MAPR™ TECHNOLOGIES

Old and New Building Blocks Come Together For Big Data

- Contact:

  tdunning@maprtech.com

  @ted_dunning

- Slides and such

  http://slideshare.net/tdunning

- Hash tags: #mapr #goto #d3 #node

Embarrassment of Riches

- d3.js allows really pretty pictures
- node.js allows simple (not just web) servers
- Storm does real-time
- Hadoop does big data
- d3 allows very cool visualizations

D3 demo

node demo

Hadoop demo

But …

- Web camp
  - everything is a service with a URL or a DOM
- Big data camp
  - non-traditional file systems
- Everybody else
  - files and databases

- They don't like to talk to each other

Why Not Tiered Architectures?

- Tiered architectures
  - translations between services and cultures
  - standard corporate answer

- Feels like molasses

The Vision

- Integrate
  - multiple computing paradigms
  - many computing communities

- How?
  - common storage, queuing and data platforms

For Example, …

- Incoming documents with text
  - store in file-based queues
  - index in real-time using Storm and Solr
  - add initial engagement class, “don’t-know”
- Search for documents using original text
  - add random noise, small for well understood docs, large for “don’t-know” docs
- Record engagement

Add Analysis

- Process engagement logs
  - item-item cooccurrence
  - user-item histories
- Update search index
  - indicator items
  - decrease uncertainty on well understood docs
- Update user profile
  - item history

Search Again

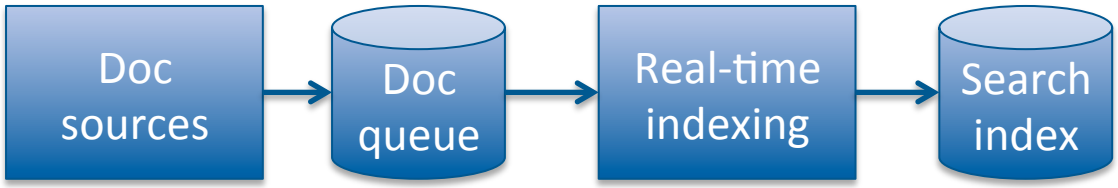
- Now searches use recent views + text
  - recent views query indicator fields
  - text queries normal text data
  - add noise as appropriate

And Draw a Picture

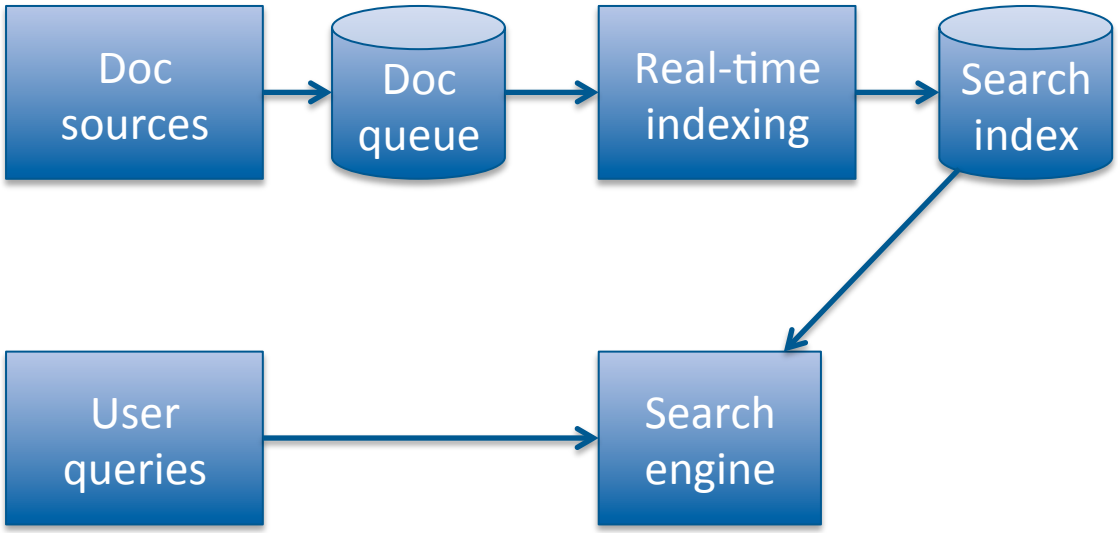
- Searches and clicks can be logged
  - real-time metrics
  - real-time trending topics
- What's hot, what's not
- Popular searches
- Document clusters
- Word clouds

