Search and Sushi; Freshness counts

by Jo Kristian Bergum, Verizon Media
@jobergum

vespa

Copyright Verizon Media

This Talk

1. Vespa introduction
2. Searching and Ranking Over Evolving Datasets
3. Vespa Real Time Indexing Architecture

Photo by Vinicius Benedit on Unsplash

Copyright Verizon Media

@vespaengine

Vespa.ai

A open-source platform for low latency computations over large, evolving data

Apache 2.0 Licensed

- ➔ Search,filter and rank structured and unstructured data
- ➔ Vector search (ANN)
- ➔ Scalable and Fast
- ➔ Advanced multiphase ranking with tensors as first class citizens
- ➔ Built-in support for importing machine learning models (TensorFlow, PyTorch,ONNX, XGBoost, LightGBM ++)
- ➔ Real time indexing and true partial updates

vespa

Copyright Verizon Media

The history of Vespa.ai

1998	2003	2004	2010	2017	2021
Alltheweb.com	**Yahoo**	**Vespa is born**	**Vespa 5.x**	**Vespa open sourced**	**Vespa Cloud**
Fast Search and Transfer Web search division	Yahoo buys Inktomi, Overture, Altavista and web division of Fast Search Transfer	Vertical Search Platform in Yahoo. Batch incremental indexing	New real time indexing	Apache 2.0 License. All development in the open at github.com/vespa-engine	Vespa Cloud released

vespa

Copyright Verizon Media

Vespa @ Yahoo/Verizon Media

Serve 25B real time queries per day, 75B writes (updates)

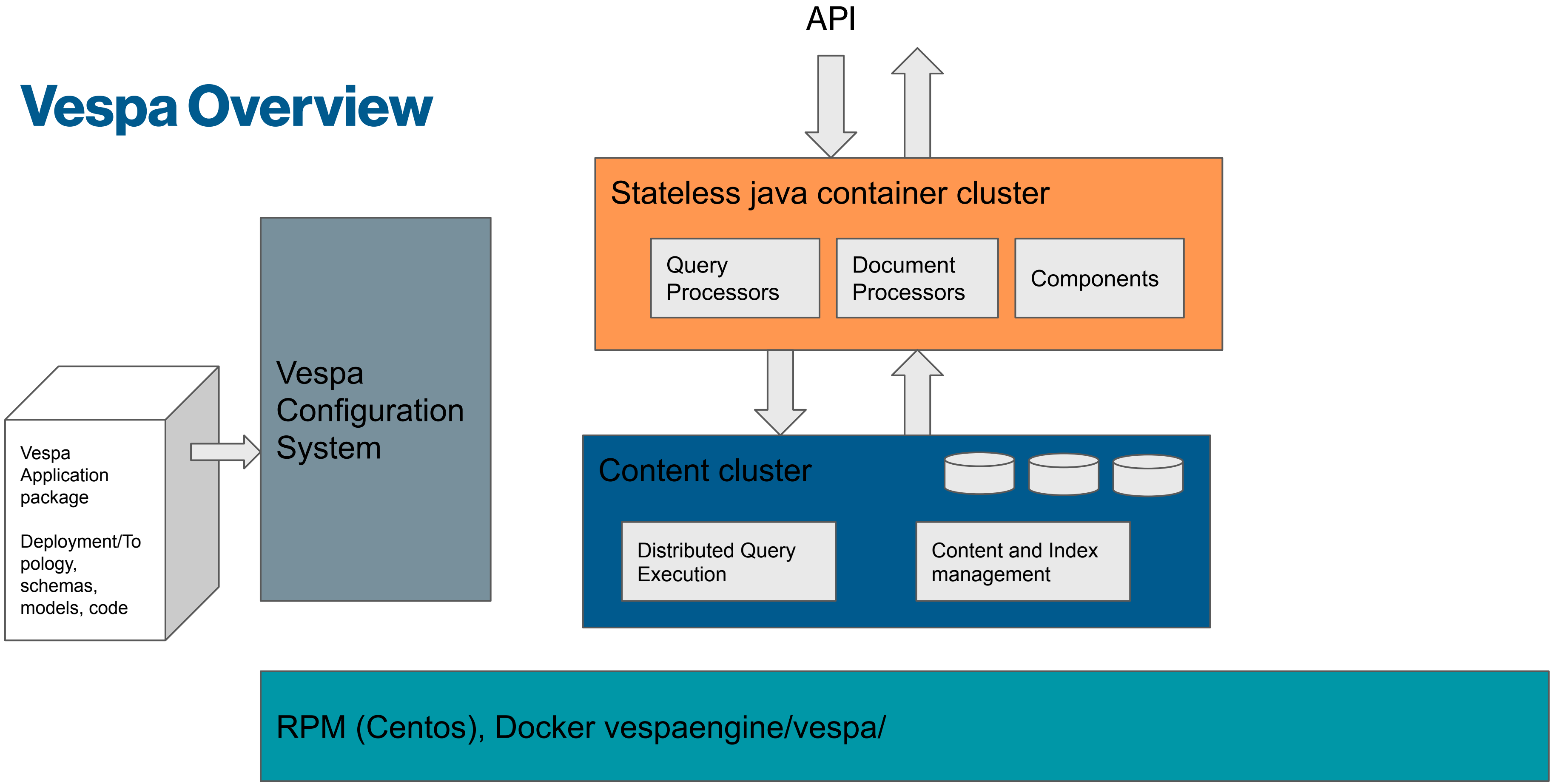
150+ different applications.

- Gemini Native Ads, Yahoo home page, Local search, News, Finance. Yahoo shopping, Gemini Product Ads and more..

