The #1 Platform for Connected Data

Neo4j 图数据库平台

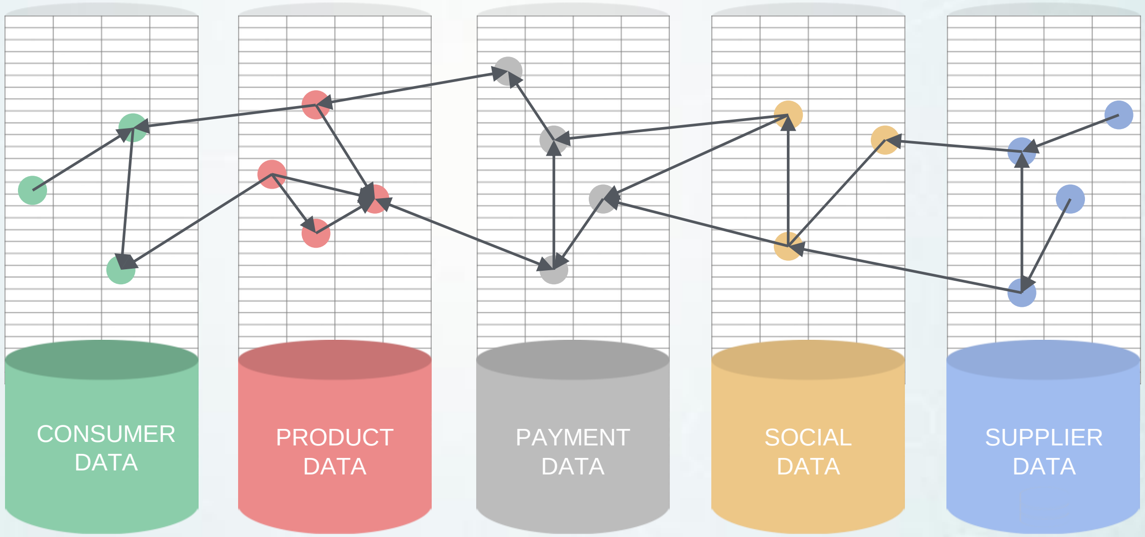
俞方桦 博士

Neo4j Beijing Meetup, 18th Mar. 2018

俞方桦 | Joshua Yu

Field Engineering, Neo4j APAC

Joshua.yu@neo4j.com



“(在大数据时代)新的竞争优势将完全来自于理解和使用各种关联来产生可执行的洞见。”

“The next wave of *competitive advantage* will be all about *using connections* to produce actionable insights.”

Agenda

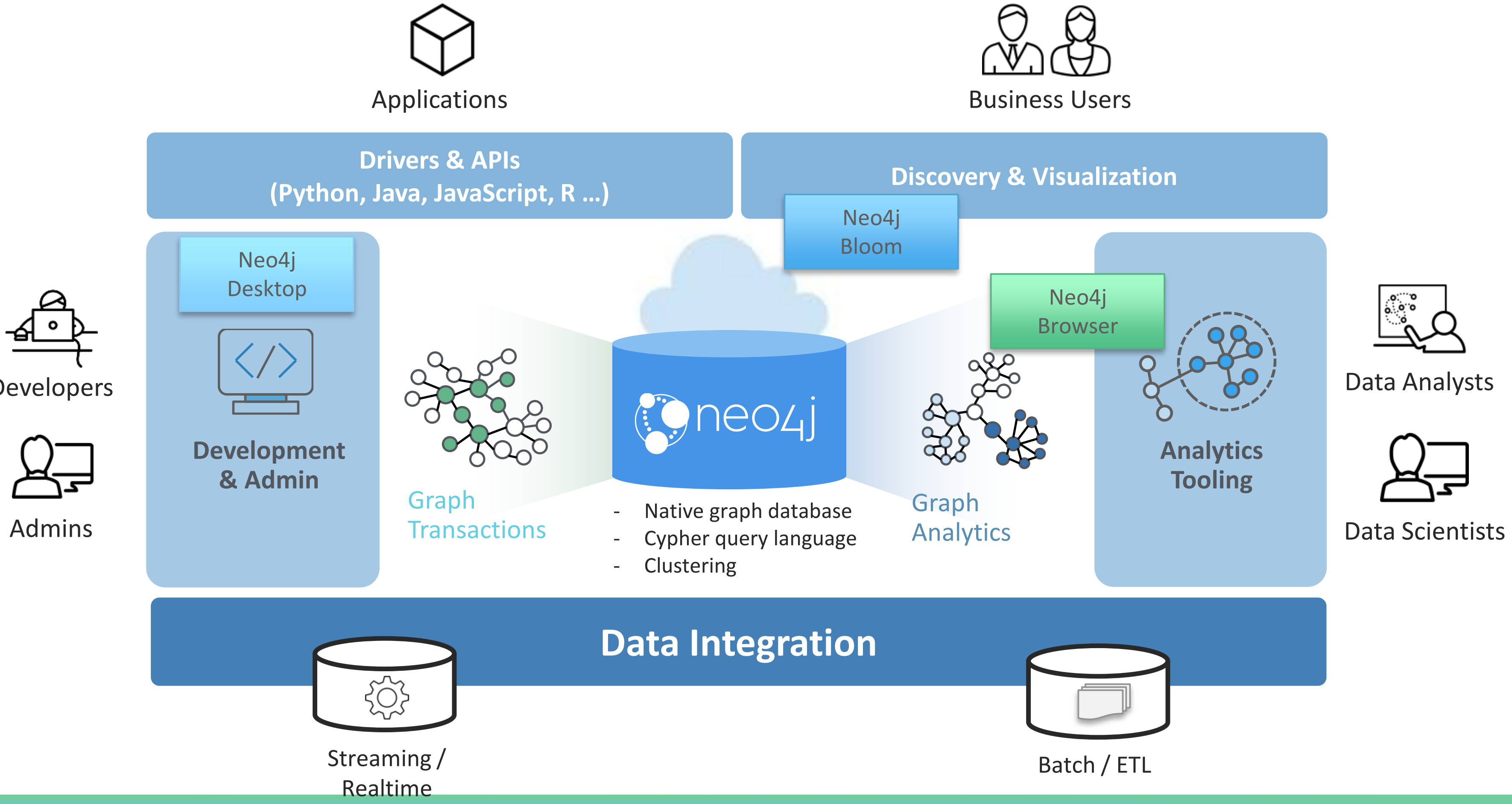
- The Graph Database Platform
- The Graph DBMS
- Cypher Query Language
- Analytical UI – Neo4j Browser
- Neo4j Desktop
- Visualization
- Enterprise Integration Patterns
- Data Migration

Agenda (cont.)

- Clustering
- Deployment
- Development
- Operational Support
- User Defined Procedures
- Security
- Business Use Cases
- Technology Use Cases
- The Community

Neo4j – 图数据库平台

Neo4j 图数据库平台



Applications

Business Users

Drivers & APIs
(Python, Java, JavaScript, R …)

Discovery & Visualization

Neo4j
Bloom

Neo4j
Desktop

Development
& Admin

Developers

Admins

Graph
Transactions

- Native graph database
- Cypher query language
- Clustering

Graph
Analytics

Neo4j
Browser

Analytics
Tooling

Data Analysts

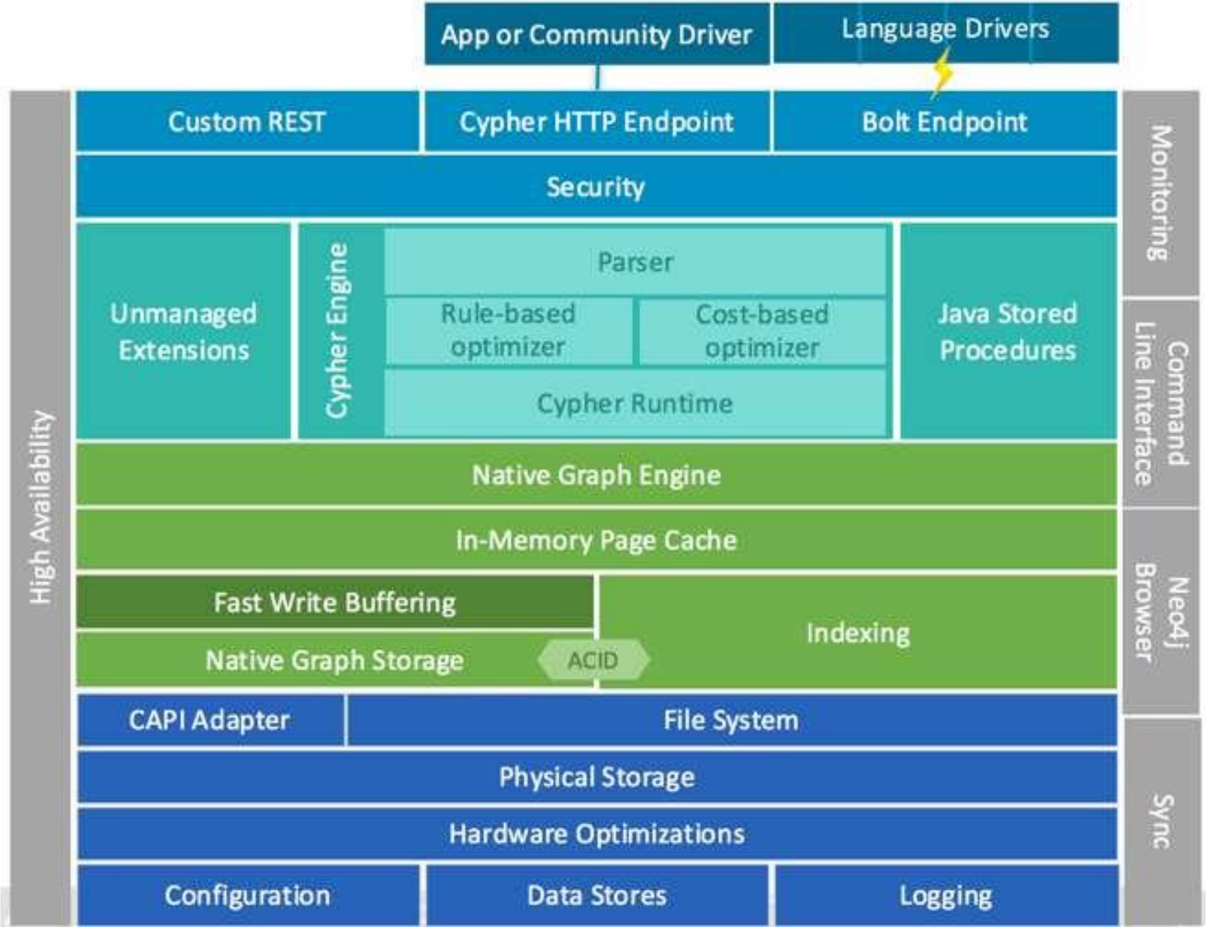
Data Scientists

Data Integration

Streaming /
Realtime

Batch / ETL

Neo4j – The Complete Application Architecture



Neo4j: 企业级的图数据库

本地/原生的图存储

Designed, built, and tested for graphs

本地/原生的图处理

For real-time, relationship-based apps

Evaluate millions of relationships in a blink

直观一致的数据建模

Faster projects compared to RDBMS

ACID 交易和数据安全

Fully ACID transactions, causal consistency and enterprise security

强大的、表达性强的查询语言

Improved productivity, with 10x to 100x less code than SQL

因果集群应用架构

Architecture provides ideal balance of performance, availability, scale for graphs

内置的数据导入

Seamless import from other databases

支持常用平台和语言的驱动

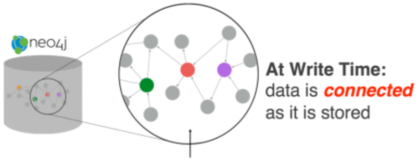
Fits easily into your IT environment, with drivers and APIs for popular languages

MATCH (A)

关键性的架构组件

1

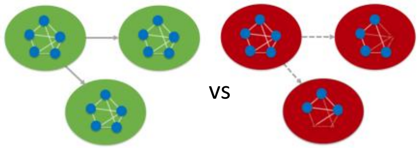
Index-Free Adjacency



In memory and on flash/disk

2

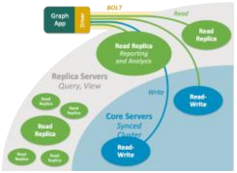
ACID Foundation



Required for safe writes

3

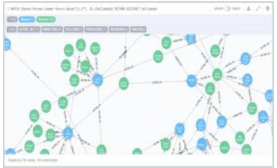
Full-Stack Clustering



Causal consistency

4

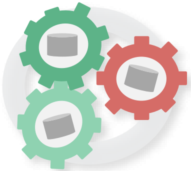
Language, Drivers, Tooling



Developer Experience,
Graph Efficiency, Type Safety

5

Graph Engine



Cost-Based Optimizer, Graph
Statistics, Cypher Runtime

6

Hardware Optimizations

For next-gen infrastructure

Neo4j公司最初是在下面哪一个城市创建的：

A. London

B. San Mateo / 硅谷地区

C. Frankfurt

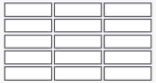
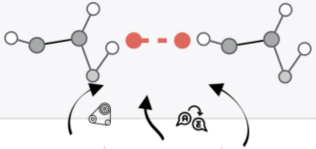
D. Malmo

