ATLANTIS PRESS

Advances in Engineering Research, volume 149

2nd International Conference on Mechanical, Electronic, Control and Automation Engineering (MECAE 2018)

Efficient DNA Sequences Storage Scheme Based on HBase

Shaoxiong Wen[1, a]

[1]School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China.

[a]wensx93@foxmail.com

Keywords: HBase, DNA sequences, DNA division, file index

Abstract. In view of the characteristics of large amount of biological sequences data, fast growth rate and high sequences repeatability, a set of DNA sequences storage schemes based on HBase is proposed and implemented in combination with the related theory and technology of HBase distributed database. The pre-splitting strategy and Rowkey optimization based on DNA classification code is designed, which solves the problem of balanced load and hot spots of server.The efficient access of data is implemented by constructing the file index to replace the specific sequences. Experiments demonstrate that the DNA sequences storage system designed by these schemes has good storage capacity and scalability.

1. Introduction

As the biological sequences data grows rapidly, The storage and maintenance of biological sequences data is a major challenge that can not be ignored in the future.At the same time, The researchers ' demand for research and analysis of biological sequences, such as DNA sequences and protein sequences, is evolving towards cloud computing, more efficient storage and application of biological sequences data is a new topic currently faced by developers and researchers.

In this paper, the DNA sequences of biomolecules is used as the research object, according to the characteristics of the current biological sequences file storage format, a DNA sequences HBase[1] system is designed by using the Pre-splitting Rowkey design strategy based on DNA division[2] and file indexing mechanism. The feasibility of the system is verified through experiments.

2. System Design and Implementation

2.1 Table Design Principles

The number of column families should not be too much, because HBase table column family corresponds to the HBase physical storage structure called Store, when the memstore of Store flushing threshold will trigger flushing operation of all column families, so the column family Try to control the number of 1 to 2. The amount of data records between the column families should not differ too much, otherwise it will lead to a relatively small amount of data the column family data spread across multiple nodes RegionServer affect query performance[3]. Therefore, according to the HBase table design related principles, the GBFF format and Fasta format should be built separately, two tables are set only one column family column accessibility, when the corresponding column name columns qualifier are designed, Specifically for the structure of the second part of the GBFF format file, which contains multiple child attributes, the column name needs to combine FEATURES as a prefix with a child attribute, such as "Features_source", which satisfies the design requirements of a single list family in the table.

2.2 Pre-Splitting and Rowkey Design Strategy for Sequences Data

Due to the huge amount of DNA sequence and the complicated relationship among sequences, the classification of DNA sequences is also one of the problems faced by biological database vendors. After long-term development and exploration, the nucleotide databases of NCBI, EMBL and DDBJ three biological research institutes Sequence of classification management gradually formed a unified standard called DNA division,including BCT(Bacterial sequences),EST(Expressed sequence

tags),HTG(High throughput genomic sequences) and so on[4]. The amount of sequence data under each classification code is different, so we can pre-allocate the appropriate number of regions for each DNA division sequences data at the time of building the table, The specific allocation strategy and the split point called splitkeys are determined by the Rowkey design optimization.

The ACCESSION number[5] of DNA sequence record is used as Rowkey in HBase, Because of the Rowkey sort default dictionary order ascending, although it is very efficient for scan operation, it is also easy to cause the local hot issues mentioned previously, so for each DNA sequence record, a new Rowkey generation strategy combining with the Pre-splitting mechanism is proposed.

Set the number of Pre-splitting region is N, the division of DNA sequence data under the classification code of DNA sequence is Ai, thus the number of region required by the classification code can be Di=N*Ai.

Using the integer part of sequence retrieval number accession to get the value of di modulo as S, and the string prefix is spliced with the integer value S.

1) The string prefix do MD5 Hashishen into a 16-bit string prefixMD5, the string prefixMD5 the first 7 bits and the search number accession stitching to 16-bit Rowkey, delimiter using "_".

2) After the above operation is completed, a required Rowkey is generated, for example, The DNA sequence with AB000100 number is BCT, the model value is 1, and the corresponding Rowkey is "74378dc_ AB000100".

