

L'importanza delle metriche applicative nel monitoraggio di servizi Cloud.

24 Novembre 2023 - 10 min. read

Oggi più che mai, l'erogazione di servizi web è parte sempre più centrale di ogni progetto.

I clienti si aspettano di fruire in modo fluido ed ininterrotto dei servizi messi a loro disposizione, indipendentemente dal momento della giornata o dal traffico gestito.

Mantenere un'esperienza utente di buona qualità è diventato quindi un elemento essenziale all'interno di un mercato altamente competitivo, dove qualsiasi interruzione o rallentamento nell'erogazione dei servizi può avere serie conseguenze, come la perdita di clienti, di opportunità di vendita o il peggioramento della reputazione aziendale.

L'esperienza utente non riguarda solo la facilità d'uso di un'applicazione, ma anche la velocità di risposta e l'affidabilità percepita. In questo contesto, il monitoraggio dei sistemi e delle applicazioni a diretto contatto con il pubblico è un aspetto critico per permettere alle aziende di reagire prontamente ad eventi avversi che potrebbero compromettere la disponibilità e fruibilità dei servizi che offrono.

Esistono svariate soluzioni di monitoraggio che promettono di fornire un quadro completo sull'operatività dei sistemi in modo semplice e senza richiedere particolari interventi.

Tuttavia, l'impiego di una soluzione generica è solo parzialmente preferibile alla totale assenza di monitoraggio. Infatti, un monitoraggio approssimativo e generico può dare la falsa sicurezza di avere tutto sotto controllo, pur essendo ciechi a situazioni che impattano l'esperienza utente e che sono visibili dagli utenti. Utilizzare metriche "di default" o più in generale non adatte al monitoraggio del caso specifico, fornisce una visione parziale o distorta della realtà, che può facilmente portare a mancate segnalazioni.

Questo tipo di situazioni, in cui i clienti sperimentano rallentamenti o errori durante la fruizione dei servizi, sono deleterie per la retention e la reputazione aziendale; diventano

ancora più gravi se l'organizzazione non è a conoscenza della situazione, e di conseguenza non può gestirla tempestivamente.

Fortunatamente, la maggior parte delle soluzioni di monitoraggio possono essere personalizzate per monitorare metriche e KPI personalizzati, che siano veramente indicativi della salute dell'applicazione. Questo tipo di configurazione è generalmente time consuming e richiede un'attenta analisi del flusso utente e degli eventi applicativi per determinare quali grandezze sono realmente rappresentative del buon funzionamento del servizio erogato, tuttavia è un'attività essenziale per avere un'immagine veritiera dello stato di salute del servizio.

Il sistema di monitoring ideale raccoglie e mette in relazione sia le metriche applicative, personalizzate e pertinenti al caso specifico, sia quelle infrastrutturali, in modo da avere un quadro completo e veritiero di come stia funzionando l'applicazione. Vedremo come l'importanza delle metriche applicative non possa essere sottovalutata per avere una visione completa e accurata dello stato di salute di un workload.

Mentre estrarre metriche infrastrutturali è solitamente assistito dal Cloud provider, non vi è nulla di standard per quanto riguarda le metriche applicative.

In questo articolo, parleremo dell'importanza di monitorare metriche che riflettano il reale funzionamento di un'applicazione, vedremo come definire e raccogliere metriche generate dall'applicazione che siano pertinenti al dominio e al contesto del servizio.

Nella trattazione racconteremo inoltre alcuni esempi concreti di come le metriche applicative possono rivelare insidie altrimenti nascoste.

Senza ulteriori indugi iniziamo ad approfondire i concetti fondamentali per arrivare alla definizione di un buon sistema di monitoring.

Il monitoring delle applicazioni in Cloud

Un sistema di monitoring è un insieme di strumenti e di processi progettati per raccogliere, analizzare e presentare dati rilevanti sull'applicazione e sull'infrastruttura cloud.

