

Rozproszone systemy operacyjne

Szczegółowa koncepcja rozwiązania oraz opis mongos

Autorzy:

Tomasz Adamiec

Piotr Cebulski

Marek Kowalski

Mateusz Rosiewicz

Paweł Sokołowski

Marcin Wnuk

Warszawa, 2013

[Interfejs programu mongod 3](#_Toc356247551)

[1.1 Czym jest mongos 5](#_Toc356247552)

[1.2 Uruchamianie mongos 5](#_Toc356247553)

[1.3 Komendy 6](#_Toc356247554)

Interfejs programu mongod

Celem projektu jest zaimplementowanie bazy danych działających zgodnie z interfejsem MongoDb. Ważnym jest więc zapoznanie się z tym interfejsem i przeprowadzenie wszelkich operacji bazodanowych w oparciu o jego komendy.

Programem odpowiedzialnym za wykonywanie operacji bazodanowych jest mongod. Mongod jest aplikacją nasłuchującą na określonym porcie (domyślnie jest to 27017, ale można go zmienić za pomocą odpowiedniego parametru wywołania programu). Komunikuje się on z klientami za pomocą odpowiednio zdefiniowanego protokołu. Wszelkie typy używane w komunikatach są zgodne z formatem BSON[[1]](#footnote-1). I tak łańcuchy znaków są typu odpowiadającego **cstring** z języka C (kodowane w UTF-8, zakończone zerem), a porządkiem bajtów we wszystkich innych typach jest **little-endian**. W skład tego protokołu wchodzi obecnie 8 różnych rodzajów wiadomości.

Każdy komunikat przesyłany z i wysyłany do **mongod** rozpoczyna się od następującego nagłówka:

struct MsgHeader {  
 int32 messageLength;  
 int32 requestID;  
 int32 responseTo;  
 int32 opCode;  
}

Składa się on z czterech czterobajtowych liczb typu integer. Pierwsza z nich określa długość całej wiadomości ( a więc 16 bajtów nagłówka powiększone o długość komunikatu specyficzną dla jego typu). Kolejną jest **requestID –** jest to identyfikator wiadomości, nadawany przez **mongod** lub też przez jego klienta. Jeżeli wiadomość jest odpowiedzią serwera bazy danych ta sama wartość umieszczana jest w **responseTo**. W pozostałych przypadkach pole to przyjmuje wartość 0. Ostatnim z elementów nagłówka jest **opCode** – jest to wartość określająca typ wiadomości. Może ona przyjmować następujące wartości:

|  |  |  |
| --- | --- | --- |
| opCode | Wartość | Komentarz |
| OP\_REPLY | 1 | Odpowiedź na rządanie klienta. Jako jedyny typ posiada ostawioną wartość responseTo. |
| OP\_MSG | 1000 | Ogólna wiadomość. Po nagłówku występuje ciąg znaków |
| OP\_UPDATE | 2001 | Aktualizacja dokumentu |
| OP\_INSERT | 2002 | Wstawienie nowego dokumentu |
| RESERVED | 2003 | Obecnie nieużywana |
| OP\_QUERY | 2004 | Zapytanie |
| OP\_GET\_MORE | 2005 | Pobiera więcej danych z zapytania |
| OP\_DELETE | 2006 | Usunięcie dokumentu |
| OP\_KILL\_CURSORS | 2007 | Zamknięcie aktualnie otwartego kursora w bazie danych. |

Przykład: OP\_INSERT

Zostanie teraz zaprezentowany jeden z typów komunikatów: **OP\_INSERT[[2]](#footnote-2)**. Struktura takiego komunikatu wygląda następująco:

struct {  
 MsgHeader header;  
 int32 flags;  
 cstring fullCollectionName;  
 document\* documents;   
}

Pierwszym polem **header** jest wcześniej omawiany podstawowy, wspólny dla wszystkich wiadomości nagłówek. Następne pole **flags** jest wektorem bitowym o długości 4 bajtów określającym opcje operacji wstawiania. Aktualnie można ustawić tylko jedną flagę (pierwszy bit) **ContinueOnError –** określa ona czy kontynuować operację wstawiania dokumentów, gdy nie powiodła się ona dla jednego z nich. Kolejne pole **fullCollectionName** zawiera pełną nazwę kolekcji do której wstawiane są dokumenty. Jest to łańcuch znaków typu **cstring**. Ostatnie pole **documents –** zawiera kolekcję wstawianych dokumentów, zakodowanych zgodnie ze standardem BSON.

