IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 25, Issue 1, Ser. II (Jan. – Feb. 2023), PP 17-20
www.iosrjournals.org

An Approach to select Optimal index selection using Q-learning algorithm for NoSQL database

V.Sumalatha[1], Dr .Suresh pabboju[2]
Ph.D Scholar, Dept of CSE, Osmania University.
Professor of IT, Director-AEC &COE CBIT Osmania University.

Abstract: *Database indexing is a vital activity to improve query performance, as indexes are the most commonly used technique to speed up query response time. Data inserts and updates take longer if the right columns are selected, which increases disk consumption and increases inserting and updating time. Therefore, it is important to optimize the performance of a NoSQL database using index selection. The code must be chosen according to the workload to ensure the efficient operation of the database. To manage the workload and index configuration selection, reinforcement learning (RL) is considered the efficient model since it is growing and can compute from scratch. Although the deep reinforcement learning model performs well for index selection, time complexity or processing time is to be optimized further. Thus, to solve this issue, an optimized deep Q-learning network is presented in this paper. To enhance the performance of the deep Q-learning network and calculating the Q value ,throughput, and average index selection.*

Keywords: RL, Optimisation Technique, NoSQL, Q-learning.

Date of Submission: 05-02-2023 Date of Acceptance: 17-02-2023

I. Introduction

The improvement of cloud computing and the Internet, databases can efficiently store and process large amount of data. NoSQL databases are increasingly used due to the high performance required during read and write. More than 225 different NoSQL repositories have been reported so far and still growing [1]. Indexes are more important in NoSQL.Providing efficient and scalable indexing services is essential for real-time data analytics in NoSQL databases [2-3]. Database indexing is a vital activity to improve query performance, as indexes are the most commonly used technique to speed up query response time. Data insert and update take longer if the right columns are selected, which increases disk consumption and increases inserting and updating time. Therefore, it is important to optimize the performance of a NoSQL database using index selection. The code must be chosen according to the workload to ensure the efficient operation of the database.

Achieving an optimal indexing configuration in a database is not so easy [4]. All query columns need to be fast indexed for fast data retrieval. However, finding a balanced trade-off between required storage and performance is more complex. Whenever the cost-based optimizer fails to find the correct solution, the DBA makes the final decision on the database architecture regarding indexing strategies. Code selection should match as many codes and most workloads as possible. A static workload varies when using a database. Also, there are large differences in choosing the optimal code in different hardware environments.

II. Literature Review

Larger data volume of various applications development leads to the today’s IT growth, now a days it is observed, with Relational database we cannot work on large data to maintain high volume data base due to the strict constraints on data structure,data relations and so on. Various formats of different industries including unstructured data has to be stored and processed in the databases. Hence, NoSQL types are the solutions for various issues of large database.NoSQL database may be good enough when not compare to the relational database because the relational database can collect data in the form of a table and can store data that is structured as well .NoSQL stands for “Not Only Sql.” But this doesn't mean that this term opposes SQL, but that is not the case. NoSQL systems can coexist relational and non-relational databases.

They are suitable in applications where there is a large amount of data is involved. Data in this case is structured, unstructured or semi-structured. There are different types of NoSQL databases. Some of these include Cassandra, Mongo DB, Couch DB and HBase among others. All of them differ in terms of structure and usage as per requirement. Depending on these factors, such databases have been further classified into various categories. Since NoSQL databases cannot be used interchangeably, in this paper practical usages of them or explained as per these categories. They are not compatible like relational databases. In situations where there is a need certain category of NoSQL you cannot use a different one. As per need each database type is having its own specialized characteristics in that particular environment so the designer can choose correct database type to implement [15].

There are four NoSQL database categories:

• Key-Valued Stores

• Column Family Stores

• Document Databases

• Graph Databases.