Check blog.vespa.ai for some interesting use cases, tensor ranking for home page recommendations for example.

Copyright Verizon Media

Vespa Overview


API
Stateless java container cluster
Query Processors
Document Processors
Components
Vespa Configuration System
Vespa Application package
Deployment/Topology, schemas, models, code
Content cluster
Distributed Query Execution
Content and Index management
RPM (Centos), Docker vespaengine/vespa/

vespa

Copyright Verizon Media

Searching and Ranking over Evolving datasets

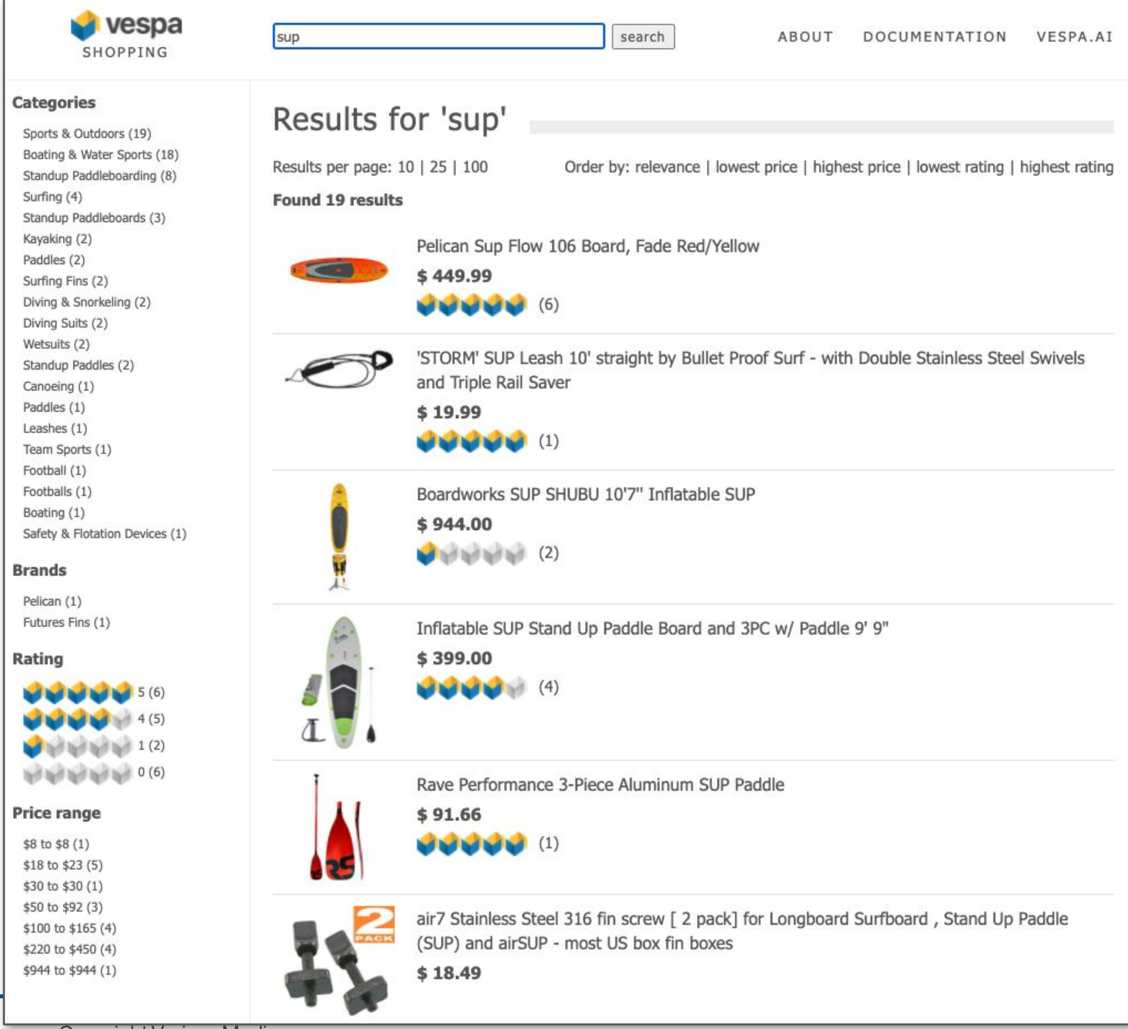
CRUD

Hard filters

- In-stock
- Price
- Rating

Soft filters (ranking)

- Click feedback
- Price



vespa SHOPPING

sup search

ABOUT DOCUMENTATION VESPA.AI

Categories

Sports & Outdoors (19)
Boating & Water Sports (18)
Standup Paddleboarding (8)
Surfing (4)
Standup Paddleboards (3)
Kayaking (2)
Paddles (2)
Surfing Fins (2)
Diving & Snorkeling (2)
Diving Suits (2)
Wetsuits (2)
Standup Paddles (2)
Canoeing (1)
Paddles (1)
Leashes (1)
Team Sports (1)
Football (1)
Footballs (1)
Boating (1)
Safety & Flotation Devices (1)

Brands

Pelican (1)
Futures Fins (1)

Rating

5 (6)
4 (5)
1 (2)
0 (6)

Price range

$8 to $8 (1)
$18 to $23 (5)
$30 to $30 (1)
$50 to $92 (3)
$100 to $165 (4)
$220 to $450 (4)
$944 to $944 (1)

Results for 'sup'

Results per page: 10 | 25 | 100   Order by: relevance | lowest price | highest price | lowest rating | highest rating

Found 19 results

Pelican Sup Flow 106 Board, Fade Red/Yellow
$ 449.99
(6)