E. Sydney

F. Beijing

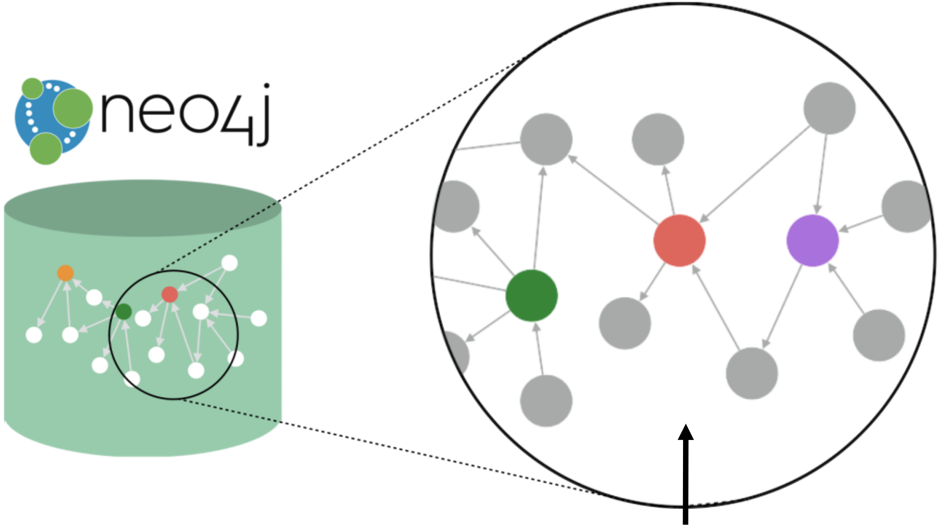
图数据库管理系统

How Neo4j Differentiates from other Databases

	neo4j **Native Graph DB**	**Non-Native Graph DB**	**RDBMS**
Visualization			
Queries	Cypher (graphs)-[are]->(everywhere)	Cypher/Gremlin /Proprietary	SQL
Processing			
Storage		Table Key-Value Column	

Optimized for graph workloads

Native Graph Architecture Advantage

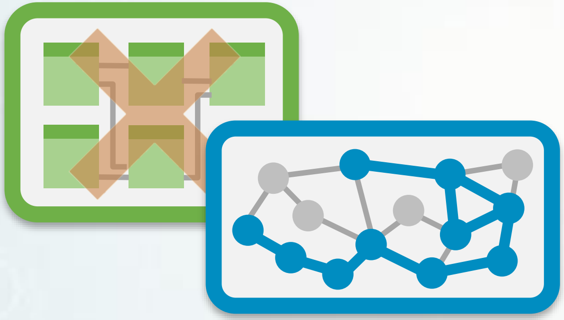


Index free adjacency

Unlike other database models Neo4j *connects* data as it is stored

Index-free adjacency ensures lightning-fast retrieval of data and relationships

Neo4j: Native Graph from the Start



Native graph storage

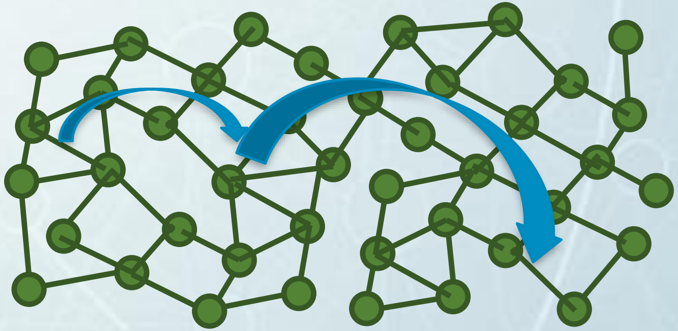
Optimized for real-time reads and ACID writes

- Relationships stored as physical objects, eliminating need for joins and join tables
- Nodes connected at write time, enabling scale-independent response times

Native graph querying

Memory structures and algorithms optimized for graphs

- Index-free adjacency enables 1M+ hops per second via in-memory pointer chasing
- Off-heap page cache improves operational robustness and scaling compared with JVM-based caches
- “Minutes to milliseconds” performance improvement



Neo4j Advantage - Performance **Neo4j Advantage - ACID Transactions**

Traversal Speeds on Amazon Retail Dataset

Social Recommendation Example

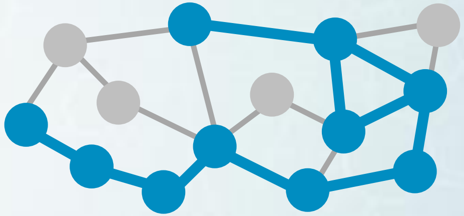
Equivalent Cypher Query

```
 MATCH (you)-[:BOUGHT]->(something)<-[:BOUGHT]-(other)-[:BOUGHT]->(reco)
WHERE id(you)={id}
RETURN reco
```

Threads	Hops per second
1	3-4 million
10	17-29 million
20	34-50 million
30	36-60 million

Relationship Queries Strain Traditional Databases

A single query can touch a ***lot of data***

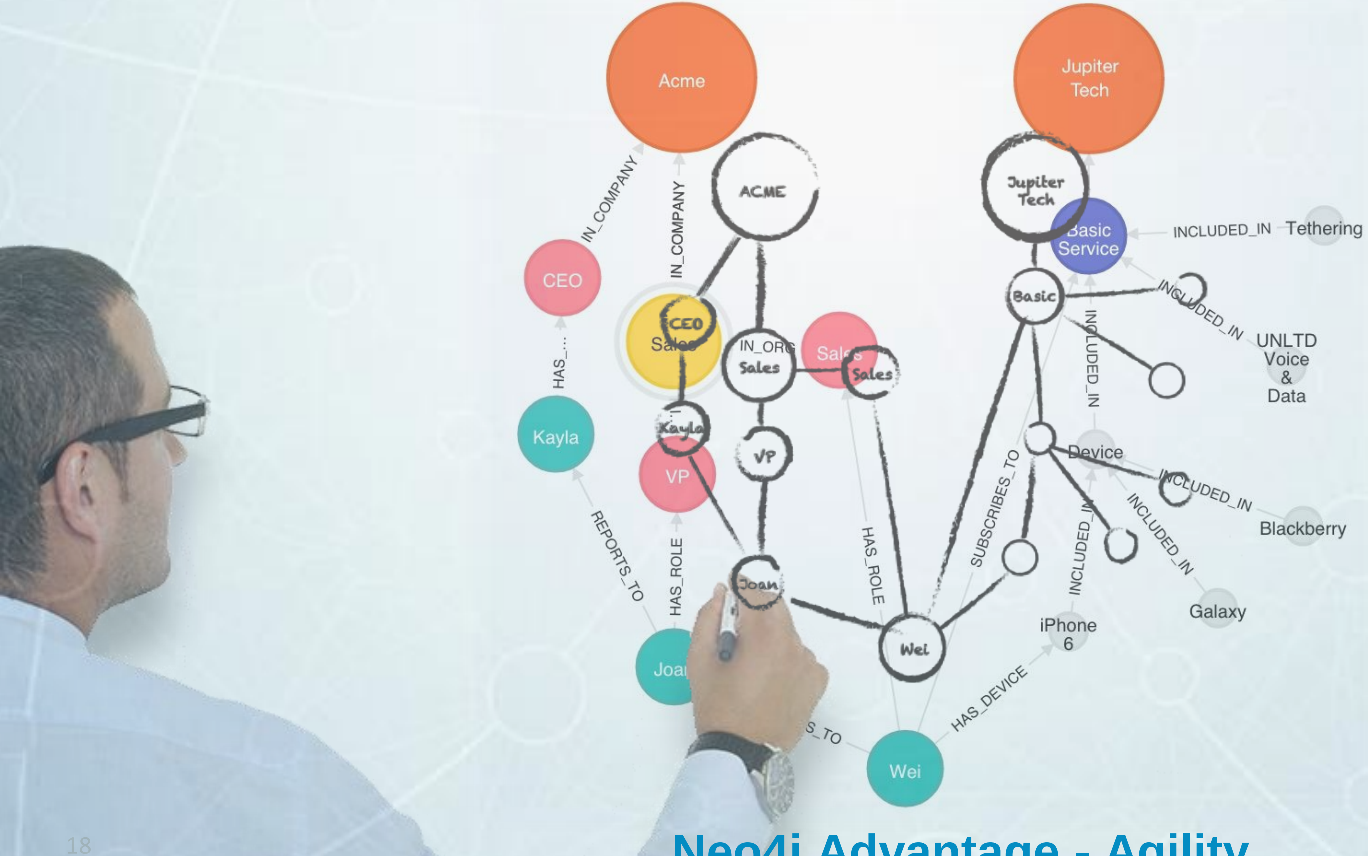


Queries can take non-sequential, ***arbitrary paths*** through data

Real-time queries need speed and consistent response times

Queries must ***run reliably*** with ***consistent results***

Neo4j Property Graph
The Whiteboard Model is the Physical Model



A unified view for ultimate agility

- Easily understood
- Easily evolved
- Easy collaboration between business and IT

Neo4j Advantage - Agility

Cypher 查询语言

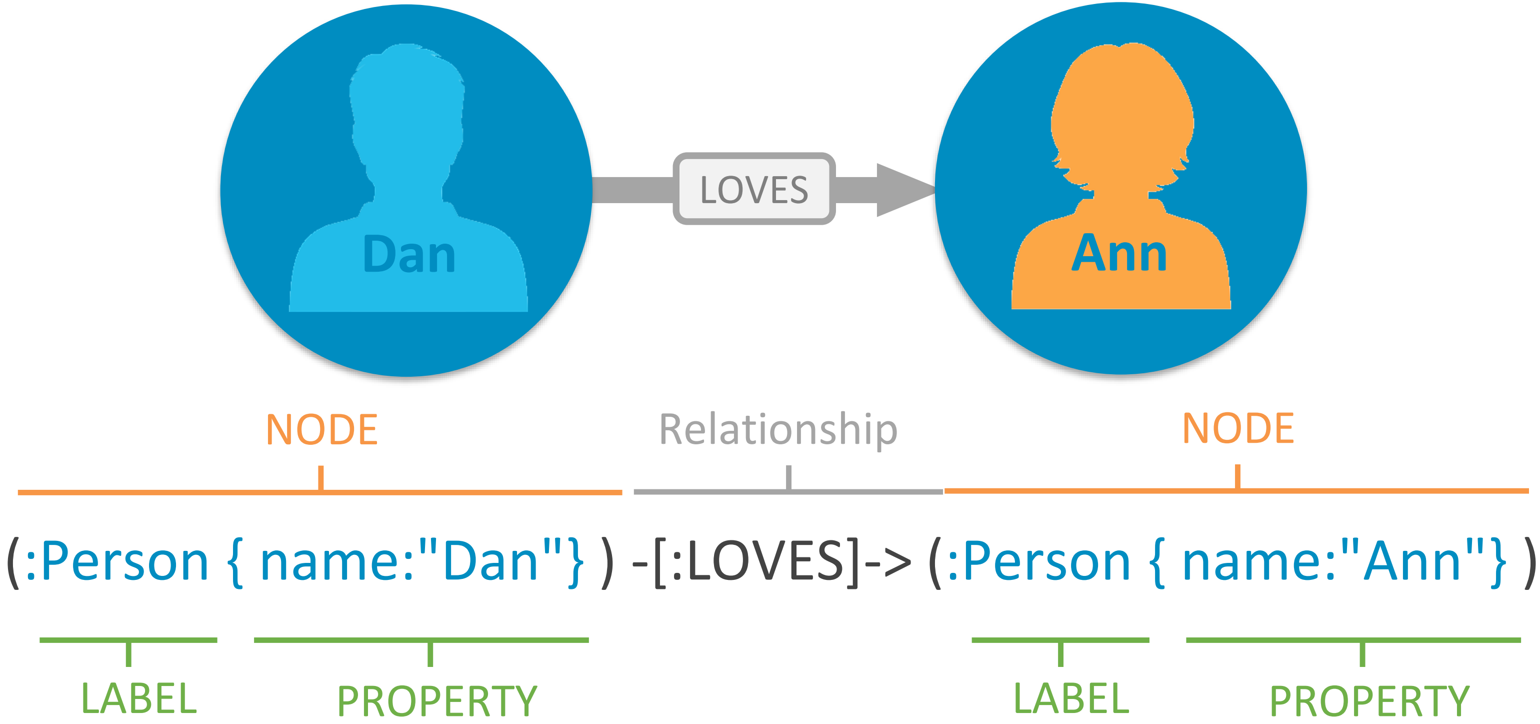
Cypher

A pattern matching query language made for graphs.

- Declarative
- Expressive
- Pattern Matching

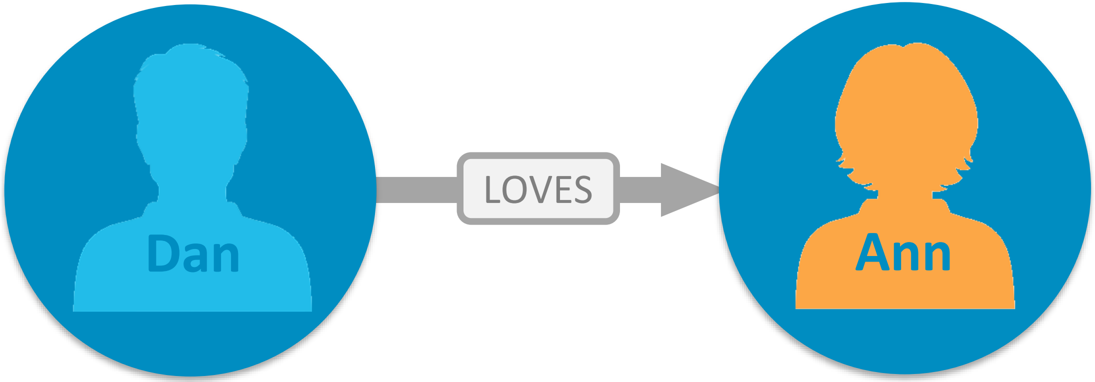
De facto industry standard query language for graph database.

