International Journal of Multidisciplinary Research and Growth Evaluation
ISSN: 2582-7138
Received: 23-04-2021; Accepted: 20-05-2021
www.allmultidisciplinaryjournal.com
Volume 2; Issue 3; May-June 2021; Page No. 571-574

Metrics-The Second Pillar of Observability

Lakshmi Narasimha Rohith Samudrala
Independent Researcher, USA

Corresponding Author: **Lakshmi Narasimha Rohith Samudrala**

DOI: https://doi.org/10.54660/.IJMRGE.2021.2.3.571-574

Abstract
Logs, Metrics, and Traces form the three pillars of observability. Each equally important as the other. Metrics, are the second pillar of observability. They providing structured numerical insights into a systems overall health. There are multiple types of metrics, each type helps organizations track either infrastructure health, application performance, business KPIs, or user experience.

This paper explores the multiple challenges that come capturing and managing metrics, such as high cardinality, excessive alert noise, missing or delayed data, and scalability. The paper provides mitigation strategies to overcome these challenges. Further, the paper provides some thoughts on upcoming future trends in metrics.

Keywords: Metrics, Observability, System Performance, Infrastructure Monitoring, Application Performance, Business KPIs, High Cardinality, Anomaly Detection, Predictive Analytics, AI-driven Monitoring, Cloud-Native Metrics, Distributed Systems, Real-Time Monitoring, Scalability, Adaptive Baselining, Self-Healing Systems, Automated Incident Response

1. Introduction
The three fundamental pieces of observability are logs, metrics, and traces [1, 2]. Each component provides valuable insights helping IT teams understand the health of the systems. Logs provide event-level information, traces provide request flow details, and metrics provide quantifiable indicators for the system help and performance [1].

A. What are metrics?
Metrics are numerical measurements that provide insights into different aspects of the IT landscape [2]. Metrics are collected at regular intervals from multiple sources. They are typically stored in time-series databases and visualized in real-time dashboards for trend analysis and anomaly detection.

B. Why are metrics important?
Metrics play a critical role in ensuring system reliability and efficiency. By continuously tracking performance indicators, teams can proactively detect and resolve issues before they impact users. Metrics help answer key questions regarding infrastructure (e.g., CPU utilization, memory consumption), application performance (e.g., latency, request success rate), application reliability (e.g., error rate, system availability), user experience (e.g., Apdex score, transaction times) etc. [4]
By leveraging metrics, organizations can enhance observability, support real-time decision-making, and enable automated incident response. Moreover, correlating metrics with logs and traces allows for deeper root cause analysis, making troubleshooting faster and more efficient.

2. Types of metrics and their use cases
Metrics provide quantifiable data to help organizations understand the health, performance, and into system health, application performance, and reliability of their systems. There are different types of metrics which are gathered from various sources. Each type of metric serves a different purpose, ranging from monitoring infrastructure resources to analyzing user experience and business performance.

A. Infrastructure Metrics
Infrastructure metrics provide insights into the health and performance of infrastructure on which the applications are running. Be it physical and virtual resources such as servers, databases, containers, and network devices. These metrics are essential to

understand the stability of the system, detecting resource bottlenecks, and planning capacity scaling.

Below figure 1 showcases an example of infrastructure metric

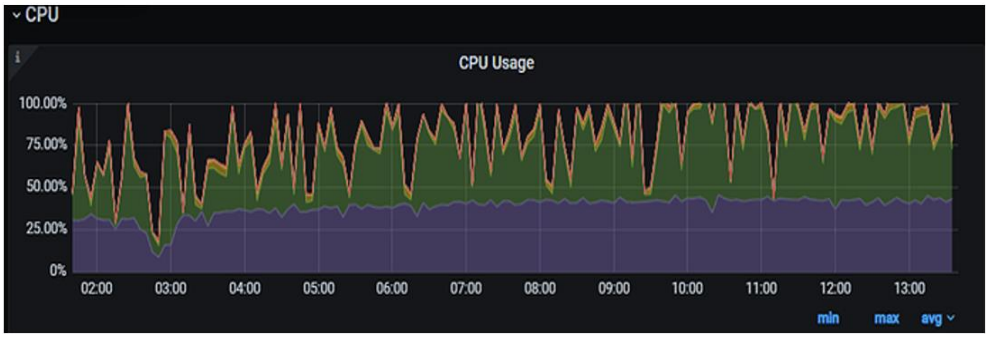

Fig 1: Example of CPU Usage metric

B. Application performance metrics

Application metrics talk about the actual software that is running on the infrastructure. These metrics help IT teams understand how well an application is performing, tracking key aspects such as response time, error rates, and request throughput [5]. These metrics help teams detect slow transactions, failures, and performance bottlenecks before they impact users. Figure 2 shows an example of Service failure rate metric.

