The Anatomy of a Real-Time Search Engine

March 30th, 2009 - Alessio Signorini

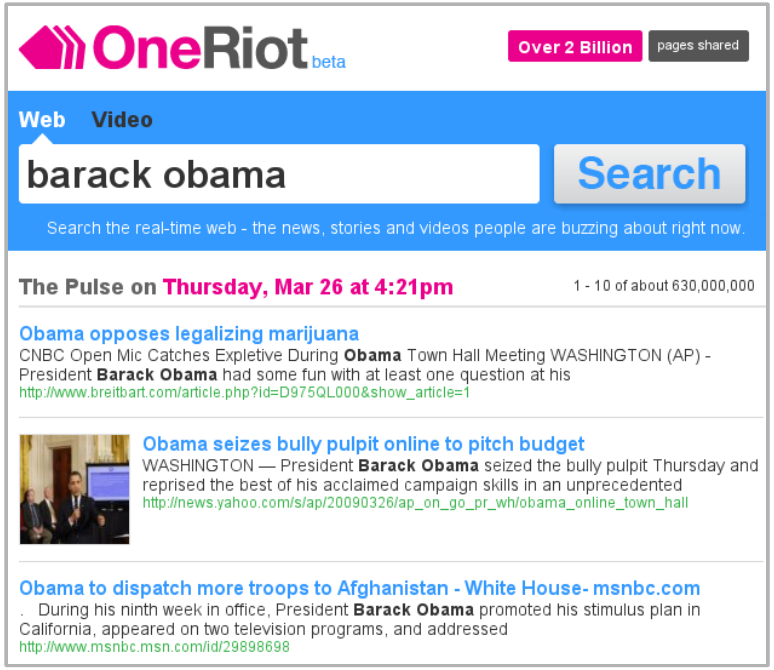
Indexing the Real-Time Web

First Real-Time Search Engine for the Social Web.

We index the stories, videos and sites that people are buzzing about right now.

Our 2 Million users share with us "what's hot" as they surf the web.

Team of 28 in Boulder, CO and San Francisco, CA.



OneRiot

About Me - Alessio Signorini

Born in Italy, before getting serious with computers I played soccer. I am a PhD Candidate at the University of Iowa with a thesis on Query Logs Analysis.

