The #1 Platform for Connected Data

WHITE PAPER

The Top 10 Use Cases of Graph Database Technology

Unlock New Possibilities with Connected Data

Jim Webber, Chief Scientist, Neo4j

White Paper

TABLE OF CONTENTS

Introduction 1
Use Case #1: Fraud Detection 2
Use Case #2: Real-Time Recommendation Engine 4
Use Case #3: Knowledge Graphs 6
Use Case #4: Anti-Money Laundering 8
Use Case #5: Master Data Management 10
Use Case #6: Supply Chain Management 12
Use Case #7: Empowering Network & IT Operations Management 14
Use Case #8: Data Lineage 16
Use Case #9: Identity & Access Management 18
Use Case #10: Bill of Materials 20
Conclusion 21

The Top 10 Use Cases of Graph Database Technology

Unlock New Possibilities with Connected Data

Jim Webber, Chief Scientist

Introduction

"Big data" grows bigger every year, but today's enterprise leaders don't only need to manage larger volumes of data. Rather, they critically need to generate insight from their existing data. So how should CIOs and CTOs generate those insights?

To paraphrase Seth Godin, businesses need to stop merely collecting data points, and start connecting them. The relationships between data points matter almost more than the individual points themselves.

Ironically, legacy relational database management systems (RDBMS) are poor at handling relationships between data points. Their tabular data models and rigid schemas make it difficult to add new or different kinds of connections. In order to leverage those data relationships, your organization needs a database technology that stores relationship information as a first-class entity. That technology is a *graph database*.

Graph technology is the future. Not only do graph databases effectively store the relationships between data points, but they're also flexible in adding new kinds of relationships or adapting a data model to new business requirements.

Beyond just the database layer itself, graph technology also includes the emerging field of graph data science for predictive analytics and machine learning as well as graph data visualization for ad hoc data discovery and exploration.

So how might your enterprise leverage graph data technology to generate competitive insights and significant business value from your connected data?

Here are the top 10 use cases of graph technology:

Use Case #1: Fraud Detection

Banks and insurance companies lose billions of dollars every year to fraud.

Traditional methods of fraud detection often fail to minimize these losses since they perform discrete analyses that are susceptible to false positives and negatives. Knowing this, increasingly sophisticated fraudsters develop a variety of ways to exploit the weaknesses of discrete analysis.

Graph technology offers new methods of uncovering fraud rings and other complex scams with a high level of accuracy through advanced contextual link analysis. As a result, graph databases are capable of stopping advanced fraud scenarios in real time.

Why Use Graph Technology for Fraud Detection?

While no fraud prevention measures are perfect, significant improvements occur when you look beyond individual data points to the connections that link them.

Understanding the connections between data and deriving meaning from these links doesn't necessarily mean gathering new data. You can draw significant insights from your existing data by simply reframing the problem as a graph.

Unlike other ways of looking at data, graphs are designed to express relatedness. Graph technology uncovers patterns that are difficult to detect using traditional representations such as tables. An increasing number of companies use graph data technology to solve a variety of connected data problems, including fraud detection.

COMPLEX LINK ANALYSIS

Uncovering fraud rings requires you to traverse data relationships with high computational complexity – a problem that's exacerbated as a fraud ring grows.

DETECTING AND PREVENT FRAUD

To prevent a fraud ring, you need real-time link analysis on an interconnected dataset, from the time a false account is created to when a fraudulent transaction occurs.

EVOLVING AND DYNAMIC FRAUD RINGS

Fraud rings are continuously growing in shape and size, and your application needs to detect these fraud patterns in this highly dynamic and emerging environment.

Example: Fortune 500 Financial Services Company

A Fortune 500 financial services company collects a huge amount of data that needs to be analyzed in real time before a transaction can be approved. While the majority of these requests are instantly approved or denied through an automated fraud detection system, potentially fraudulent requests are submitted to an analyst for manual review.

The analyst has a dedicated transaction review tool encompassing all relevant third-party data. Prior to using Neo4j, analysts had to query a Microsoft SQL Server database to review customer history for fraudulent activity.

"It was taking five minutes or more to run a query," said a product manager for fraud detection solutions at the company. "And since our analysts were having to review 10,000 daily transactions, this wasn't sustainable. Also, a relational database wasn't the right solution to perform link analysis queries so it placed a huge burden on our database."

The company needed to find a more efficient way to analyze the data and save time for both their waiting customers and analysts. Specifically, they needed to decrease the time it took to process fraud detection queries and provide analysts with simple data visualizations.

That's when they found Neo4j, which provided real-time results with connected data and data visualization that enabled analysts to make faster, more accurate decisions. This opened the door for more extensive searches, which the company hopes to expand from four to 10 degrees of separation.

Company analysts also began noticing clusters and relationships between their data, thereby uncovering new, previously unnoticed potential fraud connections. This introduced the possibility of more accurate, real-time fraud ring detection.

Data visualization made possible by Neo4j **cut the manual analyst review time in half**, allowing them to stop fraudulent transactions sooner and reduce wait times for non-fraudulent customers. In addition, they began to see possibilities far beyond their original data visualization use case.

The company is now integrating Neo4j with their real-time decision platform to instantly stop fraudulent transactions and save the company thousands of dollars per day.

