTS

A. Experimental Test Bed

To evaluate our proposed scheme, we conducted tests using a diverse array of Extreme Edge Devices (EEDs), consisting of Raspberry Pi (RPi) 4B and Nvidia Jetson Nano single-board...
computers, as shown in figure 3. The RPIs’ feature various RAM sizes (2, 4, or 8 GB) and a modifiable CPU frequency range of 0.6 to 1.8 GHz due to throttling and overclocking, and the Nano has two power modes with different performance levels, in addition to a GPU. They may also vary in bandwidth for upload and download operations. Connected via Ethernet or Wi-Fi, these devices use the axon-ECRG framework for communication within a network that hosts custom-built EEC task allocators, enabling distributed computing.

B. Preliminary Results

The EEC system’s benchmark task allocator conducts predictive scheduling, executing an array of benchmarks (CPU, memory, network, mixed) across all detected operational states, to collect comprehensive QoS data. Figure 2 shows the task allocator’s operation across two distinct worker devices, with the memory usage overlaid as a continuous line graph. This operation is crucial for characterizing each device’s performance by recording the execution times of benchmarks during state transitions. The task allocator adapts to real-time changes, optimally assigning benchmark tasks to characterize all the predicted resource usage states of the workers.

Figure 4 presents the runtime distributions for ten iterations of the CPU, memory, network, and mixed benchmarks across various operational states of the EEDs. The height of each distribution graphically represents the consistency of the obtained values, whereas the width reflects their variability. These results highlight the impact of concurrent user-end applications on computational efficiency. For instance, the idle state generally records the shortest runtime in most benchmarks, whereas more computationally intensive states lead to resource contention, thereby extending benchmark runtimes and increasing variability among states. Additionally, the disparities in the performance capabilities of the devices are evident from the variance in runtime across different benchmark-to-state combinations, enabling worker characterization.

IV. Conclusion

The proposed framework enables the timely execution of benchmark tasks to characterize heterogeneous and dynamically accessed edge devices, thereby enhancing the efficiency of task allocation schemes in distributed computing environments. Future work will involve various task allocation schemes to demonstrate how resource awareness can improve system performance. Specifically, we anticipate increased task allocation throughput, reduced average task execution times, and more efficient resource utilization of the workers.

References