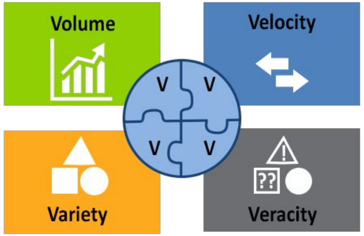
SciPi:

Scientific Publication Analytics Prototype

Dylan Vassallo & Patrick Bezzina

ICS5114 Study Unit Assignment

Introduction



Goals

- Explore dense communities within the publications network
- Track dynamics in authorship patterns over time
- Discover and visualise associations between entities

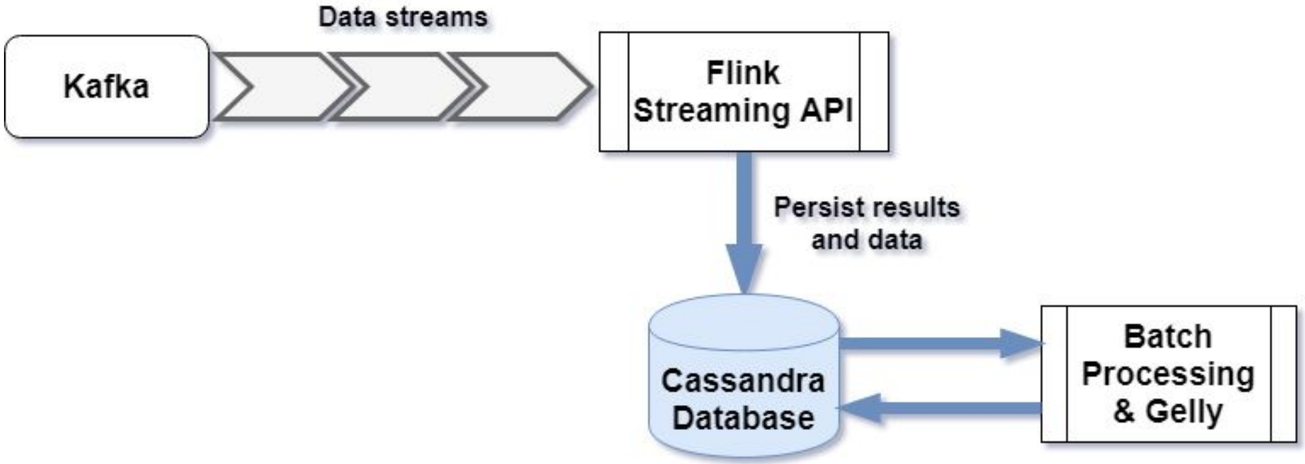
Methodology

Datasets used:

- AMiner (38GB - JSON)
- DBPL (~3GB - XML)
- Different structure, sources and size
- Pushed to Kafka streams (two topics)

