The Data Management and Real-time Search Based on Elasticsearch

XiafeiLei[1, a], ZheWang[2, b],yuzhenHe[3, c]

[1]Department of Computer Science and Technology, Jilin University, Changchun, 130000, China

[2]Department of Computer Science and Technology, Jilin University, Changchun, 130000, China

[3]WHY-E Science and Technology Co.,Ltd, Changchun,130000,China

[a]email:leixiafei1207@163.com, [b]email:wz2000@jlu.edu.cn,[c]email:heyuzhen@why-e.com.cn

Keywords: Elasticsearch, data management, real-time search, Kibana

Abstract. At present most of the databases are limited in extracting data, such as full-text search, fail to searching synonyms or correlation.The most important, it can't do real-time data processing. Elasticsearch is a open source search and analysis engine with highly extensible property.We can dig deeper into the data easily by amplifing and narrowing the range of search and analysis in real-time.This article is based on analyzing information statically such as the classes and methods of jar package in a Java project, and the static information is saved in Elaticsearch to support custom search.The process realized double-efficacy of Elasticearch, namely used as a database and a search engine, and combined with Kibana (the browser-based dashboard of Elasticsearch analysis and search) perfectly to manage data and visual search results.

Introduction

The world has been submerged by big data, the existing technology has been committed to storing big data, and be structured data warehouse carrying large amounts of information. Traditional relational databases will lead to performance degradation with setting cluster to process big data; Hbase cannot secondary indexes; the traditional lucene is not suitable for big data and cloud computing platform[1]. Elasticsearch is a distributed,extensible,and real-time search engine which can deal with large-scale data at high rate. It not only support a simple full-text search,also structured search, statistics, query filtering and geo-location,etc[2].Wikipedia uses Elasticsearch do full-text search and highlight keywords; Goldman Sachs uses it to handle 5 TB data index every day[1].The examples above shows that Elasticsearch geared to the needs of big data, it can be deployed to tens of thousands of servers to process PB levels of data, and also can run on a personal laptop .what's more,the developer set up a good default values for various options which is more convenient to new users. Because the objective conditions during the period of school, in this paper, the experimental data is not up to PB level, only in view of the common data real-time searching and management.This paper structure is as follows: the first part is the introduction; The second part is Elasticsarch principle introduction, including the concept, characteristic and processing of index; The third part is the experimental section, including environmental deployment and experiment process; Finally for the presentation and conclusion of experimental results.

Introduction of Elasticsearch Principle

Concepts and Features of Elaticsearch:

Elasticsearch (hereinafter referred to as ES) is an open source,distributed,and real-time search engine, which support for cloud services[3]. It is based on Lucene libraries to create and contribute to combining the independent function of full-text search,statistics,the distributed database system into a coherent,real-time processing as a whole, which has location-based services, search suggestions based on context, and the ability to search fragment (Snippet), besides provides a flexible query [4].Its characteristic is high availability, high extensible,distributed and almost in real time, it provide complete RESTful API to communicate with ES (and of course can be realized through the Java API).

The cluster,node and shard have to be mentioned in ES,the relation is as follow：data is stored in the shard, shard is assigned to a node, the node belong to the cluster. The storage principle of ES is to use shards to save index, at the same time it can set the number of replica. The cluster allows to has more nodes to distribute the load,any a new node to be able to add into ES cluster quickly and automatically[3][4]. By contrast, ES owns distributed function, which means don't need to change the program to achieve horizontal scalability.At the same time when a certain node break down, the rest of the nodes to ensure continuous and stable operation of the retrieval service after removal of the node, schematic diagram is in Fig1.

