Introduction to Neo4j

Table of Contents

About this module

The Neo4j Graph Platform enables developers to create applications that are best architected as graph-powered systems that are built upon the rich connectedness of data.

At the end of this module, you should be able to:

- Describe the components and benefits of the Neo4j Graph Platform.

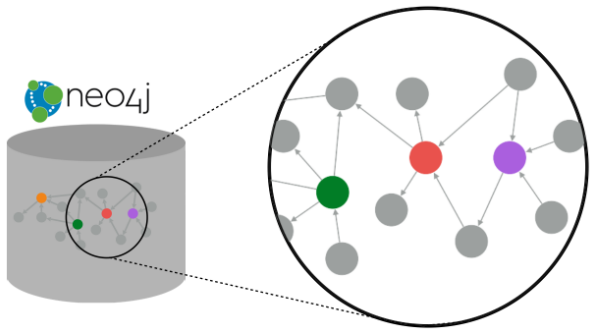
Neo4j Graph Platform

The Neo4j Graph Platform includes components that enable you to develop your graph-enabled application. To better understand the Neo4j Graph Platform, you will learn about these components and the benefits they provide.

Neo4j Database

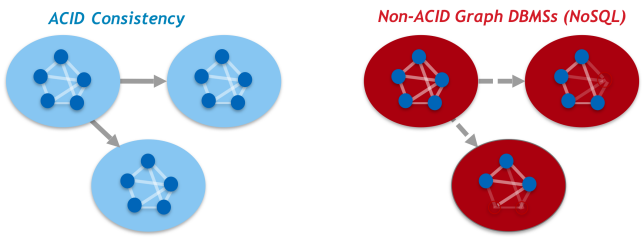
The heart of the Neo4j Graph Platform is the Neo4j Database. The Neo4j Graph Platform includes out-of-the-box tooling that enables you to access graphs in Neo4j Databases. In addition, Neo4j provides APIs and drivers that enable you to create applications and custom tooling for accessing and visualizing graphs.

Neo4j Database: Index-free adjacency



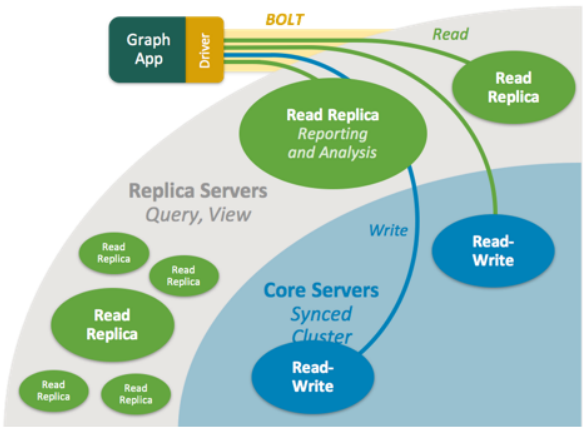
With index free adjacency, when a node or relationship is written to the database, it is stored in the database as connected and any subsequent access to the data is done using pointer navigation which is very fast. Since Neo4j is a native graph database, it supports very large graphs where connected data can be traversed in constant time without the need for an index.

Neo4j Database: ACID (Atomic, Consistent, Isolated, Durable)



ACID is extremely important for an application that requires ACID transactions. If a relationship between nodes is created, not only is the relationship created, but the nodes are updated as connected. All of these updates to the database must all succeed or fail.

Clusters



Neo4j fully supports ACID transactions that update data across locations. For some applications, this is critical. Updates to graphs that span locations must also support ACID in many applications. That is where other NoSQL databases fall short and cannot provide the propagation of transactions across clusters (locations) as Neo4j can.

Graph engine

The Neo4j graph engine is used to interpret Cypher statements and also executes kernel-level code to store and retrieve data, whether it is on disk, or cached in memory. The graph engine has been improved with every release of Neo4j to provide the most efficient access to an application's graph data. There are many ways that you can tune the performance of the engine to suit your particular application needs. Neo4j has worked closely with hardware vendors to ensure that the I/O required by Neo4j is optimal.