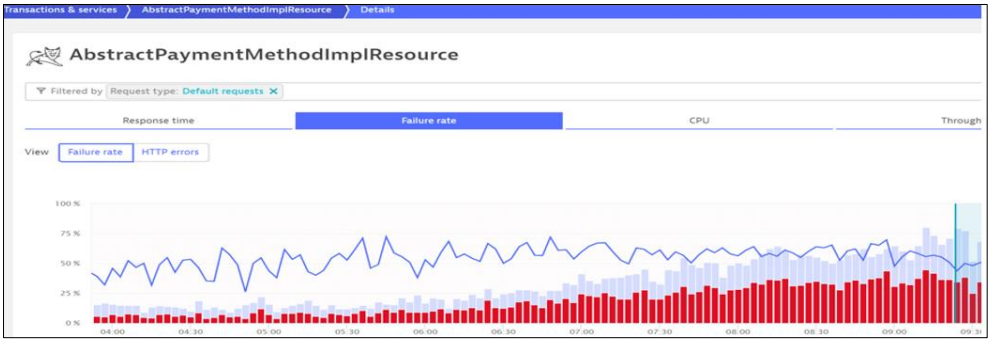

Fig 2: Example service failure rate metric [5]

C. Business and user experience metrics

Business metrics are extremely important. These metrics track key performance indicators (KPIs) that reflect user experience, revenue generation, and customer engagement. These metrics help align IT operations with business goals and customer expectations [3]. Figure 3 shows an example of apdex rating for a web application.

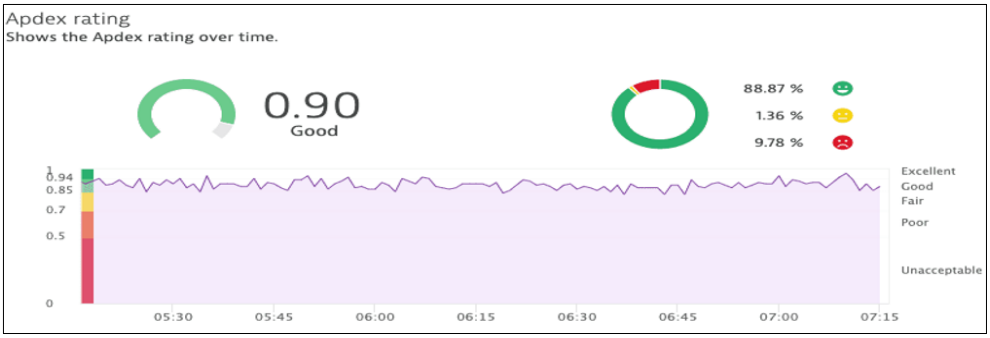

Fig 3: Example Apdex rating for a web application [4]

D. Custom metrics and domain-specific monitoring

In addition to standard infrastructure and application metrics, there would be a need for custom metrics that are specific to organizations or applications. These custom metrics tailored to either specific industry or unique business needs [3]. These can include specialized indicators for finance, healthcare, utilities, and manufacturing.

3. Challenges in metric collection and analysis

Being a critical component of observability, metrics are very useful for organizations to understand their system performance, application health, and business operations. However, metric collection, processing, storage, and analysis pose multiple challenges that organizations would need to overcome. This section explores some such challenges.

A. High cardinality and storage costs
A metric can have multiple unique values, for example session duration with a dimension of session id or IP address. Such metrics can make querying, indexing, and storage of the metrics very resource-intensive. This is a common problem in the environments with dynamic workloads, multi-tenant cloud systems, and microservices architectures.

B. Noise and false positives in alerts
Excess data poses another difficult challenge, that is alert fatigue. Complex IT environments gather a lot of metrics. If the excess data is not properly handles, it can lead to noisy alerts that are not actionable. Such noisy alerts with lead to IT teams getting alert fatigued and not pay attention to important signals. Such poorly configured metric thresholds often result in false positives, increasing investigation time and operational inefficiencies.

C. Delayed or missing data issues
Metrics should be collected, stored, and visualized in real-time, but in some cases, there could be delays in collecting data or missing complete data. This can cause issues to IT team as they would not get alerts when there is an actual problem.

D. Scaling metrics in distributed systems
With the exponentially fast growth of modern IT landscape many applications now rely on Kubernetes containers, cloud environments, and microservices. In these architectures each component generates its own set of metrics. As the number of monitored services grow, scaling metric collection without introducing overhead becomes challenging.

4. Mitigation Strategies

A. High cardinality and storage costs
To handle high cardinality and avoid excessive storage issues, organizations should leverage metric labeling. This allows organizations to intelligently aggregate the data instead of collecting granular metrics. The grouping can be done based on meaningful metadata such as geographical region, service tier, or application instance. Histogram and summary metrics can be used instead of tracking every individual event, as they provide useful statistical data (e.g., percentiles) without overwhelming storage systems.