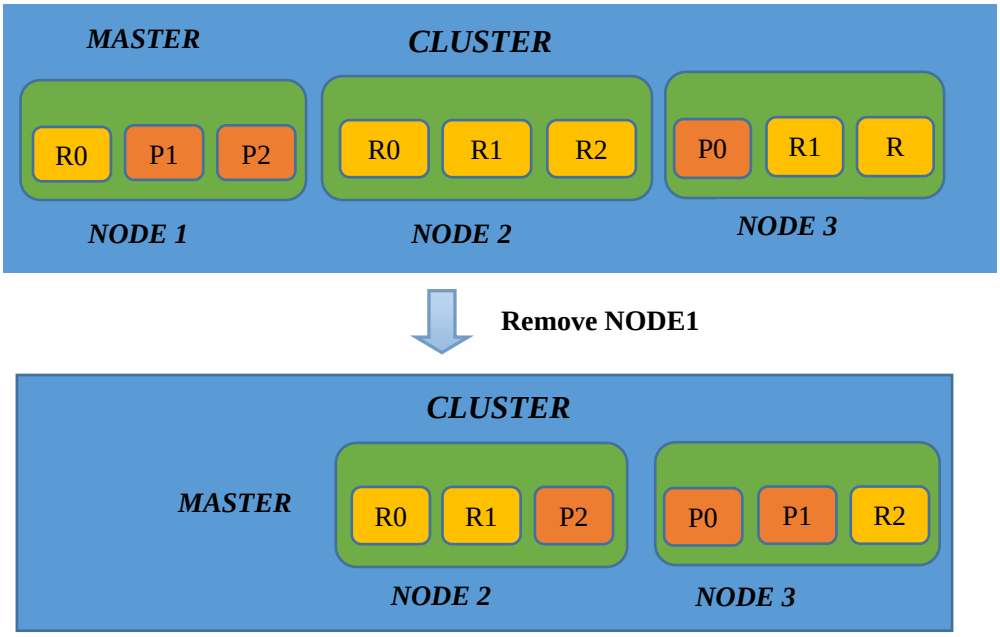

Fig1.Schematic Diagram

As can be seen from the schematic diagram of the ES, as "database",the indexes are stored in different shards of ES, The biggest difference between ES and the relational database is that the query results are sorted by correlation, that means the most accurate results in the front,but it compared with traditional database has a certain relevance, in Table I shows corresponding relation of ES and traditional relational database.

Table I.The corresponding relation of ES VS traditional relational database.

Database Name	**Corresponding Elements Name**			
Relational database	Database	Table	Rows	Columns
Elasticsearch	Index	Type	Document	Fields

Indexing process of ES

ES uses a JSON document as serialization format , so need build a new index to store them.The following for ES indexing process:

(1) Create indexes and build an the mapping of index by mapping fields.

(2) Converts the data to a JSON file (the author makes conversion use ES java API).

(3) Put the JSON file data into ES.

(4) Specified _index + _type + _id of the index(uniquely determining index).

(5) Through the CURL command or RESTClient (the author uses the plug-in) sends the request to any node of the cluster, then passed to the Master node.

(6) ES according to the _id ,_type calculate the storing shard by hash modulus.

Shard=hash (routing) % number_of_primary_shards (the routing here id).....(1)[5].

(7) Find the shard Master, judge index operation type (PUT, GET, POST, DELETE), due to ES change version for every operation(the default is increasing), so if id exist, update version; If not exist, create;find consistent with the request of the mapping and parses the JSON.

(8) Master shard send the request to the corresponding replica to index.

(9) Corresponding shard return the query results.

The experiment part

- **Experiment content:**

This experiment is based on the background of the jars package of a Java application execution. I want to parse this jar package containing the information such as the classes,the method of the class, and method called sequence staticly.The static information is stored in ES, and then do customize query and aggregation query, achieving the effect of the ES.Finally, visualizing query results by Kibana[6].

- **Procedures:**

Premise: Install the latest version of Java and Elasticsearch, and two versions are consistent; Install kibana 4.0.1;The application will be packaged to jar.

