Twitter Big Data

Colin Surprenant
@colinsurprenant
Lead ninja needium

warning: this presentation contains a few rage faces

Twitter - Spring 2012

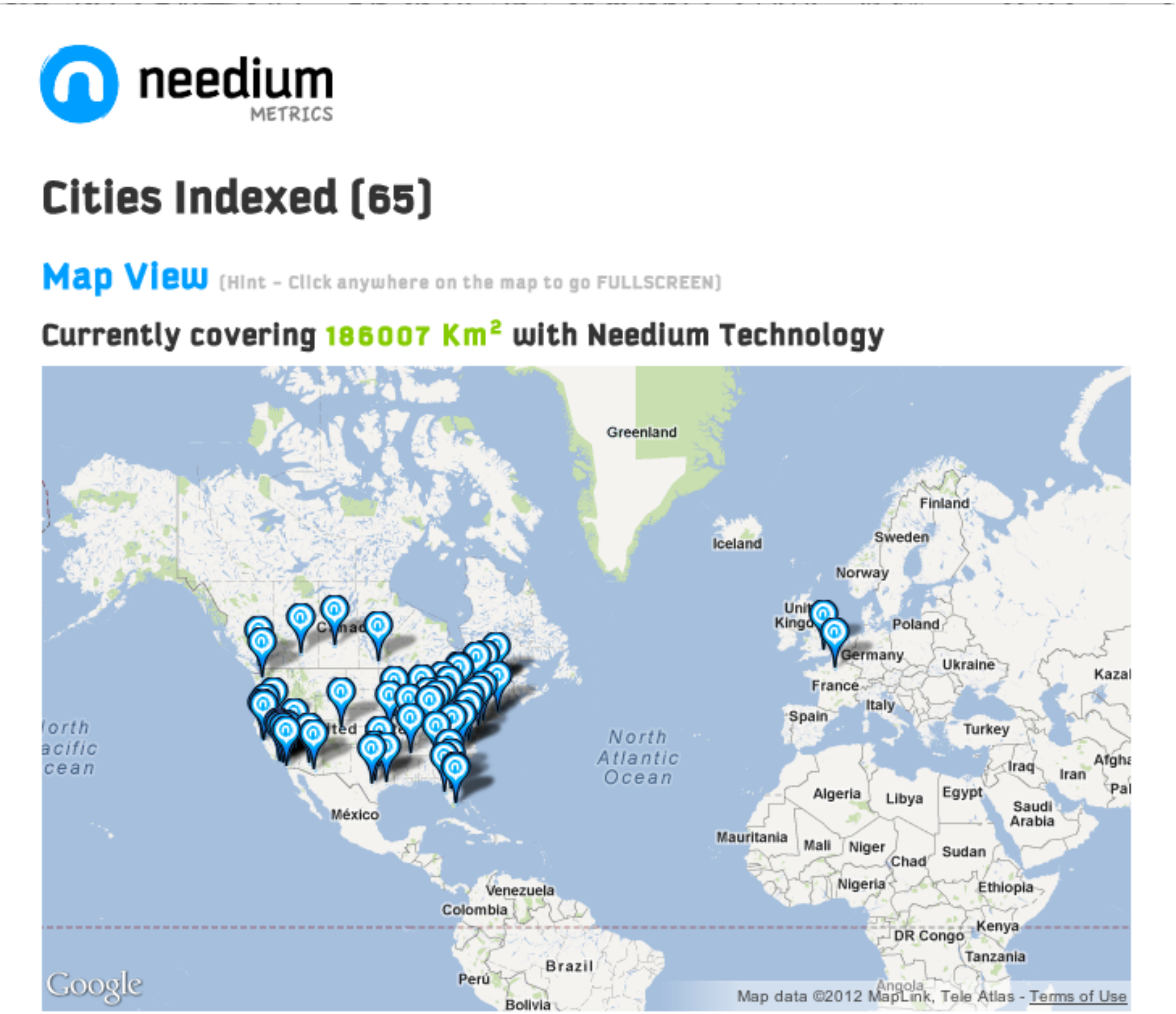
- 350 million tweets/day
- 140 million active users
- >1 million applications using API