In Pictures

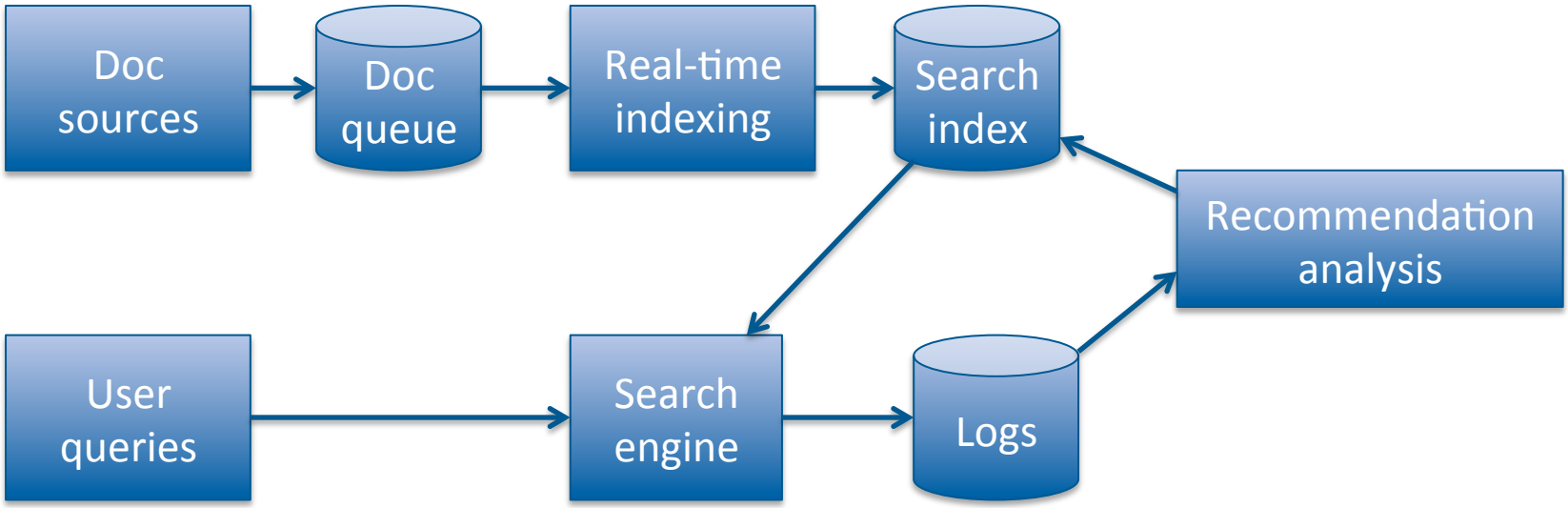
In Pictures



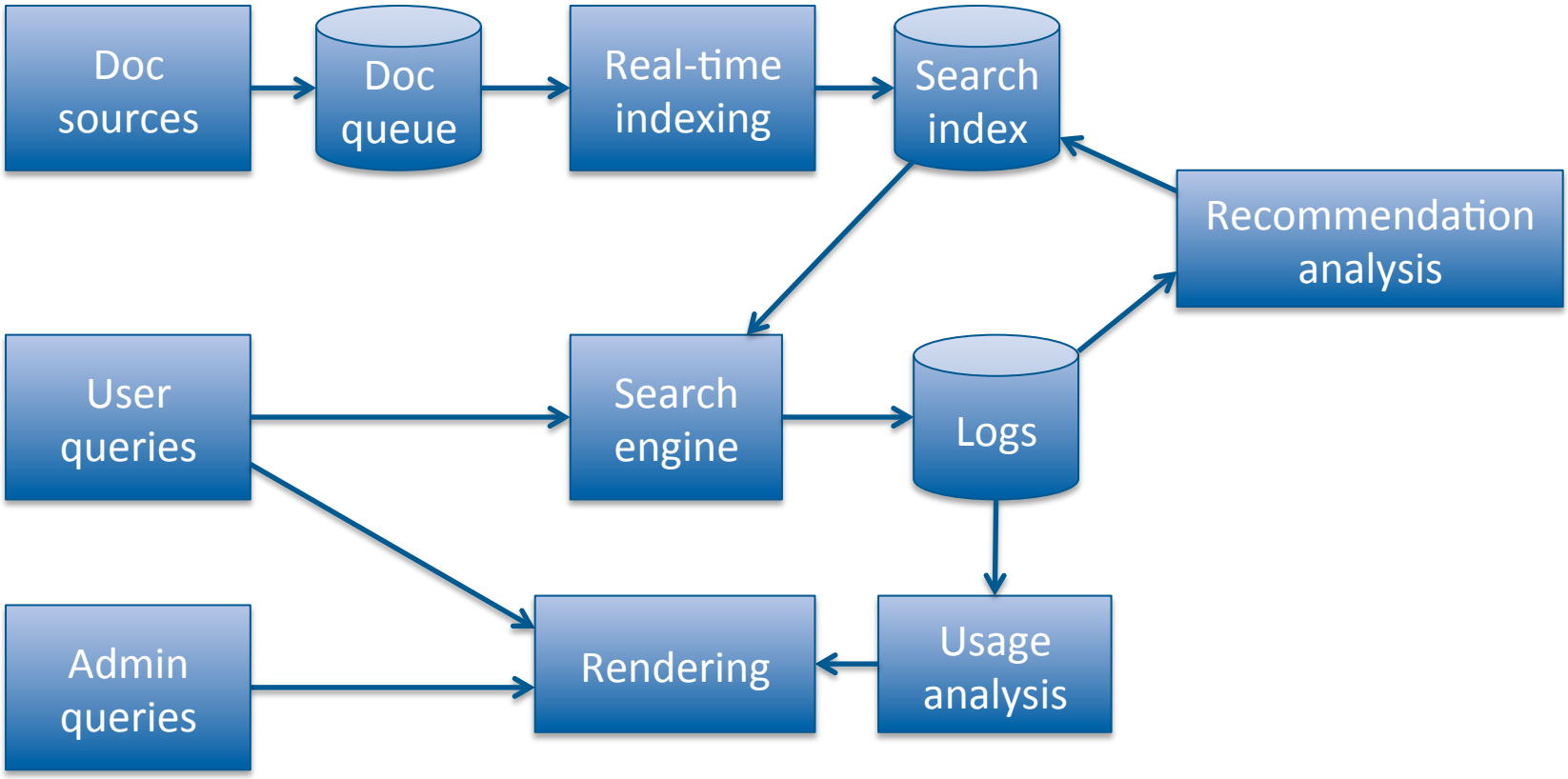
