Gossip and Inconsistency in a Search Engine

Knut Magne Risvik

Distributed Index Serving 101

Document sharded

All machines engage.
Completes the search.
Merge results.

Term sharded

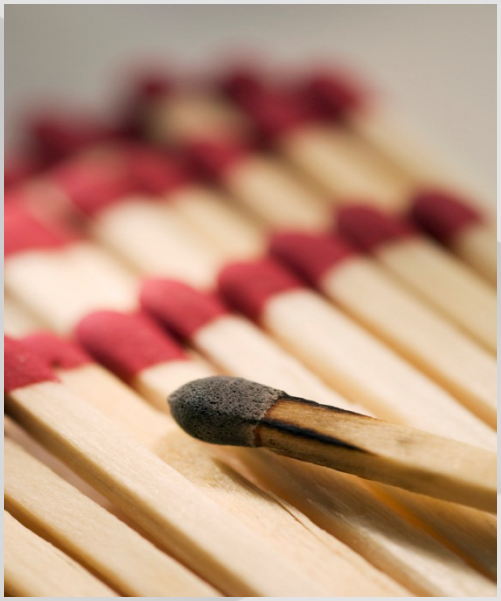
A few machines engage.
Cross talk.
Works on disks…

Search Funnel

All documents

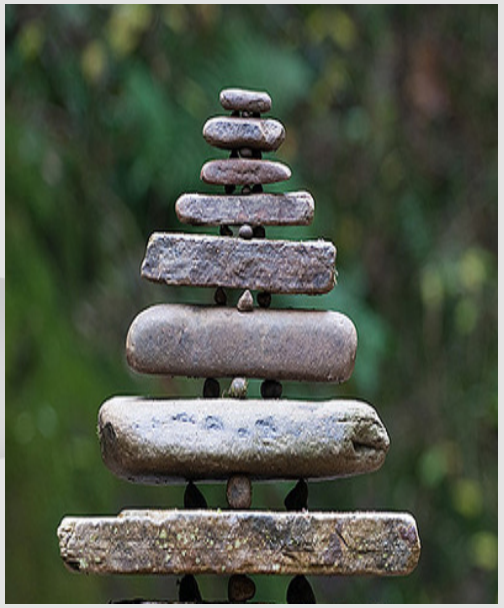
100B+

Matches



1000+
Boolean* search
Prune loosers

Ranked results



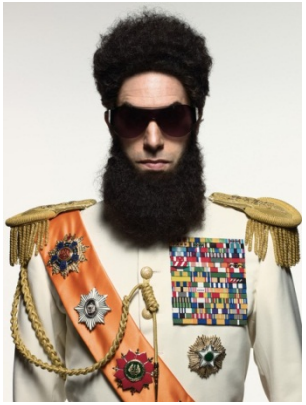
10s
Complex ranker
Find winners
Error correcting

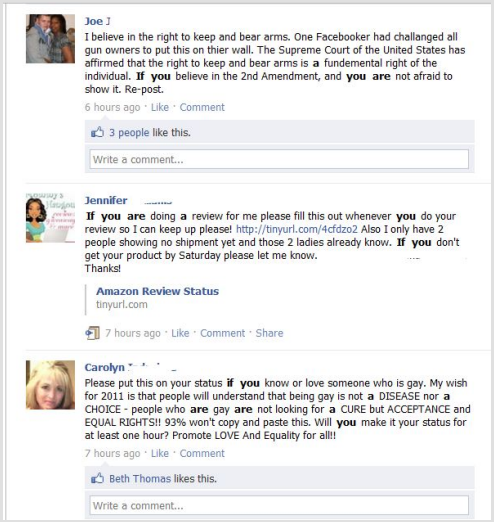
Updates

Organized
Local to each bucket

Slightly more complex
Global

Lame dictator way





Get updates

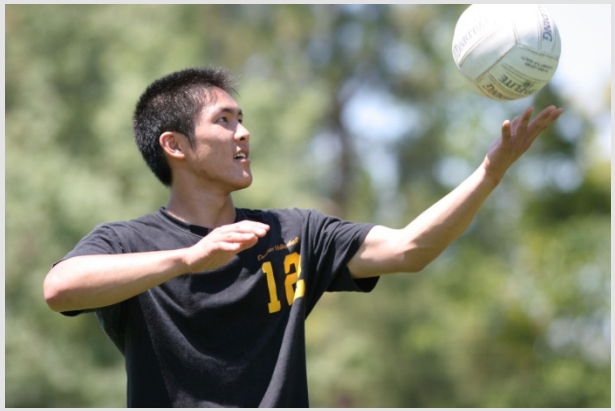
Drain traffic

Synchronize

Gossip way

Update rank data (easier)

Partition updates
Spread mutations



Immediately serve

Gossip about your state and replicas

Yummies and Yucks

No need for the dictator

Faster time to serve

Failure resilient (dictators tend to be unreliable)

False positives

Always inconsistent index

Error correction needed, might loose some

Still need some sync (rank data first)

Final thought?

- What is the real In-consistency of the data for a search engine?
- Still need to measure to possible effect from false negatives.