Neo4j - Monitoring Best Practices



neo4j

Agenda

- Neo4j Monitoring:
  - Monitor the Process
  - Monitor the Logs
  - Halin
  - Neo4j Metrics
  - Monitoring of JVM - disk utilization, CPU utilization and memory utilization
  - Query Management
  - Transaction Management
  - Connections Management
  - Monitoring of cluster endpoints

Monitor the Process

- Is the Process Running? (Red light/Green light)
- Is the http/https/bolt port reachable?
- Is the database reachable?

Neo4j logs

https://neo4j.com/docs/operations-manual/current/monitoring/logging/#logging

Log File	Description
neo4j.log	The standard log, where general information about Neo4j is written. Not written for Debian and RPM packages. *a.k.a. "user log".*
debug.log	Information useful when debugging problems with Neo4j.
http.log	Request log for the HTTP API.
gc.log	Garbage collection logging provided by the JVM.
query.log	Log of executed queries that takes longer than a specified threshold.
security.log	Log of security events.

Configuration

```
dbms.logs.<log>.enabled
               .rotation.delay
                        .keep_number
                        .size
               .level  (for security & debug)

dbms.logs.query.threshold
               .parameter_logging_enabled
               .time_logging_enabled
               .allocation_logging_enabled
               .page_logging_enabled
```

Monitor the Logs

- Parse for specific strings
- Integrate with log monitoring tools - Splunk, stackdriver etc..
  - [Configuring Stackdriver for Neo4j]

Alert Type	Alert Name	Description	Tool	Metric/log file	Comment
Neo4j security		Successful and unsuccessful login attempts	Splunk	security.log	index=neo4j* earliest=-1m@s source="/var/log/neo4j/security.log" "failed to log in" | rex "ERROR\s\[(?P<User_Name>[^\]]+)" | eventstats cou
		Password change by user, administration, or for users	Splunk	security.log	index=neo4j* source="/var/log/neo4j/security.log" "*changed password*"
			Splunk	security.log	index=neo4j* earliest=-1h@s source="/var/log/neo4j/security.log" "*tried to change password*" | rex "ERROR\s\[(?P<User_Name>[^\]]+)" | eventstats count by User_Name | search count > 3
		Users added and deleted, including failed attempts	Splunk	security.log	index=neo4j* source="/var/log/neo4j/security.log" "*created user*"
			Splunk	security.log	index=neo4j* source="/var/log/neo4j/security.log" "*deleted user*"
			Splunk	security.log	index=neo4j* source="/var/log/neo4j/security.log" "* tried to create user*"
		Custom roles added and deleted, including failed attempts	Splunk	security.log	index=neo4j* source="/var/log/neo4j/security.log" "*created role*"
			Splunk	security.log	index=neo4j* source="/var/log/neo4j/security.log" "*deleted role*"
			Splunk	security.log	index=neo4j* source="/var/log/neo4j/security.log" "*tried to create role*"
		Roles assigned and removed for users, including failed attempts	Splunk	security.log	index=neo4j* source="/var/log/neo4j/security.log" "*added role*"