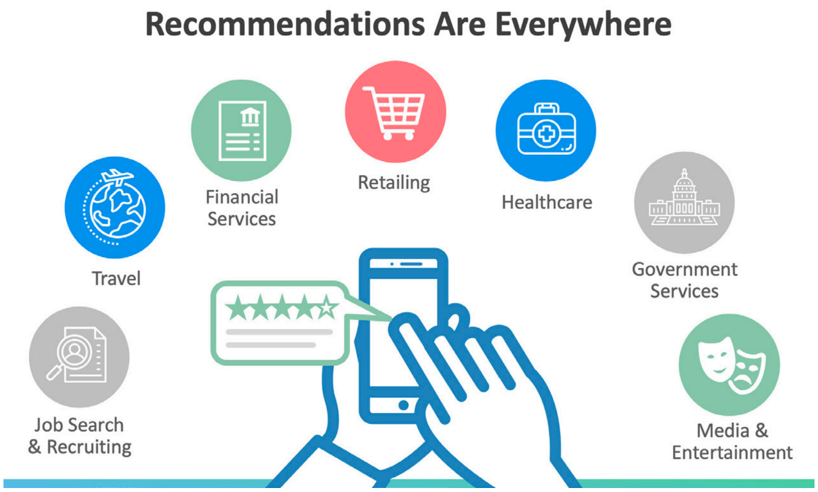
Conclusion

When it comes to graph-based fraud detection, you need to augment your fraud-detection capability with link analysis. That being said, two points are clear:

- As business processes become faster and more automated, the time margins for detecting fraud are narrowing, increasing the need for a real-time solution.
- Traditional technologies are not designed to detect elaborate fraud rings. Graph databases add value through analysis of connected data points.

Graph technology is the ideal enabler for efficient and manageable fraud detection solutions. From fraud rings and collusive groups to educated criminals operating on their own, graph database technology uncovers a variety of important fraud patterns – and all in real time.

Use Case #2: Real-Time Recommendation Engines



Whether your enterprise operates in the retail, services, media or social sector, offering your users highly targeted, real-time recommendations is essential to maximizing customer value and staying competitive. Unlike other business data, recommendations must be inductive and contextual in order to be considered relevant by your end users.

With graph technology, you're able to combine a customer's browsing behavior and demographics with their buying history to instantly analyze their current choices and immediately provide relevant recommendations – all before a potential customer clicks to a competitor's website.

Why Use Graph Technology to Power Real-Time Recommendation Engines?

The key technology in enabling real-time recommendations is the graph database, which out-classes other database technology for connecting masses of buyer and product data (or connected data in general).

Making effective real-time recommendations depends on a data platform that understands the relationships between entities, as well as the quality and strength of those connections. Only graph technology efficiently tracks these relationships according to user purchase, interactions and reviews to provide the most meaningful insight into customer needs and product trends.

Graph-powered recommendation engines often take two major approaches: identifying resources of interest to individuals, or identifying individuals likely to be interested in a given resource. With either approach, graph database technology makes the necessary correlations and connections to serve up the most relevant results for the individual or resource in question.

LARGE AMOUNTS OF DATA AND RELATIONSHIPS

Popular recommendation algorithms such as collaborative- and content-based filtering rely on the rapid traversal of continually growing and highly interconnected datasets.

REAL-TIME, RELEVANT RECOMMENDATIONS

Providing real-time contextual insights is a challenge. Therefore, the power of a suggestion system lies in its ability to make a recommendation in real time using immediate history.

NEW DATA AND RELATIONSHIPS

The accuracy and scope of recommendations increases as you add more data points. This rapid growth means the suggestion system needs to accommodate both current and future requirements.

Example: eBay

Ecommerce leader eBay is continually looking to improve ways shoppers search for items. However, existing product searches and recommendations are currently unable to provide or infer contextual information within a shopping request, explains eBay SVP & Chief Product Officer RJ Pittman.

eBay's goal was to build a real-time recommendation engine that understands and learns from the contextual language supplied by the shopper and quickly zeroes in on specific product recommendations. Specifically, to build the eBay App for Google Assistant, the knowledge graph they needed would be coupled with natural language understanding (NLU) and artificial intelligence to store, remember and learn from past interactions with shoppers.

eBay chose Neo4j as the native graph database that holds probabilistic models, which aid understanding in the conversational shopping scenario. The Neo4j graph contains both the product catalog and the attributes of shopper interactions while seeking products.

Moreover, deploying a chatbot to their user base required internet scale with a high degree of resiliency and availability, predictable responses in milliseconds and support from graph experts with experience in these types of deployments. Neo4j includes highly available clustering and exceptional write and read performance – even with millions of nodes, and is highly responsive to such user requests.

The resulting application includes the Neo4j graph database and NLUs that not only understand speech, but also include spelling and grammar intention while parsing these conversations for meaning and context.

The application is currently running in Docker containers in the cloud. Moving forward, the eBay team expects to deploy the chatbot across multiple platforms via plugins like Slack and Microsoft.

Conclusion

Real-time recommendation engines provide a key differentiating capability for enterprises in retail, logistics, recruitment, media, sentiment analysis, search and knowledge management. Storing and querying recommendation data using graph data technology allows your application to provide real-time results, rather than precalculated, stale data.

As consumer expectations increase – and their patience decreases – providing relevant, real-time suggestions will become a greater competitive advantage than ever before.

Use Case #3: Knowledge Graphs

With traditional keyword-based search, delivery results are random, diluted and low-quality. You can't really ask more precise, useful questions and get back the most relevant and meaningful information.

Much in the same way, relational databases are inflexible to future change: If you want to add new kinds of content or make structural changes, you are forced to rework the relational model.

Why Use Graph Technology for a Knowledge Graph?

The standard approach to search is a cumbersome process of repeatedly redefining your search terms until you finally hit on something of interest. This is limiting, especially as users grow accustomed to contextual searches that understand user intent.

There's also the future of search:

- Voice search means we ask and our devices react.
- Ubiquitous functionality of search will appear in emerging applications and more devices.
- Paid search is going to necessitate deeper product knowledge and purchase power.

To augment your enterprise search capabilities, you need a knowledge graph with graph-based search capabilities to deliver only relevant, contextual results.

Example: NASA

David Meza, Chief Knowledge Architect at NASA, needed a more efficient way to examine data relationships between different lessons learned over the last 50 years of space exploration.

He said, "We had been using a standard key list search," which was limited given their information was stored in a lot of separate silos.

GROWING VOLUME OF DATA ASSETS

Datasets grow bigger every second. Without effective data stores capable of handling ever-expanding data and relationships, search will always be limited.

PRECISE SEARCH QUERIES

Any search solution incapable of deep context – by pinpointing user history, intent and context – also isn't able to maintain a competitive advantage.

