IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 8, April 2023

Optimizing the Performance of Data Warehouse by Query Cache Mechanism in Big Data

Anand Tilagul[1], Praveen S[2], Rachan S H[3], Shreyas M[4]
Assistant Professor, Department of Information Science and Engineering[1]
Students, Department of Information Science and Engineering[2,3,4]
SJC Institute of Technology Chikkaballapura, India

Abstract: *In today's world of Business Intelligence (BI), fast and efficient access to data from Data Warehouses (DW) is crucial. With the increasing amount of Big Data, caching has become one of the most effective techniques for improving data access performance. DWs are widely used by organizations for managing and using data in Decision Support Systems (DSS). To optimize the performance of fetching data from DWs, various methods have been employed, and one of them is the Query Cache method. Our proposed work focuses on a cache-based mechanism that improves the performance of DWs in two ways. First, it reduces the execution time by directly accessing records from cache memory, and second, it saves cache memory space by eliminating non-frequently used data. Our goal is to fill the cache memory with the most frequently used data. To achieve this objective, we utilize an aging-based Least Frequently Used (LFU) algorithm that considers the size and frequency of data simultaneously. This algorithm manages the priority and expiry age of the data in cache memory by taking into account both the size and frequency of data. LFU assigns priorities and counts the age of data placed in the cache memory. The cache block entry with the lowest age count and priority is eliminated first. Ultimately, our proposed cache mechanism efficiently utilizes cache memory and significantly improves the performance of data access between the main DW and the business user query.*

Keywords: Business Intelligence

I. INTRODUCTION

Currently, businesses are growing fleetly, and the influenceof Data storages (DWs) has been adding day by day, being used by several associations. High client demands are driving DWs to make fast business opinions grounded on the rearmost information. Dealing with a large quantum of data can decelerate down the performance, especially in data costing, which will affect the stoner experience. Several styles have been proposed to optimize data prosecution and data processing. The author employed query cache algorithms to enhance the efficiency of data processing.. The amount of data is increasing every day asa result of advances in software technology. By using technology, businesses are automated and effects are being recorded to make it easier and for fast and secure reclamation. To, latterly on, dissect this huge volume of literal data, identify the retired patterns and make successful business opinions. As we know, the quantum of data is growing day by day, so due to the large volume of the data, the data reclamation rate is also affected. To optimize the performance of penetrating data, interpreters apply different approaches to make it easier and faster to pierce. In this regard, experimenters are concentrated on the most advanced query acceleration and cache memory ways. Experimenters used the cache- grounded system to enhance the data access performance of the DW. The cache system can maintain the performance gap by considering the prosecution time between the main database and the cache memory. In order to efficiently use cache space, colorful ways are used. Least constantly habituated (LFU) and Least lately habituated(LRU) are the two most generally used algorithms for cache relief. In the last many decades, a variety of combinations of LRU and LFU have been proposed, and these combinations are used to come up with the optimal results for cache relief. It's veritably hard to add all objects into cache memory in order to make it effective. An expansive study on cache operation has led to a large variety of algorithms, including both general- purpose and sphere specific bones.

To maximize the application of cache memory, a competent algorithm should always keep the most popular blocks in cache. The most generally used algorithms for cache relief are Least constantly habituated (LFU), utmost constantly habituated(MFU), and Least lately habituated(LRU). Cache memory is placed with the Online Analytical Process (OLAP) for answering customer- grounded queries, and the results of business druggies are brought from the OLAP garcon and stored in cache memory. Several results for DW query cache optimization were proposed by the experimenters. still, these approaches have some limitations, similar as data dispose exploiting, poor cargo balancing performing in a long response time, ad hoc query constraints, query displacing, textual, spatial, and temporal data characteristics, and resembling prosecution of complex queries, which increases computational resource operation and affects prosecution time. Along with this, some authors proposed tackle- grounded results similar as pall- grounded surroundings and ultramodern processors for optimization of DW query processing. To overcome these limitations, in thispaper, we proposed the query cache medium for optimizing DW performance.

II. LITERATURE SURVEY

Title: An O (1) algorithm for implementing the LFUcache eviction scheme
Author: Dhruv Matani, Ketan Shah, Anirban Mitra
Abstract: Cache eviction algorithms are used to speed up execution bycaching frequently used data. LRU is the most commonly used algorithm due to its O(1) speed of operation and similarity to most application behavior. LFU is also desirable, but its O(logn) runtime complexity makes it less preferred. An LFU cache eviction algorithm with an O(1) runtime complexity is introduced using a hash table to store cache entries and a linked list to maintain access order. The least frequently used entry is evicted from the tail of the list when the cache is full.
Methodology Used: Least Frequently Used(LFU) cache eviction algorithm

Advantages:

- Reduced runtime complexity.

Disadvantages:

- It is suitable only for small amount of data.