Daily Twitter @ Needium

needium

50 000 000	processed tweets
5 000	opportunities
500	messages sent
100GB	data

Needium Geo



needium METRICS

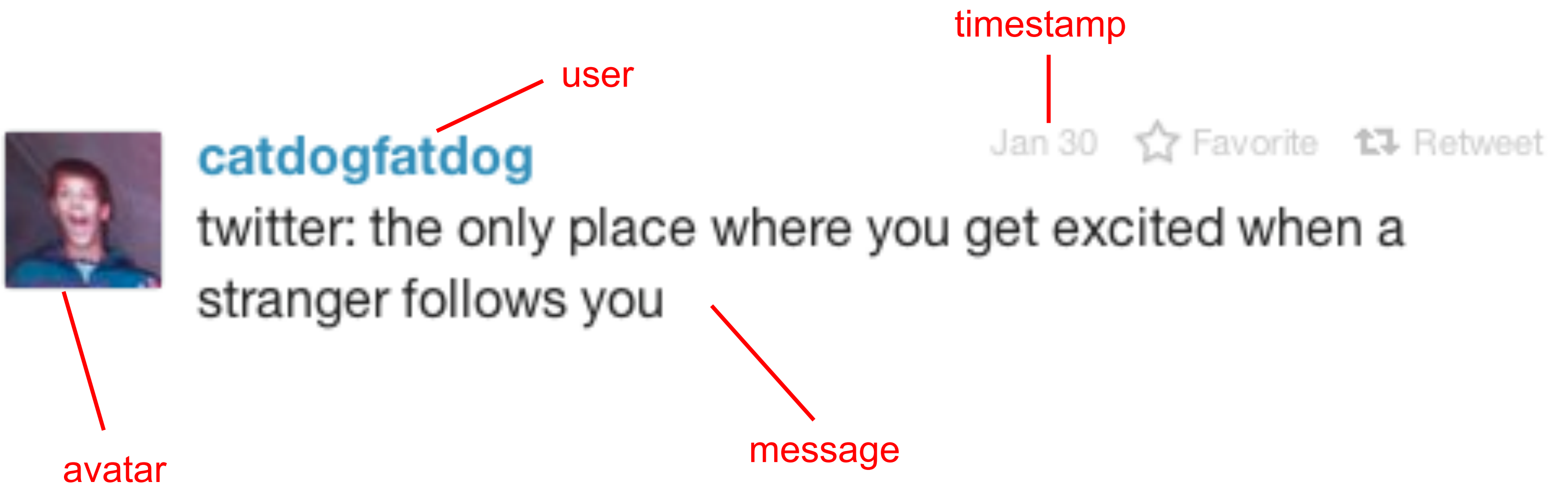
Cities Indexed (65)

Map View (Hint - Click anywhere on the map to go FULLSCREEN)

Currently covering 186007 Km² with Needium Technology

Anatomy of a tweet

What's in a tweet?

