

УДК 004.421

Самедов Р.Б.Бакинский Государственный Университет, Баку, Азербайджан
ramin.samedov@gmail.com**АЛГОРИТМ МИГРАЦИИ РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ ORACLE 12С В
ОБЛАЧНУЮ СРЕДУ NOSQL MONGODB**

Большие веб-проекты строятся в облачных средах. Однако стандартные реляционные базы данных обладают большим количеством ограничений, которые не позволяют в полной степени воспользоваться облачной средой. В статье указан ряд недостатков реляционной базы данных. В качестве используемой альтернативы показано использование NOSQL базы данных MongoDB. В статье даны алгоритм и программа, при помощи которой можно перейти из стандартной реляционной базы данных в NOSQL базу данных MongoDB.

Ключевые слова: облачные вычисления, базы данных, MongoDB, NOSQL, миграция данных.

Введение

Системы управления реляционными базами данных (СУБД) на протяжении длительного времени рассматривались как универсальное решение в области баз данных. Реляционные базы данных достигли высокого уровня зрелости в результате обширных исследований и породили большой успешный рынок и множество решений для различных областей бизнеса. Однако постоянно растущая потребность в масштабируемости и новых приложениях породила новые проблемы для традиционных реляционных СУБД, в том числе определенную неудовлетворенность результатами применения такого универсального решения в некоторых приложениях облачных сред. Для больших веб-проектов, построенных как облачные среды, таких как eBay, Amazon, Twitter и Facebook, масштабируемость и высокая доступность являются важнейшим требованием, которое должно соблюдаться в обязательном порядке. Для этих приложений даже малейшая остановка может иметь существенные финансовые последствия и отрицательно влиять на доверие конечных пользователей. Масштабируемость является главной проблемой реляционных баз данных. Так как все больше приложений работают в условиях высокой нагрузки, например таких, как веб-сервисы, их требования к масштабируемости могут очень быстро меняться и сильно расти. Реляционные базы данных хорошо масштабируются только в том случае, если располагаются на единственном сервере. В облачных средах база данных должна располагаться на нескольких десятках серверов – в итоге проблема масштабированности сильно ощущается [1].

Быстрое распространение основанных на Интернете сервисов, таких, как электронная почта, блоги, социальные сети, поисковые механизмы и электронная коммерция, существенно изменило поведение и склонности веб-пользователей в отношении создания, передачи и получения контента, обмена информацией и покупки товаров. ИТ-специалисты отмечают быстрый рост масштабов генерируемых и потребляемых данных в результате распространения вышеупомянутых систем; постоянно растущая потребность в масштабируемости и новые требования к приложениям породили новые проблемы для традиционных систем управления реляционными базами данных. Здесь на сцену выходит недорогое и высокопроизводительное программное обеспечение баз данных типа NoSQL.

База данных типа NoSQL как решение для больших веб-проектов, построенных в облачных средах

База данных типа NoSQL подчиняется известной теореме CAP (Consistency, Availability, Tolerance to Partitions – Согласованность, Доступность, Устойчивость к Разделению). Большинство проектов, построенных в облачных средах, согласно пожертвовать требованием строгой согласованности, а именно в этих системах применяется политика ослабленной согласованности под названием «согласованность в конечном счете» (eventual consistency), которая гарантирует, что если к реплицированному объекту не будут применяться никакие новые обновления, то в конечном счете каждое обращение к этому объекту будет возвращать последнее обновленное значение. В отсутствие ошибок максимальный размер окна несогласованности может быть определен на основе таких факторов, как коммуникационные задержки, нагрузка на систему и количество копий, затрагиваемых схемой репликации.

