Analyzing the Scope, Functionality, and Challenges of Neptune: A Graph Base Database Built For the Cloud

[1]G.Sathish, [2]C.Deepa, [3]C.Nandhini

Abstract— Neptune powers graph use cases such as recommendation engines, fraud detection, knowledge graphs, drug discovery, and network security. Neptune is a fast, reliable, fully managed graph database service that makes it easy to build and run applications that work with highly connected datasets. The core of Neptune is a purpose-built, high performance graph database engine that is optimized for storing billions of relationships and querying the graph with milliseconds latency. Neptune supports the popular graph query languages Apache Tinker Pop Gremlin and W3C's SPARQL, allowing you to easily build queries that efficiently navigate highly connected datasets.

———————— ◆ ————————

1 Introduction

Neptune is a fully managed graph database product byAmazon.com. It is used as a web service and is part of Amazon Web Services. It was announced on November 29, 2017. Neptune supports popular graph models Property Graph and W3C's RDF, and their respective query languages Apache Tinker Pop Gremlinand SPARQL, allowing you to easily build queries that efficiently navigate highly connected datasets.

Neptune powers graph use cases such as recommendation engines, fraud detection, knowledge graphs, drug discovery, and network security. Neptune is highly available, with read replicas, point-in-time recovery, continuous backup to Amazon S3, and replication across Availability Zones. Neptune provides data security features, with support for encryption at rest and in transit. Neptune is fully managed, so you no longer need to worry about database management tasks like hardware provisioning, software patching, setup, configuration, or backups. Graph Database Graph databases like Amazon Neptune are purpose-built to store and navigate relationships. Graph databases have advantages over relational databases for certain use cases—including social networking, recommendation engines, and fraud detection—when you want to create relationships between data and quickly query these relationships. There are a number of challenges to building these types of applications using a relational database. It requires you to have multiple tables with multiple foreign keys. The SQL queries to navigate this data require nested queries and complex joins that quickly become unwieldy. And the queries don't perform well as your data size grows over time.

2 Scopes

We include services in the scope of our compliance efforts based on the expected use case, feedback and demand. If a service is not currently listed as in scope of the most recent assessment, it does not mean that you cannot use the service. It is part of the shared responsibility for your organization to determine the nature of the data. Based on the nature of what you are building on AWS, you should determine if the service will process or store customer data and how it will or will not impact the compliance of your customer data environment. We encourage you to discuss your workload objectives and goals with your AWS account team; they will be able to evaluate your proposed use case and architecture, and how our security and compliance processes overlay that architecture.

3 Functionality

3.1 High Performance and Scalability

Amazon Neptune is a purpose-built, high-performance graph database engine. Neptune efficiently stores and navigates graph data, and uses a scale-up, in-memory optimized architecture to allow for fast query evaluation over large graphs. With Neptune, you can use either Gremlin or SPARQL to execute powerful queries that are easy to write and perform well.

3.1.1 Easy Scaling of Database Compute Resources

With a few clicks in the AWS Management Console, you can scale the compute and memory resources powering your production cluster up or down by creating new replica instances of the desired size, or by removing instances. Compute scaling operations typically complete in a few minutes.

3.1.2 Storage that Automatically Scales

Amazon Neptune will automatically grow the size of your database volume as your database storage needs grow. Your volume will grow in increments of 10 GB up to a maximum of 64 TB. You don't need to provision excess storage for your database to handle future growth.

