How Graph Databases Solve Problems in Network & Data Center Management: a Close Look at Two Deployments

An ENTERPRISE MANAGEMENT ASSOCIATES® (EMA™) White Paper

October, 2013

EMA™
IT & DATA MANAGEMENT RESEARCH,
INDUSTRY ANALYSIS & CONSULTING

How Graph Databases Solve Problems in Network & Data Center Management: a Close Look at Two Deployments

Table of Contents

Executive Introduction 1

The Neo4j Solution 1

Interview with software product developer based in North America 2

- Q: Can you describe your role and your product/ project focus? 2
- Q: Why did you choose Neo4j? 2
- Q: Do you have anything else to add about Neo4j? 2

Interview with a European Software Consultant working with a large European Telecommunications provider 3

- Q: Can you tell me a little bit more about you and your organization? 3
- Q: Can you share more specifically how you view those advantages? 3
- Q: Can you tell me a little more about the requirements of the deployment you did for a large telecommunications provider? 3
- Q: How did you get involved? 3
- Q: How many people were there on the Proof of Concept Team? And how did the deployment evolve? 4
- Q: What were some of the other benefits that the Neo4j deployment achieved there? 4

EMA Perspective 4

About Neo4j 5

How Graph Databases Solve Problems in Network & Data Center Management: a Close Look at Two Deployments

Executive Introduction

Discovering, capturing and making sense of complex interdependencies is central to running IT organizations more effectively, and it is also a critical part of running the businesses IT serves. Whether it's optimizing a network, or an application infrastructure, managing change, or providing more effective security-related access—more often than not these problems involve a complex set of physical and human interdependencies that can be quite challenging to manage. Moreover, once any two or more of these areas are brought together, the relationships are rarely linear or purely hierarchic. They form, in the computer science sense of the term, a graph. These domains are rarely static. In fact, they tend to change with reasonable frequency, as a result of factors such as reorganization and personnel changes, mergers and acquisitions, new applications being developed and old ones retired, and ongoing data center improvements.

While traditional relational databases have served the industry well in the past in enabling service and process models that tread upon these complexities, in most deployments they still demand significant overhead and expert levels of administration to adapt to change. Relational databases require cumbersome indexing when faced with the non-hierarchic relationships that are becoming yet more persistent in complex IT ecosystems, with partners and/or suppliers and service providers, as well as more dynamic infrastructures associated with cloud and agile.

This white paper examines the Neo4j graph database, which is designed to support requirements in an increasingly modern, "non-linear" and "connected" world.

This white paper examines the Neo4j graph database, which is designed to support requirements in an increasingly modern, "non-linear" and "connected" world. Two deployments are used to explore the matter through concrete examples, with salient portions of the interviews presented here. While graphs may be more commonly associated with social networking or "graph search" style recommendations, the focus here will be on the IT applications, focusing on (though not exclusively limited to) network & data center management.

The Neo4j Solution

Neo4j has become the market leader in graph databases. With its roots in Malmö, Sweden and a headquarters in Silicon Valley, Neo now has a global presence that now spans ten countries, Neo4j. Unlike relational databases, graph databases are designed to store interconnected data that's not purely hierarchic, make it easier to make sense of that data by not forcing intermediate indexing at every turn, and also making it easier to evolve models of real-world infrastructures, business services, social relationships, or business behaviors that are both fluid and multi-dimensional.

Neo4j is built "from the ground up" to support high-performance graph queries on large data sets for large enterprises with high-availability requirements. It includes its own graph query language, and uses native graph processing and a storage system natively optimized for graphs. Thirty of its current 200 customers reside in the Global 2000. The second deployment represented here is, in fact, a significant testament to Neo4j's scalability, versatility and breadth.

Its customers, typically enterprises and consultants looking for better ways to address the problems of modeling complexity, range across social networking, financial risk management, network operations root cause analysis and change impact management, to name a few. Beyond its use inside the

enterprise, which remains the predominate use, Neo4j is also used in a number of product development organizations, forming an innovative foundation for other software products. The first interview below provides an example of such a use.

Interview with software product developer based in North America

Q: Can you describe your role and your product/ project focus?