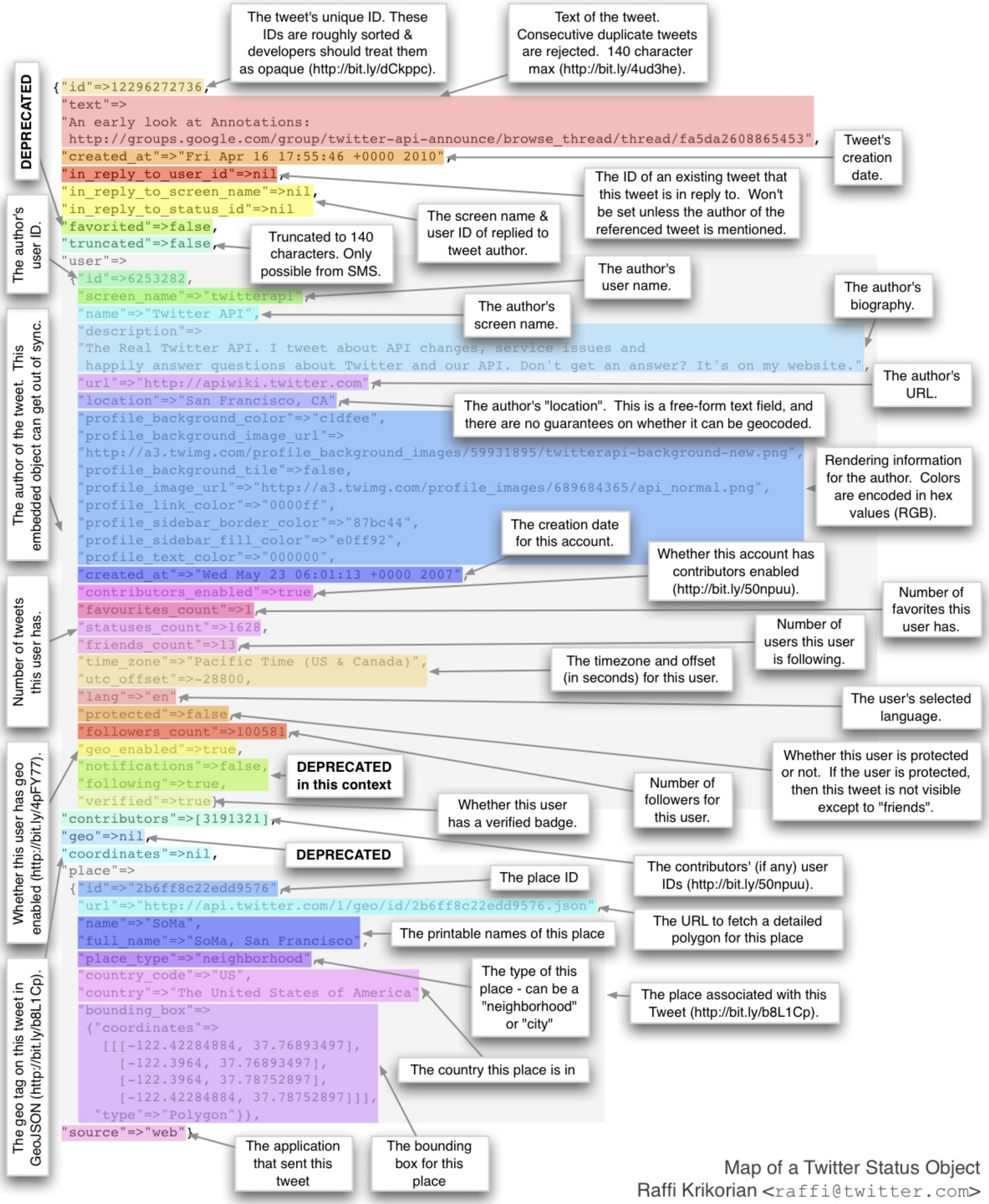


catdogfatdog

Jan 30 ☆ Favorite Retweet

twitter: the only place where you get excited when a stranger follows you

Anything else?



```
{"id"=>12296272736,
"text"=>
"An early look at Annotations:
 http://groups.google.com/group/twitter-api-announce/browse_thread/thread/fa5da2608865453",
"created_at"=>"Fri Apr 16 17:55:46 +0000 2010",
"in_reply_to_user_id"=>nil,
"in_reply_to_screen_name"=>nil,
"in_reply_to_status_id"=>nil
"favorited"=>false,
"truncated"=>false,
"user"=>
 {"id"=>6253282,
  "screen_name"=>"twitterapi",
  "name"=>"Twitter API",
  "description"=>
  "The Real Twitter API. I tweet about API changes, service issues and
   happily answer questions about Twitter and our API. Don't get an answer? It's on my website.",
  "url"=>"http://apiwiki.twitter.com",
  "location"=>"San Francisco, CA",
  "profile_background_color"=>"c1dfee",
  "profile_background_image_url"=>
  "http://a3.twimg.com/profile_background_images/59931895/twitterapi-background-new.png",
  "profile_background_tile"=>false,
  "profile_image_url"=>"http://a3.twimg.com/profile_images/689684365/api_normal.png",
  "profile_link_color"=>"0000ff",
  "profile_sidebar_border_color"=>"87bc44",
  "profile_sidebar_fill_color"=>"e0ff92",
  "profile_text_color"=>"000000",
  "created_at"=>"Wed May 23 06:01:13 +0000 2007",
  "contributors_enabled"=>true,
  "favourites_count"=>1,
  "statuses_count"=>1628,
  "friends_count"=>13,
  "time_zone"=>"Pacific Time (US & Canada)",
  "utc_offset"=>-28800,
  "lang"=>"en",
  "protected"=>false,
  "followers_count"=>100581,
  "geo_enabled"=>true,
  "notifications"=>false,
  "following"=>true,
  "verified"=>true},
"contributors"=>[3191321],
"geo"=>nil,
"coordinates"=>nil,
"place"=>
 {"id"=>"2b6ff8c22edd9576",
  "url"=>"http://api.twitter.com/1/geo/id/2b6ff8c22edd9576.json",
  "name"=>"SoMa",
  "full_name"=>"SoMa, San Francisco",
  "place_type"=>"neighborhood",
  "country_code"=>"US",
  "country"=>"The United States of America",
  "bounding_box"=>
   {"coordinates"=>
     [[[-122.42284884, 37.76893497],
       [-122.3964, 37.76893497],
       [-122.3964, 37.78752897],
       [-122.42284884, 37.78752897]]],
    "type"=>"Polygon"}},
"source"=>"web"}
```

DEPRECATED

The author's user ID.

The author of the tweet. This embedded object can get out of sync.

Number of tweets this user has.

Whether this user has geo enabled (http://bit.ly/4pFY77).

The geo tag on this tweet in GeoJSON (http://bit.ly/b8L1Cp).

The tweet's unique ID. These IDs are roughly sorted & developers should treat them as opaque (http://bit.ly/dCkppc).

Text of the tweet. Consecutive duplicate tweets are rejected. 140 character max (http://bit.ly/4ud3he).

Tweet's creation date.

The ID of an existing tweet that this tweet is in reply to. Won't be set unless the author of the referenced tweet is mentioned.

The screen name & user ID of replied to tweet author.

Truncated to 140 characters. Only possible from SMS.

The author's user name.

The author's screen name.

The author's biography.

The author's URL.

The author's "location". This is a free-form text field, and there are no guarantees on whether it can be geocoded.

Rendering information for the author. Colors are encoded in hex values (RGB).

The creation date for this account.

Whether this account has contributors enabled (http://bit.ly/50npuu).

Number of favorites this user has.

Number of users this user is following.

The timezone and offset (in seconds) for this user.

The user's selected language.

Whether this user is protected or not. If the user is protected, then this tweet is not visible except to "friends".

Number of followers for this user.

DEPRECATED in this context

Whether this user has a verified badge.

The contributors' (if any) user IDs (http://bit.ly/50npuu).

DEPRECATED

The place ID

The URL to fetch a detailed polygon for this place

The printable names of this place

The type of this place - can be a "neighborhood" or "city"

The place associated with this Tweet (http://bit.ly/b8L1Cp).