			Splunk	security.log	index=neo4j* source="/var/log/neo4j/security.log" "*removed role*"
		Users activated and suspended	Splunk	security.log	index=neo4j* earliest=-1h@s source="/var/log/neo4j/security.log" "*suspended user*"
			Splunk	security.log	index=neo4j* source="/var/log/neo4j/security.log" "*activated user*"
			Splunk	security.log	index=neo4j* source="/var/log/neo4j/security.log" "*tried to activate user*"
			Splunk	security.log	index=neo4j* source="/var/log/neo4j/security.log" "*tried to suspend user*"
		Non-admin users failed attempts to view the list of users and roles.	Splunk	security.log	index=neo4j* earliest=-15m@s source="/var/log/neo4j/security.log" "*tried to list*"
Neo4j query		Queries that run more than 5 seconds (configurable) or longer to execu	Splunk	query.log	index=neo4j* source="/var/log/neo4j/query.log" "INFO" | rex "INFO\s\s+(?<elapsed>\d+)\sms" | where elapsed>5000
		Query that have large page faults	Splunk	query.log	index=neo4j* source="/var/log/neo4j/query.log" "*page faults*" NOT "*0 page faults*"
Neo4j debug		Cluster status and node elections data	Splunk	debug.log	index=neo4j* earliest=-15m@s source="/var/log/neo4j/debug.log" "Election started" OR "Moving to CANDIDATE"
		Monitoring CC endpoint status information	Browser		See https://neo4j.com/docs/operations-manual/3.2/monitoring/causal-cluster/#causal-clustering-http-endpoints
		Warning or errors for stack traces	Splunk	debug.log	index=neo4j* source="/var/log/neo4j/debug.log" CASE("*ERROR*") NOT "Fatal error occurred when handling a client connection:" NOT "Failed write response to driver Cannot write to buffer when closed"
			Splunk	debug.log	index=neo4j* source="/var/log/neo4j/debug.log" CASE("*WARN*") NOT "Fatal error occurred when handling a client connection:" NOT "Failed write response to driver Cannot write to buffer when closed" NOT "GC Monitor: Application threads blocked for" NOT "WARN [o.n.k.i.p.Proced Failed to load" NOT "WARN [o.n.c.r.ReadReplicaStartupProcess]" NOT "WARN [o.n.c.c.CatchupServer]"
		Memory error	Splunk	debug.log	index=neo4j* source="/var/log/neo4j/debug.log" "java.lang.OutOfMemoryError"
		Server startup/shutdown information	Splunk	debug.log	index=neo4j* source="/var/log/neo4j/debug.log" "Starting server"
			Splunk	debug.log	index=neo4j* source="/var/log/neo4j/debug.log" "Shutdown started"
		Transaction timeouts or commit issues	Splunk	debug.log	index=neo4j* source="/var/log/neo4j/debug.log" "*Transaction timeout*" or "*Failed to commit transaction*"
		Checking for long GC's over 7 seconds	Splunk	debug.log	index=neo4j* source="/var/log/neo4j/debug.log" "GC Monitor" | rex "GC Monitor:\sApplication\sthreads\sblocked\sfor\s+(?<elapsed>\d+)ms where elapsed>7000
		The data from the Read Replica Neo4j node which contains the "progress" and "success" status of the backup that happens every			