Cypher: Express Graph Patterns



```
(:Person { name:"Dan"} ) -[:LOVES]-> (:Person { name:"Ann"} )
```

Cypher: CREATE Graph Patterns

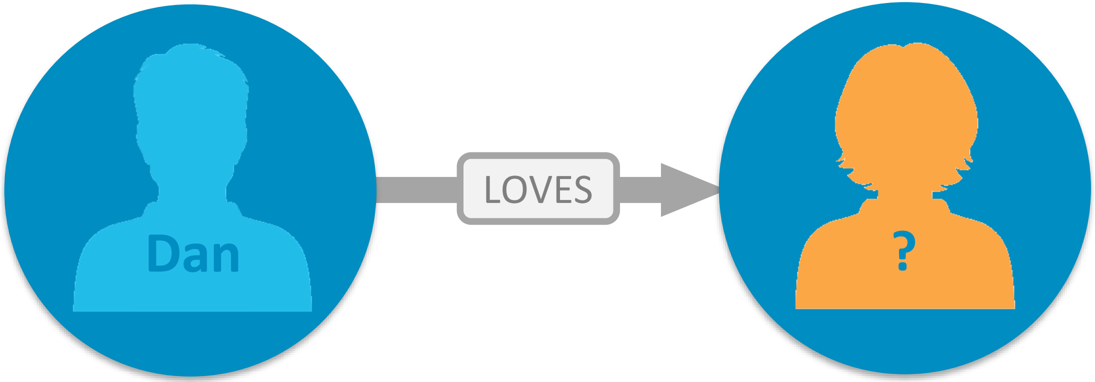


NODE Relationship NODE

```
CREATE (:Person { name:"Dan"} ) -[:LOVES]-> (:Person { name:"Ann"} )
```

LABEL PROPERTY LABEL PROPERTY

Cypher: MATCH Graph Patterns



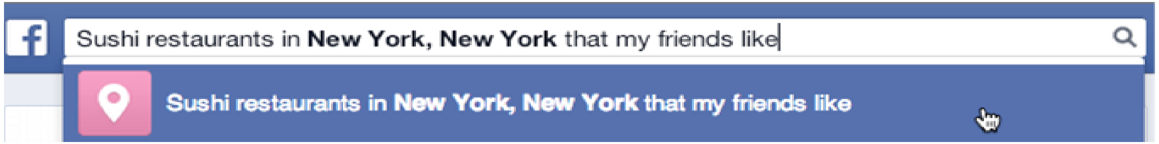
NODE Relationship NODE

```
MATCH (p:Person { name:"Dan"} ) -[:LOVES]-> ( whom ) RETURN whom
```

LABEL PROPERTY VARIABLE

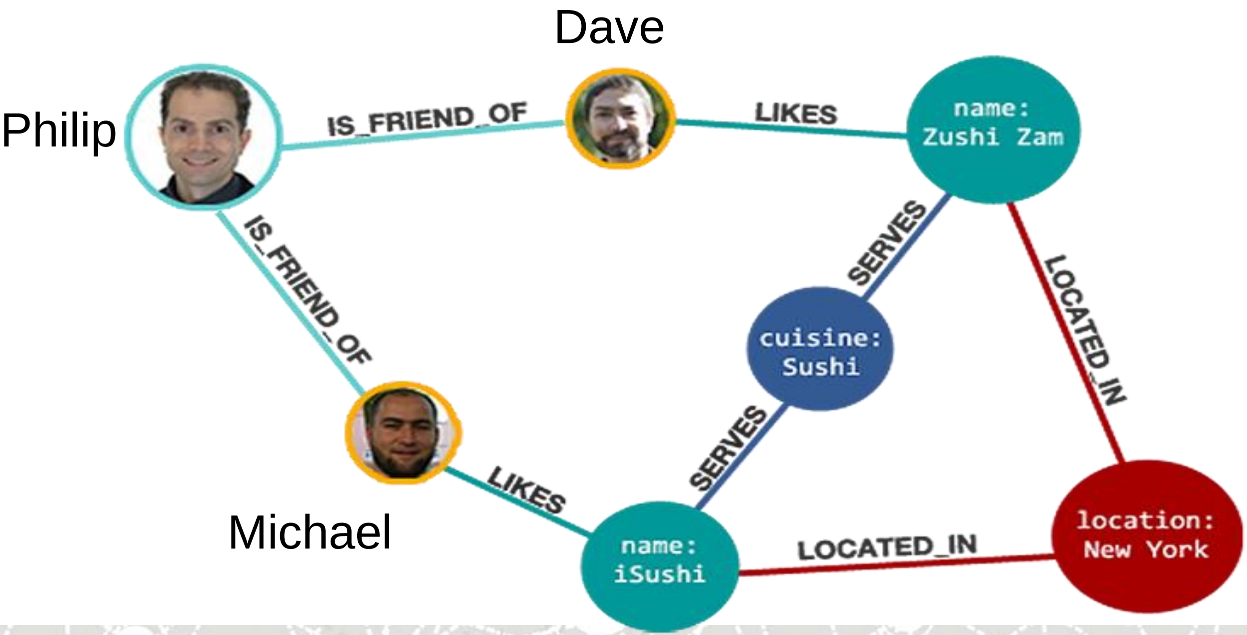
```
WHERE p.Loves_Person_Id = whom.Id
```

A social recommendation

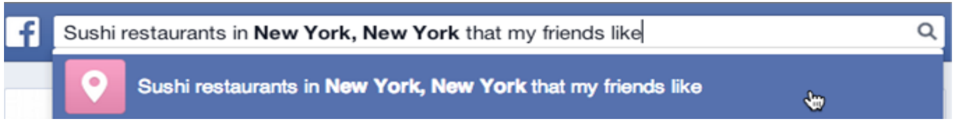


Sushi restaurants in **New York, New York** that my friends like

Sushi restaurants in New York, New York that my friends like



A social recommendation



Sushi restaurants in **New York, New York** that my friends like

Sushi restaurants in **New York, New York** that my friends like

```
MATCH (person:Person)-[:IS_FRIEND_OF]->(friend),
      (friend)-[:LIKES]->(restaurant),
      (restaurant)-[:LOCATED_IN]->(loc:Location),
      (restaurant)-[:SERVES]->(type:Cuisine)
WHERE person.name = 'Philip'
AND loc.location='New York'
AND type.cuisine='Sushi'
RETURN restaurant.name
```

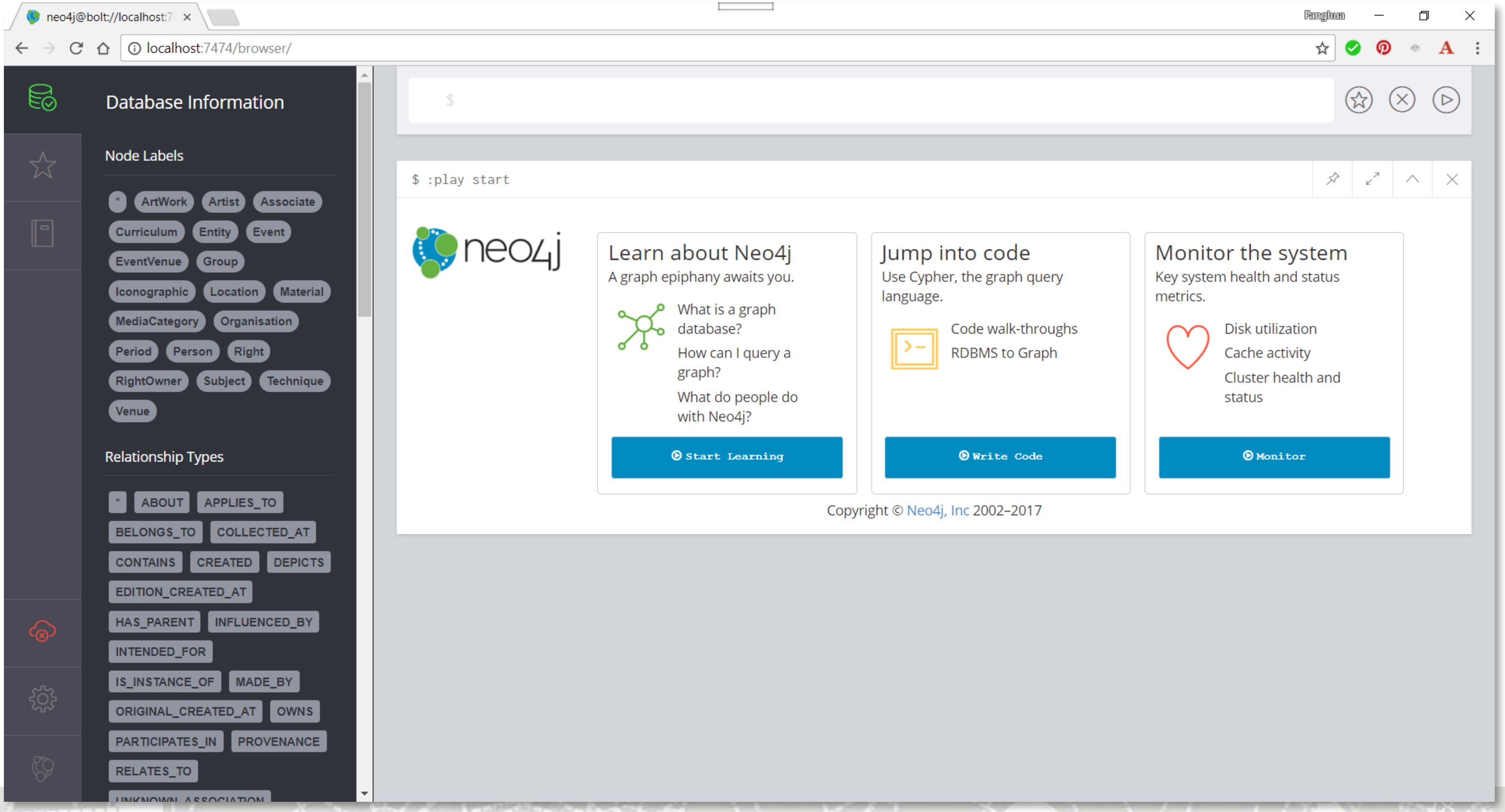
分析工具 – Neo4j Browser

Neo4j Browser

A GUI for power users and developers to access the graph database, and perform administrative tasks using Cypher.

- Running in web browser
- Installed as part of Neo4j database
- Service launched together with database
- To launch, access: http://<hostname>:7474

Neo4j Browser – Browse Database Schema



Database Information

Node Labels

* ArtWork Artist Associate Curriculum Entity Event EventVenue Group Iconographic Location Material MediaCategory Organisation Period Person Right RightOwner Subject Technique Venue

Relationship Types

* ABOUT APPLIES_TO BELONGS_TO COLLECTED_AT CONTAINS CREATED DEPICTS EDITION_CREATED_AT HAS_PARENT INFLUENCED_BY INTENDED_FOR IS_INSTANCE_OF MADE_BY ORIGINAL_CREATED_AT OWNS PARTICIPATES_IN PROVENANCE RELATES_TO UNKNOWN_ASSOCIATION

```
$ :play start
```

neo4j

Learn about Neo4j

A graph epiphany awaits you.

What is a graph database?

How can I query a graph?

What do people do with Neo4j?

Jump into code

Use Cypher, the graph query language.

Code walk-throughs

RDBMS to Graph

Monitor the system

Key system health and status metrics.

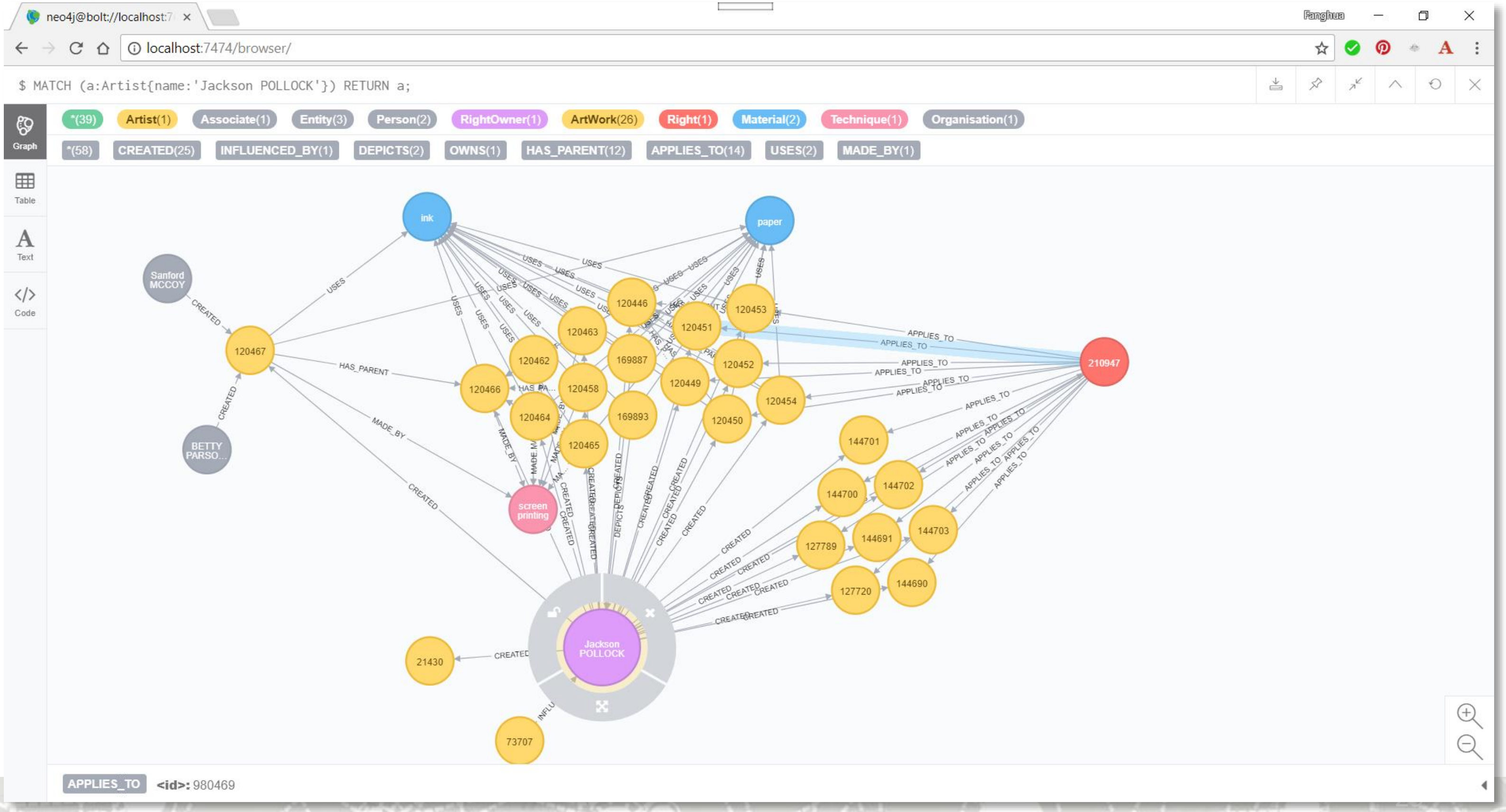
Disk utilization

Cache activity

Cluster health and status

Copyright © Neo4j, Inc 2002–2017

Neo4j Browser – Analyse Data



```
$ MATCH (a:Artist{name:'Jackson POLLOCK'}) RETURN a;
```