'STORM' SUP Leash 10' straight by Bullet Proof Surf - with Double Stainless Steel Swivels and Triple Rail Saver
$ 19.99
(1)

Boardworks SUP SHUBU 10'7" Inflatable SUP
$ 944.00
(2)

Inflatable SUP Stand Up Paddle Board and 3PC w/ Paddle 9' 9"
$ 399.00
(4)

Rave Performance 3-Piece Aluminum SUP Paddle
$ 91.66
(1)

air7 Stainless Steel 316 fin screw [2 pack] for Longboard Surfboard , Stand Up Paddle (SUP) and airSUP - most US box fin boxes
$ 18.49

Copyright Verizon Media

Real Time Indexing Architecture in Vespa, and some history

Copyright Verizon Media

Real time Indexing Architecture in Vespa

High Level Goals

- Low latency (ms)
- Visible for reads when operation is acknowledged (reply)
- High throughput
- Low impact on search serving latency

Copyright Verizon Media

Classic inverted index data structure

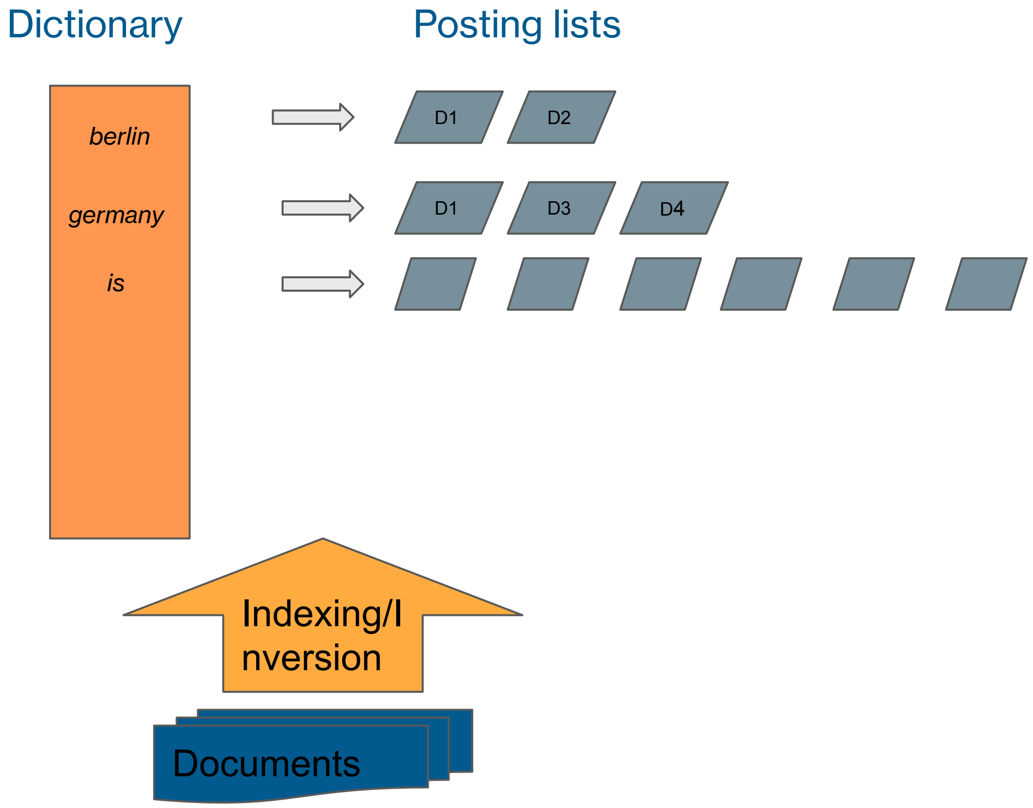
Data structure for efficient query evaluation:

text:berlin OR text:germany

text:berlin AND text:germany

text:"berlin germany"

And more...



Copyright Verizon Media

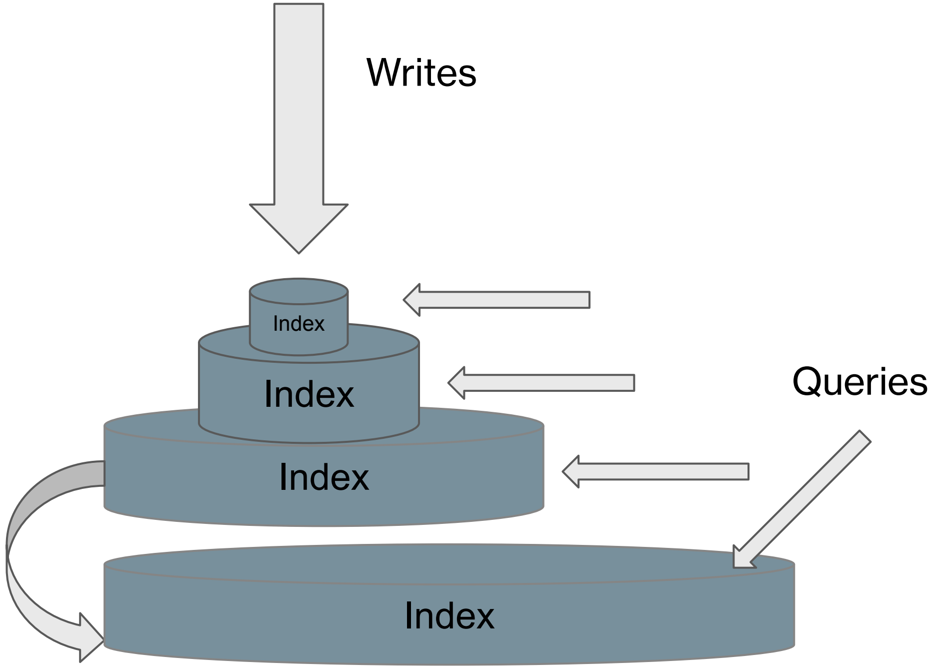
Near Real time indexing in Vespa - take 1 (2004)

Batch immutable index segments

Operations visible after some time after operation return

Queries fans out to all active indexes

Need to merge or fuse indexes in background to keep number of active indexes low



What if we can use more memory?