Essential log events

debug log

- “ERROR” messages

```
2020-07-05 20:07:10.245+0000  ERROR [o.n.c.p.h.HandshakeServerInitializer] Exception in ...
```

- Leader elections

```
2020-06-25 11:18:11.920+0000 INFO [o.n.c.c.c.RaftMachine]   Election started ...
```

- Restarts

```
2020-10-12 12:09:21.880+0000 INFO [o.n.i.d.DiagnosticsManager]                    [ System diagnostics ]
```

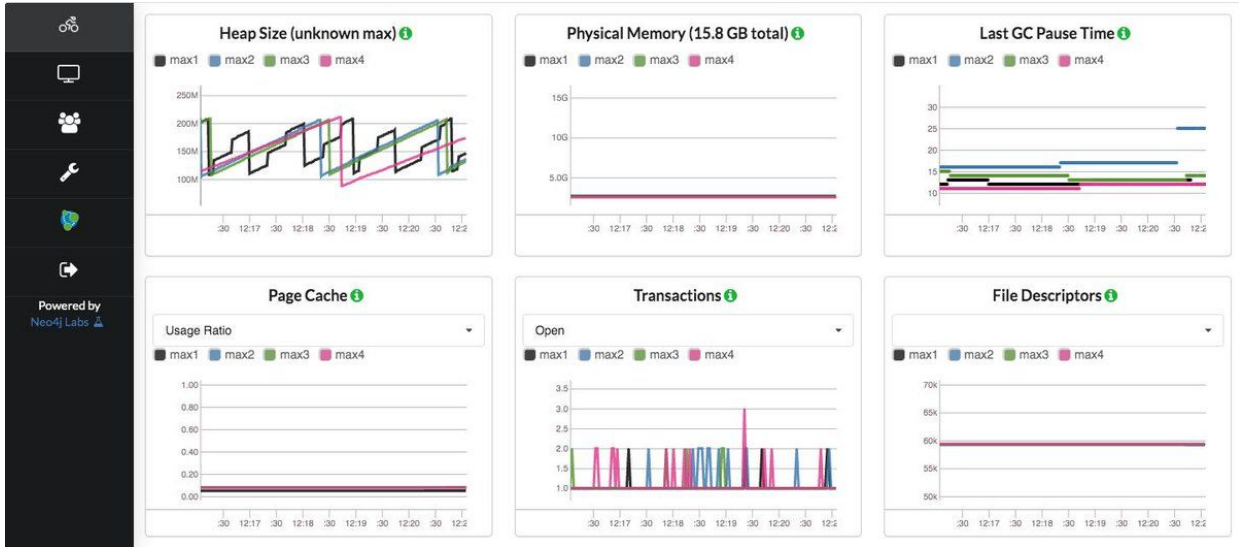
query log

- “ERROR” messages, long running queries (ex: >10s)
- memory usage, page cache hits...

neo4j

Halin

- https://neo4j.com/labs/halin/
- Halin is a cluster-enabled monitoring tool for Neo4j, that provides insights into live metrics, queries, configuration and more



Neo4j metrics

https://neo4j.com/docs/operations-manual/current/monitoring/metrics/

types

- Groups of metrics
- Global (per DBMS) `neo4j.<group>.<metric>`
- per database `neo4j.<db>.<group>.<metric>`

published as

- CSV
- Graphite
- Prometheus endpoint
- JMX

Configuration

```
metrics.enabled
metrics.neo4j.pagecache.enabled
metrics.bolt.messages.enabled
…
metrics.jvm.gc.enabled
```

```
metrics.csv.enabled=true
metrics.graphite.enabled=true
                        .server=graphitehost:2003
metrics.prometheus.enabled=true
                         .endpoint=localhost:2004
metrics.jmx.enabled=true
```


neo4j

Essential metrics

https://neo4j.com/docs/operations-manual/current/monitoring/metrics/reference/#metrics-reference

System (collectd) 

cpu usage, available memory

free disk space, open file handles

Neo4j metrics

memory	heap usage, direct memory usage, GC pauses
page cache	hit ratio, usage
transactions	active R/W Tx, rollbacks, peak concurrent
	checkpoint time
	cypher replanning events
data	ids in use
cluster	who is leader, last txID

neo4j

Sample Metrics to Monitor

Neo4j Global Metrics (graphite, prometheus, jmx or csv)

Metric Name	Metric	Description
Heap size (bytes) Heap max (bytes)	<prefix>.vm.heap.used <prefix>.vm.heap.max	This is the amount of heap space currently used at a given point in time. Monitor this to identify if you are maxing out consistently (In which case you should increase initial and max heap size) or if you are underutilizing and you can then decrease the initial and max heap sizes.
Total GC pause time (sec/min)	<prefix>.vm.gc.time.%s	Long GCs can be an indication of performance issues, or potentially instability. If this approaches the heartbeat timeout in a cluster, it may cause unwanted leader elections.
Page Cache Hit Ratio	<prefix>.page_cache.hit_ratio	Performance relies on efficiently using the page cache, so this metric should be in the 98-100% range consistently. If it is much lower than that, then the database is going to disk too often.
Page Cache Usage Ratio	<prefix>.page_cache.usage_ratio	This metric shows what percentage of the allocated page cache is actually being used. If it is 100%, then it is likely that the hitratio will start dropping, and you should consider allocating more RAM to page cache.
Open File Descriptors Max File Descriptors	<prefix>.vm.file.descriptors.count <prefix>.vm.file.descriptors.maximum	Neo4j recommends setting the ulimit to 40K file handles. This is because of the native and lucene indexing used. If this metric gets close to the upper ulimit, you should consider raising the ulimit.
Bolt Connections Running	<prefix>.bolt.connections_running	This provides the total number of bolt connections actively running on the instance. It helps to determine the load on the system
Page Cache Total Faults	<prefix>.page_cache.page_faults	The total number of page faults happened in the page cache. If this continues to rise over time, it could be an inidication that more pagecache is needed.

