A neo4j powered social networking and Question & Answer application to enhance scientific communication.

René Pickhardt, Heinrich Hartmann

related-work.net

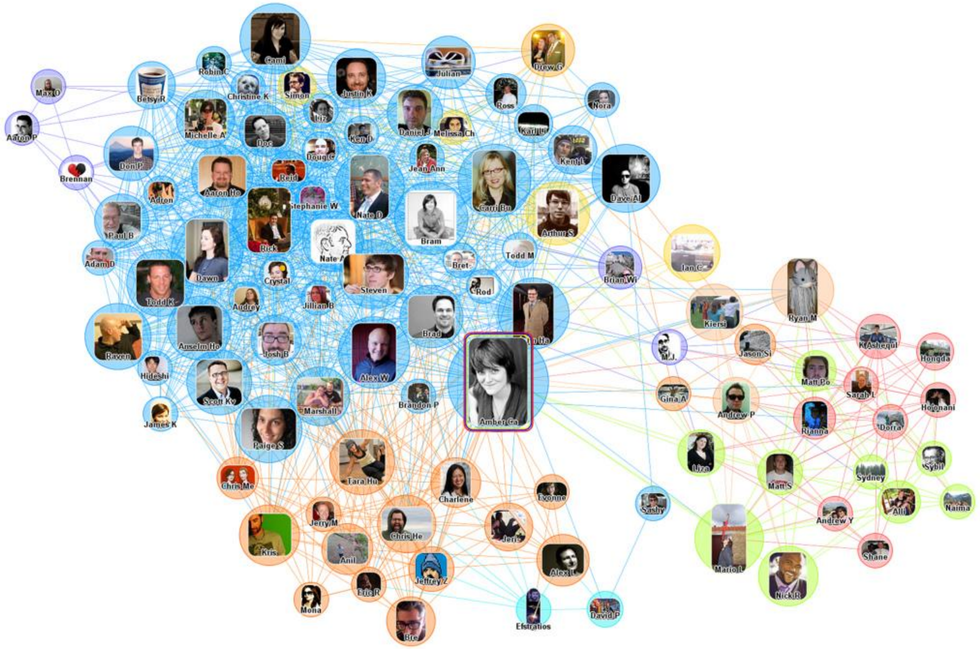
Roadmap

- Introduction
- Data structures for Q & A Systems
- Benchmarking Q & A Systems
- Our data model and requirements
- Lessons learnt from handling graphs in neo4j
- GWT(P)
- Demo Time
- (Personalized) Auto Completion

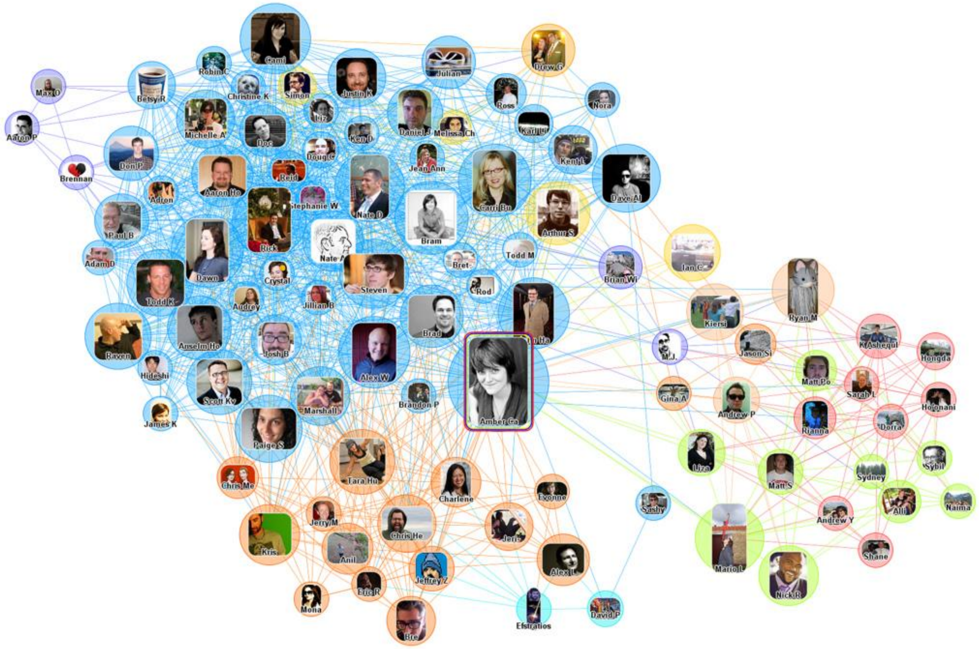
Central problems in Academia

1. Finding new relevant publications

2. Connecting people interested in the same topic



Central problems in Academia



Solution: The academic graph

- Social contacts
- Citation between papers

No Open Access to academic data: Citations/Fulltext!

We need open Access in the scientific community

Our society pays

1. scientists to do the research
2. scientists to do the peer reviewing
3. Conference fees (not here (:)
4. subscription fees for scientific journals (through libraries)

==> Open access could give us the citation graph.

Tim Berners Lee - 2012 at WWW conference:

"Think of how you want the world to be.

Just imagine it !

and then:

Go geek an code it up"

No open platform for scientists exists on the web

	Open Source	Open Data	User Generated Content
Google Scholar	NO	NO	Not really
Microsoft Academic	NO	NO	Not really
SciVerse/Elsevier	NO	NO	NO
CiterSeerX	YES	YES (Quality problematic)	NO
Mendeley	NO	Not really API (no citation data)	YES
ResearchGate	NO	NO	YES
Sciweavers	NO	NO	YES
RelatedWork	YES	YES	YES

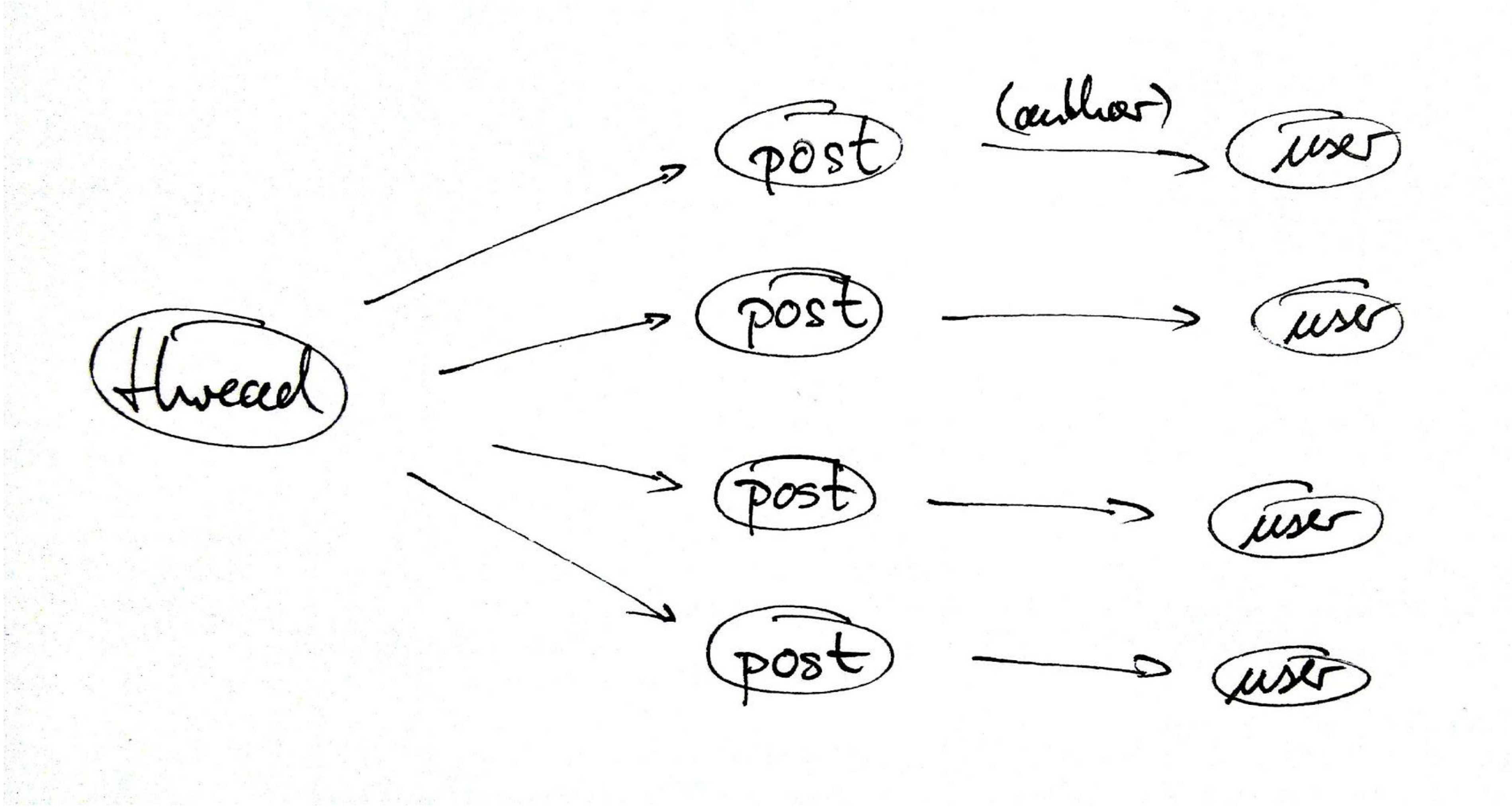
Our vision on blog.related-work.net/proposal