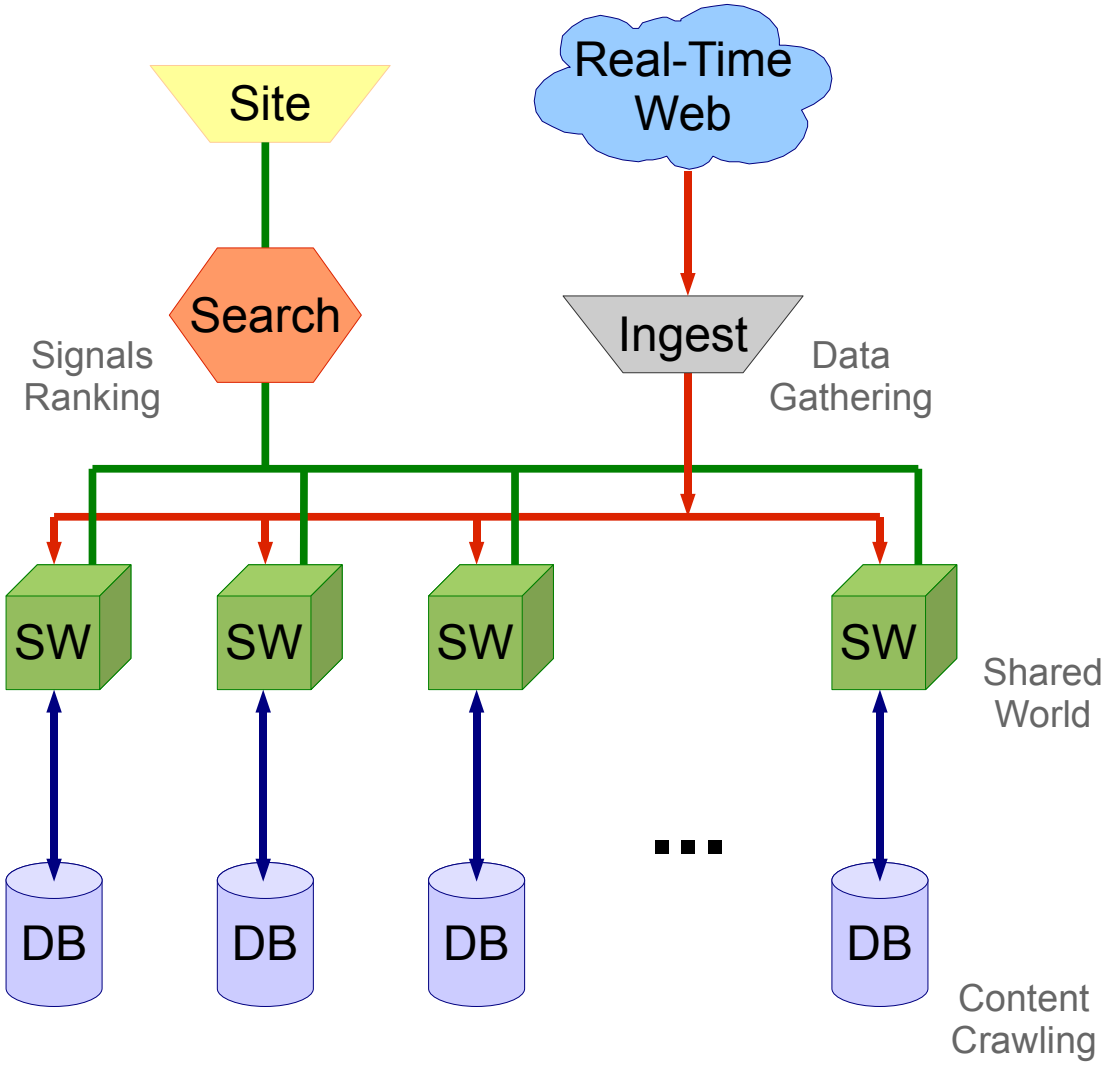
Until the end of last year I was Director of Technology at Ask.com.

- Query and Page Classification
- Ranking and Optimization
- Knowledge Extraction and Answers
- Vertical Search
- Personalized Search

In December 2008 I joined OneRiot as Director of Search and Pulse Technology.

OneRiot

An Overlook of Our Systems



Some Statistics

Total URLs Shared
2+ Billion

Shared URLs/day
30+ Million

New URLs
Available in 43s

Search Time
Less than 500ms

OneRiot

The Curse of Freshness

In a Real-Time engine the freshness of a page is an extremely important signal.

Unfortunately, it is also very hard to balance with relevance and authority. For example, “Barack Obama” has millions of relevant pages but also always something fresh.

Freshness is also often technically hard to handle. Range queries are never efficient and sorting millions of documents is expensive.

We keep our posting lists sorted by freshness and extract the top N pages which meet our criteria for relevance.

OneRiot

Popular Pages may be Boring

At first, one might think that every popular page in the Shared World is interesting. Unfortunately, that is not the case.

At any given moment, cnn.com, google.com and yahoo.com are among the most popular and trafficked pages of the Web.