DISAMBIGUATING SEARCH QUERIES

As data volume increases, overlapping but irrelevant search results increase. Search quality inherently suffers from a larger quantity of results.

David and his team of engineers and scientists weren't able to connect the dots of their data from a relational model, resulting in time-consuming and costly projects. Not only that, but a common goal of these projects is to find patterns and solve problems to avoid repeating them in the future.

"We have a 'big data' problem – not only in structured data, but unstructured data – and we are continually gathering more data," David said. "I need a product or an application that can go across and develop millions if not billions of nodes, connect that information, and at fast speeds."

David and his team discovered Neo4j to be the scalable tool they needed, one that is capable of looking at millions and millions of nodes.

"Previously, this information was stored in separate silos," he said. "Now we're able to connect the dots and uncover trends across the years as to what systems and subsystems may've been impacted by experiments we're running in the space station."

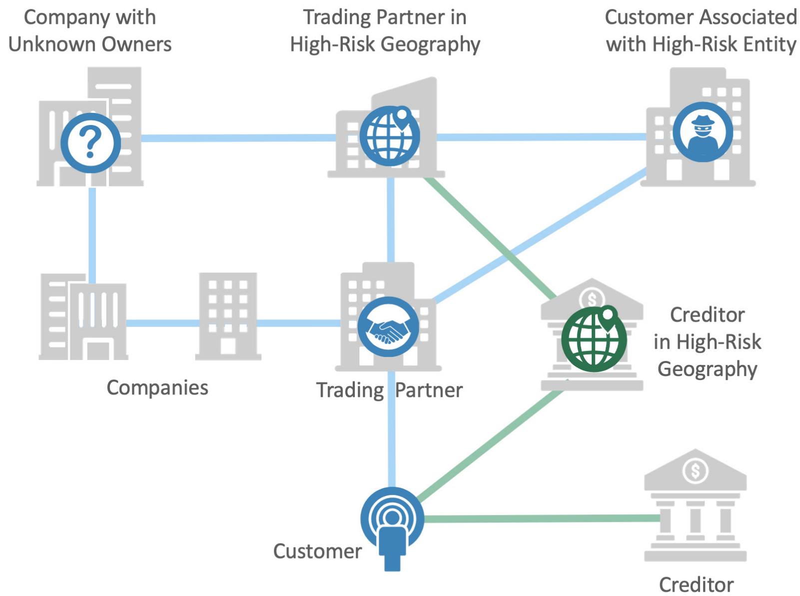
For NASA's "lessons learned" project, using Neo4j found certain information regarding Orion and Apollo eras that prevented an issue, saving well over two years of work and one million dollars of taxpayer funds.

Conclusion

When users are given the most relevant (and possibly unexpected) search results, engagement and satisfaction is higher. But, without the context of relationships and metadata, searches fail to provide precise results a user is looking for.

With large and growing volumes of data assets, you need a knowledge graph to accommodate the relationships inherent in your dataset. By utilizing a graph database, you'll have improved access to information, where users and customers find the product, service or digital asset they need most.

Use Case #4: Anti-Money Laundering



Anti-Money Laundering (AML) schemes today are more sophisticated, and often involve indirection to mislead and delude dubious activity. Traditional technologies, however, aren't designed to connect the dots across many intermediate steps.

Inspectors typically spend an exhaustive amount of time poring over reams of data, often taking months doing so, all while the daily transactions pile up.

Why Use Graph Technology for Anti-Money Laundering?

Many traditional technologies aren't designed to connect the dots, so detecting money-laundering schemes requires a tremendous amount of laborious effort. Teams of inspectors are burdened with manually going through gobs of data.

Then, there's the sheer variety of money-laundering tactics taking place today:

- Structuring (aka smurfing)
- Bulk cash smuggling
- Cash-intensive businesses
- Trade-based laundering
- Shell companies and trusts
- Round-tripping
- Bank capture
- Casinos and other gambling
- Real estate
- Cash salaries
- Life insurance business

FALSE POSITIVES

Alerts of false positives continue to proliferate, eroding confidence in data accuracy, and cost the financial industry billions in wasted efforts.

THE REGULATORY LANDSCAPE

Regulations are calling for better monitoring and analysis. Data reporting transparency is critical to avoiding noncompliance penalties and prosecution.

LEGACY SOLUTIONS

When it comes to money laundering, time is everything. Current AML processes deflate productivity and present efficiency obstacles.

With large and growing volumes of data assets, you need a technology solution that accommodates the size and variety of connection in your dataset.

Graph databases capable of recognizing complex data relationships with real-time query performance are a powerful weapon against the murky world of money laundering and embezzlement.

Example: Money Transfer Service

A Compliance Manager at a Money Transfer Service knew they needed a software tool that would allow AML investigators to see patterns of transfers, which reach into the hundreds of thousands daily transactions.

This Money Transfer Service, who moves nearly $600 billion a year, needed a way to detect "smurfing" activity, which thrives on splitting large sums of illicit funds to a hidden network of beneficiaries.

Rightfully so, the international money transfer industry is highly regulated, and the company complies with anti-money laundering (AML) requirements in every country where it operates (150 worldwide).

"We are continuously looking to improve our systems and processes," the Compliance Manager said, "and we reviewed mapping technology with the FIU (European Financial Intelligence Unit) and immediately saw the potential this type of tool could have for our business."

"When we saw Neo4j, we understood that it was the best tool because it offered a dynamic way of looking into the data," they said. "Our [traditional] tools were all based on SQL technology, and it's impossible to have the dynamic approach Neo4j can give us in this investigation process."

In just a few clicks, Neo4j processes this company's immense amount of data dynamically and in real time, allowing the money transfer company greater efficiency in both compliance and investigatory process.

In terms of one big case spanning several countries and at least two continents, the Compliance Manager said, "Neo4j allowed us to see the whole picture ... The critical element of the suspicion was the journey – and it was obvious when you looked at it with Neo4j."