- Social network for scientists
- Q&A System to enhance scientific communication
- Open Access (to all publications)
- in particular: Open Citation Network
- Linked Open Data of discussions and meta data
- Strong recommenders
- More Trust Altmetrics
- Personalized news Feed for hot publications (last year)

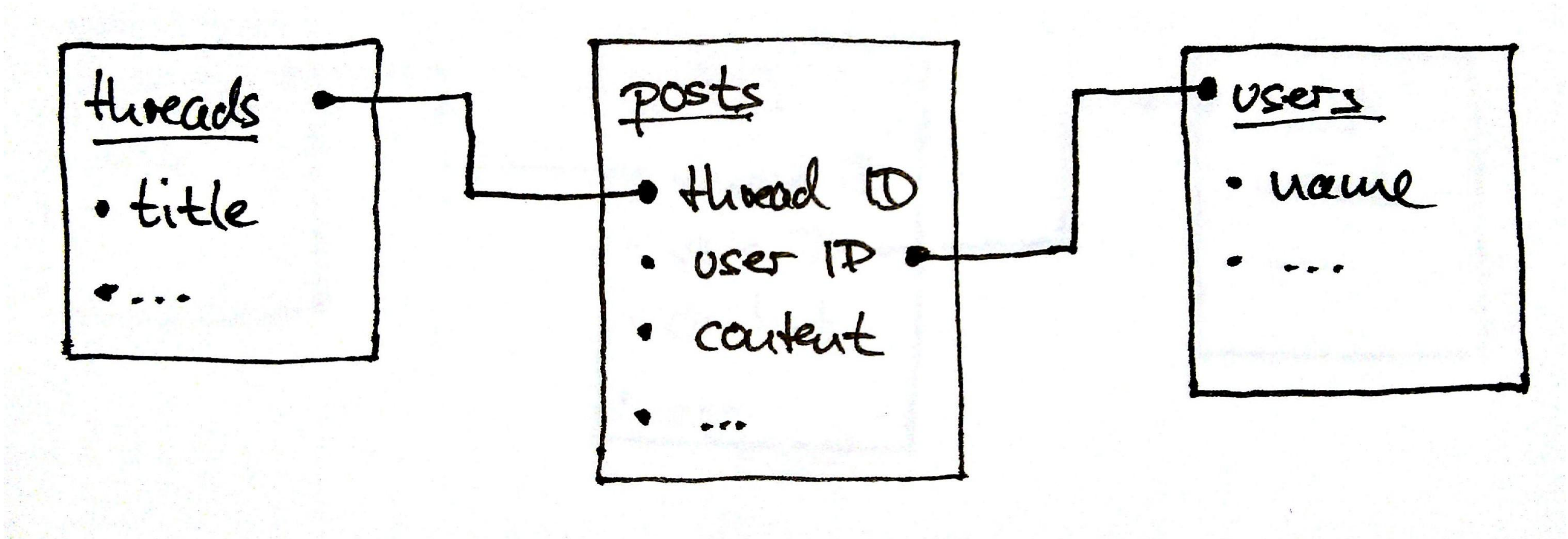
Roadmap

- Introduction
- Data structures for Q & A Systems
- Benchmarking Q & A Systems
- Our data model and requirements
- Lessons learnt from handling graphs in neo4j
- GWT(P)
- Demo Time
- (Personalized) Auto Completion