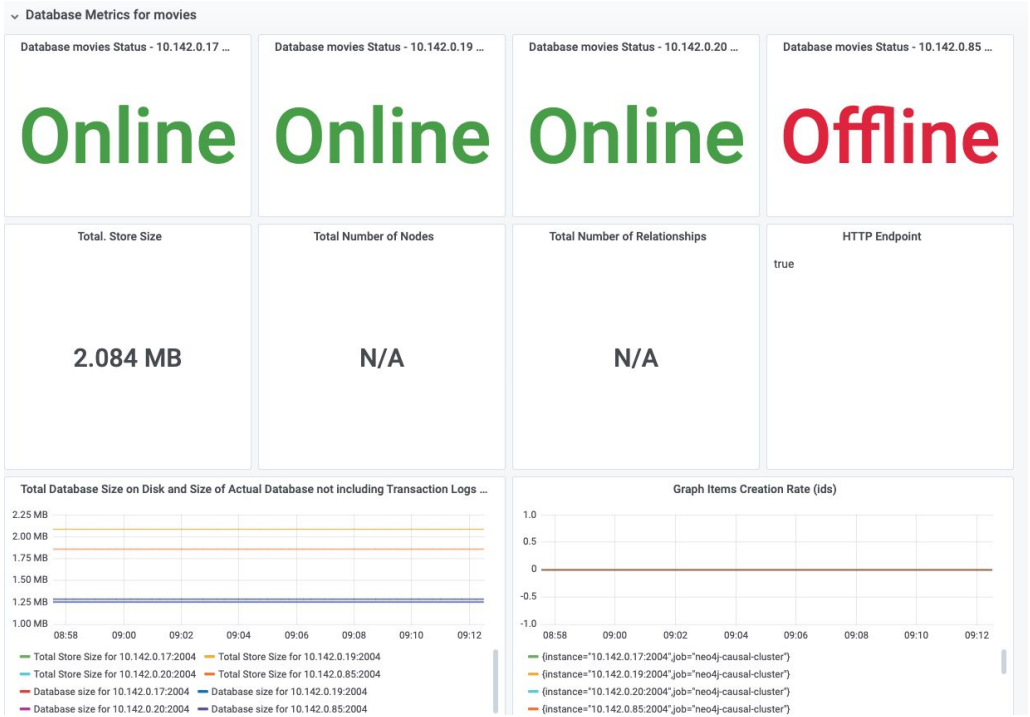
Neo4j Metrics per DB (graphite, prometheus, jmx or csv)