Language and driver support

Because Neo4j is open source, you can delve into the details of how the Neo4j Database is accessed, but most developers simply use Neo4j without needing a deeper understanding of the underlying code. Neo4j provides a full stack that implements all levels of access to the database and clustering layer where you can use our published APIs. The language used for querying the Neo4j database is Cypher, an open source language.

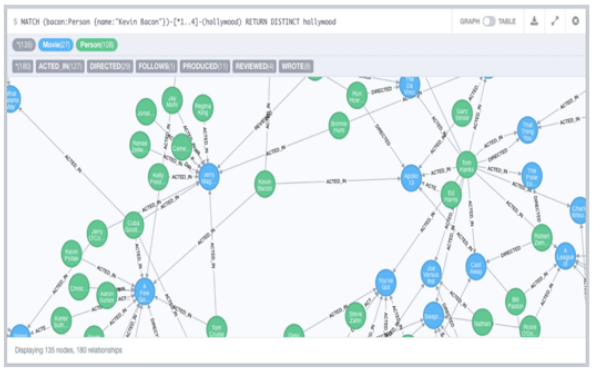
In addition, Neo4j supports Java, JavaScript, Python, C#, and Go drivers out of the box that use Neo4j's bolt protocol for binary access to the database layer. Bolt is an efficient binary protocol that compresses data sent over the wire as well as encrypting the data. For example, you can write a Java application that uses the Bolt driver to access the Neo4j database, and the application may use other packages that allow data integration between Neo4j and other data stores or uses as common framework such as spring.

It is also possible for you to develop your own server-side extensions in Java that access the data in the database directly without using Cypher. The Neo4j community has developed drivers for a number of languages including Ruby, PHP, and R. You can also extend the functionality of Neo4j by creating user defined functions and procedures that are callable from Cypher.

Libraries

Neo4j has a published, open source Cypher library, Awesome Procedures on Cypher (APOC) that contain many useful procedures you can call from Cypher. Another Cypher library is the Graph Algorithms library, shown here, that can help you to analyze data in your graphs. Graph analytics are important because with Neo4j, the technology can expose questions about the data that you never thought to ask. And finally, you can use the GraphQL library (tree-based subset of a graph) to access a Neo4j Database. These libraries are available as plug-ins to your Neo4j development environment, but there are many other libraries that have been written by users for accessing Neo4j.