In Pictures



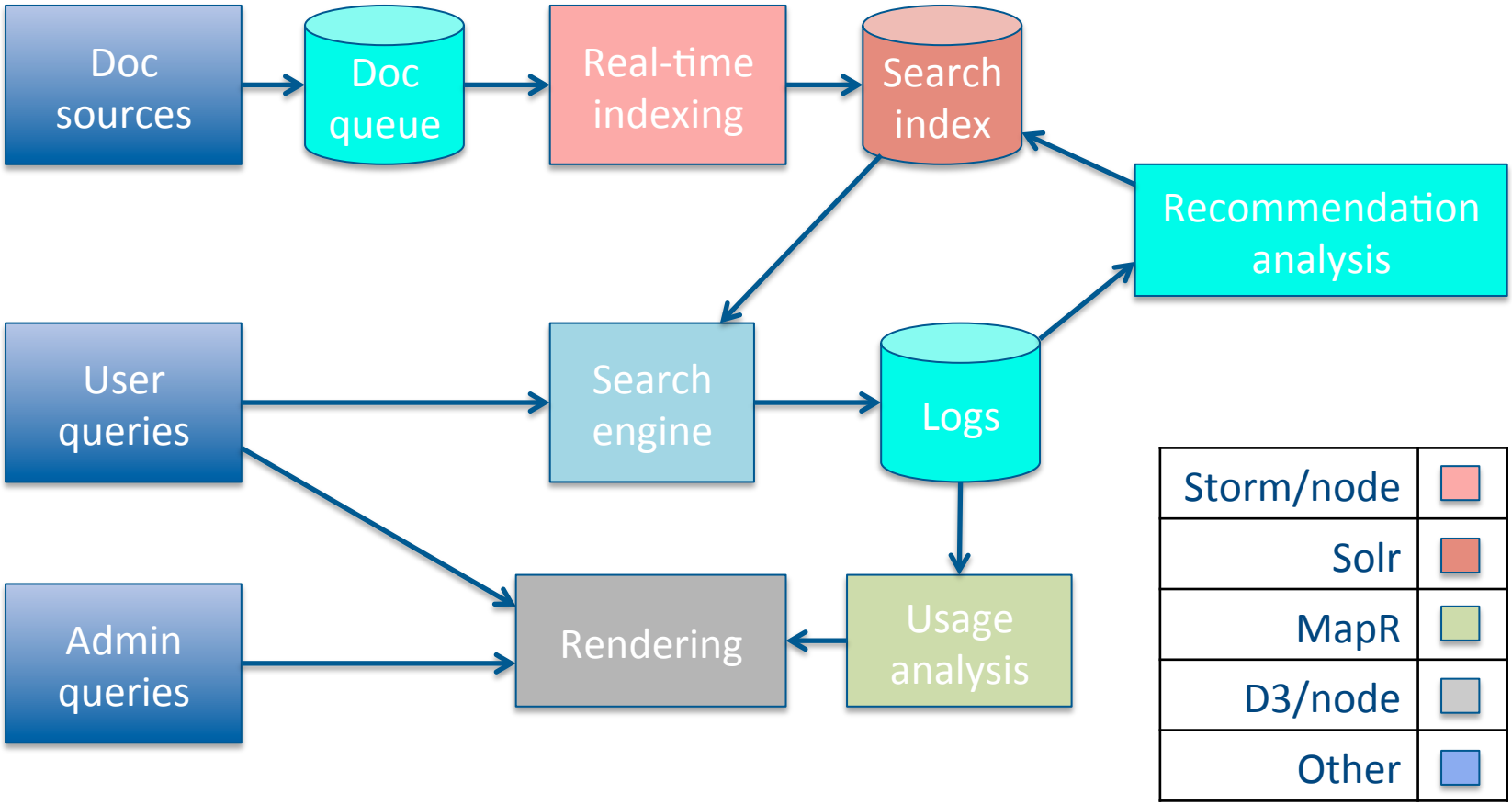
In Pictures



In Pictures



Which Technology?