Zostanie teraz zaprezentowany przykład przedstawiający prostą wiadomość wstawienia dokumentu. Wywołania z powłoki Mongo polecenia:

db.entites.insert({Name: „Tom”})

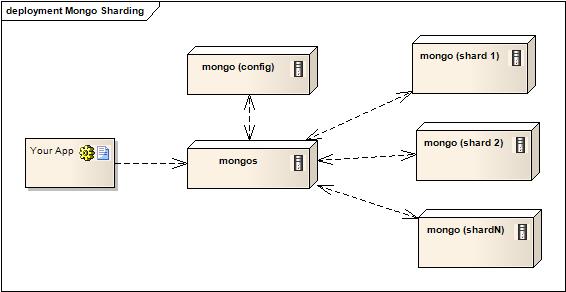
Powoduje wysłania następującego ciągu bajtów (każdy z bajtów zapisany jest szesnastkowo):

46-00-00-00-04-00-00-00-00-00-00-00-D2-07-00-00-00-00-00-00-74-65-73-74-2E-65-6E-74-69-74-69-65-73-00-24-00-00-00-07-5F-69-64-00-51-75-A7-20-41-B6-76-09-20-E2-9A-08-02-4E-61-6D-65-00-04-00-00-00-54-6F-6D-00-00

Pierwsze cztery czwórki bajtów reprezentują nagłówek. 46-00-00-00 = 70 jest długością wiadomości. 04-00-00-00 = 4 jest identyfikatorem wiadomości. Kolejne cztery bajty: 00-00-00-00 = 0 zgodnie z protokołem przyjmują wartość 0, gdyż nie jest to odpowiedź serwera. Ostatnie cztery bajty nagłówka przyjmują wartość: D2-07-00-00 = 2002; wartość ta także jest zgodna z oczekiwaniami – odpowiada on typowi komunikatu: OP\_INSERT. Dalsza część wiadomości odpowiada specyficznym polom dla operacji wstawiania dokumentu. **Flags** przyjmuję wartość 00-00-00-00 = 0, a więc żadna z flag nie została ustawiona. Kolejnym elementem komunikatu jest zakodowana pełna nazwa kolekcji: 74-65-73-74-2E-65-6E-74-69-74-69-65-73-00 = „test.entities”. Pozostałe bajty: 24-00-00-00-07-5F-69-64-00-51-75-A7-20-41-B6-76-09-20-E2-9A-08-02-4E-61-6D-65-00-04-00-00-00-54-6F-6D-00-00 reprezentują dokument w postaci BSON.

* 1. Czym jest mongos

Mongos jest to usługą swoistego routingu dla MongoDB. Usługa ta działa na warstwie aplikacji oraz określa ona lokalizacje dla danych w klastrze. W celu lepszego zobrazowania problem przedstawiony jest na ilustracji zamieszczonej poniżej.



* 1. Uruchamianie mongos

Po poprawnym zainstalowaniu i uruchomieniu MongoDB można uruchomić usługę mongos. Jednak by to zrobić należy zdefiniować ścieżki dostępu przedstawione poniżej:

> sudo mkdir -p /db/data/config  
> sudo mkdir -p /db/data/shard1  
> sudo mkdir -p /db/data/shard2

Pierwsza z powyższych komend wskazuje na położenie serwera z konfiguracją z której będzie korzystał mongos. Pozostałe komendy utworzą instancję dla shardingu.

Kolejnym krokiem jest uruchomienie serwera konfiguracyjnego:

> sudo mongod --dbpath "/db/data/config" --port 10381 –configsvr

Powyższa komenda utworzy i uruchomi demona konfiguracyjnego MongoDB. W tym przypadku będzie to przykładowo port 10381, parametr configsvr pozwoli MongoDB zidentyfikować instancję jako serwer konfiguracyjny.

Kiedy powyższe kroki zostaną wykonane można uruchomić usługę mongos komendą:

> sudo mongos --configdb localhost:10381 --port 10382 --chunkSize 1

Komenda ta utworzyła usługę mongos która korzysta z serwera konfiguracyjnego configdb oraz nasłuchuje na porcie z numerem 10382. chunkSize określa maksymalną wielkość danych wyrażoną w megabajtach.

Kolejnym krokiem jest utworzenie shard boxes zawierających nasze dane.

