International Journal for Innovative Engineering and Management Research

PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL

www.ijiemr.org

IJIEMR Transactions, online available on 8th Aug 2024. Link

https://ijiemr.org/downloads.php?vol=Volume-13& issue=issue08

DOI: 10.48047/IJIEMR/V13/ISSUE 08/21

Title Parallel Query Processing in Big Data: A Hadoop Framework Comparison

Volume 13, ISSUE 08, Pages: 141 - 146

Paper Authors
Dr. G.V. Ramesh Babu , Venkatanna Gari Sreevidya

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

Parallel Query Processing in Big Data: A Hadoop Framework Comparison

Dr. G.V. Ramesh Babu
Associate Professor, Department of Computer Science, Sri Venkateswara University, Tirupati
gvrameshbabu74@gmail.com
Venkatanna Gari Sreevidya
Master of Computer Applications,
Sri Venkateswara University, Tirupati
sreevidyavenkatannagari17@gmail.com

Abstract
The metamorphosis of high-level questions into operational processes that can be successfully implemented in sequence on a multiprocessor machine is referred to as simultaneous query processing. This is accomplished by utilizing the parallelization of knowledge and the concurrent database system's different execution strategies. While processing these queries, not only transforming them into an execution process but also executing them precisely and redundant is important. Even though many number of different methodologies were developed in recent times that are based on advancing technologies, retrieving effectively, and lacking in the performance of the developed systems have been seen. To make the users and developers understand deeper into these concepts, we proposed a survey over the recent advancements in this industry in detail including their demerits to make sure that a developer to look through this article and understand quick to make their research regarding their concern issue and focus on it. We listed out the technologies involved, their advantages, and concentrated areas as well.
Keywords: Query Processing, Big Data, Apache Tomcat, Map Reduce, Hadoop, SQL Analytics.

Introduction

Request processing is the technique for responding to a query from a repository or information system, which typically entails comprehending the query, searching for information for capacity storage, and retrieving the information. A request statement was previously processed across a single server procedure without the sequential request processing characteristic. Multi request processing refers to the capacity to handle numerous jobs at the same time using a single query statement. We could process numerous in parallel efficiently over multiple server processes by identifying these statements that need to be processed and employing various industry services, which perform faster and more effectively than a single process server. The key benefit of this multi-processing system is that it can handle sophisticated inquiries in conjunction with decision-making applications or in bigger querying settings. During their execution, these queries are usually parallelized automatically.

Simultaneous request processing refers to the translation of high-level inquiries into concurrent implementation schedules by a multiprocessor computer. This is accomplished through the use of parallel content positioning and the various implementation mechanisms provided by the linear database architecture. Simultaneous programming benefits include machines that can better read instructions, saving time and money by filtering through "vast data" faster than ever before. Parallel computing has the potential to solve increasingly challenging problems and provide more resources. For analytically expensive issues, concurrent processors are used, implying that a large number of calculations are required. Simultaneous processing may be appropriate if the problem is very difficult to solve or if the results must be produced quickly. Geo processing, weather modeling, farming predictions, economic threat control, video-color restoration, fluid mechanics, diagnostic imaging, and drug development are all important uses. The structure of a concurrent operating system is a little tricky. Multi-processing has numerous advantages, but it also has drawbacks. There are some organizations that necessitate the use of specialized programming techniques. Energy consumption is high due to the multi-core design.

Literature Review:

[1] According to the author, a variety of interpretation machines aid data engineers by emphasizing similarities within data points and conducting feature extraction over exponentially rising and elevated data aggregation. While handy for a variety of tasks like user behavior insights, deployable event computation, and root-cause assessment, today's clarification systems are designed as self-contained computational instruments that do not interact with conventional SQL analytics frameworks, limiting their validity and expandability. They responded by proposing the DIFF operator, a relational consolidation command that combines the essential capability of these systems with expressive relational querying in the process. They demonstrated how DIFF could offer the same meanings as current descriptive engines while covering a large set of real use cases in commerce, by implementing single-node and shared implementations of the DIFF function in MB SQL, a MacroBase expansion. They have also shown how this expressive data interpretation technique provides greater physical and logical search efficiencies.