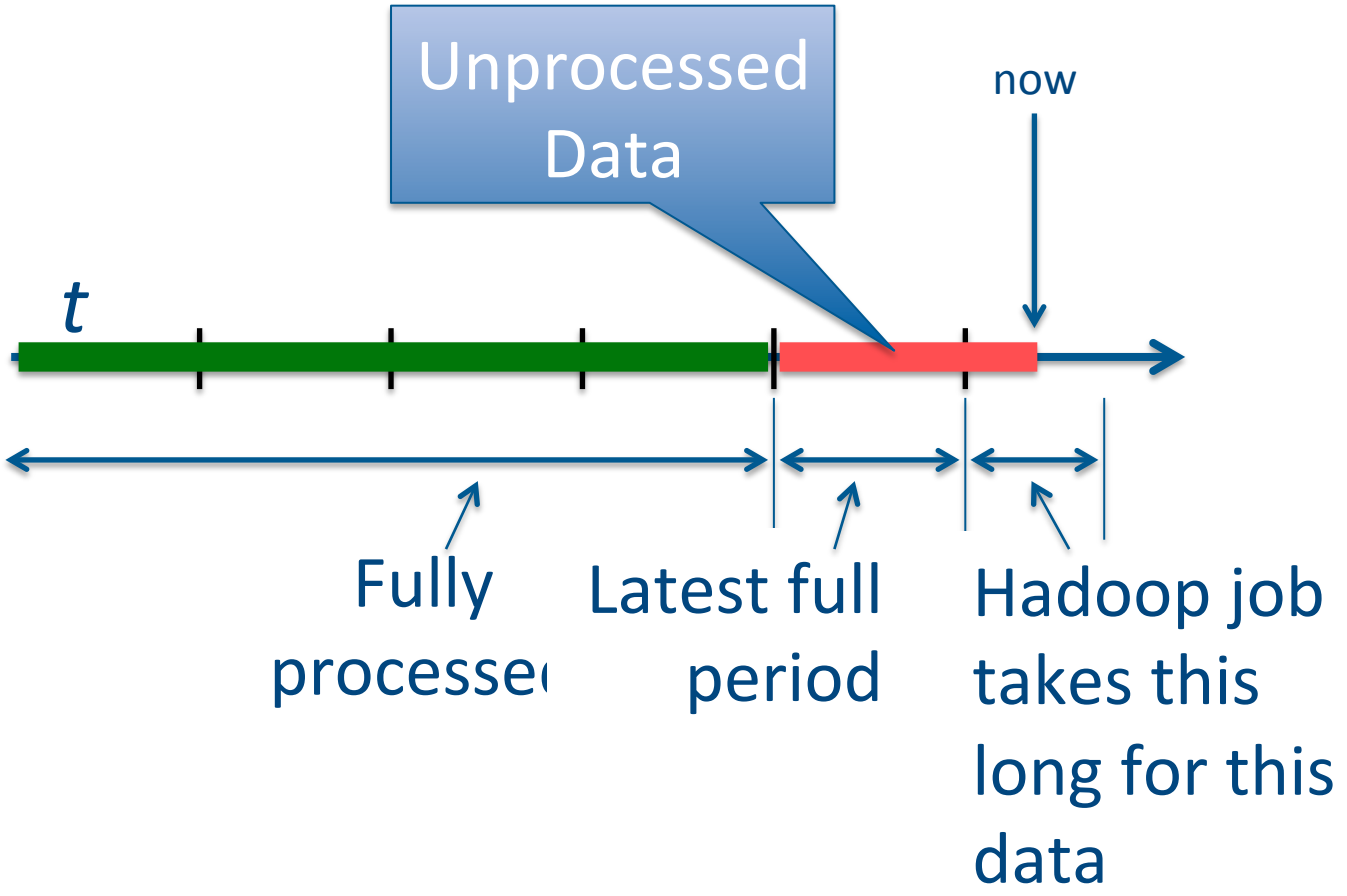


MAPR TECHNOLOGIES

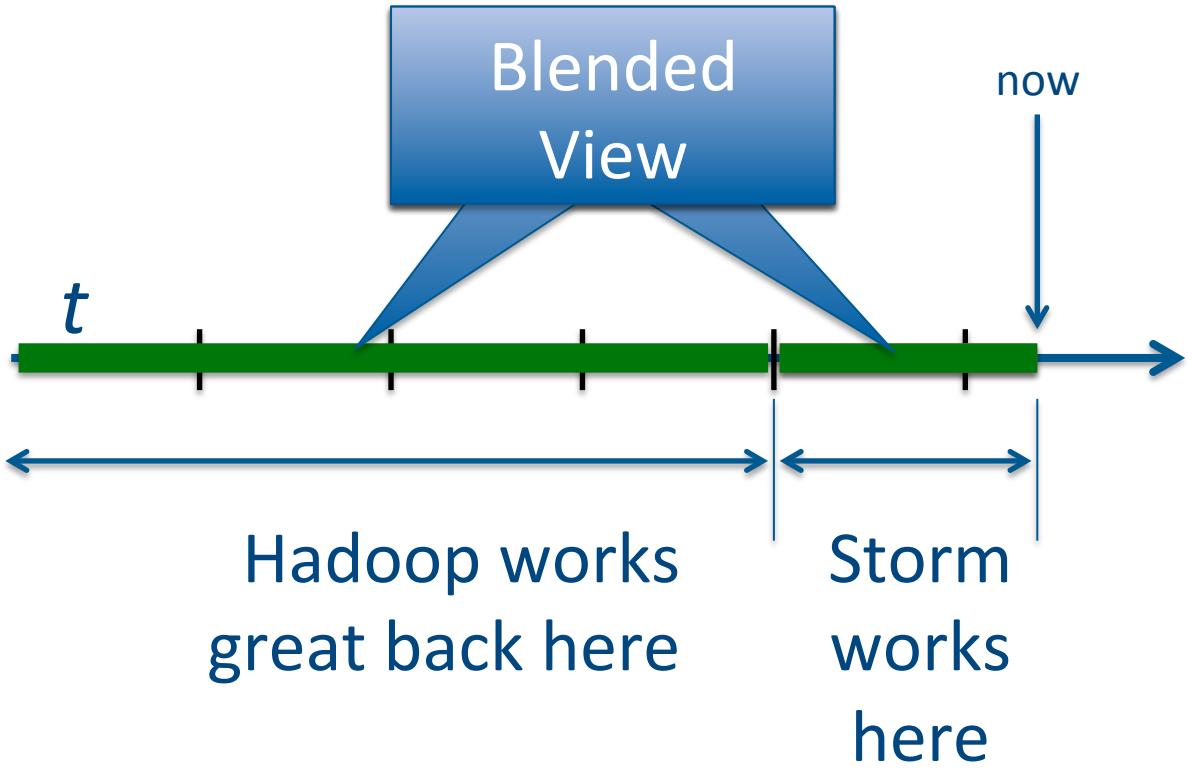
Yeah, But …

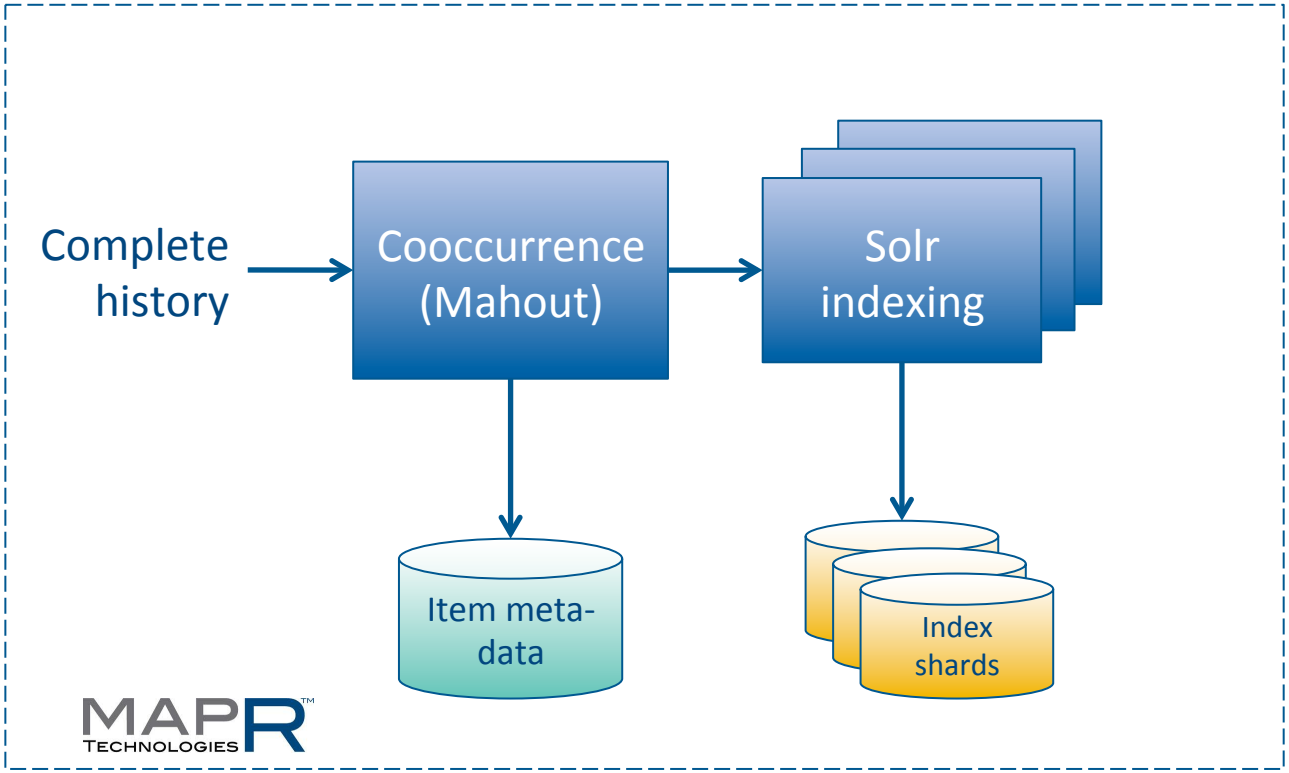
- This isn't as easy as it looks
- Take the real-time / long-time part