- $1,000 $/GB in 2000
- 10$/GB in 2010

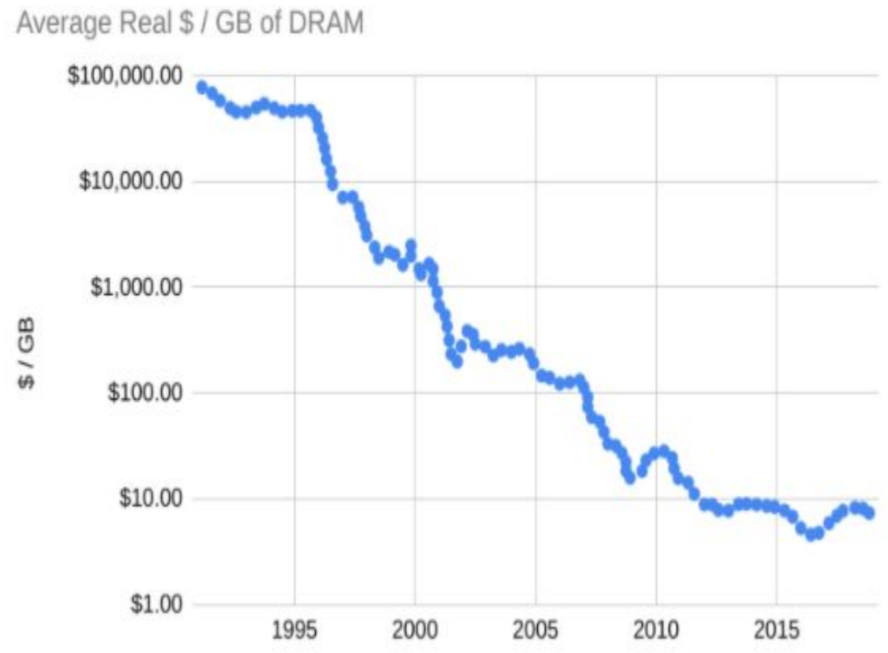


Figure 2: Average $ / GB of DRAM from 1991 to 2019 according to Objective Analysis. Dollars are 2020 dollars.

