



Performance Benchmarking for Link Prediction Algorithms in Social Networks

Team Members :

Neeraj Kavassery Parakkat

Sumon Biswas

Prachi Patel



Outline



- Link Prediction
- Existing Work
- Overview of Metrics
- Project Goal
- Datasets Used
- Implementation and Schema
- Experimental Setup and Challenges Faced
- Results and Conclusions
- References



Link Prediction

Why link prediction?

- Used in social networks analysis.
- Helps detect communities in a large graph.
- Similar nodes tend to have similar properties and are likely to get connected.

How is similarity measured?

- Based on metrics.
- The type of metric is defined by the type of similarity we want to measure.
- Vary from each other on complexity of computation.



Existing Work

Motivation and Reference paper:

- **“Implementing Link-Prediction for Social Networks in a Database System”**, *DBSocial '13* New York, NY USA. Authors: Sara Cohen, Netanel Cohen-Tzemach
- The paper focuses on implementing seven link metrics on relational, graph and Key-value based databases and comparing each of them.
- Relational : MySQL 5.5 , Graph; Neo4J, Key-value Store: Redis.
- Goal: To identify which metric suits best for what kind of a database. 6 out of the 7 metrics were benchmarked in the project.

An overview of the metrics

- **Common Neighbors:** $\text{score}(x,y) = |\Gamma(x) \cap \Gamma(y)|$

Intuition: The probability of authors collaborating increases with the number of other collaborators they have in common.

- **Jaccard Similarity:** $\text{score}(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|}$

Intuition: May be they have common neighbors just because each may have lot of neighbors, not because they are strongly related to each others.

- **Preferential Attachment:** $\text{score}(x,y) = |\Gamma(x)| \cdot |\Gamma(y)|$

Intuition: The probability of co-authorship of x and y is correlated with the product of the number of collaborators of x and y.

An overview of the metrics

- **Graph Distance** : Shortest-path distance between two nodes

- **Katz measure:**
$$score(x, y) = \sum_{l=1}^{\infty} \beta^l \cdot |paths_{x,y}^{<l>}|$$

where $paths_{x,y}^{<\ell>} := \{\text{paths of length exactly } \ell \text{ from } x \text{ to } y\}$

weighted: $paths_{x,y}^{<1>} := \text{number of collaborations between } x, y.$

unweighted: $paths_{x,y}^{<1>} := 1 \text{ iff } x \text{ and } y \text{ collaborate.}$

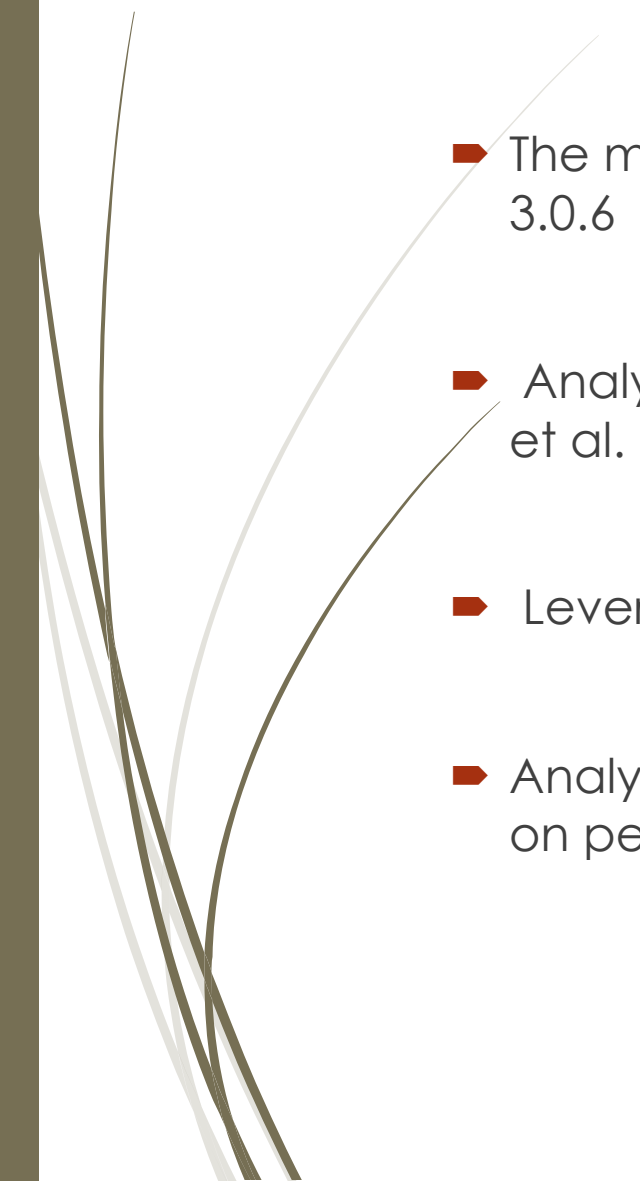
- **Rooted Page Rank:** Computes a general importance value for a node

$$PR(A) = (1-d) + d(PR(T1)/C(T1) + \dots + PR(Tn)/C(Tn))$$

damping factor, $d = 0.85$



Goal of the project

- The main objective of the project is to re-benchmark the metrics with latest neo4j v 3.0.6 and compare its performance with MySQL 5.6.
 - Analyze and implement how changes to the graph structure initially used by Cohen et al. improves performance.
 - Leverage latest features of neo4j like attribute indexes and cost based optimizer.
 - Analyze the change in query syntax between neo4j versions and record its impact on performance.
- 

Datasets Used

- All graphs contain only one type of node (author) and one type of edge (co-author)
- Datasets used : Coauthorship datasets from DBLP, and friendship network from Facebook

Type	Name	Nodes	Edges
Co-authorship	dblp-all-core3	366,600	4,349,796
Co-authorship	dblp-2010-2012	248,695	2,589,320
Co-authorship	dblp-2002-2009	182,493	1,621,846
Co-authorship	ca-CondMat	23,133	186,936
Co-authorship	ca-AstroPh	18,771	396,160
Co-authorship	ca-HepPh	12,006	237,010
Co-authorship	ca-HepTh	9,875	51,971
Co-authorship	ca-GrQc	5,241	28,980
Social Network	Facebook	4,039	170,174

Implementation in MySQL

- Version 5.6 / Cache: 128 MB
- The graph representation in MySql is exactly the **same as used by Cohen et al.**
- Edges are represented by edge tables, with 2 columns representing each node.
- Link prediction metrics were implemented as stored procedures, in order to avoid computation costs.
- Following helper tables were created to help speed up computation :
 - Neighbors Count Table : Stores no. of neighbors per node
 - Common Neighbors Table : Has no. of common neighbors for all possible node pairs.
 - Popular Nodes Table : Contains top n nodes with largest no. of neighbors

MySQL Schema

- The following is the schema in MySql
- id : author id

Table Name	Description	Schema
cn	Common Neighbors	(id1 int, id2 int, id3 int)
cnc	Common Neighbors Count	(id1 int, id2 int, count int)
edges	Co-authorship relation amongst authors	(id1 int id2 int)
neighbors	Neighbor count for every author	(id int, neighbours int)
nodes	Author Details	(id int, name varchar(64))
topn	Top 100 Nodes with highest neighbors count	(id int, neighbors int)