3) The integer value s obtained by taking the model ensures that the DNA records can be hashed into different region under the same classification code, but the DNA sequences in region still belong to a classification code, which control the dispersed granularity of the DNA sequence well.

2.3 File Index

In addition to the high repetitiveness of the DNA sequence, the sequence also shows as follows: 1) the sequence length of the sequence record is different. For example, the length of the Hepatitis A virus RNA gene fragment with the ACCESSION number of EU416249 is 348 and the ACCESSION number is the zebrafish DNA of BX927392 The length of the sequence fragment has reached 158543. 2) The amount of base sequence data under some classification codes is huge. For example, the length of DNA base sequence belonging to the HTG classification code reaches an average of 100,000, and the corresponding text size is about 100KB. From the database optimization point of view, the base sequence data is not suitable for direct storage in the column field, the structure of the file index is a common solution to such problems.

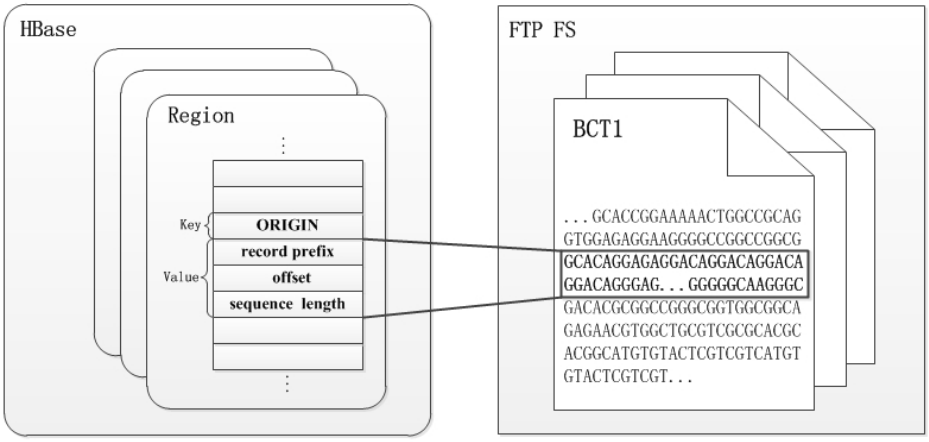

Fig. 1 The principle of File Index architecture

Taking GBFF format storage as an example, a large base sequence base file is pre-created in the ftp file system. The file name uses a prefix generated in a Rowkey optimization scheme to form a mapping relationship with the region generated by the HBase precodel in the previous section. When subsequent sequence records are submitted The base sequence is directly extracted according to the corresponding modulo value S and appended to the corresponding large file. Since the content of the file is stored as a continuous character string mainly composed of A, G, C and T, the byte offset of the

file can be summed Sequence length length to identify the base sequence in the file location. Schematic structure of the program as shown in Figure 2, this time HBase table ORIGIN field content by the record prefix, offset and sequence length and as a base sequence file index, follow-up experiments show that the program can greatly improve DNA sequence data Access efficiency and server cluster performance.

3. Experiment

HBase version 1.2.6, hadoop version 2.7.4, zoomkeeper version 3.4.10, jdk version 1.8.0, using four servers to build HBase cluster.Experimental data were collected from the GenBank partial nucleotide sequence dataset downloaded from the NCBI official website. The dataset consists of multiple GBFF-formatted sequence files compressed by gzip. After decompression, the total size is about 165.6G, and the sequence record size ranges from 2KB to 250KB, for a total of 45632843 sequence records.

Table 1 Server configuration

attribute	parameter
CPU	Intel(R) Xeon(R) CPU E5-2682, 2.50GHz
RAM	4GB
Operating System	CentOS 7.3 64
Bandwidth	1Gbps
Hard Disk	SATA 500GB

In order to compare the experimental results, the original scheme and optimization scheme of GBFF format data writing and querying are designed, and the corresponding system is implemented. In the original scheme, the data table does not do the pre-splitting processing, uses the retrieval number accession directly as the Rowkey, the field Origin stores the real nucleotide sequence, writes directly to the HBase data table. The optimization scheme adopts the pre-splittinging and the optimization strategy based on the classification code, each sequence records the field Origin construction file index instead of the nucleotide sequence, and writes the data and the base sequence text into the HBase data table and the FTP file system respectively. The experiment designed the corresponding system interface for the data query to obtain the complete sequence record as valid query.Both scenarios divide the data into four servers and deploy the scripts written in the above scenario, the following figure shows the average throughput comparison between the two scenarios when importing different data volumes.