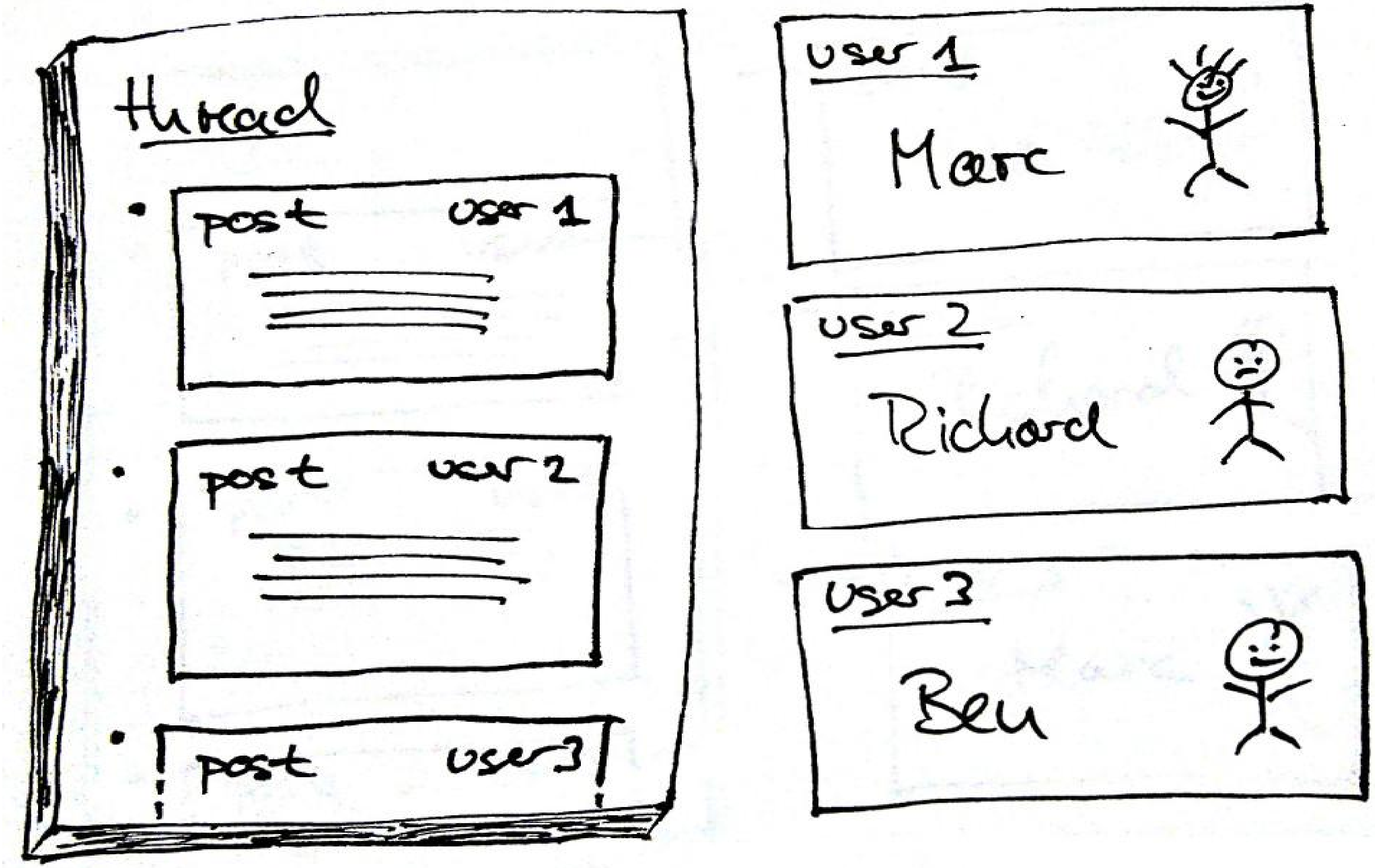
A graph model for discussions



Relational model for discussions



Discussions in document stores

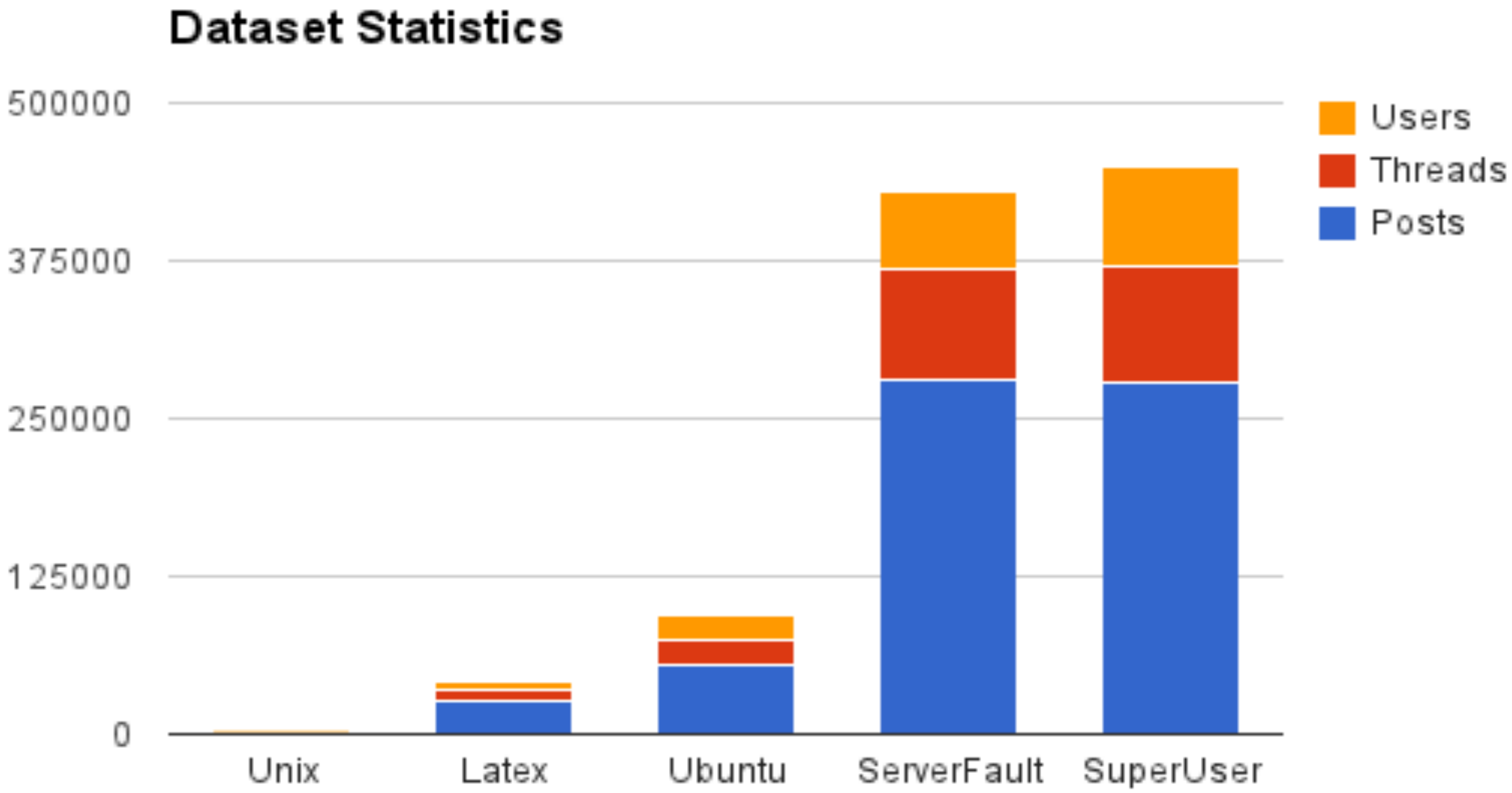


Roadmap

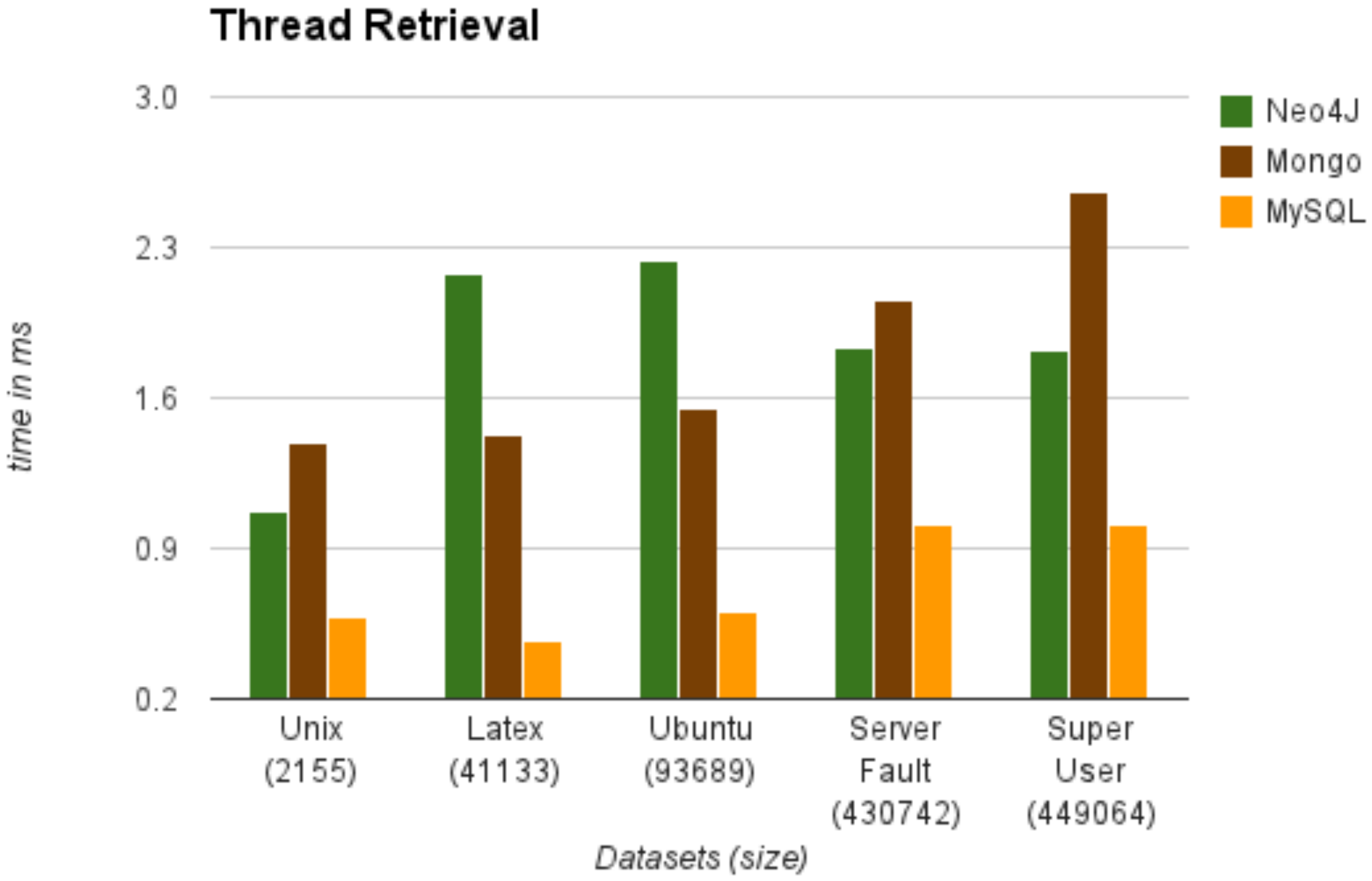
- Introduction
- Data structures for Q & A Systems
- Benchmarking Q & A Systems
- Our data model and requirements
- Lessons learnt from handling graphs in neo4j
- GWT(P)
- Demo Time
- (Personalized) Auto Completion