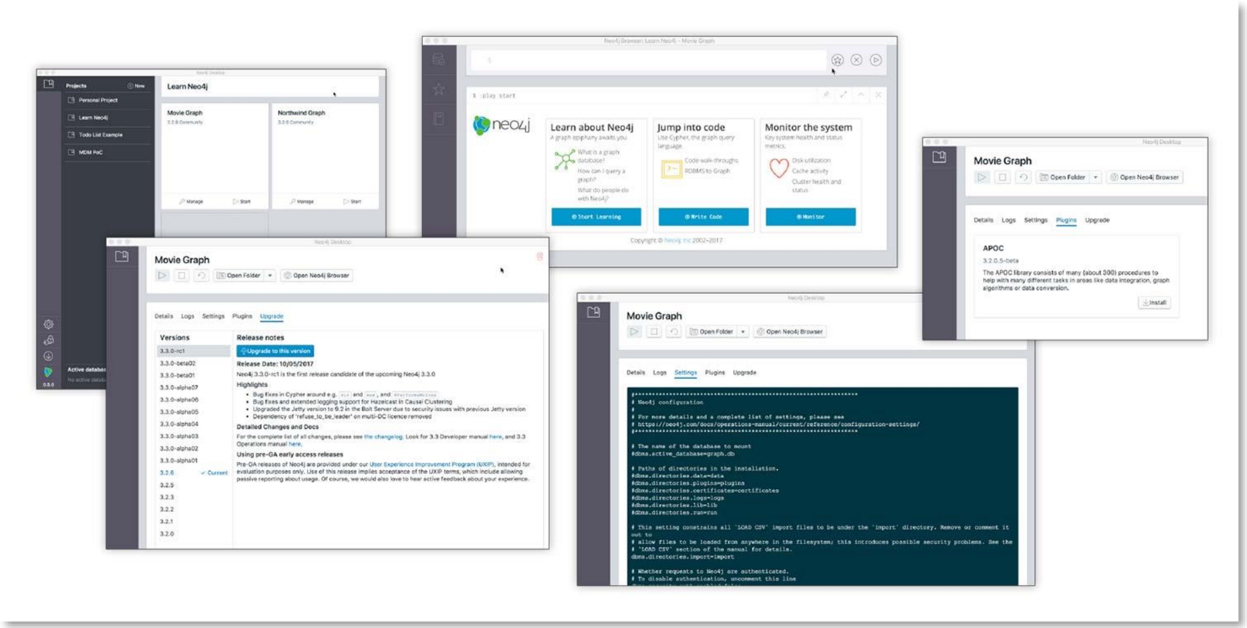
*(39) Artist(1) Associate(1) Entity(3) Person(2) RightOwner(1) ArtWork(26) Right(1) Material(2) Technique(1) Organisation(1)

*(58) CREATED(25) INFLUENCED_BY(1) DEPICTS(2) OWNS(1) HAS_PARENT(12) APPLIES_TO(14) USES(2) MADE_BY(1)

APPLIES_TO <id>: 980469

Neo4j Desktop

Neo4j Desktop (Mac & Windows)



Learn about Neo4j
A graph epiphany awaits you
What is a graph database?
How can I query a graph?
What do people do with Neo4j?

Jump into code
Use Cypher, the graph query language.
Code walk-throughs
RDBMS to Graph

Monitor the system
Key system health and status metrics.
Disk utilization
Cache activity
Cluster health and status

Copyright © Neo4j, Inc 2002-2017

APOC
3.2.0.5-beta
The APOC library consists of many (about 300) procedures to help with many different tasks in areas like data integration, graph algorithms or data conversion.

Versions
3.3.0-rc1
3.3.0-beta02
3.3.0-beta01
3.3.0-alpha07
3.3.0-alpha06
3.3.0-alpha05
3.3.0-alpha04
3.3.0-alpha03
3.3.0-alpha02
3.3.0-alpha01
3.2.6 ✓ Current
3.2.5
3.2.3
3.2.2
3.2.1
3.2.0

Release notes
Upgrade to this version
Release Date: 10/05/2017
Neo4j 3.3.0-rc1 is the first release candidate of the upcoming Neo4j 3.3.0
Highlights
- Bug fixes in Cypher around e.g. `min` and `max`, and `#PerformsWrites`
- Bug fixes and extended logging support for Hazelcast in Causal Clustering
- Upgraded the Jetty version to 9.2 in the Bolt Server due to security issues with previous Jetty version
- Dependency of 'refuse_to_be_leader' on multi-DC licence removed

Detailed Changes and Docs
For the complete list of all changes, please see the changelog. Look for 3.3 Developer manual here, and 3.3 Operations manual here.

Using pre-GA early access releases
Pre-GA releases of Neo4j are provided under our User Experience Improvement Program (UXIP), intended for evaluation purposes only. Use of this release implies acceptance of the UXIP terms, which include allowing passive reporting about usage. Of course, we would also love to hear active feedback about your experience.

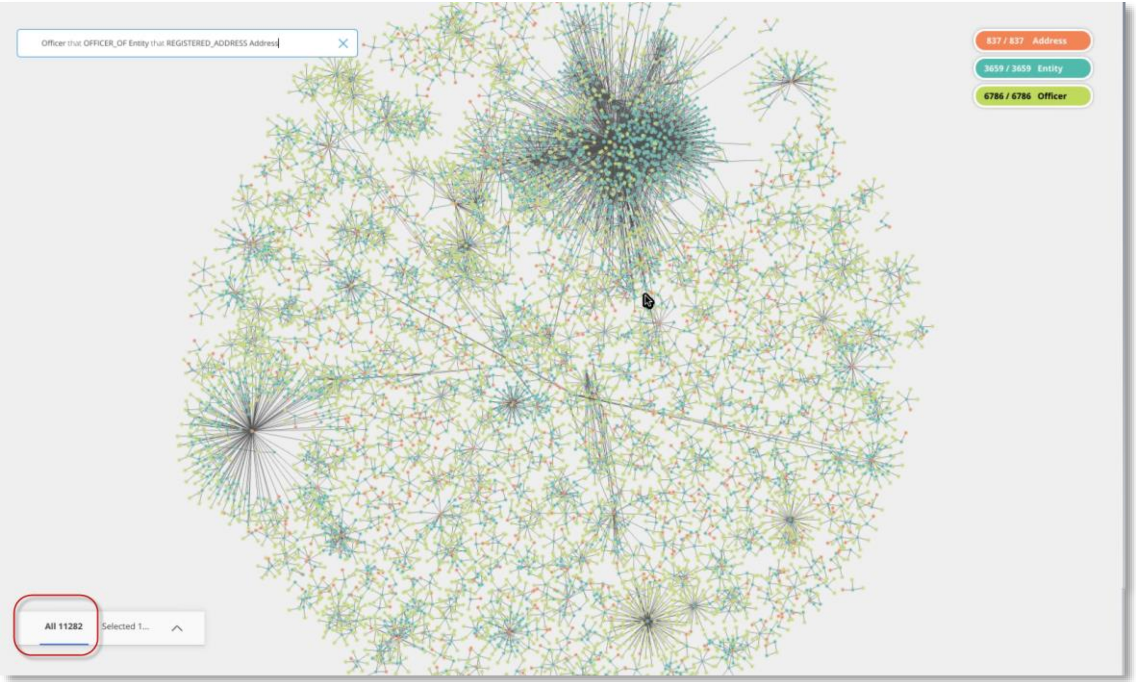
```
#*****************************************************************
# Neo4j configuration
#
# For more details and a complete list of settings, please see
# https://neo4j.com/docs/operations-manual/current/reference/configuration-settings/
#*****************************************************************

# The name of the database to mount
#dbms.active_database=graph.db

# Paths of directories in the installation.
#dbms.directories.data=data
#dbms.directories.plugins=plugins
#dbms.directories.certificates=certificates
#dbms.directories.logs=logs
#dbms.directories.lib=lib
#dbms.directories.run=run

# This setting constrains all `LOAD CSV` import files to be under the `import` directory. Remove or comment it
out to
# allow files to be loaded from anywhere in the filesystem; this introduces possible security problems. See the
# `LOAD CSV` section of the manual for details.
dbms.directories.import=import

# Whether requests to Neo4j are authenticated.
# To disable authentication, uncomment this line
```

- The new mission control centre for developers
- Connecting to all your Neo4j databases within a single application
- Installing extensions like graph algorithms or ETL components
- Free to download, bundled with an Enterprise version of Neo4j for Developers

Visualization

Neo4j Bloom* – To be released in Q2 18

- A powerful tool for business users to explore data and connections
- Support of natural language input
- Support of editing nodes and relationships on the graph
- And many more…

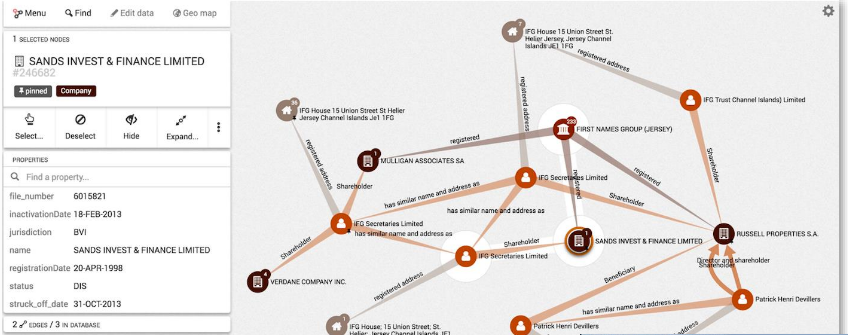


Officer that OFFICER_OF Entity that REGISTERED_ADDRESS Address

** Note: at this stage all features mentioned are for your references ONLY, and subject to change due to schedule and priority.*

3rd Party Tools

- Linkurious



Menu
Find
Edit data
Geo map
Select...
Deselect
Hide
Expand...
IFG House 15 Union Street St. Helier Jersey, Jersey Channel Islands JE1 1FG
registered address
IFG House 15 Union Street St Helier Jersey Channel Islands Jel 1FG
IFG Trust Channel Islands) Limited
FIRST NAMES GROUP (JERSEY)
registered
MULLIGAN ASSOCIATES SA
Shareholder
IFG Secretaries Limited
has similar name and address as
IFG Secretaries Limited
SANDS INVEST & FINANCE LIMITED
RUSSELL PROPERTIES S.A.
Director and shareholder
VERDANE COMPANY INC.
Beneficiary
Patrick Henri Devillers
IFG House; 15 Union Street; St. Helier; Jersey Channel Islands JF1 1FG
LINKURIOUS

1 SELECTED NODES

SANDS INVEST & FINANCE LIMITED
#246682

pinned Company

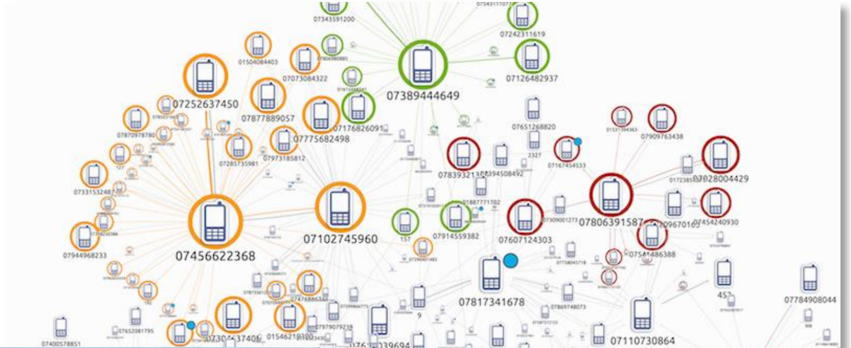
PROPERTIES

Find a property...