I helped found the Linux High Availability and the Pacemaker projects, and have spent time at Bell Labs and IBM, among others, before my current project, which uses Neo4j. The goal of my work now is to develop a way of managing critical systems infrastructures for high availability by capturing how devices are interrelated across the infrastructure. Our agent-based discovery doesn't require generating traffic over the network, or sending out packets. I like to say that it's amazing what you can learn just by listening—in this case listening to CDP or LLPD and ARP packets and OS resources on listening ports—as each machine effectively monitors those it's connected to.

We also have a security play. We can tell you what servers you lost track of in a way that won't annoy your security people. And we can also tell you here's what you're monitoring and here's what you're not monitoring. They're related of course. Monitoring and "keeping bad guys out." Thirty percent of all machines that are penetrated are those that are "not known" or forgotten about.

Q: Why did you choose Neo4j?

I didn't see any other viable solution. Quite on my own I understood that my problem was a graph problem. They didn't have to tell me.

A graph database is an excellent format for capturing and modeling the interdependencies that can help to diagnose failures. If I were to explain any IT project and design in more depth, I'd draw a graph—that's how people understand IT infrastructure. We whiteboard it. Without this type of modeling, once things fell out of sync, I'd have to crawl all over the machines to figure out what they were doing, and then reassemble the pieces. For example, you might have a set of interdependencies in either direction and Neo4j will let you capture that easily and naturally without having to define a whole mess of linear relationships between each device.

I didn't see any other viable solution. Quite on my own I understood that my problem was a graph problem. They didn't have to tell me.

Scalability was also key. You don't have to do a relational join between every machine with every other machine fifteen machines die and you don't know which one caused the problem—now imagine that's in a collection of 100 machines, or a thousand machines. Neo4j is easily extensible to supporting complex environments.

Q: Do you have anything else to add about Neo4j?

Yes, actually. They have a really excellent community—and work hard to support their customers as well. Since I also depend on building a solid community around my work, and have a long record of being involved with open source initiatives, I can really appreciate what they're doing there. Frankly, I'm impressed with how they're managing it all. I've watched them work. And that's in part what made me decide that we can develop our own solution by leveraging Neo4j.

Interview with a European Software Consultant working with a large European Telecommunications provider

Q: Can you tell me a little bit more about you and your organization?

My firm is a software consultancy and I work closely with many Neo4j deployments an eye to modeling, problem solving and innovation. I see some distinctive advantages in graph databases, and in particular in Neo4J's offering.

Q: Can you share more specifically how you view those advantages?

The graph model is unique with its ability to accommodate highly connected, partially structured datasets that can evolve over time in terms of complexity and structure. Graphs are also naturally capable of providing a wide range of evolvable ad-hoc queries on top of these datasets. This not only makes for much improved flexibility in design. It also enables relationships to be easily captured that are unsuited to traditional hierarchic models. It also allows for much better adaptability to changes when the changes themselves are less predictable or not strictly hierarchic in nature. One of the things I especially appreciate is that Neo4j makes it simple to model real-life or business situations—it provides a much better working foundation for key stakeholders who are not necessarily technical.

The graph model is unique with its ability to accommodate highly connected, partially structured datasets that can evolve over time

Q: Can you tell me a little more about the requirements of the deployment you did for a large telecommunications provider?

This company had a very large complex network with many silos and processes—including network information spread across more than thirty systems. The large number of data sources was in part due to network complexity, and in part due to different business units, as well as organic growth through mergers and acquisitions. These different sources also created a very non-linear fabric that had to be modeled and understood from various dimensions.

Previous to Neo4j, they had different network layers stored in different systems—for instance, one system might be dedicated to cell towers, another fiber cables, and another devoted to information about consumers, or conversely enterprise customers.

The company needed a way to predict and warn customers in advance of any service interruptions in order to maintain customer service agreements and avoid financial penalties due to unplanned downtime. With daily changes required to optimize the network infrastructure, managing this effectively was definitely a challenge. One of their business process challenges was around maintenance and ensuring redundancy—they needed to know if they took a device down for maintenance, exactly who might be impacted and what the penalties might be, as well as what alternate routes might better mitigate the impact. There was also a more proactive, planning requirement—e.g. planning to lay an alternate cable for backup and knowing how things are connected so best-case alternate paths can be identified. What are all the upstream interdependencies? Downstream interdependencies? Etc.

Q: How did you get involved?