Methodology

Overview of the process flow

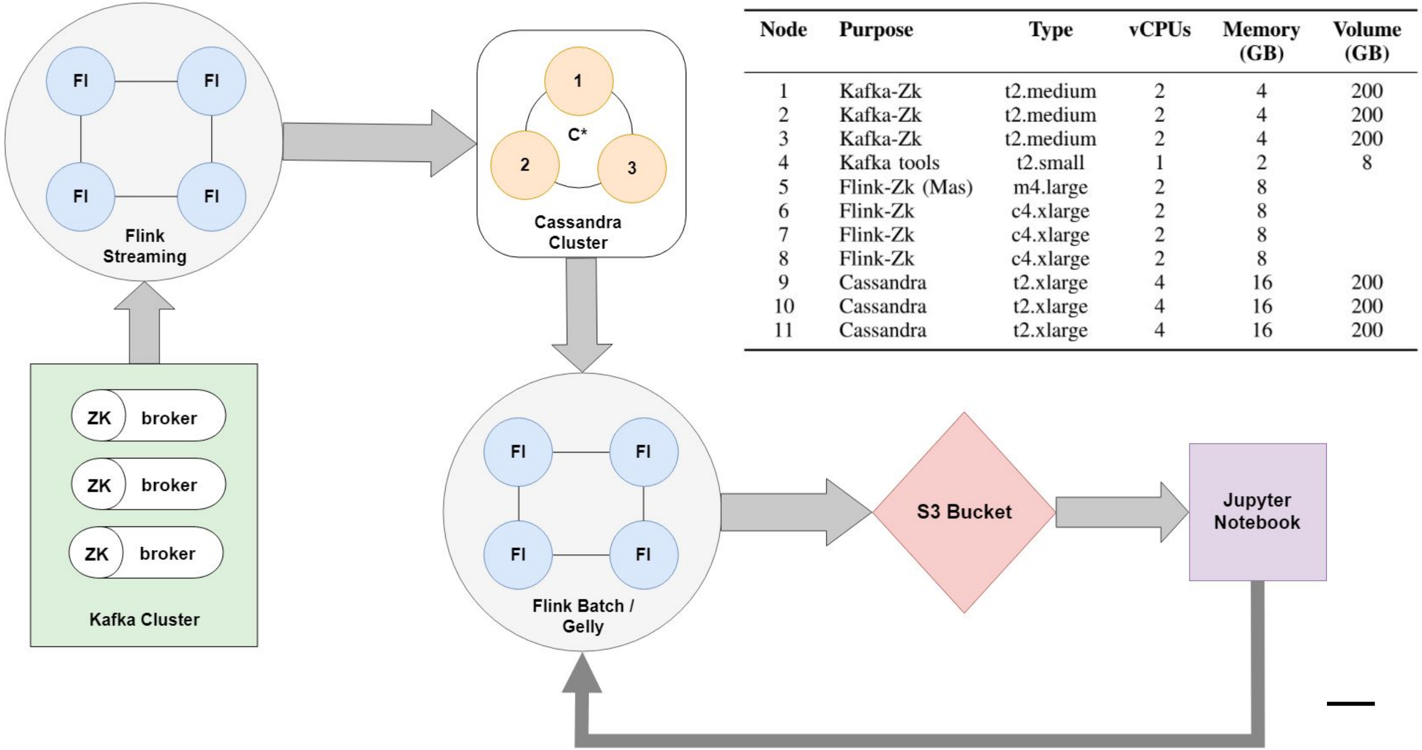


Cloud Setup (AWS)

- 3 nodes for Kafka with Zookeeper
- 1 node for Kafka web tools (monitoring)
- 4 nodes for Flink (on Yarn) : 1 master node + 3 slave nodes
- 3 nodes for Cassandra Database

Cloud Setup Architecture

High level Architecture

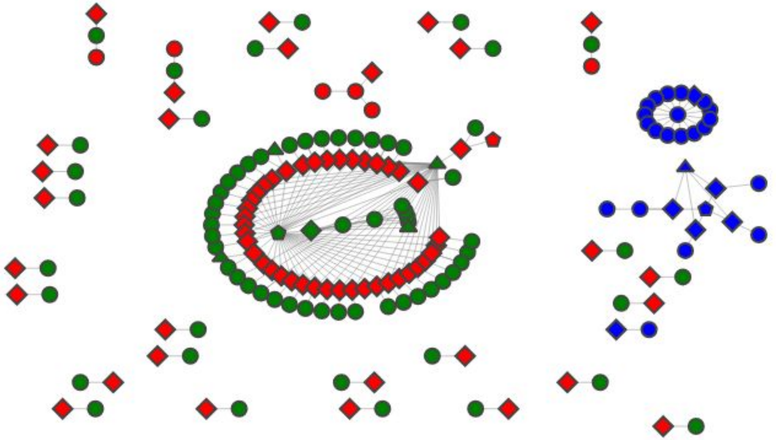


Node	Purpose	Type	vCPUs	Memory (GB)	Volume (GB)
1	Kafka-Zk	t2.medium	2	4	200
2	Kafka-Zk	t2.medium	2	4	200
3	Kafka-Zk	t2.medium	2	4	200
4	Kafka tools	t2.small	1	2	8
5	Flink-Zk (Mas)	m4.large	2	8	
6	Flink-Zk	c4.xlarge	2	8	
7	Flink-Zk	c4.xlarge	2	8	
8	Flink-Zk	c4.xlarge	2	8	
9	Cassandra	t2.xlarge	4	16	200
10	Cassandra	t2.xlarge	4	16	200
11	Cassandra	t2.xlarge	4	16	200

