Rozproszone systemy operacyjne

Szczegółowa koncepcja rozwiązania

Autorzy:

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

Warszawa, 2013

[Interfejs programu mongod 4](#_Toc356056626)

[Przykład: OP\_INSERT 4](#_Toc356056627)

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. : <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)