The importance of application metrics in monitoring Cloud services.

24 November 2023 - 10 min. read

Today, more than ever, offering services via web or mobile applications is an increasingly central part of every project.

Customers expect to enjoy the services made available to them in a fluid and uninterrupted way, regardless of the time of day or the traffic the service manages.

Maintaining a good quality user experience has become an essential element within a highly competitive market, where any interruption or slowdown in the provision of services can have serious consequences, such as the loss of customers, sales opportunities, or the worsening of the company's reputation.

User experience is not just about the ease of use of an application or website but also about the speed of response and perceived reliability. In this context, monitoring systems and applications in direct contact with the public is critical to allow companies to react promptly to adverse events that could compromise the availability and usability of services.

Various monitoring solutions promise to provide a complete picture of the operation of the systems in a simple way and without requiring particular interventions.

However, using a generic solution is only partially preferable to the total absence of monitoring. Approximate and generic monitoring can give the false security of having everything under control when, in reality, problems are happening but are visible to users and not detected by the monitoring system. Using "default" metrics, or more generally not suitable for monitoring the specific case, provides a partial and distorted view of reality, quickly leading to missed reports.

This type of situation, in which customers experience slowdowns or errors when using services, is detrimental to retention and company reputation; they become even more

severe if the organization is unaware of the situation and consequently cannot manage it promptly.

Fortunately, we can customize most monitoring by introducing custom metrics and KPIs that indicate the application's health. This type of configuration is generally time-consuming and requires careful analysis of user flows and application events to determine which metrics genuinely represent the state of the service provided.

The ideal monitoring system collects and correlates both application and infrastructure metrics to have a complete and truthful picture of how the application is functioning. The importance of application metrics must be considered when you want to have a complete and accurate view of the health of a workload.

While extracting infrastructure metrics is usually assisted by the Cloud provider or the provider where the application is hosted, there is nothing standard regarding application metrics.

In this article, we will talk about the importance of monitoring metrics that reflect the accurate state of an application or system; we will see how to define and collect metrics generated by the application that are relevant to the domain and context of the service.

In the discussion, we will also give concrete examples of how application metrics can reveal otherwise hidden pitfalls.

Without further ado, let's begin to delve into the fundamental concepts to arrive at the definition of a sound monitoring system.

Monitoring of applications in the Cloud

A monitoring system is a set of tools and processes designed to collect, analyze, and present relevant data about the application and cloud infrastructure.

The data should allow administrators and developers to profile solution performance, identify potential issues, and resolve them promptly, ensuring a flawless user experience. The same data can be exploited to more deeply understand the implications of infrastructure and software architecture choices and to make improvements and optimizations.

A good monitoring system should offer near real-time visibility into the operation of the application and infrastructure. It should be easily integrated with the application and any pre-existing management tools.

As previously mentioned, we can identify two main categories of metrics: infrastructural ones and application ones.

Infrastructure Monitoring

The most widespread and immediately adoptable monitoring is the infrastructural type, which focuses on extracting metrics relating to the infrastructure to support the applications. It includes metrics such as CPU usage, bandwidth, disk throughput, RAM usage, and the number of HTTP requests or incoming connections.

Infrastructure metrics can provide a detailed view of the physical and virtual resources used by the application within the Cloud environment; they are essential to ensure that the infrastructure provides the resources needed by the application and that the latter operates efficiently.

The main metrics that cloud providers make available are:

- Percentage CPU usage
- Percentage or amount of RAM used (often requires an agent)
- Disk usage, both in terms of storage and I/O operations
- Network usage, generally in terms of throughput
- Cloud Services-specific metrics, such as the number of messages in a queue or events injected into an event bus.
-

There may need to be more than an infrastructural monitoring system to obtain a truthful general picture of the health and performance of a service. In fact, there are situations in which exclusively infrastructural monitoring may not be able to detect problems that impact the user experience, even in extreme cases where they are blocking for end users.

Keep in mind that infrastructure metrics only provide part of the overall picture of your application's performance. Hence, it would help if you thought of a way to get more insights into how your application is performing.

What can go wrong?

At this point, asking what situations an infrastructure monitoring system can miss is legitimate.

We can explore two typical scenarios to understand in which situations infrastructure metrics are insufficient.

Logical error or bug in the application

Let's assume that with the release of a new application version, a bug has been introduced that prevents end users from acting, for example, due to a front-end rendering problem.

From the point of view of infrastructure metrics, this error would not be detectable, except perhaps following abandonment by users, which would cause a lower-than-normal use of resources. In this situation, the problem will likely become known from user reports before it can be deduced from the metrics collected, effectively decreeing the simultaneous failure of the release and quality control procedures and the monitoring system.

We can also imagine a scenario in which the release is successful, working smoothly for days or weeks until an unexpected condition renders some service features unusable. While most unhandled errors can be caught by infrastructure metrics, for example, by counting the number of HTTP errors, many other errors are "handled" by displaying a message to the user. If the application is not designed to output a metric specification or an error notification, the monitoring system will have no way of detecting the criticality.