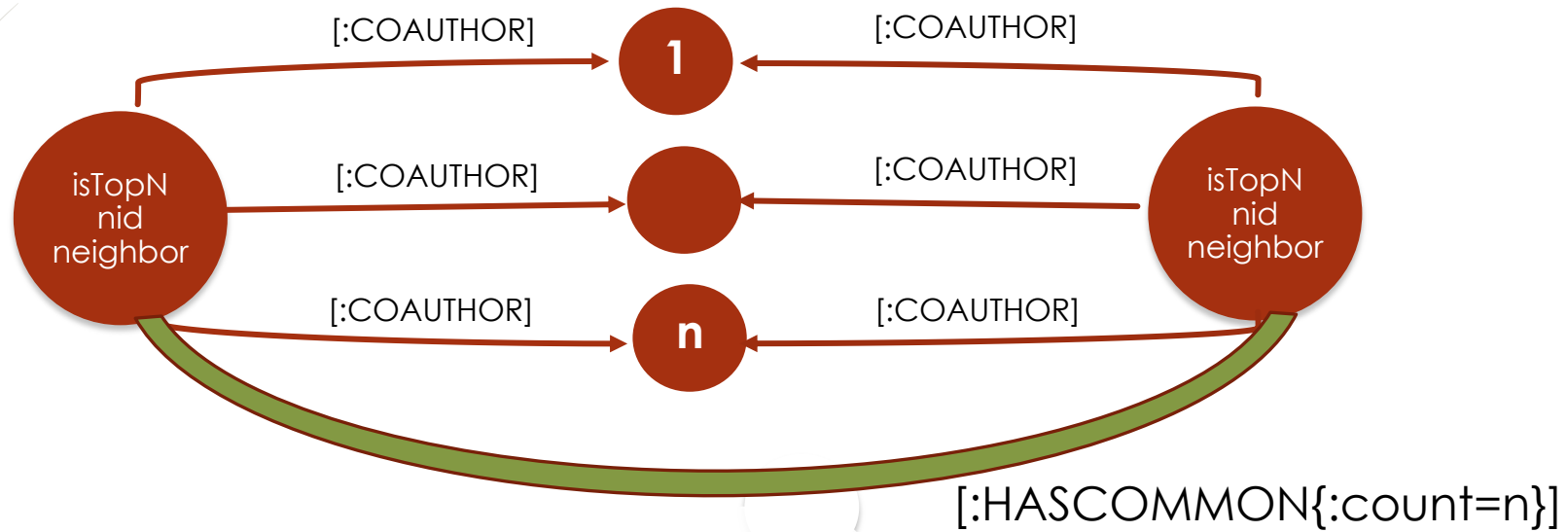
Schema in Neo4J

- Version 3.0.6 / Page cache: 5GB
- Nodes: Author nodes
- Edges: CoAuthorship



- Attributes
 - Nid: unique id of the node
 - Neighbors: neighbor count of the node
 - isTopN: is the node among the nodes with top 100 most neighbors.

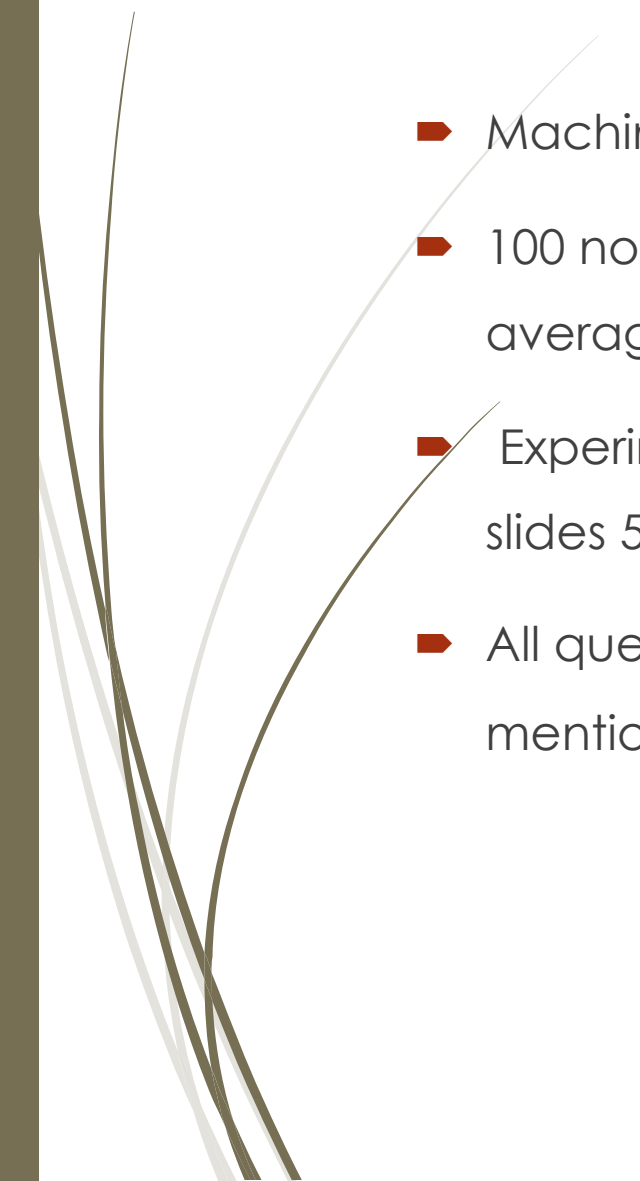
Changes to Schema in Neo4J



- Indexes Introduced:
 - Index on isTopN, neighbors
- HasCommon edge introduced to ease computing of common neighbors.
- Query changes:
 - Converted 'start' based queries to match.
 - Forced use of index.
 - Rewrote queries so as to leverage the new edge introduced.



Experimental Setup

- Machine : Core i7, 2.5Ghz, 16GB RAM
 - 100 nodes were randomly chosen, and each metric was computed 100 times and their average and total time were measured.
 - Experiment was run on all the 9 datasets listed in slide 8 and on all 6 metrics listed in slides 5 and 6.
 - All queries : MySQL - old queries and Neo4J - old and new both, were run on the above mentioned configuration
- 



Challenges Faced

➤ Neo4J :

- Version migration of existing queries used for running the metrics.
- Broken APIs due to new version of neo2py library.
- Collection of graphs used by the earlier experiments and importing them into the new version of neo4j.

