JES
Journal of Engineering Sciences
Vol 10, Issue 11, NOV/2019
ISSN NO:0377-9254

CDVC FRAMEWORK USING STORM INDEXING ALGORITHM FOR MAP-REDUCING

[1]KAKARLA SAMSON PAUL, [2]POLIKANTI GOUTHAM KRISHNA
[1]Assistant Professor, [2]Pg Scholar
DEPARTMENT OF CSE
Dr. K. V. Subba Reddy Institute of Technology, Kurnool, AP.

ABSTRACT:
CDVC is a cloud-based similarity search framework based on the DVC-based strategy that supports the following functionalities: (1) parallel pre-processing of dataset's descriptors for the storage management in the distributed databases over the cloud. (2) Indexing of a new descriptor q in real-time; (3) parallel query processing (similarity search) in the cloud. In this paper we propose two architectures, one is baseline architecture and another is in-memory architecture. The first architecture is used to utilizes a disk-based processing strategy. Another one uses the Apache Flink distributed stream processing framework. And also evaluate to conduct the two image datasets. The main aim of this paper is to develop a parallel media storm indexing algorithm using the map-reduce in the CDVC framework.

Key Terms—Stream processing, multimedia big data, indexing, cloud computing.

I. INTRODUCTION

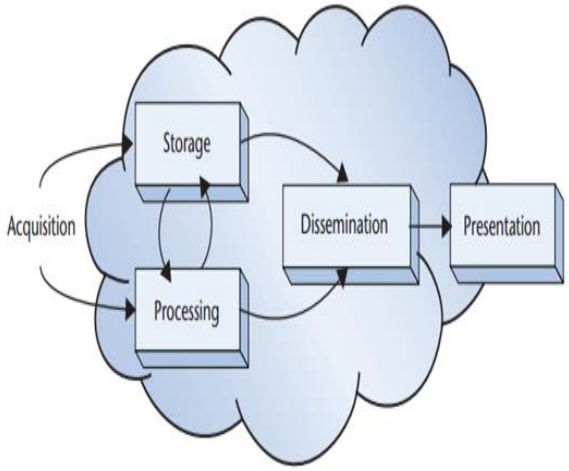
With the outburst of the Internet and user-generated content, and the increasing frequency of cameras, mobile phones, and social media, gigantic amounts of multimedia data are actuality produced, forming a distinctive kind of big data. Multimedia big data fetches marvellous probabilities for multimedia applications and services—such as multimedia searches, endorsements, advertisements, healthcare services, and smart cities. The need to compute such massive datasets is converting how we deal with multimedia computing.

Researchers have premeditated some of the problems in big data computing (see the related sidebar), but multimedia big data has its own characteristics related to multimodality, real-time information, quality of experience, and so on. For example, some multimedia learning applications, games, or 3D rendering might require GPU processing. Consequently, methods for general big data might not directly apply to multimedia big data.

Compared to approaches of general text-based big data computing, multimedia big data computing faces additional compression, storage, transmission, and analysis tasks in terms of

- Establishing unstructured and dissimilar data,
- Dealing with cognizance and thoughtful complexity,
- Addressing real-time and quality-of-service requirements, and
- Ensuring scalability and computing efficiency.

Here, we consider these technical challenges and the related scientific problems for multimedia big data computing, introducing various research directions and emerging technologies.



The emergence of big data computing is affecting the life cycle of multimedia content

II. LITERATURE SURVEY

D. Moise (2013) [3]

In this paper, author has focused on Hadoop, the open-source implementation of the Map-Reduce paradigm. Using as case-study a Hadoop-based application, i.e., image similarity search, author has presented the experiences with the Hadoop framework when processing terabytes of data. The scale of the data and the application workload allowed the author to test the limits of Hadoop and the efficiency of the tools it provides.

J. M. Banda (2014) [4]

In this work author has presented an alternative approach for large-scale retrieval of solar images using the highly-scalable retrieval engine Lucene. While Lucene is widely popular among text-based search engines, significant adjustments need to be made to take advantage of its fast indexing mechanism and highly-scalable architecture to enable search on image repositories.

S. Antaris (2015) [5]

Author has proposed a similarity search strategy over the cloud, based on the dimension's value cardinalities of image descriptors. Author's strategy has low pre-processing requirements by dividing the computational cost of the pre-processing steps into several nodes over the cloud and locating the descriptors with similar dimension's value cardinalities logically close. New images are inserted into the distributed databases over the cloud efficiently, by supporting dynamical update in real-time.

III. SYSTEM ANALYSIS

Proposed System

In this project we proposed an efficient parallel media storm indexing algorithm in the CDVC framework, built on Flink. From this framework we disclosed that a high indexing speed up factor (ISF) in all settings. Where the media storm distribution will match better with the examined application, to authenticate that the media storms have been correctly indexed in CDVC, we evaluate the search accuracy of CDVC by indexing media storms with the sequential indexing algorithm. An approximate indexing mechanism is proposed using the Map-Reduce paradigm to reduce the high latency and high CPU issue. Two different architectures, a disk-based and in-memory architecture, are presented in order to evaluate the benefit of the in-memory processing in the proposed mechanism, while the performance of the proposed media storm indexing mechanism is examined on different burst incoming rates within several "storming time frames", that is, the duration that the storm lasts. And we verify that the media storms have been correctly indexed in the DVC-based data structure by performing search queries, outperforming state-of-the-art methods in terms of indexing, search time and accuracy.