NoSQL базы данных обладают следующими особенностями:

- Масштабируемость: возможность горизонтального масштабирования пропускной способности с охватом множества серверов. Система в состоянии поддерживать очень большие базы данных с очень высокой частотой запросов при очень низкой задержке.
- Эластичность: простой интерфейс/протокол уровня вызова (в отличие от SQL-связывания). При этом система в состоянии удовлетворять меняющиеся требования к приложениям в обоих направлениях, как в увеличении масштаба, так и в случае уменьшения масштаба.
- Оптимизация маршрута доступа: поддержка более слабых моделей согласованности, чем ACID-транзакции в большинстве традиционных реляционных СУБД, что позволяет выполнять высокую интенсивность операций чтения, а также операций записи.
- Производительность: эффективное использование распределенных индексов и оперативной памяти для хранения данных.
- Динамичность: возможность динамического описания новых атрибутов или схемы данных [2].

Основные особенности NOSQL базы данных MongoDB

Очень актуально создать механизм миграции из реляционных баз данных в NOSQL базу данных, учитывая, что большинство уже существующих приложений написано для реляционных баз данных. Для начала дадим шесть основных концепций NOSQL базы данных на примере одной из самых распространенных баз данных MongoDB:

1. MongoDB – состоит из ноля или более баз данных, каждая из которых является контейнером для прочих сущностей. Все данные хранятся в базе данных.
2. База данных может иметь ноля или более «коллекций». Коллекция сильно напоминает традиционную «таблицу». Таким образом справедливо сравнение реляционной таблицы с «коллекцией».
3. «Коллекции» состоят из ноля или более «документов». Документ можно рассматривать как «строку» в реляционной базе данных.
4. Документ состоит из одного или более «полей», подобно «столбцам» в реляционной базе данных.
5. «Индексы» в MongoDB почти идентичны таковым в реляционных базах данных.
6. «Курсоры» отличаются от предыдущих пяти концепций – они очень важны, так как, запрашивая у MongoDB какие-либо данные, база данных возвращает курсор, с которым возможно сделать все что угодно – подсчитывать, пропускать

определенное число предшествующих записей, но при этом не загружая сами данные [3].

Таким образом для миграции из реляционной базы данных Oracle 12C в NOSQL базу данных MongoDB необходимы первые пять концепций, то есть все, за исключением курсоров, так как курсоры будут использоваться исключительно после миграции.

Алгоритм, при помощи которого возможно выполнять миграцию данных из реляционной базы данных в NOSQL базу данных

Шаг 1: В существующей схеме HR базы данных Oracle 12C создаются две пустые таблицы EMP и DEPT. В последующих шагах эти таблицы будут заполнены данными и перенесены в NOSQL базу данных MongoDB. Программа скрипта создания указанных таблиц следующая:

```
CREATE TABLE emp (
  empno decimal(4,0) NOT NULL,
  ename varchar(10) default NULL,
  job varchar(9) default NULL,
  mgr decimal(4,0) default NULL,
  hiredate date default NULL,
  sal decimal(7,2) default NULL,
  comm decimal(7,2) default NULL,
  deptno decimal(2,0) default NULL
);
CREATE TABLE dept (
  deptno decimal(2,0) default NULL,
  dname varchar(14) default NULL,
  loc varchar(13) default NULL
);
```

В результате получаются две таблицы, как показано на рисунке 1 и на рисунке 2, где поле Column Name – имя столбца, ID – порядковый номер, Index Pos – позиция индекса, Null? может принимать значение столбца Null, Data Type – тип данных столбца.

Шаг 2: Таблицы EMP и DEPT заполняются тестовыми данными. Программа заполнения таблиц данными:

```
INSERT INTO emp VALUES ('7369','SMITH','CLERK','7902','1980-12-17','800.00',NULL,'20');
INSERT INTO emp VALUES ('7499','ALLEN','SALESMAN','7698','1981-02-20','1600.00','300.00','30');
INSERT INTO emp VALUES ('7521','WARD','SALESMAN','7698','1981-02-22','1250.00','500.00','30');
INSERT INTO emp VALUES ('7566','JONES','MANAGER','7839','1981-04-02','2975.00',NULL,'20');
INSERT INTO dept VALUES ('10','ACCOUNTING','NEW YORK');
INSERT INTO dept VALUES ('20','RESEARCH','DALLAS');
INSERT INTO dept VALUES ('30','SALES','CHICAGO');
INSERT INTO dept VALUES ('40','OPERATIONS','BOSTON');
Commit;
```