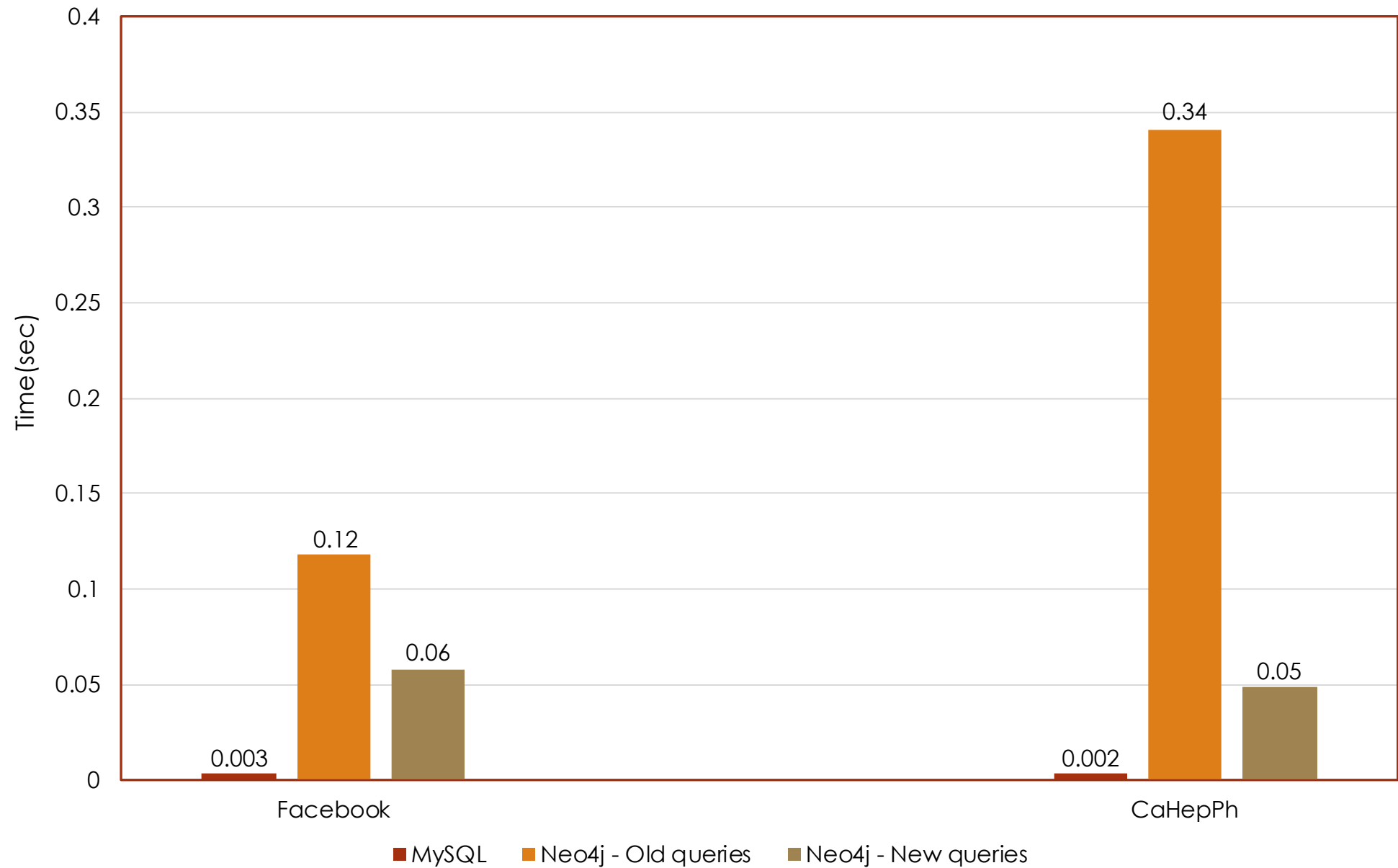
➤ MySQL :

- MySQL operations become slow for large datasets like dblp-2010- 2012 having hundreds of millions of rows.
- Iterative MySQL procedures often result into crash.