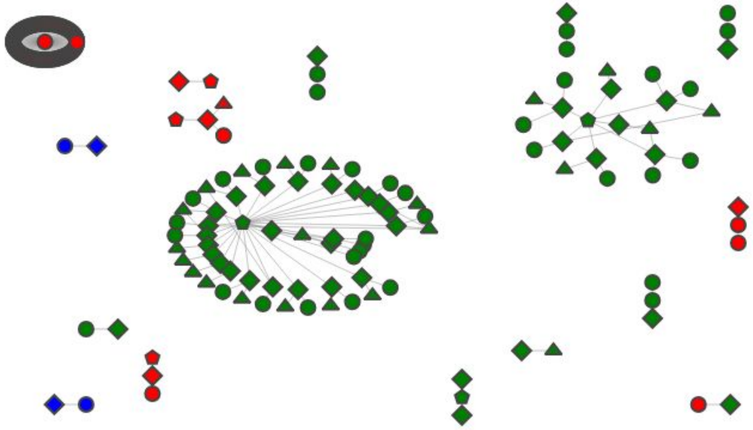
Implementation

Goal #1 : Dense communities

- Construct graph for the publications network
- Algorithm to detect communities (CommunityDetection)
- Metric to gauge strength of the discovered communities
- Evaluate strength between multiple communities



Dense Communities in Publications Network (Layout: twopi)



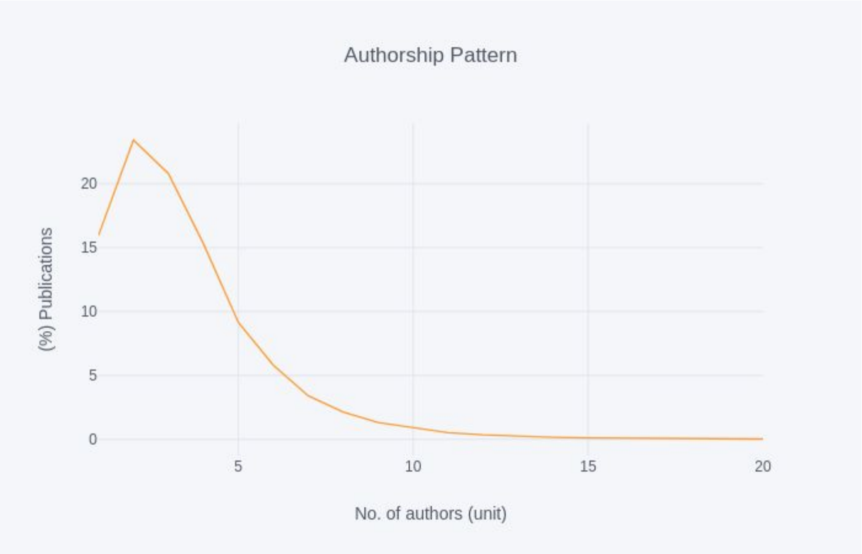
Dense Communities in Publications Network (Layout: twopi)

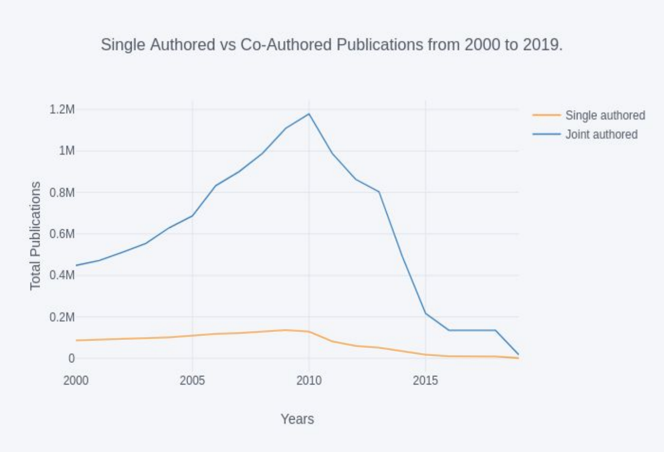
Implementation (2)

