An Overview of Dashboard made using DynamoDB

Pulkit Kakkar - 40160971
Dinesh Sankuri - 40041026
Bo Wang - 40183470

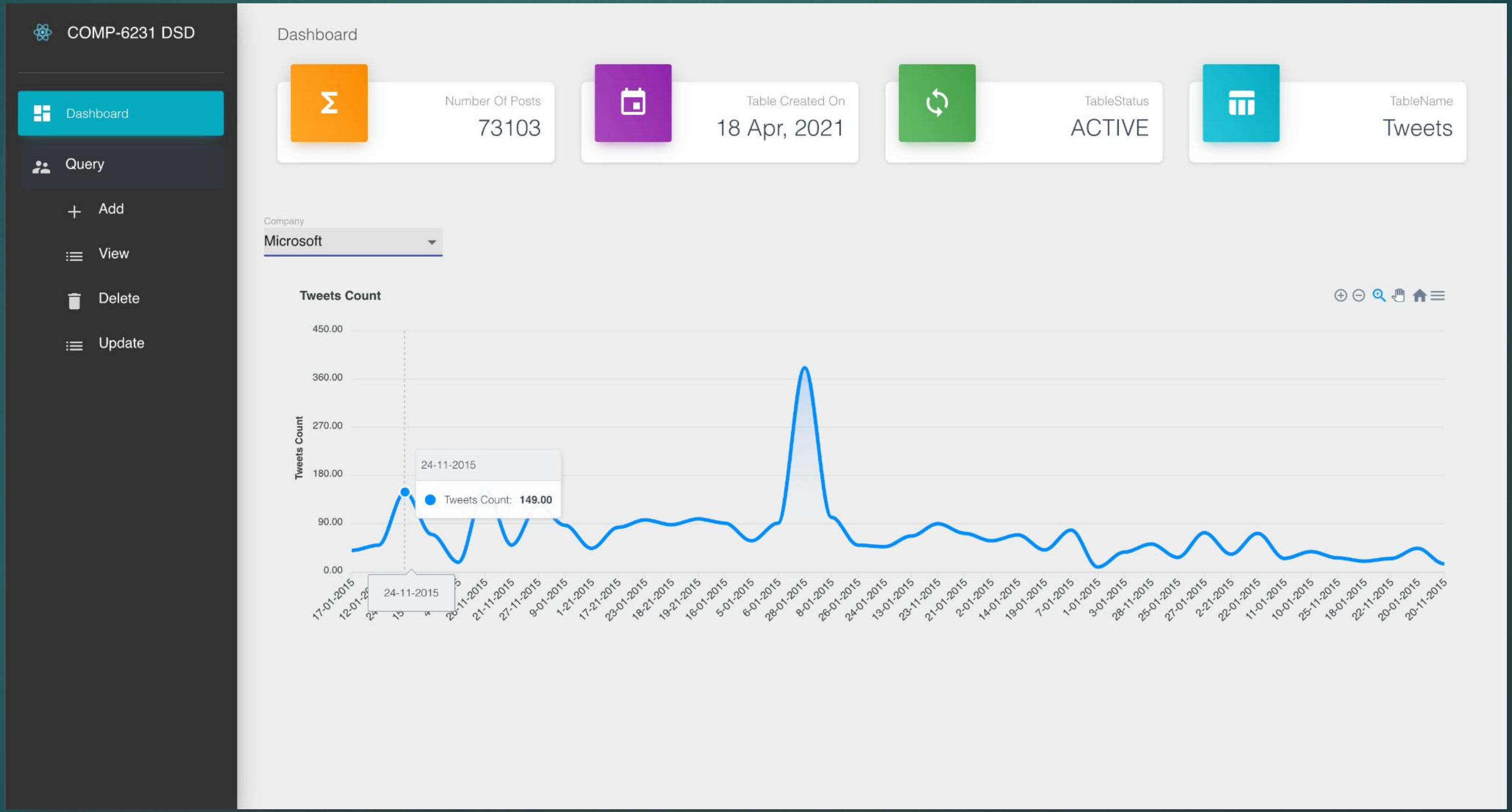
DynamoDB

DynamoDB is a NoSQL database service provided by AWS.

- It is completely managed by AWS.
- It provides simple APIs, which can add, delete, modify, and check data.
- As the amount of data increases sharply, it can still provide stable performance output .

