ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2016

Genetic Sequence Matching Using D4M Big Data Approach

Priyanka H.Y, Kavitha G

M. Tech, Department of CS& E, UBDT College, Davangere, India

Assistant Professor, Department of CS& E, UBDT College, Davangere, India

ABSTRACT: the term Big Data is usually used to describe huge amount of data that is generated by humans from digital media such as cameras, internet, phones, sensors etc. By building advanced analytics on the top of big data, one can predict many things about the user such as behavior, interest etc. However before one can use the data, one has to address many issues for big data storage. Two main issues are the need of large storage devices and the cost associated with it. Sequence of DNA storage seems to be an appropriate solution to address these issues of the big data. In this paper we are also using initial segmentation process as pre-processing step, Mapping by Tag generating, and D4M algorithm.

KEYWORDS: Big Data, DNA Sequence, Segmentation, Mapping, Tag,D4M.

I. INTRODUCTION

The challenges associated with big data are Volume, Velocity and Variety .Big data volume stresses the storage, memory and compute capacity of a computing system and requires access to a computing cloud. The velocity of big data stresses the rate at which data can be absorbed and meaningful answers produced. Big data variety makes it difficult to develop algorithms and tools that can address that large variety of input data. The ability to collect and analyze large amounts of data is a growing problem within the scientific community. The growing gap between data and users calls for innovative tools that address the challenges faced by big data volume, velocity and variety. Numerous tools exist that allow users to store query and index these massive quantities of data. Each storage or database engine comes with the promise of dealing with complex data. When using multiple technologies, however, there is significant trouble in designing the movement of information between storage and database engines to support an end-to-end application.

The standard approach for handling the increasing complexity of operations is to increase the size of the functions at the cost of increasing the effort required to build these functions. Solving this problem requires a composable mechanism (e.g., multi-dimensional associative arrays) for creating operations of increasing complexity without increasing their relative effort. A final challenge is that the above problems are ant correlated. Addressing one problem will typically result in the other issue becoming more difficult. For example, adding computing and network resources to a database system increases the difficulty of adding more diverse data and adding more complex operations. Thus, a viable solution must address all of the issues simultaneously.

D4M architecture addresses the challenges presented in the previous section by using a layered software architecture that addresses each challenge in its own layer. The top layer consists of compostable associative arrays that provide a one-to-one correspondence between database queries and linear algebra. Associative arrays can be both the input and output of a wide range of database operations and allow complex operations to be composed with a small number of statements. The middle layer consists of several parallel computation technologies that allow associative arrays to be distributed efficiently across a parallel computer. Furthermore, the pieces of the associative array can be bound to specific parts of one more databases to optimize the performance of data insertion and query across a parallel database system. The bottom layer consists of databases running on parallel computation hardware. D4M can use any type of database. D4M can fully exploit the power of databases that use an internal sparse tuple representation (e.g., a row/col/val triple store) to store all data regardless of type. The D4M approach provides several advantages and improvements over existing methods, specifically: D4M represents complex database operations and queries as

ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2016

compostable algebraic operations on associative arrays; D4M Provides distributed arrays for parallel database operations; D4M Transparently handles diverse data types using a tuple store.

II. RELATED WORK

Waldvogelet.al [1] explained the Scalable Distributed Information Management System (SIDM) is a hierarchical tree based information management system that aggregates data about a large scale networked system. It provides scalability through hierarchical aggregation and flexibility to accommodate a wide range of applications and data attributes. Furthermore, it performs lazy aggregation, on demand re-aggregation, and tunable spatial replication to ensure robustness.

Renesse, et.al [3] it is based on Astrolabe [59] which is highly robust due to its unstructured gossip protocol for data distribution and replication of all aggregated attribute values associated with a sub tree to all peers in the sub tree. SIDM has extra initial processing overhead to build the aggregation trees it needs to operate, as well as for additional overhead during DHT maintenance