Datasets from StackExchange

StackExchange -- **OPEN DATA!**



Graph dbs are competitive

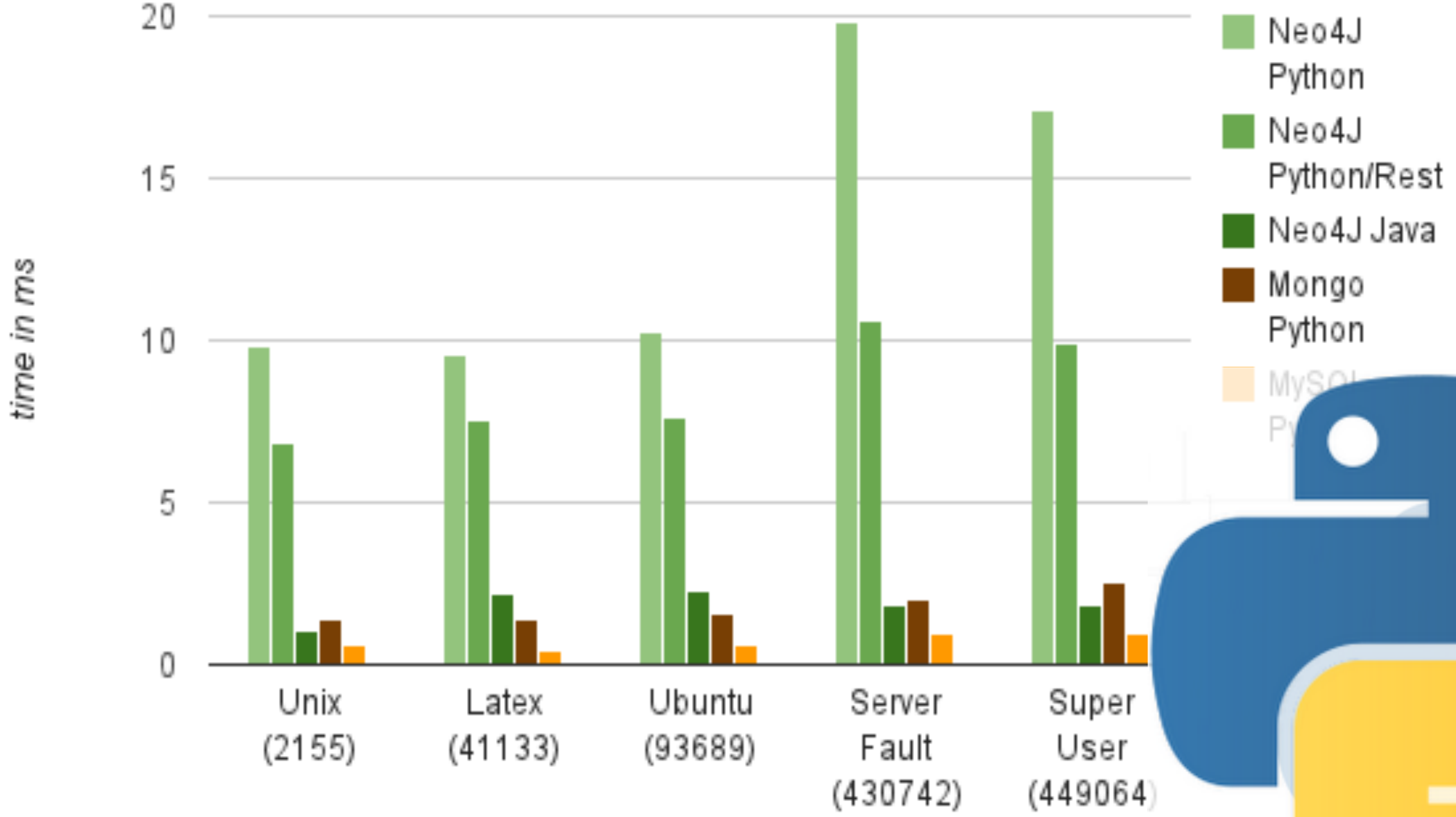


Who likes Python?

Who likes Neo4J?

Neo4J/Python performance???

Neo4j
The World's Leading Graph Database



Neo4J/Python -- WTF??

Neo4j
The World's Leading Graph Database

By marysia_@flickr

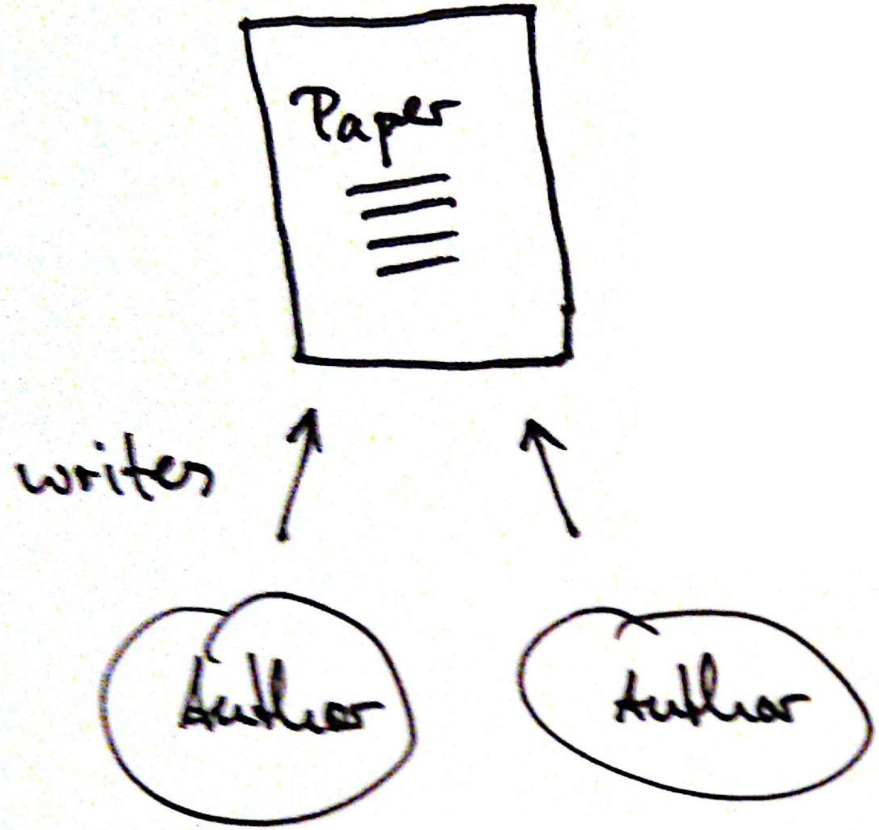
Neo4j
The World's Leading Graph Database

Please give us a fast API!!

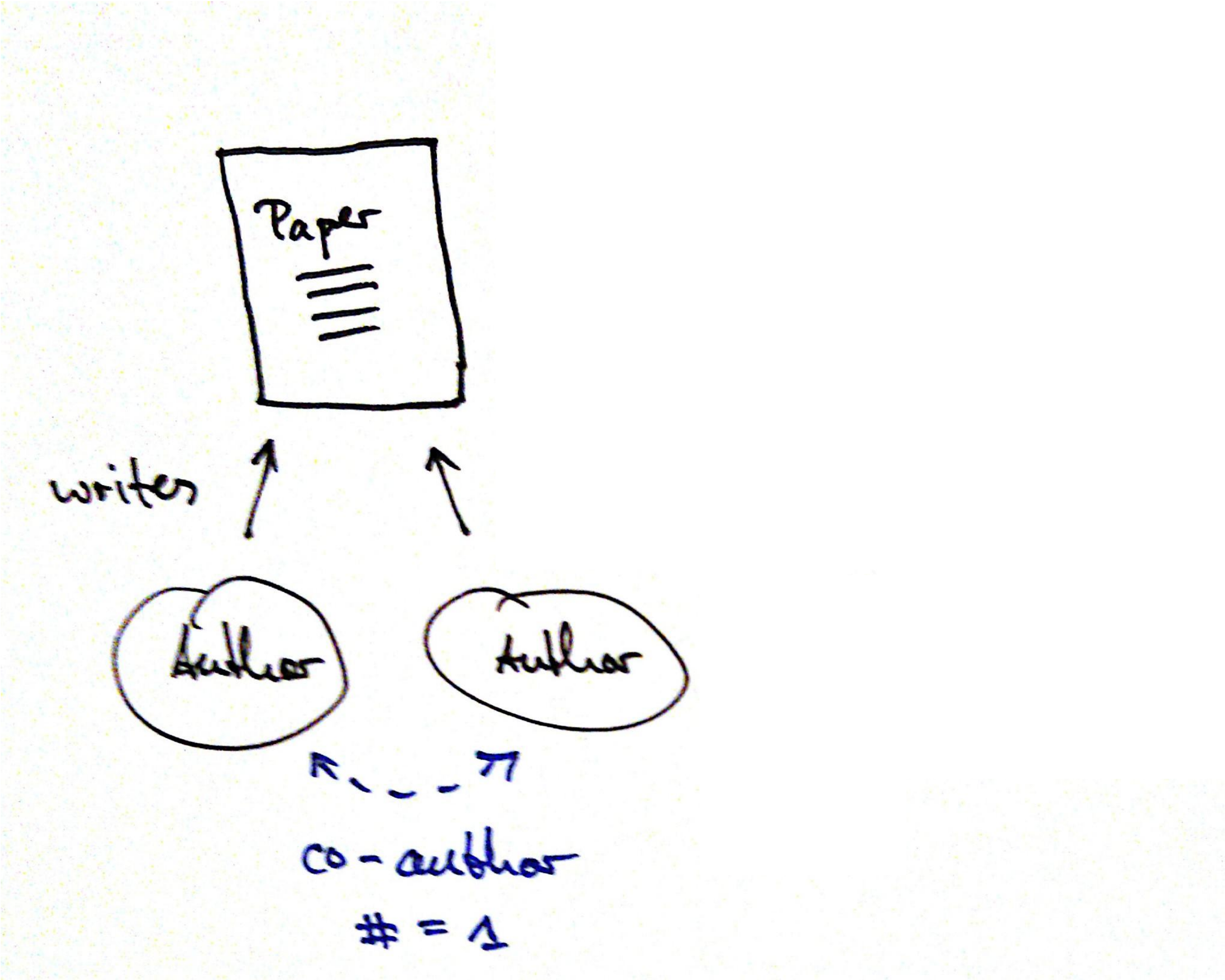
Roadmap

- Introduction
- Data structures for Q & A Systems
- Benchmarking Q & A Systems
- Our data model and requirements
- Lessons learnt from handling graphs in neo4j
- GWT(P)
- Demo Time
- (Personalized) Auto Completion