I dati dovrebbero consentire agli amministratori e agli sviluppatori di profilare le performance della soluzione, identificare potenziali problemi e risolverli tempestivamente, garantendo un'esperienza utente priva di criticità. Gli stessi dati possono essere sfruttati per capire più profondamente le implicazioni delle scelte infrastrutturali e di architettura del software, e per apportare miglioramenti e ottimizzazioni.

Un buon sistema di monitoring dovrebbe offrire visibilità in near real-time sull'operatività dell'applicazione e dell'infrastruttura, e dovrebbe essere facilmente integrabile sia con l'applicazione che con eventuali strumenti di gestione preesistenti.

Come precedentemente accennato, possiamo identificare due principali categorie di metriche: quelle infrastrutturali, e quelle applicative.

Monitoring Infrastrutturale

Il monitoring più diffuso e di immediata adozione è quello di tipo infrastrutturale, che si concentra sull'estrazione di metriche relative all'infrastruttura a sostegno degli applicativi. Include metriche come l'utilizzo della CPU, la larghezza di banda utilizzata, il throughput del disco, l'utilizzo di RAM, il numero di richieste HTTP o di connessioni in ingresso.

Spesso, queste metriche possono essere fornite direttamente dal cloud provider, e non è richiesta alcuna modifica applicativa per raccoglierle.

Le metriche infrastrutturali possono fornire una visione dettagliata delle risorse fisiche e virtuali utilizzate dall'applicazione all'interno dell'ambiente Cloud, sono essenziali per garantire che l'infrastruttura fornisca le risorse necessarie all'applicazione e che quest'ultima operi in modo efficiente.

Le principali metriche che i cloud provider mettono a disposizione sono:

- Utilizzo percentuale della CPU
- Percentuale o quantità di RAM utilizzata (spesso richiede un agent)
- Utilizzo del Disco, sia in termini di storage che di operazioni di I/O
- Utilizzo della Rete, generalmente in termini di throughput
- Metriche specifiche dei Servizi Cloud, come per esempio numero di messaggi in una coda, o di eventi immessi in un event bus.

Un sistema di monitoraggio infrastrutturale, sebbene fondamentale, può risultare non sufficiente per ottenere un quadro generale veritiero dello stato di salute e delle performance di un servizio. Esistono infatti situazioni in cui un monitoraggio esclusivamente infrastrutturale può non essere in grado di rilevare problemi che impattano l'esperienza utente, fino, in casi estremi, in cui risultano bloccanti per gli utenti finali.

Occorre considerare che le metriche infrastrutturali forniscono solo una parte del quadro complessivo delle prestazioni dell'applicazione, pertanto è necessario pensare ad un modo per ottenere maggiori insight su come l'applicazione stia performando.

Cosa può andare storto?

A questo punto, è lecito domandarsi quali situazioni possono sfuggire ad un sistema di monitoraggio dell'infrastruttura.

Per capire in quali situazioni le metriche infrastrutturali non sono sufficienti, possiamo esplorare due scenari tipici.

Errori logici o bug nell'applicazione

Ipotizziamo che con il rilascio di una nuova versione dell'applicazione sia stato introdotto un baco che impedisce agli utenti finali di svolgere un'azione, per esempio per un problema di rendering del front-end.

Dal punto di vista delle metriche infrastrutturali, tale errore non sarebbe rilevabile, se non forse in seguito all'abbandono da parte degli utenti, che provocherebbe un utilizzo di risorse inferiore alla norma. In questa situazione, è probabile che si venga a conoscenza del problema dalle segnalazioni degli utenti prima che sia deducibile dalle metriche raccolte, decretando, di fatto, il fallimento simultaneo delle procedure di rilascio e quality control, sia del sistema di monitoraggio.

Possiamo anche immaginare uno scenario in cui il rilascio sembra essere andato a buon fine, con un funzionamento regolare di giorni o settimane, fino a quando una condizione non prevista rende inutilizzabile alcune funzioni del servizio. Mentre una parte degli errori non gestiti può essere intercettato dalle metriche infrastrutturali, per esempio conteggiando il numero di errori HTTP, molti altri errori vengono "gestiti" mostrando un messaggio all'utente, e se l'applicazione non è progettata per emettere una metrica specifica, o una notifica di errore, il sistema di monitoraggio non avrà alcun modo di rilevare la criticità.