> sudo mongod --port 10383 --dbpath /db/data/shard1 --shardsvr  
> sudo mongod --port 10384 --dbpath /db/data/shard2 –shardsvr

Powyższa komenda utworzyła dwie instancję shardów dla MongoDB. Obie instancję otrzymały unikalne numery portów oraz zostały przypisane do flagi shardsvr.

Kolejnym krokiem jest dodanie shard servers do mongos.

> mongo localhost:100382  
> use admin  
> db.runCommand({addshard: "localhost:10383", allowLocal: true})  
> db.runCommand({addshard: "localhost:10384", allowLocal: true})

Pierwsza z komend łączy instancję mongos. Druga komenda przełącza w tryb admin w celu możliwości wykonania kolejnych komend. Kolejne komendy dodają shardy w raz z określeniem ich portów. W celu dodania większej ilość shardów po prostu należy wywołać komendę wielokrotnie z uwzględnieniem unikalnego numeru portu dla poszczególnych shardów.

Po wykonaniu powyższych kroków środowisko gotowe jest do przyjmowania danych.

* 1. Komendy

--help, -h

Zwraca podstawową pomoc.

--version

Pokazuje wersje mongod.

--config <filename>, -f <filename>

Określa plik konfiguracyjny który może zostać użyty w celu załadowania ustawień dla mongos

--verbose, -v

Zwieksza ilość sprawozdawczości wewnętrznej dla standardowego wyjścia lub w pliku z logami określonego w [--logpath](http://docs.mongodb.org/manual/reference/mongos/#cmdoption-mongos--logpath).

--quiet

Uruchamia instancję mongos w trybie quite, który ogranicza ruch generowany na wyjściu.

--port <port>

Określa port TCP dla mongos na którym jest prowadzony nasłuch dla klientów, którzy się łączą. Domyślnym portem jest port numer 27017.

--bind\_ip <ip address>

Określa adres IP interfejsu na którym mongos będzie nasłuchiwał połączeń. Domyślnie mongos nasłuchuje na wszystkich interfejsach. Można to zmienić, jednak podczas dodawania nowych interfejsów należy upewnić się że zostały przeprowadzone kroki mające na celu zapewnienie bezpieczeństwa dla integralności bazy danych.

--maxConns <number>

Określa maksymalną ilość jednoczesnych połączeń, które zostaną przyjęte przez mongos. Ustawienie to nie ma wpływu jeśli wartość tego parametru jest wyższa niż systemu operacyjnego na którym uruchomiony jest mongos dla maksymalnego progu śledzenia połączeń. Wartość nie może być większa niż 20000.

--objcheck

Wymusza na mongos sprawdzenie poprawności wszystkich zapytań otrzymywanych od klientów. Chroni przed wprowadzaniem niepoprawnych obiektów do bazy danych. Opcja ta ma wpływ na wydajność i jest domyślnie wyłączona.

--logpath

Określa ścieżkę do pliku z logami

--logappend

Zapisuje logi w dzienniku na końcu pliku nie nadpisując zawartości po restarcie mongos.

--syslog

Wysyła wszystkie logi do systemu Syslog.

--pidfilepath <path>

Określa położenie pliku w którym składowane są informację na temat „PID”, id procesow

--keyFile <file>

Określa ścieżkę do pliku w którym przechowywany jest klucz w celu autentykacji połączenia pomiędzy mongos, a klastrami shardów.

--nounixsocke

Wyłącza nasłuch na socketach UNIX.

--unixSocketPrefix <path>

Określa ścieżkę do socketa UNIX

--configdb <config1>,<config2><:port>,<config3>

Określa konfigurację bazy danych.

--test

Opcja ta wykonywania wewnętrznych testów jednostkowych.

--upgrade

Opcja ta aktualizuje meta dane wykorzystywane przez konfigurację bazy danych.

--ipv6

Uruchamia wsparcie dla IPv6. Domyśle funkcjonalność ta jest wyłączona.

--jsonp

Zezwala JSONP na dostęp poprzez interfejs http.

--noscripting

Wyłącza silnik skryptowy.

--nohttpinterface

Wyłącza interfejs http

--localThreshold

Wpływa na logikę działania mongos podczas wyboru członków replikacji

--noAutoSplit

Zapobiega automatycznemu wstawiania meta danych do kolekcji podczas procesu shardingu.

1. : <http://bsonspec.org/#/specification> [↑](#footnote-ref-1)
2. Po dokładny opis reszty odsyłam do: <http://docs.mongodb.org/meta-driver/latest/legacy/mongodb-wire-protocol/>. [↑](#footnote-ref-2)