In 2016,Ameri [7] proposed a self-tuning approach to manage the indexes of a database using its query optimizer. Considering the wide usage of databases, ever increasing data size and the demand for fast data access, it is crucial to automate the management of database design and the database performance tuning process.One crucial aspect of database performance tuning is the creation of appropriate indexes for a given workload of sample queries and write operations issued to the database.A self-tuning mechanism that adopt the number of indexes and choose the proper attributes for indexing based on the ratio of read to write The databases have their own cost functions implemented in their query optimizer, a reasonable way of choosing indexes is to present a limited number of candidate indexes to the optimizer and recommend its chosen ones. The databases have their own cost functions implemented in their query optimizer, a reasonable way of choosing indexes is to present a limited number of candidate indexes to the optimizer and recommend its chosen ones.

AbbasiKamel and TaharEzendine [9] presented a method based on the knapsack algorithm to exploit benefits over the greedy algorithm in 2020. A dynamic selection approach for indexes and materialized views with the Knapsack algorithm maintain the optimal solution even after the data modification. Knapsack algorithm is used to selection of indexes and materialized views on finding estimate cost.They proposed an optimal dynamic selection based on the use of indexes and materialized views.The solution is said to be optimal when it answers all the requests with a minimum response time. This approach allows to optimize the execution of the requests and to adapt an optimal solution to the modifications of data in the base one applies the Knapsack algorithm.

In 2019,Neuhaus, et al., [10] developed GADIS, an algorithm-based approach for selecting indexes automatically. By using this approach, you can find database configurations that exceed all baselines while saving on storage. Genetic Algorithm is a search based optimization technique based on the principles of genetics and natural selection.GA is finding an optimal index configuration like find the fitness function. There are two fitness functions : An optimization objective to maximize the database performance considering INSERT, DELETE and SELECT queries and Designed to optimize the query response time by search for faster index.

In 2020, Maryam, Eslam, and Amir [13] developed a feedback control loop for continuous monitoring and lightweight workload analysis in NoSQL-wide column stores. This loop described the design pattern for the self-tuning feature and was utilized to forecast workload changes necessary for automatic schema database re-tuning. The workload model was based on construction using a reconfigurable colored Petri-net model. Results of the article showed that the construction time and adaptation time of the model increasedgradually with workload level.

Reinforcement Learning (RL) is a machine learning domain that focuses on building self-improving systems that learn for their own actions and experiences in an interactive environment. In RL, the system (learner) will learn what to do and how to do based on rewards[16].

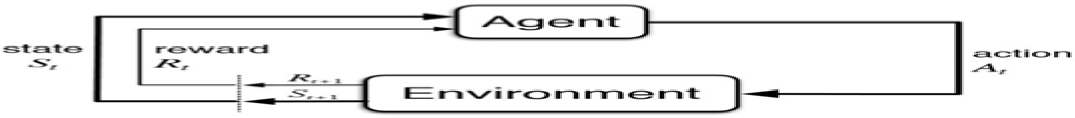


Fig 1: RL interact with Agent and Environment

- ➢ Agent – learner who takes decisions based on previously earned rewards.
- ➢ Action – the step an agent takes in order to gain a reward.
- ➢ Environment – a task which an agent needs to explore in order to get rewards.
- ➢ State – in an environment, the state is a situation or position where an agent is present.

➢ The present state contains information about the previous state of the agent which helps them with the next course of action.

➢ Reward- an Agent receives rewards or punishments in response to the actions performed.

III. PROPOSED MODEL

➢ For optimal index selection, an optimized deep Q-learning network is presented in this work.

➢ To enhance the performance of the deep Q-learning network, an improved heuristic algorithm is presented.

➢ Using this algorithm, the best action sequences are obtained. These optimal sequences and their corresponding rewards are given as input to the Q network for calculating the Q value. At final, the action sequence with maximum reward is selected as the optimal index configuration using NoSQL database.

Q-learning

Q Learning is a model-free value-based Reinforcement Algorithm. The focus is on learning the value of an action in a particular state[17].

