Sommario

NoSQL è l'acronimo di "Not only SQL" e viene usato generalmente per indicare quei database che non usano un modello di dati relazionale e quindi potrebbero non usare SQL come linguaggio di interrogazione

Vantaggi di Nosql

Una delle principali motivazioni per l'uso di tali database è rappresentata dalla scalabilità. La scalabilità è un requisito sempre più importante per le applicazioni web, e ciò è dovuto a molti fattori: l'esplosione del numero di utenti della rete, la sempre maggiore diffusione di OpenID, quindi la sinergia tra i fornitori di servizi, ma anche la crescente disponibilità di dispositivi con accesso ad Internet come smartphone, tablet e altri dispositivi portatili.

Scalabilità orizzontale

La scalabilità orizzontale si ha se l'aumento delle risorse si riferisce all'aumento dei nodi nel sistema, cioè il sistema riesce a parallelizzare il carico di lavoro.

- Il vantaggio più importante dato da questo tipo di scalabilità è il costo: infatti, con un'applicazione perfettamente scalabile, si potrebbero impiegare molti nodi a basso costo.
- Un altro vantaggio è la maggior fault-tolerance: l'errore in un nodo non pregiudica totalmente il funzionamento dell'applicazione.

Scalabilità verticale

La scalabilità verticale si ottiene quando, per aumentare le prestazioni dell'intero sistema, si aumentano le risorse di un singolo nodo del sistema, ad esempio utilizzando una CPU con frequenza maggiore o incrementando la memoria disponibile.

- Il vantaggio di questo tipo di scalabilità è che generalmente non è necessario modificare le applicazioni, e non sono richiesti interventi amministrativi.
- Lo svantaggio consiste innanzitutto nel costo, perché l'aggiornamento spinto di una macchina può essere economicamente molto più gravoso dell'acquisto di una ulteriore macchina di pari potenza.

Due tipologie a confronto:

I database relazionali si basano su modelli relazionali progettati per creare schemi di database, inserire, modificare e gestire dati memorizzati, interrogare i dati memorizzati creare e gestire strumenti di controllo e accesso ai dati.

I database NoSQL sono utilizzati principalmente per la loro facilità di sviluppo, la loro alta funzionalità e la scalabilità orizzontale delle prestazioni. Questo tipo di database è ottimizzato per applicazioni che necessitano di grandi quantità di dati e modelli di dati flessibili.

Mongo db

- MongoDB è un DBMS non relazionale, orientato ai documenti; è un database NoSQL, infatti si allontana dalla struttura tradizionale basata su tabelle relazionali in favore di documenti in stile JSON con schema dinamico, rendendo l'integrazione di dati di alcuni tipi di applicazioni più facile e veloce. Rilasciato sotto Apache, MongoDB è unsoftware libero e open source.
- Sviluppato inizialmente dalla società di software 10gen (ora diventata MongoDB Inc.) nell'ottobre 2007 come un componente di un prodotto di platform as a service, l'azienda si è spostata verso un modello di sviluppo open source nel 2009, con 10gen in grado di offrire un supporto commerciale e altri servizi. Da allora, MongoDB è stato adottato come backend da un alto numero di grandi siti web e societa di servizi come eBay, Foursquare, SourceForge e il New York Times, tra gli altri. MongoDB è il più popolare database NoSQL.

Open Source

- MongoDB è disponibile gratuitamente sotto la GNU Affero General Public License.
- Le librerie per i vari linguaggi sono disponibili sotto la Apache License. In aggiunta, MongoDB Inc. offre licenze commerciali per MongoDB.

Caratteristiche principali

- MongoDB supporta ricerche per campi, intervalli e regular expression. Le query possono restituire campi specifici del documento e anche includere funzioni definite dall'utente in JavaScript.
- Qualunque campo in MongoDB può essere indicizzato (gli indici in MongoDB sono concettualmente similari a quelli dei tradizionali RDBMS). Sono disponibili anche indici secondari, indici unici e indici sparsi.
- MongoDB fornisce alta disponibilità e aumento del carico gestito attraverso i replica set. Un replica set consiste in due o più copie dei dati. Ogni replica può avere il ruolo di copia primaria o secondaria in qualunque momento. La replica primaria effettua tutte le scritture e le letture. Le repliche secondarie mantengono una copia dei dati della replica primaria attraverso un meccanismo di replicazione incluso nel prodotto. Quando una replica primaria fallisce, il replica set inizia automaticamente un processo di elezione per

determinare quale replica secondaria deve diventare primaria. Le copie secondarie possono anche effettuare letture, con dati eventualmente consistenti di default.

