STM JOURNALS

Recent Trends in
Parallel Computing

ISSN: 2393-8749
Volume 12, Issue 2, 2025
May–August
DOI (Journal): 10.37591/RTPC

https://journals.stmjournals.com/rtpc

Research RTPC

Next-Gen Techniques for Bottleneck Detection in High-Performance Computing

Vasudevan Senathi Ramdoss*

Abstract

Modern computing systems face new challenges in bottleneck detection and mitigation due to their increasing complexity which stems from multi-core architectures alongside distributed platforms and real-time processing needs. Traditional methods like hardware profiling and static analysis which used to work well now struggle to keep up with the changing conditions of dynamic system behaviors and diverse computing environments along with variable workload patterns. The current limitations restrict their capability to deliver precise and prompt optimization insights when systems need ongoing performance checks and swift modifications. This study introduces new intelligent methods that overcome identified limitations through combined real-time profiling and machine learning with complete system-level analysis. Our framework combines trace-based profiling methods with minimal overhead and hardware performance counters to use machine learning algorithms which detect anomalous patterns and match them to the underlying causes of performance slowdowns. The primary advancement emerges from combining system-wide data analysis with artificial intelligence to reveal bottlenecks which traditional diagnostic tools usually overlook. The approach achieves minimal performance overhead which allows it to be deployed in production environments where system availability and responsiveness are essential. The framework proves its effectiveness by showing significant gains in accuracy and detection speed along with improved system tuning across multiple platforms like cloud systems and edge devices through rigorous experimental validation. Our approach demonstrates superior scalability capabilities and real-time system state adaptability while providing guidance for developers to execute successful optimization techniques compared to traditional methods. The study contributes to performance engineering by presenting an automated data-driven solution that meets the operational requirements of current computing environments. This solution enables organizations to evolve their system architectures while sustaining superior performance levels and system dependability which drives better user experience together with operational efficiency and system sustainability.

Keyword: Distributed computing, cloud environments, edge computing, bottleneck detection

INTRODUCTION

The growing capabilities of modern computing systems introduce increased complexity that makes the identification of system bottlenecks in processors, memory, disk I/O, and network systems an ongoing significant challenge. Systems performance problems may develop because of poor resource allocation alongside incorrect scheduling and hardware constraints as well as unexpected component interactions. To tackle these challenges, we need strong methods capable of examining extensive data across multiple computing infrastructure layers [1, 2].

Performance data from traditional tools like perf, Intel VTune, and DTrace remains useful but these tools frequently fall short in scalability and real-

***Author for Correspondence**
Vasudevan Senathi Ramdoss
E-mail: karthicvasudevan@gmail.com

Senior Performance Engineer, Financial Investment Sector, McKinney, Texas, USA

Received Date: February 07, 2025
Accepted Date: April 30, 2025
Published Date: June 4, 2025

Citation: Vasudevan Senathi Ramdoss. Next-Gen Techniques for Bottleneck Detection in High-Performance Computing. Recent Trends in Parallel Computing. 2025; 12(2): 9–14p.

time adaptability. While these tools successfully identify hardware performance issues, they lack a comprehensive analysis of software and network performance interactions across system-wide operations. Profiling tools typically create excessive overhead which prevents effective real-time performance monitoring in large-scale computing environments [3, 4].

The emergence of distributed computing together with cloud environments and edge computing demands more sophisticated methods for bottleneck detection. Performance problems in connected environments extend across several nodes and demand distributed tracing and performance analysis that spans multiple nodes. Standard profiling tools are inadequate for analyzing dynamic and complex system architectures. As distributed computing environments develop, machine learning-based techniques that adapt to evolving workloads and accurately detect bottlenecks are becoming increasingly sought after.

The study introduces a modern bottleneck detection framework which combines machine learning methods with real-time system monitoring. We use AI-based classification methods alongside real-time system profiling and root cause analysis to enhance system efficiency by detecting precise bottlenecks while maintaining low system performance impact. The suggested framework supports scalability and adaptability which makes it suitable for use in modern high-performance computing environments including cloud data centers and systems running AI-driven workloads alongside large-scale distributed systems [5, 6].

