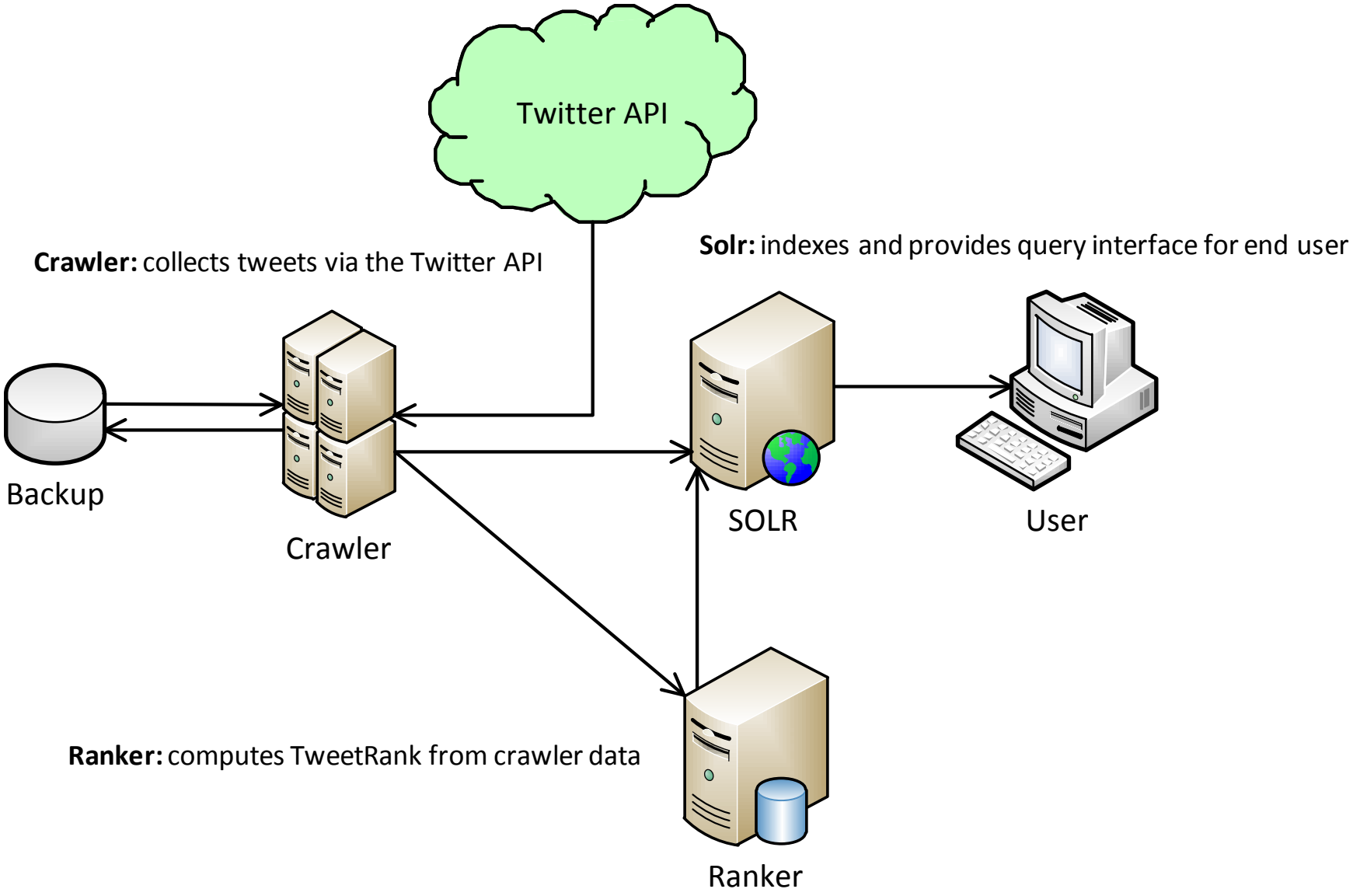
TweetRank

TweetRank is an attempt to apply the PageRank algorithm on Twitter statuses (tweets).

It uses a different rank calculation which considers attributes such as number of replies/retweets and hashtags.

Victor Hallberg	victorha@kth.se
Johan Stjernberg	stjer@kth.se
Joan Puigcerver I Perez	joanpip@kth.se
Alexander Hjalmarsson	alehja@kth.se
Christoffer Rydberg	chrryd@kth.se

Components


Twitter API
Crawler: collects tweets via the Twitter API
Solr: indexes and provides query interface for end user
Backup
Crawler
SOLR
User
Ranker: computes TweetRank from crawler data
Ranker

Crawler

- **Uses the Twitter *HTTP REST* API**
  - Twitter limits the number of queries to 150 per hour
  - Crawler gathers as much data as possible from each query
  - Use multiple proxies to bypass the query limit
  - Runs on multiple threads in multiple machines