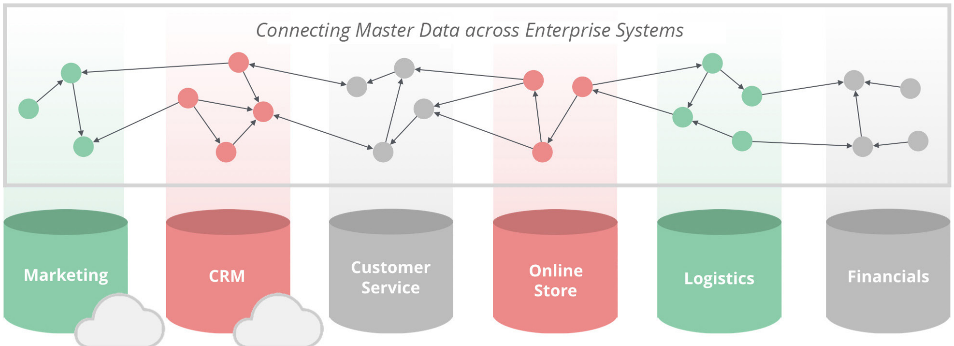
Since fully deploying Neo4j, the company is able to pursue criminal cases 20 times faster than with traditional tools, and they have "defined the benchmark" for investigatory compliance in their industry.

Conclusion

With a graph database, you continually improve AML detection by accommodating new data sources and types – without a rewrite of your data model. Built-in high availability features ensure user data is always accessible to your mission-critical AML engine.

Unlike relational databases, a graph stores interconnected data, making it easier to detect fraudulent activity, regardless of the depth or shape of the data.

Use Case #5: Master Data Management



Master data is the lifeblood of your enterprise, including data such as:

- Users
- Customers
- Products
- Accounts
- Partners
- Sites
- Business units

Many business applications use master data. It's often held in many different places, with lots of overlap and redundancy, in different formats, and with varying degrees of quality. Master data management (MDM) is the practice of identifying, cleaning, storing, and – most importantly – governing this data.

MDM best practices vary, from merging all master data into a single location to managing data assets for easy access from a single service or application. In both cases (or any hybrid solution), enterprise data architects need a data model that provides for ad hoc, variable and exceptional structures as business requirements change. This rapidly evolving model fits best with a graph database.

Why Use Graph Technology for Master Data Management?

Because master data is highly connected and shared, poorly built MDM systems cost business agility in a way that ripples throughout your enterprise. Most legacy MDM systems rely on a relational database that isn't optimized for traversing relationships or rapid responsiveness.

These data connections and relationships in your master datasets are essential to competitive advantage as business analytics evolve. The good news is that graph databases are ideal for modeling, storing and querying the hierarchies, metadata and connections in your master data.

With graph databases, your master data is much easier to model and costs fewer resources (modelers, architects, DBAs and developers) than building a relational solution. In addition, you don't have to migrate all of your master data into a single location. Graph relationships easily connect your siloed data between CRM systems, inventory systems, accounting and point-of-sale systems to provide a consistent vision of your enterprise data.

COMPLEX AND HIERARCHICAL DATASETS

Master data, such as organizational and product data, has deep hierarchies with top-down, lateral and diagonal connections. Managing such data models with a relational database results in complex and unwieldy code that is slow to run, expensive to build and time-consuming to maintain.

REAL-TIME STORAGE AND QUERY PERFORMANCE

The master data store must integrate with and provide data to a host of applications within the enterprise – sometimes in real time. However, traversing a complex and highly interconnected dataset to provide real-time information is a significant challenge.

DYNAMIC STRUCTURE

Master data is dynamic in nature, making it harder for developers to design systems that accommodate its evolution.

Example: Airbnb

In a large, complex organization such as Airbnb, an ever-growing landscape of internal and external data resources – especially when scattered across various platforms – eventually becomes unmanageable and restrictive.

Airbnb Software Engineer John Bodley recognized that Airbnb's data was prohibitively siloed, inaccessible or lacked proper context. He also noticed that employees were relying on tribal knowledge for answers to questions, which stifled productivity.

"We often run an employee survey," he said, "and we consistently scored really poorly around the question: 'The information I need to do my job is easy to find.'" Bodley's team knew they needed to democratize data so any employee, regardless of data-literacy level, was empowered to find resources, fully confident the results were relevant and reliable.

His team set off developing the Dataportal, a self-service, integrated data-space that presents a contextual, holistic view of Airbnb data for employees to navigate their data landscape whenever they need access or answers for their daily working needs.

Bodley and his team quickly realized their entire data ecosystem was best represented as a graph. That led them to the Neo4j graph database, which offered the fastest way to search through millions of data connections per second. Neo4j also integrated well with Airbnb's preferred programming languages while also allowing them to enrich search rankings by taking advantage of the graph topology.