Aim

To develop a dashboard which communicates with DynamoDB to illustrate results of various operations.



Dashboard

Number Of Posts 73103

Table Created On 18 Apr, 2021

TableStatus ACTIVE

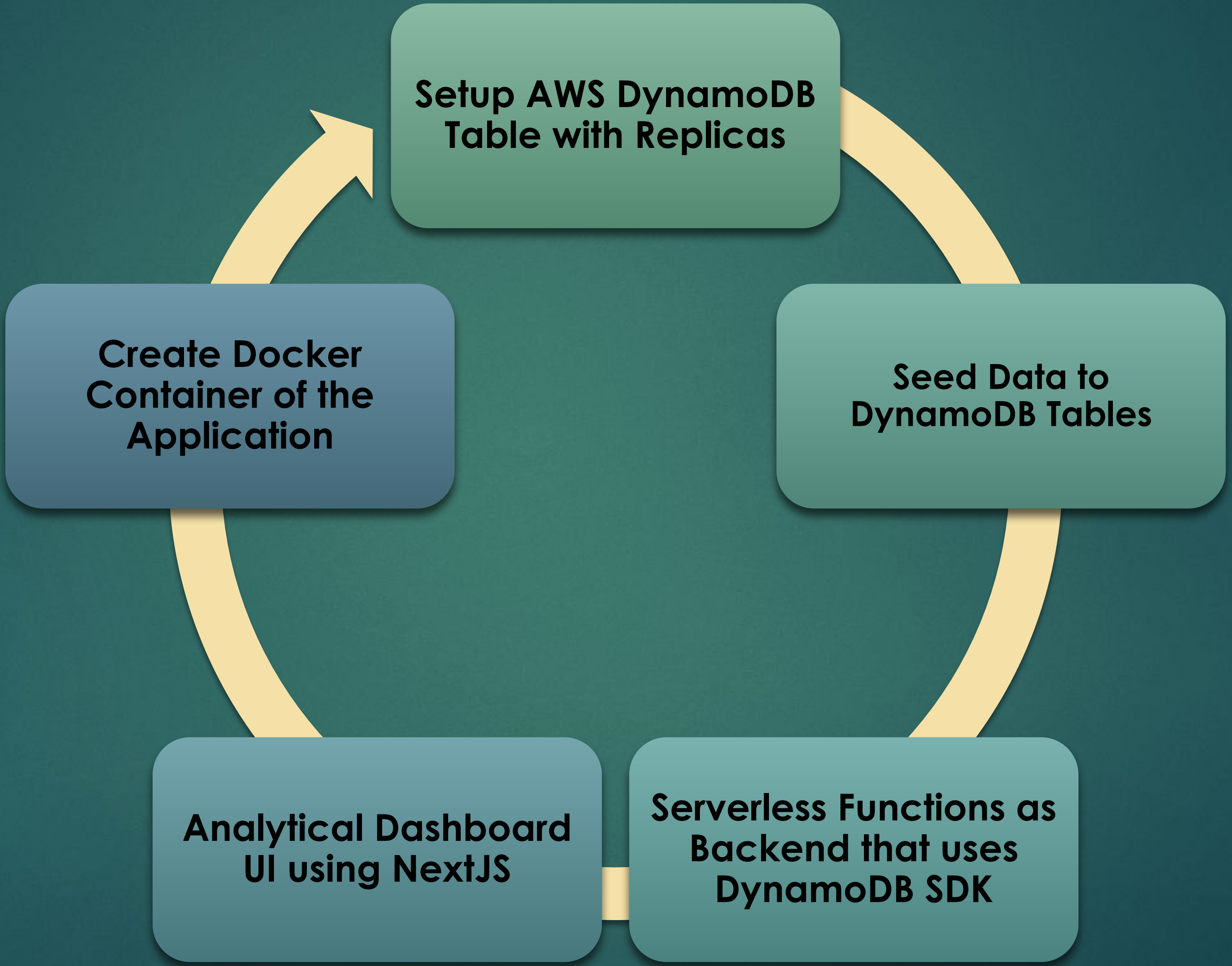
TableName Tweets

Company: Microsoft

Tweets Count

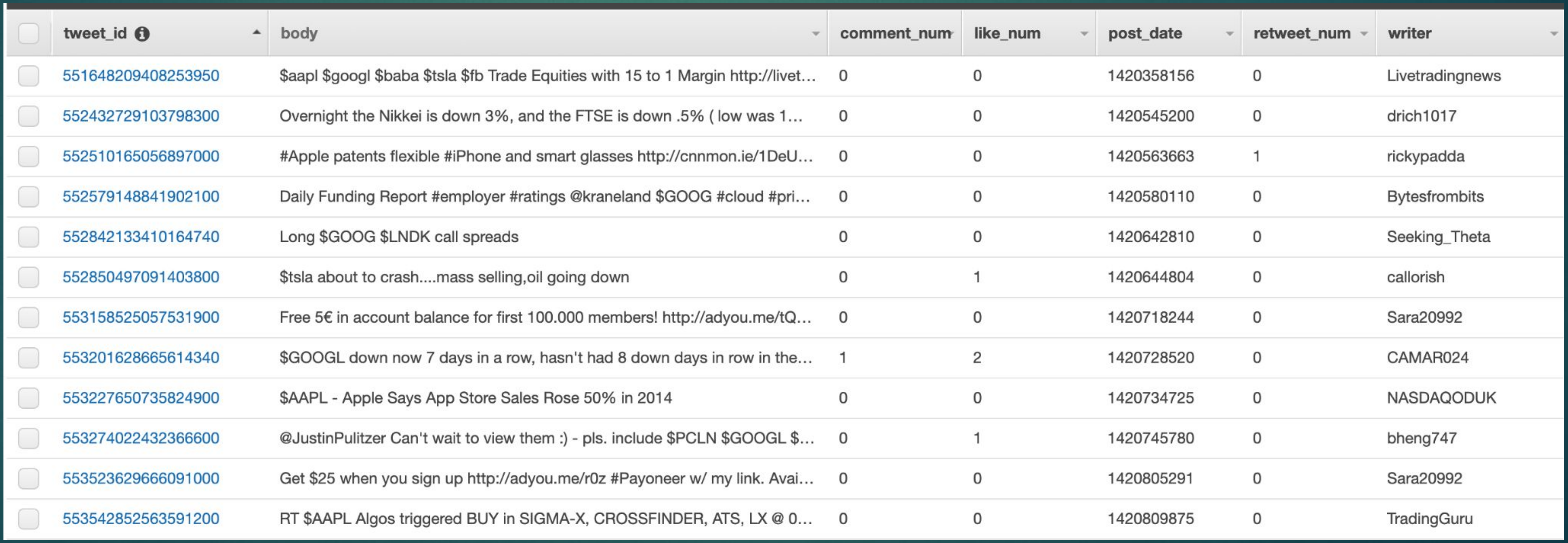
24-11-2015
● Tweets Count: **149.00**

Project Phases


Setup AWS DynamoDB Table with Replicas
Seed Data to DynamoDB Tables
Serverless Functions as Backend that uses DynamoDB SDK
Analytical Dashboard UI using NextJS
Create Docker Container of the Application

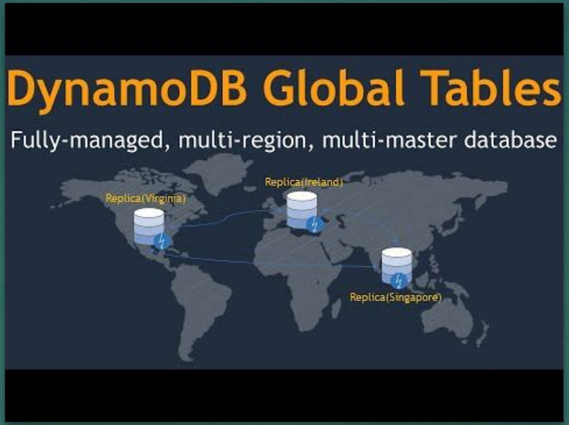
Data Set Used

- An **open-source dataset** from Kaggle
- Contains Information about **tweets** that are written about Amazon, Apple, Google, Microsoft, and Tesla by using their appropriate share tickers from 2015 to 2020.
- **Table Schema:**