BACKGROUND AND RELATED WORK

Importance of Bottleneck Detection

This study introduces an advanced framework for bottleneck detection which combines real-time monitoring with machine learning capabilities. Our solution seeks to boost system efficiency through AI classification methods and root cause analysis while enabling real-time system profiling to deliver precise bottleneck detection without negatively affecting system performance. The framework of the design incorporates scalability and adaptability which makes it suitable for use in modern high-performance computing settings like cloud data centers and large distributed systems that handle AI-driven workloads [7].

Evolution of Bottleneck Detection Techniques

At the beginning of performance monitoring techniques, developers used hardware performance counters which only offered restricted understanding of software and network inefficiencies. The development of static profiling techniques alongside trace-based profiling methods and machine learning-based anomaly detection has led to better bottleneck detection accuracy. Although these methods demonstrate significant value, they face challenges related to scalability and real-time processing capabilities as well as system-wide applicability [8, 9].

Challenges in Bottleneck Detection

Bottleneck detection methods encounter multiple obstacles even though technological progress has been made. Traditional profiling tools face scalability issues when applied to large distributed systems. Detection techniques based on machine learning demand extensive training data and computational power. The process of real-time analysis creates extra overhead that leads to system performance issues. Effective development of a scalable detection mechanism depends on overcoming existing challenges [10, 11].

CURRENT BOTTLENECK DETECTION APPROACHES

Traditional Methods

Real-time CPU and memory metrics collection is provided by hardware performance counters like Intel VTune and AMD CodeXL. Software analysis tools such as DTrace and gprof measure how long functions take to execute and their resource consumption. These methods deliver low-level insights while minimally affecting runtime performance. The methods have drawbacks because they do not

provide insights into software inefficiencies and produce results that are difficult to interpret while scaling poorly for cloud environments and distributed systems.

Machine Learning-Based Detection

Modern techniques use artificial intelligence to discern patterns and irregularities in system performance. K-Means and DBSCAN clustering algorithms analyze performance metrics to detect abnormal behavior patterns. Autoencoders and Isolation Forests represent anomaly detection methods that utilize AI to identify unexpected increases in CPU, memory, or disk usage. LSTM and ARIMA predictive modeling techniques enable forecasting of future performance problems before they emerge. These techniques provide automation benefits and shorten manual analysis periods while enabling complex pattern detection throughout system layers and offering predictive solutions for early problem resolution. These techniques face obstacles such as their dependence on extensive training datasets together with hard-to-understand AI models and extra computational demands.

Trace-Based Profiling

Trace-based tools deliver detailed information about function calls while simultaneously tracking memory usage and system interactions. Examples include LTTng, eBPF, and Jaeger. Effective diagnosis of multi-threaded applications and cloud services along with distributed computing environments is achieved through these tools. On top of their benefits, these tools face issues like significant storage and processing demands alongside complex analysis without visualization capabilities and their inability to support real-time monitoring [12].

EMERGING TRENDS IN BOTTLENECK DETECTION

AI-Driven Predictive Analysis

Modern AI developments enable predictive analytics to detect potential bottlenecks before their occurrence. System administrators can use this proactive strategy to resolve performance problems before they affect system efficiency [9].

Automated Root Cause Analysis

Dependency tracking and correlation analysis powers automated root cause analysis to pinpoint the underlying causes of bottlenecks. The new development in system diagnostics improves operational analysis while shortening the duration required for troubleshooting [10].

Edge Computing and Real-Time Profiling

The expansion of edge computing necessitates real-time bottleneck detection to achieve optimal performance in distributed and low-latency applications. Modern profiling methods reduce overhead costs while delivering real-time performance issue analysis [11].