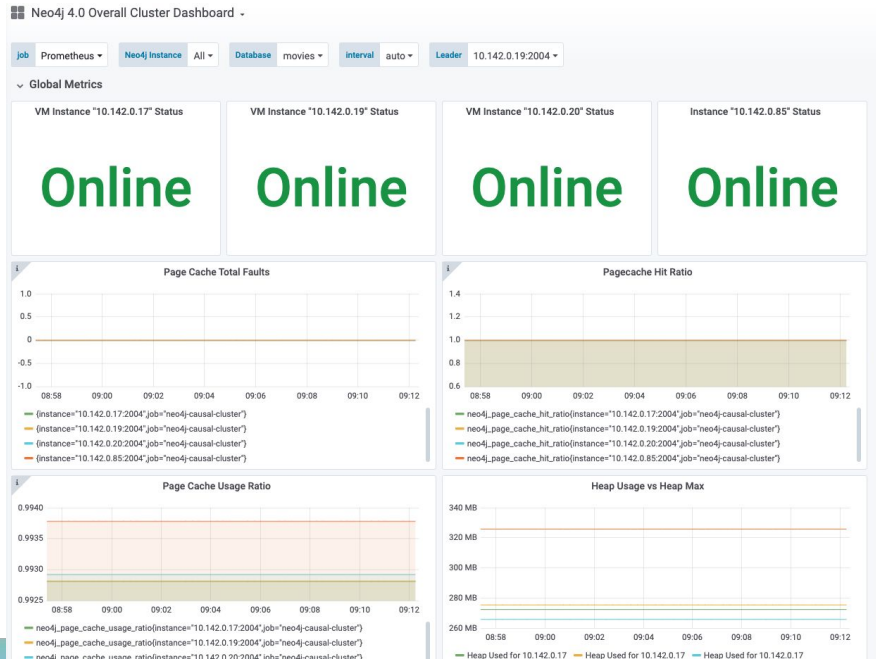
Metric Name	Metric	Description
Who thinks they are raft leader?	<prefix>.causal_clustering.core.is_leader	Track this for each Core cluster member. It will report 0 for "I am not leader" and 1 for "I am leader". The sum of all of these should always be 1. If it is 0 for more than 30 seconds, action may be needed.
Last committed tx id	<prefix>.transaction.last_committed_tx_id	Track this for each Core cluster member, and each Read Replica. Might break into separate charts. It should show one line, ever increasing, and if one of the lines levels off or falls behind, it is clear that this member is no longer replicating data and action is needed to rectify this situation.
Total IDs in Use	<prefix>.ids_in_use.relationship_type <prefix>.ids_in_use.property <prefix>.ids_in_use.relationship <prefix>.ids_in_use.node	This is informational, but good for identifying spikes or large increases in this, indicating large data loads, which could correspond with some behavior you are investigating. This is in itself not an indication of any issue.
Throughput (tx/sec)	<prefix>.transaction.committed <prefix>.transaction.committed_read <prefix>.transaction.committed_write	This is informational, but good for identifying spikes or large increases in this, indicating large data loads, which could correspond with some behavior you are investigating. This is in itself not an indication of any issue.
Active transactions	<prefix>.transaction.active <prefix>.transaction.active_read <prefix>.transaction.active_write	This is informational, but good for identifying spikes or large increases in this, indicating large data loads, which could correspond with some behavior you are investigating. This is in itself not an indication of any issue.
Peak Concurrent Transactions	<prefix>.transaction.peak_concurrent	The highest peak of concurrent transactions. This is a useful value to understand to help design for highest load scenarios and whether the bolt thread settings should be altered.
Rolled Back Write Transactions	<prefix>.transaction.rollbacks_write	Seeing a lot of writes rolled back may indicate various issues with locking, transaction timeouts etc..
Cypher re-planning events (per sec)	<prefix>.cypher.replan_events	Neo4j caches 1000 plans by default. Seeing sustained replanning events or large spikes could indicate an issue that needs to be investigated.
Check Pointing Duration (ms)	<prefix>.check_point.duration	Checkpoints should generally take several seconds to several minutes. Long checkpoints can be an issue, as these are invoked when the database stops, when a hot backup is taken, and periodically as well. Values over 30 minutes or so should be cause for some investigation.
Database Total Store Size	<prefix>.store.size.total <prefix>.store.size.database	The total size of the database and transaction logs, in bytes. The total store size of the databases helps determine how much page cache is required. It also helps compare the Total disk space used by the data store and how much is available.