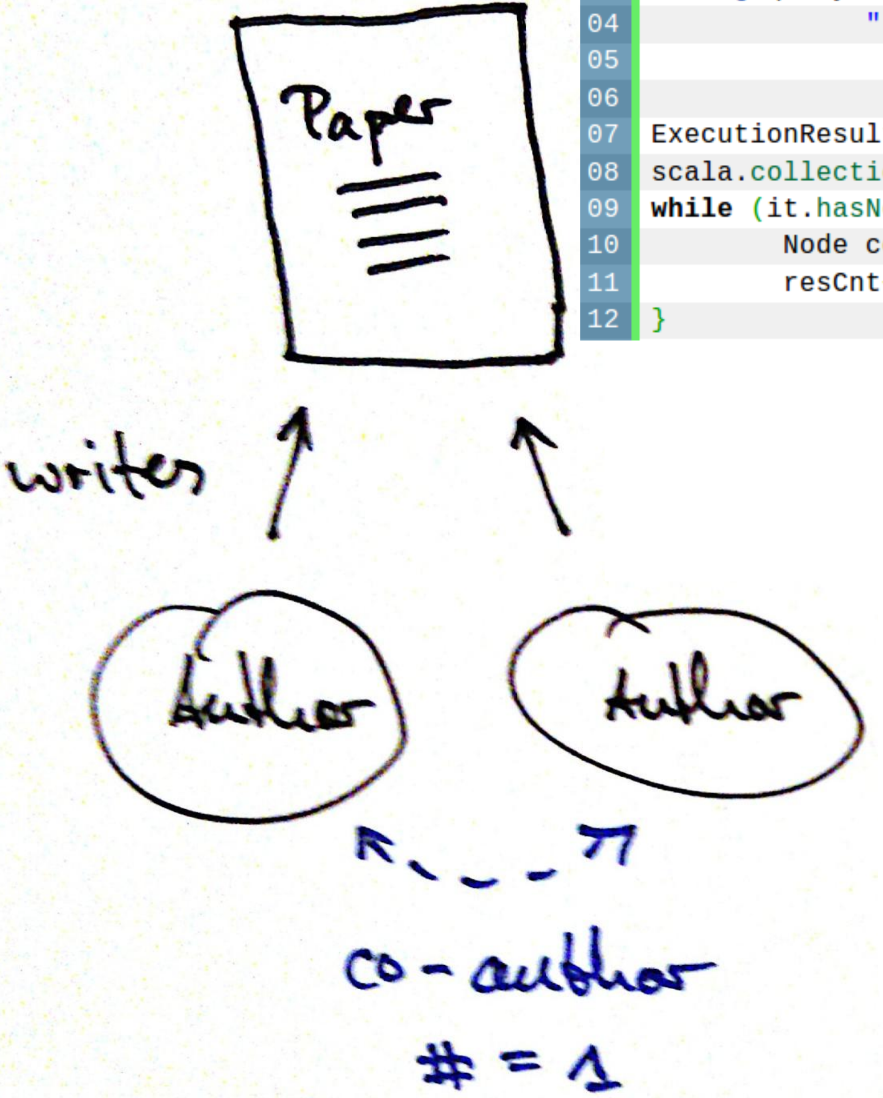
Papers and Authors make up our graph

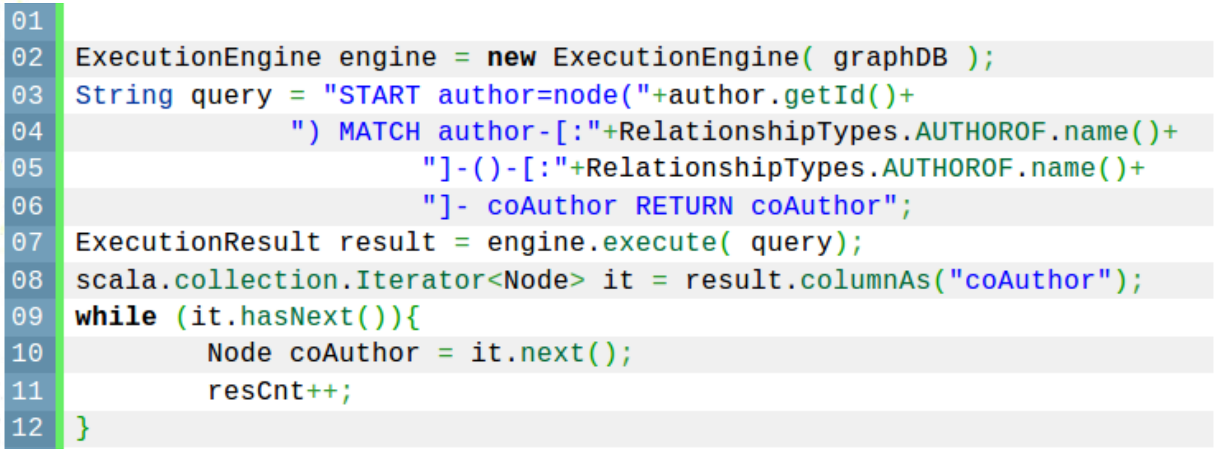


Caclulating Co-Authorship is BFS depth 2



Caclulating Co-Authorship is BFS with depth = 2

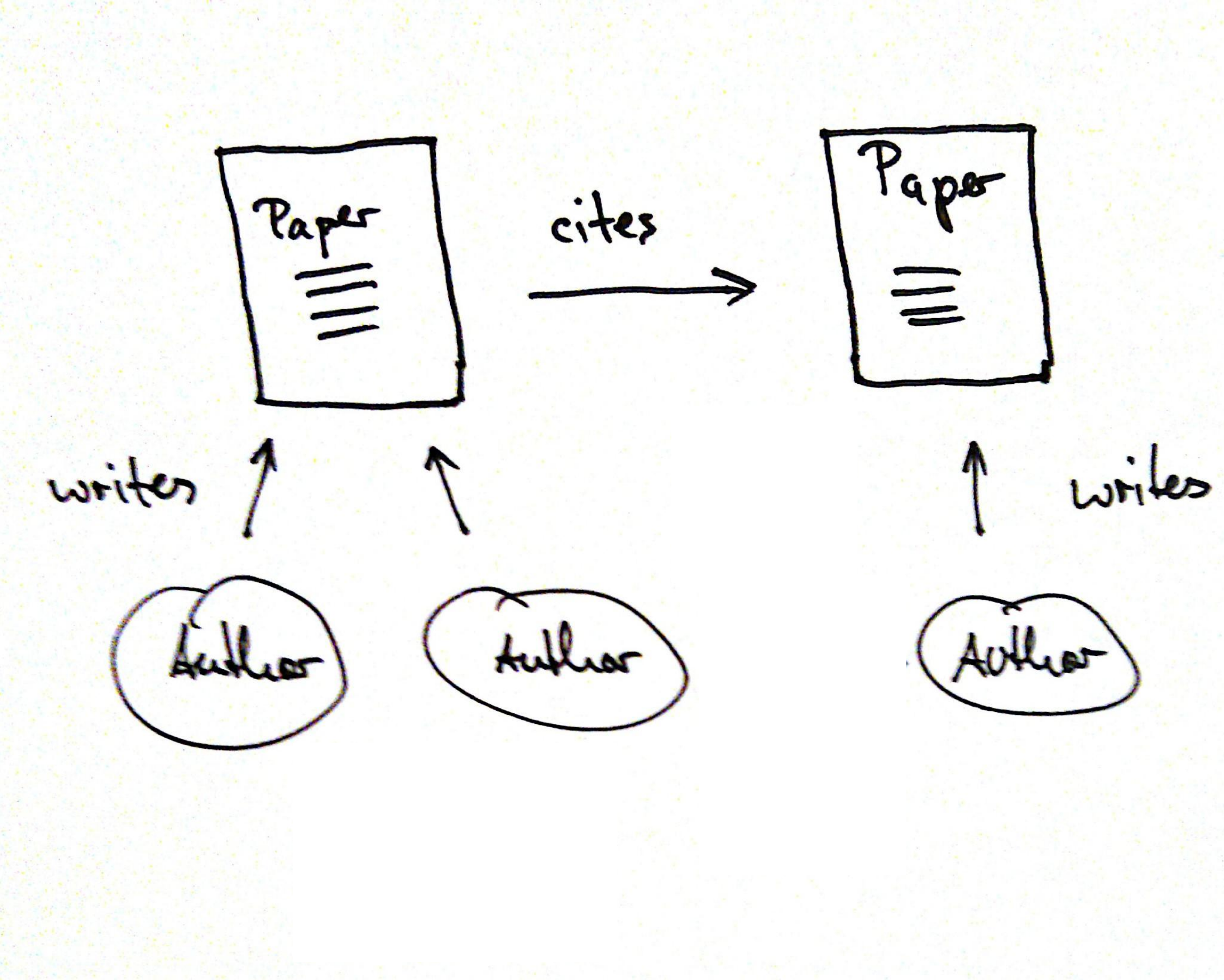




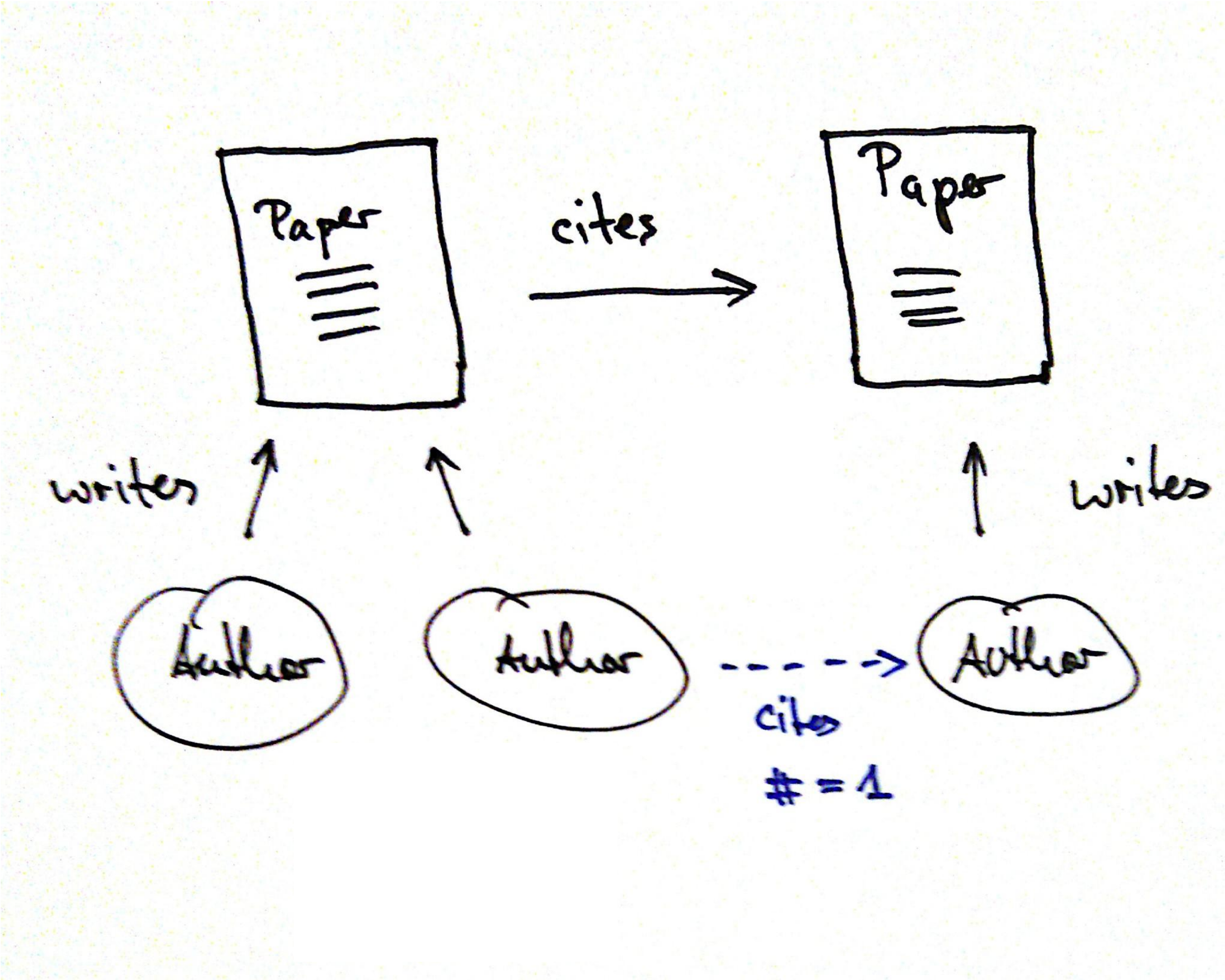
```

ExecutionEngine engine = new ExecutionEngine( graphDB );
String query = "START author=node("+author.getId()+
                "") MATCH author-[:"+RelationshipTypes.AUTHOROF.name()+
                        "]-()-[:"+RelationshipTypes.AUTHOROF.name()+
                        "]- coAuthor RETURN coAuthor";
ExecutionResult result = engine.execute( query);
scala.collection.Iterator<Node> it = result.columnAs("coAuthor");
while (it.hasNext()){
        Node coAuthor = it.next();
        resCnt++;
}
```