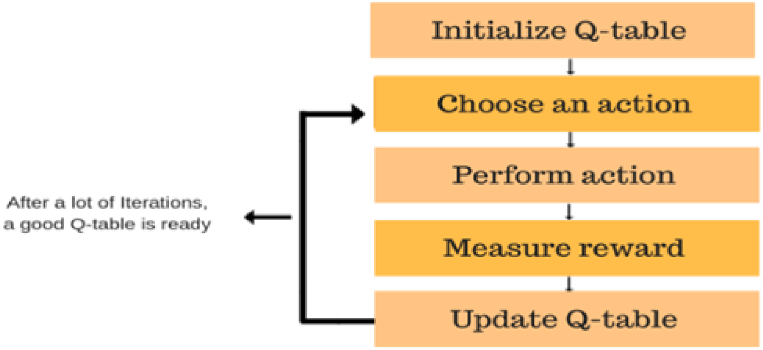


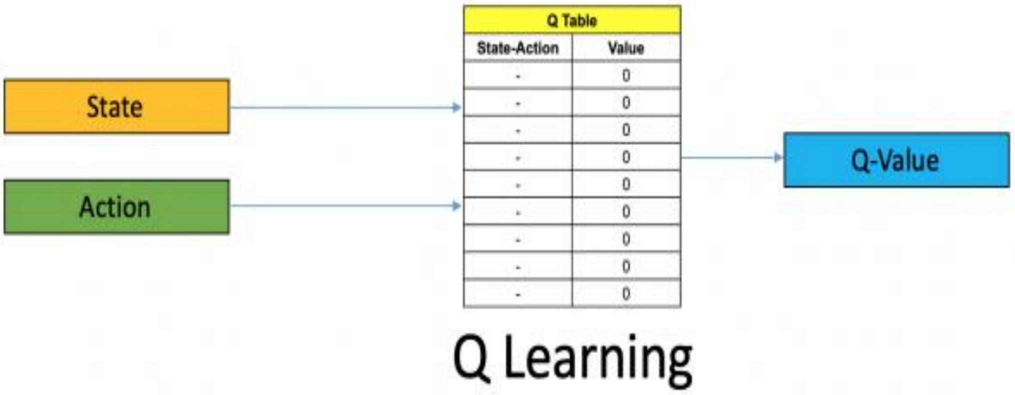
Fig 2: Q- learning table

➢ Q-Table contains the Q-score, the maximum expected future reward that the agent will get if it takes a specified action.

➢ Each row signifies a particular state in the environment and each column is dedicated to the actions.

➢ Initially, in Q-learning, we explore the environment and update the Q-table until it's ready and contains information about better actions for each state to maximize the rewards.

The value of a state V(s) is the maximum of all the possible Q-values.



Q Table	
State-Action	Value
-	0
-	0
-	0
-	0
-	0
-	0
-	0
-	0
-	0

Fig 3: Q-learning architecture

Deep Q-Learning

In deep Q-learning uses of neural networks. In terms of the neural network we feed in the state, pass that through several hidden layers (the exact number depends on the architecture) and then output of the Q-values[17].

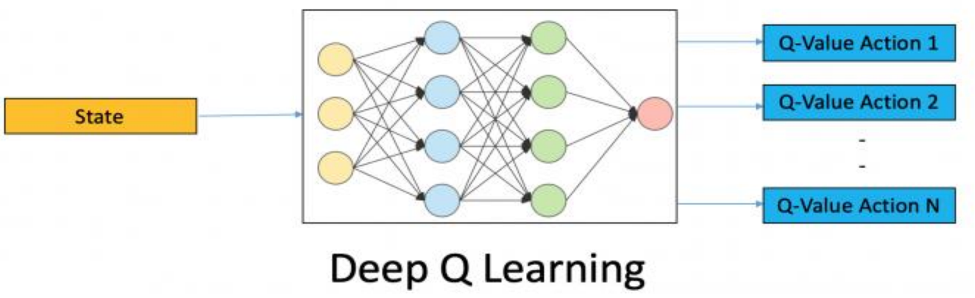


Fig 4: Deep Q-learning architecture

IV. CONCLUSION

There are so many optimization techniques for sequential databases (SQL) but there are very less techniques for NoSQL databases optimization, performance improvement and optimal index selection. By using Q-learning algorithm itself is RL with deep learning model, obviously which enhances the index selection process.

