NOSQL Databases & DynamoDB

DynamoDB - Architecture

DynamoDB is a NoSQL fully managed Database-as-a-Service (DBaaS) product available within AWS. This is the traditional serverless DB route in AWS

- uses Key/Value &/or Document data. "Wide column DB"
- no self-managed servers or infrastructure
- Highly resilient and can be optionally Globally resilient
- Single digit millisecond speeds -- SSD based
- manual or auto scaling or on-demand (set and forget)
- RCU - read capacity units, WCU - write capacity units

DynamoDB - Tables

- Table is a grouping of ITEMS with the same PRIMARY key. Item is like a row in a traditional DB
    - item max size 400KB. item can have all, none, or some of the attibutes/columns in an item(row)
- Table has two primary keys to pick from:

1. Partition key. made up of just a partition key
2. Composite primary key. made up of a partition key and a sort key

DynamoDB - Backups

Two types of backups:

1. On-demand backup. Full copy of table retained until removed
2. Point-in-time Recovery. Not enabled by default. 35 day window to restore to 1 second granularity, PER table restores EXAM - NoSQL mentioned? DynamoDB EXAM - Key/Value DB mentioned? DynamoDB EXAM - Relational Data/DB? NOT DynamoDB

DynamoDB - Operations, Consistency and Performance

Key elements of READS and WRITES to DynamoDB and step through how the QUERY AND SCAN operations work.

- When you create a DynamoDB table, you choose from two capacity modes: on-demand and provisioned -- On-Demand: If you need low admin overhead or traffic is unknown/unpredictable. Price per million R or W units. More expensive -- Provisioned: RCU/WCU capacity set per table basis
- 1 RCU = 1 x 4KB read ops per second, 1 WCU is 1 x 1KB write ops per second -- Every table has RCU/WCU Burst Pool (300 seconds)

DynamoDB - Query and Scan

- Query. One way to retrieve data. Accepts a single PK value and optionally an SK or range.
- Scan. Least efficient but most flexible way to get data in DynamoDB. Scans through table item-by-item - consumed capacity is all items read/scanned

DynamoDB - Consistency model in DynamoDB

In DynamoDB, each piece of data is replicated into different AZs known as Storage Nodes, and one is the Leader storage node. Leader storage nodes receive the reads/writes and write updates are replicated by leader to other nodes

- Eventually Consistent Reads. You are not guaranteed to see the most updated data if you're send to a node that hasn't received the replicated new data yet. Cheaper option
- Strongly Consistent Reads. Always reads from Leader node to guarantee data up to date NOTE: Not every app can tolerate eventual consistency (some things MUST be accurate)

Calculate WCU

Problem: Need to store 10 items per second, 2.5K average size per item

1. Round up average size to nearest whole number. 2.5kB/1kB -> 3
2. Multiply item size x # items/sec. 3 x 10 = 30 = 30 WCU required

Calculate RCU

Problem: Need to retrieve 10 items per second, 2.5K average size per item

1. Divide Item size / 4 then round up. 2.5K/4kB rounded up is 1.
2. Multiply item size x # items/sec. 1 x 10 = 10 = 10 RCU required for Strongly Consistent. Eventual is 50% cost, so 5RCU required for Eventually Consistent Reads

DynamoDB - Local and Global Secondary Indexes

Local Secondary Indexes (LSI) and Global Secondary Indexes (GSI) allow for an alternative presentation of data stored in a base table.

- Indexes improve the efficiency of operations in DynamoDB
- LSI allow for alternative Sort Key's (SAME PK) whereas with GSIs you can use alternative PK and SK.
    - LSIs only created at Table creation
- GSIs can be created any time. 20 limit of GSIs per base table. Uses alternate PK and SK. GSIs are ALWAYS eventually consistent, so to use GSI your tables needs to be able to cope with eventualy consistency NOTE: Use GSI as default and LSI only when STRONG consistency index is required

DynamoDB - Streams & Lambda Triggers

DynamoDB Streams are a 24 hour rolling window of time ordered list of changes to ITEMS in a DynamoDB table

- Streams have to be enabled on a per table basis , and have 4 view types:

1. KEYS_ONLY. Only keys of item that changed
2. NEW_IMAGE. New item after change
3. OLD_IMAGE. Old item before change
4. NEW_AND_OLD_IMAGES. Both old and new states

- Lambda can be integrated to provide trigger functionality - invoking when new entries are added on the stream.