- MongoDB memorizza i dati in documenti flessibili JSON-like; i campi possono variare da un documento all'altro ed è possibile modificare nel tempo la struttura dei dati.

Il modo migliore per lavorare con i dati

Il modello flessibile di dati a documento di MongoDB rende intuitivo il lavoro con i dati, sia che tu stia creando un'applicazione da zero o che stia aggiornandone una esistente.

Introduzione a Mongo Db database NoSql

MongoDb un database NoSql che serve a gestire grandissime quantità di dati.

È un database document-based che offre la scalabilità e flessibilità insieme alle funzionalità di query e indicizzazione.

MongoDB memorizza i dati in documenti flessibili JSON-like; i campi possono variare da un documento all'altro ed è possibile modificare nel tempo la struttura dei dati ed è utilizzato per le query ad hoc, l'indicizzazione e l'aggregazione in tempo reale dei dati.

Obiettivi principali:

- Scalabilità grazie principalmente alle nuove tecnologie hardware e i big Data
- Facilitare lo sviluppo
- Rappresentazione dei dati intuitiva. Serve per salvare dati semi strutturati (come file xml) e non strutturati (file o immagini) e strutture complesse di dati

Il vantaggio principale di MongoDb è essere scalabile ***orizzontalmente***, infatti, con un'applicazione perfettamente scalabile, si potrebbero impiegare molti nodi a basso costo.

Altri vantaggi sono:

- Fail-hover se una macchina smette di funzionare le sue attività vengono indirizzate verso un'altra macchina attiva
- Alta disponibilità: assicura la disponibilità del servizio durante varie tipologie di failure
- Disaster recovery: garantire un'erogazione continua dei servizi RPO(recovery point objective)
- Open Source: MongoDB è disponibile gratuitamente sotto la GNU Affero General Public License
  le librerie per i vari linguaggi sono disponibili sotto la Apache License. In aggiunta, MongoDB Inc. offre licenze commerciali per MongoDB.

Svantaggio di MongoDB:

- Per poter essere una via di mezzo tra tutti i tipi di server avendo sia una buona scalabilità che un'ottima funzionalità ha dovuto perdere alcune funzioni SQL: JOIN e le TRANSACTION.

Alternativa al join:
Come creare qualcosa che sostituisca un comando importantissimo come il Join?

Offrendo un altro modo di accesso ai dati con un'alternativa a quelle due funzioni.

Questo è stato possibile con la creazione di qualcosa di potente agile e flessibile: Documented-Oriented dove i singoli record sono memorizzati come documenti;

Questo documento è un equivalente ad un oggetto della programmazione ad oggetti, questo porta una corrispondenza alla OOP e i dati del database, che rende il documento oriented molto agile e flessibile, in questo modo chi utilizza MongoDb può adattarsi molto velocemente.

Questi documenti sono codificati tramite il formato JSON che è un formato molto leggibile al contrario di un Xml.

Grazie a questo metodo MongoDb è riuscito a creare le Pre-Join o Embedding: quando noi effettuiamo un JOIN tra due tabelle ci aspettiamo che il risultato sia l'unione di esse.

Le pre-join fanno il contrario, partono direttamente dal join delle tabelle, questo è possibile perché già prima di salvare il nostro dato sul database, noi includiamo al suo interno anche altre informazioni come se fosse il risultato di un join.

Esempio pratico:
Abbiamo due tabelle

Tabella Cliente

id	name	surname	City
1	Matteo	Aimi	Parma
2	Federico	Chiaffi	Parma

Tabella veicoli

id	model	year	Id_cliente
1	Polo	2018	1
2	207	2014	2
3	Touran	2010	1

Con una normale join possiamo ottenere tutti i veicoli per ogni cliente

Non avendo le join in Mongo prima di salvare le tabelle dobbiamo aggiungerci dati in più creando un documento Json :

```
{
        name:”Matteo”,
        surname:”Aimi”,
        city:”Parma”
        cars[
                {model: “Polo”,
                years:2018},
                {model:”Touran”,
                years:2010}
        ],
}
```

In questo documento abbiamo unito i dati di due tabelle in un unico documento che contiene tutte le informazioni che vogliamo, ad esempio un indirizzo email o numero di telefono. Ovviamente per creare un nuovo Cliente dovremo creare un nuovo documento uguale a quello crato prima solo con i dati diversi.

Tutto ciò assicura a MongoDb delle ottime performance con delle semplici query otteniamo lo stesso risultato che otterremmo con una join relazionale

Fonte http://university.mongodbitalia.it/

Spiegazione utilizzo di MongoDb : https://www.youtube.com/watch?v=LWFC-PWDPIM (video dello sviluppatore di mongodbitalia.it)