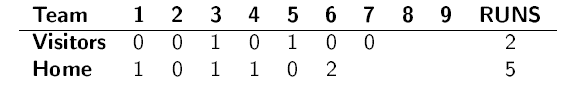
Abstract

Sistemi di storage moderni replicano i dati su più macchine al fine di garantirne persistenza, low latency e tolleranza ai guasti. La presenza di più repliche di un medesimo dato genera quindi problemi di coerenza fra i dati stessi. Tale coerenza può essere garantita mediante diversi modelli, ognuno con i propri punti di forza e le proprie debolezze. Chiaramente, non ne esiste uno valido in generale e la scelta deve essere fatta tenendo in considerazione quali proprietà del sistema riteniamo maggiormente rilevanti ai nostri scopi. Pertanto, prendendo in esame i due sistemi di key-value store distribuiti Amazon Dynamo e COPS, è stato possibile verificare, anche tramite confronti, come questi compromessi vengono raggiunti.

Introduzione

Ogni applicazione necessita di un livello di coerenza che cambia a seconda degli scopi che vuole raggiungere e delle priorità che vuole ottenere. Infatti, nel 2000 è stato congetturato, e successivamente dimostrato, un risultato teorico che prova che in ogni sistema distribuito non è possibile soddisfare contemporaneamente tutte e tre le seguenti proprietà: *availability*, ossia ogni richiesta riceve sempre una risposta su ciò che è riuscito o fallito non rimanendo mai in attesa indefinitamente, *partition-tolerance*, ossia il sistema continua a funzionare nonostante arbitrarie perdite di messaggi, *consistency*, ossia tutti i nodi vedono gli stessi dati simultaneamente. Questa conclusione è stata dimostrata nel 2002, dopo quasi trenta anni di ricerca, e va sotto il nome di *Teorema CAP*.

Per descrivere alcuni differenti modelli di coerenza, utilizzeremo come esempio i punteggi di una partita di baseball, memorizzati all’interno di un key-value store distribuito, come mostrato di seguito:



**Strong consistency.**  La garanzia più alta che possiamo raggiungere fornisce ad ogni client che effettua operazioni di lettura sempre l’ultimo valore aggiornato. Per implementare tale livello, è necessario un alto livello di sincronizzazione tra i vari nodi, che per essere raggiunto esige attese, causando un calo delle performance e la partecipazione attiva di tutti i nodi. In riferimento all’esempio, è possibile che ci venga ritornato unicamente il punteggio 2-5.

**Eventual Consistency.** Formalmente l’eventual consistency consente di ritornare un qualunque valore che è stato scritto su un dato che il client vuole leggere. In pratica, quello che questo livello garantisce è che se non vengono eseguiti nuovi aggiornamenti su un oggetto, eventualmente tutti gli accessi a quell’oggetto ritorneranno l’ultimo valore aggiornato.

Dettagli Tecnici

Introduzione generale sui sistemi

Cops è distribuito su scala geografica con repliche identiche degli stessi dati distribuite in più datacenter.

Dynamo è costituito da nodi potenzialmente distribuiti in più datacenter distinti in cui sono distribuiti e replicati i dati.

In COPS la specifica afferma che i dati non sono partizionati fra i vari datacenter che formano il sistema, ma replicati tra essi.

Per fare un paragone tra i due sistemi, consideriamo in prima istanza i datacenter di COPS come nodi atomici alla pari dei nodi di Dynamo. In realtà, ogni datacenter è costituito da un cluster locale formato da una serie di nodi, che implementano strong consistency, per servire la base di client affiliata.

**Replicazione.** In COPS ogni datacenter possiede una replica di tutti i dati. In Dynamo è più complesso, i dati non sono replicati su tutti i nodi ma solo su un numero N (N << # totale nodi)

COPS usa la replicazione principalmente per avvicinare i dati ai diversi client distribuiti geograficamente, oltre che per garantire chiaramente persistenza dei dati stessi. In Dynamo invece l’accento è posto proprio sulla fault tolerance e sulla availability, nonché sulle performance.

In COPS non è contemplata la perdita totale dei dati di un datacenter, nella specifica non è esplicitato se le richieste della base di client di riferimento vengono interrotte o reindirizzate ad un altro datacenter, potenzialmente distante geograficamente.

In Dynamo le richieste sono accolte automaticamente dai nodi attivi che mantengono le repliche dei dati interessati.

(\* versionamento \*)

**Coerenza.** In COPS implementiamo causal consistency+, in quanto si vuole garantire la coerenza più forte possibile che permetta, al contempo, di soddisfare le caratteristiche ALPS, nel rispetto del Teorema CAP. È impossibile quindi implementare livelli di coerenza maggiori, quali la strong consistency o la sequential consistency.

In Dynamo, sono implementati due modelli di coerenza, strong consistency ed eventual consistency, impostabili mediante dei parametri che governano direttamente anche performance, availability e persistenza dei dati. Questi ultimi parametri, sempre nel rispetto del Teorema CAP, consentono di bilanciare il sistema, permettendone un diverso uso in base alle esigenze.

**Come COPS garantisce coerenza.**

**Come Dynamo garantisce coerenza.** La coerenza delle operazioni è gestista con la tecnica del *quorum* [cite]. Il sistema permette di impostare due parametri: W indica il numero di nodi dai quali viene atteso un acknowledge a seguito di una scrittura, R il numero di nodi che partecipa ad un’operazione di lettura.

Facendo riferimento alla [rif figura], nell’operazione di lettura (*get(k)*)