Sample Metrics to Monitor

These are more Server level items that should also be monitored	
Metric Name	**Description**
Is Process Up?	Monitor the process and/or ports or endpoints to ensure instance is running
Free space by disk (bytes)	This might be tracked in another system. If not, use this to monitor free disk space and take action/alert when this is less than 10%.
CPU usage (jiffies/tx)	Indication of CPU utilization spikes, and if this is maxed out, additional CPU capacity may be needed. This metric mainly focuses on identifying spikes in high CPU queries.
Idle CPU (jiffies)	Indication of CPU utilization spikes, and if this is maxed out, additional CPU capacity may be needed. This metric mainly focuses on identifying when CPU is saturated in general.
Available memory	Because the OS and lucene, plus some other things, will need some amount of memory, make sure this metric shows consistenly available memory for those things, based on calculations done when sizing the system (i.e. leftover from Total Memory - max heap - page cache)
Used Memory	This metric will tell you if you are close to using all available memory on the server. It is good practice to make sure that your max used spikes are bit below 100%, something like 95% or lower, to reduce the risk of an OOM situation.
Disk I/O stats	When the Graph is larger then the Pagecache setting, the Disk I/O can have a large impact on the performance of the system.

Sample Grafana Dashboard





Monitoring - JVM

- Monitoring JVM
  - Recommend to monitor CPU, memory utilization, disk utilization, heap usage and Garbage Collection (gc)
  - Customers can use third party tools like DataDog, AppDynamics etc to monitor the JVM utilization

Query Logging

- Query Logging Best practice
  - Enable query logging
  - Default is to log all queries but it is recommended to log for queries exceeding a certain threshold.
  - Helps with analysis of long running queries and does not impact system performance
  - Logs
    - Allocated bytes for the executed query
    - Page hits and page faults for the executed query
    - Parameters for the executed query (though can be disabled to not display those in case of sensitive information)
    - Logs detailed timed information for executed query (CPU time, IO time etc.. - this parameter does impact performance so only recommended when required)

Query Management

- For Monitoring
  - list of Active Queries - Call dbms.listQueries()
  - list of Active Transactions using - Call dbms.listTransactions()
  - list of Active Connections - Call dbms.listConnections()
  - List of Active locks - call dbms.listActiveLocks(queryId)
- For Terminating
  - A single query - call dbms.killQuery(queryId)
  - Multiple queries - call dbms.killQueries(<list of queryIds>)
  - Single Transaction - call dbms.killTransaction(transactionId)
  - Multiple transactions - call dbms.killTransactions(<list of transaction ids>)
  - Single Connection - call dbms.killConnection(connectionId)
  - Multiple Connections - call dbms.killConnections(list of connection ids)

Transaction Management

- Transaction Management Best Practice
  - Configure Neo4j to terminate transactions whose execution time has exceeded the configured timeout.
    - *dbms.transaction.timeout=10s*
    - *Can set this dynamically on each instance (Read Replicas only if required)*
      - Call dbms.setConfigValue('dbms.transaction.timeout','10s')
  - An executing transaction may get stuck while waiting for some lock to be released by another transaction. To kill that transaction and remove the lock set
    - dbms.lock.acquisition.timeout=10s
    - Cannot be set dynamically
  - Best practice is to use both the `dbms.transaction.timeout` and `dbms.lock.acquistion.timeout`

Causal Cluster Monitoring - Using Stored Procedure

- Find the Role of cluster member
  - *CALL dbms.cluster.role()*
- Gain an overview over the instances in the cluster
  - *CALL dbms.cluster.overview()*
- Get Routing Recommendations
  - *CALL dbms.cluster.routing.getServers()*

Causal Cluster Monitoring - Using APIs

A unified set of endpoints exist, both on Core Servers and on Read Replicas, with the following behavior:

- /db/{dname}/cluster/writable — Used to direct write traffic to specific instances.
- /db/{dname}/cluster/read-only — Used to direct read traffic to specific instances.
- /db/{dname}/cluster/available — Available for the general case of directing arbitrary request types to instances that are available for processing read transactions.
- /db/{dname}/cluster/status — Gives a detailed description of this instance's view of its own status within the cluster. Useful for monitoring and coordinating rolling upgrades.
- Note - If authentication and authorization is enabled in Neo4j, the Causal Clustering status endpoints will also require authentication credentials. For some load balancers and proxy servers, providing this with the request is not an option.

  `dbms.security.causal_clustering_status_auth_enabled=false`