

ABSTRAK

Perusahaan XYZ dalam perkembangannya beralih dari arsitektur monolitik menjadi arsitektur mikroservis. Permasalahan yang ditemui sebelumnya terkait ketika terjadi lonjakan data secara signifikan dan ketika salah satu servis mengalami kondisi mati yang dapat menyebabkan kurangnya efektivitas dari layanan tersebut dan juga diperlukan sistem yang dapat mengatasi ketika terjadinya lonjakan data yang cukup signifikan. Peralihan arsitektur dari monolitik menjadi mikroservis diharapkan dapat mengatasi permasalahan tersebut. Tetapi dengan penerapannya arsitektur mikroservis diperlukan suatu sistem untuk melakukan *monitoring* terhadap kualitas dari masing – masing servis, sehingga nantinya kualitas dari masing – masing servis dapat terlihat performanya.

Penggunaan metode *Distributed Tracing*, terutama melalui alat bantu seperti *OpenTelemetry*, menjadi krusial untuk mempermudah pengembangan dengan fitur debugging dan fixing, sekaligus memberikan informasi detail tentang proses, kinerja, dan kesalahan pada suatu sistem. Dengan menggunakan metode *distributed tracing* diharapkan dapat menyediakan informasi secara *end-to-end* terhadap proses dari sinkronisasi. Metode ini.

Implementasi sistem monitoring mikroservis di perusahaan XYZ melibatkan langkah-langkah instrumentasi yang berhasil menciptakan sistem monitoring yang efektif. Penerapan metode *Application Performance Monitoring* (APM) telah memberikan kontribusi positif dengan memilah dua jenis telemetri data, yaitu metrics dan trace, untuk memberikan gambaran detail mengenai performa layanan *microservices*. Namun, perlu dicatat bahwa penerapan *OpenTelemetry*, meskipun memberikan keuntungan dalam pemantauan performa, juga menyebabkan dampak negatif pada penggunaan memori aplikasi mikroservis. Terdapat peningkatan yang cukup signifikan sebesar 20.68% dan 40.61 % pada penggunaan memori mikroservis A dan B setelah dilakukan instrumentasi. Hasil ini menunjukkan bahwa walaupun monitoring mikroservis dapat memberikan wawasan yang bernilai, perlu dilakukan evaluasi lebih lanjut terkait dampak penggunaan alat monitoring tertentu pada sumber daya sistem mikroservis.

Kata kunci : Mikroservis, Distributed Tracing, Monitoring, Telco, Grafana, Prometheus, Jaeger, OpenTelemetry, Kualitas Servis, Metrics, Trace.

ABSTRACT

XYZ Company, in its development, has transitioned from a monolithic architecture to a microservices architecture. Previously encountered issues were related to significant data spikes and service downtime, leading to a lack of effectiveness in those services. A system was needed to address significant data spikes effectively. The transition from a monolithic to a microservices architecture is expected to alleviate these issues. However, with the implementation of a microservices architecture, a system for monitoring the quality of each service is essential. This ensures that the performance of each service can be observed.

The use of Distributed Tracing, especially through tools like OpenTelemetry, is crucial to facilitate development with debugging and fixing features. It also provides detailed information about the processes, performance, and errors within a system. Utilizing distributed tracing is expected to provide end-to-end information on the synchronization process. This method is crucial for gaining insights into the performance and interactions of each service within the microservices architecture.

The implementation of microservices monitoring systems at XYZ involves instrumentation steps that have successfully created an effective monitoring system. The adoption of Application Performance Monitoring (APM) methods has positively contributed by categorizing two types of telemetry data, namely metrics and trace, to provide detailed insights into the performance of microservices. However, it is noteworthy that the implementation of OpenTelemetry, while advantageous for performance monitoring, also has a negative impact on the memory usage of microservices applications. There is a significant increase of 20.68% and 40.61% in the memory usage of microservices A and B, respectively, after instrumentation. These results indicate that although microservices monitoring can provide valuable insights, further evaluation is needed regarding the impact of using specific monitoring tools on the resources of microservices systems.

Keywords: Microservices, Distributed Tracing, Monitoring, Telco, Grafana, Prometheus, Jaeger, OpenTelemetry, Service Quality, Metrics, Trace.