REFERENCES

[1]. Nosql database list, [EB/OL], https://hostingdata.co.uk/nosql-database/

[2]. Gayathiri, N. R., D. David Jaspher, and A. M. Natarajan. "Big Data retrieval techniques based on Hash Indexing and MapReduce approach with NoSQL Database." In 2019 International Conference on Advances in Computing and Communication Engineering (ICACCE), pp. 1-8. IEEE, 2019.

[3]. Gao, Xiaoming, and Judy Qiu. "Supporting queries and analyses of large-scale social media data with customizable and scalable indexing techniques over NoSQL databases." In 2014 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, pp. 587-590. IEEE, 2014.

[4]. E. Petraki et al., "Holistic indexing in main-memory column-stores," in SIGMOD. ACM, 2015, pp. 1153–1166.

[5]. Yu Yan, Shun Yao, Hongzhi Wang, Meng Gao, "Index selection for NoSQL database with deep reinforcement learning" , Information Sciences 561 (2021) 20–30 Elsevier 26 January 2021.

[6]. Alsayoud, F. and Miri, "March. Index Selection on MapReduce Relational-Databases". In 2015 IEEE First International Conference on Big Data Computing Service and Applications (pp. 302-307). IEEE.

[7]. Ameri, " On a self-tuning index recommendation approach for databases". In 2016 IEEE 32nd International Conference on Data Engineering Workshops (ICDEW) (pp. 201-205).IEEE.

[8]. Alavala, M. and Alhamdani, "Automatic database index tuning using machine learning". In 2021 6th International Conference on Inventive Computation Technologies (ICICT) (pp. 523-530). IEEE.

[9]. Kamel, A. and Ezzedine, " Dynamic selection of indexes and views materialize with algorithm Knapsack". In 2019 International Conference on Internet of Things, Embedded Systems and Communications (IINTEC) (pp. 214-219). December 2019 IEEE.

[10]. Neuhaus, P., Couto, J., Wehrmann, J., Ruiz, D.D.A. and Meneguzzi, F.R., "GADIS: A genetic algorithm for database index selection". In The 31st International Conference on Software Engineering & Knowledge Engineering, 2019, Portugal.(SEKE).

[11]. Ravat, F., Song, J., Teste, O. and Trojahn, C., "Efficient querying of multidimensional RDF data with aggregates: Comparing NoSQL, RDF and relational data stores". International Journal of Information Management, 54, p.102089.

[12]. Kain, R., Manerba, D. and Tadei, "The index selection problem with configurations and memory limitation: A scatter search approach". Computers & Operations Research, 133, p.105385.May 2021 ,Elsevier Ltd.

[13]. Mozaffari, M., Nazemi, E. and Eftekhari-Moghadam, "Feedback control loop design for workload change detection in self-tuning NoSQL wide column stores". Expert Systems with Applications, 142, p.112973,2020.

[14]. Zahra Sadri, Le Gruenwald Eleazar Leal,"Online index selection using deep reinforcement learning for a cluster database" ,July 26,2020 IEEE 36th International Conference on Data Engineering Workshops (ICDEW).

[15]. V. Sumalatha,Dr. Suresh Pabboju, "Overview of NoSQL Databases and A Concise Description of MongoDB",International Journal of Engineering Research & Technology (IJERT). Published by Vol. 9 Issue 12, December-2020.

[16]. https://www.guru99.com/reinforcement-learning-tutorial.html.

[17]. https://www.mlq.ai/deep-reinforcement-learning-q-learning/

V.Sumalatha, et. al. " An Approach to select Optimal index selection using Q-learning algorithm for NoSQL database." *IOSR Journal of Computer Engineering (IOSR-JCE)*, 25(1), 2023, pp. 17-20.