Column Name	ID	PK	Index Pos	Null?	Data Type
DEPTNO	1	1	1	N	NUMBER (2)
DNAME	2			Y	VARCHAR2 (14)
LOC	3			Y	VARCHAR2 (13)

Рис.1. Структура таблицы DEPT

Column Name	ID	PK	Index Pos	Null?	Data Type
EMPNO	1	1	1	N	NUMBER (4)
ENAME	2			Y	VARCHAR2 (10)
JOB	3			Y	VARCHAR2 (9)
MGR	4			Y	NUMBER (4)
HIREDATE	5			Y	DATE
SAL	6			Y	NUMBER (7,2)
COMM	7			Y	NUMBER (7,2)
DEPTNO	8			Y	NUMBER (2)

Рис.2. Структура таблицы EMP

Шаг 3: создается пустая схема в NOSQL базе данных MongoDB с таким же именем, что и в реляционной базе данных. После подключения к базе данных MongoDB скрипт создания схемы следующий:

```
C:\Users\ramin>mongo.exe
MongoDB shell version: 2.6.3
connecting to: test
```

```
При помощи команды use создается схема SCOTT
> use SCOTT
switched to db SCOTT
```

Шаг 4: создается коллекция на основе таблиц реляционной базы данных, которые необходимо перенести. Создается ровно одна коллекция, идентичная таблицам EMP и DEPT, так как база данных типа NOSQL работает иначе с коллекциями, нежели реляционные базы данных с таблицами. [3] Программа следующая:

```
> db.runCommand( { create: "employees", capped: true, size: 64 * 1024 } )
{ "ok" : 1 }
```

Шаг 5: в реляционной базе данных пишется и запускается SQL скрипт, при помощи которого получают документы для переноса непосредственно самих данных. Создаются

UOT 004.421

Səmədov Ramin B.

Bakı Dövlət Universiteti, Bakı, Azərbaycan

ramin.samedov@gmail.com

Oracle 12C relyasiya verilənlər bazasının NoSQL MongoDB hesablama bulud mühitinə miqrasiyası algoritmi

İri veb-proyektlər hesablama buludlarında yaradılır. Lakin standart relyasiya verilənlər bazalarında bir neçə çatmazlıqlar mövcuddur və bu çatmazlıqlar hesablama bulud mühitindən tam istifadə etmək imkanını vermir. Bu məqələdə çatmazlıqlar haqqında məlumat verilib. Alternativ həll kimi NOSQL MongoDB verilənlər bazasından istifadəsi məsləhət görülüb. Məqələdə Oracle 12C relyasiya verilənlər bazasının cədvəllərinin, məlumatlarının, sxemlərinin hesablama bulud mühitində yaradılmış NoSQL MongoDB verilənlər bazasına miqrasiyası algoritmi və proqramı göstərilmişdi.

Açar sözlər: hesablama buludları, verilənlər bazası, MongoDB, NOSQL, məlumatların miqrasiyası.

Ramin B. Samadov

Baku State University, Baku, Azerbaijan

ramin.samedov@gmail.com

The algorithm for migrating relational database Oracle 12C into a cloud database NOSQL MongoDB

Large web projects are built in a cloud environment. However, standard relational database has a large number of restrictions that do not allow to use the full extent of the cloud environment. This paper identifies a number of shortcomings of relational databases. NOSQL database MongoDB is shown as an alternative to such databases. The article presents an algorithm and a program which can help you to switch easily from standard relational database to NOSQL database MongoDB.

Keywords: Cloud computing, databases, MongoDB, NOSQL, data migration.