Title: Grosbeak: A data warehouse supportingresource-aware incremental computing
Author: Zuozhi Wang, Kai Zeng, Botong Huang, Wei Chen, Xiaozong Cui, Bo Wang, Ji Liu, Liya Fan, Dachuan Qu, Zhenyu Hou, Tao Guan, Chen Li, Jingren Zhou
Abstract: This paper introduces Grosbeak, a novel data warehouse that uses Efficiently processing recurring routine jobs with resource-aware incremental computing, leading to a reduction in resource skew and optimizing resource usage. Unlike traditional data warehouse, Grosbeak breaks analysisjobs into small batches that incrementally process continuously ingested data and schedules them intelligently when the cluster has free resources.
The system is demonstrated using real-world analysis pipelines, allowing users to register recurring queries and observe the incremental scheduling behavior and smoothed resource usage pattern.
Methodology Used:

- Clustering

Advantages:

- It automatically optimizes the way resources are utilized by the underlying cluster.

Disadvantages:

- It takes more time to generate clusters.

Title: Enhancing data quality at ETL stage of datawarehousing
Author: Neha Gupta, SakshiJolly
Abstract: This study proposes improvements to various clustering algorithms for handling inappropriate symbols in datasets, such as dummy values, cryptic values, or missing values. The proposed improvements include using the expectation- maximization algorithm with dot product for cryptic data, the DBSCAN method with Gower metrics for dummy values, the Wards algorithm with Minkowski distance for contradicting data, and the K-means algorithm with Euclidean distance metrics for missing values. The proposed improvements have been shown to improve the quality of the data and ensure consistency when loading datainto a data warehouse.
Methodology Used:
- Expectation-Maximization algorithmand K-means algorithm

Advantages:
- The proposed methods enhanced the quality of the data and ensured consistency before loadingit into the data warehouse.

Disadvantages:
- It considering about data quality.

Title: Parallel query processing in a polystore Author: Pavlos Kranas, Boyan Kolev, OleksandraLevchenko
Abstract: With the increasing volume of data, polystores have become a crucial area of interest in the big data and cloud landscape. To make the most of the parallel processing capabilities of data processing platforms and underlying data stores, a polystore must (i) maintain the expressiveness of each data store's native query or scripting language and (ii) utilize a distributed architecture that allows for parallel data integration, including joins, on top of parallel retrieval of partitioned datasets. This paper addresses these requirementsby (i) using the CloudMdsQL query language's polyglot approach to enable the combination of native queries and SQL statements for ad-hoc integration, and (ii) Incorporating this approach within the LeanXcale distributed query engine, thereby allows for the parallel processing of native scripts at data store shards. Moreover,
effective optimization methods, including bind join, canbe used to enhance the performance of selective joins. Our experimental validation demonstrates the performance advantages of leveraging parallelism, high expressivity, and optimization.
Methodology Used:
- Distributed Query Engine

Advantages:
- Parallelism in combination with high expressivity and optimization.

Disadvantages:
- It needs more resources for implement.
- Expensive

Title: STAR: A cache-based distributed warehousesystem for spatial data streams
Author: Zhida Chen, Gao Cong, Walid G. Aref
Abstract: With the widespread use of mobile phones and location- based services, spatial data has become increasingly prevalent. As a result, there is a need for data stream warehouse systems that can provide real-time analytical results using the latest integrated spatial data. the STAR (Spatial Data Stream Warehouse) system, a distributed in- memory spatial data stream warehouse system that provides low-latency and up-to-date analytical results over a fast spatial data stream. The STAR system supports a broadrange of aggregate queries for spatial data analytics, such as comparing the frequencies of spatial objects in differentregions or displaying the most popular topics tweeted in various cities. STAR accomplishes aggregate queries by preserving distributed materialized views. Furthermore, STAR enables dynamic load adjustment, making it bothscalable and adaptable. We showcase STAR's capabilities using real datasets on top of Amazon EC2 clusters.

Methodology Used:

- Aggregate queries over spatial data streams

Advantages:

- It repartitions the workload dynamically to adapt tochanges in the workload.

Disadvantages:

- High Implementation Cost

Title: Distributed query optimization strategies forcloud environment
Author: Mostafa R. Kaseb, Samar Sh. Haytamy & Rasha M. badry
Abstract: Cloud computing provides various services over theinternet such as computing, servers, storage, databases, networking, and software that customers can use and payfor according to their usage. Database as a Service (DBaaS) is becoming increasingly popular in cloud computing due to its faster response times and increased reliability. Distributed databases are used in the cloud to store and manage data. Enterprises are transferring their stored data to the cloud computing center. However, in a distributed database system via cloud environment, the required relations by a query plan may be stored at numerous sites, leading to an exponential increase in the number of potential alternative plans to find an optimal QueryExecution Plan. Exhaustively exploring all potential plansin a large search space is not computationally feasible. Therefore, query optimization mechanisms are crucial for improving performance in the cloud, optimizing data processing time, and maximizing resource utilization. The aim of this paper is to provide researchers with a comprehensive overview of query processing and its optimization techniques by reviewing various methods usedfor query optimization.

Methodology Used:

- The techniques used to optimize query processing.