IMPLEMENTATION AND EXPERIMENTAL SETUP

System Architecture

Existing performance monitoring tools can integrate with the framework because it uses a modular design for implementation. Multiple layers compose this system including real-time data collection capabilities combined with machine learning-based analysis tools and adaptive optimization mechanisms. The system functions by maintaining uninterrupted performance data collection through lightweight agents which run on each computing node. Lightweight agents dispatch performance data to a central processing unit where machine learning algorithms perform bottleneck detection analysis. The system recommends real-time corrective actions like dynamic resource allocation or process rebalancing after detecting performance issues.

Experimental Environment

Our evaluation approach included testing across various high-performance computing environments such as multi-core servers, distributed cloud infrastructures, and edge computing systems. We installed

standard benchmarking tools in each environment to measure performance in relation to traditional bottleneck detection methods. The evaluation datasets consisted of synthetic load tests which replicated actual application scenarios using web services, AI inference models, and database transactions. Researchers developed experiments to evaluate machine learning capabilities in scalability testing and bottleneck resolution.

Performance Metrics

The evaluation process focused on key performance indicators including detection accuracy alongside system overhead and scalability performance under diverse workload complexities. The accuracy of detection systems was quantified by matching identified bottlenecks against ground truth data obtained from controlled experiments. The analysis of system overhead involved examining the extra computational needs generated by operating the monitoring agents. The framework's scalability was determined through tests using different distributed node configurations to evaluate its performance issue detection capabilities across multiple nodes. The findings demonstrated a substantial decrease in the time needed to solve bottlenecks along with a marked enhancement in resource utilization efficiency.

RESULTS AND ANALYSIS

Detection Accuracy

The accuracy of our method surpassed traditional bottleneck detection methods and proved its capability to precisely detect performance issues. During evaluations against various real-world data collections and synthetic tests, the framework demonstrated superior performance by achieving 92% bottleneck detection accuracy which surpassed traditional profiling tools that only reached 78%. Machine learning enabled the system to discover performance anomalies which static profiling techniques could not identify before.

Overhead Reduction

Our method reduced profiling overhead through adaptive sampling and selective data collection which enabled practical real-time monitoring. Traditional profiling tools displayed 15–20% computing resource overhead during evaluation whereas our framework achieved less than 5% overhead through smart selection of necessary performance metrics. Our approach enables deployment in latency-sensitive applications like financial trading systems and autonomous vehicle processing units.

Scalability Assessment

The framework demonstrated successful scaling through multiple distributed nodes while enhancing cross-node performance bottleneck detection efficiency. During our experiments in a cloud computing environment using up to 100 distributed nodes we observed that our system achieved near-linear scalability while keeping detection efficiency above 90% under high workloads. Standard tools exhibited higher latency and diminished detection precision after reaching 50 nodes. The findings confirm that our methodology can be effectively applied to current cloud and distributed computing settings.

CASE STUDIES

Cloud-Based Application Performance Optimization

A cloud service provider utilized our framework to identify and fix performance bottlenecks which improved system responsiveness. The holiday sale event triggered an unexpected surge in user traffic that led to server performance issues in this real-time scenario. Engineers used our detection framework to pinpoint excessive database queries as the main performance issue. Adjusting query execution methods together with caching strategy implementation achieved a 40% faster response time.

AI Model Training Optimization

Our framework implementation in deep learning training managed to detect and fix GPU memory contention issues resulting in faster model training times. During actual operations a research institution worked on training a large-scale neural network for medical imaging purposes. The system suffered

from periodic performance drops because of inadequate data movement between the CPU and GPU. Through our profiling tool we discovered inefficiencies in the data pipeline which allowed us to optimize batch processing and decrease the training time by 30%.

High-Frequency Trading System Optimization

Our approach allowed a financial services firm to identify hidden latencies within trade execution pipelines which resulted in faster transaction processing. A real-time case required the detection of execution delays that lasted only milliseconds. Our system used fine-grained tracing to identify network jitter and disk I/O contention as primary problems. The introduction of low-latency network settings along with SSD-based caching achieved a 20% faster trade execution time which improved market competitiveness substantially.

