

Rozproszone systemy operacyjne

Zapis, przechowywanie danych oraz struktura w MongoDB

Autorzy:

* Tomasz Adamiec
* Piotr Cebulski
* Marek Kowalski
* Mateusz Rosiewicz
* Paweł Sokołowski
* Marcin Wnuk

Warszawa, 2013

* 1. Wyższość MongoDB podczas zapisu

MongoDB charakteryzuje się ogromną szybkością i pojemnością. Wszystkie dane przechowywane są w pamięci RAM i dopiero z stamtąd zapisywane na dysk. Powoduje to, że taka baza jest wymarzonym narzędziem dla wielu funkcjonalności społecznościowych. Funkcje takie jak ściana lub tablica, znajomi, chmura tagów, opierają się zasadniczo na jednej podstawowej zasadzie. To co pisze jedna osoba widoczne jest u wielu osób a dodatkowo nie można tu stosować mechanizmów keszowania, ponieważ użytkownicy społecznościowi oczekują natychmiastowej reakcji na ich akcje (jeżeli piszę coś na ścianie mojego znajomego, on powinien to zobaczyć w tym samym momencie, jeżeli piszę coś na swojej ścianie, to wszyscy moi znajomi muszą zobaczyć to od razu!). Taki system z konieczności musi mieć możliwość ogarnąć ogromne ilości jednoczesnych zapisów z jednoczesnymi odczytami. W wypadku nawet niewielkich społeczności (wiadomosci24.pl) gdzie liczba użytkowników jest stosunkowo niewielka (20000) wymagałoby to od bazy MySQL wydajności na poziomie przekraczającym możliwości pojedynczego serwera, a pozostał by i tak problem blokad tabel na czas zapisu. W mongoDB nie ma takiego problemu, zapisy nie blokują odczytów i są wykonywane na pamięci RAM.

* 1. Struktura

MongoDB może być uruchomiona jako pojedyncza instancja serwera dokumentowej bazy danych. Może także zostać skonfigurowana do pracy w systemie rozproszonym na wielu węzłach. Jest to tzw. tryb sharding, umożliwiający skalowanie poziome bazy MongoDB. Pojedyncza instancja, oparta na procesie mongodb, może być skonfigurowana do wersji rozproszonej z dostępem przez kontroler mongos. Wersja rozproszona może zawierać do 1000 węzłów. Przy skalowaniu poziomym, dane są podzielone na wiele serwerów, natomiast z zewnątrz, za pośrednictwem mongos, widzimy je jako jeden serwer. W ramach poszczególnych części dane mogą być także zreplikowane w trybie master-slave, czyli zapis następuje do węzła master, a odczyt jest możliwy z dowolnego węzła. W takim środowisku dzięki użyciu kontrolera mongos, wiele operacji może być wykonywanych równolegle. Są one jednak wykonywane niezależnie od siebie bez blokowania innych operacji. Otrzymujemy dzięki temu dużą wydajność serwera tzn. szybkość przetwarzanych danych

Analogia do SQL:

|  |  |
| --- | --- |
| **MongoDB** | **Baza SQL** |
| Dokument | Wiersz / rekord |
| Kolekcja | Tabela |
| \_id | Klucz główny |
| Zagnieżdżenie | Relacja 1:N |
| Tablica referencji do obiektów | Relacja M:N |
| Indeks | Indeks |

**Nie ma kolumn i tabel - są dokumenty oraz kolekcje:**

* kolekcja to zbiór dokumentów
* dokument składa się z kluczy (pól)
* dokument to jeden rekord
* wiele wartości dla jednego klucza
* szybkie dodawanie kolejnych kluczy nawet dla bardzo rozbudowanych aplikacji
* nie ma potrzeby tworzenia oddzielnych tabel aby normalizować bazę danych
* możemy definiować różne klucze w obrębie jednej kolekcji

// tablica $users to kolekcja, elementy tej tablicy to dokumenty

$users = array(

array('login' => 'a', 'pass' => 'b'),

array('first\_name' => 'a', 'last\_name' => 'b')

);

**Nie ma schematu:**

* jeśli chcesz stworzyć nowy dokument to dodajesz obiekt z danymi do bazy
* jeśli chcesz zmienić pola w dokumencie to:
  + ładujesz dokument
  + dodajesz nowy klucz lub/i edytujesz wartości kluczy
  + zapisujesz