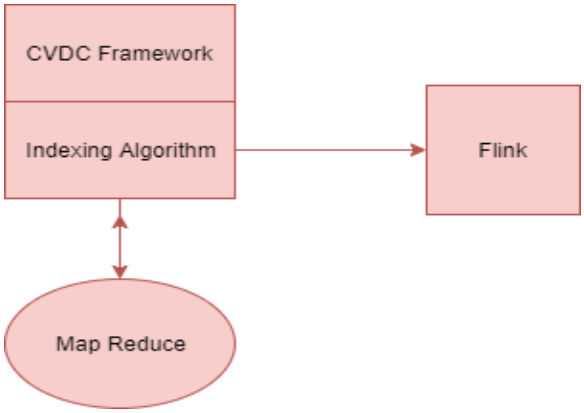


Fig.1 Proposed Architecture

IV.IMPLEMENTATION

Indexing

Given a new descriptor vector v_q, the goal is to identify the position pos_q in the double linked list L based on the priority index p. The correct position pos_q is located by identifying the set V_{pk} ($|V_{pk}| << N$), which is the set of the already stored image descriptors that have the same value with the new descriptor v_q at the first sorted dimension. We set a primary key P_k to the value of the first sorted dimension, corresponding to the dimension with the highest DVC. Then, the new descriptor is inserted in the logical position pos_q in $(O|V_{pk}| .log |V_{pk}|) + O(D)$ updating the double linked list from L to L'. Since ($|V_{pk}| << N$) the indexing algorithm is independent of the dataset size N.

Query Processing

Given the updated double linked list L^1 and the position pos_q of the image query q, a set V_{2W} of image descriptors is generated which is located in the updated double linked list L^1 in W previous and W next to the position pos_q, corresponding to a search radius 2W. To retrieve the top-k results, the distances between the set V_{2W} and the query image descriptor vector v_q are calculated using M computational nodes with T parallel threads.

Pre-processing

According to, Given M computational nodes and N descriptor vectors with D dimensions, CDVC performs a multi-sort algorithm in parallel. The aspects of image descriptors with high DVC are prioritized, assuming that these dimensions are more discriminative. The priorities based on DVC are stored in a priority index vector p. After the multi-sort algorithm has finished, the logical positions of the descriptors are stored in a double linked list L, which is the index of CDVC. In this study, we assume that the descriptors have been extracted locally and not in the cloud. Nevertheless, several works follow parallel strategies to speed up the extraction of descriptors.

V. CONCLUSION

In this project we conclude the Exact Indexing Mechanism with Map Reduce (EMR) mechanism presents high latency and high CPU cost because of the number of descriptors which have to be retrieved from the distributed databases and the number of required comparisons to index the incoming descriptors. In contrast, the Approximate Indexing Mechanism with Map-Reduce (AMR) mechanism overcomes the high latency and high CPU issues of EMR, by following an approximate strategy. AMR minimizes the number of descriptors retrieved from the distributed databases and compared with the incoming descriptor, achieving a speedup factor of 2.37 on average in the two evaluation datasets. The proposed AMR media storm indexing mechanism indexes the incoming descriptors efficiently in order to preserve the high search accuracy of EMR. Instead of using a subset of descriptors with the same primary key, the approximate mechanism of AMR uses a root descriptor and indexes the incoming descriptors in a relative position that is close to the position that the exact mechanism of EMR indexes.

REFERENCES

[1] X. Wu et al., "Data Mining with Big Data,"IEEE Trans. Knowl-edge and Data Eng., vol. 26, no. 1, 2014, pp. 97–107.

[2] P. Russom et al., "Big Data Analytics,"TDWI Best Practices Report, Fourth Quarter, 2011.

[3] D. Moise, D. Shestakov, G. T. Gudmundsson, and L. Amsaleg, "Terabytescale image similarity search: Experience and best practice," in Proceedings IEEE International Conference on Big Data, 2013, pp. 674–682.

[4] J. M. Banda and R. A. Angryk, "Scalable solar image retrieval with lucene," in Proceedings IEEE International Conference on Big Data, 2014, pp. 11–17.

[5] S. Antaris and D. Rafailidis, "Similarity search over the cloud based on image descriptors' dimensions value cardinalities," ACM Transactions on Multimedia Computing, Communications and Applications, vol. 11, no. 4, pp. 51:1–51:23, Jun. 2015.

[6] [4] M. Norouzi, A. Punjani, and D. J. Fleet, "Fast exact search in hamming space with multi-index hashing," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 36, no. 6, pp. 1107–1119, 2014.

[7] M. Muja and D. G. Lowe, "Scalable nearest neighbour algorithms for high dimensional data," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 36, no. 11, pp. 2227–2240, 2014.

[8] S. Schelter, S. Ewen, K. Tzoumas, and V. Markl, "All roads lead to Rome: Optimistic recovery for distributed iterative data processing," in 22nd ACM International Conference on Information and Knowledge Management, CIKM'13, 2013, pp. 1919–1928.

[9] A. Alexandrov, R. Bergmann, S. Ewen, J.-C. Freytag, F. Hueske, A. Heise, O. Kao, M. Leich, U. Leser, V. Markl, F. Naumann, M. Peters, A. Rheinlander, M. J. Sax, S. Schelter, M. Hoger, K. Tzoumas.