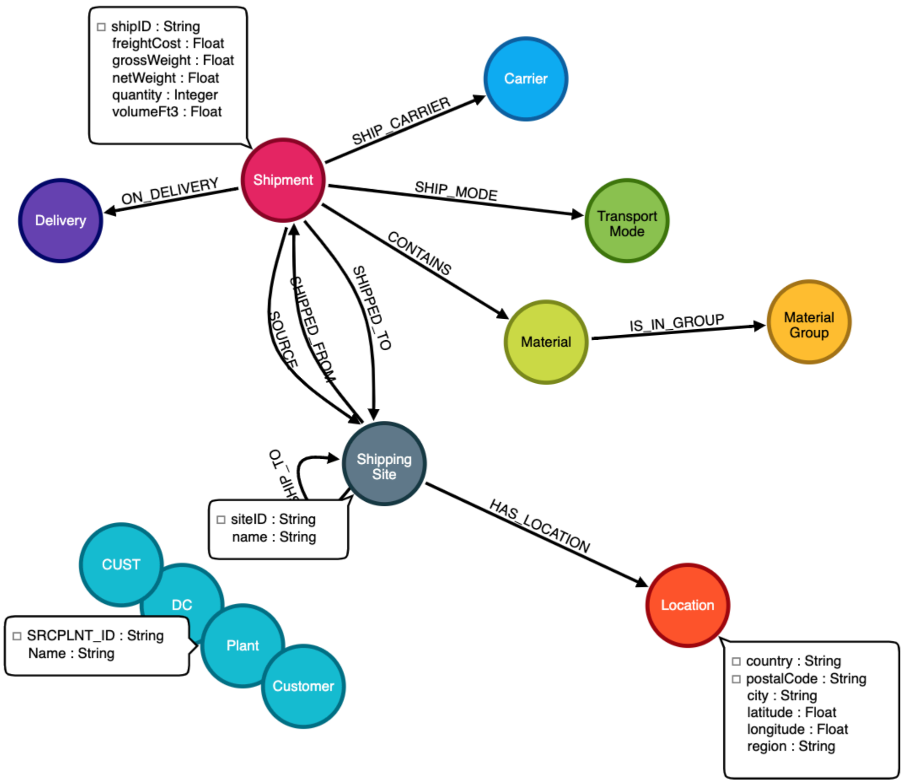
With Neo4j, Airbnb was able to tie together their entire data ecosystem and make it searchable, relevant and trustworthy. Instead of far-flung staffers relying on tribal knowledge, the Dataportal is Airbnb's one-stop resource for finding relevant data, especially for employee- and team-centric information critical to daily performance.

Conclusion

The best data-driven business decisions aren't based on stale information silos. Instead, you need real-time master data with information about data relationships.

Graph databases are built from the ground up to support these relationships. With more efficient modeling and querying, organizing your master data in a graph yields relevant answers faster and more flexibly than ever before.

Use Case #6: Supply Chain Management



By nature, supply chain management is dynamic, with many moving parts, and where bottlenecks may occur at any given point. The challenge is that the volume and detail of data generated by traditional databases lacks real-time, accurate information processing capabilities.

Big data is driving many (if not all) industries today, but supply chain management really is the perfect use case for data-driven insights, predictive analytics and productivity assurance. As data only stands to increase, you'll increasingly face challenges to keep track and maintain control.

Why Use Graph Technology for Supply Chain Management?

Given that master data, by definition is highly shared, this struggle tends to cost business agility in a way that ripples throughout the organization. Our architectures have focused on getting data to fit a single definition of the truth, something most of us come to realize is not a feasible solution in the long run.

Supply Chain Management Data Segments

- Distributing network resources to meet demand
- Route planning for time and transport costs
- Customer management to keep them coming back

DELIVERY ROUTES, THE LAST MILE

The need for greater insight into all of the challenges, efforts and costs that bog down the final leg of the transportation process.

TRANSPARENCY AND RELIABILITY

Only estimating the time and costs associated with each new customer is serving neither party.

SIMPLIFYING DISTRIBUTION NETWORKS

With a series of warehouses, factories and distribution centers, their interrelationship presents many challenges.

PINPOINTING FUTURE DEMAND

Market demand is volatile, and without a big data solution, there's no real insight into customers, suppliers and sensors, plus external factors.

- Marketing to acquire new customers and drive revenue
- Risk analysis of potential accidents and extraordinary events
- Maintenance work, renovations and asset purchases
- Distribution centers and warehouse management for proper allocation of capacity and available space

Example: Transparency-One

Today's supply chains are vast and wide-ranging, which makes them fertile ground for risk. This makes transparency both more vital and more complex.

Recognizing this problem, Transparency-One developed a platform which allows manufacturers and brand owners to learn about, monitor, analyze and search their supply chain, and to share significant data about production sites and products.

After 2013, it was deemed necessary to expand this mapping to include more detailed information about all elements in the supply chain (products, supply chain mapping, etc).

Chris Morrison, Chief Executive Officer at Transparency-One, said: "The challenge was even greater as the area was very new to our clients, and no market solution offered real complete transparency."

To develop the solution, Transparency-One initially turned to a classic SQL database-type solution. But, it quickly realized that the volume and structure of information to be processed would have a major impact on performance, causing considerable problems.

So, the Transparency-One team began to look at graph databases.

"As we had decided on a graph database, we looked at which databases of this type were used by leading players," Morrison said. "The answer was clear and unequivocal: It was Neo4j, the world leader in this field with an established reputation."

Transparency-One chose Neo4j because only a graph database could meet the requirements of the platform; the decisive factors were Neo4j's capacity to manage large volumes of data, and the fact that it is the most widely used database of this type in the world, by both large companies and startups.

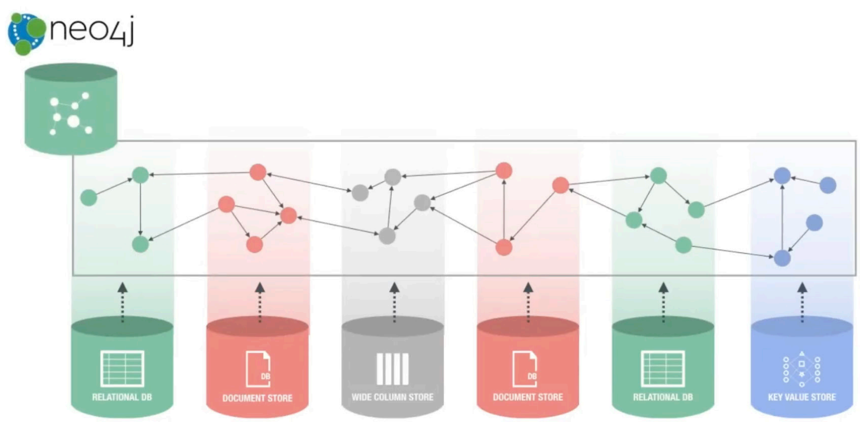
The project was soon up and running and a prototype was built in less than three months. Since then, Transparency-One has extended the solution with the addition of new modules, and it is currently being deployed by several companies.

Conclusion

Traditional relational systems have their benefits, but traversing a network of master data is expensive, less agile and slow.