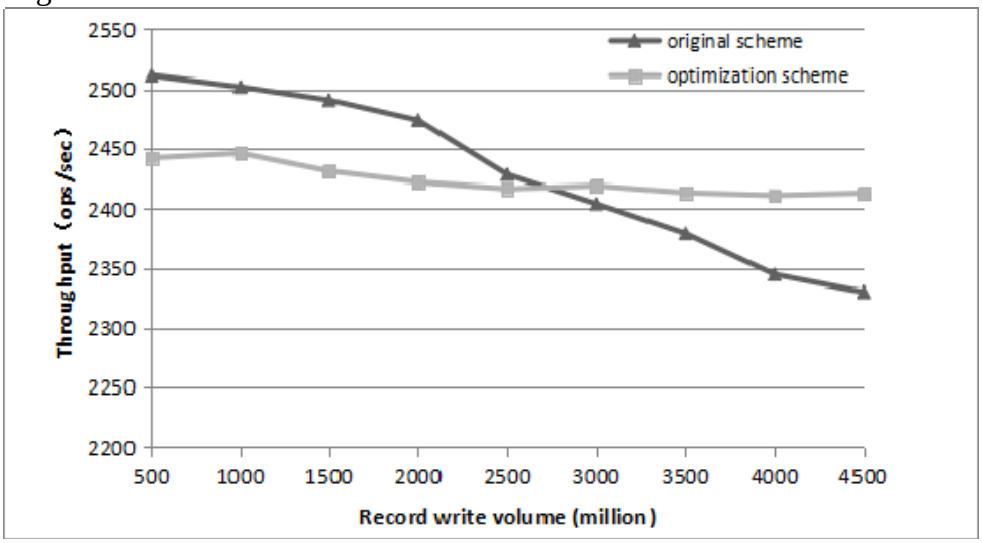

Fig. 2 Average throughput comparison of two schemes to import different amounts of data

Figure 3 shows the stability and efficiency of the optimized data writes, the amount of data that is imported is about 25 million before the optimization scheme throughput TPS is lower than the original scheme because the optimization scheme requires more preprocessing of the source data set,

including the optimization design and the construction of the file index and upload sequence files for the retrieval number of each record. However, with the increase of the data volume, the hotspot problem of the server of the original scheme appears gradually, and the base sequence directly as the origin field value leads to the HBase table size increase too fast, which leads to the node region a large number of flush and split operations, and the cluster load is more and more large, It can be seen that the average TPS of the original scheme is decreasing, while the optimization scheme keeps a high and stable TPS.

4. Summary

Aiming at the problem of storage maintenance caused by the rapid growth of biological sequence data over the years, this paper studies the structure of DNA sequence data and the current mainstream sequence file storage format, combining the storage model of HBase database with the related characteristics of HBase distributed storage. The storage of DNA sequence in HBase database is optimized in multidimensional degree. The experimental results show that the optimized DNA sequence storage scheme has better memory and query scanning performance. But the database function is unitary, therefore the future research work mainly aims at the biological sequence data retrieval comparison and the further performance optimization.

References

[1]. George L. HBase : the definitive guide[J]. Andre, 2011, 12(1):1 - 4.

[2]. Ouellette B F, Boguski M S. Database divisions and homology search files: a guide for the perplexed.[J]. 1997, 7(10):952-955.

[3]. Apache HBase ™ Reference Guide[OL]. [2013-7-1]. http://hbase.apache.org/book.html/

[4]. Wang Y Z, Chen S, Yuan L H. Introduction to Polymer Science[M]. Science Press, 2010.

[5]. Sample GenBank Record[OL]. [2006-10-23]. https://www.ncbi.nlm.nih.gov/