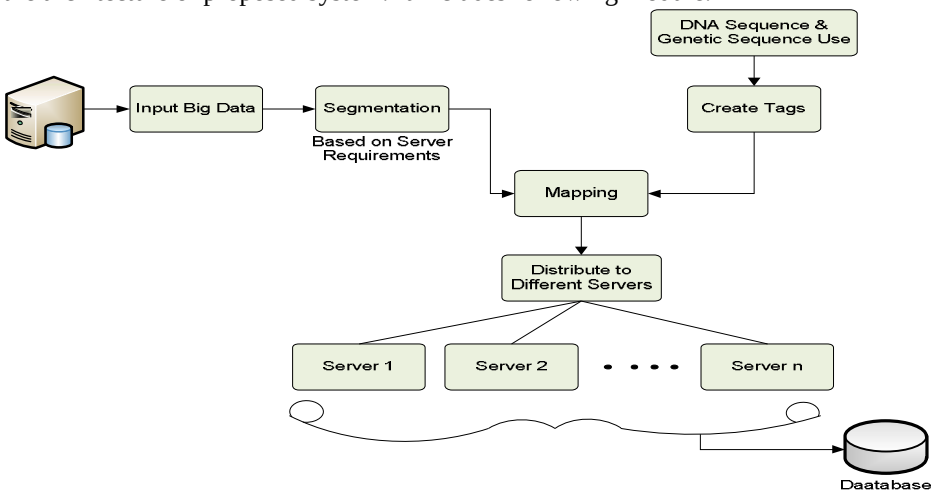
Bhagwan et.al [2] a comparative study of simple replication schemes and erasure coding schemes showed that an erasure coding scheme provides higher data availability than a simple replication scheme, but in the case of a P2P distributed dependency management system (D4M) where dependent artifacts are distributed across the network, the erasure coding scheme leads to high traffic overheads when it

.

Sting et .al [4] proposed the replica placement protocol uses a co-ordinate and controlled strategy which is globally known to each peer in the network in order to place the replicas. It uses the globally known hashing allocation function h(m, d) where m ≥ 1 is the index number of a replica instance and d is the identifier of each document. The allocation (hashing) function provides the address of a DHT peer on which a replica can be placed. Using this hash function with either the actually number of replicas present or the location of the closest replica, any peer in the network can find a potential replica"s address. In this replica selection protocol, locks are used during replica addition/deletion. These locks are used to avoid temporary inconsistencies in lookup while replica rd is being modified. These temporary inconsistencies may lead to these replicas not being selected for document retrieval in the worst case. These inconsistencies cannot be tolerated in a P2P dependency management system (D4M) where most of the artifacts depend on other artifacts in the system.

III. PROPOSED SYSTEM

Figure 1 shows the architecture of proposed system. It includes following module.



ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2016

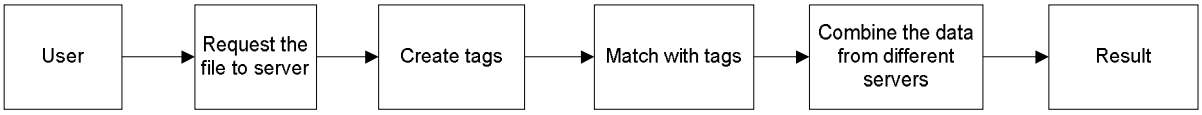


Figure 1: Shows the architecture of our proposed system.

a) Segmentation:

Sentence Segmentation It is a crucial first step in text processing. Segmentation a text into sentences is generally based on punctuation. The sentences boundaries and phrase boundaries can be estimated according to punctuation marks. we tokenize the sentences in the text file to create the words.

Word Segmentation It is getting words from text. The space is a good separator for this task but it will not work with special cases as compound words. Some compound words are written with a space in the middle even though they are single words. Such cases must be solved at this stage. It means that we must have knowledge base with similar words. With solving this problem, this stage is relatively easy.

b) Generating tags:

We are proposing a method of generating tags using Genetic Sequence or DNA sequence. If suppose we consider a genetic sequence.

$$S= \{ w_0, w_1, \ldots . . w_n \}$$

Tagging is a process of assigning each word w_i of S a corresponding tag. Tag is generated using attributes of text file i.e., text filename, Username, Server name. The use of DNA sequence has a single occurrence in the genome and whose location and base sequence are known.

Tag will be generated for text files and store it in a database. When a user sends the request for a file again a tag will be generated and matches with the tags that are stored in the database.

c) Mapping:

A Mapping Algorithm is used to identify the location in a reference genome. Mapping process executes when the user request a file from the server as shown below in figure 2, tag name is generated based on filename, username and server name. This tags is compared with already generated tags, if it matches the particular file is mapped from server to user. If it does not matched then user is requesting unknown file or access not permitted to that file. Use of DNA sequence automatically maps the matched tagged with the database to the user.

The Overall Model of our proposed system is D4M Scheme. We are using D4M approach for big data.

IV. RESULTS AND DISCUSSION

This section explains the results of our proposed system. Successful retrieval of text files to valid users is counted as hit count. Below figure3 shows the comparison graph for our proposed method to existing method.

.