With a graph database, you achieve global storage for a 360-degree view to effectively manage and leverage your supply chain management data connections. The graph model also makes it easy to evolve as data sources change over time, giving you flexible performance and scalability enabling real-time decision making no matter how large the dataset.

Use Case #7: Empowering Network & IT Operations Management



Data Stored in Disparate Silos

By their nature, networks are graphs. Graph databases are, therefore, an excellent fit for modeling, storing and querying network and IT operational data, no matter which side of the firewall your business is on.

Today, graph databases successfully employed in the areas of telecommunications, network management, impact analysis, cloud platform management and data center and IT asset management.

In all of these domains, graph databases store configuration information. This is used in real time to alert operators to potential design flaws and shared failure modes in the infrastructure, and to reduce problem analysis and resolution times from hours to seconds.

Why Use Graph Technology for Network and IT Operations?

As with master data, a graph database is used to bring together information from disparate inventory systems, providing a single view of the network and its consumers – from the smallest network element all the way to the applications, services and customers who use them.

A graph representation of a network enables IT managers to catalog assets, visualize their deployment and identify the dependencies between the two. The graph's connected structure enables network managers to conduct sophisticated impact analyses, answering questions like:

- Which parts of the network – applications, services, virtual machines, physical machines, data centers, routers, switches and fiber – do particular customers depend on? (Top-down analysis)
- Conversely, which applications, services and customers in the network will be affected if a particular network element fails? (Bottom-up analysis)
- Is there redundancy throughout the network for the most important customers?

TROUBLESHOOTING

Whether you're managing a major network change, bolstering your security-related access or optimizing an application infrastructure, the physical and human interdependencies are extremely complex. This makes it difficult to troubleshoot.

IMPACT ANALYSIS

Relationships among the various nodes in your network are neither purely linear nor hierarchical, making it difficult to determine the interdependencies of network elements on each other. These relationships become even more complex when two or more systems are brought together.

GROWING PHYSICAL AND VIRTUAL NODES

With rapid growth in network sizes and a rapid increase in elements to support new network services and devices, your IT organization must develop systems that accommodate both current and future requirements.

A graph database representation of the network can also enrich operational intelligence based on event correlations. Whenever an event correlation engine (such as a Complex Event Processor) infers a complex event from a stream of low-level network events, it assesses the impact of that event against the graph model and triggers any necessary compensating or mitigating actions.

Example: Comcast

Companies around the world have come a long way towards making homes smarter.

Smart devices send alerts when the front door is unlocked, the house alarm is disengaged, or someone is at the front door. These connected devices naturally interact with one another but often lack capabilities like natural language processing. The result is a large collection of connected devices that can't be automated.

"A person is not just an ID. A person is a set of relationships to personal information, locations, people and devices," said Jessica Lowing, Director of Project Management at Comcast.

Homes aren't able to perform tasks like "turn off the lights in Lily's room" because these requests require insight into complex semantic and social relationships. Personalization is also incredibly limited. To address these challenges, Comcast put together a team dedicated to creating and perfecting an xFi smart home prototype.

The first step was to develop rich definitions for all the terms in the Comcast profile graph. For example, a "person" definition needed much more than a unique ID. It had to include a unique set of relationships to personal information, locations, people, and devices. "Since people are at the center of these smart homes, they also need to be at the center of our automation," Lowing said, "which brings us back to modeling social and semantic relationships."

The team recognized that the real value of this rich data was the relationships between them, which would require a native graph database structure. Ultimately, the team would also need to build a shared platform at the household level.

This resulted in the Xfinity profile graph, a scalable, flexible, multi-tenant user-profile service for extending personal information and relationships across Xfinity products. It models customers' real-life relationships, and provides context so that Xfinity applications provide a more personalized experience for users.

Conclusion

Discovering, capturing and making sense of complex interdependencies is central to effectively running network and IT operations, which is critical to running an enterprise. Whether it's optimizing a network or application infrastructure or providing more efficient security-related access, these problems involve a complex set of physical and human interdependencies that are challenging to manage.

The relationships between network and infrastructure elements are rarely linear or purely hierarchical. But, graph databases are designed to store that interconnected data, making it easy to translate network and IT data into actionable insights.

Use Case #8: Data Lineage

Risk modeling is a multiplex of requirements. It also demands organizations – especially large banks, hedge funds and aggressive investment houses – to track data connections across a complex network of investments, holdings, financial instruments and granular pricing data.

Why Use Graph Technology for Data Lineage?

Banks and other regulated financial organizations like yours are required to trace data dependencies through many intricate levels before reaching original, authoritative data sources – a crucial underlying requirement most systems simply can't address. Data lineage challenges stemming from stringent requirements like BCBS 239 require a lot more flexibility and persistence than traditional data storage systems offer.

You're required to backwards trace data through discrete data silos until its lineage ends with an authoritative source. Likewise, even though entities require standard identifiers, business groups often use their own terminology and algorithms – sometimes within the same organization.

The structure and location of data often makes it all but impossible to address in a single, centralized data store. And ironically, moving everything into a single repository may make tracing data lineage even more difficult.

Integrating information into a single, enterprise-wide logical data model requires graph technology. With a graph database, you get a single source of truth and complex or hidden data connections are queried and revealed within milliseconds.

Example: UBS

Sid Hannif, a Solution Architect at UBS, struggled with the hundreds of systems and siloed data he needed to join to make customer-use experiences seamless with up-to-date, real-time data.

"We're supposed to provide timely, accurate and complete data, and here the consumer had to go and join the data themselves," he said.

UBS has been around since 1854. There's approximately 60,000 employees, and the financial enterprise is split into different divisions, covering wealth management, investment bank, asset management, a corporate center and data services.

With the data services group, Sid and his team look after 12 data domains and sub datasets, where the data is sent out over a number of channels. Account data lived in relational databases. The number of integrations were a big obstacle and the UBS customer reference data distribution platform was suffering as a result.

