

STATEFUL VS. STATELESS: COSA È MEGLIO?

Software as a Service (SaaS)

Statefull

Stateless



beSharp | 30 Dicembre 2019

Nell'era dei **servizi SaaS**, la costruzione di microservizi stateless sembra essere l'unico modo per raggiungere il successo. Ma è sempre possibile progettare servizi stateless? E quando è opportuno progettare invece microservizi di tipo Stateful?

Cominciamo dando una prima definizione dei due termini:

Parliamo di **microservizi Stateless** quando nessun dato viene memorizzato direttamente sull'host. Il servizio è perfettamente in grado di funzionare servendosi solo delle informazioni disponibili nel payload delle richieste, oppure acquisendo le informazioni necessarie da un servizio Stateful dedicato, come ad esempio un database.

Nel caso di **microservizi Stateful**, al contrario, le richieste possono essere servite esclusivamente grazie ad un sistema di memorizzazione dei dati sull'host.

Stateful vs Stateless: cosa cambia?

Stateless

Il server elabora le richieste sulla base delle sole informazioni trasmesse **ad ogni singola richiesta**, non servendosi di informazioni relative a richieste precedenti (in questo caso, non c'è alcun bisogno di mantenere informazioni di stato tra una richiesta e l'altra), oppure serve le richieste attingendo, per le informazioni di stato, da un servizio esterno, come un **database**.

E' interessante sottolineare che il fatto che qualsiasi istanza possa recuperare tutte le informazioni di stato necessarie per eseguire un comportamento consente di ottenere **resilienza** ed **elasticità**, oltre che la possibilità per qualsiasi istanza di eseguire qualsiasi attività. **L'efficienza**, inoltre, è garantita dal fatto che richieste diverse possono essere elaborate anche da server diversi.

Stateful

Alcuni esempi di servizi a stato sono **database** e servizi basati su protocolli internet che necessitano di una stretta **gestione dello stato** sui singoli host. Il server elaborerà le richieste sia in base alle informazioni trasmesse ad ogni richiesta, che alle informazioni memorizzate da richieste precedenti. Nel caso di microservizi Stateful, **lo stesso server dovrà servire tutte le richieste** collegate alle stesse informazioni di stato (oppure saranno le informazioni di stato a dover essere condivise con tutti i server che ne hanno bisogno per servire richieste).

Cosa comporta la gestione Stateful di carichi di lavoro?

Isolamento delle risorse

Molte delle attuali soluzioni di orchestrazione dei container sul mercato sono ancora basate su un approccio di tipo “best-effort” per quanto riguarda l’allocazione di risorse come CPU, memoria e storage. Questo risulta essere comunque un approccio rischioso che può causare la perdita di transazioni dei clienti o di dati a causa di prestazioni inaffidabili.

Storage persistente

Ogni servizio Stateful può richiedere e supportare tipi diversi di storage (ad esempio dispositivi a blocchi o file system distribuiti). Determinare il tipo di storage di supporto migliore per un servizio a stato può essere impegnativo.

Possiamo ritenere questi due limiti due conseguenze del fatto che **molti microservizi Stateful sono stati pensati e costruiti per ambienti legacy** e probabilmente **monolitici**.

Per superare limiti di questo tipo, le organizzazioni potrebbero iniziare tentando di **containerizzare i loro servizi Stateful**. Per poterlo fare, però, avrebbero bisogno di sviluppare sia strumenti altamente specifici al fine di riuscire a coordinare tutte le istanze correlate ed ottenere così **High Availability**, sia strategie sofisticate per distribuire, gestire o far funzionare al meglio questi servizi. Questo potrebbe causare all’azienda un notevole **overhead manuale** probabilmente dispendioso in termini di tempo e costi, oltre che far nascere la necessità di sviluppare **operation personalizzate** per ogni singolo servizio con conseguente notevole rischio operativo.

E il SaaS?

Il SaaS (**Software as a Service**) è un modello di licenza e fornitura dei software. Questi, hostati in modo centralizzato, vengono erogati su abbonamento.

Contrariamente a quanto si pensa, il **SaaS non richiede necessariamente microservizi di tipo Stateless**, anzi: non tutti i processi possono essere resi tali (o non per tutti vale la pena essere Stateless!) è comunque possibile costruire servizi SaaS di successo sia Stateless, che Stateful. Resta comunque vero che un servizio monolitico di tipo Stateful sarà probabilmente più costoso e difficile da mantenere, oltre che meno scalabile. Inoltre, avrà bisogno di una gestione particolare sia per i backup, che per questione di HA.

Conclusione

Se è vero che progettare i **microservizi Stateless** ha molti vantaggi, soprattutto in termini di **scalabilità, costi ed efficienza** (si pensi ad esempio ad una infrastruttura di notevoli dimensioni, con un bacino di utenti considerevole e distribuita geograficamente a livello mondiale), a volte progettare Stateful è una necessità.

Ora sappiamo che questo non preclude affatto la possibilità di ottenere un **SaaS di successo** sia dal punto di vista tecnico, che di business.

Tutto ciò che occorre fare in più è garantire un **buon livello di scalabilità** e introdurre un ottimo **piano di backup e Disaster Recovery**.

Volete qualche consiglio in più al riguardo? Continuate a seguirci per carpire i trucchi del mestiere





beSharp

Dal 2011 beSharp guida le aziende italiane sul Cloud. Dalla piccola impresa alla grande multinazionale, dal manifatturiero al terziario avanzato, aiutiamo le realtà più all'avanguardia a realizzare progetti innovativi in campo IT.

Get in touch

beSharp.it
proud2becloud@besharp.it

Copyright © 2011-2021 by beSharp srl - P.IVA IT02415160189