The country this place is in

The application that sent this tweet

The bounding box for this place

Map of a Twitter Status Object
Raffi Krikorian <raffi@twitter.com>
18 April 2010

How to get the tweets?

- Streaming API

  Subscribe to realtime feeds moving forward

- REST Search API

  Search request on past data (1 week)

Streaming API

public statuses from all users

- status/filter

  track/location/follow
  - 5000 follow user ids
  - 400 track keywords
  - 25 location boxes
  - rate limited

- status/sample
  - 1% of all public statuses (message id mod 100)
  - two status/sample streams will result in same data

Streaming API

per user streams

- User Streams
  - all data required to update a user's display
  - **requires user's OAuth token**
  - statuses from followings, direct messages, mentions
  - cannot open large number of user streams from same host
- Site Streams
  - multiplexing of multiple User Streams

Streaming API

Firehose

need more/full data?   only through partners

- gnip.com
- datasift.com

- filtering/tracking
- partial to full Firehose

What's the catch?

Streaming API

Firehose

Base Twitter data license

$0.10 per 1000 tweets

~$1 million/month

approx for full Firehose

Streaming API

Firehose

startup?

Search API

- REST API (http request/response)
    - search query
    - geocode (lat, long, radius)
    - result type (mixed/recent/popular)
    - since id

- max 100 rpp and 1500 results
- rate limited (~1 request/sec)

Twitter Geo

NO simple way to grab

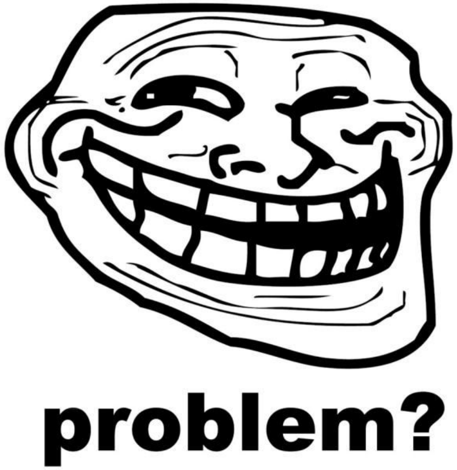
ALL tweets for a given region

OH NOES!

Twitter Geo

Streaming API

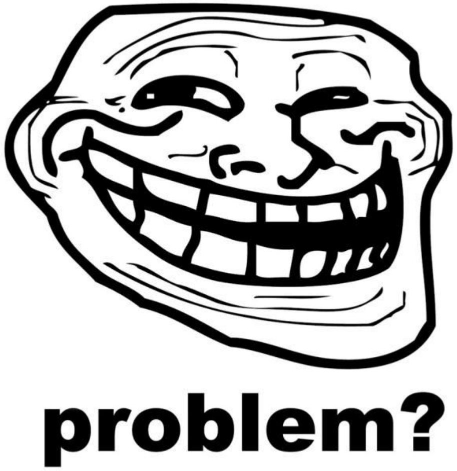
- status/filter + location (bounding box)
  - only tweets with explicit coordinates
  - < 10% of all tweets



Twitter Geo

Streaming API

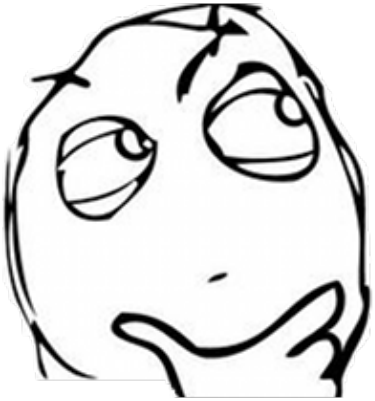
- Firehose
  - < 10% of all tweets contains explicit coordinates
  - must do reverse geocoding on user profile location
  - user profile location is free form



Twitter Geo

Search API

- geocode (lat, long, radius)
  - tweets with explicit coordinates
  - tweets reverse geocoded from user profile location
  - location field: free form text (Montreal / Montreal,Qc / Mtl / Mourial)
  - false positives
  - REST API: not for frequent polling
  - rate limited (1 req/sec/ip)



Twitter Geo

Solutions?

That's your job!

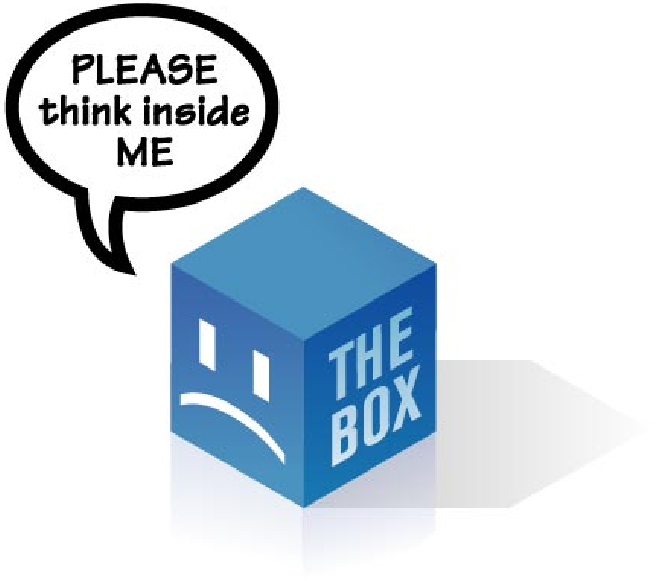
But seriously?