Step:

1.Deploy the environment, start Elasticsearch by CMD

2.Mapping index(called utf_test) in the program, including field "_index", "type", "id", "ClassName","MethodName","ClassAndMethodHashcode","TraceMethodVisitorHashcode", "createDate".

3.Through the java API of ES, put the static information into ES(The process is impleted by the program), temporarily make ES as database.

4. Start Kibana by CMD and do custom query(this experiment only three queries).

(1) Paragraph query : ClassAndMethodHashcode = 1734536807;

(2)Custom query: method begin with the “get” AND ClassAndMethod Hashcode = 1734536807;

(3)Aggregation query: statistical field “MethodName “Top15 and “ClassNameAnd Method “Top30 (the number of experimental data is 10786) .

- **Results:**

Experiment 1:

Paragraph query:To match the records including ClassAndMethod Hashcode = 1734536807.

Operation:Input "match_phrase 1734536807" in the search box,the result is in Fig2.

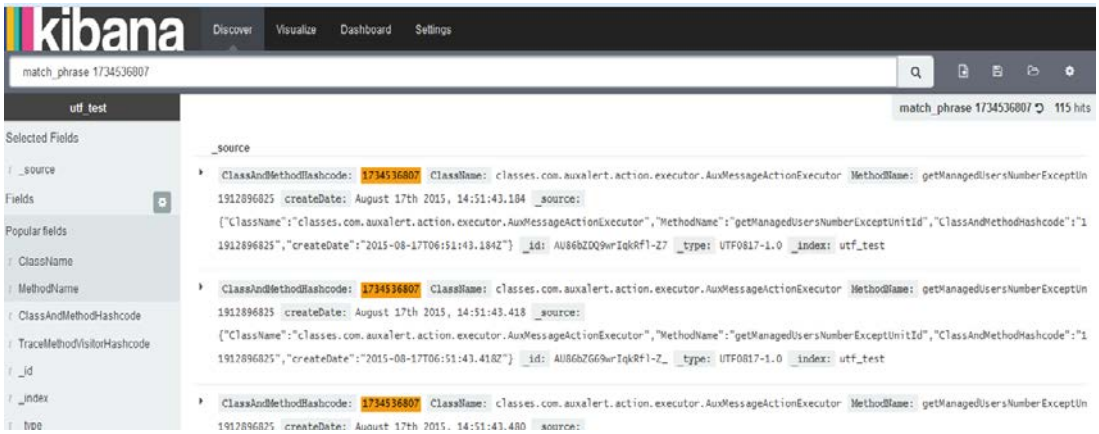

Fig2.The result of Experiment 1

Experiment 2:

Custom query: To find methods begin with the “get” AND ClassAndMethod Hashcode = 1734536807;

Operation: Input "1734536807" AND get *, the result is in Fig3.

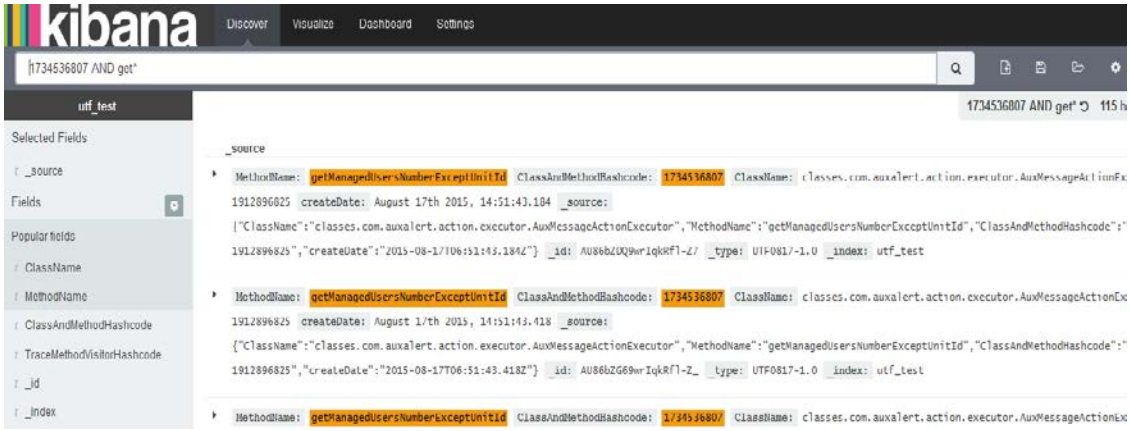

Fig3.The result of Experiment 2

Experiment 3:

