Rozproszone systemy operacyjne

Nazwa projektu „nie wiem jaka jest dokładna nazwa ☺”

Autorzy:

Imię nazwisko

Imię nazwisko

Imię nazwisko

Imię nazwisko

Imię nazwisko

Warszawa, 2013

[1 Wstęp 3](#_Toc352778950)

[1.1 Możliwości MongoDB 3](#_Toc352778951)

[1.2 Wykorzystanie 3](#_Toc352778952)

[1.3 MongoDB podstawowe metody 4](#_Toc352778953)

[find 4](#_Toc352778954)

[findAndModify 4](#_Toc352778955)

[findOne 4](#_Toc352778956)

[Insert 5](#_Toc352778957)

[Save 5](#_Toc352778958)

[Update 5](#_Toc352778959)

[sdfsfswef **Błąd! Nie zdefiniowano zakładki.**](#_Toc352778960)

1. Wstęp

MongoDB jest to otwarty i nierelacyjny system zarządzania bazą danych napisany w języku C++. Prace nad systemem rozpoczęła firma 10gen w 2007 roku. Pierwsze wydanie stabilnej wersji nastąpiło w 2009 roku. MongoDB charakteryzuje się dużą skalowalnością oraz wydajnością. Nie posiada ściśle określonej struktury obsługiwanych baz danych. Dane składowane są jako dokumenty w stylu JSON, co umożliwia aplikacjom bardziej naturalne ich przetwarzanie, przy zachowaniu możliwości tworzenia hierarchio raz indeksowania.

MongoDB charakteryzuje się następującymi właściwościami:

* jednorodne wsparcie dla standardu unicode ,
* obsługa danych w innych kodowaniach w formacie binarnym,
* duża liczba obsługiwanych typów danych,
* obsługa kursorów,
* zapytania ad-hoc,
* zapytania do zagnieżdżonych pól dokumentów,
* indeksowanie,
* wsparcie dla agregacji danych,
* możliwość składowania plików w bazie,
* architektura zaprojektowana z myślą o łatwej replikacji.

Wewnętrznym językiem do definiowania zapytań oraz funkcji agregujących jest [JavaScript](http://pl.wikipedia.org/wiki/JavaScript) wykonywany bezpośrednio przez serwer MongoDB. Interfejsy programistyczne pozwalające obsługiwać bazy MongoDB powstały dla wszystkich wiodących języków programowania, w tym dla C, C++, C#, Javy, PHP, Perla, Pythona, Rubiego.

MongoDB posiada ograniczone wsparcie dla transakcji - ich zasięg jest ograniczony do zmian w pojedynczym dokumencie, aczkolwiek zmiany te mogą być bardzo skomplikowane. Z tego powodu część użytkowników ogranicza zastosowanie MongoDB do niekrytycznych danych informacyjnych, pozostawiając obsługę krytycznych operacji (np. obsługa zamówień w sklepie) relacyjnym bazom danych, gdzie takie ograniczenia nie występują.

1. Cechy MongoDB

W celu utworzenia bazy danych wystarczy polecenie use <nazwa bazy> , baza jest gotowa do użycia. Domyślnie przy użyciu wiersza poleceń ( mongo ) używana jest baza test.

Baza jest w stu procentach dynamiczna. Ma to swoje plusy i minusy. Bazy danych i kolekcje (odpowiedniki tabel) tworzone są dynamicznie – oznacza to, że baza nie krzyczy na nas za literówki. Jeżeli raz napiszemy

db.users.insert({"name": "Student" });

(komenda wstawiająca dokument – odpowiednik rekordu – do kolekcji *user* z polem – kolumną - *name* która ma wartość *“Student”*) to Mongo sprawdzi czy w używanej przez nas bazie jest już kolekcja *users*, jeżeli nie to ją utworzy. Gdy za drugim razem wpiszemy

db.usrs.find();

(komenda listująca wszystkie dokumenty z kolekcji *usrs* ) to Mongo nie nakrzyczy na nas, tylko wyświetli pusty zbiór wyników. Gdy wpiszemy

db.usrs.insert( {"name": "Student" } );

to Mongo stworzy kolekcję *usrs*, wstawi do niej dokument z polem *name* o wartości *Student* i będzie się cieszyć z dobrze wykonanego zadania.

Co ciekawe każdy dokument przy insercie dostaje generowany przez bazę identyfikator. Dotyczy to też sterowników do PHP – po wywołaniu inserta obiekt który dodawaliśmy dostaje automatycznie pole \_id .

* 1. API

find

db.collection.find();

Metoda find jest to odpowiednik selecta w relacyjnych bazach danych. Składnia tej metody została przedstawiona poniżej.

db.products.find( { qty: { $gt: 25 } } );

Powyższy zapis należy rozumień jako: znajdź wszystkie dokumenty o nazwie products w których wartość dla pola gty jest większa niż 25.

findAndModify

db.collection.findAndModify( {

query: <document>,

sort: <document>,

remove: <boolean>,

update: <document>,

new: <boolean>,

fields: <document>,

upsert: <boolean>

} );

Przykład wykorzystania został zamieszczony poniżej.

db.people.findAndModify( {

query: { name: "Tom", state: "active", rating: { $gt: 10 } },

sort: { rating: 1 },

update: { $inc: { score: 1 } }

} )

Powyższy zapis należy rozumień jako: znajdź dokumenty z kolekcji people, gdzie wartość dla pola name jest równa Tom, dla pola state jest równa active, dla pola rating jest większa niż 10. Następnie posortuj rosnąco dokumenty z pierwszego zapytania jeśli takie istnieją i wybierz pierwszy dokument. Następnie zwiększ wartość dla pola score o 1.

findOne

db.collection.findOne();

Powyższe polecenie zwraca tylko jeden dokument spełniający warunek zadany w kwerendzie. Jeśli wiele dokumentów spełnia warunek, wyświetlony zostanie tylko pierwszy dokument zgodnie z porządkiem jaki jest on umieszczony w bazie.

Insert

db.collection.insert(document);

Przykład zastosowania:

db.products.insert( { item: "card", qty: 15 } );

Operacja ta wstawi nowy dokument do kolekcji protucts z polem item z wartością card oraz polem qty z wartością 15, oraz z unikalnym kluczem ObjectID.

Save

Save();

Metoda save aktualizuje istniejący dokument, lub wstawia dokument w zależności od parametru.

Przykład zastosowania:

db.products.save( { item: "book", qty: 40 } );

Jeśli wartość w dokumencie dla pola \_id będzie podana oraz wartość ta będzie już istniała wówczas dokument zostanie zaktualizowany z wyszczególnionymi wartościami dla danych pól. Jeśli w dokumencie nie będzie istniała wartość dla pola \_id wówczas do dokumentu zostaną dodane wyszczególnione pola z ich wartościami.

Update

db.collection.update(query, update[, options]);

Metoda update aktualizuje istniejący dokument, lub dokumenty w kolekcji. Domyślnie update aktualizuje jeden dokument.

Przykład zastosowania:

db.products.update( { item: "book", qty: { $gt: 5 } }, { $set: { x: 6 }, $inc: { y: 5} } );

Metoda ta aktualizuje dokument w kolekcji products, która odpowiada kryterium zapytania i ustawia wartość pola x na 6 oraz zwiększa wartość dla pola y o 5. Wszystkie pozostałe pola w dokumencie pozostają niezmienione.

* 1. Replikacja danych
  2. Bezpieczeństwo