Twitter Geo

- search API *intelligent* polling farm
  - adjust polling interval to minimize polling in relation to traffic

- streaming API status/filter/follow reader farm?
  - find N *relevant* users from city, # stream readers = N / 5000
  - must do reverse geocoding
  - user list dynamic update

- TOS gray zone



Storm

Distributed and fault-tolerant realtime computation

https://github.com/nathanmarz/storm

Storm

The promise

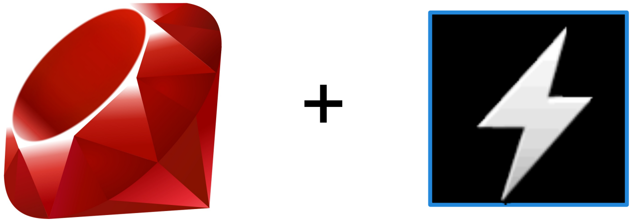
- Guaranteed data processing
- Horizontal scalability
- Fault-tolerance
- No intermediate message brokers
- Higher level abstraction than message passing
- Just work

RedStorm

JRuby integration & DSL for Storm

Simplicity of Ruby + power of Storm

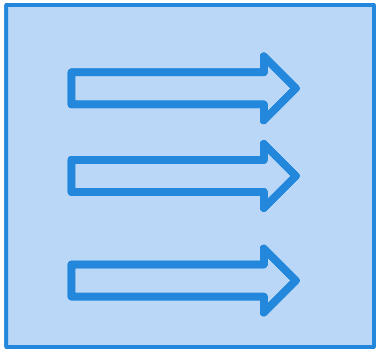
https://github.com/colinsurprenant/redstorm



