

Is Your Application Performance Monitoring Tool Cutting Corners?

Trace sampling can lead to revenue loss and hurt an organization's brand.

The transition to public cloud and cloud-native applications unlocked new capabilities that bring many advantages to organizations — easier scaling, faster compute capacity adjustment and reduced need to operate datacenters, to name a few — but it also introduced a new set of challenges. Application performance monitoring (APM) can help address these challenges, but only if done correctly. Most APM solutions use sampling to detect slow performance and errors, but the simple truth is that unless a solution ingests and analyzes all of the data, it will inevitably fail to provide insights on the erroneous behavior. Only APM solutions that ingest ALL transaction data can help organizations elevate user experience, uphold brand image and accelerate time to market.

In the past, applications tended to be designed as what is known in the industry as monoliths — a single application, running on a single server or virtual machine, written in a single programming language and updated about once per year. Such a deployment model made sense when hardware refresh cycles were long and acquisition of new hardware took several weeks or months.

Cloud technologies gave rise to a new application architecture known as microservices. A microservices-based application consists of multiple separate components, each performing a small, well-defined task, running on its own container or serverless function, all of which work together to perform the functions for which the application was designed.

The main advantage of microservices is that they can be managed, scaled and developed independently. However, every user transaction, also known as a “trace,” traverses over multiple (sometimes hundreds) of microservices, which makes understanding how the microservices interact with each other very challenging. But that understanding is essential in order to provide an exceptional user experience. A misbehaving microservice can lead to significant delays in user transactions or cause outright failures. Failed or delayed transactions can lead to loss of revenue and hurt an organization's brand.

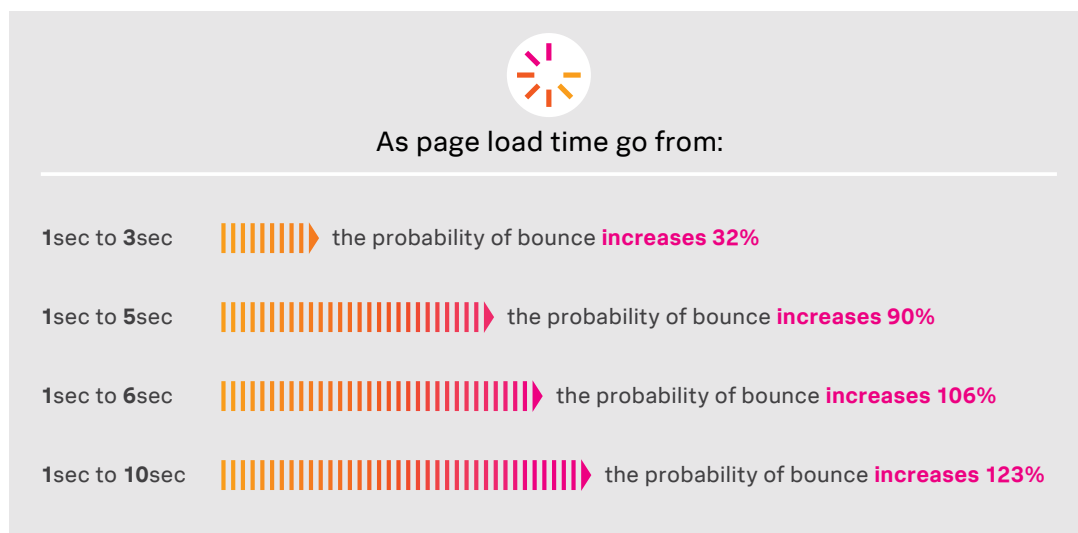


Image 1: Persistently poor performance can have significant business implications¹

The market is full of APM tools that are designed to ensure applications provide the right level of service without interruption. There are several approaches in the market to handle tracing as can be seen in the table below:

	Random/Head-based Sampling	Smart/Tail-based sampling	No Sampling/Complete trace ingestion
Description	The APM tool randomly saves traces at the time of their initiation to estimate the performance of the system	The APM tool analyzes all completed traces, and saves the ones deemed “interesting” based on the APM vendor’s definitions or user configuration. Saved traces are later used to estimate the performance of the system	All traces are saved on the APM SaaS environment and are used to provide a complete picture of the system performance
Completeness of data needed for troubleshooting issues	Very partial	Partial	Complete — all data is available
Compute Resources Used in Customer Environment	Low	High	Low
Customer Configuration and Maintenance Effort	Low	High	Low
Likelihood of poor user experience	High	High	Low

One point that clearly comes across from Table 1, is that in order to have complete data you must leverage an architecture that does not use sampling. What does incomplete data mean from a business perspective?

Sampled traces mean that user experience problems persist, because the data needed to solve them has not been sampled. Engineering and Ops teams can only wait until the sampling mechanism picks up the problematic traces in a process that can take days or even weeks, with significant negative impact on brand and digital business. Industry benchmarks show that poor user experience caused by slow load times more than doubles the probability of customer churn (see Image 1).

“This is finally the APM for the modern microservices age I have been looking for.”²

Splunk designed Splunk APM especially for modern environments, introducing a unique architecture called NoSample™ that analyzes and stores ALL traces. By analyzing and storing all traces our customers were able to reduce troubleshooting times from days to minutes, enhance the performance of their applications, maintain their brand image and maximize their online revenue, and break free of the bottlenecks imposed by legacy APM solutions.

1 <https://www.thinkwithgoogle.com/marketing-resources/data-measurement/mobile-page-speed-new-industry-benchmarks/>

2 https://www.gartner.com/reviews/market/application-performance-monitoring/vendor/splunk/product/signalfx_microservices_apm/review/view/1307851