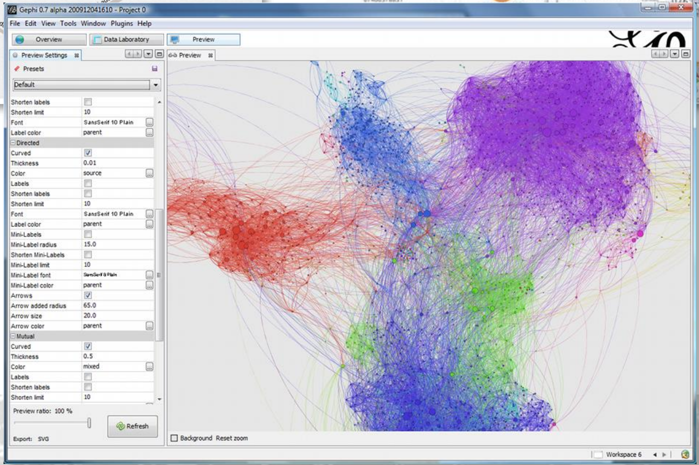
file_number	6015821
inactivationDate	18-FEB-2013
jurisdiction	BVI
name	SANDS INVEST & FINANCE LIMITED
registrationDate	20-APR-1998
status	DIS
struck_off_date	31-OCT-2013

2 EDGES / 3 IN DATABASE

- Keyline



- Gephi



Default

Shorten labels	□
Shorten limit	10
Font	SansSerif 10 Plain
Label color	parent
Directed	
Curved	☑
Thickness	0.01
Color	source
Labels	□
Shorten labels	□
Shorten limit	10
Font	SansSerif 10 Plain
Label color	parent
Mini-Labels	□
Mini-Label radius	15.0
Shorten Mini-Labels	□
Mini-Label limit	10
Mini-Label font	SansSerif 8 Plain
Mini-Label color	parent
Arrows	☑
Arrow added radius	65.0
Arrow size	20.0
Arrow color	parent
Mutual	
Curved	☑
Thickness	0.5
Color	mixed
Labels	□
Shorten labels	□
Shorten limit	10

Preview ratio: 100 %

Export: SVG

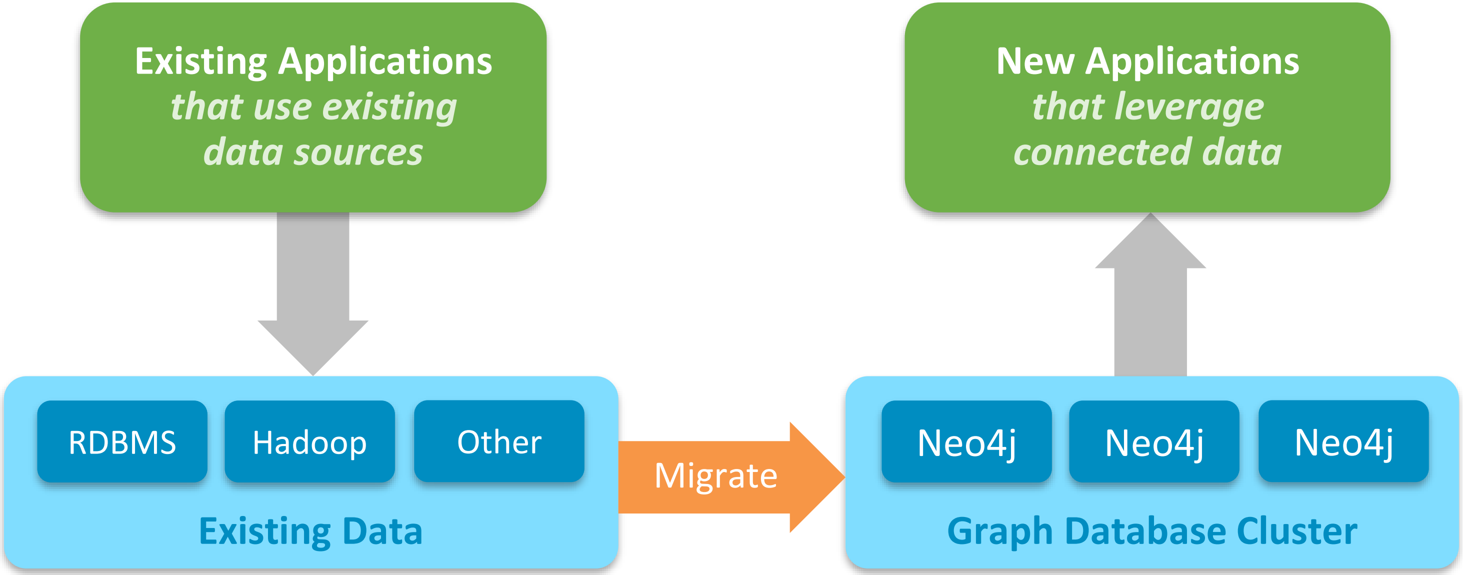
Open Source Libraries

- Sigma
- NeoVis
- D3.js
- Cytoscape

Do a search on Github and you will find several more.

与企业的现有系统集成

Neo4j in your Environment
Usage Pattern 1 – 共存



Existing Applications
that use existing data sources

New Applications
that leverage connected data

RDBMS

Hadoop

Other

Existing Data

Migrate

Neo4j

Neo4j

Neo4j

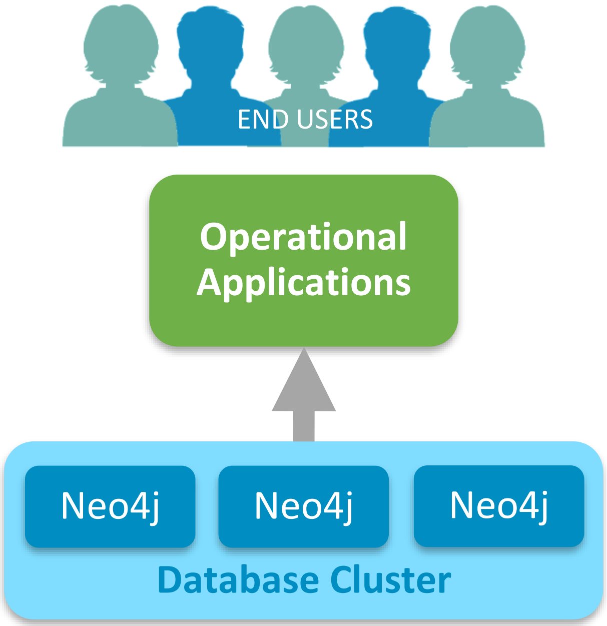
Graph Database Cluster

Load connected data to Neo4j using ETL or Messaging system

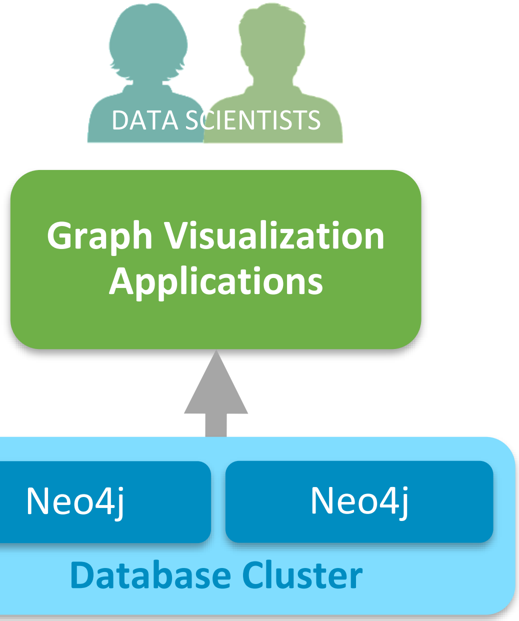
Neo4j in your Environment

Usage Pattern 2 – 分治

Operational Applications

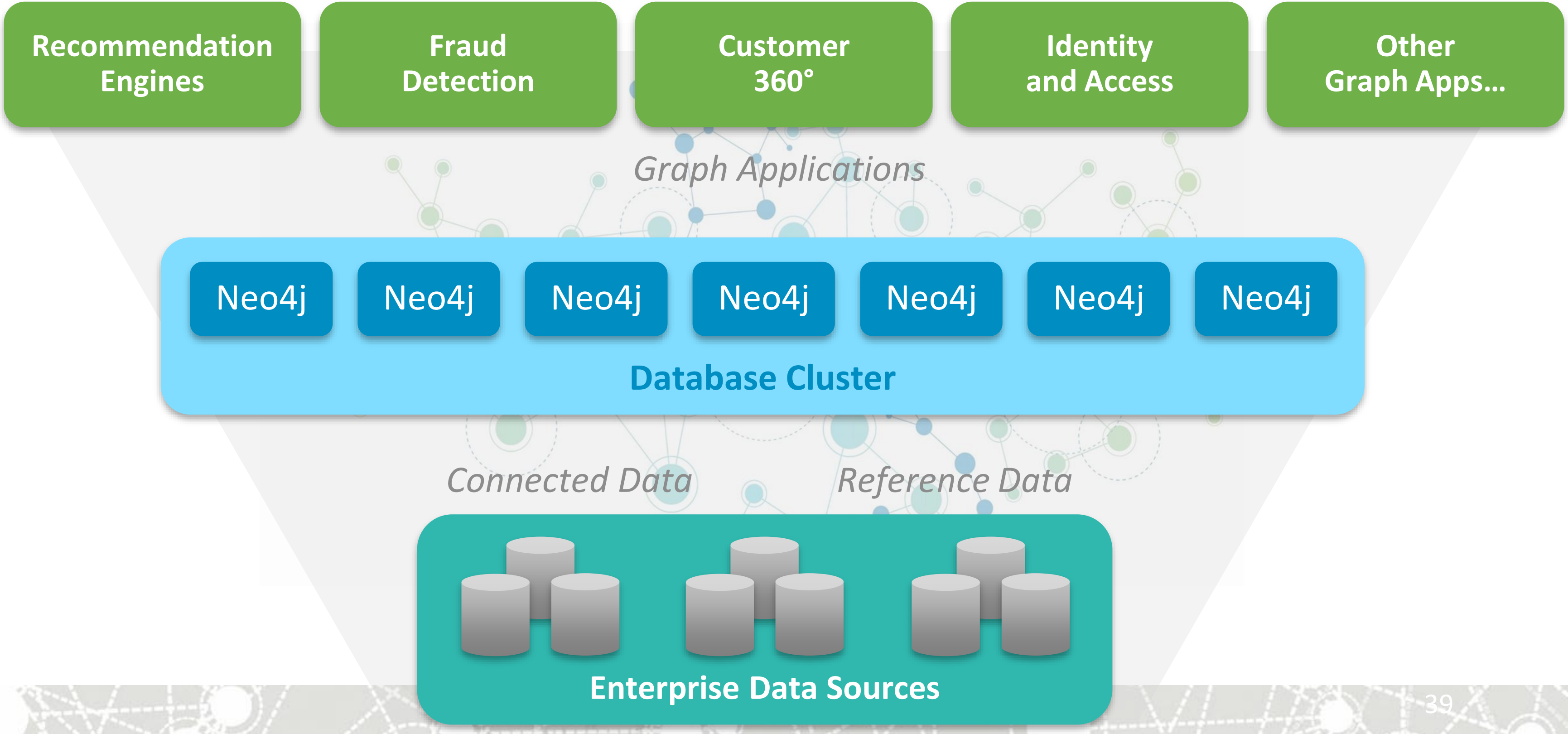


Analytic Applications



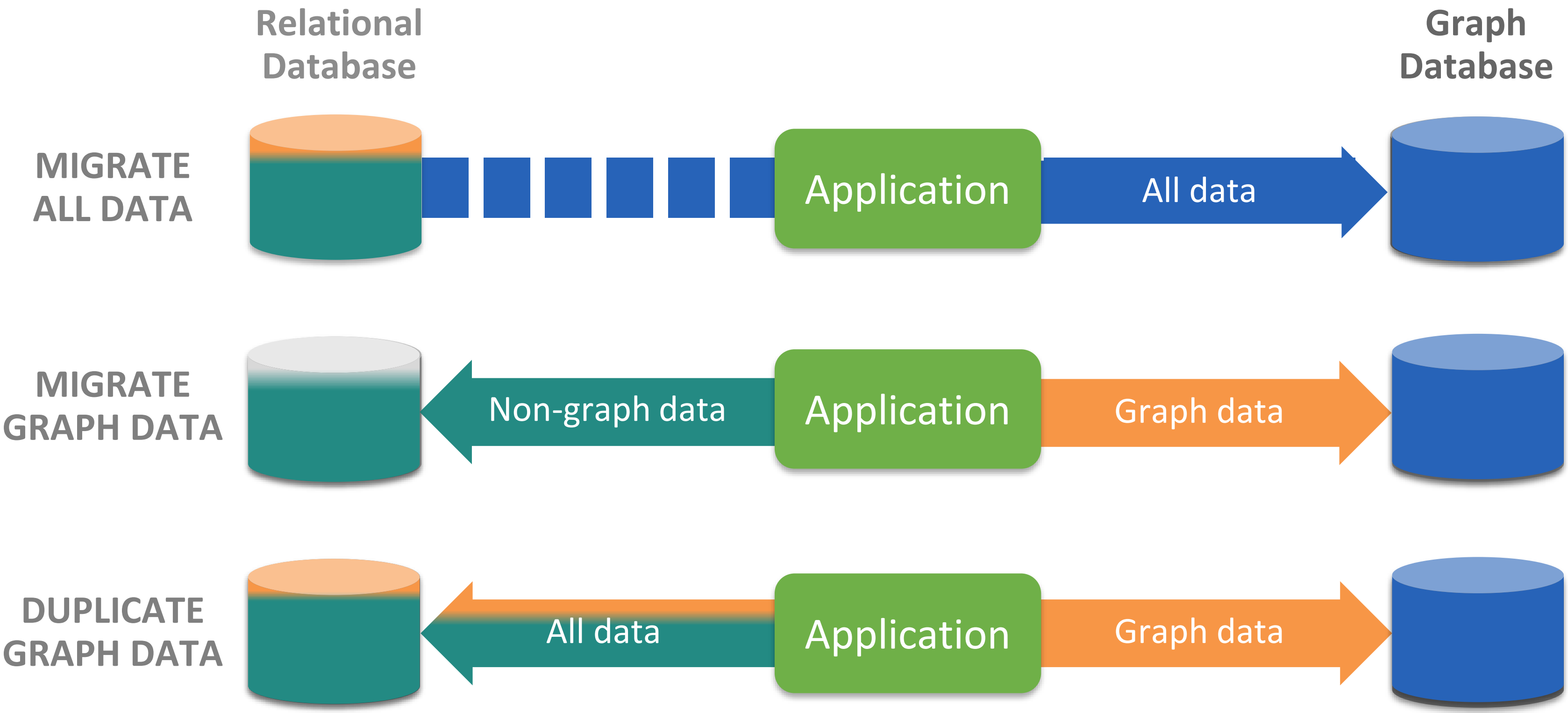
Custom (D3, Vizjs)
Linkurious
Tom Sawyer