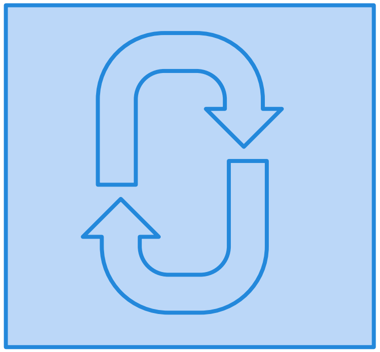
Storm

Typical use cases

Stream processing



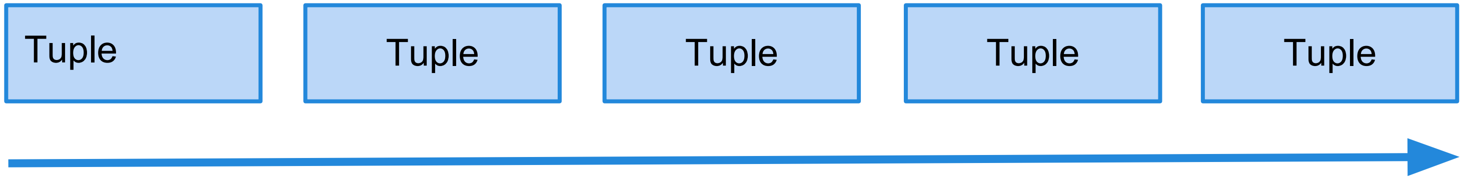
Continuous computation



Storm

Concepts

Streams

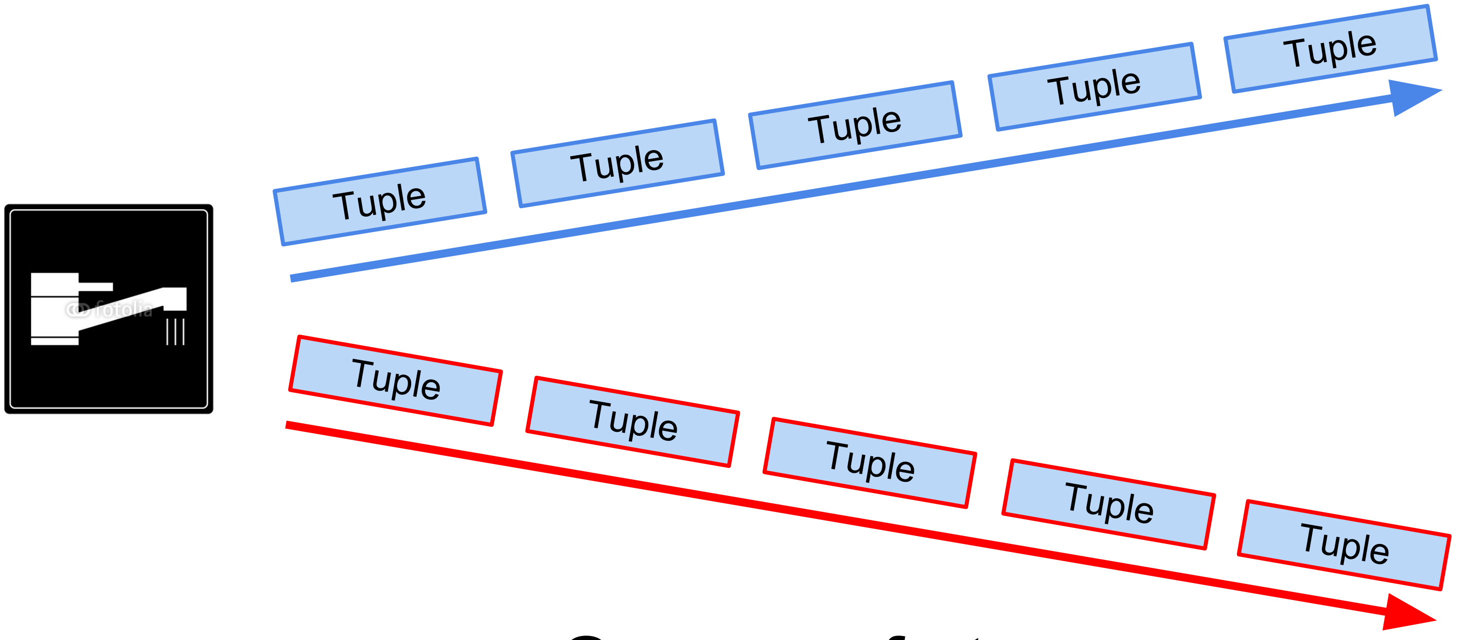


Unbounded sequence of tuples

Storm

Concepts

Spouts



Tuple Tuple Tuple Tuple Tuple

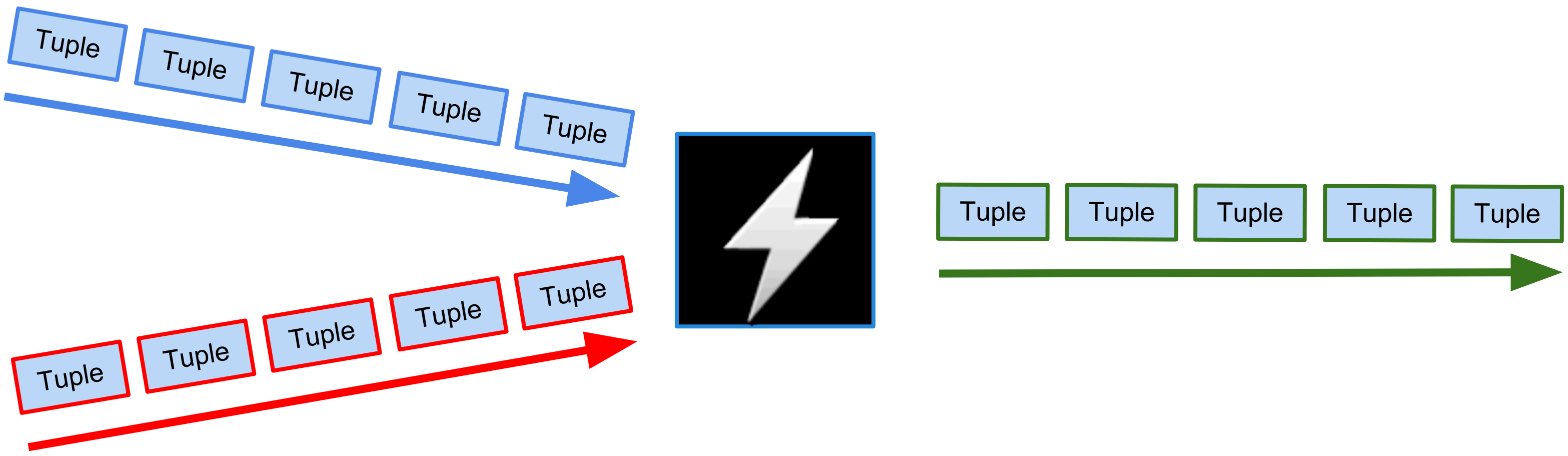
Tuple Tuple Tuple Tuple Tuple

Source of streams

Storm

Concepts

Bolts