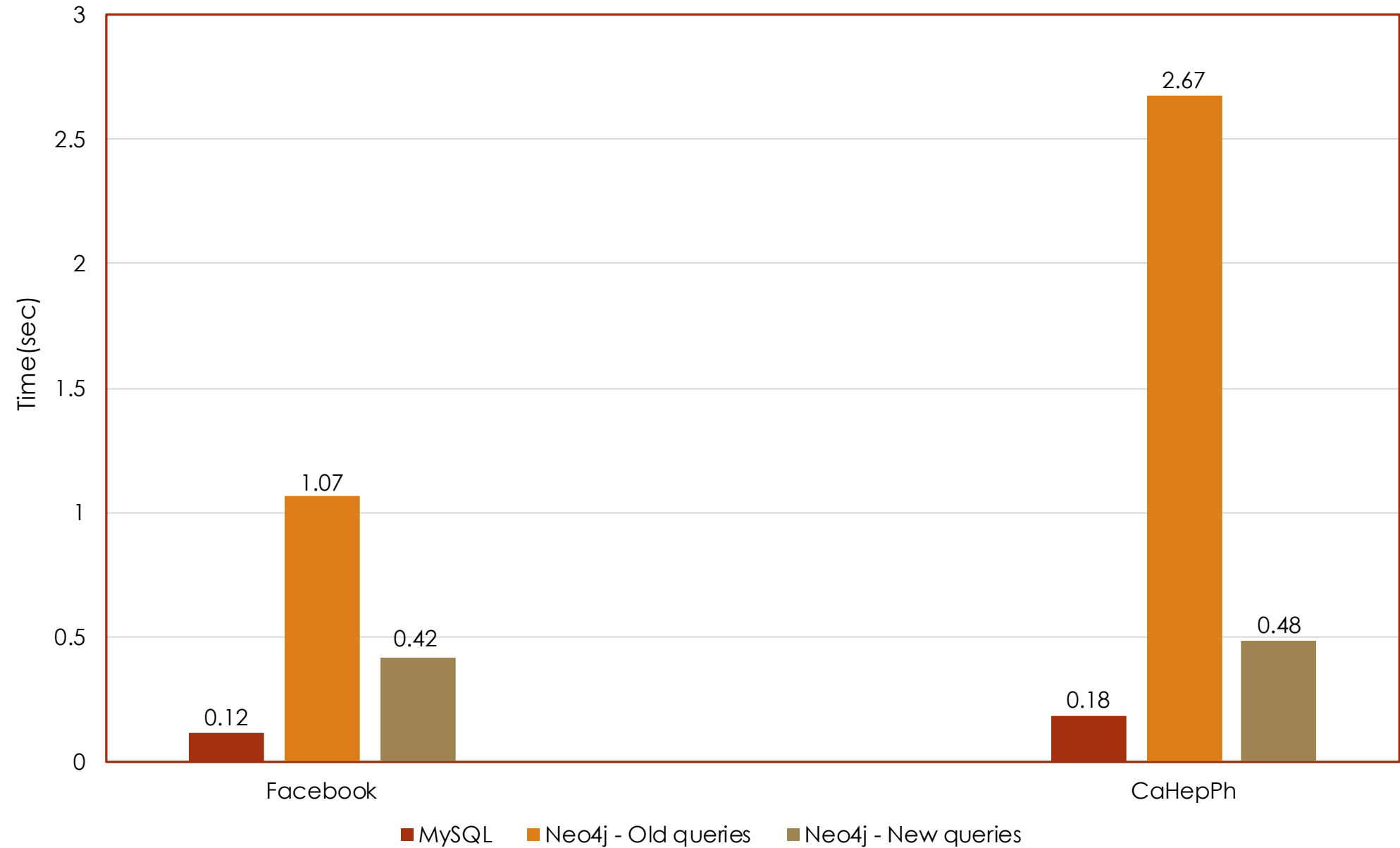


Experimental Results and Analysis

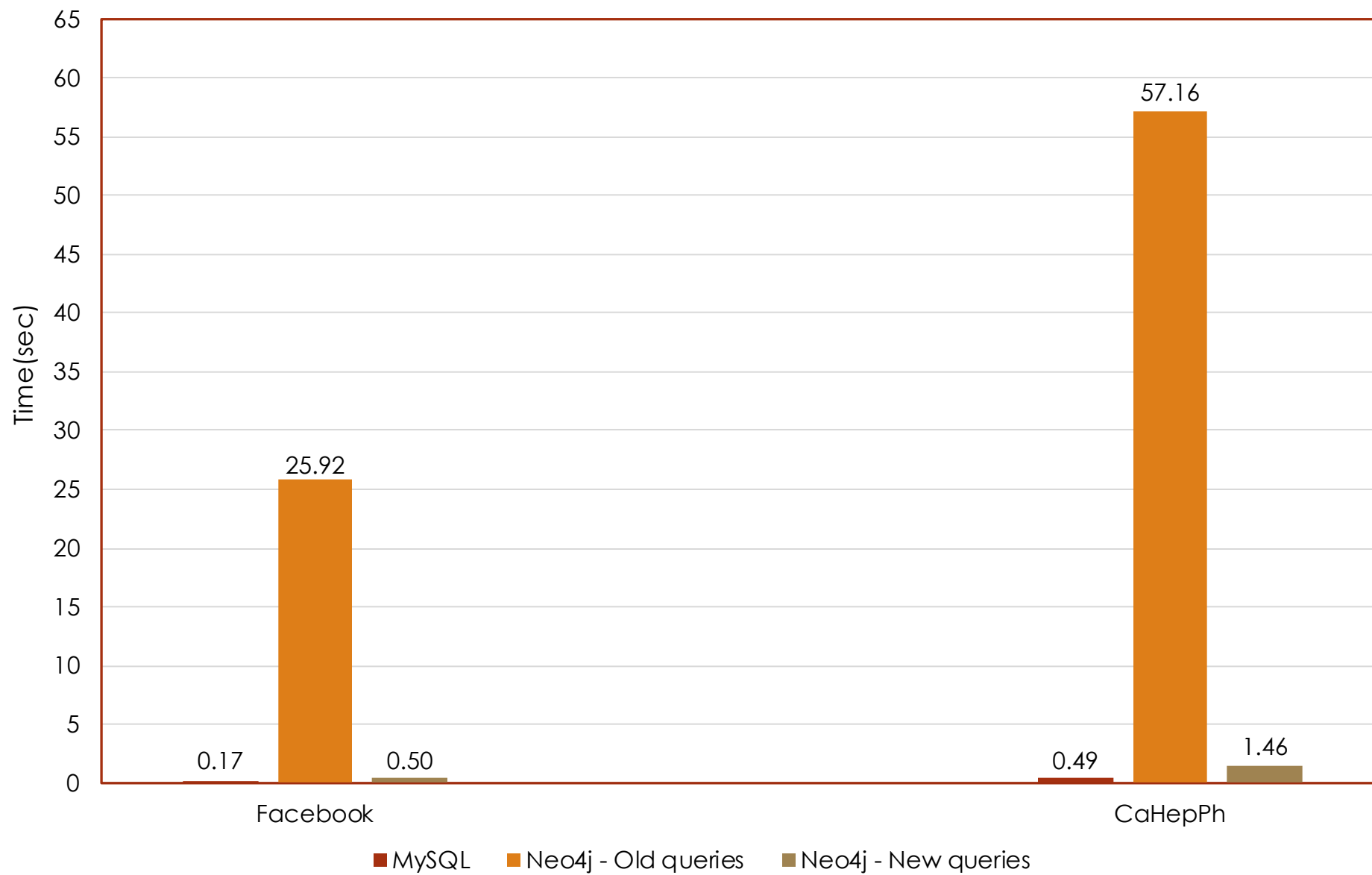
bPreferentialAttachment



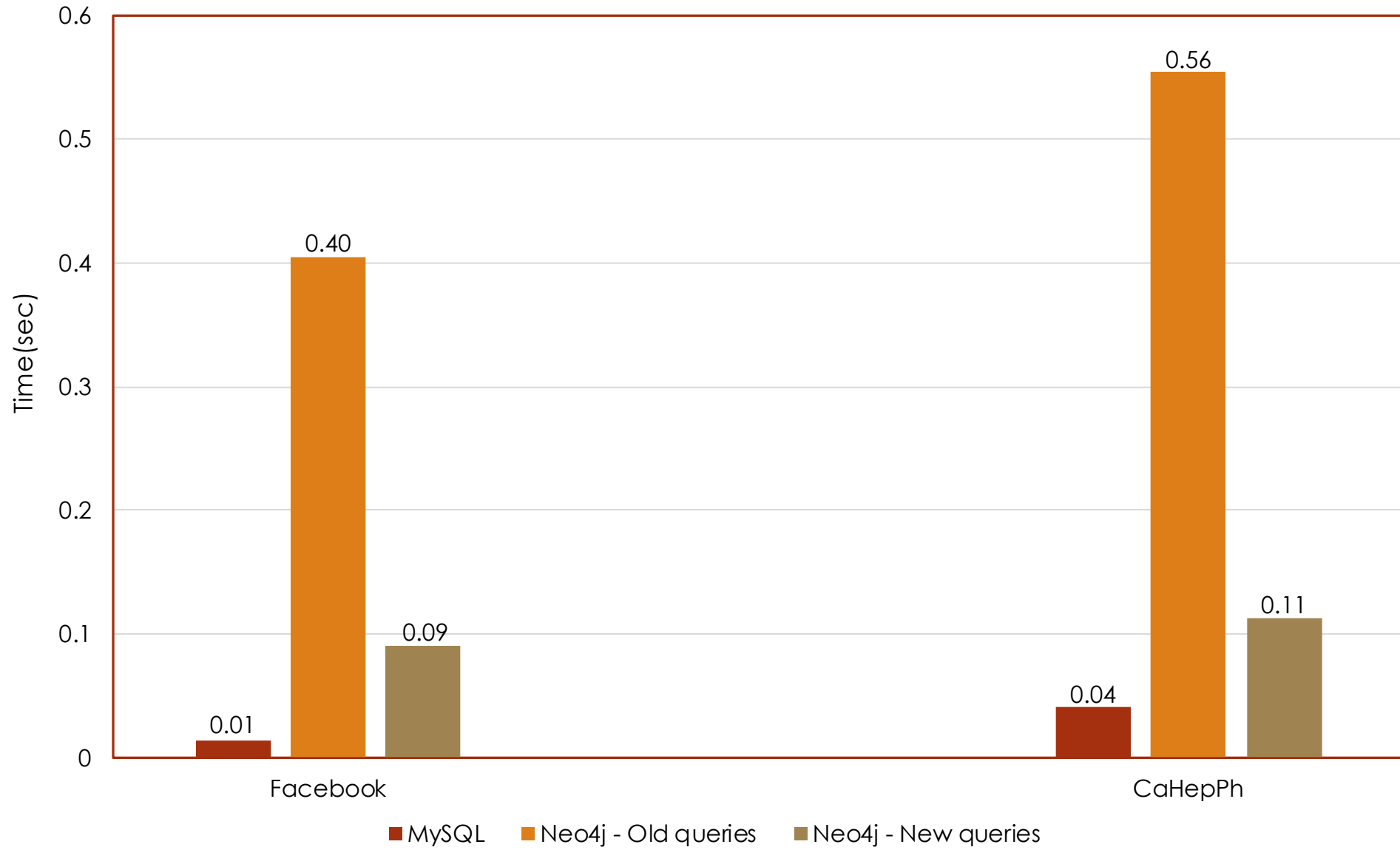
cNeighboursIndex



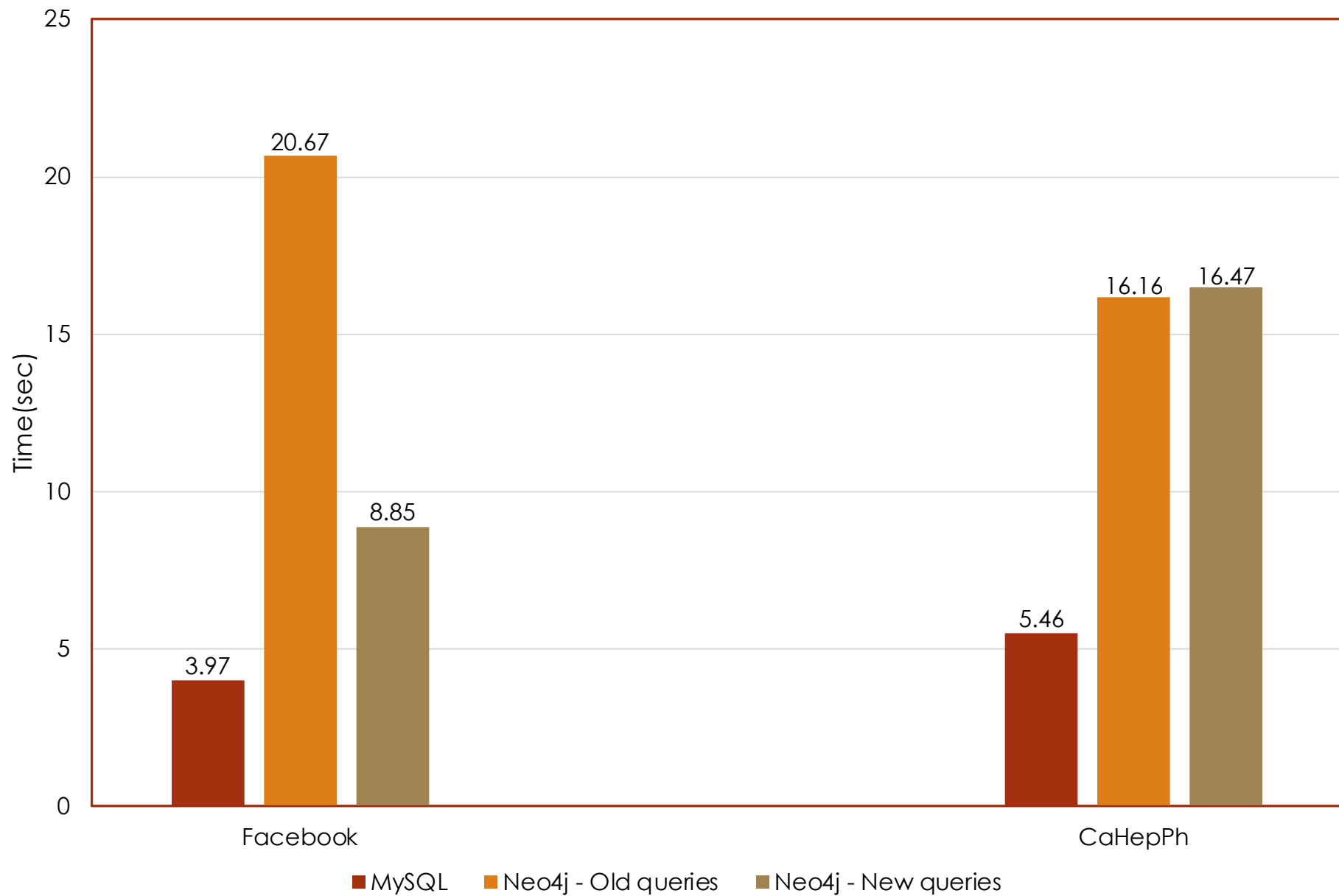
bCommonNeighbours



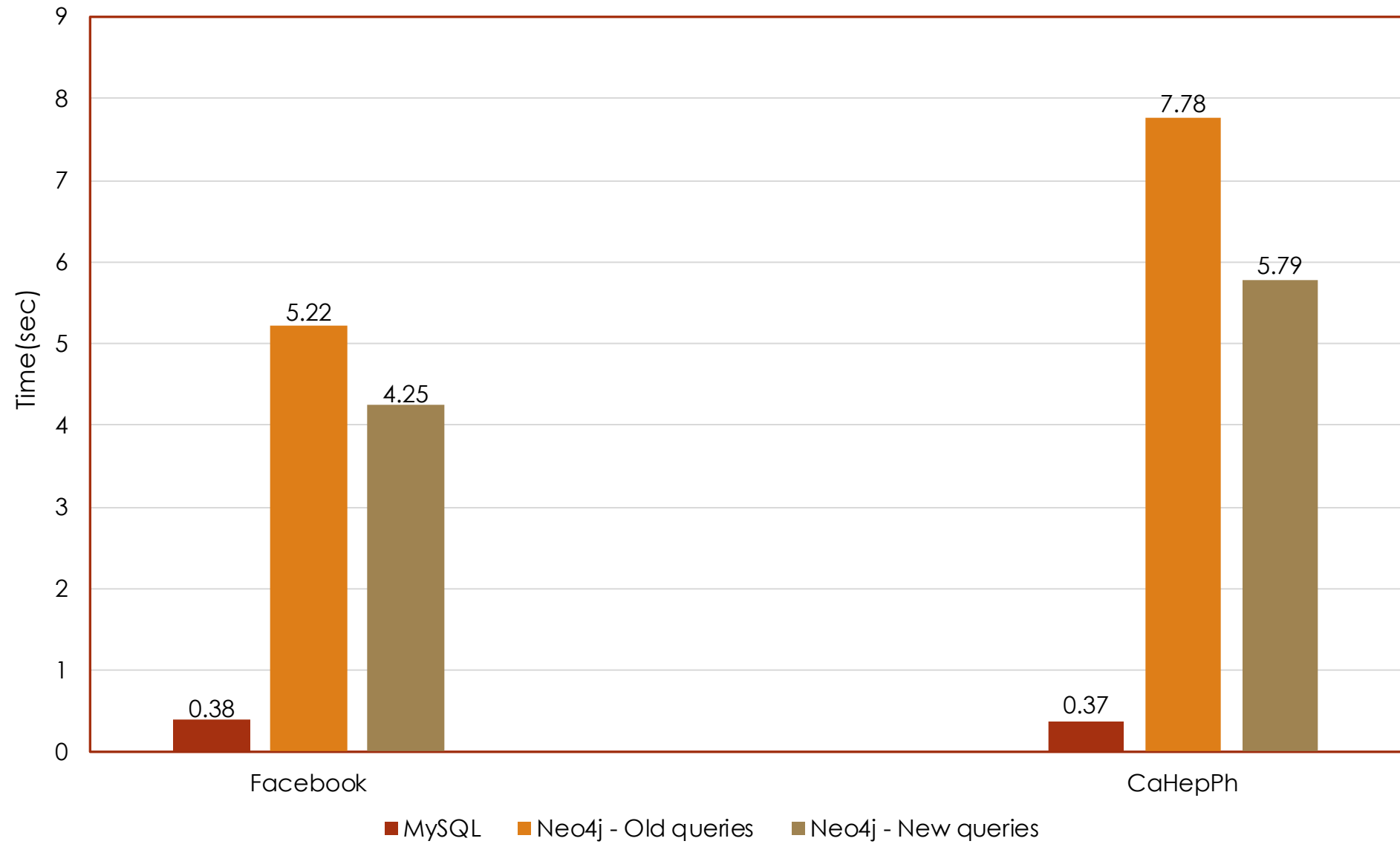
cTopIndex



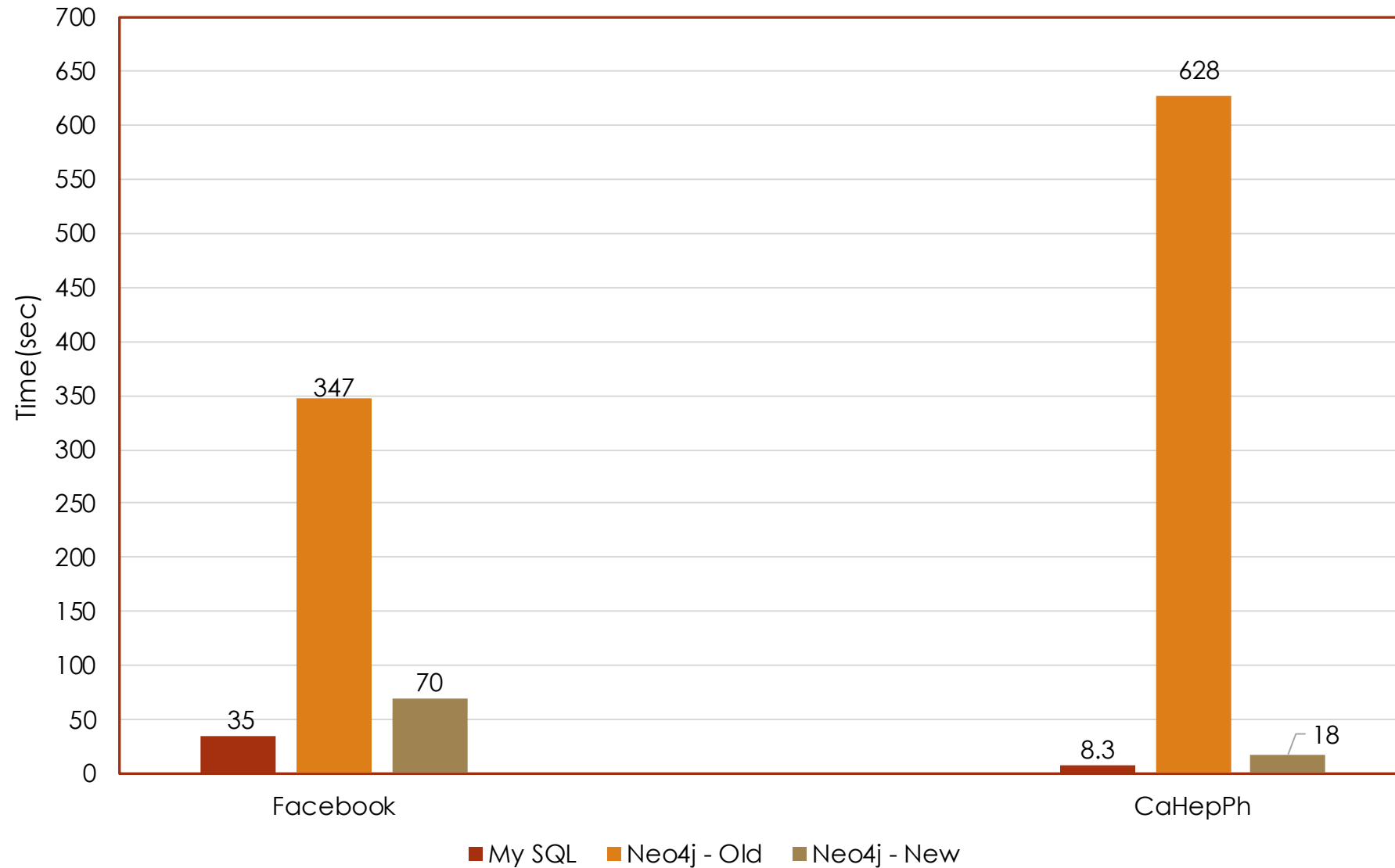
xJaccardsCoefficient



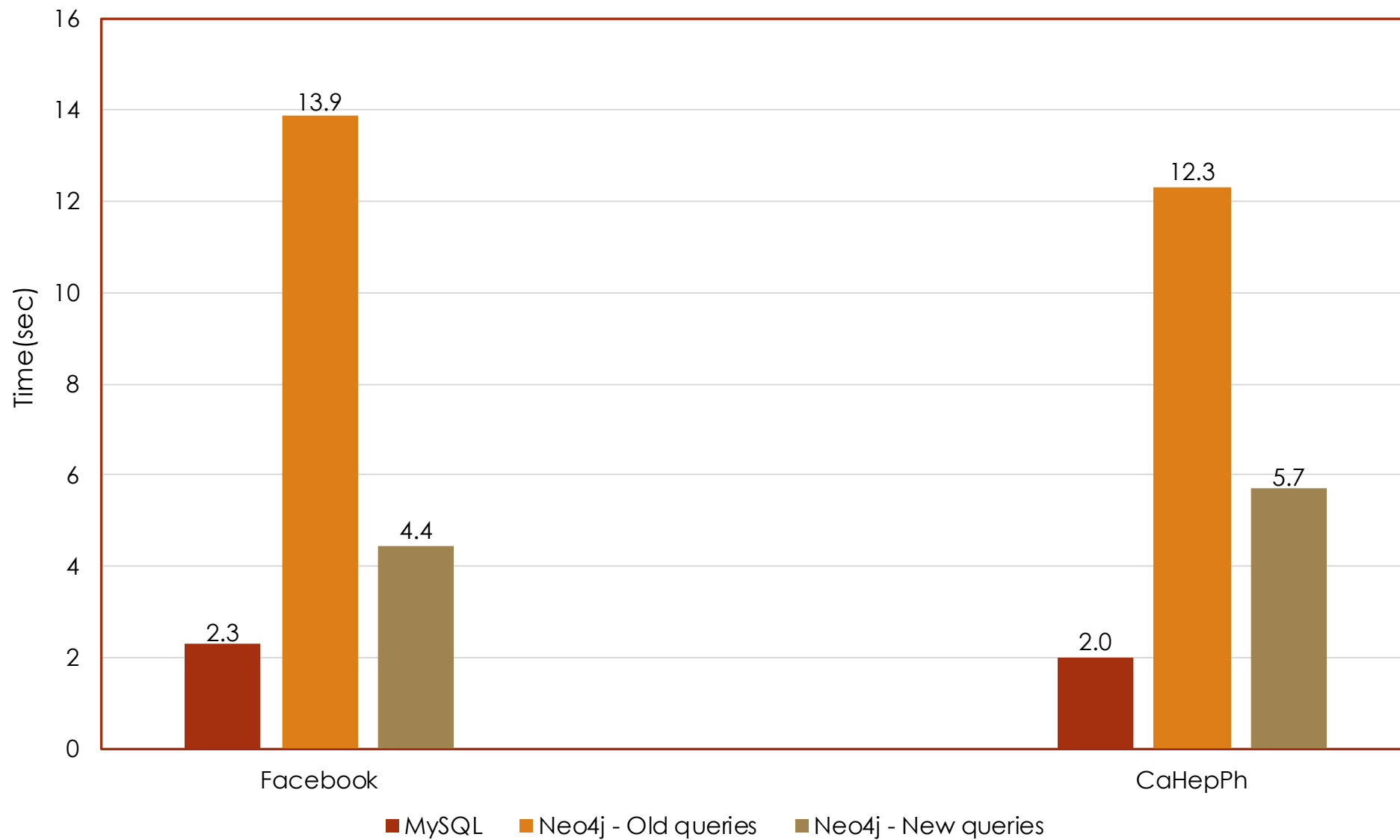
xPreferentialAttachment



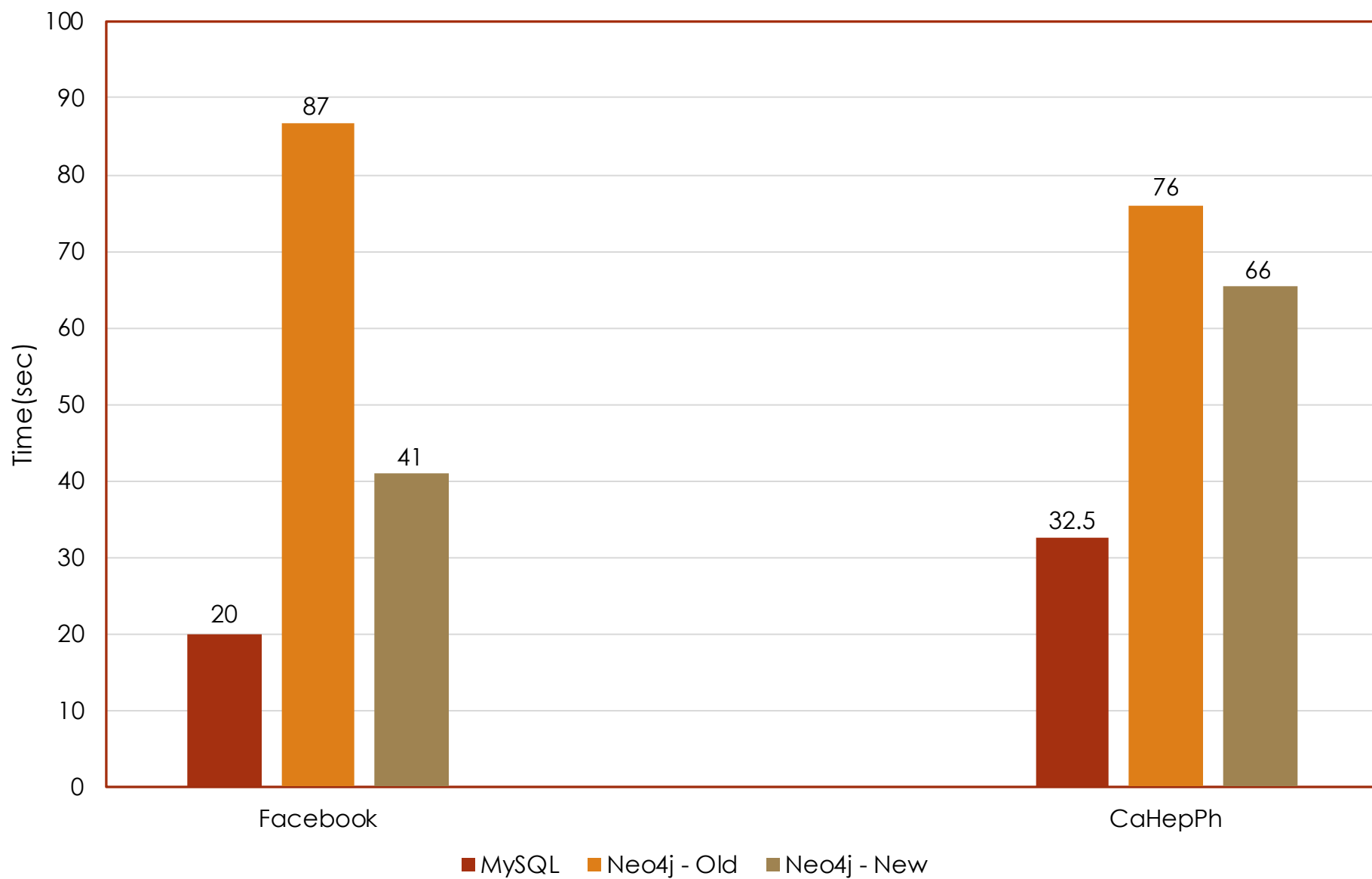
xKatz



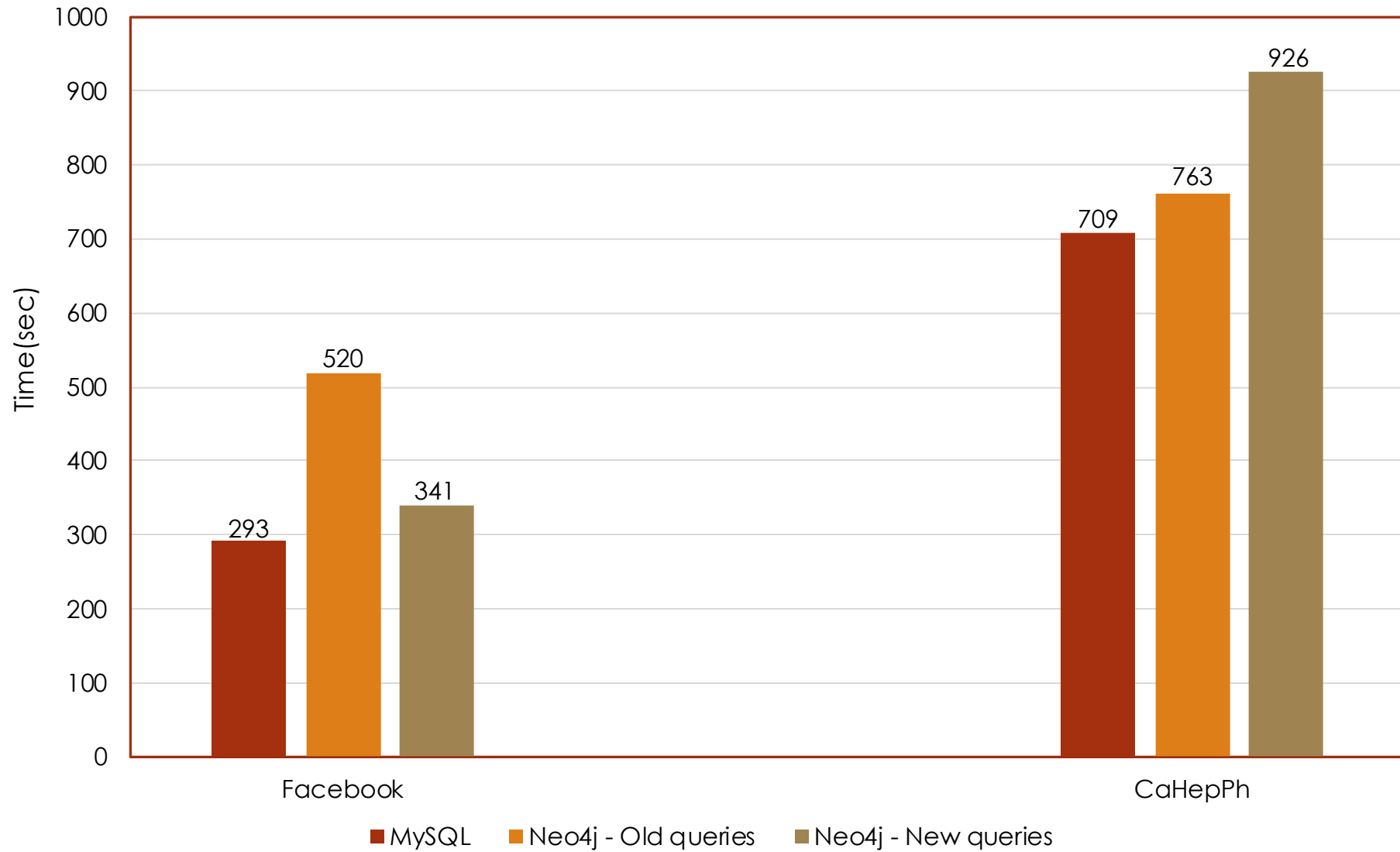