tweet_id	body	comment_num	like_num	post_date	retweet_num	writer
551648209408253950	$aapl $googl $baba $tsla $fb Trade Equities with 15 to 1 Margin http://livet…	0	0	1420358156	0	Livetradingnews
552432729103798300	Overnight the Nikkei is down 3%, and the FTSE is down .5% (low was 1…	0	0	1420545200	0	drich1017
552510165056897000	#Apple patents flexible #iPhone and smart glasses http://cnnmon.ie/1DeU…	0	0	1420563663	1	rickypadda
552579148841902100	Daily Funding Report #employer #ratings @kraneland $GOOG #cloud #pri…	0	0	1420580110	0	Bytesfrombits
552842133410164740	Long $GOOG $LNDK call spreads	0	0	1420642810	0	Seeking_Theta
552850497091403800	$tsla about to crash....mass selling,oil going down	0	1	1420644804	0	callorish
553158525057531900	Free 5€ in account balance for first 100.000 members! http://adyou.me/tQ…	0	0	1420718244	0	Sara20992
553201628665614340	$GOOGL down now 7 days in a row, hasn't had 8 down days in row in the…	1	2	1420728520	0	CAMAR024
553227650735824900	$AAPL - Apple Says App Store Sales Rose 50% in 2014	0	0	1420734725	0	NASDAQODUK
553274022432366600	@JustinPulitzer Can't wait to view them :) - pls. include $PCLN $GOOGL $…	0	1	1420745780	0	bheng747
553523629666091000	Get $25 when you sign up http://adyou.me/r0z #Payoneer w/ my link. Avai…	0	0	1420805291	0	Sara20992
553542852563591200	RT $AAPL Algos triggered BUY in SIGMA-X, CROSSFINDER, ATS, LX @ 0…	0	0	1420809875	0	TradingGuru

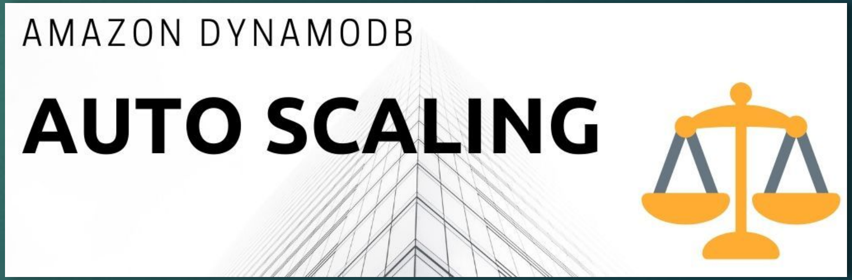
Distributed Properties



Containers

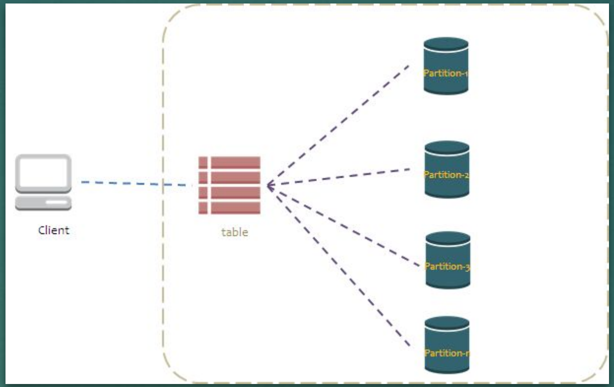
Reliable Communication & Availability

Replicas

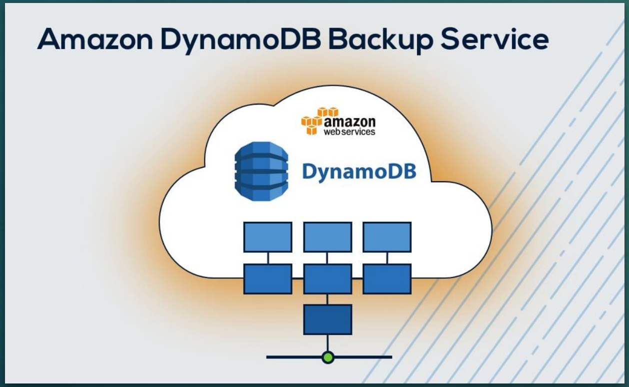


Distributed Properties

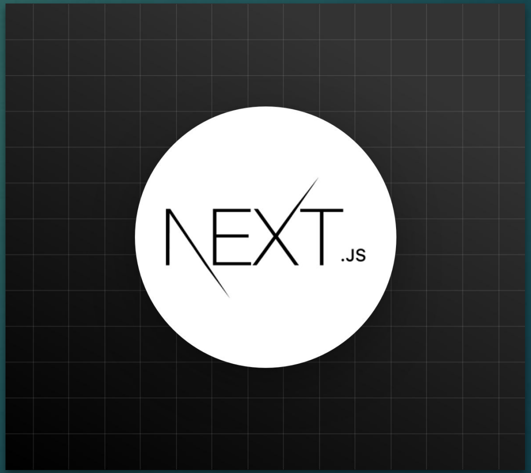
Recovery from Failure with Point-in-Time Recovery for DynamoDB