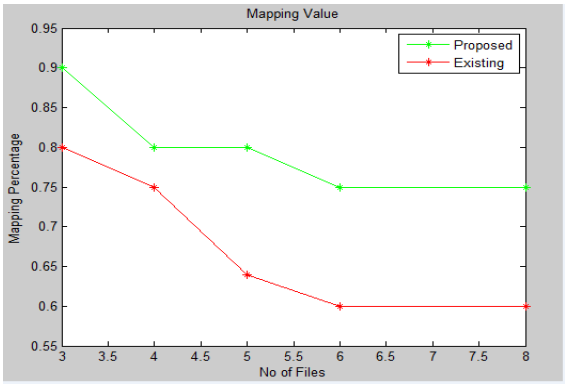


Figure 3: Graph for Mapping Values.

ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2016

V. CONCLUSION

The goal of the Dynamic Distributed Dimensional Data Model (D4M) is to combine the advantages of distributed arrays, tuple stores, and multi-dimensional associative arrays to create a database and computation system that solves the challenges associated with increasing data size, data diversity and operation complexity. In this paper the implementation of D4M has demonstrated simultaneous improvement in all of these dimensions when compared to current standard approaches (e.g., Java + SQL).The high hit count of our proposed method shows the accuracy and efficiency of system.

REFERENCES

1. M. Waldvogel, P. Hurley, and D. Bauer, "Dynamic Replica Management in Distributed Hash Tables", in Research Report RZ–3502, IBM, 2003.
2. R. Bhagwan, K. Tati, Y. Cheng, S. Savage, and G. Voelker. Total recall: System support for automated availability management. In Proceedings of NSDI, 2004.
3. R. V. Renesse, K. P. Birman, and W. Vogels, „Astrolabe: A Robust and Scalable Technology for Distributed System Monitoring, Management, and Data Mining", ACM Transactions on Computer Systems, Vol. 21, pp. 2003, 2001.
4. D. Stingl, C. Gross, J. Rückert, L. Nobach, A. Kovacevic, and R. Steinmetz. PeerfactSim.KOM: "A Simulation Framework for Peer-to-Peer Systems." In Proc. of the IEEE International Conference on High Performance Computing & Simulation (IEEE HPCS "11), 2011
5. K. Saller, D. Stingl, and A. Schürr. "D 4M, a Self-Adapting Decentralized Derived Data Collection and Monitoring Framework". Workshops of the conference scientific communication in Distributed Systems, Vol. 37, pp. 245 – 256, 2011.
6. The Annotated Gnutella Protocol Specification v0.4, 2003:[www]http://rfcgnutella.sourceforge.net/developer/stable/index.html#t1. Last access on 2012-05-17
7. D. Karger, E. Lehman, F. Leighton, M. Levine, D. Lewin, and R.Panigraphy, Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the World Wide Web. In Proceedings of the 29th Annual ACM Symposium on Theory of Computing, pp. 654–663,2000.
8. [5]. A. Rowstron and P. Druschel, Pastry: "Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems." Lecture Notes in Computer Science, pp. 22-25, 2001.
9. M. Ripeanu, I. Foster, and A. Iamnitchi, Mapping the Gnutella Network: Properties of Large-Scale Peer-to-Peer Systems and Implications for System Design. IEEE Internet Computing Vol. 6, No.1, 2002.
10. Y. Chawathe, S. Ratnasamy, L. Breslau, S. Shenker, and N. Lanham. GIA: Making Gnutella-like P2P Systems, Scalable. ACM SIGCOMM 2003.
11. J. Liang, R. Kumar, and K. Ross." The FastTrack Overlay: A Measurement Study. Computer Networks",Vol. 50,pp. 842–858,2006.
12. R. Morselli, B. Bhattacharjee, A. Srinivasan, and M. Marsh,"Efficient lookup on unstructured topologies",Proceedings of the Twenty-Fourth Annual ACM Symposium on Principles of Distributed Computing, Las Vegas, NV, USA, pp. 17 – 20, 2005
13. Löser, S. Staab, and C. Tempich, Semantic Social Overlay Networks. IEEE J. Sel. Areas. Communications, Vol. 25, No.1,pp. 5–14,2005.
14. J. Pouwelse, P. Garbacki, J. Wang, A. Bakker, J. Yang, A. Iosup, D. Epema, M. Reinders, M. van Steen, and H. Sips. Tribler: "A Social-based Peer-to-Peer syste". Proc. of the 5thInternational Workshop on Peer-to-Peer Systems.
15. I. Clarke, O. Sandberg, B. Wiley, and T. Hong, Freenet: A Distributed Anonymous Information Storage and Retrieval System, in Lecture Notes in Computer Science : Designing Privacy Enhancing Technologies: International Workshop on Design Issues in Anonymity and Unobservability, Berkeley, CA, USA, Proceedings:, Vol. 2009, Springer, pp.46–66,2000.