Neo4j in your Environment
Usage Pattern 3 – 联合



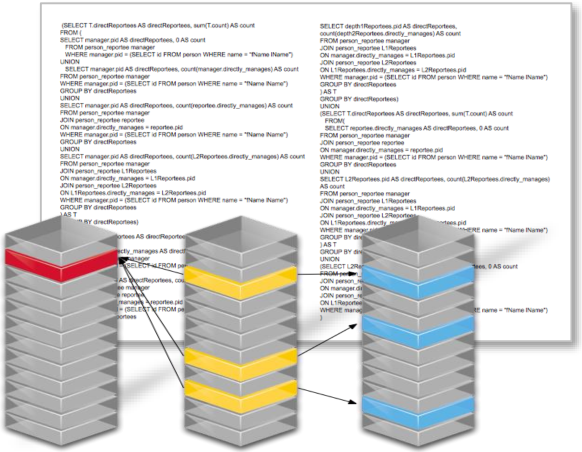
数据迁移/更新

Ways to migrate data to Neo4j



Accessing Relational Data

- ***Dump to CSV*** all relational database have the option to dump query results and tables to CSV
- ***Access with DB-Driver*** access DB with JDBC/ODBC or other driver to pull out selected datasets
- ***Use APOC procedures*** apoc.load.jdbc
- ***Use built-in or external endpoints*** some databases expose HTTP-APIs or can be integrated
- ***Use ETL-Tools*** existing ETL Tools can read from relational and write to Neo4j e.g. via JDBC



```
(SELECT T.directReportees AS directReportees, sum(T.count) AS count
FROM (
SELECT manager.pid AS directReportees, 0 AS count
  FROM person_reportee manager
  WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
UNION
  SELECT manager.pid AS directReportees, count(manager.directly_manages) AS count
FROM person_reportee manager
WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
GROUP BY directReportees
UNION
SELECT manager.pid AS directReportees, count(reportee.directly_manages) AS count
FROM person_reportee manager
JOIN person_reportee reportee
ON manager.directly_manages = reportee.pid
WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
GROUP BY directReportees
UNION
SELECT manager.pid AS directReportees, count(L2Reportees.directly_manages) AS count
FROM person_reportee manager
JOIN person_reportee L1Reportees
ON manager.directly_manages = L1Reportees.pid
JOIN person_reportee L2Reportees
ON L1Reportees.directly_manages = L2Reportees.pid
WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
GROUP BY directReportees
) AS T
```

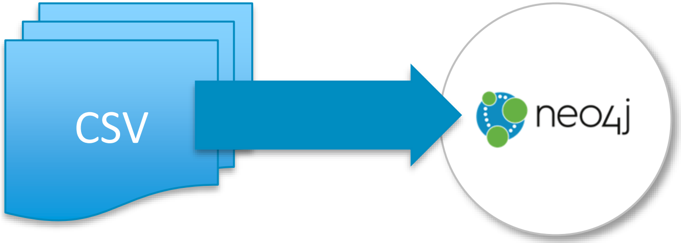
```
SELECT depth1Reportees.pid AS directReportees,
count(depth2Reportees.directly_manages) AS count
FROM person_reportee manager
JOIN person_reportee L1Reportees
ON manager.directly_manages = L1Reportees.pid
JOIN person_reportee L2Reportees
ON L1Reportees.directly_manages = L2Reportees.pid
WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
GROUP BY directReportees
) AS T
GROUP BY directReportees)
UNION
(SELECT T.directReportees AS directReportees, sum(T.count) AS count
  FROM(
  SELECT reportee.directly_manages AS directReportees, 0 AS count
FROM person_reportee manager
JOIN person_reportee reportee
ON manager.directly_manages = reportee.pid
WHERE manager.pid = (SELECT id FROM person WHERE name = "fName lName")
GROUP BY directReportees
UNION
SELECT L2Reportees.pid AS directReportees, count(L2Reportees.directly_manages)
AS count
FROM person_reportee manager
JOIN person_reportee L1Reportees
ON manager.directly_manages = L1Reportees.pid
JOIN person_reportee L2Reportees
```

Getting Data into Neo4j: CSV

Cypher-Based “LOAD CSV” Capability

10M

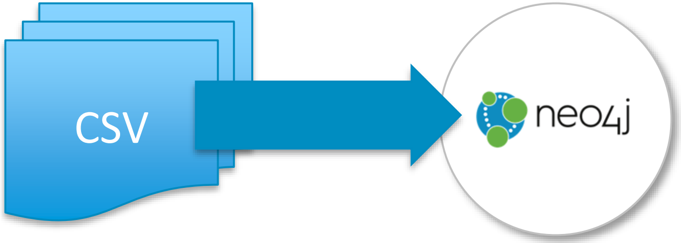
- Transactional (ACID) writes
- Initial and incremental loads of up to 10 million nodes and relationships
- From HTTP and Files
- Power of Cypher
  - Create and Update Graph Structures
  - Data conversion, filtering, aggregation
  - Destructuring of Input Data
- Transaction Size Control
- Also via Neo4j-Shell / Cypher-Shell



Getting Data into Neo4j: CSV

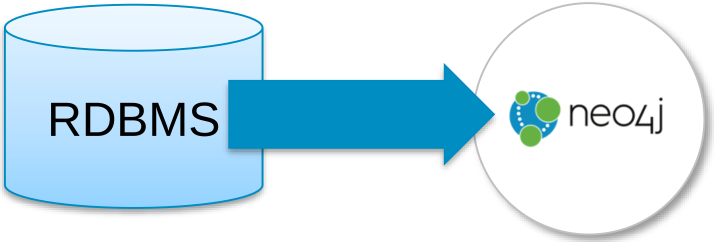
Command line bulk loader - `neo4j-admin import`

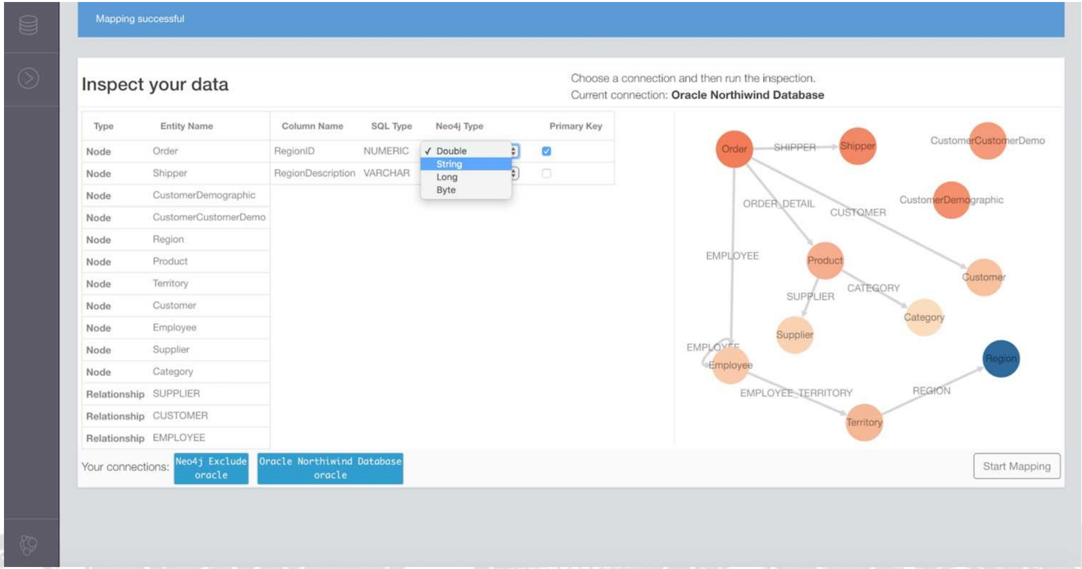
- For initial database population
- Scale across CPUs and disk performance
- Efficient RAM usage
- Split- and compressed file support
- For loads up to 10B+ records
- Up to 1M records per second



Neo4j ETL

- A GUI to create ETL mapping and jobs
- Loaded from Neo4j Desktop
- Reading data using JDBC from sources
- Landing data as CSV files
- Loading data into Neo4j by scheduled jobs





Mapping successful

Inspect your data

Choose a connection and then run the inspection.
Current connection: **Oracle Northiwind Database**

Type	Entity Name
Node	Order
Node	Shipper
Node	CustomerDemographic
Node	CustomerCustomerDemo
Node	Region
Node	Product
Node	Territory
Node	Customer
Node	Employee
Node	Supplier
Node	Category
Relationship	SUPPLIER
Relationship	CUSTOMER
Relationship	EMPLOYEE

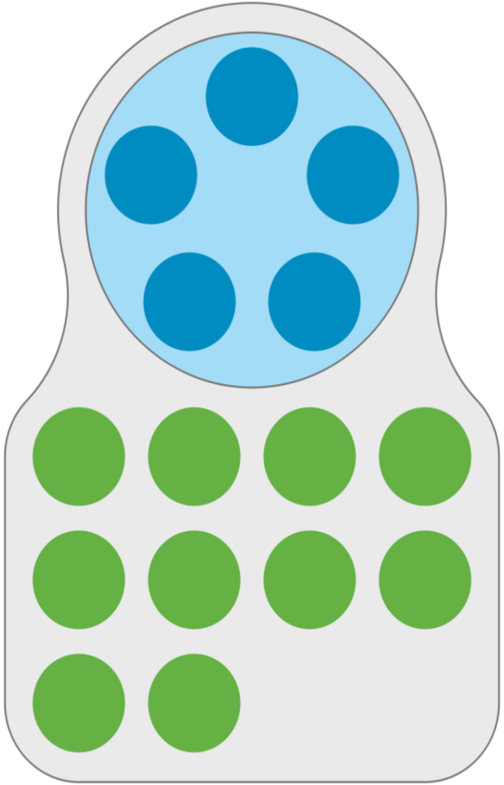
Column Name	SQL Type	Neo4j Type	Primary Key
RegionID	NUMERIC	✓ Double String Long Byte	☑
RegionDescription	VARCHAR		□

Your connections: Neo4j Exclude oracle | Oracle Northiwind Database oracle

Start Mapping

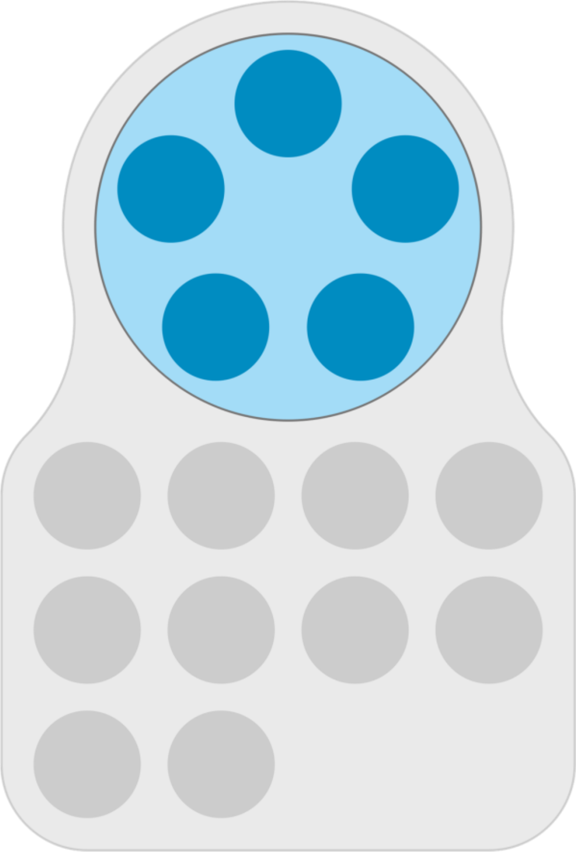
集群(Clustering)

因果集群(Causal Clustering)