Cloud Functions (Like RPCs)



Flat Naming (Hash Table)



DynamoDb Operations Demonstrated

Batch Write To Seed Data in DB

GET to get attributes for provided primary key

describeTable to get table information for replica

DELETE, PUT, POST to update table and trigger replica update

Scan to execute complex query on non-indexed key

Demo

Amazon DynamoDB

Pulkit Kakkar
Concordia University
Montreal, Quebec
pulkitkkr@gmail.com

Bo Wang
Concordia University
Montreal, Quebec
bowangmontreal@gmail.com

Dinesh Kumar
Concordia University
Montreal, Quebec
dineshsnkumar@gmail.com

ABSTRACT

The advent of NoSQL databases has increased the amount of data storage, improved scalability and enhanced the flexibility of the data model. In this paper, we are introducing the Amazon DynamoDB, which is a key-value NoSQL Database. The concepts in distributed systems in dynamodb such as Scalability, Reliability and Consistency are analyzed. Also the techniques used in DynamoDB implementation are introduced. The access commands used and backup, replication, auto-scaling services are explained in detail.

1 INTRODUCTION

DynamoDB is a NoSQL database service provided by AWS. It is fully hosted on AWS. Developers only need to define the data access mode and some key information to use it through HTTP API. It has the following characteristics:

- As the amount of data increases sharply, it can still provide stable performance output
- It is completely managed by AWS. Developers do not need to SSH to their server, do not need to manage the server, do not need to update the OS patch and encryption library on the server, etc.
- It provides simple APIs, which can add, delete, modify, and check data. In addition, developers can also define query modes according to the scenarios in which the data is obtained[1]

2 CONCEPTS OF DISTRIBUTED SYSTEMS IN DYNAMODB

2.1 Scalability

Dynamo supports incremental expansion, because the increase and decrease of nodes must have good scalability, and reduce the data flow during the period as much as possible, thereby reducing the performance jitter of the cluster. Dynamo chooses to use a consistent hash algorithm to handle the addition and deletion of nodes.

Let's imagine what the limitations of traditional hashing algorithms are. Once I have given the total number of nodes h,the node to which the data is divided is fixed (x mod h). At this time, once I increase or decrease the size of h, then all the data is The mapping relationship has to be changed, and the solution can only be data migration, but consistent hashing can prioritize the data area that each node is responsible for on a circle. In this way, every time a node is added or deleted, the scope of influence is limited to a small part of the data.

Consistent hashing has disadvantages. If you just map each node directly to a ring, it may cause the complexity of the range between nodes to be large or small, resulting in unbalanced data distribution and machine load. Therefore, there is an optimization measure for

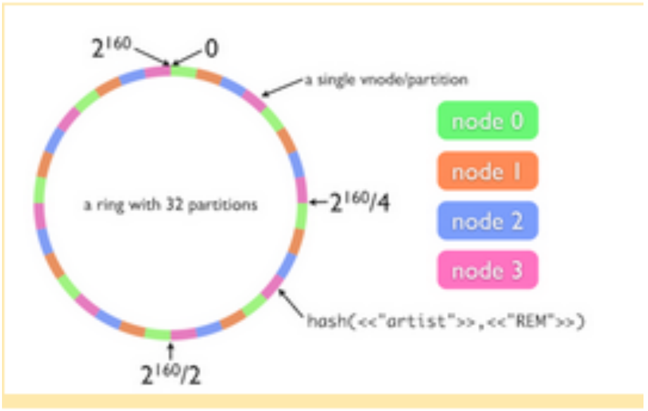


Figure 1: Scalability
[1]

consistent hashing, which is to introduce virtual nodes. In fact, I introduce a middle layer decoupling. The virtual nodes fall on the circle on average, and then the mapping of the actual nodes is linked to certain virtual nodes, indicating that I am here. Each physical node is actually responsible for the data range of these virtual nodes, so as to achieve the role of load balancing.[13]

2.2 Reliability

In the design of Dynamo, in order to ensure disaster recovery, data is replicated to N hosts, where N is the number of redundant copies of data. Each node in Dynamo has a module called a request coordinator[12]. It receives some After the data key value K, it will be replicated to N-1 nodes behind the circle to ensure that the key value K has N copies. Therefore, in fact, each node in Dynamo stores both the data it receives for itself and stores other nodes' replica data.