xCommonNeighbors



xGraphDistance



xRootedPageRank





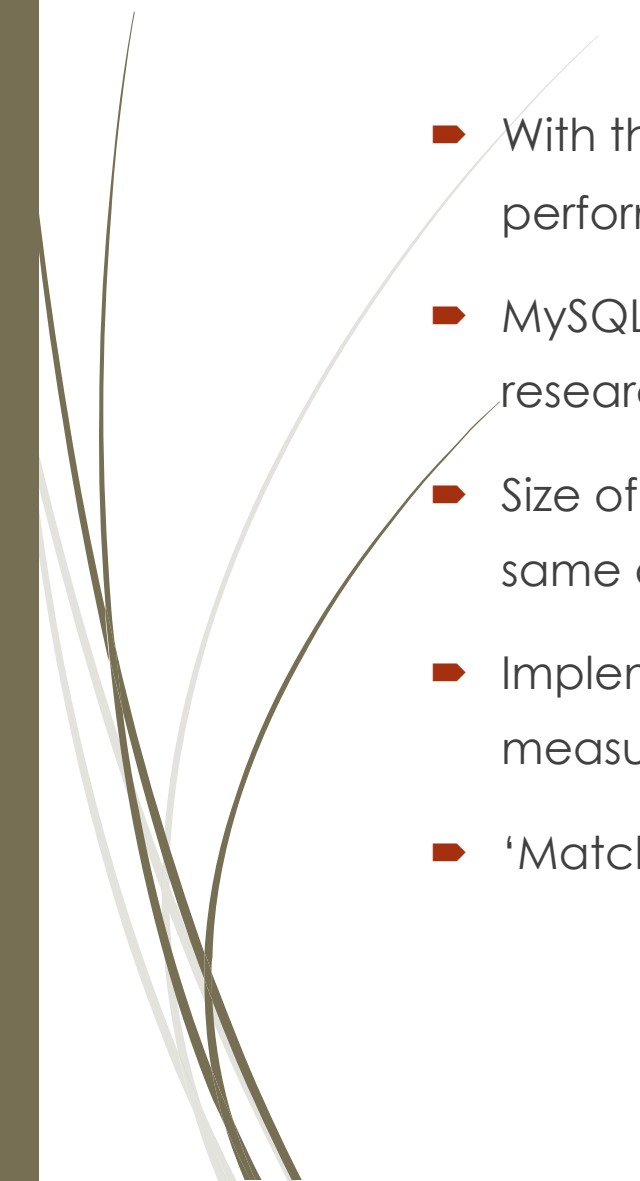
Analysis of the Graphs



- ▶ Queries not using the new edges but only indexing also showed large performance improvement due to indexing used on neighbors attribute:
 - Queries for : Katz measure, Preferential Attachment, Jaccard's Coefficient, TopNIndex, etc saw higher performance as compared to older queries
- ▶ Queries using the new edge performs nearly as better as mysql