3.1.3 Low Latency Read Replicas

Increase read throughput to support high volume

- [1]G.Sathish, Assistant Professor, Department of Computer Applications, Priyadarshini engineering college, Vaniyambadi, Email: satpec@gmail.com
- [2]C.Deepa, II-MCA, Department of Computer Applications, Priyadarshini engineering college, Vaniyambadi, Email: deepacapr20@gmail.com
- [3]C.Nandhini, III-MCA, Department of Computer Applications, Priyadarshini engineering college, Vaniyambadi, Email: nandhinicapr20@gmail.com

application requests by creating up to 15 database read replicas. Amazon Neptune replicas share the same underlying storage as the source instance, lowering costs and avoiding the need to perform writes at the replica nodes. This frees up more processing power to serve read requests and reduces the replica lag time – often down to single digit milliseconds. Neptune also provides a single endpoint for read queries so the application can connect without having to keep track of replicas as they are added and removed.

3.1.4 High Availability and Durability Instance Monitoring and Repair

The health of your Amazon Neptune database and its underlying EC2 instance is continuously monitored. If the instance powering your database fails, the database and associated processes are automatically restarted. Neptune recovery does not require the potentially lengthy replay of database redo logs, so your instance restart times are typically 30 seconds or less. It also isolates the database buffer cache from database processes, allowing the cache to survive a database restart.

3.1.5 Multi-AZ Deployments with Read Replicas

On instance failure, Amazon Neptune automates failover to one of up to 15 Neptune replicas you have created in any of three Availability Zones. If no Neptune replicas have been provisioned, in the case of a failure, Neptune will attempt to create a new database instance for you automatically.

3.1.6 Fault-tolerant and Self-healing Storage

Each 10GB chunk of your database volume is replicated six ways, across three Availability Zones. Amazon Neptune uses fault-tolerant storage that transparently handles the loss of up to two copies of data without affecting database write availability and up to three copies without affecting read availability. Neptune's storage is also self-healing; data blocks and disks are continuously scanned for errors and replaced automatically.

4 Automatic, Continuous, Incremental Backups and Point-in-time Restore

Amazon Neptune's backup capability enables point-in-time recovery for your instance. This allows you to restore your database to any second during your retention period, up until the last five minutes. Your automatic backup retention period can be configured up to thirty-five days. Automated backups are stored in Amazon S3, which is designed for 99.999999999% durability. Neptune backups are automatic, incremental, and continuous and have no impact on database performance.

4.1 Database Snapshots

Database Snapshots are user-initiated backups of your instance stored in Amazon S3 that will be kept until you explicitly delete them. They leverage the automated incremental snapshots to reduce the time and storage required. You can create a new instance from a Database Snapshot whenever you desire.

4.2 Open Graphics Supports Property Graph Apache Tinker Pop Gremlin

Property Graphs are popular because they are familiar to developers that are used to relational models. Gremlin traversal language provides a way to quickly traverse Property Graphs. Amazon Neptune supports the Property Graph model using the open source Apache TinkerPop Gremlin traversal language and provides a Gremlin Websockets server that supports TinkerPop version 3.3. With Neptune, you can quickly build fast Gremlin traversals over property graphs. Existing Gremlin applications can easily use Neptune by changing the Gremlin service configuration to point to a Neptune instance.

5 Supports W3C's Resource Description Framework (RDF) 1.1 and SPARQL 1.1

RDF is popular because it provides flexibility for modeling complex information domains. There are a number of existing free or public datasets available in RDF including Wikidata and PubChem, a database of chemical molecules. Amazon Neptune supports the W3C's Semantic Web standards of RDF 1.1 and SPARQL 1.1 (Query and Update), and provides an HTTP REST endpoint that implements the SPARQL Protocol 1.1. With Neptune, you can easily use the SPARQL endpoint for both existing and new graph applications.

5.1 Network Isolation

Amazon Neptune runs in Amazon VPC, which allows you to isolate your database in your own virtual network, and connect to your on-premises IT infrastructure using industry-standard encrypted IPsec VPNs. In addition, using Neptune's VPC configuration, you can configure firewall settings and control network access to your database instances.