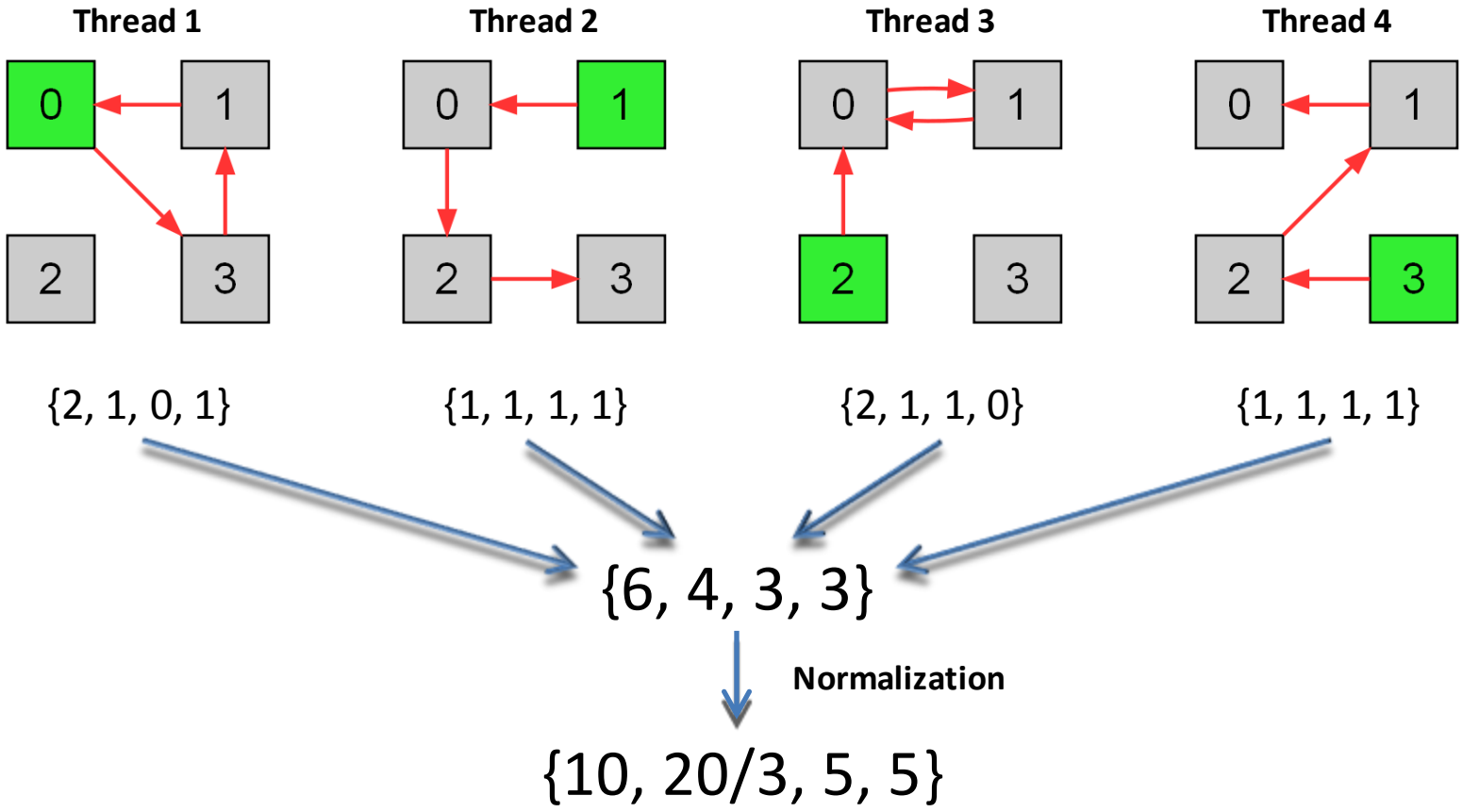
- **How does it work?**
  1. Start with a queue of some users
  2. Pop the first user from the queue and query the Twitter API for tweets and friends for it
  3. Add friends and user mentions in each tweet to the user queue
  4. Send tweet data to Solr and the ranker
  5. Go to 2

Rank algorithm

- Uses the complete path *Monte Carlo* algorithm, stopping at dangling nodes
- Starts a randomized walk from each node
  - at least 100 times
  - at most total number of tweets / 100
- Random path length
  - 20% chance at each node that the surfer stops
- Probability of visiting tweet *x* from *y* estimated from:
  1. Random access
  2. Retweeted or replied
  3. Author of *x* mentioned by tweet *y*
  4. Author of *x* followed by author of *y*
  5. Hashtag shared by tweets *y* and *x*
- A stochastic matrix is built with these probabilities
- TweetRank is the eigenvector of this matrix

Ranker

- Ranker runs on multiple threads, where each thread computes one walk at a time
- Rank for each tweet is calculated as the sum of visits for each node from every walk
- Normalization - divide by the maximum number of visits for a node and then multiply by 10



Solr / Lucene

- Handles indexing and searching
- Crawler sends tweets to be indexed through *HTTP POST* requests
- Scores are calculated as a product of:
  - TweetTrank
  - *tf-idf* (hashtag matches are boosted)
- Current TweetRank data is fetched from a text file on the server
  - Enables rank updates without having to replace (re-index) existing documents
  - Utilizes the *ExternalFileField* feature in Solr

Results (TODO)

(INSERT QUERY INTERFACE SCREENSHOT)