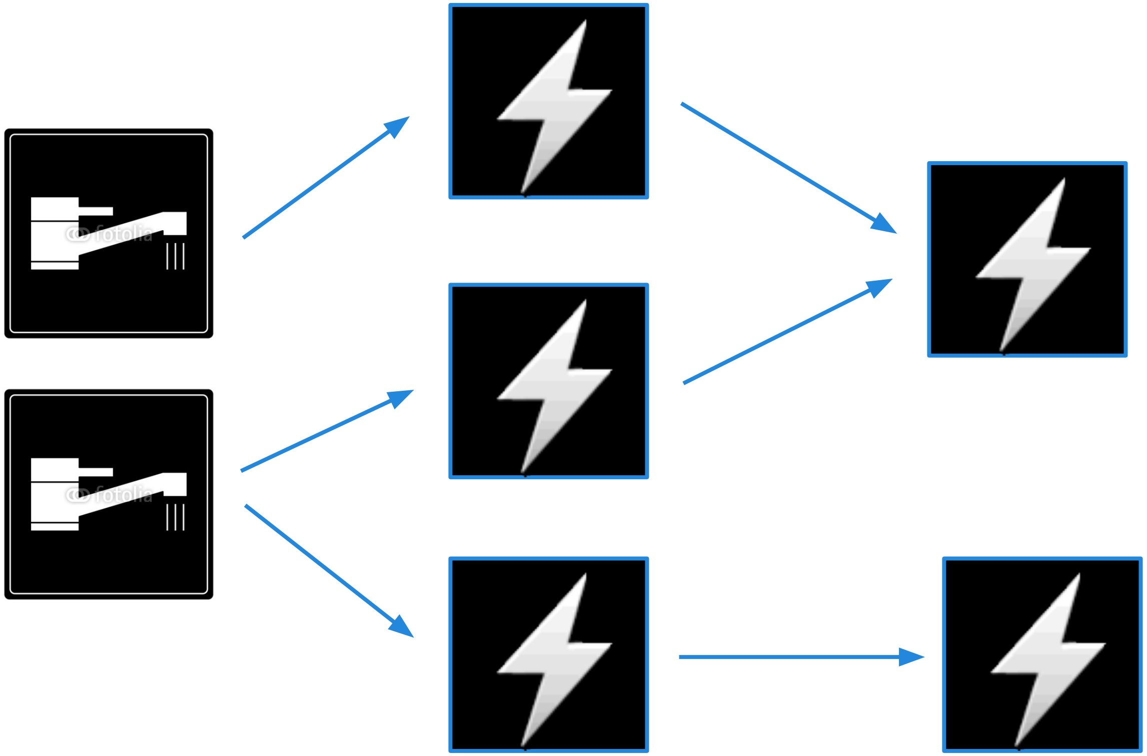


Processes input streams and produce new streams

Storm

Concepts

Topology



Network of spouts and bolts

Storm

What Storm does

- Distributes code
- Robust process management
- Monitors topologies and reassigns failed tasks
- Provides reliability by tracking tuple trees
- Routing and partitioning of stream

Tweitgeist

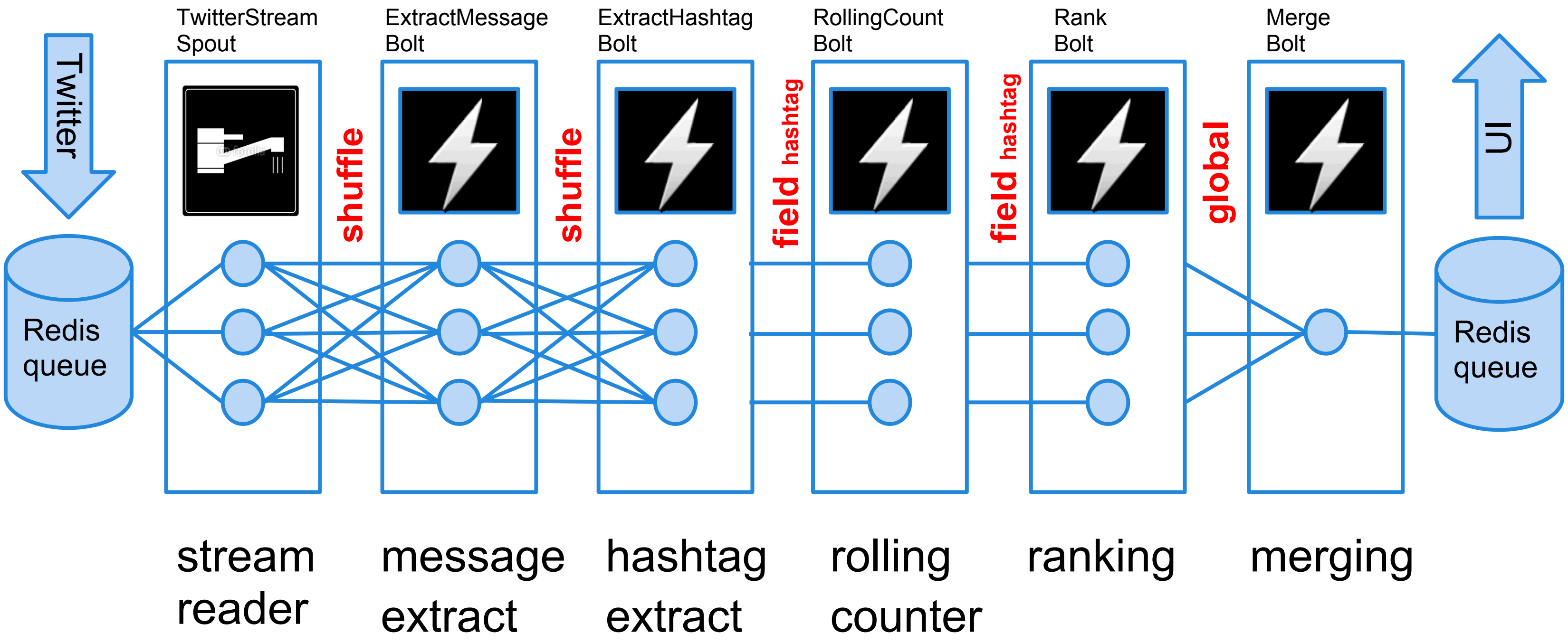
Live top 10 trending hashtags on Twitter