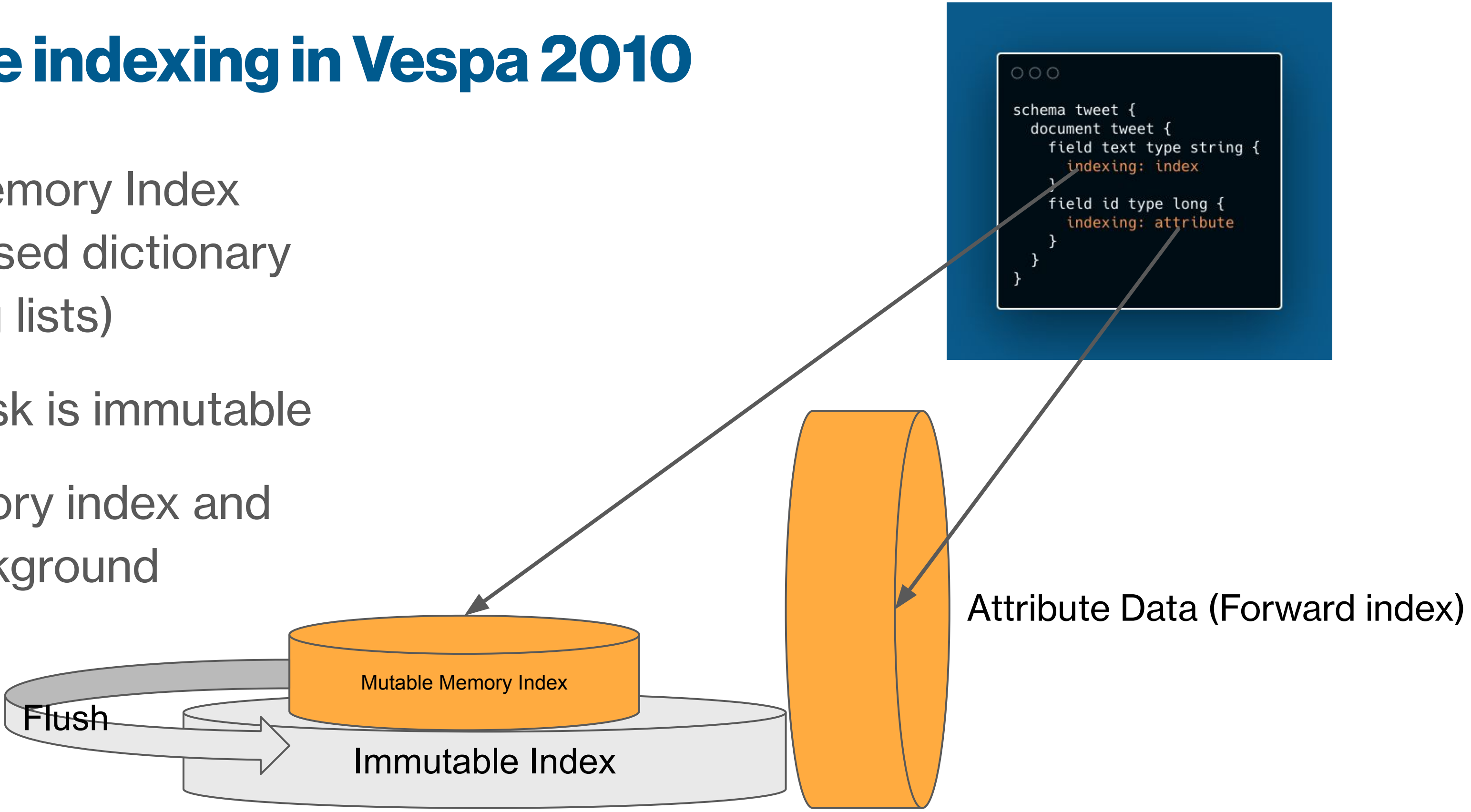
Copyright Verizon Media

Real time indexing in Vespa 2010

Mutable Memory Index (B+ tree based dictionary and posting lists)

Index on disk is immutable

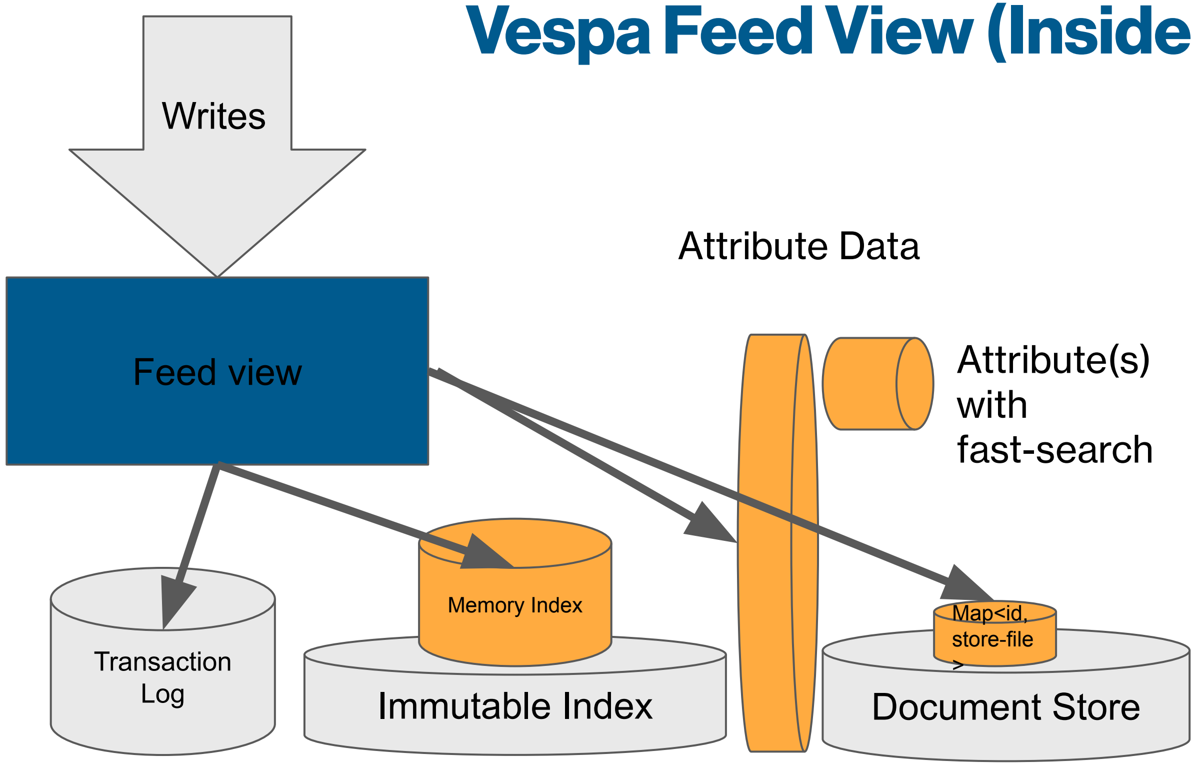
Flush memory index and fuse in background