Aggregation function, statistical field“MethodName” Top15 and “ClassNameAnd Method” Top30,the result is in Fig4.

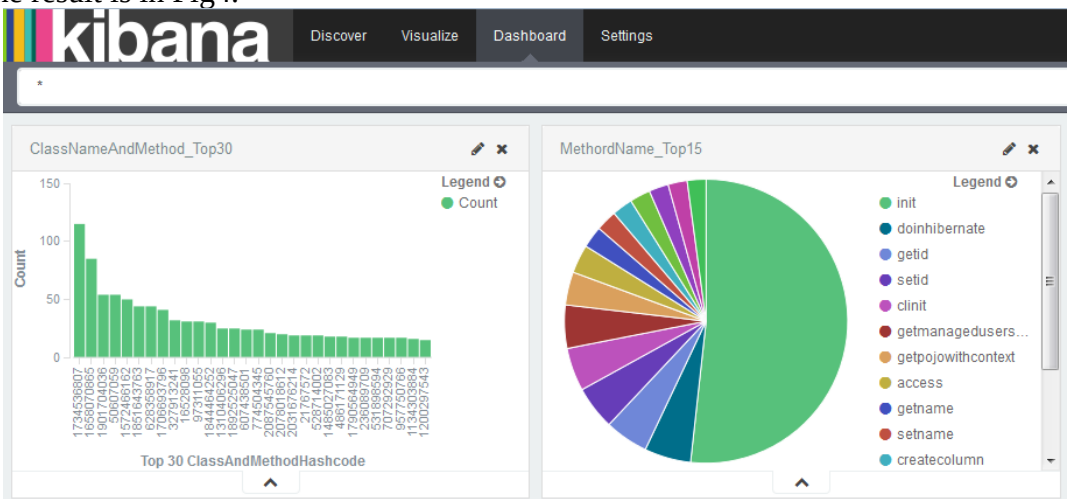

Fig4.The result of Experiment 3

Conclusion

This paper implements the double efficacy of ES , it is used as a "database"and as a "search engine". As "database", it is based on JSON (JavaScript Object Notation), so not only can achieve full text search, also can realize more flexible structured search; As a "search engine", it is real-time, but also highlight the search results, and support aggregation, summary,etc.

Acknowledgement

This work was financially supported by the National Science and Technology Support Projects - Jilin Financial Institutions Application Demonstration Project (Project No.2013BAH07F05).

References

[1]JunBai, HeBinGuo. Software Integration Research of Large-scale Logs Real-time Search Based on ElasticSearch[J].Journal of Jilin Normal University (Natural Science Edition),2014(2).

[2]XiaoLeiLu.Elasticsearch Authoritative Guide (Chinese version)[OL].http: //es.xiaoleilu.com/

[3]Junjie Chen, Guofan Huang. Reconstruct library search engine based on Elasticsearch[J]. Information Research,2015(11).

[4]KangJiang,JunFeng,ZhiXianTang,ChaoWang.The Metadata Searching and Sharing Platform Based on ElasticSearch[J].Computer and Modernization.2015(2).

[5]DaHongXu, Application Research in Vehicle License Plate Recognition System Based on Elaticsearch[J].Computer Era. 2014,(12).

[6]DouShi San, Kibana user guide in Chinese[OL].http://kibana.net.2014.06