Infine, il caso più banale è la semplice propagazione di una versione che contiene un errore logico, dove una o più funzioni dell'applicazione non registrano errori, ma forniscono un risultato errato all'utente, che gli impedisce quindi di continuare a fruire correttamente del servizio.

Errori di integrazione o indisponibilità di servizi terzi

Molte applicazioni fanno affidamento su servizi di terze parti, per esempio per la gestione dei pagamenti online, oppure per l'autenticazione degli utenti. Problemi di integrazione o malfunzionamenti nei servizi di terze parti possono influenzare notevolmente l'esperienza utente, ed è facile che tali situazioni non vengano rilevate dalle metriche infrastrutturali.

Supponiamo, ad esempio, che l'applicazione utilizzi un servizio di pagamento, e che quest'ultimo smetta di processare correttamente i pagamenti relativi ad un circuito di pagamento. Spesso, l'errore di pagamento viene restituito direttamente all'utente, mentre l'applicazione non subisce un fallimento e continua ad operare correttamente invitando l'utente a riprovare. In questo scenario, le metriche infrastrutturali, potrebbero non rendere visibile che una porzione dell'utenza non riesce a finalizzare il pagamento.

Le metriche dell'infrastruttura sono un aspetto fondamentale del monitoraggio delle applicazioni cloud, tuttavia non sono sufficienti a garantire che gli utenti stiano fruendo del servizio in modo soddisfacente. Per ottenere una visione completa delle prestazioni dell'applicazione e garantire un'esperienza utente di alta qualità, è essenziale predisporre anche il monitoraggio delle metriche applicative per valutare l'applicazione in modo più completo e accurato.

Monitoraggio applicativo

Arriviamo infine a parlare delle metriche applicative. Queste sono essenzialmente delle metriche che possono essere estrapolate dall'applicazione, per esempio effettuando il parsing dei log, oppure facendo query sul database.

Nelle applicazioni più moderne, o in fase di sviluppo, è possibile fare in modo che l'applicazione stessa emetta in modo esplicito le metriche di interesse, inviandole regolarmente al servizio di monitoring che le raccoglie e le rende disponibili agli operatori.

Le metriche applicative forniscono informazioni sul funzionamento interno dell'applicazione, dando visibilità su elementi vicini alla business logic e sull'interazione che il cliente ha con il software.

Per entrare nell'ottica del tipo di metriche che dobbiamo imparare ad estrarre, supponiamo di gestire un e-commerce; l'obiettivo del nostro servizio è quindi quello di vendere il più possibile e di soddisfare i clienti con una buona esperienza utente.

In questo scenario, una metrica applicativa interessante potrebbe essere il numero di utenti loggati che stanno facendo shopping, misurato dall'applicazione emettendo il numero di sessioni attive che hanno fatto richieste negli ultimi 30 minuti. La stessa informazione potrebbe anche essere ricavata da un servizio dedicato, che esegue periodicamente query sul database dell'e-commerce al fine di ottenere la metrica, e provvede ad emetterla verso il sistema di monitoraggio.

Un'altra metrica rappresentativa dello stato dell'applicazione potrebbe essere il numero di carrelli con almeno un articolo, che ci da un'idea di quante persone abbiano intenzione di

acquistare qualcosa, e quindi della pipeline delle potenziali vendite. E ancora, il numero totale di articoli nei carrelli di tutta la base utenti, o l'importo totale delle vendite potenziali. Infine, la metrica più rappresentativa del raggiungimento dell'obiettivo è sicuramente il numero di acquisti finalizzati.

Per scegliere le metriche più adatte da far emettere all'applicazione, bisogna considerare il tipo di applicazione e gli obiettivi che l'organizzazione vuole raggiungere mediante l'esposizione del servizio ai clienti. Occorre quindi individuare le funzioni chiave, quelle più importanti e collegate al raggiungimento degli obiettivi.