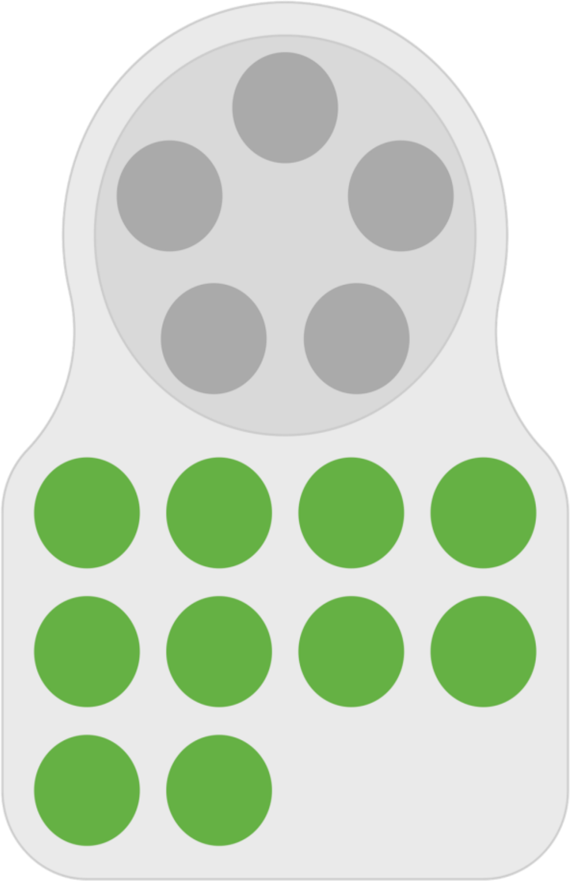
Core

Replica



Core

- Small group of Neo4j databases
- Implements Consensus Commit
- Responsible for data safety

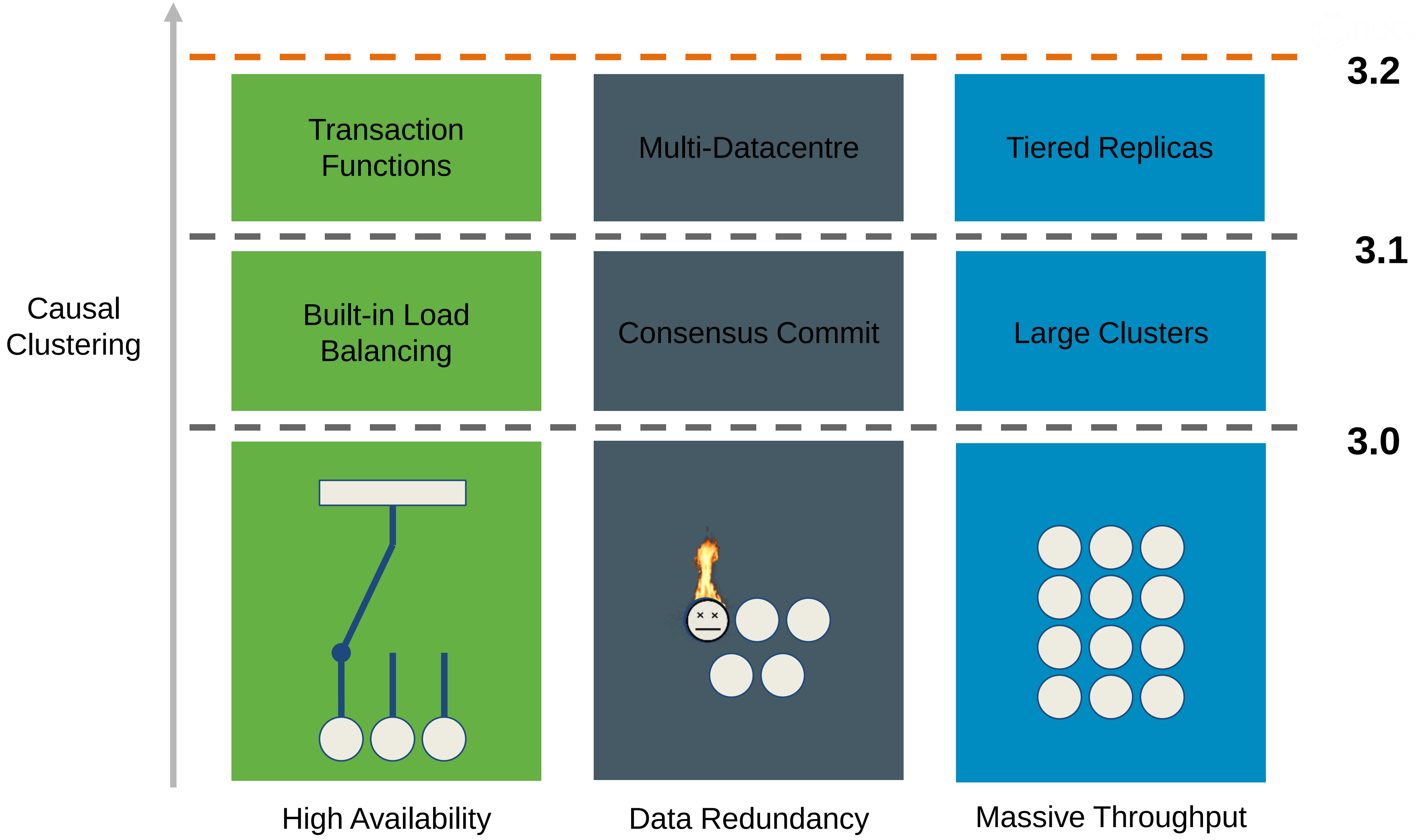


Replica

- For massive query throughput
- Read-only replicas
- Not involved in Consensus Commit => less overhead
- Disposable, suitable for auto-scaling

Causal Clustering - Features

- Two Zones – Core + Edge
- Group of **Core Servers** – Consistent and Partition tolerant (CP)
  - Transactional Writes
  - Quorum Writes, Cluster Membership, Leader via Raft Consensus
- Scale out with **Read Replicas**
- Smart **Bolt Drivers** with
  - Routing, Read & Write Sessions
  - Causal Consistency with Bookmarks



Causal Clustering

	High Availability	Data Redundancy	Massive Throughput
3.2	Transaction Functions	Multi-Datacentre	Tiered Replicas
3.1	Built-in Load Balancing	Consensus Commit	Large Clusters
3.0			

部署(Deployment)

Neo4j Supported Platforms

neo4j

On-Premise Platforms

Linux

Linux
IBM POWER

solaris

Windows

hp UX

macOS
For Development

Cloud Platforms and Containers

amazon web services™

Microsoft Azure

docker

… and others

开发(Development)

Neo4j的编程语言驱动

- Hosted on GitHub
- Apache licensed
- Versioned and released independently of Neo4j
- One driver can support multiple server versions
- Quarterly releases (on average)
- Published to Maven Central, PyPI, npm, etc

Neo4j Drivers

Java

```
<dependency>
    <groupId>org.neo4j.driver</groupId>
    <artifactId>neo4j-java-driver</artifactId>
    <version>X.Y.Z</version>
</dependency>
```

Python

```
pip   install   neo4j-driver
```

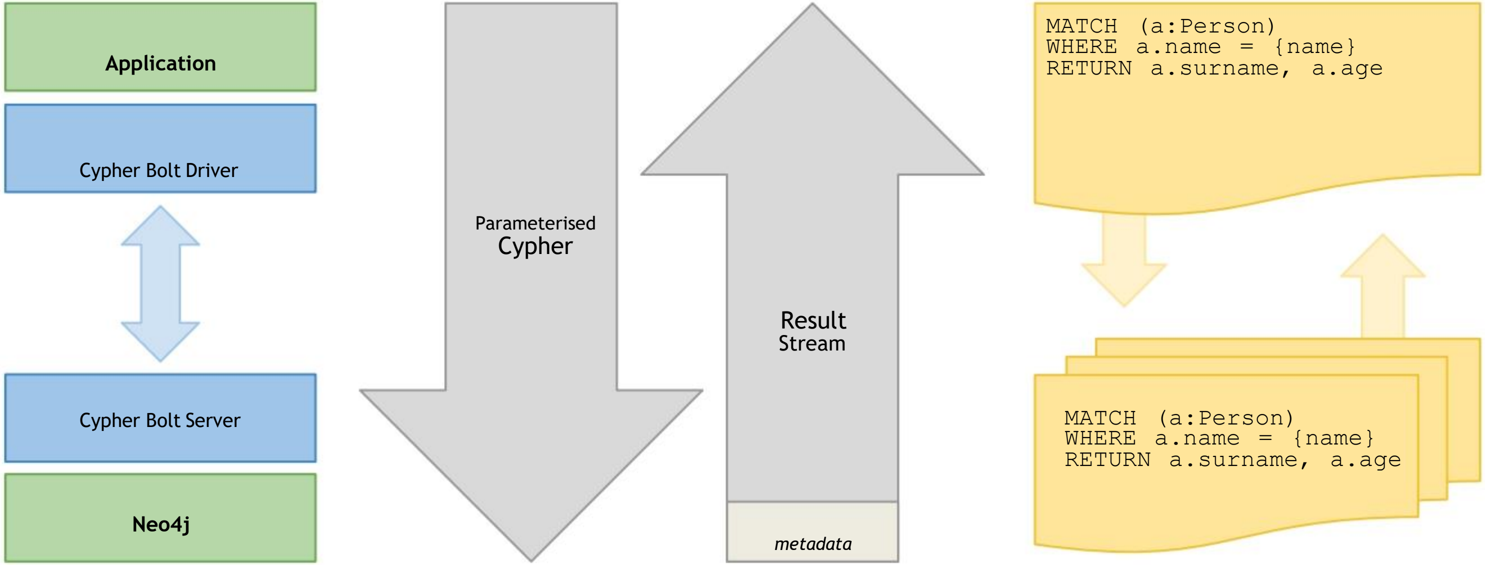
.NET

```
PM>   Install-Package   Neo4j.Driver
```

Javascript

```
npm   install   neo4j-driver
```

Data Flow



```
MATCH (a:Person)
WHERE a.name = {name}
RETURN a.surname, a.age
```

```
MATCH (a:Person)
WHERE a.name = {name}
RETURN a.surname, a.age
```

使用Java访问Neo4j数据库的方法

- Official Driver
- User Defined Functions & Procedures
- Core API
- Neo4j-OGM
- Spring Data Neo4j
- ~~Unmanaged Extensions~~

Resources

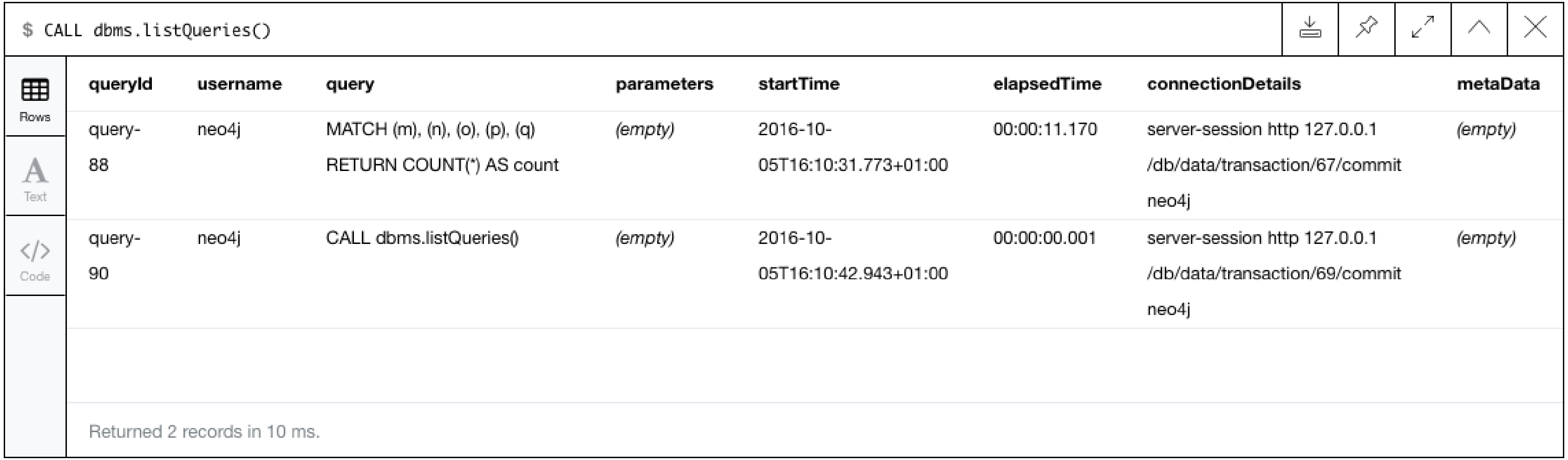
- https://neo4j.com/developer/java/
- https://neo4j.com/docs/java-reference/current
- https://neo4j.com/docs/developer-manual/current/extending-neo4j/procedures/
- https://neo4j.com/docs/operations-manual/current/security/securing-extensions/
- https://projects.spring.io/spring-data-neo4j/

运营支持

Currently running queries

List the running queries :

```
CALL dbms.listQueries()
```



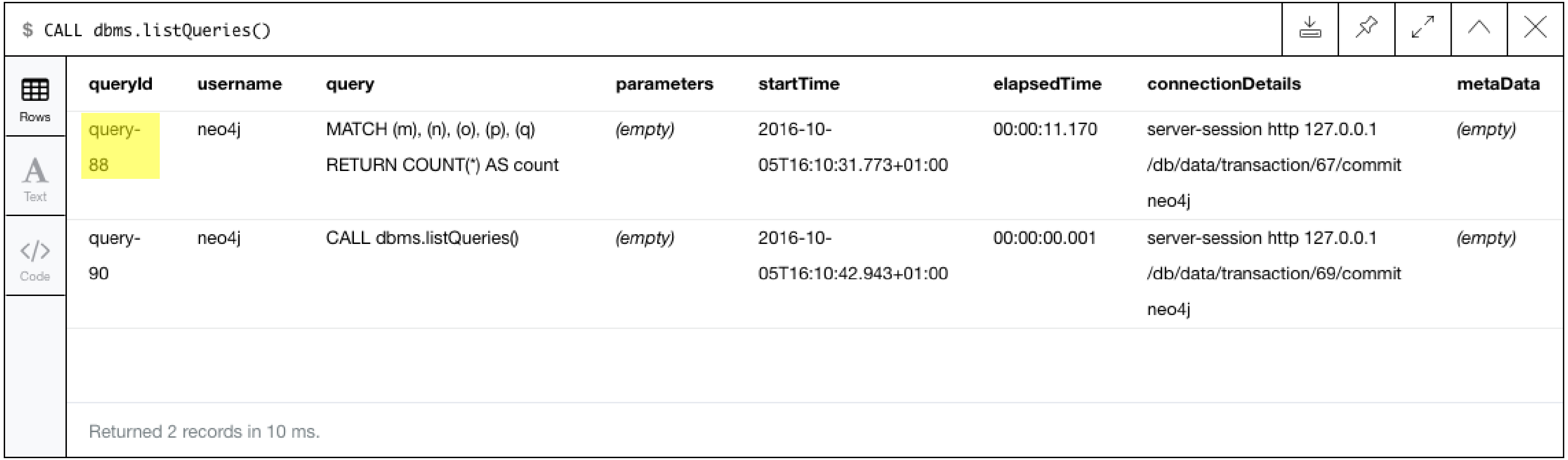
```
$ CALL dbms.listQueries()
```

queryId	username	query	parameters	startTime	elapsedTime	connectionDetails	metaData
query-88	neo4j	MATCH (m), (n), (o), (p), (q) RETURN COUNT(*) AS count	*(empty)*	2016-10-05T16:10:31.773+01:00	00:00:11.170	server-session http 127.0.0.1 /db/data/transaction/67/commit neo4j	*(empty)*
query-90	neo4j	CALL dbms.listQueries()	*(empty)*	2016-10-05T16:10:42.943+01:00	00:00:00.001	server-session http 127.0.0.1 /db/data/transaction/69/commit neo4j	*(empty)*

Returned 2 records in 10 ms.

