**Scelta del database NoSQL**

Deve essere realizzato un **URL shortner**, un servizio che, dato un link in input, ne restituisca un altro abbreviato, generato in modo casuale oppure in modo personalizzato a partire da una stringa inserita dall'utente. Si deve consentire anche di visualizzare alcune statistiche relative ad uno short url generato dal sistema, relative al numero di visite e la loro provenienza geografica in base all'IP.

Obiettivo primario sarà quindi quello di memorizzare la corrispondenza tra link in input (long url) e link abbreviati generati dal sistema (short url), in modo che ogniqualvolta un utente cliccherà su uno short url verrà reindirizzato al long url corrispondente. Oltre a tener memoria della corrispondenza long-short url, si dovranno anche registrare ulteriori informazioni, da cui poter poi elaborare delle statistiche.

Si è deciso quindi che il miglior metodo per poter aggregare questi dati, fosse quello di unirli in un documento, ci siamo perciò indirizzati verso un database NoSQL Document-oriented.

La scelta del database SQL è stata effettuata considerando anche il **CAP Theorem**, secondo cui è impossibile che un sistema distribuito soddisfi contemporaneamente tutte e tre le caratteristiche di **consistenza**, **disponibilità** e **tolleranza della partizione**, ma al massimo 2. Sono state considerate di maggior rilevanza le caratteristiche di tolleranza della partizione, che garantisce il servizio anche qualora una partizione non comunichi con le altre, e di consistenza, in modo che i dati rimangano consistenti dopo l'esecuzione di una scrittura o di un update.

**MongoDB** è il database scelto, visto che appartiene alla categoria dei database NoSQL orientati ai documenti e fa parte di quei database che soddisfano contemporaneamente le caratteristiche evidenziate in precedenza; MongoDB risiede dunque su una definizione **CP** (Cap Theorem), ovvero assicura che, pur potendo essere qualche dato inaccessibile, il resto sia accurato e consistente.

MongoDB è il più popolare database document-oriented e NoSQL, con più di 10 millioni di downolad, migliaia di clienti e più di 1000 technology e service partners.

I documenti di MongoDB sono in stile JSON composti da campi chiave-valore, come ad esempio: { "name": "mongo", "age": 5 }. MongoDB memorizza i documenti sul disco in formato BSON, dove BSON è la rappresentazione binaria di documenti JSON.

MongoDB garantisce **performance elevate**, in particolare:

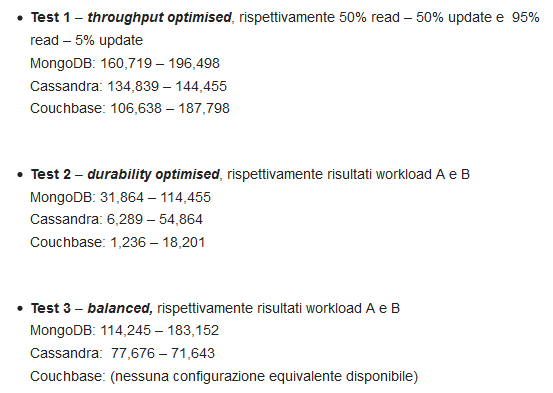
* il supporto per i modelli di dati embedded riduce le attività di I/O su database
* gli indici garantiscono query più veloci e possono includere chiavi da documenti embedded e array.

Fornisce **alta disponibilità** attraverso una funzione di replicazione (Replica Set) che fornisce un failover automatico e ridondanza dei dati. Un Replica Set è un gruppo di MongoDB server che mantengono lo stesso data set, fornendo ridondanza e incrementando la disponibilità dei dati.

Infine MongoDB assicura **scalabilità orizzontale** tramite sharding automatico, distribuendo i dati attraverso cluster, e Replica Set, che forniscono letture eventually-consistent.

Da un report pubblicato dalla United Software Associates (USAIN), dove vengono messe a confronto le prestazioni di tre database NoSQL, Cassandra, Couchbase e MongoDB, quest’ultimo è stato quello che si è distinto nei test prestazionali (*benchmark*) effettuati dai ricercatori. I database sono stati messi alla prova dallo Yahoo! cloud standard benchmark (YCSB) con l’intenzione di accertarne l’effettiva durabilità; il metro di giudizio verte quindi sull’idea che le applicazioni debbano focalizzarsi più sulla durabilità che sulle performance, non tollerando perdite di dati.

Di seguito i risultati dei test, considerando che tutti i valori indicano il numero di operazioni al secondo :



Come è possibile osservare dai risultati MongoDB risulta il migliore in termini di **durabilità**. Anche per questo la nostra scelta è stata indirizzata verso questo database, dato che la consideriamo abbastanza rilevante in un servizio di url shortening. Si desidera infatti che rimanga “intatta” a lungo l’associazione tra uno short url e il corrispettivo long url, ovvero che cliccando su uno short url, anche a distanza di tempo dalla sua creazione, si venga correttamente reindirizzati.

Fonti:

1. Introduction to MongoDB [[https://docs.mongodb.org/manual/core/introduction/]](https://docs.mongodb.org/manual/core/introduction/%5d)
2. Il databse NoSQL più performante? È MongoDB [http://www.hostingtalk.it/il-database-nosql-piu-performante-mongodb/]