This company had some choices between Neo4j and some very rigid and expensive tools designed to fit specific needs. For instance they already had an impact analysis system from which they were extracting

spreadsheets and a team of about ten people doing manual work on the spreadsheets, which is expensive and error-prone. But a small team at that company did a proof of concept with Neo4j and felt that it had many advantages—both in terms of immediate benefits and potential—given the graph nature of many network interdependencies across various processes. Once the POC team showed some initial potential, they got the buy-in to move forward to next steps, and I came on board.

Q: How many people were there on the Proof of Concept Team? And how did the deployment evolve?

There were only three: two developers plus the project manager. It only took a few months to show the benefits. As the deployment evolved, we added someone to support needed integrations. Within four to six months we were able to match the pre-existing system and to demonstrate benefits and advantages. These included fast and powerful queries, along with a custom visualization module. Then we proceeded to take the next steps to support more complex analysis for root cause—e.g. *if you do this or of this occurs, it will cause this specific problem.* Or conversely, *This is the reason that you experienced this problem.* All along the way there was fierce competition to show value, as this telecommunications provider was very serious about managing its costs.

One of the things I like best about Neo4j is that it supports incremental development. You don't have to get all the data at once to get value from it. You can build your graph in an incremental way as opposed to more rigid approaches, and then add other layers to accommodate more data and more complex or new relationships.

One of the things I like best about Neo4j is that it supports incremental development. You don't have to get all the data at once to get value from it.

It was almost a dream business case because you could measure the benefit of the project as the telecommunications provider began to manage production-level changes that impacted its many actual customers. Every time they got something wrong there were immediate costs in penalties. And the values were huge.

Q: What were some of the other benefits that the Neo4j deployment achieved there?

After implementation of the model and the impact analysis queries, it was easy to extend the application to support single point of failure detection thanks to the flexibility of the graph model. Also, by providing an effectively unified cross-domain view, experts from different silos could work together for the first time and agree on a common domain terminology.

EMA Perspective

It's not often easy to clarify the real values of "enablers" versus the applications that run on them, but the two examples above provide insights as to why Neo4j, in particular, and graph databases more generally, can deliver meaningful benefits. In EMA's current research on application discovery and dependency mapping, one vendor's choice for a graph database vs. relational was quite articulate and explicit:

"We feel that relational databases are a much poorer choice because it is harder to represent the data in a natural way, harder to extend it, and harder to query data where the relationships are complex and sometimes cyclic."

EMA is excited about the prospects for Neo4j as the leader in graph databases (with more than 50,000 downloads a month and a significant number of Global 2000 companies as commercial customers) given the growing need to answers exactly these requirements for a wide variety of use cases. What Neo likes to call the "whiteboard" nature of graph is telling—as the modeling doesn't require an intermediary join table to link up what domain experts, on their own, might naturally draw on a white board. The more that technology can support human expertise and intuitions directly, the more likely it is to deliver superior results in terms of accuracy, adaptability, ease of deployment and overall control. EMA looks forward to seeing strong continued growth for Neo4j as a true "enabler" for a still growing number of values across a widening number of verticals, IT stakeholders and business professionals.

EMA is excited about the prospects for Neo4j as the leader in graph databases (with more than 50,000 downloads a month and a significant number of Global 2000 companies as commercial customers).

About Neo4j

Neo Technology researchers have been pioneering graph databases since 2000, and have been instrumental in bringing the power of the graph to numerous organizations worldwide, including 25 Global 2000 customers, such as Cisco, Accenture, Deutsche Telekom, Telenor, and HP. Serving customers in production for over a decade, Neo4j is the world's leading graph database with the largest ecosystem of partners and tens of thousands of successful deployments.

For more information about Neo Technology, please visit www.neotechnology.com. To download Neo4j, please visit www.neo4j.org.

About Enterprise Management Associates, Inc.

Founded in 1996, Enterprise Management Associates (EMA) is a leading industry analyst firm that provides deep insight across the full spectrum of IT and data management technologies. EMA analysts leverage a unique combination of practical experience, insight into industry best practices, and in-depth knowledge of current and planned vendor solutions to help its clients achieve their goals. Learn more about EMA research, analysis, and consulting services for enterprise line of business users, IT professionals and IT vendors at www.enterprisemanagement.com or blogs.enterprisemanagement.com. You can also follow EMA on Twitter or Facebook.

Corporate Headquarters:
1995 North 57th Court, Suite 120
Boulder, CO 80301
Phone: +1 303.543.9500
Fax: +1 303.543.7687
www.enterprisemanagement.com

2800.100313