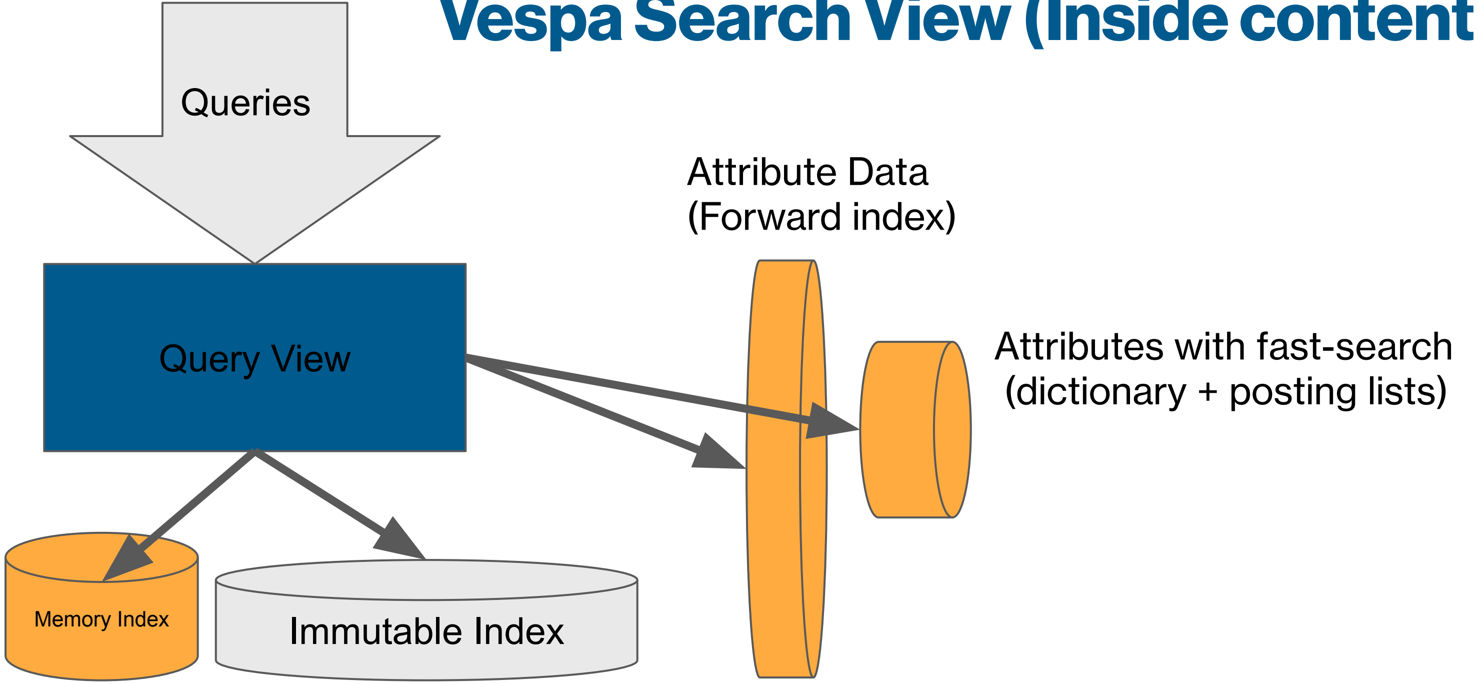
```
schema tweet {
  document tweet {
    field text type string {
      indexing: index
    }
    field id type long {
      indexing: attribute
    }
  }
}
```

Copyright Verizon Media

Vespa Feed View (Inside content node)



Vespa Search View (Inside content node)



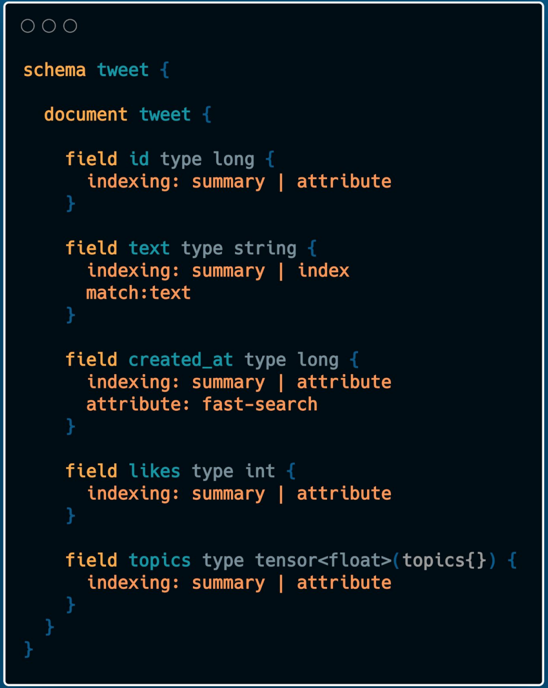
Copyright Verizon Media

Vespa Schema Language

Vespa is strongly typed

Schema declares field, types, matching

Key distinction index versus attribute



```
schema tweet {

  document tweet {

    field id type long {
      indexing: summary | attribute
    }

    field text type string {
      indexing: summary | index
      match:text
    }

    field created_at type long {
      indexing: summary | attribute
      attribute: fast-search
    }

    field likes type int {
      indexing: summary | attribute
    }

    field topics type tensor<float>(topics{}) {
      indexing: summary | attribute
    }
  }
}
```