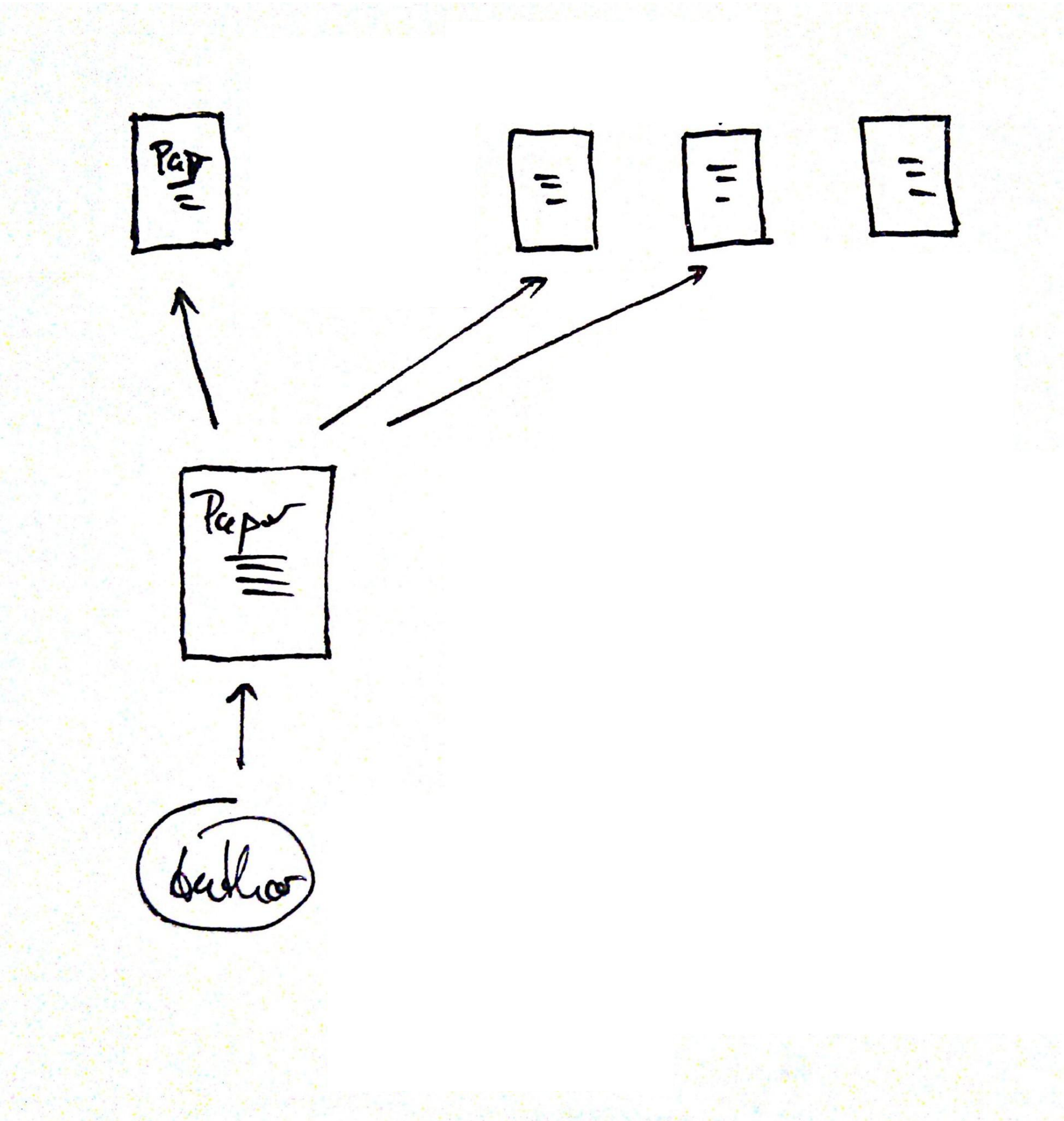
We are also interested in Citations



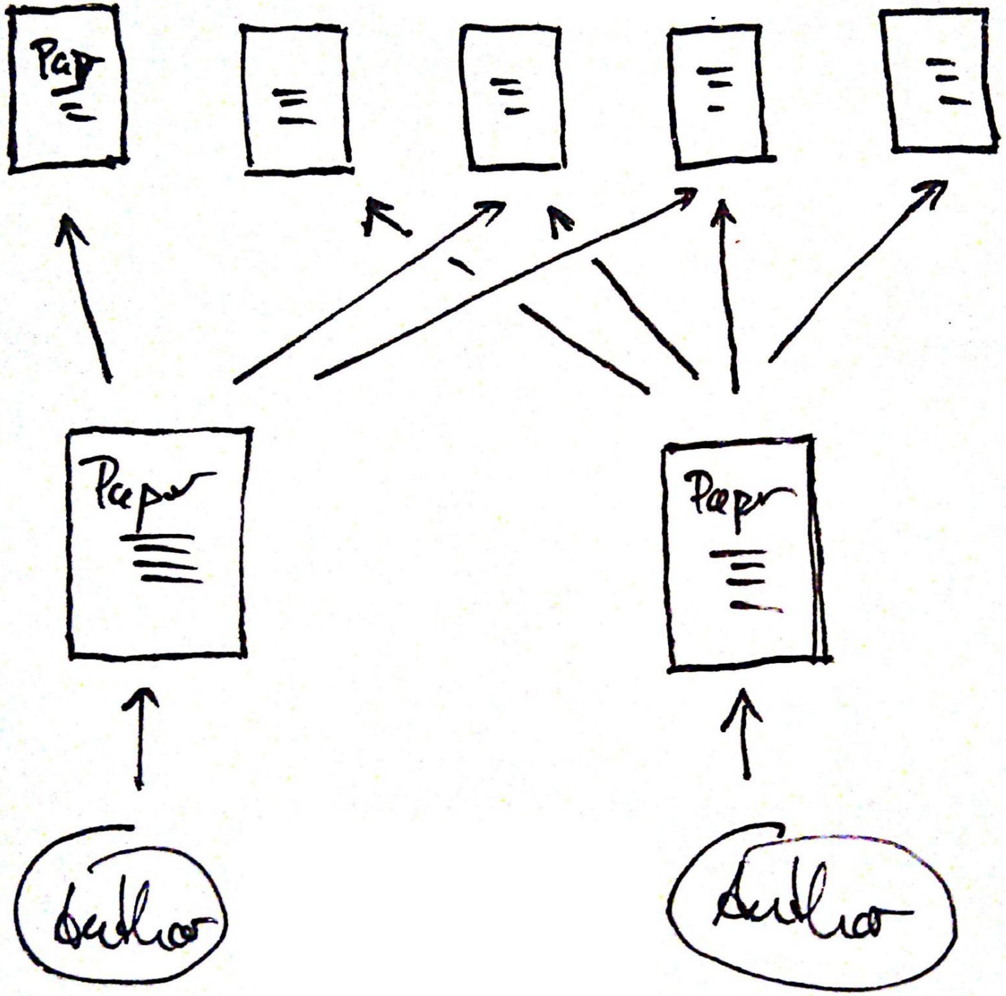
Authors citing other authors is BFS with depth = 3



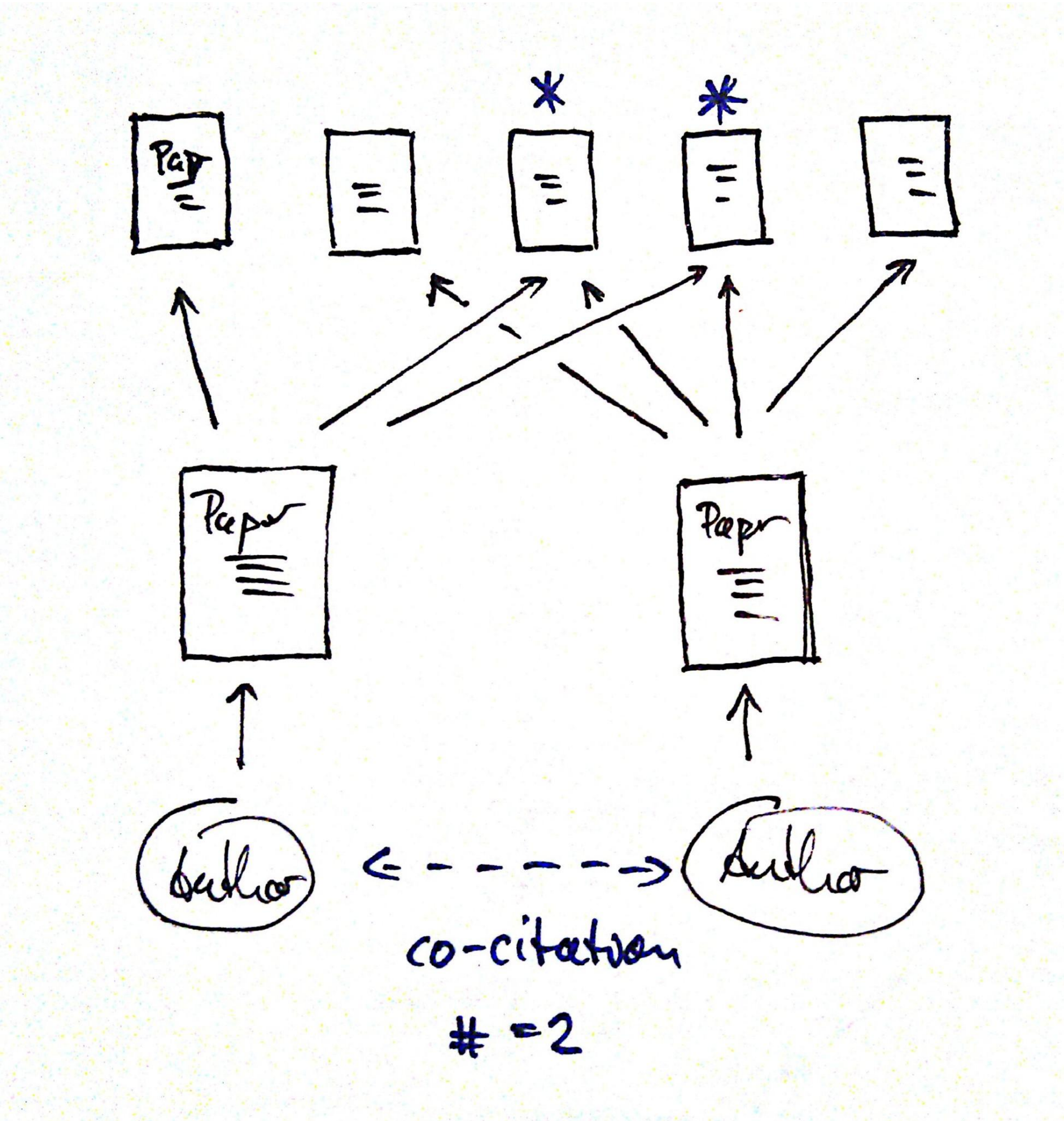
We want to see how similar two authors are



Co citations as a similarity measure



Calculating Co-Citations is BSF with depth = 4



All queries are local Graph traversals

