RightSizing-SpikeServer

1 Caso di Studio

Il sistema oggetto di studio è un'architettura di data center per un Internet Service Provider, progettata per gestire dinamicamente le fluttuazioni di carico e garantire la Quality of Service (QoS) ottimizzando al contempo l'uso delle risorse.

Il problema principale affrontato è il "right-sizing", ovvero come evitare sia il sovradimensionamento (spreco di risorse) sia il sottodimensionamento (violazione degli SLA e degrado delle prestazioni), specialmente in presenza di fluttuazioni di carico a breve e lungo termine.

L'architettura proposta, come descritto nel caso di studio 6.2 del libro di testo "Performance Engineer", si basa su un livello di scaling verticale che gestisce i picchi di carico improvvisi e di breve durata. Questo livello introduce uno Spike Server dedicato. Un Load Controller monitora un indicatore di picco (Spike Indicator, SI), definito come il numero di richieste concorrenti in esecuzione su un Web Server.

Il comportamento del sistema seguirebbe quanto descritto:

- Quando l'indicatore SI supera una soglia di allarme SI_max, le nuove richieste in arrivo non vengono più inviate al Web Server congestionato, ma vengono reindirizzate allo Spike Server.
- Quando il carico sul Web Server diminuisce e SI scende al di sotto della soglia, il routing delle richieste torna alla normalità.

2 Obiettivi dello studio

Lo studio si pone l'obiettivo di analizzare e validare l'efficacia del modello di autoscaling gerarchico attraverso la simulazione. Gli obiettivi specifici sono:

- identificare il valore di carico di lavoro che supera i tempi di risposta richiesti
- Identificare il valore ottimale di SI_max: Per un dato carico di lavoro (es. 40,000 richieste/ora), determinare il valore di SI_max che minimizza il tempo di risposta medio del sistema o lo mantiene al di sotto di un valore target definito da un Service Level Agreement, come gli 8 secondi utilizzati nel libro.

- Studiare il comportamento del sistema sotto carichi crescenti: Analizzare le prestazioni del sistema (in particolare il tempo di risposta) al variare del tasso di arrivo delle richieste (carico leggero, medio, pesante) utilizzando la soglia SI_max.
- Studiare il sistema con fluttuazioni a breve e a lungo termine: Analizzare le prestazioni del sistema introducendo fluttuazioni a breve e lungo termine, utilizzando sempre la soglia SI_max
- Trovare due nuovi SI_max per fluttuazioni a breve e a lungo termine per rimanere sotto un certo tempo di risposta (8 secondi)

3 Modello concettuale

Il modello descritto può essere schematizzato nel seguente modo:

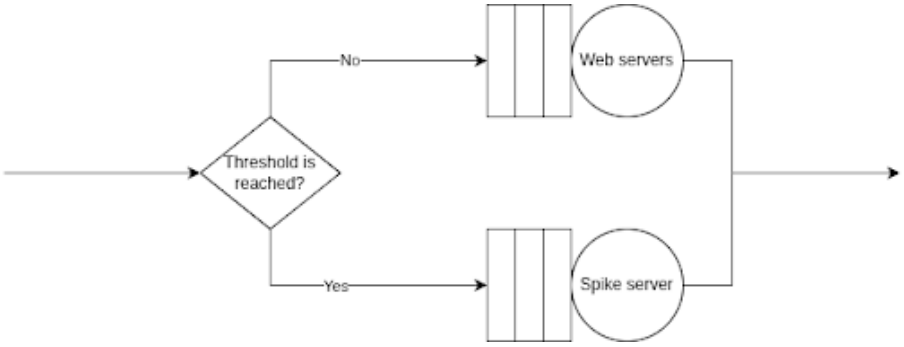

Figure 1: Modello concettuale del sistema di autoscaling gerarchico con Spike Server

I job arrivano e a seconda del livello di pienezza dei webservers. Nonostante il sistema possa sembrare troppo semplice per una analisi simulativa, ci sono una serie di aspetti che lo rendono complesso e difficilmente modellabile solo matematicamente senza semplificazioni:

- Tempi di arrivo iperesponenziali: Questi arrivi e i servizi non esponenziali vengono utilizzati in questo contesto per modellare le fluttuazioni del carico. Matematicamente non sono facilmente modellabili se non considerando le relazioni che valgono per delle distribuzioni generiche, Il che porterebbe a meno informazioni di valore per l'analisi.
- Il routing non è probabilistico: il routing dei job non è semplicemente probabilistico (40% su uno e 60% su un altro), ma dipende strettamente dallo stato dei webservers nel momento del routing. Questa complicazione rende molto difficile un'analisi statica, soprattutto nel transiente.

Spiegazione tempi di servizio esponenziale: Nel caso di studio affrontato nel libro viene utilizzata una distribuzione iperesponenziale nel tassi di servizio per modellare il fatto che ad un server arrivano job di dimensione molto variabile. Ovviamente questa cosa potrebbe essere modellata anche utilizzando delle size dei job differenti anziché agire sul tasso di servizio. Tuttavia ho deciso di attenermi al testo originale e utilizzare anche io dei tassi di servizio iperesponenziali per modellare questo comportamento.