

Scalable Real Time Inefficiency Detection on HPC Systems

Umbreen Sabir Mian[†], Robert Schöne

Centre for Information Services and High Performance Computing (ZIH)
Technische Universität Dresden, 01062, Dresden, Germany
{umbreen.mian, robert.schoene}@tu-dresden.de

ABSTRACT

Most of the existing performance monitoring tools available for High Performance Computing (HPC) systems are either based on application level instrumentation or sampling an application during its execution by recording its state at regular intervals. The performance analysis is then done by users offline on the collected data. In this study, we propose a scalable framework which employs software sampling to collect the application performance data, streams the collected samples through a message broker using the publish-subscribe model. The framework does performance analysis by using Machine Learning (ML) online to detect inefficiencies in running applications.

1 INTRODUCTION

HPC and supercomputers are used in almost every field of science but the efficient usage of the computing resources by the software is usually low because multitudes of applications running on the machine suffer from efficiency issues. In order to utilize the resources on a supercomputer optimally, it is inevitable to identify and detect applications that use the given resources suboptimal. Users can analyze their programs using performance analysis tools. The process of performance analysis consists of data collection, data storage and data presentation for analysis [1]. A multitude of monitoring systems are available, which continuously collect a huge amount of data on HPC systems, including system logs, resource utilization data, and data from different temperature and energy sensors. Furthermore, current state-of-the-art performance monitoring systems collect hardware resource utilization data and hardware events via hardware performance counters [2], [3]. The data collected by monitoring systems is then used either offline to do post-mortem performance analysis or recent works also use ML to automatically detect performance problems. In this work, we use a combinations of software sampling and hardware performance counters to collect application information and then detect performance inefficiencies in HPC application using ML online. With the growing size of HPC machines, it is inevitable to analyze the data automatically. For this purpose, ML techniques are employed [4], [5].

2 RESEARCH PROBLEM

The first step to implement an online inefficiency detecting framework is to select the type of data required. Application instrumentation is not suitable because the overhead gets higher with the number of calls and can get infeasible to employ at real time. The sampling of software and hardware is considered to limit the overhead on the running applications. Moreover, in order to

correlate the high level job information to the collected samples, batch system information should also be collected. Since data is collected on all individual nodes of the machine, a lightweight data transport protocol is also required. Finally, the analysis tool needs to employ data storage, which can later be used to perform offline training of the ML algorithm. The trained ML algorithm should then be able to find irregular usage and inefficiency patterns with new data being published to the system.

3 PROPOSED SYSTEM

Figure 1 presents the proposed framework. It collects software samples and operating system data using the Linux perf events interface [6], hardware events via ProPE [3] and batch system data using a Slurm plugin. All data is sent through a message broker to the data storage and the data analysis steps. Data is stored in a scalable database and the data analysis employs ML to detect performance inefficiencies in real time.

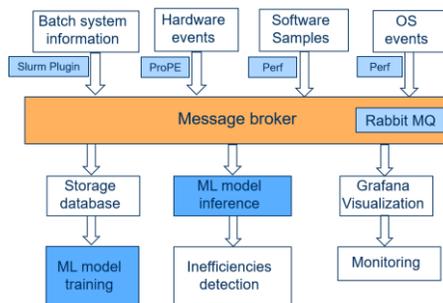


Figure 1: Proposed Inefficiencies Detection Framework

REFERENCES

- [1] Thomas Ilsche et. al, 2014, Combining Instrumentation and Sampling for Trace-based Application Performance Analysis, *In Tools for High Performance Computing*, Springer, Cham 2014.
- [2] Alessio Netti et. al, 2019, From Facility to Application Sensor Data: Modular, Continuous, and Holistic Monitoring with DCDB. *In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC '19)*. ACM Press, New York, NY
- [3] Dieter an May et. al. 2019, PROPE: Process.Oriented Performance Engineering Service Infrastructure for Scientific Software at German HPC Centers, *White Paper Draft*. Available: <https://blogs.fau.de/prope/files/2019/05/ProPE.pdf>
- [4] Denis Shaykhislamov and Vadim Voevodin. 2018. An Approach for dynamic detection of inefficient supercomputer applications, *Procedia Computer Science* 136 (2018), 35-43.
- [5] Tuncer O. et al. 2017, Diagnosing Performance Variations in HPC application using Machine Learning. *In High Performance Computing. ISC 2017. Lecture Notes in Computer Science*, Vol. 10266
- [6] Vincent M.Weaver, 2015, Self-monitoring Overhead of the Linux perf_event Performance Counter Interface, *In IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, 2015.