B. Reducing noise and false positives in alerts
Static alerts are primarily the culprits for alert fatigue. Organizations can replace static thresholds with dynamic baselining. AI-powered monitoring tools like Dynatrace, AppDynamics, New Relic, etc. can learn normal behavior for a metric and identify anomalies to the normal behavior. This way the team can be sure that when an alert is generated it is actionable as the AI would only trigger an alert if the metric exceeds its expected variation. Also making the metric context aware is very helpful, as this way multiple alerts can be grouped into one if contextually, they are related to the same issue.

C. Addressing delayed or missing metrics
To mitigate the impact of delayed or missing metrics, organizations should implement highly available data collection mechanisms to ensure reliability. Deploying multiple metric collectors across different regions prevents single point of failure. Additionally, using buffering and caching mechanisms ensures that metric data is temporarily stored and transmitted once the network or storage system recovers. It is also advisable to configure health checks or heart bear monitoring, which periodically checks whether data sources and data collectors are still active.

D. Scaling metric collection in distributed systems
In the modern IT landscape, the metrics are generated via various sources. To handle such a massive scale of metric collection, organizations should implement hierarchy-based collection and aggregation techniques. Instead of sending raw metrics from every container or microservice, the data should be aggregated at the node, cluster level, or agent before being sent to the central monitoring servers. These collection points for data can compress, encrypt, and transfer the data hence reducing the overall load.

5. Future of metrics in observability

As IT environments become more complex and data intensive, traditional methods of metric collection and analysis would also need to evolve. The increasing usage of AI and ML is shaping to be the future of observability. With the rapid growth of cloud-native architectures and the need for real-time decision-making, organizations must adopt monitoring advancements to stay ahead of the curve and avoid user impacts.

A. AI-driven adaptive baselining and anomaly detection
Traditional static threshold-based monitoring is being replaced with AI-powered adaptive baselining. Adaptive baselining automatically learns the shift in application behavior based on historic trends, seasonality and adjusts its baseline accordingly. This allows AI to detect subtle deviations that may indicate early signs of failure. This approach reduces false positives and ensures that only meaningful anomalies trigger alerts, improving incident response efficiency.
Machine learning models are also being used to correlate multiple metrics, allowing organizations to pinpoint root causes faster by analyzing dependencies between performance indicators.

B. Predictive analytics for proactive monitoring
Most of the monitoring that leverages metrics are reactive as they trigger alerts when a particular threshold or condition is breached. With the advancements in machine learning, the historical data can be used to predict data points for a given metric. This means, organizations would be able to predict incidents before they actually occur. This ability would prove to be a game changer for organizations as they would now have the power to proactively remediate issues before occurring, therefore provide superior service to their customers.

C. Automated incident response and self-healing systems
The ultimate goal of any monitoring is to not only detect but also to resolve the issue automatically. Future observability platforms will integrate metrics with automated remediation workflows, allowing systems to self-correct in response to anomalies.

6. Conclusion

Metrics are one of the key cornerstones of observability. They provide quantifiable near real-time insights to system

health, application performance, and business operations. By leveraging infrastructure metrics (CPU, memory, network), application performance metrics (response time, error rates), and business KPIs (transactions per second, user engagement), teams can make data-driven decisions to improve system reliability and operational efficiency.

However, collecting and managing metrics pose multiple challenges such as high cardinality, excessive alert noise, missing data, and storage overhead. Organizations must adopt best practices such as AI-powered alerting, dynamic baselining, redundant data collection, and hierarchical metric aggregation to ensure efficient and scalable monitoring.

Looking ahead, AI and ML play a very prominent role in monitoring and observability. AI-driven monitoring, predictive analytics, and automated incident response will shape the future of metric collection and analysis. By adapting to these advancements, organizations can move towards a proactive and self-healing infrastructure. That will enable reducing downtime, improved system reliability, and user satisfaction.

7. References

1. Samuel. The 3 pillars of system observability: logs, metrics, and tracing. IOD - The Content Engineers. 2020 Dec 15 [cited 2025 Mar 27]. Available from: https://iamondemand.com/blog/the-3-pillars-of-system-observability-logs-metrics-and-tracing/
2. SentinelOne. Three pillars of observability: Do you have all of them? 2019 Jun 18 [cited 2025 Mar 27]. Available from: https://www.sentinelone.com/blog/three-pillars-of-observability/
3. Analytix Accounting. Importance of measurable business metrics | Analytix Accounting. 2018 Jul 17 [cited 2025 Mar 27]. Available from: https://www.analytixaccounting.com/importance-of-measurable-business-metrics/
4. Enzenhofer K. Easily measure and manage user experience with more flexible Apdex calculations. Dynatrace News. 2019 Dec 6 [cited 2025 Mar 27]. Available from: https://www.dynatrace.com/news/blog/easily-measure-and-manage-user-experience-with-more-flexible-apdex-calculations/
5. Duchateau F. How to fine tune failure detection. Dynatrace News. 2019 Jul 2 [cited 2025 Mar 27]. Available from: https://www.dynatrace.com/news/blog/how-to-fine-tune-failure-detection/.