2.3 Consistency

In the case where there are N redundant copies of data, to ensure strong consistency, we need to wait for all copies to be written before returning to the client to write successfully, but this will reduce performance. Dynamo allows the user to set at least W copies to be written before returning, and when reading, the value needs to be read from R copies to return. Therefore, as long as W + R> N, the correct value can be guaranteed to be read.

But there is a problem in how to determine which of the returned R values is the latest, that is, each data should have a version information. In order to solve this problem, Dynamo introduced the concept of vector clock and used vector clock to record version

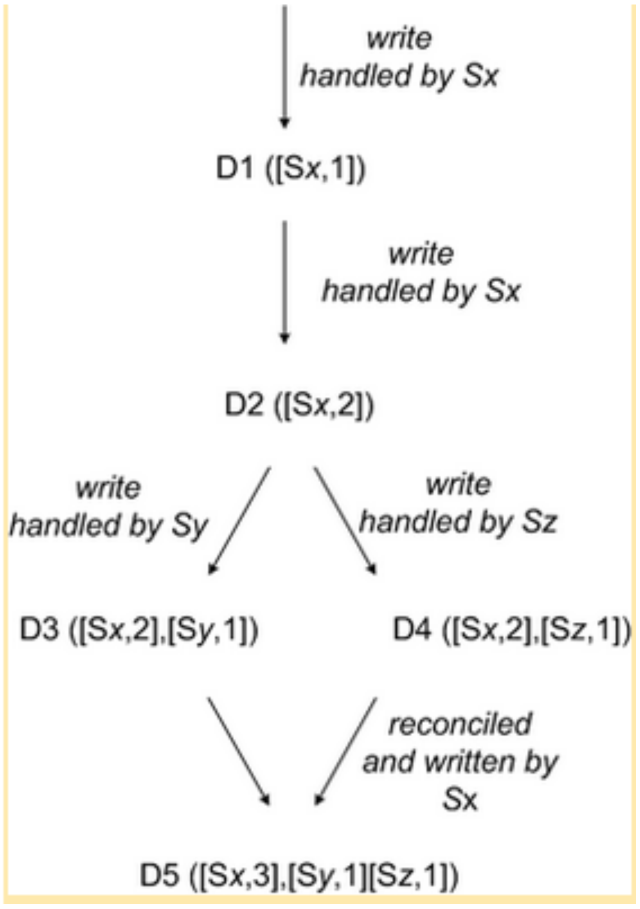


Figure 2: Consistency
[6]

information. Simply put, every time a write operation is performed, the written copy will add an updater and version number vector group <updater, version> for this data change as the version information, which will also be brought in the subsequent copy process This part of the information. When a read operation occurs, multiple versions are returned, and the client's business layer resolves the conflict and merges the various versions. Of course, the client can also choose the simplest strategy, that is, the most recent write overwrites the previous one [6]

3 TECHNIQUES IN IMPLEMENTATION OF DYNAMODB

3.1 DynamoDB Batch Write

DynamoDB doesn't allow writing CSV directly. Instead, you can insert up to 25 items at a time using batch write Query. We had a huge DB of over 3.1 millions rows. In order to seed DB with these rows,we used BatchWrite Query. We first loaded CSV in memory, created batches of 25 items per request and hit AWS using AWS batch write command and added these rows to table [11]

3.2 Partition Key and Sort Key

DynamoDB uses the partition key value as input to the internal hash function. The output from the hash function determines the partition (physical storage inside DynamoDB) where the item will be stored. All items with the same partition key are stored together in the sort order of the sort key value.[9]

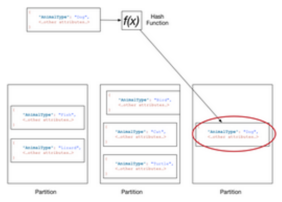


Figure 3: Patyion key and Sort Key

In a Figure with a partition key and a sort key, two items can have the same partition key value, but the two items must have different sort key values.