DDB - Triggers

- Streams are the foundation of Triggers
- Triggers allow fo actions to take place in the event of data change, use can use Lambda to perform a compute action in response. View type data is send with trigger (e.g. notification of new message can be send to users etc.)

DynamoDB - Global Tables

DynamoDB Global Tables provides multi-master global replication of DynamoDB tables which can be used for performance, High Avail or Disaster Recovery/Biz Continuity reasons.

- GLOBAL table and then multiple regions with tables becoming REPLICA tables
- LAST WRITER WINS is used for conflict resolution; the most recent Write is what is replicated to other tables
- Global app must tolerate eventual consistency. However, region reads in the same region as writes are strongly consistent

DynamoDB - Accelerator (DAX)

DynamoDB Accelerator (DAX) is an in-memory cache designed specifically for DynamoDB. It should be your default choice for any DynamoDB caching related questions.

- Global table has sub-second replication between table replicas
- DAX is less overhead than normal caching
- Can scale up or out (bigger or more DAX instances)
- Supports "write-through"
- Good for read-heavy work loads

DynamoDB - TTL

Amazon DynamoDB Time to Live (TTL) allows you to define a per-item timestamp to determine when an item is no longer needed.

- Shortly after the date and time of the specified timestamp, DynamoDB deletes the item from your table without consuming any write throughput.
- TTL is provided at no extra cost as a means to reduce stored data volumes by retaining only the items that remain current for your workload’s needs.

Amazon Athena

Amazon Athena is serverless querying service which allows for ad-hoc questions where billing is based on the amount of data consumed.

- Athena is an underrated service capable of working with unstructured, semi-structured or structured data.

- tl;dr take data stored in S3 and perform ad-hoc queries on that data. Pay only for the data consumed while running query (and s3 storage)
- Uses process called "schema-on-read", a table-like translation --> original data on S3 never changes !!
- Usefull when loading or transformation of data isn't needed or desired (just query data) EXAM - Best option for querying AWS logs; VPC flow logs, CloudTrail, ELB Logs, cost reports, Glue, Web Server, etc
- Athena Federated Query -- a way for Athena to query outside S3
- Athena = occasional querries because you don't need to load data (stream is querried, no infrastructure (serverless). Redshift is not ad-hoc (not serverless, data is needed to load)

Elasticache

Elasticache is a managed in-memory cache which provides a managed implementation of the redis or memcacheD engines.

- Useful for `READ-HEAVY workloads` that demand low latency
- scaling reads in a cost effective way: in-memory database for high performance (reduces database workloads and costs (less reads))
- allows for externally hosted user session state (if 1 EC2 instance fails another will take over but user session is kept alive, user doesn't notice)

Two engines are supported:

1. Redis. supports advanced data structures (lists, sets, etc). Multi-AZ. Backup and restore. Transactions
2. Memcached. Supports simple data structures (strings). No backups. Multi-threaded EXAM - Can store Session Data (for stateless servers) EXAM - Elasticashe requires application code changes to implement (handle caching)

Redshift Architecture

Redshift is a column based, petabyte-scale, data warehousing product within AWS

- It's designed for OLAP (Online Analytical Processing / column based) products within AWS/on-premises to add data to for long term processing, aggregation and trending. (Not OLTP which is row/transaction; inserts/modifies/deletes)
- Redshift Spectrum. Direct Query S3 without having to load it into redshift in advance
- Federated Query. Query data in remote data sources
- Redshift is server based (not serverless)
- EXAM: Redshift has Enhanced VPC Routing:
    - VPC networking for advanced networking control can be used --> can be controlled using Security Groups, Network Access Contorl Lists and use custom DNS, VPC gateways or other gateways to reach AWS external services.
- AZ specific (runs in 1 AZ and uses nodes architecture: 1 leader node and different compute nodes)
- automatic backups to S3 are done to prevent data loss (every 8 hours or every 5GB, whatever happens first)

Redshift DR and Resilience

- RedShift only exists in 1 AZ. There are a number of recovery features:
- snapshots:
    - Either automatic (1-35 day retention)
    - or manual (no retention period).
    - Since data is backed up to S3, you have S3 protections against failure
    - snapshots can be sent to other regions for disaster recovery with a separate configurable retention period
- SERVER based