	BFS
Papers written by author?	1
Which papers cite a paper?	
Which papers are cited by a paper?	
What are frequent co-authors of an author?	2
Which authors does author X like to cite?	3
By which authors is a given author often cited?	
Similar papers.	4
Who are the similar authors?	

Did you guys see the tables?

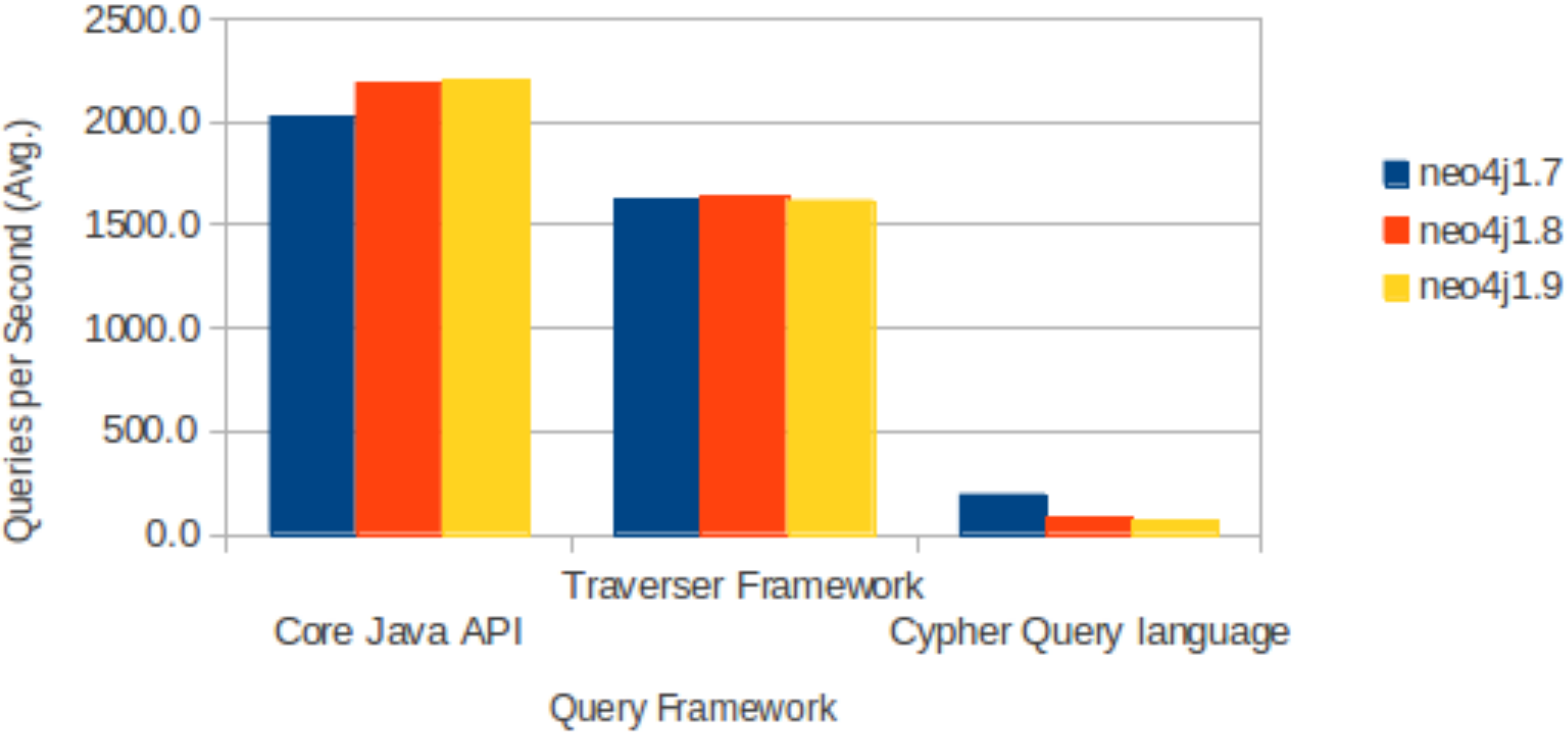
Did you guys see the tables?