Advantages:

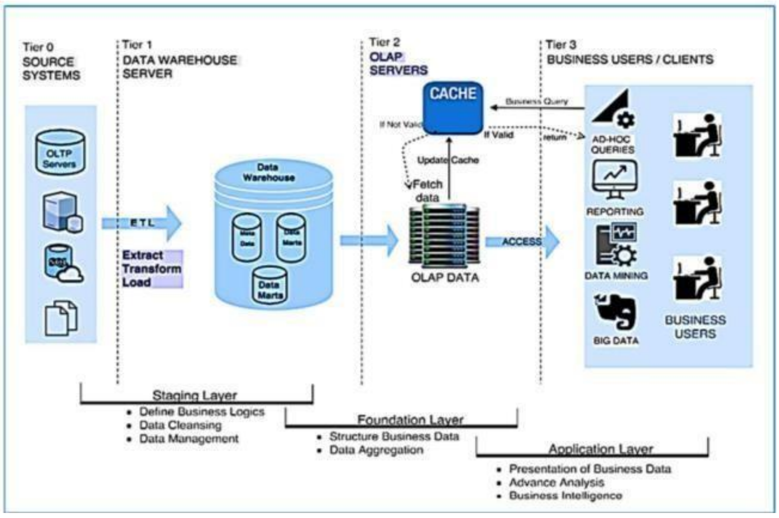
- Using distributed query optimization techniques can speed up query processing.

Disadvantages:

- Extensive resource waste.
- Extra tine consumption.

III. PROPOSED SYSTEM

The amount of data is increasing day by day due to the usage of software technology. By using technology, businesses are automated and things are being recorded to make it easier andfor fast and secure retrieval. To, later on, analyze this huge volume of historical data, identify the hidden patterns and make successful business decisions. As we know, the amountof data is growing day by day, so due to the large volume of the data, the data retrieval rate is also affected.



IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 8, April 2023

The issue of cache pollution is faced by the LFU (Least Frequently Used) algorithm. This happens when objects with extensive reference counts are replaced, even if the cached objects are not re-accessed again. To address this problem and save memory from non-frequently used data, a work [size-based cache] proposes the addition of an aging factor. The aging-based LFU adds a key value to the cached data based on its age. In the proposed solution for cache replacement technique, the cache mechanism is placed with OLAP after ETL operations. When Ad-hoc queries from business users are requested, the system first checks theresult in the cache memory (as shown in Figure 1). The ETL framework of a DW involves three steps:

- Extraction.
- Transformation.
- Loading.

IV. CONCLUSION

The choice of a cache replacement technique such as LFU, LRU, or SIZE is dependent on the data environment and domain. However, the implementation of cache memory in a DW with increasing data poses a challenge for OLAP. To address this, we have proposed a cache mechanism that combines frequency, size, and aging-based policies, which we believe will outperform using LFU, LRU, or SIZE-based policies alone. Moving forward, we plan to implement our proposed algorithm in a real-life testing scenario for an OLAP database server. Additionally, we will optimize cache performance using materialized views and concatenation to improve analytical query response time in a real-time data warehousing environment.

REFERENCES

[1]. W. Moudani, M. Hussein, M. Moukhtar, and F. Mora- Camino, ‘‘An intelligent approach to improve the performance of a data warehouse cache based on association rules,’’ J. Inf. Optim. Sci., vol. 33, no. 6, pp. 601–621, Nov. 2012.

[2]. A. Simitsis, P. Vassiliadis, and T. Sellis, ‘‘Optimizing ETL processes in data warehouses,’’ in Proc. 21st Int. Conf. Data Eng. (ICDE), Tokyo, Japan, 2005, pp. 564–575.

[3]. D. Matani, K. Shah, and A. Mitra, ‘‘An O (1) algorithm for implementing the LFU cache eviction scheme,’’ Oct. 2021, arXiv:2110.11602.

[4]. S. Huang, Q. Wei, D. Feng, J. Chen, and C. Chen, ‘‘Improving flash-based disk cache with lazy adaptive replacement,’’ ACM Trans. Storage, vol. 12, no. 2, pp. 1–24, Mar. 2016.

[5]. M. S. A. Khaleel, S. E. F. Osman, and H. A. N. Sirour, ‘‘Proposed ALFUR using intelegent agent comparing with LFU, LRU, SIZE and PCCIA cache replacement techniques,’’ in Proc. Int. Conf. Commun., Control, Comput. Electron. Eng. (ICCCCEE), Khartoum, Sudan, Jan. 2017, pp. 1–6.

[6]. Z. Wang, K. Zeng, B. Huang, W. Chen, X. Cui, B. Wang,

[7]. J. Liu, L. Fan, D. Qu, Z. Hou, T. Guan, C. Li, and J. Zhou, ‘‘Grosbeak: A data warehouse supporting resource-aware incremental computing,’’ in Proc. ACM SIGMOD Int. Conf. Manage. Data, Jun. 2020, pp. 2797–2800.