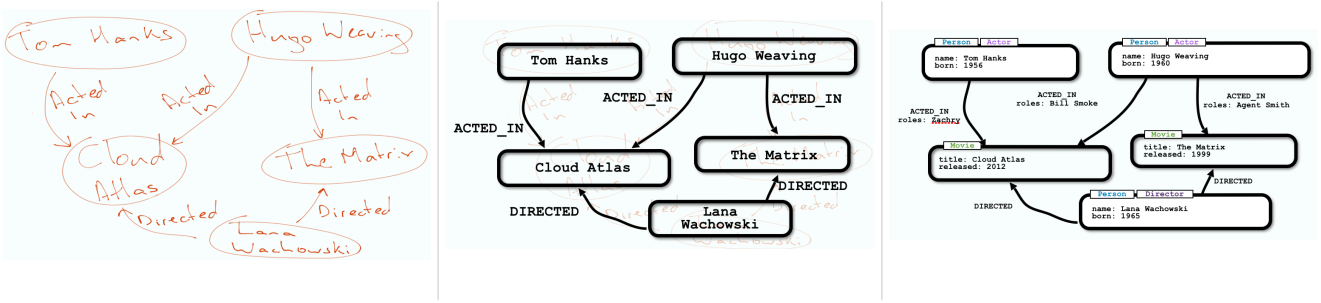
Tools



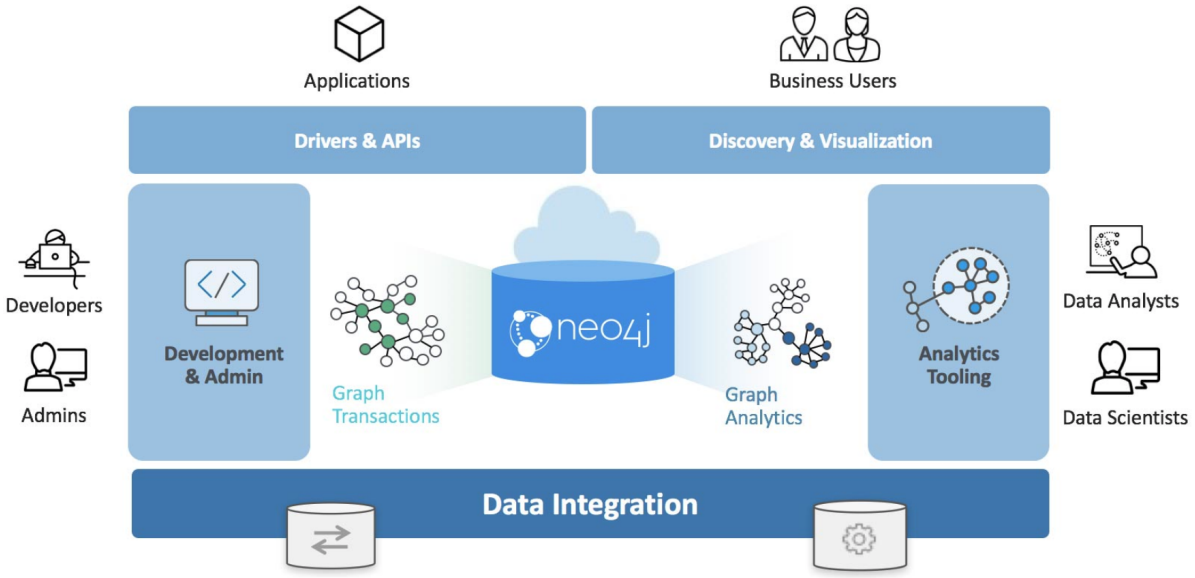
In a development environment, you will use the Neo4j Browser or a Web browser to access data and test your Cypher statements, most of which will be used as part of your application code. Neo4j Browser is an application that uses the JavaScript Bolt driver to access the graph engine of the Neo4j database server. Neo4j also has a new tool called **Bloom** that enables you to visualize a graph without knowing much about Cypher. In addition, there are many tools for importing and exporting data between flat files and a Neo4j Database, as well as an ETL tool.

Whiteboard modeling

With a property graph model, it is very easy to collaborate with colleagues to come up with a whiteboard model of your data that is easy to understand and easy modify. You then use the model to create the nodes, relationships, labels, and properties you will use for your Neo4j data. Even after the graph has been defined and populated with data, it is easy to modify the graph as your application needs change.



Neo4j Graph Platform architecture



Here is the big picture of the Neo4j Graph Platform. The Neo4j Database provides support for graph transactions and analytics. Developers use the Neo4j Desktop, along with Neo4j Browser to develop graphs and test them, as well as implement their applications in a number of languages using supported drivers, tools and APIs. Administrators use tools to manage and monitor Neo4j Databases and clusters. Business users use out-of-the box graph visualization tools or they use custom tools. Data analysts and scientists use the analytics capabilities in the Graph Algorithm libraries or use custom libraries to understand and report findings to the enterprise. Applications can also integrate with existing databases (SQL or NoSQL), layering Neo4j on top of them to provide rich, graph-enabled access to the data.

Check your understanding

Question 1

What are some of the benefits provided by the Neo4j Graph Platform?

Select the correct answers.

- ▢ Database clustering
- ▢ ACID
- ▢ Index free adjacency
- ▢ Optimized graph engine

Question 2

What libraries are included with Neo4j Graph Platform?

Select the correct answers.

- ▢ APOC
- ▢ JGraph
- ▢ GRAPH ALGORITHMS
- ▢ GraphQL

Question 3

What are some of the language drivers that come with Neo4j out of the box?

Select the correct answers.

- ▢ Java
- ▢ Ruby
- ▢ Python
- ▢ JavaScript

Summary

You should now be able to:

- Describe the components and benefits of the Neo4j Graph Platform.