Copyright Verizon Media

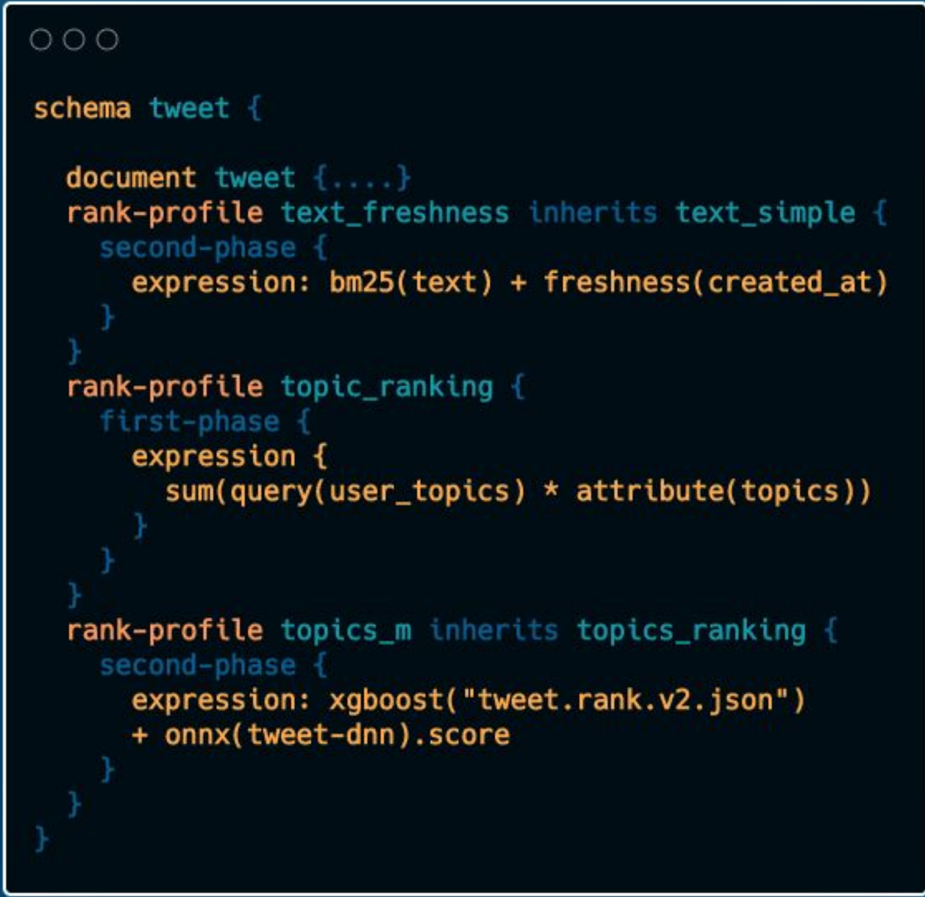
Vespa Ranking

Flexible ranking framework

Read attribute values (scalar, multi-valued or tensors) and use in ranking expression

Easy to express custom ranking logic

ML first class citizen

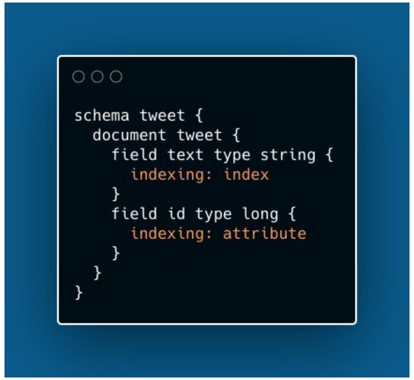


```
schema tweet {

  document tweet {....}
  rank-profile text_freshness inherits text_simple {
    second-phase {
      expression: bm25(text) + freshness(created_at)
    }
  }
  rank-profile topic_ranking {
    first-phase {
      expression {
        sum(query(user_topics) * attribute(topics))
      }
    }
  }
  rank-profile topics_m inherits topics_ranking {
    second-phase {
      expression: xgboost("tweet.rank.v2.json")
      + onnx(tweet-dnn).score
    }
  }
}
```

Copyright Verizon Media

Attribute versus index


```
schema tweet {
  document tweet {
    field text type string {
      indexing: index
    }
    field id type long {
      indexing: attribute
    }
  }
}
```

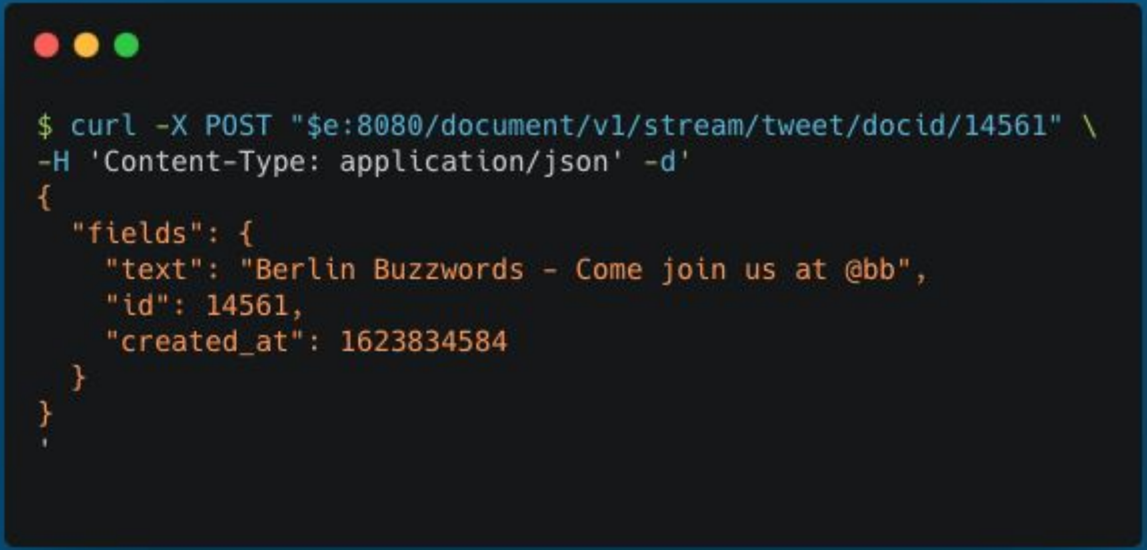
Indexing	Document store read for partial update	Fast matching using inverted index	Ranking
index	Y	Y	Y (text ranking)
attribute	N	Y (with fast-search)	Y