The data services team wanted to use metadata to have the engines interrogate how the data shaped, how it links with other data, and then do the joining. And, that's exactly what a knowledge graph does as a subcomponent of the system.

From there, UBS chose Neo4j's graph database to model their data.

"The governance of data – moving from place to place – is a lot easier," said Sid. "We understand who's taking our data and through what channels. But more importantly, our consumers are able to understand what datasets we offer and how they relate to other ones."

GOVERNANCE AND INFRASTRUCTURE

Banks must utilize data governance and integrated data taxonomies, as well as group-wide metadata including identifiers for entities, counterparties, customers and accounts.

RISK DATA AGGREGATION

Banks must generate accurate, consistent and reliable risk data while maintaining full visibility back to authoritative data sources. These datasets must satisfy a full spectrum of requests from managers and regulators.

RISK REPORTING

Management and regulatory reports must represent risk in a precise and auditable manner, and reconcile to the complexity of the bank's risk model and operations.

SUPERVISORY REVIEW

Bank risk supervisors are required to review their institution's ongoing compliance with BCBS 239 principles.

Conclusion

With a graph database for privacy and risk reporting compliance, there are a number of beneficial outcomes traditional systems don't offer:

- Trace the lineage of risk factors back to their original, authoritative data sources
- Span pricing, position, cash management and other data silos into a unified dataset
- Work with regulators to visualize and modify risk model graph diagrams
- Enable the easy modification of risk models to keep pace with changing market conditions, organizational changes and investment strategies
- Handle mergers, divestitures and reorganizations that affect the historical and future operation and performance of trading desks.

The Neo4j graph database also supports global financial terminology standards backed by professional services that guarantee success and provide new visibility into your compliance efforts and day-to-day operations.

Use Case #9: Identity & Access Management

Identity and access management (IAM) solutions store information about parties (e.g., administrators, business units, end-users) and resources (e.g., files, shares, network devices, products, agreements), along with the rules governing access to those resources. IAM solutions apply these rules to determine who can access or manipulate a resource.

Traditionally, identity and access management has been implemented either by using directory services or by building a custom solution inside an application's backend. Hierarchical directory structures, however, can't cope with the complex dependency structures found in multi-party distributed supply chains. Custom solutions that use non-graph databases to store identity and access data become slow and unresponsive as their datasets grow in size.

Why Use Graph Technology for Identity and Access Management?

A graph database can store complex, densely connected access control structures spanning billions of parties and resources. Its richly and variably structured data model supports both hierarchical and non-hierarchical structures, while its extensible property model allows for capturing rich metadata regarding every element in the system.

With a query engine that can traverse millions of relationships per second, graph database access lookups over large, complex structures execute in milliseconds – not minutes or hours.

As with network and IT operations, a graph database access control solution allows for both top-down and bottom-up queries:

- Which resources – company structures, products, services, agreements and end users – can a particular administrator manage? (Top-down)
- Given a particular resource, who can modify its access settings? (Bottom-up)
- Which resource can an end-user access?

Access control and authorization solutions powered by graph databases are particularly applicable in areas like content management, federated authorization services, social networking preferences and software as a service (SaaS) offerings, where they realize minutes-to-milliseconds increases in performance over their relational database predecessors.

Example: Telenor Norway

Telenor Norway is an international communications services company. For several years, it has offered its largest business customers the ability to self-service their accounts. Using a browser-based application, administrators within each of these customer organizations can add and remove services on behalf of their employees.

To ensure users and administrators see and change only those parts of the organization and the services they are entitled to manage, the application employs a complex identity and access management system which assigns privileges to millions of users across tens of millions of product and service instances.

INTERCONNECTED PERMISSIONS DATA

To verify an accurate identity and its access permissions, the system needs to traverse through a highly interconnected dataset that is growing in size and complexity.

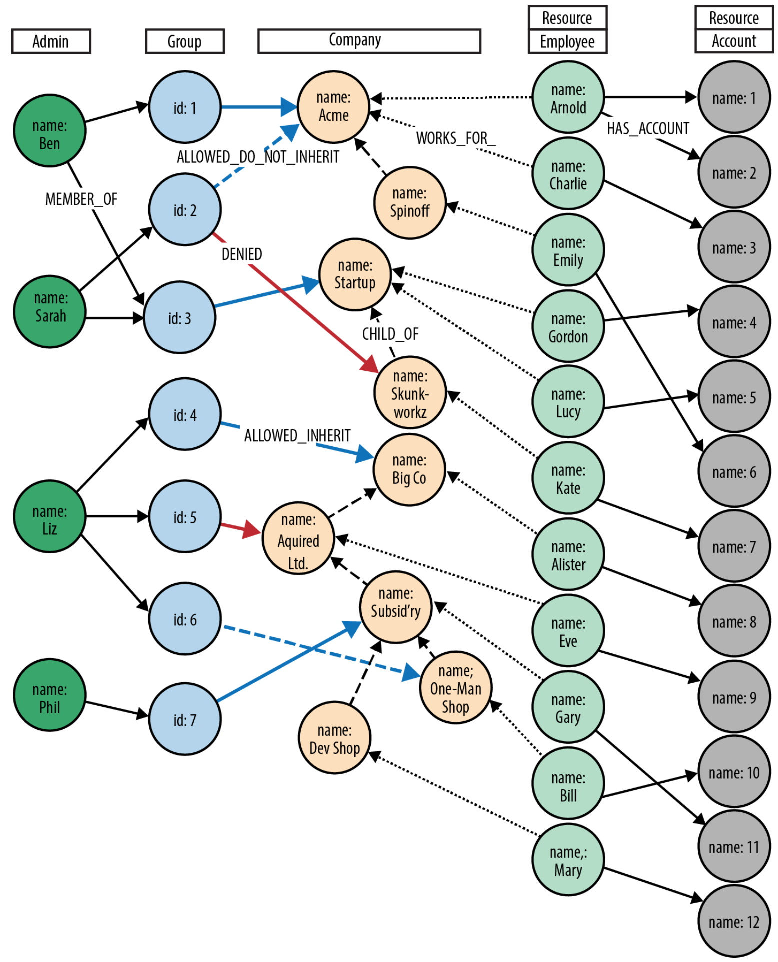
PRODUCTIVITY AND CUSTOMER SATISFACTION

As users, products and permissions grow, traditional systems no longer deliver responsive query performance, resulting in diminished user experience and frustration for users.