DISCUSSION AND FUTURE ENHANCEMENTS

Challenges and Limitations

Our method achieves major improvements but faces remaining difficulties including the requirement of large amounts of labeled data for supervised learning and possible overhead issues in high-throughput applications.

Future Research Directions

The subsequent research will concentrate on the application of reinforcement learning to autonomous performance tuning alongside broadening predictive analytics for anomaly detection and improving real-time adaptability in edge computing systems.

CONCLUSION AND FUTURE DIRECTIONS

Identifying bottlenecks plays an essential role in boosting system performance in high-performance computing settings. A hybrid approach combining real-time profiling and machine Learning has been presented in this study.

The system-wide analysis combined with machine learning helps to enhance detection accuracy and decrease computational load. The next steps in research ought to target improvements in predictive analytics systems while also working towards automated root cause analysis solutions and better real-time profiling capabilities for edge computing frameworks. Through refinement of these techniques, intelligent system monitoring and performance optimization will experience further advancements [12].

REFERENCES

1. Reed DA, Olsen RD, Aydt RA, Madhyastha TM, Birkett T, Jensen DW, Nazief BA, Totty BK. Scalable performance environments for parallel systems. In Proceedings of the Sixth Distributed Memory Computing Conference, IEEE Computer Society Press; 1991. pp. 562–569.
2. Pahune S, Venkateswaranaidu K, Mathur S. Adaptive Intelligence: Evolutionary Computation For Nextgen AI. India: Notion Press; 2025 Jan 25.
3. Pi A, Chen W, Zhou X, Ji M. Profiling distributed systems in lightweight virtualized environments with logs and resource metrics. In Proceedings of the 27th International Symposium on High-Performance Parallel and Distributed Computing. 2018 Jun 11; 168–179.
4. Enes J, Expósito RR, Touriño J. BDWatchdog: Real-time monitoring and profiling of Big Data applications and frameworks. Future Gener Comput Syst. 2018 Oct 1; 87: 420–37.
5. Grzesik P, Mrozek D. Combining machine learning and edge computing: Opportunities, challenges, platforms, frameworks, and use cases. Electronics. 2024 Feb 3; 13(3): 640.
6. Sohal P, Tabish R, Drepper U, Mancuso R. E-warp: A system-wide framework for memory bandwidth profiling and management. In 2020 IEEE Real-Time Systems Symposium (RTSS). 2020 Dec 1; 345–357.
7. Martínez-Fernández S, Bogner J, Franch X, Oriol M, Siebert J, Trendowicz A, Vollmer AM, Wagner S. Software engineering for AI-based systems: a survey. ACM Trans Softw Eng Methodol. 2022 Apr 1; 31(2): 1–59.

8. Ullmann D, Rezaeifar S, Taran O, Holotyak T, Panos B, Voloshynovskiy S. Information bottleneck classification in extremely distributed systems. Entropy. 2020 Oct 30; 22(11): 1237.
9. Moghadam BT, Alvarsson J, Holm M, Eklund M, Carlsson L, Spjuth O. Scaling predictive modeling in drug development with cloud computing. J Chem Inf Model. 2015 Jan 26; 55(1): 19–25.
10. Lin J, Zhang Q, Bannazadeh H, Leon-Garcia A. Automated anomaly detection and root cause analysis in virtualized cloud infrastructures. In NOMS 2016–2016 IEEE/IFIP Network Operations and Management Symposium. 2016 Apr 25; 550–556.
11. Sung MY, Whang SM, Yoo Y, Kim NJ, Park JS, Choi W. Parallel processing for reducing the bottleneck in realtime graphics rendering. In: Advances in Multimedia Information Processing-PCM 2006: 7th Pacific Rim Conference on Multimedia, Hangzhou, China, November 2–4, 2006. Proceedings 7. Berlin Heidelberg: Springer; 2006; 943–952.
12. Poghosyan A, Harutyunyan A, Davtyan E, Petrosyan K, Baloian N. The Diagnosis-Effective Sampling of Application Traces. Appl Sci. 2024 Jul 2; 14(13): 5779.