DEMO

https://github.com/colinsurprenant/tweitgeist

Live demo: http://tweitgeist.needium.com

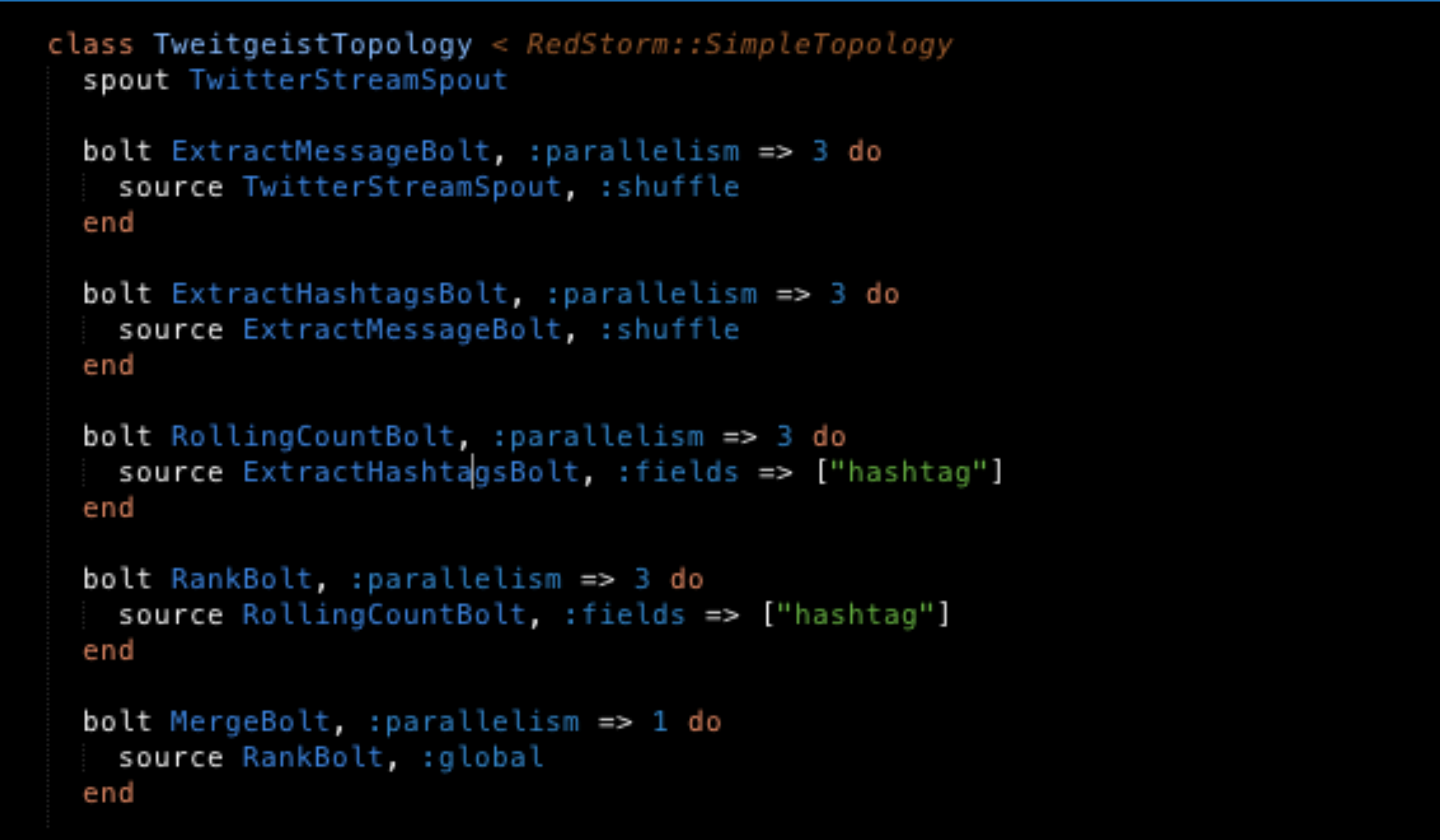
Tweitgeist



Shuffle grouping:	Tuples are randomly distributed across the bolt's tasks
Fields grouping:	The stream is partitioned by the fields specified in the grouping
Global grouping:	The entire stream goes to a single one of the bolt's tasks

Tweitgeist

Topology definition



```
class TweitgeistTopology < RedStorm::SimpleTopology
  spout TwitterStreamSpout

  bolt ExtractMessageBolt, :parallelism => 3 do
    source TwitterStreamSpout, :shuffle
  end

  bolt ExtractHashtagsBolt, :parallelism => 3 do
    source ExtractMessageBolt, :shuffle
  end

  bolt RollingCountBolt, :parallelism => 3 do
    source ExtractHashtagsBolt, :fields => ["hashtag"]
  end

  bolt RankBolt, :parallelism => 3 do
    source RollingCountBolt, :fields => ["hashtag"]
  end

  bolt MergeBolt, :parallelism => 1 do
    source RankBolt, :global
  end
```