3.3 GetItem, Query and Scan command

These three operations are query operations for dynamoDB, and the efficiency is: getItem> query> scan

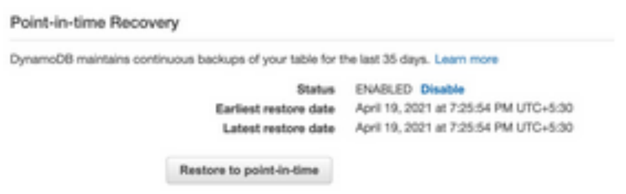
getItem is like what it sounds, it gets an item from the hash based on the primary key. It's very fast, but it applies only to a single query with the given primary key. We used it to pass tweet id and get all info about that tweet id (Body, postDate, writer etc.)

Query is the most common way. It returns information about the table, including the current status of the table, when it was created, the primary key schema, and any indexes on the table. We used this command to show information about the table. as you can see in the video we are showing Tweets Count, Table Name, Table Status, etc.. basically, it gets metadata about the table.

Scan is a full table scan, and it is the slowest one. For non hashed or non Primary keys, We use this command. A Scan operation in Amazon DynamoDB reads every item in a table or a secondary index. We used this command to Query DB to find the tweets which contained the company Stock Name. eg check tweet which contains $AAPL in tweet body. we used Filter Expression for SCAN to filter all tweets which contain passed substring [5].

3.4 Backup and Recovery Service

With on-demand backup, you can create full backups of Amazon DynamoDB tables for data archiving, which will help you meet corporate and government regulatory requirements[7]. Regardless of whether the amount of data contained in the table is a few MB or hundreds of TB, you can back it up without affecting the performance and availability of the production application. Regardless of the size of your table, the backup will be completed in a few seconds, so you don't need to worry about backup schedules or long-running processes. In addition, all backups will be automatically encrypted,



Point-in-time Recovery

DynamoDB maintains continuous backups of your table for the last 35 days. Learn more

Status	ENABLED Disable
Earliest restore date	April 19, 2021 at 7:25:54 PM UTC+5:30
Latest restore date	April 19, 2021 at 7:25:54 PM UTC+5:30

Figure 4: BackUp and Recovery Time

cataloged, easy to find and keep until you explicitly delete them [4].

Point-in-time recovery (PITR) can continuously back up your DynamoDB table data. Once enabled, DynamoDB will maintain incremental backups of the last 35 days for your tables until you explicitly disable them[10].

3.5 Multi-Region Replication

With global tables you can specify the AWS Regions where you want the table to be available. DynamoDB performs all of the necessary tasks to create identical tables in these Regions and propagate ongoing data changes to all of them. DynamoDB implements multi-master table writes in the background to ensure that the last write to a specific item prevails. When using a global table, each item will contain a timestamp attribute to indicate the time of the most recent write. Updates are propagated asynchronously to other regions via DynamoDB Streams, usually within one second[7]. Figure 7

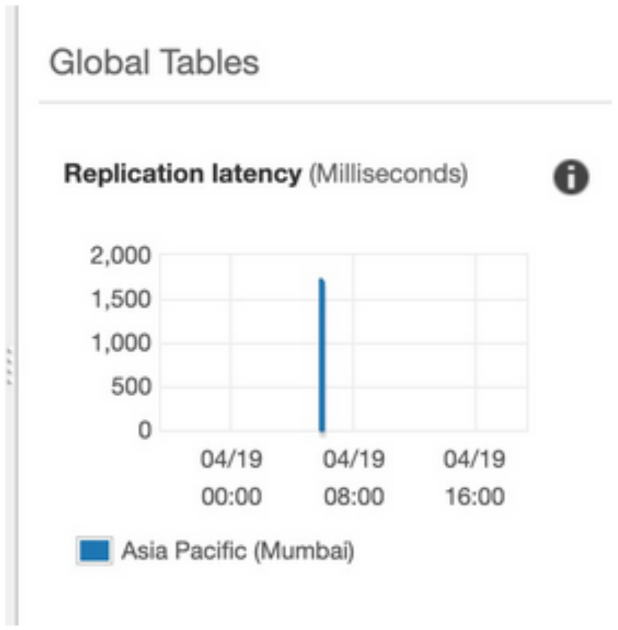


Figure 5: Replication

3.6 DAX

Some use cases often require a high request rate (up to millions of requests per second), low and predictable latency, and strong reliability. In order to cope with the surge in read volume or to ensure sub-millisecond read latency, we can turn to the caching mechanism provided by DynamoDB Accelerator (DAX)