DYNAMIC STRUCTURE AND ENVIRONMENT

With rapid growth in the size of users and their associated metadata, your application needs to accommodate both current and future identity management requirements.

Below is an example of Telenor's data model.



Caption: A sample of Telenor Norway's data model showing their identity and access management application.

Due to performance and responsiveness issues, Telenor decided to replace its existing IAM system with a graph database solution. Their original system used a relational database, which employed recursive JOINs to model complex organizational structures and product hierarchies. Because of the join-intensive model, their most important queries were unacceptably slow.

In contrast, once they implemented a graph database solution, Telenor realized the performance, scalability and adaptiveness necessary for handling their identity and access management needs, reducing queries that once took many minutes to milliseconds.

Conclusion

For your enterprise organization, managing multiple changing roles, groups, products and authorizations is an increasingly complex task. Relational databases simply aren't up to the task of managing your identity and access needs as queries are far too slow and unresponsive.

Using a graph database, you seamlessly track all of your identity and access relationships in real time. With an interconnected view of your data, you have better insights and controls than ever before.

Use Case #10: Bill of Materials

The problem isn't that the Bill of Materials (BOM) process is dead due to the era of digital transformation – it's still the crux of manufacturing processes – but rather that new manufacturing data is everywhere and is poorly managed.

A lack of real-time collaboration with a distributed group of people (contractors, suppliers) – where products are multidisciplinary (mechanical, electro mechanical) and there's new design-to-manufacturing workflows – spreadsheets and siloed data are barriers to success.

Why Use Graph Technology for Bill of Materials?

Most enterprise manufacturers use vendor applications: CRM systems, work management systems, accounts payable, accounts receivable, point of sales systems, and so on. Due to this approach, you need to store and model data as a graph where a native graph stores interconnected master data that's neither purely linear or hierarchical.

Likewise, emerging BOM trends add even more layers of data challenges. Compliance requirements are more strict. Counterfeit parts are pervading the market. There's consolidation among parts manufacturers, and parts life cycles are becoming shorter. Meanwhile, tech innovation has shown to increase risk of availability while simultaneously contributing to obsolescence.

A graph database flexible data model makes it easy to evolve master data as needs change over time.

Example: U.S. Army

"The U.S. Army procures and tracks millions of equipment components every year," said Preston Hendrickson, Technical Lead for the U.S. Army Project, CALIBRE.

The challenge was to rapidly collect and combine a massive bill of materials (BoM) information – every component and its cost, what equipment it relates to, and its expected lifespan/ average time to failure.

With over 1 million active, guard and reserve soldiers and around 200,000 civilian staff, the Army also deploys a staggering amount of equipment – small arms, rifles and machine guns, tanks, trucks and armored vehicles, and thousands of ships, helicopters and aircraft.

Buying and managing this scale of equipment is a significant logistical challenge. It involves acquiring millions of parts for hundreds of thousands of weapons and vehicles every year, maintaining these components – sometimes in deadly, far-flung territories. It also involves the responsibility of knowing that doing the job properly saves not just costs, but lives.

This level of data management was becoming increasingly difficult, even impossible, on the Army's aging mainframe-based system.

"Thanks to Neo4j's graph database, the U.S. Army can now quickly store, explore and visualize their wealth of logistical data," said Hendrickson. He noted that the contrast with their previous system was stark.

"Typically, it would take 60 person-hours to load data so the Army could understand 'we're going to need to replace X, Y, or Z parts' or provide cost estimates and analyses," said Hendrickson. "Now it's down to seven to eight hours."

Bill of Materials Challenges

PRODUCT MANUFACTURABILITY

Keeping the BOM up to speed with the lifecycle of product development.

MULTI-DISCIPLINARY BOM

Manufacturing now goes beyond mechanical parts, including electronic, software and more.

DESIGN WORKFLOWS

No global traceability and audit, no centralized catalog for inventory management or the design to order process.

Components are ordered sometimes millions at a time. With Neo4j, the Army anticipates the demand for spare parts and spread it across multiple time periods, instead of one quarter at a time. "The result is better ordering and budgeting," Hendrickson said. "The Army gets a fuller total cost estimate, but it's also a more predictable total cost of ownership (TCO) as well."

"Neo4j enables analysts to save huge amounts of time," Hendrickson continued. "Now everything is in the graph, we are able to see more detailed data that previously had been glossed over. We now have original data, and much better detail in analysis. Answers are immediate. As a result, the parts delivery is more accurate and order turnaround is much faster."

Conclusion

Using graph technology, BOM teams receive consistent, meaningful views of master data to then identify relationships between people, accounts, business entities, transactions and other data.

With a 360-degree view of your BOM, you may leverage meaningful, contextual data connections. With a versatile graph model, it's simple to evolve the system as data sources change over time. Performance and scalability enable real-time decision making – no matter how large the dataset.

Conclusion

These 10 use cases of graph databases are hardly a comprehensive list, but they do highlight impactful and profitable applications of graph technologies. Even so, there are plenty of other use cases for graph technologies, including the life sciences, social networking, gaming, government, sports and even non-profits.

Today's CIOs and CTOs are under increasing pressure to provide actionable insights from their big data, even as datasets grow larger and more unwieldy. What they need is technology that determines the connections between data points and derives appropriate cogent conclusions.

Graph databases are that technology solution. They allow data professionals at every level to exploit the potential of their data relationships, rather than just individual data points. The only limit to how those relationships might be harnessed is up to the imagination of the database user.

In this way, graph databases are a rising tide – not merely a passing fad – in the world of big data insights.

Questions about Neo4j?

Contact us around the globe:
info@neo4j.com
neo4j.com/contact-us