 - For eg : bCommonNeighbors – execution time for MySQL : 0.17 and for Neo4J : 0.50
- ▶ For distance-based metrics, Neo4j works better on dense graphs
 - For eg : Graph Distance – for Facebook {dense graph} : we can see an improvement of around 46 secs, whereas for CaHepPh {sparse graph} : we just have an improvement of 10 secs.



Conclusion

- With the changes in the neo4j schema, we were able to closely match the performance of MySql for few of the metrics.
 - MySQL still outperforms because of the optimizations on joins that are a result of the research carried over for the past many years.
 - Size of neo4j databases are atleast 4 times more than MySql databases for the same datasets.
 - Implementation of edge based metrics (rootedPageRank, xDistance, katz measure) is easier in graph databases when compared to relational.
 - 'Match' based queries perform better than 'start' based queries.
- 



References

- **“Implementing Link-Prediction for Social Networks in a Database System”** - DBSocial 2013 Proceedings of the ACM SIGMOD Workshop on Databases and Social Networks . Pages 37-42 Authors: Sara Cohen, Netanel Cohen-Tzemach
- **“Link Prediction and Recommendation across Heterogeneous Social Networks”** - : Data Mining (ICDM), 2012 IEEE 12th International Conference. Pages: 181 – 190 Authors: Yuxiao Dong, Jie Tang, Sen Wu, Jilei Tian, Nitesh V. Chawla, Jinghai Rao, Huanhuan Cao
- **“A multilayer approach to multiplexity and link prediction in online geo-social networks”** - EPJ Data Science (2016) Authors: Desislava Hristova , Anastasios Noulas , Chloe Brown , Mirco Musolesi, Cecilia Mascolo