6 Resource-Level Permissions

Amazon Neptune is integrated with AWS Identity and Access Management (IAM) and provides you the ability to control the actions that your AWS IAM users and groups can take on specific Neptune resources including Database Instances, Database Snapshots, Database Parameter Groups, Database Event Subscriptions, and Database Options Groups. In addition, you can tag your Neptune resources, and control the actions that your IAM users and groups can take on groups of resources that have the same tag (and tag value). For example, you can configure your IAM rules to ensure developers are able to modify "Development" database instances, but only Database Administrators can modify and delete "Production" database instances.

7 Encryption

Amazon Neptune allows you to encrypt your databases using keys you create and control through AWS Key Management Service (KMS). On a database instance running with Neptune encryption, data stored at rest in the underlying storage is encrypted, as are the automated backups, snapshots, and replicas in the same cluster. Neptune uses TLS (AES-256) to secure data in transit.

8 Advanced Auditing

Amazon Neptune allows you to log database events with minimal impact on database performance. Logs can later be analyzed for database management, security, governance, regulatory compliance and other purposes. You can also monitor activity by sending audit logs to Amazon Cloud Watch.

9 FULLY MANAGED

Easy to Use

Getting started with Amazon Neptune is easy. Just launch a new Neptune database instance using the AWS Management Console. Neptune database instances are pre-configured with parameters and settings appropriate for the database instance class you have selected. You can launch a database instance and connect your application within minutes without additional configuration. Database Parameter Groups provide granular control and fine-tuning of your database.

10 Fast Parallel Bulk Data Loading

Property Graph Bulk Loading

Amazon Neptune supports fast, parallel bulk loading for Property Graph data that is stored in S3. You can use a REST interface to specific the S3 location for the data. It uses a CSV delimited format to load data into the Nodes and Edges. See the Neptune Property Graph bulk loading documentation for more details.

11 RDF Bulk Loading

Amazon Neptune supports fast, parallel bulk loading for RDF data that is stored in S3. You can use a REST interface to specific the S3 location for the data. The N-Triples (NT), N-Quads (NQ), RDF/XML, and Turtle RDF 1.1 serializations are supported. See the Neptune RDF bulk loading documentation for more details.

12 Cost-Effectiveness

12.1 Pay Only for What You Use

There is no up-front commitment with Amazon Neptune; you simply pay an hourly charge for each instance that you launch. And, when you're finished with a Neptune database instance, you can easily delete it. You do not need to over-provision storage as a safety margin, and you only pay for the storage you actually consume. To see more details, visit the Neptune Pricing page.

12.2 Challenges

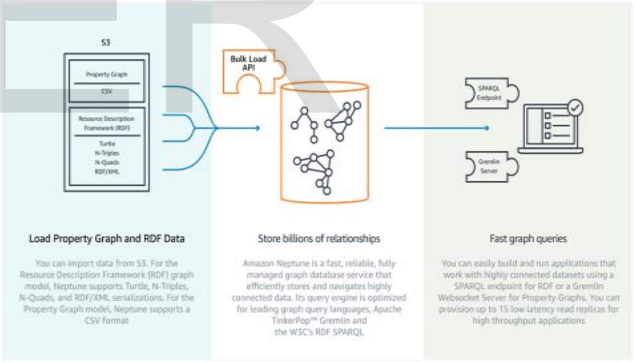
Amazon has launched Neptune, a graph database that can be used to create relationship graphs of highly connected datasets. Neptune is a fully managed service, and is among the slew of announcements made at AWS reinvent.

While the announcements about the new product make no mention of the history of the product, it seems to be based on Blazegraph, a well-known graph database that rumor has it was acquired by Amazon last year. The domain was certainly acquired by Amazon, and a number of key Blazegraph developers now work at Amazon, including Brad Bebee, CEO of Blazegraph until this year, and now giving sessions on Neptune at Re:Invent. The name Neptune may well have been decided on because Janus and Titan are both big graph databases.