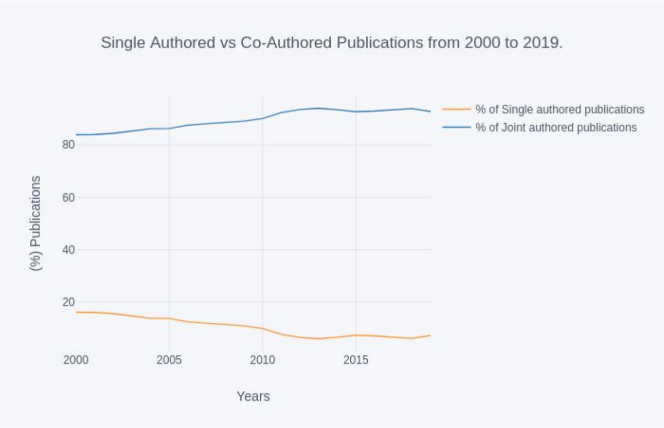
Goal #2 : Dense communities

- Evaluate whether joint authorship is common
- Show single and joint authorships how they vary over time (year-wise distribution and average paper per author)
- Find papers that go hyper with authorship (> 100 authorship)

Single vs joint authorships

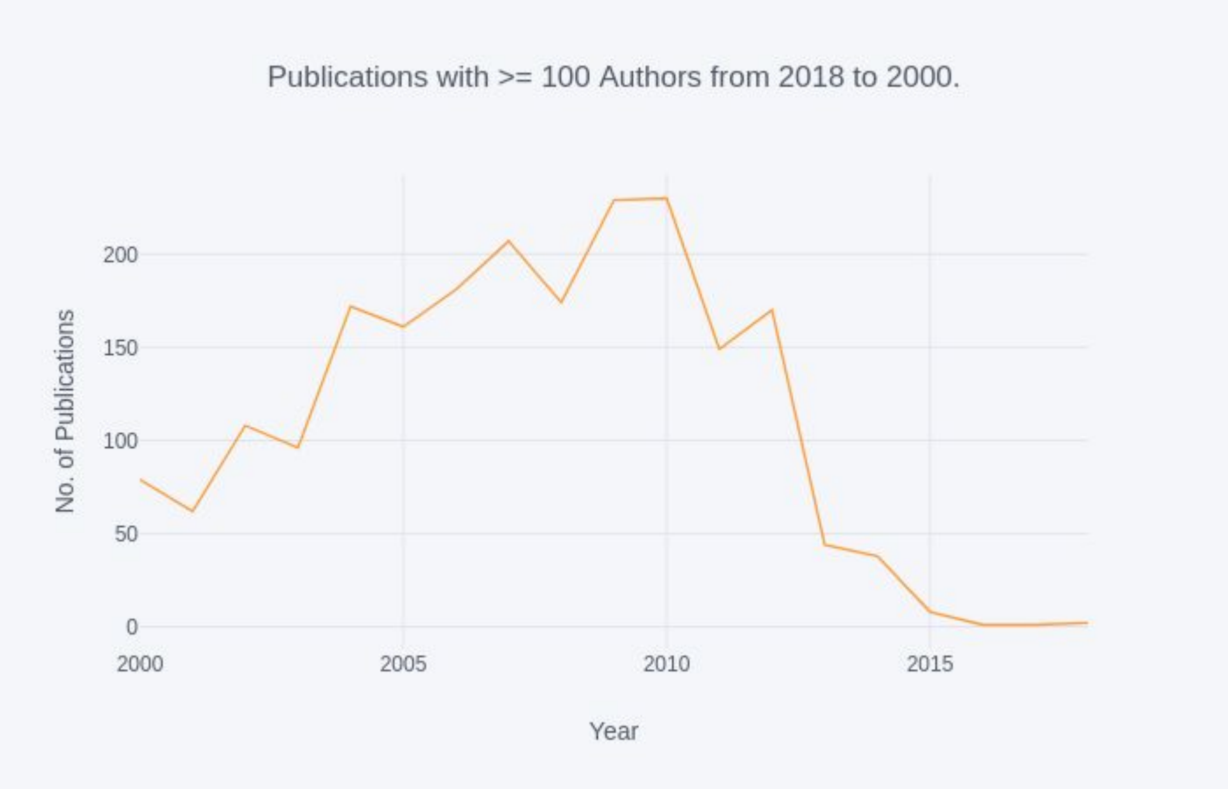






Dynamics over time

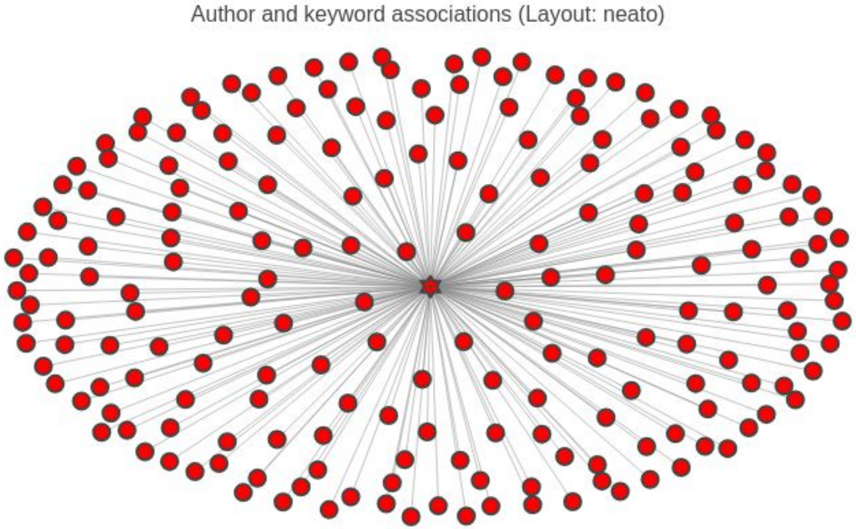
Authorship goes hyper

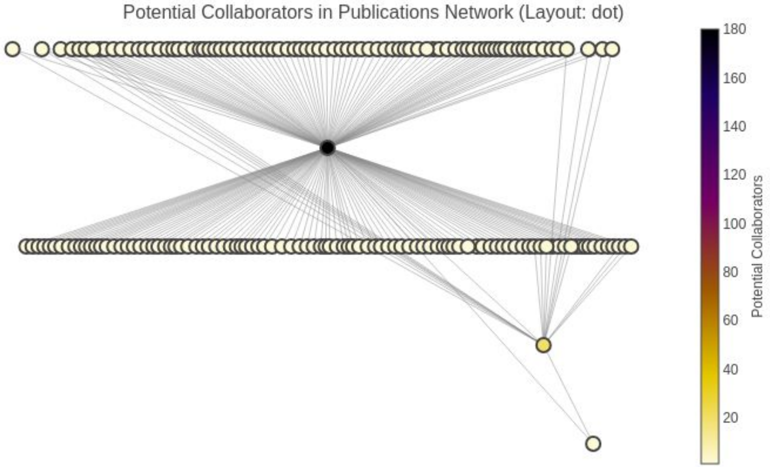


Implementation (3)

Goal #3 : Discover and visualise associations between entities

- Associations between keywords and authors (using the title)
- Graph clustering algorithm
- Recommend potential collaborations


Author and keyword associations (Layout: neato)


Potential Collaborators in Publications Network (Layout: dot)
180
160
140
120
100
80
60
40
20
Potential Collaborators

Challenges

- Streaming data with different structure Kafka
- Setting up the Clusters on AWS
- Choosing the database
  - Initially opted for MongoDB (issues with data sink)
  - Choose Cassandra DB
    - Apache product and it integrates easily with Kafka and Flink
    - Query language (SQL like)
    - Data sink mechanism
- Creating Bipartite Graph with the normal graph structure in Gelly

Challenges

- Recommending potential collaborations
  - Used bipartite graphs
- Finding associations between authors and keywords
  - Used Cosine similarity provided
- Visualising large amounts of data as a graph network
  - Showed only samples when it comes to graph visualisation
  - Positioning vertices in graph visuals is not so easy

Future Improvements

- Gelly streaming (still an early project)
- Graph persistence (Neo4j)
- A better way to find associations between keywords and text (more complex similarity metrics)
- Process publications with different languages
- Real-time visualisations and something more suited for big data (Kibana + Elasticsearch)
- Better way to identify different entities as being the same entity

Demo

GitHub Page:

https://github.com/achmand/SciPi

Two part video

One for the streaming part and the other for the batch processing part