Kill running queries

We can also kill the query manually using the kill query procedure:



```
$ CALL dbms.listQueries()
```

queryId	username	query	parameters	startTime	elapsedTime	connectionDetails	metaData
query-88	neo4j	MATCH (m), (n), (o), (p), (q) RETURN COUNT(*) AS count	*(empty)*	2016-10-05T16:10:31.773+01:00	00:00:11.170	server-session http 127.0.0.1 /db/data/transaction/67/commit neo4j	*(empty)*
query-90	neo4j	CALL dbms.listQueries()	*(empty)*	2016-10-05T16:10:42.943+01:00	00:00:00.001	server-session http 127.0.0.1 /db/data/transaction/69/commit neo4j	*(empty)*

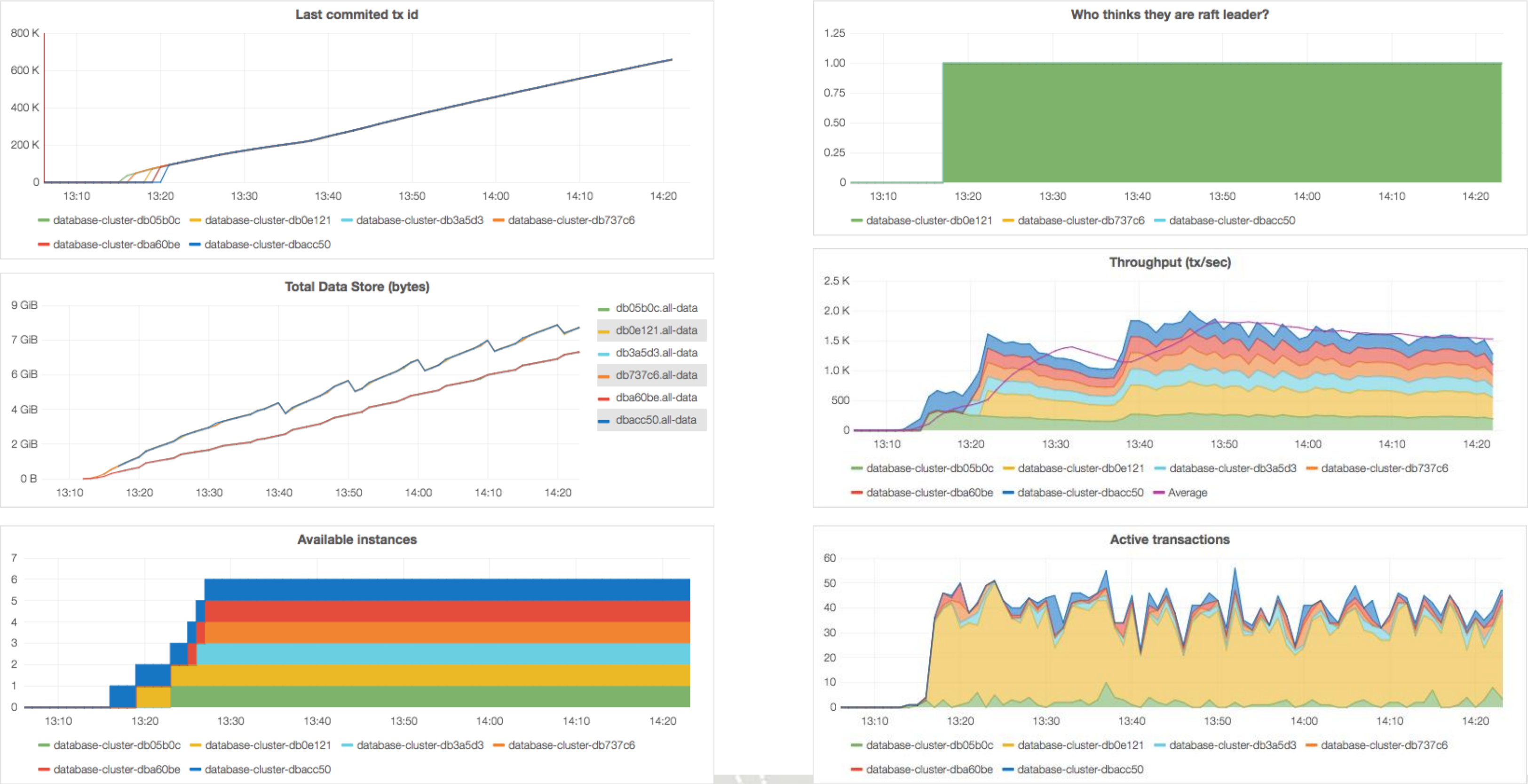
Returned 2 records in 10 ms.

```
CALL dbms.killQuery("query-88")
```

Enabling JMX metrics

```
// default setting for enabling all supported metrics

metrics.enabled=true

metrics.graphite.enabled=true

metrics.graphite.server=<ip>:2003

metrics.graphite.interval=<how often to send data, defaults to 3s>

metrics.prefix=<Neo4j instance name, e.g. wwwneo1>
```

Grafana as the Monitoring dashboard



用户自定义过程

安全性

Flexible Authentication Options

Choose authentication method

- Built-in native users repository

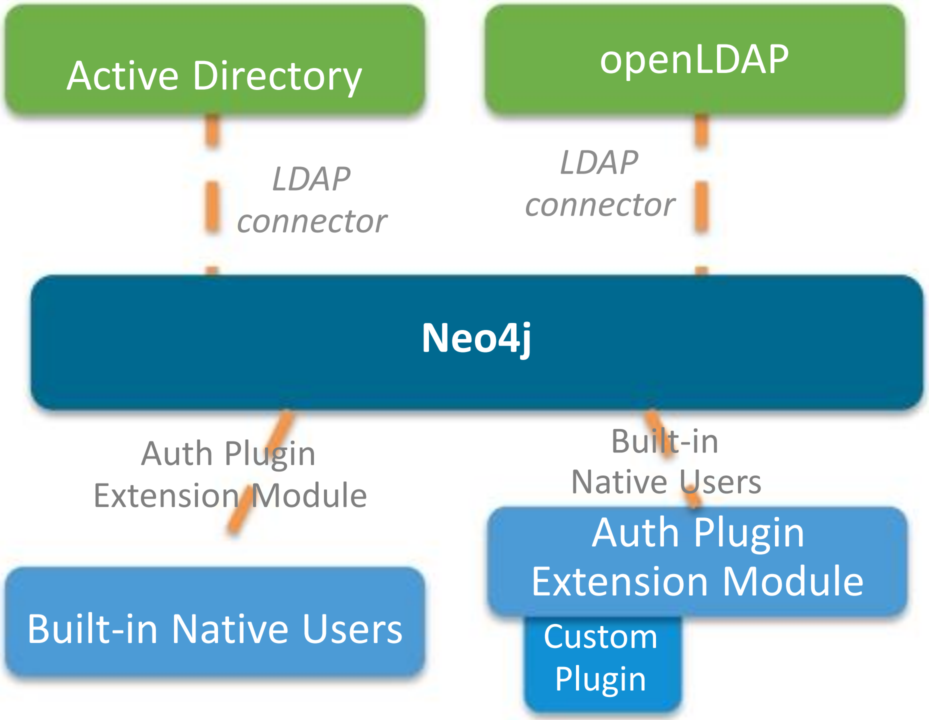
  Testing/POC, single-instance deployments

- LDAP connector to Active Directory or openLDAP

  Production deployments

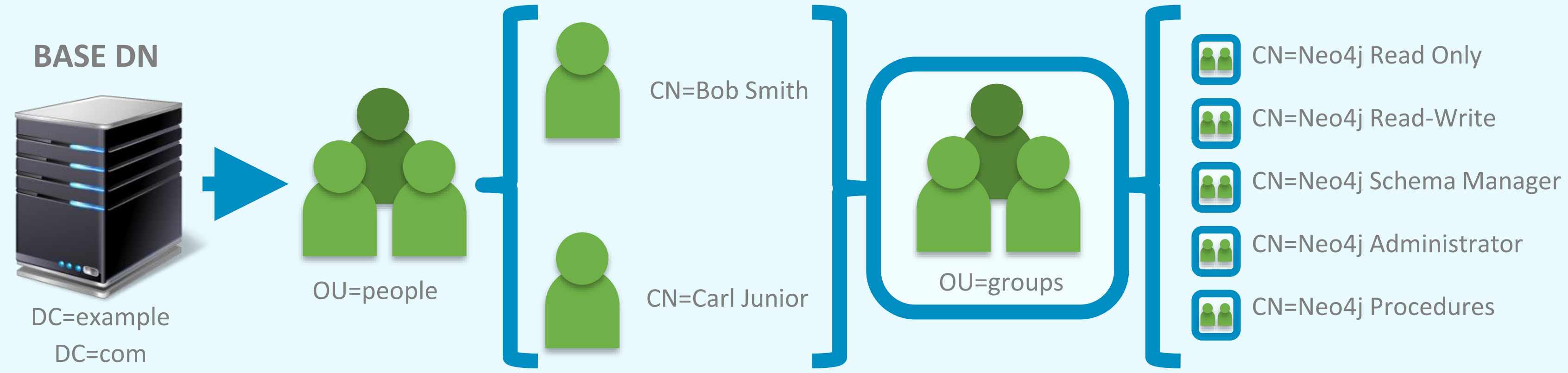
- Custom auth provider plugins

  Special deployment scenarios



Flexible Authentication Options

LDAP Group to Role Mapping



```
# Configure the actual mapping between groups in the LDAP and roles in
Neo4j
dbms.security.ldap.authorization.group_to_role_mapping= \
  "CN=Neo4j Read Only,OU=groups,DC=example,DC=com"       = reader; \
  "CN=Neo4j Read-Write,OU=groups,DC=example,DC=com"      = publisher; \
  "CN=Neo4j Schema Manager,OU=groups,DC=example,DC=com" = architect; \
  "CN=Neo4j Administrator,OU=groups,DC=example,DC=com"  = admin; \
  "CN=Neo4j Procedures,OU=groups,DC=example,DC=com"      = allowed_role
```

./conf/neo4j.conf

业务应用场景

Common Graph Technology Use Cases

Real-Time Recommendations	Fraud Detection	Network & IT Operations	Master Data Management	Knowledge Graph	Identity & Access Management
Walmart	Itaú	hp	cisco	AirBnb	UBS

Neo4j图数据库应用领域广泛，然而以下哪些领域并**不适合**使用图数据库？

A. 社交网络分析

B. 在线推荐引擎

C. 欺诈团伙检测

D. 财务报表

E. 身份管理和访问控制

F. 媒体数据管理和存储

G. 知识图谱

H. 数据地图/元数据管理

I. 网路和设备管理

技术应用场景

Neo4j Technology Use Cases*

- Integration with Hadoop (and other No-SQL databases)
- Containerization / Docker images of Neo4j
- Cloud deployment: AWS and Azure
- Natural Language Processing
- Integration with ElasticSearch
- Large scale and complex calculation: Spark + Neo4j
- Artificial Intelligence: R + Neo4j
- Internet of Things
- Machine Learning
- … … … …

** Demo available on request*

Neo4j的社区

Largest pool of graph technologists

Downloads

10M+

3M+ from Neo4j Distribution
7M+ from Docker

Events

400+

Approximate Number of
Neo4j Events per Year

Trained Developers

50k+

Trained/certified Neo4j
professionals

Meetups

50k+

Number of Meetup
Members Globally

- Neo4j中文社区：neo4j.com.cn
- 中国图数据库技术群
- 中国用户和技术聚会

Finding Help

- Website: http://neo4j.com
- Twitter: https://twitter.com/neo4j
- Slack: search for neo4j
- Technical questions： http://stackoverflow.com
- APOC on Github: https://github.com/neo4j-contrib/neo4j-apoc-procedures
- QQ Group： Neo4j Online Community / 547190638 (for Chinese users)

Contact us via email: apac@neo4j.com (Asia Pacific regions)

是时候活动一下筋骨、放松一下大脑了。。。

THANK YOU!