In altre parole bisogna identificare il **valore** che il servizio offre ai clienti, e definire come misurarlo con i dati a disposizione del software. Per fare un'altro esempio in un campo diverso, supponiamo di voler raccogliere metriche per un servizio di streaming musicale: il valore è certamente il numero di brani ascoltati, ma anche il tempo di ascolto. Altre metriche utili potrebbero rendere visibile il tasso di fidelizzazione e di acquisizione di nuovi clienti, per esempio misurando il tasso di abbonamento, il tasso di disiscrizione, ecc.

Una volta che è chiaro il valore che si vuole offrire ai clienti, si possono definire delle metriche che riflettano gli aspetti più rilevanti del proprio business per il monitoraggio dell'applicazione e la valutazione delle sue performance.

In secondo luogo, occorre identificare i **processi** aziendali che sono coinvolti nell'erogazione del servizio, e capire se parte del loro output può essere misurato usando dati ed eventi cui il servizio ha visibilità.

Tornando all'esempio dell'e-commerce, i processi che incidono visibilmente sull'esperienza utente, e di cui è facilmente possibile ottenere informazioni sulle performance mediante il servizio potrebbero essere il tempo di preparazione dei pacchi, il tempo di consegna, il numero di ordini, il numero di reclami, e così via. Monitorare questa tipologia di metriche permette di identificare una possibile inefficienza o un problema in modo automatico, e di agire tempestivamente.

A questo punto dovrebbe essere chiaro che raccogliendo in modo centralizzato metriche applicative, più vicine al business, e metriche infrastrutturali, il quadro sulla salute dell'applicazione diventa molto più chiaro e completo. Diventa possibile mettere in relazione entrambi le tipologie di metriche, per identificare una vasta gamma di criticità ed intervenire in modo mirato.

La maggior parte delle soluzioni di monitoring permettono la raccolta di metriche personalizzate, è quindi una buona pratica utilizzare l'applicazione, agent o moduli

addizionali per emettere le metriche applicative, ed inviarle al sistema di monitoring centralizzato.

Conclusioni

In conclusione, abbiamo visto come le metriche applicative siano uno strumento fondamentale per valutare il reale funzionamento di un servizio, da abbinare anche alle metriche infrastrutturali per ottenere il quadro completo dello stato dell'applicazione.

Abbiamo anche parlato di come definire e raccogliere metriche che siano allineate agli obiettivi di business, in modo da poter monitorare gli aspetti più critici e rilevanti per il successo dell'applicazione.

Infine, abbiamo presentato alcuni esempi di come le metriche applicative possano rivelare insidie altrimenti nascoste, come errori silenziosi, degradazioni progressive o anomalie comportamentali.

Speriamo che questo deep-dive sia stato utile.

Ci vediamo tra 14 giorni con un nuovo articolo!

About Proud2beCloud

Proud2beCloud è il blog di [beSharp](#), APN Premier Consulting Partner italiano esperto nella progettazione, implementazione e gestione di infrastrutture Cloud complesse e servizi AWS avanzati. Prima di essere scrittori, siamo Solutions Architect che, dal 2007, lavorano quotidianamente con i servizi AWS. Siamo innovatori alla costante ricerca della soluzione più all'avanguardia per noi e per i nostri clienti. Su Proud2beCloud condividiamo regolarmente i nostri migliori spunti con chi come noi, per lavoro o per passione, lavora con il Cloud di AWS. Partecipa alla discussione!



Alessio Gandini

Cloud-native Development Line Manager @ beSharp, DevOps Engineer e AWS expert. Computer geek da quando avevo 6 anni, appassionato di informatica ed elettronica a tutto tondo. Ultimamente sto esplorando l'esperienza utente vocale e il mondo dell'IoT. Appassionato di cinema e grande consumatore di serie TV, videogiocatore della domenica.