[2] When data retrieval approaches are combined with quality assurance methodologies, that urges the need of maintaining the quality criteria such as reliability, correctness, and thoroughness that can be enforced as components of ad-hoc query formulation and implementation, resulting in better query responses. Regarding the introduction of new data integrity assessment technologies, there are few investigations evaluating the effectiveness and sustainability of data durability evaluation tasks while information retrieval. This article reports the findings of an exploratory work aimed at determining whether Spark could provide superior improvement and extensibility benefits when performing data integrity in query processing activities on a variety of computational systems, along with a single asset multi-core device and a cluster-based framework, for a variety of requirements. Our findings show that combining efficient data science packages with the distributed and parallel abilities of big data processing can lead to significant scalability and performance advantages.

[3] According to the author, DAGs are extensively used for analyzing huge data in group-based shared computing frameworks. Developing an effective approach for DAG on a parallel data framework is crucial to designing automatic self-maintaining big data machines. A perfectly operating system is required to allocate a precautionary machine with all the relevant requirements within a parallel operating task that may dynamically variate during the execution. Allocating the necessary resources during the execution makes a system calculate the accurate computational time. The researchers of this paper dealt with this challenge by developing a new low cost-effective model namely, BOE to determine the allocation of the necessary resources by identifying the congestion accurately and forecasting its execution time. Within a DAG work process, they have also introduced a relatively standard approach that heuristically allocates the resource allocation equity within among the levels for foreseeing a plan. Several cost-effective models were also implemented to estimate the working of this model.

[4] Nested queries are complex queries that are also combinatorial queries that are challenging to write and are highly time-

consuming to compute. Much research was performed to rewrite these complex queries and divide them to make them understand, easy to compute, and increase the performance. As it is a definable approach that requires more logic and constructional efforts this is majorly not a free service among the most available database systems. Thus, the developers here have introduced a general utility machine and a GPU simulated mechanism that aims in performing high at the least cost for development. As said the authors here have observed in depth among the nested and unnested queries for identifying their liberties and limits for processing in GPU. They have also developed a hybrid model which is said to be algorithmically effective and simple, with lower space complexity having a generic framework. Parallel computing reading with indices, smart vectorization to accelerate join procedures, leveraging cache proximity for loops, and optimal GPU storage handling are just a few of the essential architecture optimizations the developers undertook. The possible improvements have been incorporated in NestGPU, a GPU-based sectional database system that is GPU hardware ambivalent.

[5] According to the author, AQP could involve the mechanisms of ML to work efficiently and enhance the performance and accuracy in query execution when compared to those of standard measures. The developers said that after their intense research about this topic, they have concluded to opt for a salient framework DBEst++ to derive lightweight ML methods that cover a large part of the known workload in analytical queries with the usage of huge memory storage pre acquittance. The architectural property of the opted methodology depends on the word embedding methods with NN associated with dependencies-based forecasting for estimating the density and correlated attribute values. The developers here have presented the design properties and motivations related to the proposed DBEst+ and discussed how the ML models could be included. They also discussed which kind of challenging tasks could be considered, how to deal with such tasks including high cardinality types of attributes by retaining high preciseness under the updating data. The experimentation results also include memory spaces and response time as well.

[6] In the advent of overcoming, the issues regarding the accuracy and time complexity for processing a multitask query, the developers of this research have proposed an innovative framework named WAPBC – CSMR. This framework is said to be incorporated over a large dataset with a greater number of queries. Initially, the developers have performed WAPB for classifying the dataset into several of its respective categories which in return for overcoming the classification time, a special method named, GDR was implemented that categories within a minimum time. A likelihood ratio is calculated from the weaker to stronger learner outcomes and the classified data is restored in a prefetcher cache. Rigorous steps were induced later in this framework that also works on eliminating the queries that get stuck in the cache and consumes the processing time, thus avoiding the computational time. Many other issues like the accuracy of query processing, rate of error, and processing time for the verified user queries were also considered.

