Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Кафедра системотехніки

Дисципліна: «СПЕЦІАЛІЗОВАНІ БАЗИ ДАНИХ (NOSQL)»

**ПРАКТИЧНЕ ЗАНЯТТЯ № 6**

**«ВИВЧЕННЯ КОМАНД АГРЕГАЦІЇ ДАНИХ В СУБД MONGODB»**

|  |  |  |
| --- | --- | --- |
| Виконав:  ст. гр. ІТКН-18-2  Гуляев Д.С. |  | Прийняв:  Доц. каф. Хряпкін О.В. |

Харків 2020

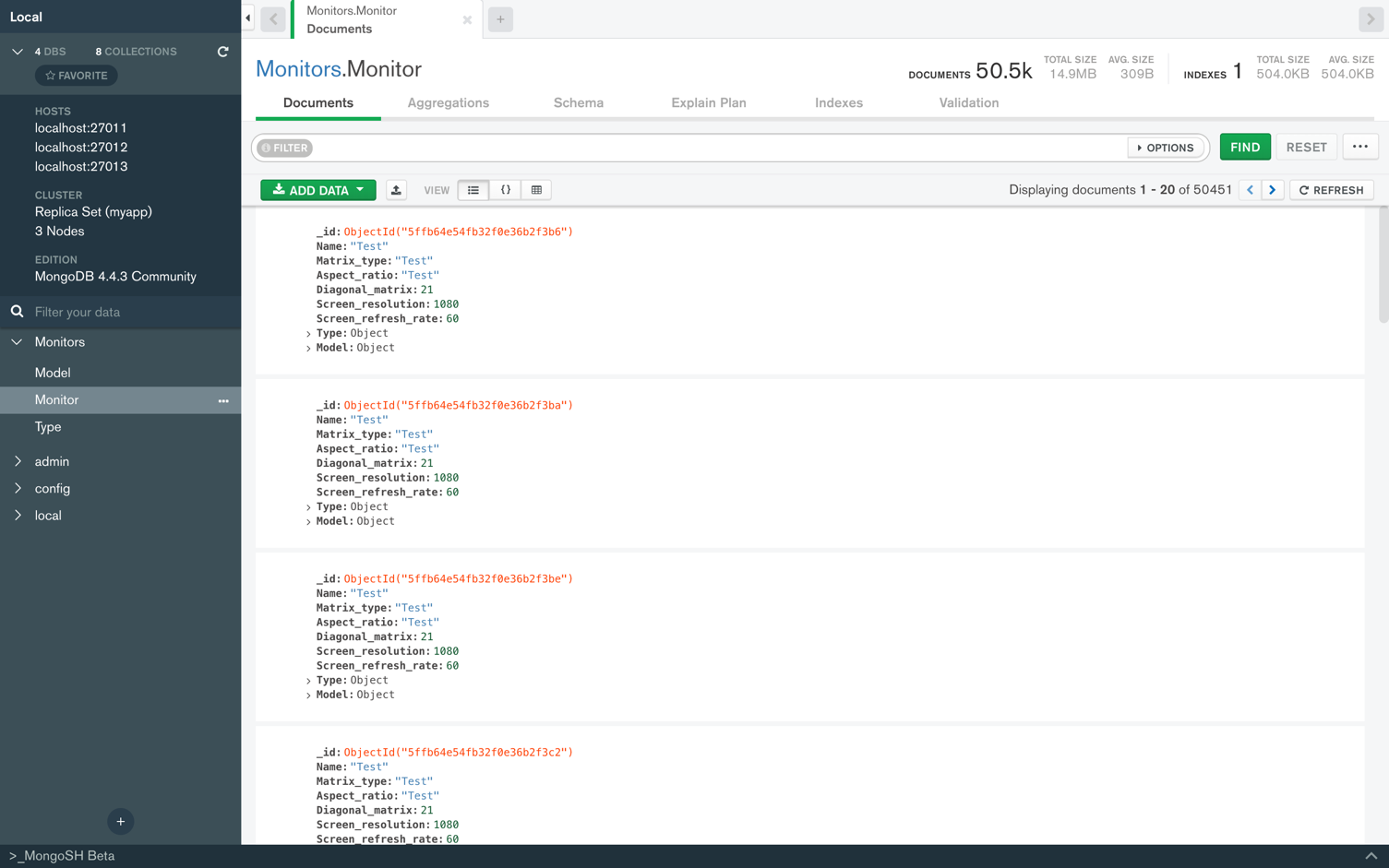
**1 Вивчення команд агрегації даних в СУБД MongoDB**