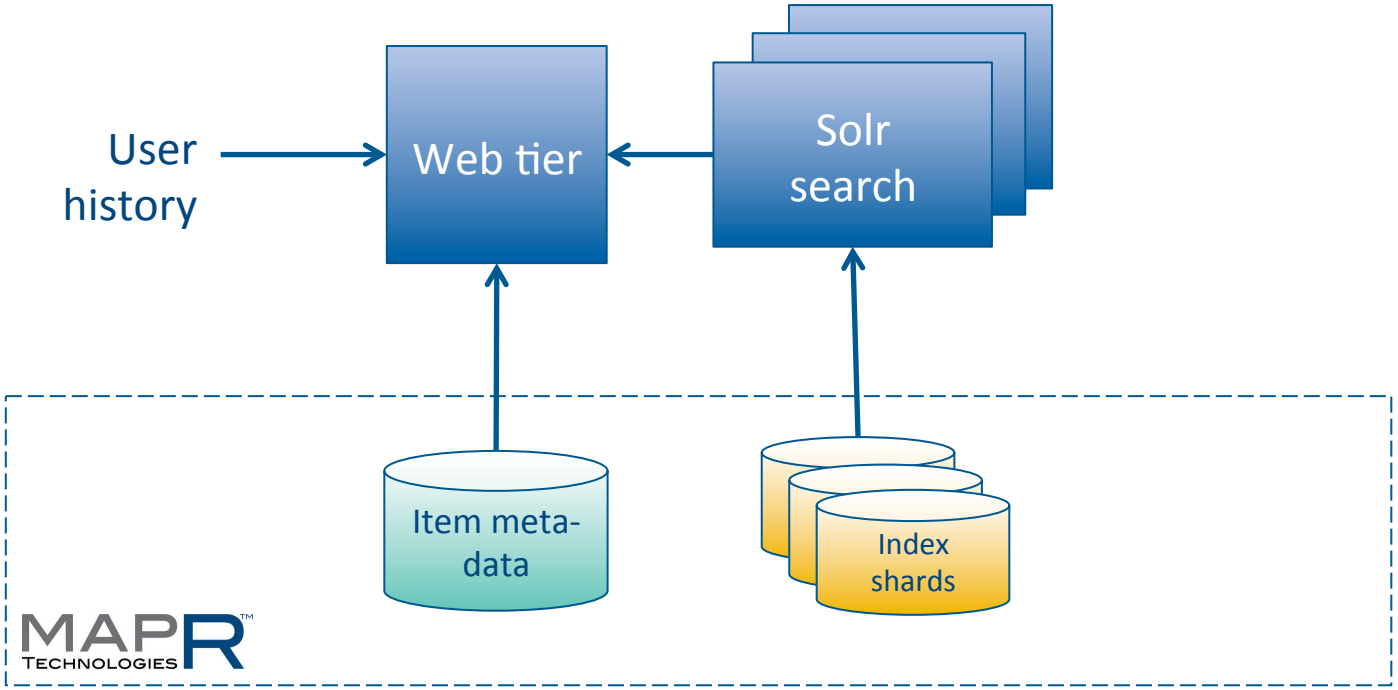
Hadoop is Not Very Real-time

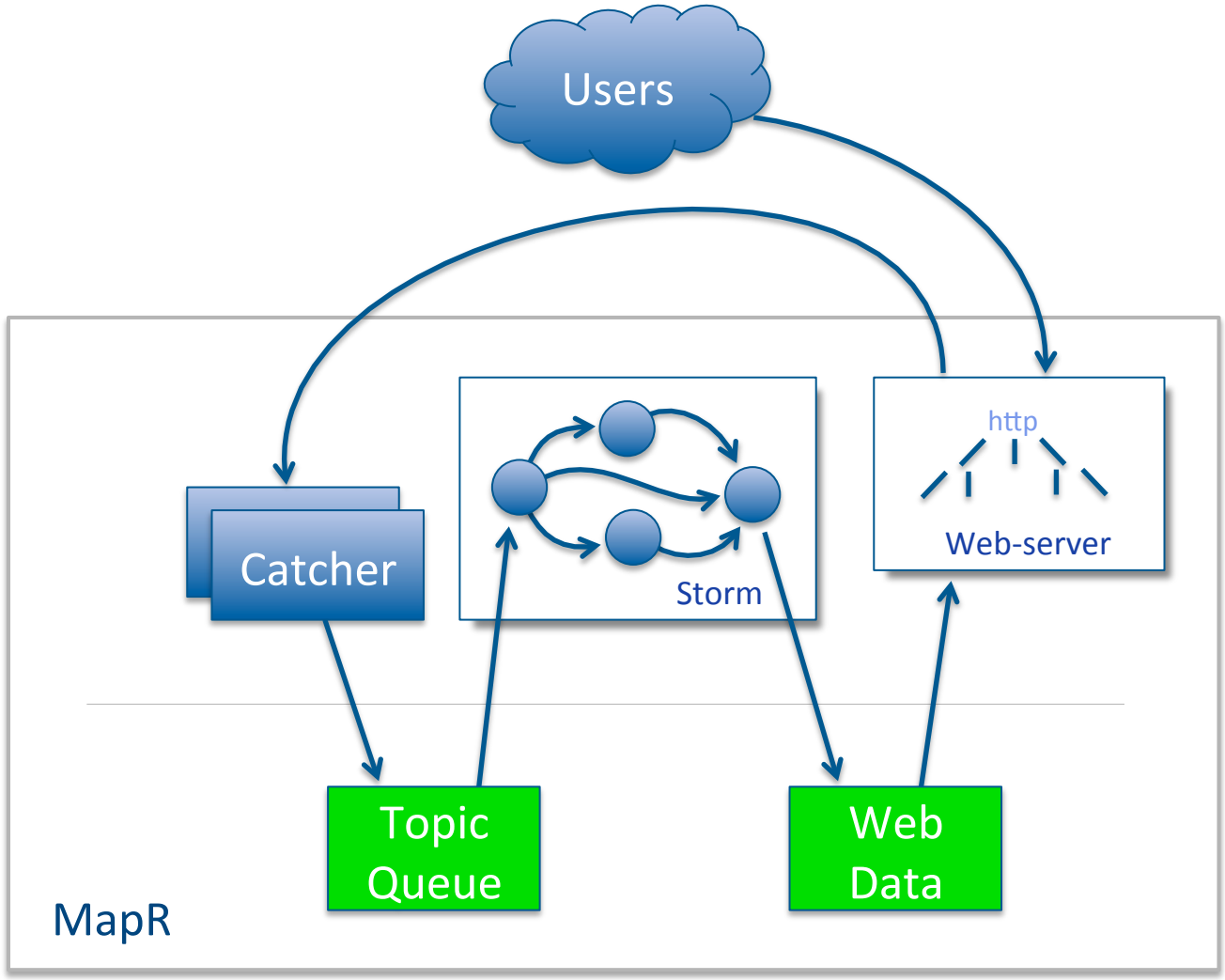


Real-time and Long-time together

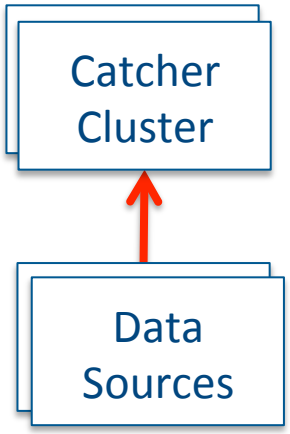



Complete history
Cooccurrence (Mahout)
Solr indexing
Item meta-data
Index shards
MAPR TECHNOLOGIES





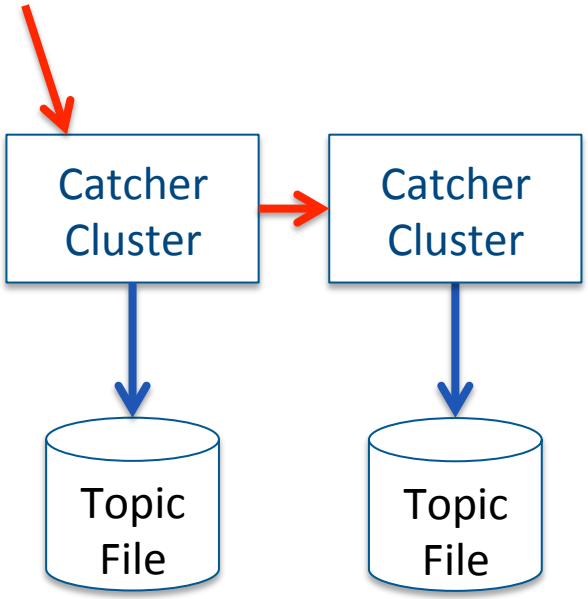
Closer Look – Catcher Protocol



The data sources and catchers communicate with a very simple protocol.

Hello() => list of catchers
Log(topic,message) =>
(OK|FAIL, redirect-to-catcher)

Closer Look – Catcher Queues

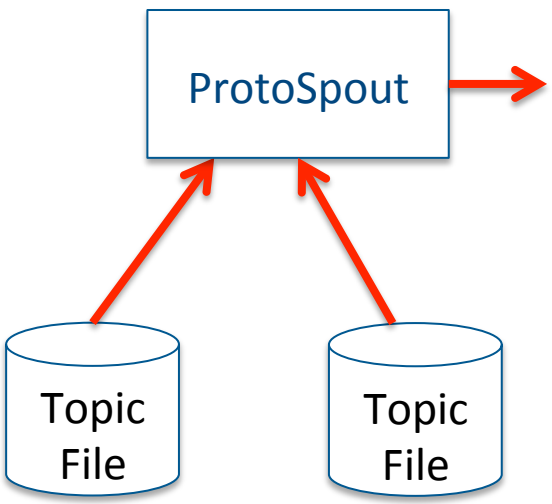


The catchers forward log requests to the correct catcher and return that host in the reply to allow the client to avoid the extra hop.

Each topic file is appended by exactly one catcher.

Topic files are kept in shared file storage.

Closer Look – ProtoSpout



The ProtoSpout tails the topic files, parses log records into tuples and injects them into the Storm topology.

Last fully acked position stored in shared file system.

Yeah, But …

- What was that about adding noise in scoring?
- Why would I do that??
- Is there a simple answer?

Thompson Sampling

- Select each shell according to the probability that it is the best

- Probability that it is the best can be computed using posterior

$$P(i \text{ is best}) = \int I\left[\mathbf{E}[r_i \mid \theta] = \max_j \mathbf{E}[r_j \mid \theta]\right] P(\theta \mid D)\, d\theta$$