DAX does all the heavy lifting required to add in-memory acceleration to your DynamoDB tables, without requiring developers to manage cache invalidation, data population, or cluster management [2]

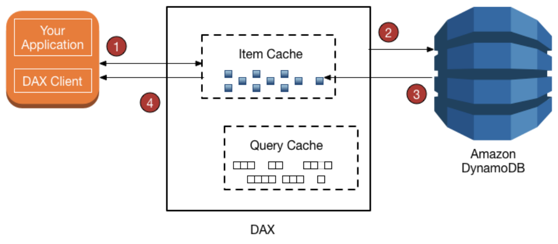


Figure 6: DAX
[2]

3.7 Auto Scaling

In certain applications we can't predict exactly what the usage will be. DynamoDB Auto Scaling provides a mechanism whereby we can allocate resources based on the usage. In Auto Scaling we provide the upper, lower capacity of the resources and target utilization by means of which we can fluctuate the resources based on the usage. Autoscaling policy can be applied to table or a global secondary index

Consider an example of a table where we have an auto scaling policy enabled. DynomoDB monitors this table using CloudWatch and whenever the tables consumed capacity increases or decreases for a specified length of time it triggers the autoscaling on that table [8].

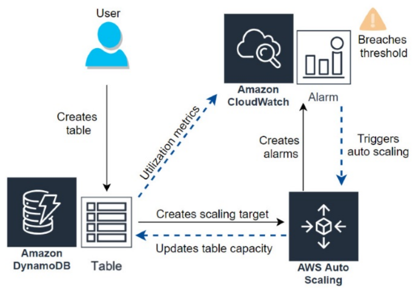


Figure 7: Autoscaling
[3]

4 CONCLUSION

With Amazon DynamoDB, there are no servers to provision, patch, or manage, and no software to install, maintain, or operate. DynamoDB automatically scales tables to adjust for capacity and maintains performance with zero administration. Availability and fault tolerance are built in, eliminating the need to architect your applications for these capabilities.

REFERENCES

[1] Amazon. [n.d.]. Amazon DynamoDB. https://aws.amazon.com/dynamodb/?nc1=h_ls. Accessed March. 27, 2021.
[2] Amazon. [n.d.]. Amazon DynamoDB Accelerator. https://aws.amazon.com/dynamodb/dax/ Accessed April. 17, 2021.
[3] Amazon. [n.d.]. Amazon DynamoDB auto scaling: Performance and cost optimization at any scale. https://aws.amazon.com/blogs/database/amazon-dynamodb-auto-scaling-performance-and-cost-optimization-at-any-scale/ Accessed April. 17, 2021.
[4] Amazon. [n.d.]. Backup and Restore. https://aws.amazon.com/dynamodb/backup-restore/ Accessed April. 17, 2021.
[5] Amazon. [n.d.]. Best Practices for Querying and Scanning Data. https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/bp-query-scan.htm Accessed April. 10, 2021.
[6] Amazon. [n.d.]. DAX and DynamoDB Consistency Models. https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/DAX.consistency.html Accessed April. 17, 2021.
[7] Amazon. [n.d.]. Global Tables: Multi-Region Replication with DynamoDB. https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/GlobalTables.html Accessed April. 17, 2021.
[8] Amazon. [n.d.]. Managing Throughput Capacity Automatically with DynamoDB Auto Scaling. https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/AutoScaling.html Accessed April. 17, 2021.
[9] Amazon. [n.d.]. Partitions and Data Distribution. https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/HowItWorks.Partitions.html Accessed March. 27, 2021.
[10] Amazon. [n.d.]. Using the DynamoDB Document Client. https://docs.aws.amazon.com/sdk-for-javascript/v2/developer-guide/dynamodb-example-document-client.html Accessed April. 17, 2021.
[11] Mohd Belal. [n.d.]. Writing Millions of records in DynamoDB. https://www.codementor.io/@mohdbelal/writing-millions-of-records-in-dynamodb-14zgyszj41 Accessed April. 10, 2021.
[12] programmersought.com. [n.d.]. Rereading Amazon Dynamo felt paper. https://programmersought.com/article/85692546759/ Accessed March. 27, 2021.
[13] All things Distributed. [n.d.]. Amazon's Dynamo. https://www.allthingsdistributed.com/2007/10/amazons_dynamo.html. Accessed March. 27, 2021.