[7] By the author, RDF has been a prominent framework that defines data with an ambiguous or fluctuating schema utilizing the idea of triples due to its versatility. Its prominence has generated in a slew of large-scale transdisciplinary datasets, prompting the creation of RDF handling algorithms. There are two types of existing practices: initially, a relational paradigm to store the triples in records, and the latter uses data structures to express RDF data as a tree. Because they use optimization tactics from the conceptual data, such as tagging and segmentation, the former group's techniques are more expandable. Yet, when coping with sophisticated queries that are durable in current systems, these methods have a lot of latency. Graph-based techniques, however, use more complicated data formats that fail to handle main memory adequately and are not extensible in computer equipment with restricted resources. In this article, they presented a way of querying RDF data that combines both RDF graph traversal and actual triple segmentation. The structure of

this system is then clearly described, as well as the method for managing main memory during query execution, which is built on the Volcano execution environment.

[8] From the developer's context, considering a high-dimensional massive data collection, skyline request computation takes a long duration. To handle this difficulty, parallel processing approaches are required, with MapReduce is among the most prominent mechanisms for processing huge data. In the research, a large variety of effective MapReduce skyline technologies have been suggested, with most of them focusing on splitting and trimming the provided dataset. There are, nevertheless, chances for more redundancy. Using a unique LShape segmentation technique and an efficient Propagation Refinement technique, the developers have offered two simultaneous skyline computational methods in this paper.2Phase LShape and 1Phase LShape are the two computational models for numerous removers and solitary suppressants that were introduced, correspondingly. We demonstrate that our techniques outperformed the compared standard techniques, particularly for high-dimensional and numerous amounts of data collection, through comprehensive trials.

[9]With the advancements in technology and usage of the cloud, many companies have been accessing the cloud for having the advantage of the service DBaaS. But, according to the author, one may face many issues while getting access to this service regarding flexibility and managing the data scalability. Recently, many companies have been opting for No SQL databases, but depending on the necessity of relational databases in the cloud to manage the data which is crucial in decision making. OLAP questionaries require a majority of the time for processing and demand high-performance abilities within their corresponding DBs to achieve outcomes in a feasible period. The developers of this research have established a middleware solution named C-ParGRES that could be employed on a cloud that could explore the replicated data and handle the interquery and intraquery parallelly and supports OLAP. This framework also generates variant independent virtual groups for different DB and its clients.

[10] The developer of this research paper explains that for effective inference-depended understanding, it is becoming increasingly crucial to concentrate on explosive development and a huge number of substantive RDF information. PTIF, CTIF, and DRTF are conventional methodologies for determining inference-depended reasoning conclusions over large amounts of RDF data. It is sufficient for executing semantic web search interpretation; nonetheless, these methods can be applied due to their logical inference architectures. The sensitiveness challenge for large RDF datasets argumentation is addressed with MLMIS, which is performed within the MapReduce architecture. The experimental analysis has utilized Dbpedia, a benchmarked RDF dataset to measure the responsiveness of numerous requests across alternative Hadoop workflows in order to demonstrate the performance of PTIF, CTIF, and DRTF. The developers state that their model works better when compared to the Cassandra model.

Results and Discussion:

Table 1: Comparative Study

S.No.	Author	Algorithms	Merits	Demerits
1.	Abuzaid	Diff, Anti-diff commands	The introduced commands are said to increase interoperability.	In the future, the application of these commands in real-time could be implemented to boost productivity.

2.	Cisneros-Cabrera	Spark	The developers here also included the guidelines necessary for choosing the required computational structure for processing the queries of DQ.	In the future, the extended library could be used and deeper experimentation with Apache Spark tuning could be investigated.
3.	Juwei Shi	BOE	The developed model automatically identifies the bottleneck situation of a resource at different levels.	The skew impact would be considered in the cost estimation process for further studies.
4.	S. Floratos	NestGPU	Best for nested methods. Also developed a cost model that predicts the computational time for the nested query that allows the query optimizer to select its own least execution procedure.	
5.	Qingzhi Ma	DBEst++	Showed higher accuracy and greater response time.	Future enhancements could be developed over light AQP and its integration into DB engines.
6.	Tamil Selvan	WAPBC - CSMR	The developed framework as said increases the accuracy of query processing reduces time complexity and error rate.	The redundancy of this method is not explained when applied in several other smaller or bigger applications.
7.	Khelil	RDF	Achieved scalability	Limited Main memory.
8.	H. Wijayanto	MapReduce	Greater performance over high dimensional data and massive data.	Variant skyline query measures need to be verified as all processes may not be compatible.
9.	W. M. Ribeiro	C-ParGRES	Cost-effective solution and provides elasticity.	Further, this could be worked out in applying for enormous databases and checking performance variations in VMs.

