Distributed Computing Architecture for High-Volume Semiconductor Test Data Processing

Sarah Mitchell

Australia

Abstract

This study proposes a distributed computing architecture for high-volume semiconductor test data processing, focusing on improving computational scalability, processing efficiency, and data management reliability. The proposed framework integrates distributed data ingestion, parallel processing, and scalable storage mechanisms to manage large datasets generated during wafer probing, electrical characterization, and final device testing. A microservices-based architecture employs distributed message queues and partitioned data processing to support continuous acquisition and analysis of semiconductor test records. Parallel computing techniques are incorporated to accelerate data preprocessing, statistical analysis, anomaly detection, and yield estimation across multiple computing nodes. Dynamic workload balancing and fault-tolerant processing mechanisms are introduced to maintain system availability under fluctuating data volumes and computational demands. Performance evaluation considers data throughput, processing latency, resource utilization, fault recovery time, and horizontal scalability. The proposed architecture supports efficient semiconductor test analytics, timely quality assessment, and reliable data-driven decision-making in high-volume electronic manufacturing environments.

Keywords: Distributed Computing; Semiconductor Test Data; Parallel Data Processing; Scalable Architecture; Manufacturing Analytics.

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, 7]. 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 [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. rao Gondi, S. (2025). Multi-Modal AI-Driven Electrical and Optical Characterization Framework for Sub-5 nm Semiconductor Defect Localization. Research Briefs on Information and Communication Technology Evolution, 11, 188-198.
2. Tsung, C. K., Hsieh, H. Y., & Yang, C. T. (2019). An implementation of scalable high throughput data platform for logging semiconductor testing results. *IEEE Access*, *7*, 26497-26506.
3. Misaki, M., Tsuda, T., Inoue, S., Sato, S., Kayahara, A., & Imai, S. I. (2016, December). Distributed database and application architecture for big data solutions. In *2016 International Symposium on Semiconductor Manufacturing (ISSM)* (pp. 1-4). IEEE.
4. Tsai, C. P., Hsiao, H. C., Chao, Y. C., Hsu, M., & Chang, A. R. (2018, December). Bridging the gap between big data system software stack and applications: The case of semiconductor wafer fabrication foundries. In *2018 IEEE International Conference on Big Data (Big Data)* (pp. 1865-1874). IEEE.
5. rao Gondi, S. (2020). Intelligent Fault Detection and Classification in Semiconductor Manufacturing Using Statistical Learning Techniques. Transactions on Smart Electrical and Computational Systems, 7, 26-31.
6. Suryajaya, B., Chen, C. C., Hung, M. H., Liu, Y. Y., Liu, J. X., & Lin, Y. C. (2017, August). A fast large-size production data transformation scheme for supporting smart manufacturing in semiconductor industry. In *2017 13th IEEE Conference on Automation Science and Engineering (CASE)* (pp. 275-281). IEEE.

7. Kim, J., & Joe, I. (2024). Chip-level defect analysis with virtual bad wafers based on huge big data handling for semiconductor production. *Electronics*, *13*(11), 2205.
8. Chen, D., Zhang, J., Wu, L., Zhang, P., & Wang, M. (2024). Industrial data space application framework for semiconductor wafer manufacturing system scheduling. *Journal of Manufacturing Systems*, *77*, 464-482.
9. Webb, J. W., & Webb, J. (2016, September). A method for storing semiconductor test data to simplify data analysis. In *2016 IEEE AUTOTESTCON* (pp. 1-10). IEEE.
10. Gondi, S. R. (2024). Data-Centric Yield Engineering Frameworks for Next-Generation Semiconductor Production Systems. Transactions on Smart Electrical and Computational Systems, 11, 37-44.
11. Chen, T. C. T. (2022). *Production planning and control in semiconductor manufacturing: Big data analytics and industry 4.0 applications*. Springer Nature.