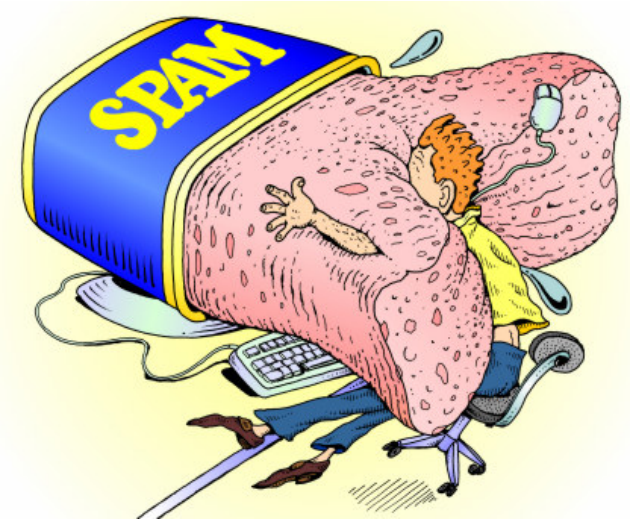
Being too sensitive to accelerations might surface grandma's Flickr account, but not enough will miss breaking news.

To rapidly identify hot pages we monitor many time intervals (from 30 mins to 1 year before), their momentum and ratio of increase.

OneRiot

Re-thinking Ranking for the Shared Web

As for any successful web product, the Shared Web is already getting spammed.



On Twitter and Digg it is already possible to find shared links to paid content, subscription porn sites and advertising.

There is no time to link. People use search. The fast pace at which pages appear and become viral makes it hard for PageRank to work.

A winning strategy must consider where the traffic comes from, the time spent on the page, and the relative importance of the user.

OneRiot

Not Everybody Wants to Read

The Web is not anymore just a collection of “serious” HTML pages. By itself, text relevance does not make users happy. For example:

Searching for “celebrities” users want

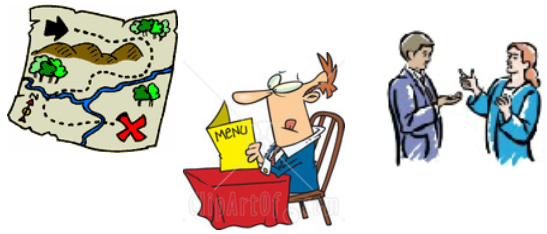
Gossip, Pictures, News, ...

Searching for “movies” users want

Trailers, Show times, Reviews, ...

Searching for “restaurants” users want

Directions, Menu, Opinions, ...



OneRiot

Questions?

Mixed Feelings about Freshness

In a Real-Time engine the freshness of a page is an extremely important signal, but is very hard handle it in Lucene:

Solution 1

Create a field with repeated symbol. Newest pages have more symbols.

Search for that symbol in the field.

Solution 2

Create a new field as in Solution 1 (e.g. "1 1 1 1").

In scoring function ignore initial 80% of max symbols.

Solution 3/4

Have some small and fresh DBs. Use MultiSearcher.

Create timestamp field and use range queries.

OneRiot

Filter by Relevance, sort by Freshness

What we did:

- Save the TimeStamp at first crawling
- Ignore Freshness, rank only by Relevance
- Filter out results below a certain threshold
- Sort by Freshness

Technically:

- Create a subclass of TopFieldDocCollector introducing filter
- Modify Collector() creation specifying TimeStamp in SortField

The initialization of a Collector creates a cache for each sorting field specified for sorting. Remember to warm up your searchers!

OneRiot

Not every Busy Page is Hot

It might seem easy to identify hot sites if you can look at users traffic. Unfortunately, big pages like yahoo.com or cnn.com are always busy.

Solution 1

Compare traffic with previous hour.

Traffic on general websites tend to decrease over night.

Solution 2

Compare traffic against 24h ago.

Weekly events create peaks for sites like ESPN.com.

Solution 3

Compare traffic with last week.

One-Day Sales bring everybody on Target.com.

OneRiot

Combine Everything and some more...

What we did:

- Consider total amount of traffic
- Keep at least 1 year usage of statistics
- Use flexible data structures
- Always use smallest integer to contain data

Technically:

- Use relative increase of hits. Factor in ratio of increase.
- Consider multiple accelerations. Account for site importance.

New pages are tricky: with no history they have big accelerations. You might miss breaking news or surface grandma's flickr account!

OneRiot