10.	SaikishoreYagala	Map Reduce	Higher data retrieval time.	

Conclusion:

This study has provided most of the information regarding the parallel computing problems and their respective frameworks. Each article represented in the written literature explains a specific problem while dealing with parallel computing and respective efficient algorithm to solve that issue. No matter the type and size of data, most of the listed articles show an innovative approach that could be deployed over other supporting environments as well. Finally, every developer desires an efficient and precise framework that reduces their burden over the problem of working over multiple queries at a time. This paper presented a detailed view of the types of problems a user might face, and their accurate respective solution. Understanding complex queries, fragmenting them in a meaningful order, and then processing them was also explained in one literature. Limitations with productivity, parameter tuning, cost estimations, and memory management were also discussed.

References:

[1] Abuzaid, F., Kraft, P., Suri, S. et al. DIFF: a relational interface for large-scale data explanation. The VLDB Journal 30, 45–70 (2021). https://doi.org/10.1007/s00778-020-00633-6.

[2] Cisneros-Cabrera, S., Michailidou, A.-V., Sampaio, S., Sampaio, P., &Gounaris, A. (2021). Experimenting with big data computing for scaling data quality-aware query processing. Expert Systems with Applications, 178, 114858. doi:10.1016/j.eswa.2021.114858.

[3] J. Shi and J. Lu, "Performance Models of Data Parallel DAG Workflows for Large Scale Data Analytics," 2021 IEEE 37th International Conference on Data Engineering Workshops (ICDEW), 2021, pp. 104-111, doi: 10.1109/ICDEW53142.2021.00026.

[4] S. Floratos et al., "NestGPU: Nested Query Processing on GPU," 2021 IEEE 37th International Conference on Data Engineering (ICDE), 2021, pp. 1008-1019, doi: 10.1109/ICDE51399.2021.00092.

[5] Qingzhi Ma, Ali Mohammadi Shanghooshabad, Mehrdad Almasi, Meghdad Kurmanji, Peter Triantafillou. Learned Approximate Query Processing: Make it Light, Accurate and Fast. In 11th Conference on Innovative Data Systems Research, CIDR 2021, Virtual Event, January 11-15, 2021, Online Proceedings. www.cidrdb.org, 2021

[6] Tamil Selvan, S., Balamurugan, P. & Vijayakumar, M. Prefetched wald adaptive boost classification based Czekanowski similarity MapReduce for user query processing with bigdata. Distrib Parallel Databases 39, 855–872 (2021). https://doi.org/10.1007/s10619-020-07319-6.

[7] Khelil, A., Mesmoudi, A., Galicia, J. et al. Combining Graph Exploration and Fragmentation for Scalable RDF Query Processing. Inf Syst Front 23, 165–183 (2021). https://doi.org/10.1007/s10796-020-09998-z.

[8] H. Wijayanto, W. Wang, W. Ku and A. Chen, "LShape Partitioning: Parallel Skyline Query Processing using MapReduce," in IEEE Transactions on Knowledge and Data Engineering, doi: 10.1109/TKDE.2020.3021470.

[9] W. M. Ribeiro, M., A. B. Lima, A., & Oliveira, D. (2019). OLAP parallel query processing in clouds with C-ParGRES. In Concurrency and Computation: Practice and Experience (Vol. 32, Issue 7). Wiley. https://doi.org/10.1002/cpe.5590.

[10] Saikishore Yagala, Myneni Madhu Bala (2020). MULTI- INFERENCE APPROACH FOR EFFICIENT DISTRIBUTED REASONING OF LARGE-SCALE RDF DATA. Journal of Xi'an University of Architecture & Technology, Volume XII, Issue IX, 2020