- But I promised a *simple* answer

Thompson Sampling – Take 2

- Sample θ

$$\theta \sim P(\theta \mid D)$$

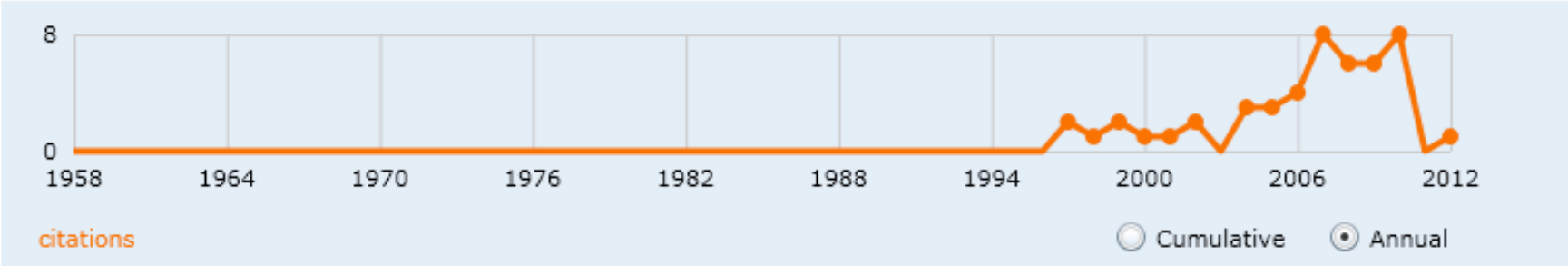
- Pick *i* to maximize reward

$$i = \underset{j}{\operatorname{argmax}}\, \mathbf{E}[r \mid \theta]$$

- Record result from using *i*

Nearly Forgotten until Recently

- Citations for Thompson sampling



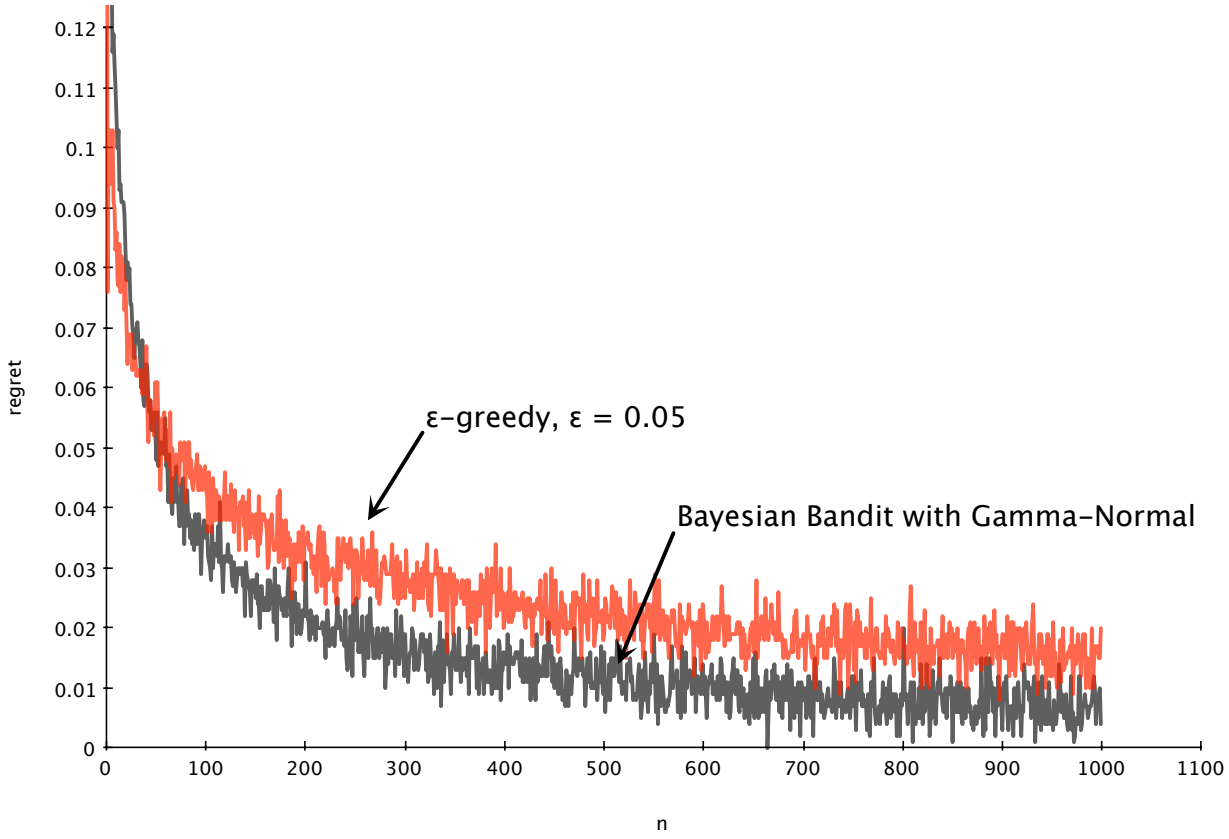
Bayesian Bandit for the Search

- Compute distributions based on data so far
- Sample scores s_1, s_2 …
  - based on actual score
  - plus per doc noise from these distributions
- Rank docs by s_i

- Lemma 1: The probability of showing doc *i* at first position will match the probability it is the best

- Lemma 2: This is as good as it gets

And it works!



Yeah, But …

- Isn’t recommendations complicated?
- How can I implement this?

Recommendation Basics

- History:

User	Thing
1	3
2	4
3	4
2	3
3	2
1	1
2	1

Recommendation Basics

- History as matrix:

	t1	t2	t3	t4
u1	1	0	1	0
u2	1	0	1	1
u3	0	1	0	1

- (t1, t3) cooccur 2 times,
- (t1, t4) once,
- (t2, t4) once,
- (t3, t4) once

A Quick Simplification

- Users who do *h*

$$\mathbf{Ah}$$

- Also do *r*

$$\mathbf{A}^T\left(\mathbf{Ah}\right)$$ User-centric recommendations

$$\left(\mathbf{A}^T\mathbf{A}\right)\mathbf{h}$$ Item-centric recommendations

Recommendation Basics

- Coocurrence

	t1	t2	t3	t4
t1	2	0	2	1
t2	0	1	0	1
t3	2	0	1	1
t4	1	1	1	2

Problems with Raw Cooccurrence

- Very popular items co-occur with everything
  - Welcome document
  - Elevator music
- That isn't interesting
  - We want ***anomalous*** cooccurrence

Recommendation Basics

- Coocurrence

	t1	t2	t3	t4
t1	2	0	2	1
t2	0	1	0	1
t3	2	0	1	1
t4	1	1	1	

	t3	*not* t3
t1	2	1
***not* t1**	1	1

Spot the Anomaly

0.44

	A	*not* A
	13	1000
	1000	100,000

0.98

	A	*not* A
	1	0
	0	2

2.26

	A	*not* A
	1	0
not B	0	10,000

7.15

	A	*not* A
	10	0
not B	0	100,000

- Root LLR is roughly like standard deviations

Root LLR Details

- In R

```
entropy = function(k) {
  -sum(k*log((k==0)+(k/sum(k))))
}
rootLLr = function(k) {
  sign = …
  sign * sqrt(
    (entropy(rowSums(k))+entropy(colSums(k))
    - entropy(k))/2)
}
```

- Like sqrt(mutual information * N/2)

  See http://bit.ly/16DvLVK

Threshold by Score

- Coocurrence

	t1	t2	t3	t4
t1	2	0	2	1
t2	0	1	0	1
t3	2	0	1	1
t4	1	1	1	2

Threshold by Score

- Significant cooccurrence => Indicators

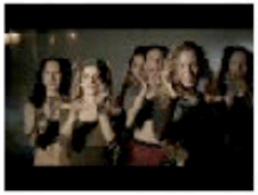
	t1	t2	t3	t4
t1	**1**	0	0	**1**
t2	0	**1**	0	**1**
t3	0	0	**1**	**1**
t4	**1**	0	0	**1**

Yeah, But …

- Why go to all this trouble?
- Does it really help?