Finally, the most trivial case is the simple propagation of a version containing a logical error, where one or more application functions do not register errors but provide an incorrect result to users, preventing them from continuing to use the service correctly.

Integration errors or errors in third-party services

Many applications rely on third-party services, such as managing online payments or user authentication. Integration issues or malfunctions in third-party services can significantly impact user experience, and it's easy for such situations to go undetected by infrastructure metrics.

Suppose, for example, that the application uses a payment service and that the latter stops correctly processing payments relating to a payment circuit. Often, the payment error is returned directly to the user, while the application does not suffer a failure and continues to operate correctly. In this scenario, the infrastructure metrics may not make it visible that a portion of the user cannot finalize the payment.

Infrastructure metrics are a critical aspect of cloud application monitoring. However, they are insufficient to ensure that users enjoy the service satisfactorily. To gain a complete view of application performance and provide a high-quality user experience, it is essential also to

set up application metrics monitoring to evaluate the application more thoroughly and accurately.

Application monitoring

Finally, we come to talk about application metrics. These metrics can be extrapolated from the application, for example, by parsing logs or querying the database.

In more modern applications, or those under development, it is possible to ensure that the application explicitly emits the metrics of interest, sending them regularly to the monitoring service that collects them and makes them available to operators.

Application metrics provide information on the internal functioning of the application, giving visibility on elements close to the business logic and on the customer's interaction with the software.

To get into the perspective of the metrics we need to learn to extract, let's suppose we manage an e-commerce; our service's goal is to sell as much as possible and satisfy customers with a good user experience.

In this scenario, an interesting application metric could be the number of logged-in users who are shopping, measured by the application by outputting the number of active sessions that have made requests in the last 30 minutes. The same information could also be obtained from a dedicated service, which periodically executes queries on the e-commerce database to obtain the metric and sends it to the monitoring system.

Another metric representative of the application's state could be the number of carts with at least one item, which gives us an idea of how many people intend to buy something and, therefore, the pipeline of potential sales. And again, the total number of items in the carts of the entire user base or the total amount of potential sales. Finally, the most representative metric of achieving the objective is the number of finalized purchases.

To choose the most suitable metrics to emit from the application, it is necessary to consider the type of application and the objectives the organization wants to achieve by displaying the service to customers. It is, therefore, essential to identify the key functions, the most important ones, and those linked to achieving the objectives.

In other words, we need to identify the service's **value** to customers and define how to measure it with the data available to the software. To give another example in a different field, let's suppose we want to collect metrics for a music streaming service: the value is undoubtedly the number of songs listened to and the listening time. Other valuable metrics

could make new customers' retention and acquisition rate visible, for example, by measuring the subscription rate, unsubscription rate, etc.

Once you know the value you want to offer customers, you can define metrics that reflect the most relevant aspects of your business for monitoring the application and evaluating its performance.

Secondly, it is necessary to identify the **processes** involved in the service provision and understand whether part of their output can be measured using data and events to which the service has visibility.

Returning to the example of e-commerce, the processes that visibly impact the user experience and whose performance can easily be obtained through the service could be package preparation time, delivery time, number of orders, number of complaints, and so on Street. Monitoring this type of metrics allows you to identify a possible inefficiency or problem and act promptly and automatically.

By centrally collecting application metrics closer to the business and infrastructure metrics, the picture of the application's health becomes much more precise and complete. It becomes possible to relate both types of metrics, identify a wide range of critical issues, and intervene in a targeted way.

Most monitoring solutions allow the collection of custom metrics, so it is a good practice to use the application, agent, or additional modules to output application metrics and send them to the centralized monitoring system.

Conclusions

In conclusion, we have seen how application metrics are a fundamental tool for evaluating the actual functioning of a service, which must also be combined with infrastructure metrics to obtain a complete picture of the application status.

We also talked about how to define and collect metrics aligned with business objectives so you can monitor the most critical and relevant aspects for the application's success.

Finally, we presented examples of how application metrics can reveal otherwise hidden pitfalls, such as silent errors, progressive degradations, or behavioral anomalies.

We hope this can help understand more about how to monitor your systems efficiently.

Keep following us for new articles! See you in 14 days.

About Proud2beCloud

Proud2beCloud is a blog by beSharp, an Italian APN Premier Consulting Partner expert in designing, implementing, and managing complex Cloud infrastructures and advanced services on AWS. Before being writers, we are Cloud Experts working daily with AWS services since 2007. We are hungry readers, innovative builders, and gem-seekers. On Proud2beCloud, we regularly share our best AWS pro tips, configuration insights, in-depth news, tips&tricks, how-tos, and many other resources. Take part in the discussion!

Alessio Gandini

Cloud-native Development Line Manager @ beSharp, DevOps Engineer and AWS expert.Since I was still in the Alfa version, I'm a computer geek, a computer science-addicted, and passionate about electronics all-around.At this moment, I'm hanging out in the IoT world, also exploring the voice user experience, trying to do amazing (Io)Things.Passionate about cinema and great TV series consumer, Sunday videogamer

---