* 1. Мета заняття

Розробити програмний засіб, що виконує операції агрегування даних з використанням можливостей Aggregation Framework СУБД MongoDB. Отримати практичні навички виконання основних агрегують запитів до БД.

* 1. Хід роботи

Кількість даних – >500000

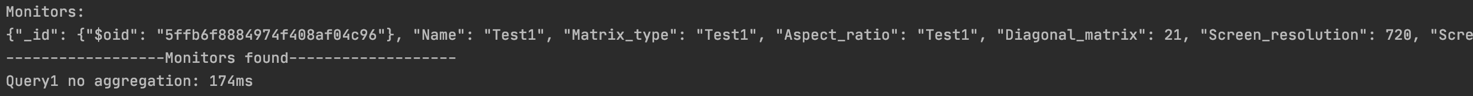


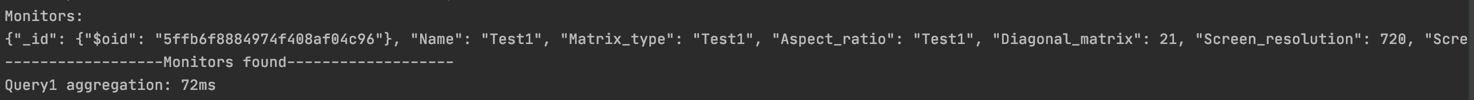
**Код класу Agregation**. В даному класі реалізовані запити як з використанням Aggregation Framework, так і без нього

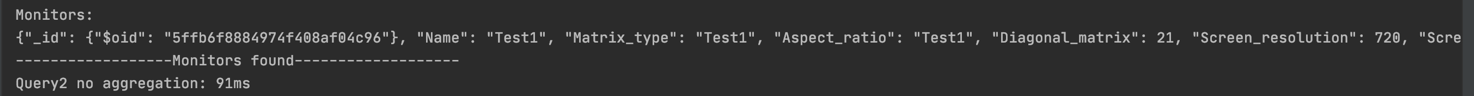
package Aggregation;  
  
import com.mongodb.client.MongoClient;  
import com.mongodb.client.MongoClients;  
import com.mongodb.client.MongoCollection;  
import com.mongodb.client.MongoDatabase;  
import org.bson.Document;  
import org.bson.conversions.Bson;  
  
import java.util.ArrayList;  
import java.util.Arrays;  
import java.util.List;  
  
import static com.mongodb.client.model.Accumulators.*push*;  
import static com.mongodb.client.model.Accumulators.*sum*;  
import static com.mongodb.client.model.Aggregates.\*;  
import static com.mongodb.client.model.Filters.*eq*;  
import static com.mongodb.client.model.Indexes.*descending*;  
import static com.mongodb.client.model.Projections.\*;  
  
public class Aggregation {  
  
 public void testTimeAggregation(){  
 //QUERY 1 NO Aggregation  
 try (MongoClient mongoClient = MongoClients.*create*("mongodb://localhost:27017,localhost:27011,localhost:27012,localhost:27013/?replicaSet=myapp&retryWrites=true")) {  
 MongoDatabase DB = mongoClient.getDatabase("Monitors");  
 MongoCollection<Document> Collection = DB.getCollection("Monitor");  
  