Amazon describes Neptune as being based on a purpose-built, high-performance graph database engine optimized for storing billions of relationships and querying the graph with milliseconds latency. Recommended uses for Neptune include recommendation engines, fraud detection, knowledge graphs, drug discovery, and network security.

Amazon Neptune supports two open standards for describing and querying graphs - Apache TinkerPop3 style Property Graphs queried with Gremlin, and Resource Description Framework (RDF) queried with SPARQL. Gremlin is a graph traversal language where a query is a traversal made up of discrete steps following an edge to a node, while SPARQL is a declarative language based on Semantic Web standards from W3C. Data can be imported from S3, and for the Resource Description Framework (RDF) graph model, Neptune supports Turtle, N-Triples, N-Quads, and RDF/XML serializations. Property graphs are limited to CSV.

12.3 How It Works



You can import data from S3. For the Resource Description Framework (RDF) graph model, Neptune supports Turtle, N-Triples, N-Quads, and RDF/XML serializations. For the Property Graph model, Neptune supports a CSV format

Amazon Neptune is a fast, reliable, fully managed graph database service that efficiently stores and navigates highly connected data. Its query engine is optimized for leading graph query languages, Apache TinkerPop™ Gremlin and the W3C's RDF SPARQL

You can easily build and run applications that work with highly connected datasets using a SPARQL endpoint for RDF or a Gremlin Websocket Server for Property Graphs. You can provision up to 15 low latency read replicas for high throughput applications

12.4 Graph Database Uses

Graph databases are useful for connected, contextual, relationship-driven data. An example is modeling social media data, as shown in the previous section. Other examples include recommendation engines, driving directions (route finding), logistics, diagnostics, and scientific data analysis in fields like neuroscience.

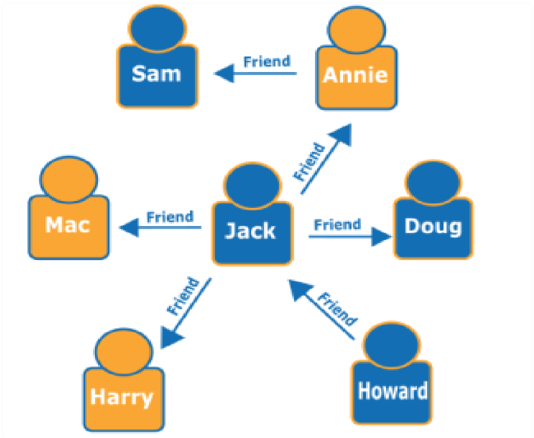
- ➢ Fraud Detection
- ➢ Recommendation Engines
- ➢ Knowledge Graphs
- ➢ Life Sciences
- ➢ Network / IT Operations

12.5 Graph Queries And Traversals

Neptune supports two different graph query languages: Gremlin (Apache TinkerPop3) and SPARQL (SPARQL 1.1).

- Gremlin is a graph traversal language and, as such, a query in Gremlin is a traversal made up of discrete steps. Each step follows an edge to a node.
- SPARQL is a declarative query language based on graph pattern-matching standardized by the W3C.

Given the following graph of people (nodes) and their relationships (edges), you can find out who the "friends of friends" of a particular person are—for example, the friends of Howard's friends.



Looking at the graph, you can see that Howard has one friend, Jack, and Jack has three friends: Annie, Harry, and Mac. This is a simple example with a simple graph, but these types of queries can scale in complexity, dataset size, and result size.

12.6 Working with Backups

Preview Release Amazon Neptune Preview is available only to white listed customers. To request access to Neptune, see the information on the Amazon Neptune Preview page. Amazon Neptune creates and saves automated backups of your DB instance. It creates a storage volume

Snap shot of your DB instance, backing up the entire DB instance and not just individual databases. Neptune creates automated backups during the backup window of your DB instance. It saves the backups according to the backup retention period that you specify. If necessary, you can recover your database to any point in time during the backup retention period. Your DB instance must be in the ACTIVE state for automated backups to occur. If your database is in another state, for example STORAGE_FULL, automated backups don't occur. We can also back up our DB instance manually by creating a DB snapshot. For more information about creating a DB snapshot. We can copy both automatic and manual DB snapshots, and share manual DB snapshots.