Real-life example

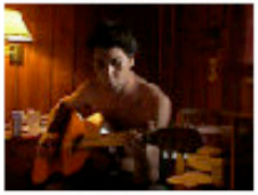
CONCIERTO CIUDAD DE LAS IDEAS PARTE FINAL
Music
58 views



Siudy / Buleria
Music
722 views

Vicente Amigo 2ª parte Ciudad de las Ideas
Music
124 views

Van Halen's Eruption
Music
4400 views



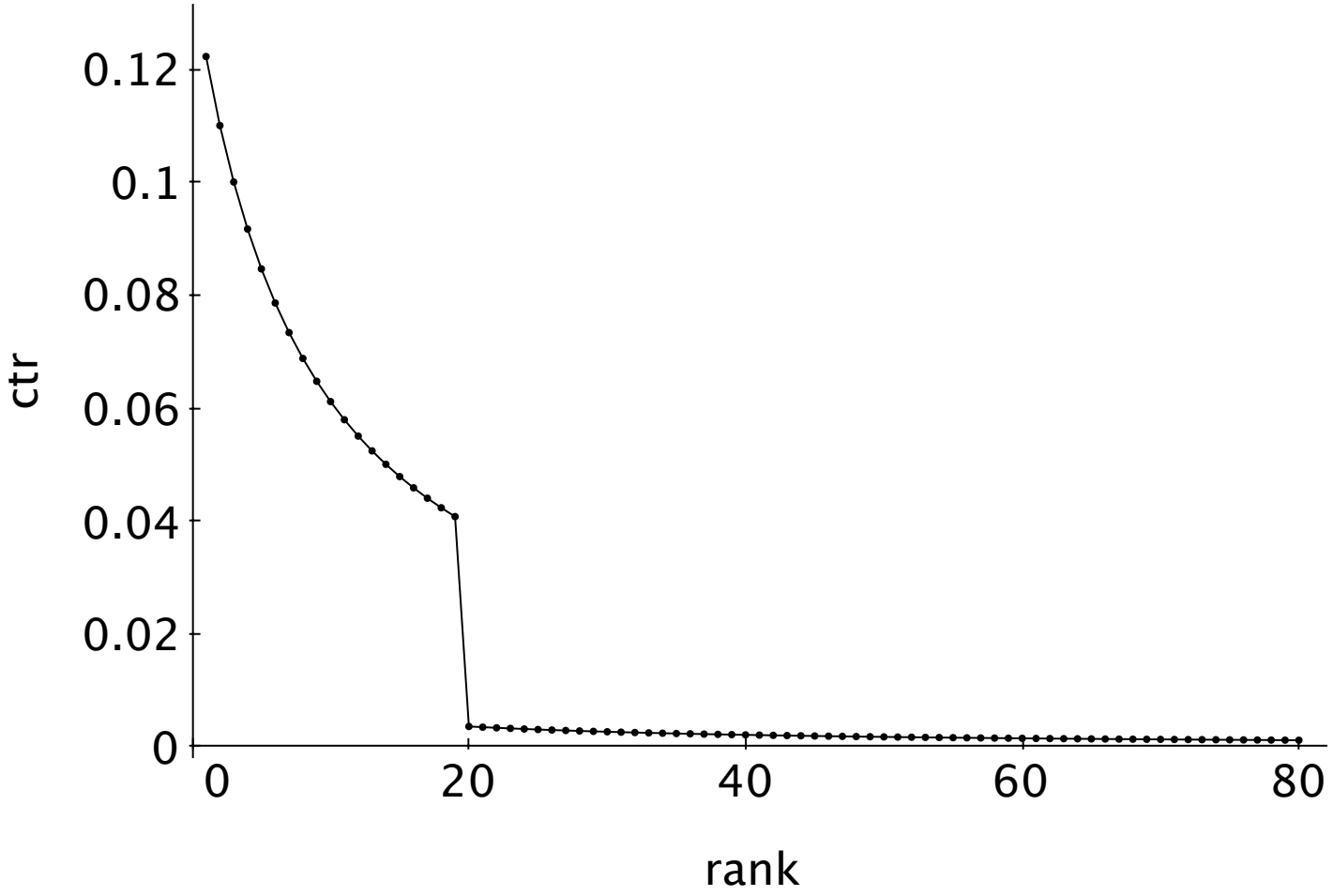
Freestyle Flamenco
Music
653 views

The Real Life Issues

- Exploration
- Diversity
- Speed

- Not the last percent

The Second Page



Make it Worse to Make It Better

- Add noise to rank

  1 2 8 7 6 3 5 4 10 13 21 18 12 9 14 24 34 28 32 17
  11 27 40 30 41 49 16 15 35 23 19 22 26 31 20 43 25 29 33 62
  38 60 74 53 36 37 39 70 45 44 46 71 42 69 47 63 52 57 51 48

- Results are worse today
- But better tomorrow

Anti-Flood

- 200 of the same result is no better than 2
- The recommender list is a portfolio of results
  - If probability of success is highly correlated, then probability of at least one success is much lower
- Suppressing items similar to higher ranking items helps

The Punchline

- Hybrid systems really can work today
- Middle tiers aren't as interesting as they used to be
  - No need for Flume … queue directly in big data system
  - No need for external queues, tail the data directly with Storm
  - No need for query systems for presentation data … read it directly with node
- Absolutely require common frameworks and standard interfaces
- You can do this today!

- Contact:

  tdunning@maprtech.com

  @ted_dunning

- Slides and such

  http://slideshare.net/tdunning

- Hash tags: #mapr #goto #d3 #node