 long startTime = System.*currentTimeMillis*();  
 List<Document> monitorList = Collection.find(new Document("Screen\_refresh\_rate", 75)).into(new ArrayList<>());  
 long endTime = System.*currentTimeMillis*();  
 System.*out*.println("Monitors: ");  
 for (Document monitor : monitorList) {  
 System.*out*.println(monitor.toJson());  
 }  
 System.*out*.println("------------------Monitors found-------------------");  
 System.*out*.println("Query1 no aggregation: " + (endTime - startTime) + "ms");  
 }  
 catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 //QUERY 1 Aggregation  
 try (MongoClient mongoClient = MongoClients.*create*("mongodb://localhost:27017,localhost:27011,localhost:27012,localhost:27013/?replicaSet=myapp&retryWrites=true")) {  
 MongoDatabase DB = mongoClient.getDatabase("Monitors");  
 MongoCollection<Document> Collection2 = DB.getCollection("Monitor");  
  
 Bson match = *match*(*eq*("Screen\_refresh\_rate", 75));  
 long startTime = System.*currentTimeMillis*();  
 List<Document> monitorList2 = Collection2.aggregate(Arrays.*asList*(match)).into(new ArrayList<>());  
 long endTime = System.*currentTimeMillis*();  
 System.*out*.println("Monitors: ");  
 for (Document monitor : monitorList2) {  
 System.*out*.println(monitor.toJson());  
 }  
 System.*out*.println("------------------Monitors found-------------------");  
 System.*out*.println("Query1 aggregation: " + (endTime - startTime) + "ms");  
  
 }  
 catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 //QUERY 2 NO Aggregation  
 try (MongoClient mongoClient = MongoClients.*create*("mongodb://localhost:27017,localhost:27011,localhost:27012,localhost:27013/?replicaSet=myapp&retryWrites=true")) {  
 MongoDatabase DB = mongoClient.getDatabase("Monitors");  
 MongoCollection<Document> Collection3 = DB.getCollection("Monitor");  
  
 long startTime = System.*currentTimeMillis*();  
 List<Document> monitorList = Collection3.find(new Document("Test", 60).append("Screen\_resolution", 720)).into(new ArrayList<>());  
 long endTime = System.*currentTimeMillis*();  
 System.*out*.println("Monitors: ");  
 for (Document monitor : monitorList) {  
 System.*out*.println(monitor.toJson());  
 }  
 System.*out*.println("------------------Monitors found-------------------");  
 System.*out*.println("Query2 no aggregation: " + (endTime - startTime) + "ms");  
 }  
 catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 //QUERY 2 Aggregation  
 try (MongoClient mongoClient = MongoClients.*create*("mongodb://localhost:27017,localhost:27011,localhost:27012,localhost:27013/?replicaSet=myapp&retryWrites=true")) {  
 MongoDatabase DB = mongoClient.getDatabase("Monitors");  
 MongoCollection<Document> Collection4 = DB.getCollection("Monitor");  
  
 Bson match = *match*(*eq*("Test", 60));  
 Bson match2 = *match*(*eq*("Screen\_resolution", 720));  
 long startTime = System.*currentTimeMillis*();  
 List<Document> monitorList2 = Collection4.aggregate(Arrays.*asList*(match)).into(new ArrayList<>());  
 long endTime = System.*currentTimeMillis*();  
 System.*out*.println("Monitors: ");  
 for (Document monitor : monitorList2) {  
 System.*out*.println(monitor.toJson());  
 }  
 System.*out*.println("------------------Monitors found-------------------");  
 System.*out*.println("Query2 aggregation: " + (endTime - startTime) + "ms");  
  
