Resource-Aware Task Scheduling in Heterogeneous Computing Platforms for Industrial Analytics

Noah van Dijk

Netherlands

Abstract

This study proposes a resource-aware task scheduling framework for heterogeneous computing platforms supporting industrial analytics, focusing on improving computational efficiency, workload distribution, and resource utilization. The proposed methodology integrates CPU, GPU, and edge computing resources within a unified scheduling architecture to accommodate diverse industrial data processing requirements. A dynamic task allocation model considers computational complexity, memory availability, processing deadlines, communication latency, and energy consumption to determine suitable execution resources. Machine learning-based workload prediction is incorporated to estimate task execution times and anticipate resource demands under changing operating conditions. A multi-objective optimization strategy balances processing throughput, task completion time, energy efficiency, and system reliability while minimizing scheduling overhead. The framework evaluates alternative scheduling policies through simulation scenarios involving real-time sensor analytics, predictive maintenance, and industrial process monitoring. Performance assessment considers scheduling latency, resource utilization, deadline satisfaction, and computational scalability. The proposed approach supports efficient industrial analytics across distributed heterogeneous computing infrastructures.

Keywords: Resource-Aware Scheduling; Heterogeneous Computing; Industrial Analytics; Workload Optimization; Edge Computing.

1. Introduction

Enterprise applications depend on reliable diagnostics to detect runtime errors, failed workflows, API issues, slow transactions, and integration faults [1]. Logs are the main evidence source for understanding what happened before, during, and after an incident. Effective application diagnostics require structured logging, traceability, contextual metadata, error classification, and operational visibility [2]. However, fixed logging strategies often create two problems. Low-detail logs may miss important causes, while high-detail logs may generate excessive storage cost and alert noise [3].

The main challenge is that diagnostic needs change during runtime. A normal transaction may require only basic logs, but a failing workflow, repeated exception, or security-sensitive event may need deeper traces [4]. Adaptive logging improves this process by changing log depth according to system condition, transaction risk, and error behavior [5]. AI-assisted monitoring can identify when additional diagnostic detail is needed and when logging can return to normal levels. This article proposes an adaptive logging framework for enterprise application diagnostics.

2. Methodology

The proposed framework captures log metadata from application servers, API gateways, databases, background jobs, authentication modules, and integration services [6]. It records event type, severity, timestamp, user role, transaction ID, request path, error frequency, response time, affected component, and workflow state. These features are converted into diagnostic context profiles. Adaptive logging requires context because the same event may be low-risk in one workflow but critical in another [7, 8].

The logging controller classifies runtime states as normal, warning, error-prone, performance-risk, security-sensitive, workflow-critical, or incident-active [9]. Based on this classification, it adjusts log level, trace depth, parameter capture, correlation ID tracking, and diagnostic enrichment. Lightweight scoring and policy rules are used so logging decisions remain fast and explainable. Sensitive fields are masked before storage [10], and transaction-critical events are linked with workflow and database identifiers for easier diagnosis.

The framework is evaluated using simulated enterprise applications involving APIs, database workflows, authentication events, scheduled jobs, and service integrations. Static logging is compared with adaptive logging under repeated exceptions, slow requests, failed jobs, API timeout, suspicious login, and workflow interruption scenarios [11]. Evaluation metrics include diagnosis time, log volume reduction, root-cause evidence quality, missed-event rate, storage overhead, alert usefulness, and administrator acceptance rate. Feedback from resolved incidents, ignored logs, and administrator notes is used to refine logging policies over time.

3. Results and Discussion

The results show that adaptive logging improves diagnostics by capturing richer evidence only when runtime risk increases. In the baseline condition, static logs either lack enough detail or produce large volumes of low-value records. With the proposed framework, detailed traces are activated for high-risk events and reduced during normal execution. The strongest improvement appears in intermittent failures, where deeper logs are needed only around the failure window.

The discussion shows that adaptive logging must balance diagnostic depth with privacy and storage cost. Excessive logging can expose sensitive data or overload monitoring systems, while weak logging may hide failure causes. The main limitation is that log-level decisions depend on accurate runtime classification. Regular policy review and masking controls are necessary for safe long-term use.

4. Conclusion

This article presented an adaptive logging framework for enterprise application diagnostics. The framework adjusts log detail using runtime context, risk scoring, workflow state, and error behavior. It improves troubleshooting by increasing diagnostic evidence during abnormal conditions while reducing unnecessary log noise during normal operation. The study shows that adaptive logging can strengthen enterprise diagnostics when combined with structured metadata, privacy controls, and continuous incident feedback.

References

1. Sulaiman, M., Halim, Z., Lebbah, M., Waqas, M., & Tu, S. (2021). An Evolutionary Computing-Based Efficient Hybrid Task Scheduling Approach for Heterogeneous Computing Environment: An evolutionary computing-based efficient hybrid task scheduling approach for heterogeneous computing environment. *Journal of Grid Computing*, *19*(1), 11.
2. Deng, Z., Yan, Z., Huang, H., & Shen, H. (2020). Energy-aware task scheduling on heterogeneous computing systems with time constraint. *IEEE Access*, *8*, 23936-23950.
3. Gondi, S. R. (2023). AI-Assisted Fault Localization in Semiconductor Wafer Testing and Manufacturing Operations. Perception, Navigation and Intelligent Devices, 10, 1-8.
4. Stan, R. G., Băjenaru, L., Negru, C., & Pop, F. (2021). Evaluation of task scheduling algorithms in heterogeneous computing environments. *Sensors*, *21*(17), 5906.
5. Matrouk, K., & Alatoun, K. (2021). Scheduling algorithms in fog computing: A survey. *International Journal of Networked and Distributed Computing*, *9*(1), 59-74.
6. Kaur, N., Kumar, A., & Kumar, R. (2021). A systematic review on task scheduling in Fog computing: Taxonomy, tools, challenges, and future directions. *Concurrency and Computation: Practice and Experience*, *33*(21), e6432.
7. Jamil, B., Ijaz, H., Shojafar, M., Munir, K., & Buyya, R. (2022). Resource allocation and task scheduling in fog computing and internet of everything environments: A taxonomy, review, and future directions. *ACM Computing Surveys (CSUR)*, *54*(11s), 1-38.
8. rao Gondi, S. (2023). Data-Driven Process Analytics for Early Defect Detection in High-Volume Semiconductor Fabrication. Cross-Disciplinary Knowledge Systems, 10, 37-42.

9. Bansal, S., Aggarwal, H., & Aggarwal, M. (2022). A systematic review of task scheduling approaches in fog computing. *Transactions on Emerging Telecommunications Technologies*, *33*(9), e4523.
10. Bisht, J., & Vampugani, V. S. (2022). Load and cost-aware min-min workflow scheduling algorithm for heterogeneous resources in fog, cloud, and edge scenarios. *International Journal of Cloud Applications and Computing (IJCAC)*, *12*(1), 1-20.
11. Hosseinzadeh, M., Azhir, E., Lansky, J., Mildeova, S., Ahmed, O. H., Malik, M. H., & Khan, F. (2023). Task scheduling mechanisms for fog computing: a systematic survey. *IEEE Access*, *11*, 50994-51017.