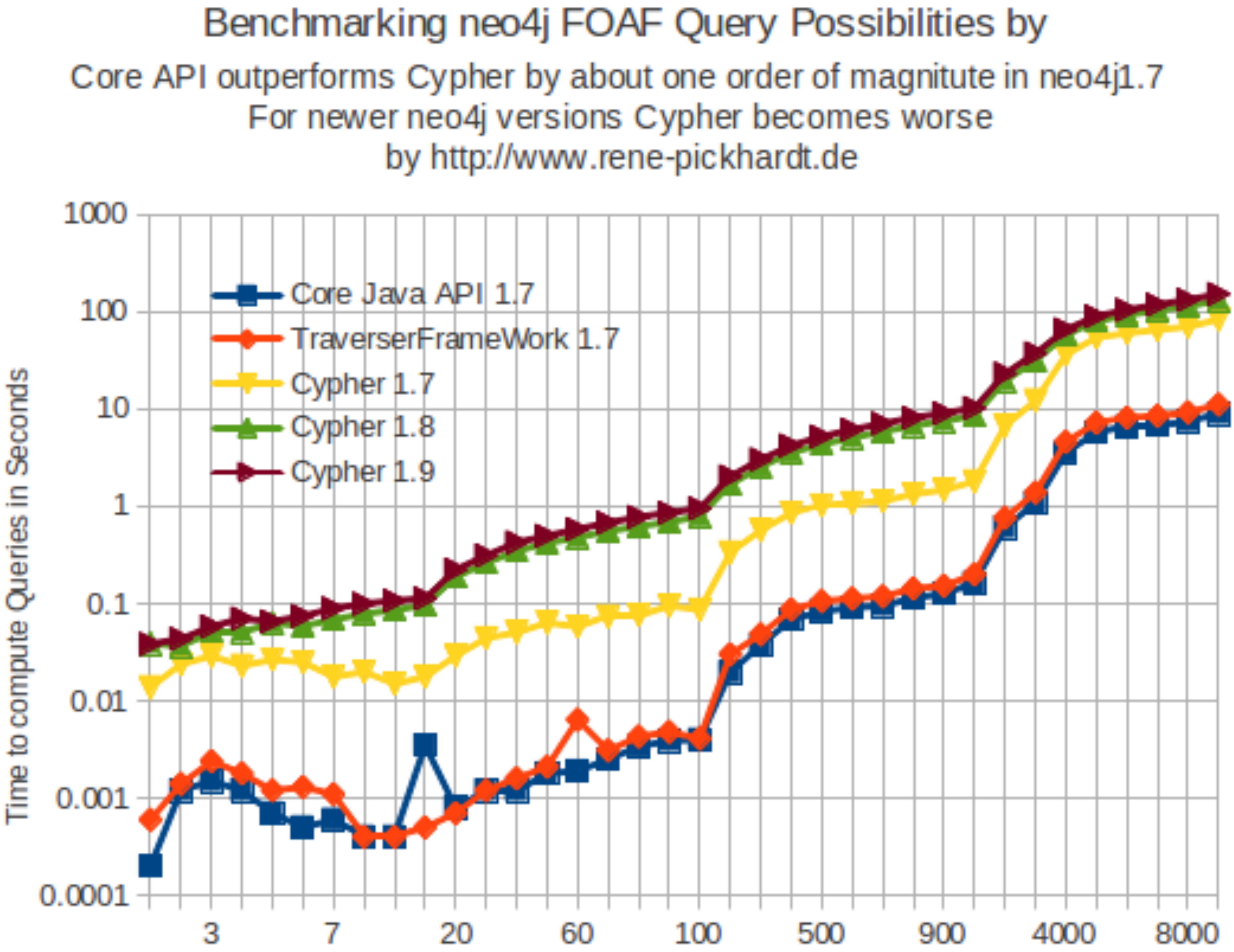
Roadmap

- Introduction
- Data structures for Q & A Systems
- Benchmarking Q & A Systems
- Our data model and requirements
- Lessons learnt from handling graphs in neo4j
- GWT(P)
- Demo Time
- (Personalized) Auto Completion

FOAF like Queries per Second with various neo4j Query frame works

by http://www.rene-pickhardt.de
averages are built on experimants with more than 1k queries





Benchmarking neo4j FOAF Query Possibilities by

Core API outperforms Cypher by about one order of magnitute in neo4j1.7
For newer neo4j versions Cypher becomes worse
by http://www.rene-pickhardt.de

Insert new edges for traversal results

- 1 mio. traversals per second seems a lot but be aware:

- FOAF query (co-authors) <==> BFS depth = 2:
  - 2200 requests per second on average

- Similar authors (co-citation) <==> BFS depth 4
  - 5 requests per sec.

- With existing similar author edges:
  - about 4000 requests per sec!

Try to avoid hitting the disc during data mining

- We calculate Page rank iteratively for ranking in search
- We store page Rank as a node Property
- While calculation keep.
    - HashMap<Node, Float> pageRankMap;
- If you need to hit the disc try to group about 50k writes to a single transaction.

4 Lessons

- Just don't use python with neo4j
- Use neo4j's Java Core API instead of Cypher Query Language
- Store the results of big and expensive Traversals in the Graph
    - calculate via cronjob as a data mining task
    - have a processing priority queue coming from your application
- Writing properties is slow. Try to avoid this

Roadmap

- Introduction
- Data structures for Q & A Systems
- Benchmarking Q & A Systems
- Our data model and requirements
- Lessons learnt from handling graphs in neo4j
- GWT(P)
- Demo Time
- (Personalized) Auto Completion

GWT(P) powers our application

GWT

+ clear separation of server and client side code
+ AJAX: Only transfer data
+ Robust Java server application which integrates smoothly with neo4j, lucene, tomcat,...
+ powerfull API's for web programming including html5

GWTP

+ MVP framework
+ Injection Handling
+ History Support
+ Code Split
+ Faster testing (if we ever start doing this this)
++ we promise it's on our todo

Roadmap

- Introduction
- Data structures for Q & A Systems
- Benchmarking Q & A Systems
- Our data model and requirements
- Lessons learnt from handling graphs in neo4j
- GWT(P)
- Demo Time
- (Personalized) Auto Completion

Roadmap

- Introduction
- Data structures for Q & A Systems
- Benchmarking Q & A Systems
- Our data model and requirements
- Lessons learnt from handling graphs in neo4j
- GWT(P)
- Demo Time
- (Personalized) Auto Completion

Some facts about our search architecture

- Search is based on lucene
  - 1.) its well integrated in neo4j
  - 2.) good search infrastructure also coming in java

- Autocompletions:
  - We didn't go for solr
  - SuggestTree Class by Nicolai Diethelm. (Source Forge)

- Personalized autocompletion:
  - gwtp: do as much computation on the client as possible
  - Pagerank projected onto extended ego network
  - Completely cached on the client.

Further reading and resources

- https://github.com/renepickhardt/related-work.net
- https://github.com/opencitations/OpenCitationsCorpus
- https://github.com/HeinrichHartmann/DiscussionBenchmark
- http://blog.related-work.net/data/ (data sets later LOD)
- http://blog.related-work.net/proposal/ (Proposal)
- http://www.rene-pickhardt.de/get-the-full-neo4j-power-by-using-the-core-java-api-for-traversing-your-graph-data-base-instead-of-cypher-query-language/
- http://www.rene-pickhardt.de/graphity

2 way - Q & A

- Did we miss some technology?
- How do we scale beyond 1 machine?
- Any ideas on absolutly mandetory Features we should not miss?
- What about your Questions?

Thank you very much

Thank you very much!