 }  
 catch (Exception e) {  
 e.printStackTrace();  
 }  
  
// //QUERY 3 NO Aggregation  
// try (MongoClient mongoClient = MongoClients.create("mongodb://localhost:27017,localhost:27011,localhost:27012,localhost:27013/?replicaSet=myapp&retryWrites=true")) {  
// MongoDatabase DB = mongoClient.getDatabase("Monitors");  
// MongoCollection<Document> Collection4 = DB.getCollection("Monitor");  
//  
// long startTime = System.currentTimeMillis();  
// List<Document> monitorList = Collection4.find(new Document("Test", 60).append("Screen\_resolution", 720)).into(new ArrayList<>());  
// long endTime = System.currentTimeMillis();  
// System.out.println("Monitors: ");  
// for (Document monitor : monitorList) {  
// System.out.println(monitor.toJson());  
// }  
// System.out.println("------------------Monitors found-------------------");  
// System.out.println("Query3 no aggregation: " + (endTime - startTime) + "ms");  
// }  
// catch (Exception e) {  
// e.printStackTrace();  
// }  
 //*TODO create query 3 no aggregation* //QUERY 3 Aggregation  
 try (MongoClient mongoClient = MongoClients.*create*("mongodb://localhost:27017,localhost:27011,localhost:27012,localhost:27013/?replicaSet=myapp&retryWrites=true")) {  
 MongoDatabase DB = mongoClient.getDatabase("Monitors");  
 MongoCollection<Document> Collection5 = DB.getCollection("Monitor");  
  
 Bson match = *match*(*eq*("Screen\_refresh\_rate", 80));  
 Bson sort = *sort*(*descending*("Diagonal\_matrix"));  
 Bson group = *group*("$Diagonal\_matrix", *sum*("count", 1L), *push*("Diagonal\_matrix", "$Diagonal\_matrix"));  
 Bson project = *project*(*fields*(*excludeId*(), *computed*("tag", "$\_id"), *include*("count", "Diagonal\_matrix")));  
 Bson limit = *limit*(3);  
 long startTime = System.*currentTimeMillis*();  
 List<Document> monitorList2 = Collection5.aggregate(Arrays.*asList*(match, group, project, sort, limit)).into(new ArrayList<>());  
 long endTime = System.*currentTimeMillis*();  
 System.*out*.println("Monitors: ");  
 for (Document monitor : monitorList2) {  
 System.*out*.println(monitor.toJson());  
 }  
 System.*out*.println("------------------Monitors found-------------------");  
 System.*out*.println("Query3 aggregation: " + (endTime - startTime) + "ms");  
  
 }  
 catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 }  
}

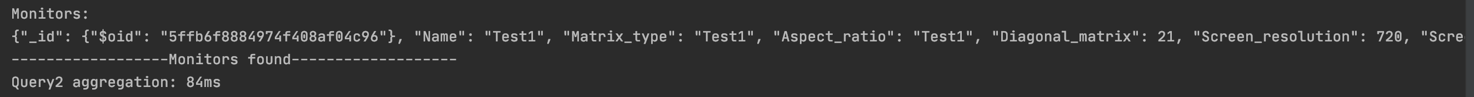
//Aggregation  
new Aggregation().testTimeAggregation();

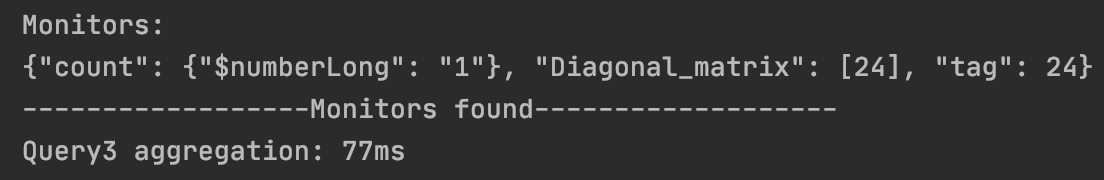
Результати роботи програми











Графіки залежності виконання запитів від часу

**Висновки**

Як можемо бачити, використання Aggregation Framework дає виграш в часі обробці великих даних, а також спрощує програмний код.

Під час виконання даного практичного заняття було отримано навички з використання Aggregation Framework , та отримані навички з використання основних агрегованих функцій для СУБД MongoDB.