12.7 Backup Storage

Our Neptune backup storage for each AWS Region is composed of the automated backups and manual DB snapshots for that Region. our backup storage is equivalent to the sum of the database storage for all instances in that Region. Moving a DB snapshot to another Region increases the backup storage in the destination Region. All automated backups are deleted when you delete a DB instance. After you delete a DB instance, the automated backups can't be recovered. If you choose to have Neptune create a final DB snapshot before it deletes your DB instance, you can use that to recover your DB instance. Manual snapshots are not deleted.

12.8 Backup Window

Automated backups occur daily during the preferred backup window. If the backup requires more time than allotted to the backup window, the backup continues after the window ends, until it finishes. The backup window can't overlap with the weekly maintenance window for the DB instance.

13 Amazon Neptune User Guide

13.1 Working with Backups

During the automatic backup window, storage I/O might be suspended briefly while the backup process initializes (typically under a few seconds). You might experience elevated latencies for a few minutes

during backups for Multi-AZ deployments. If you don't specify a preferred backup window when you create the DB instance, Neptune assigns a default 30-minute backup window. This window is selected at random from an eight-hour block of time per Region.

13.2 Backup Retention Period

You can set the backup retention period when you create a DB instance. If you don't set the backup retention period, the default backup retention period is seven days if you create the DB instance using the AWS Management Console. For DB clusters, the default backup retention period is one day regardless of how the DB cluster is created.

After you create a DB instance, you can modify the backup retention period. You can set the backup retention period to between 1 and 35 days. we can also set the backup retention period to 0, which disables automated backups. Manual snapshot limits (100 per AWS Region) don't apply to automated backups.

13.3 Disabling Automated Backups

In certain situations, you might want to disable automated backups temporarily; for example, while loading large amounts of data. We highly discourage disabling automated backups because it disables point-in-time recovery. Disabling automatic backups for a DB instance deletes all existing automated backups for the instance. If you disable and then re-enable automated backups, you can only restore starting from the time you re-enabled automated backups. In this example, you disable automated backups for a DB instance named my db instance by setting the backup retention parameter to 0.

13.4 Disabling Automated Backups Using The Console

Follow these steps to use the AWS Management Console to disable automated backups immediately for your DB instance.

13.5 To Disable Automated Backups Immediately

1. Sign in to the AWS Management Console, and open the Amazon Neptune console at https://yukon.aws.amazon.com/neptune?region=us-east-1.

2. In the navigation pane, choose DB Instances, and then choose the DB instance that you want to modify.

3. Choose Instance Actions, and then choose Modify. The Modify DB Instance window appears.

4. For Backup Retention Period, choose 0.

5. Choose Apply Immediately.

6. Choose Continue.

14 Conclusion

Neptune can quickly and easily process large sets of user's profiles and interactions to build social networking applications. Neptune enables highly interactive graph queries with high throughout to bring social features into our applications. Neptune allows us to store relationships between such as customer interests, friends, and purchase history in a graph an quickly query it to make recommendations that are personalized and relevant. We can use relationships to process financial and purchase transactions in near real time to easily detect fraud patterns. Neptune provides a fully managed service to execute fast graph queries to detect that a potential purchaser is using the same email address and credit card as a known fraud case. Neptune helps us build knowledge graph applications

References

[1] https://aws.amazon.com/neptune
[2]https://docs.aws.amazon.com/neptune/latest/userguide/graph-database.html
[3]https://aws.amazon.com/blogs/aws/amazon-neptune-a-fully-managed-graph-database-service
[4]https://aws.amazon.com/nosql/graph
[5]https://www.infoq.com/news/2017/12/amazon-neptune