A Digital Library for Crowds on the Real-Time Social Web

Jeff McGee and Krishna Y. Kamath

Problem Description:

In this project, we want to build a digital library for transient crowds in highly-dynamic social messaging systems like Twitter and Facebook. A crowd is a short-lived ad-hoc collection of users, representing a “hot-spot" on the real-time web. For example, an event like super-bowl might result in formation of crowds that are discussing various happenings in the game. A set of crowds might be discussing some interesting advertisement while a different set might be discussing a player who might have been involved in an important play. The discovery and tracking of are challenging in comparison to the more static and long-lived group-based membership offered on many social networks (e.g., fan of the LA Lakers on Facebook). Successful detection of these hot-spots can positively impact related research directions in online event detection, content personalization, social information discovery, etc. We will build a framework that allows an analyst or curious user to find interesting crowds and see how they evolve. This framework will have three main parts: a database to store crowds, users, and their messages; a set of crowd detection algorithms and filters; and a tool for searching for crowds by topic, geography, or user-name.

In particular the goals of this project are as follows:

1. Develop a framework to collect and process data from social media websites.
2. Implement modules to discover crowds occurring on the real-time web.
3. Archive the crowds discovered, so that they can be searched and browsed.
4. Provide RESTful APIs for clients to access the collection of crowds.

Architecture for the Proposed Solution:

An architecture for the proposed solution is shown in Figure 1. The various modules of this architecture are as follows:

Crawler: This module deals with interacting with social media websites to collect information. For this project we will be collecting data from Twitter using Twitter Streaming APIs.
Filter: This module removes unwanted data like spam messages, from data collected by the crawler.
Grouper: The grouper module analyses the data collected, to determine crowds and adds them to the DBMS.

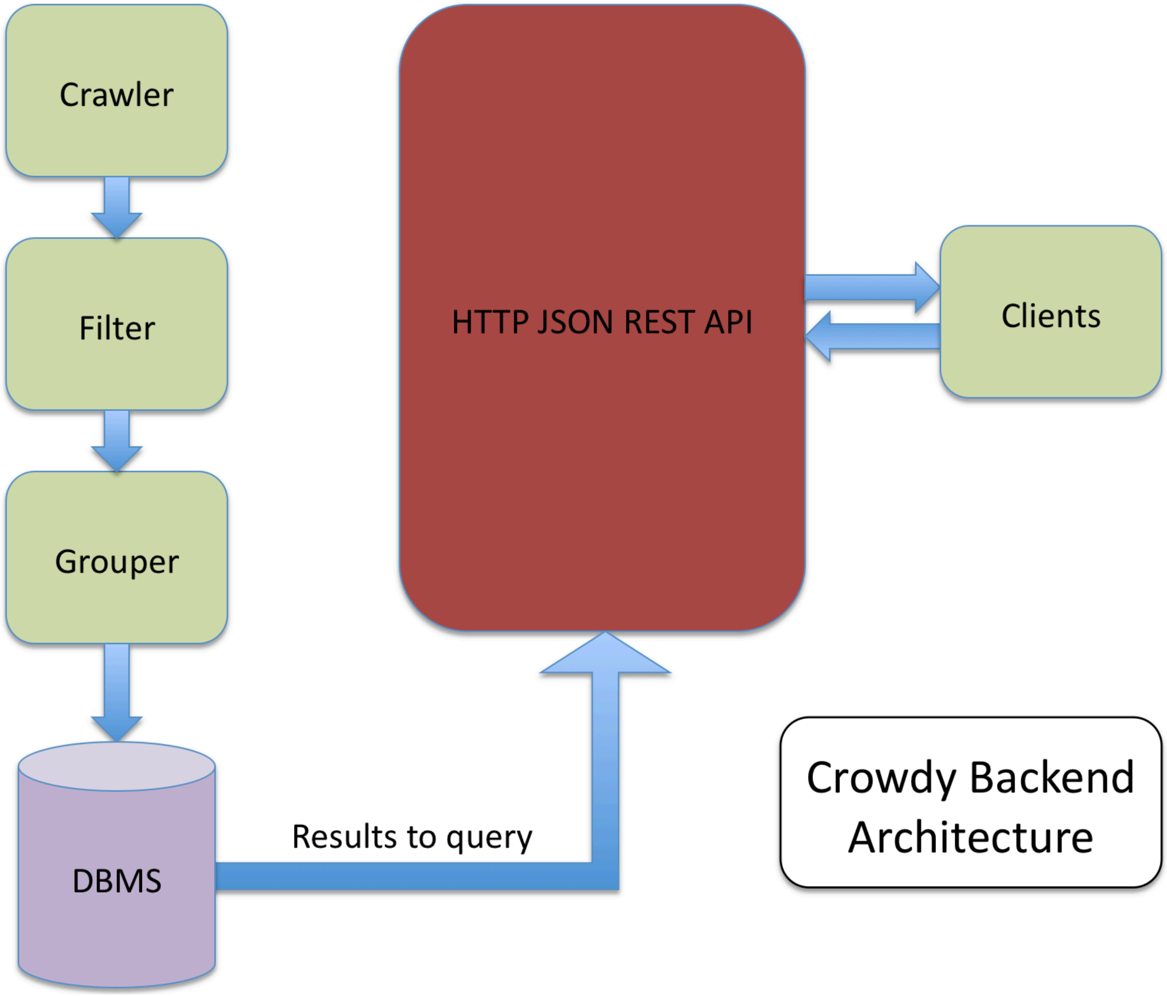

Figure 1

HTTP JSON RESTful APIs: This module will expose a HTTP-based API that clients can query to get data from the collection. The API will return crowds, users, and their tweets.

Tools used in the Project

The tools used in this project are described below:

Beanstalk: Beanstalk is a simple, fast work queue. We use it to connect the crawlers, filters, and groupers together.

Tornado: Tornado is an open source, non-blocking web server. We choose it because it is fast and scalable.

MongoDB: MongoDB is a scalable, high-performance, open source, document-oriented database. We store the crowds discovered by the framework in this DB. The API module interacts with this database to retrieve collections as required by the user query.

Project Timeline

- Mar 1: Working proof-of-concept
- Mar 15: Stable API
- Mar 31: Search functionality
- Apr 15: Support for multiple crawlers, filters, and groupers
- May 1: Working system