Copyright Verizon Media

API Examples

Copyright Verizon Media

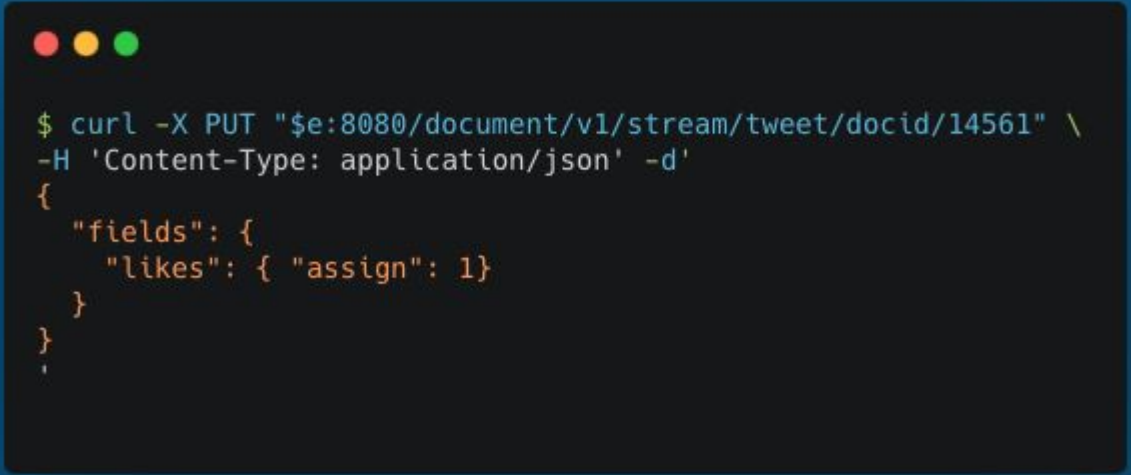
Create Document



```
$ curl -X POST "$e:8080/document/v1/stream/tweet/docid/14561" \
-H 'Content-Type: application/json' -d'
{
  "fields": {
    "text": "Berlin Buzzwords - Come join us at @bb",
    "id": 14561,
    "created_at": 1623834584
  }
}
'
```

Copyright Verizon Media

Update Document

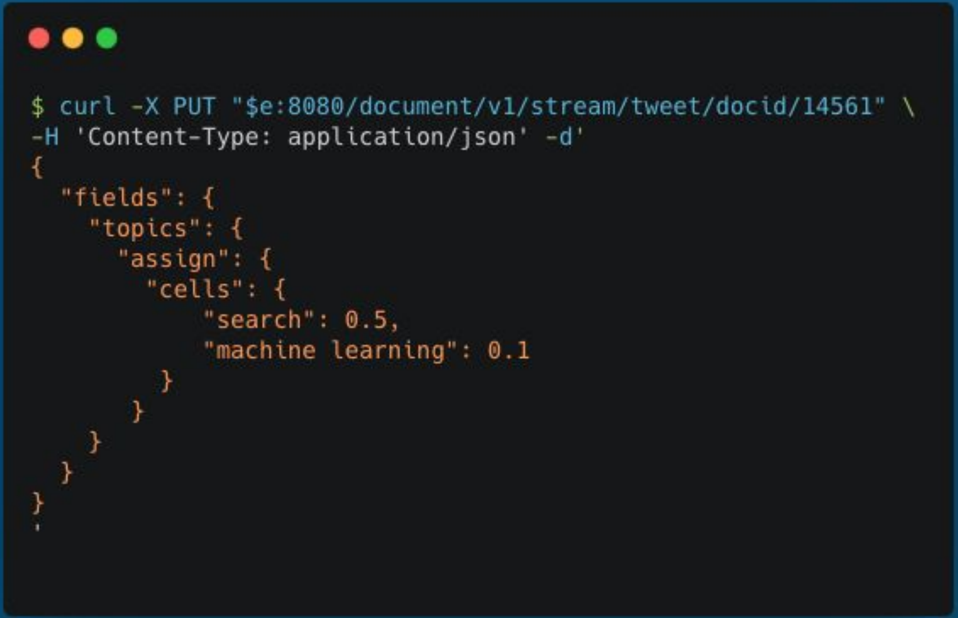


```
$ curl -X PUT "$e:8080/document/v1/stream/tweet/docid/14561" \
-H 'Content-Type: application/json' -d'
{
  "fields": {
    "likes": { "assign": 1}
  }
}
'
```

Copyright Verizon Media

Update Document

Assign topics tensor

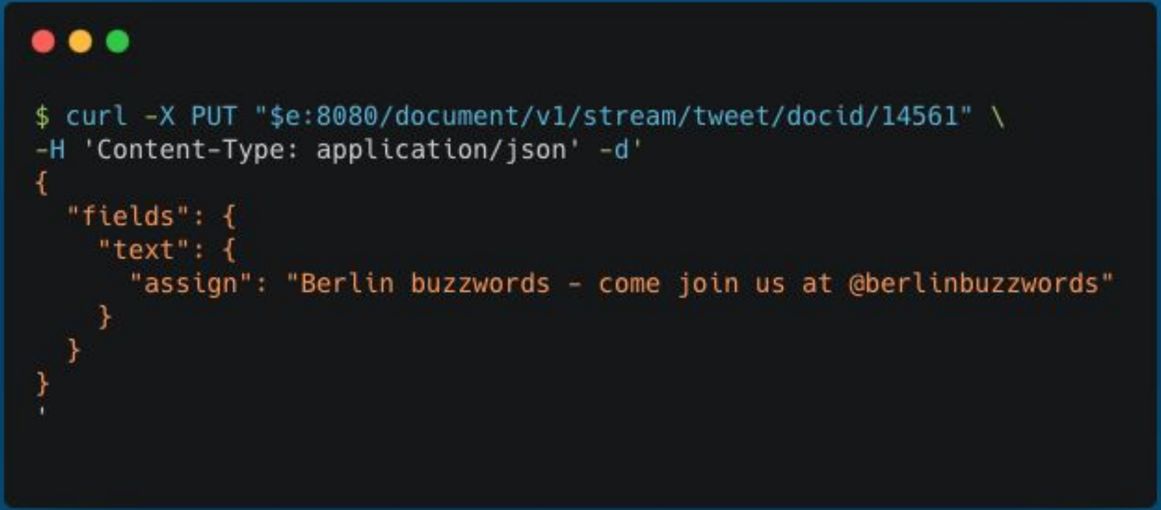


```
$ curl -X PUT "$e:8080/document/v1/stream/tweet/docid/14561" \
-H 'Content-Type: application/json' -d'
{
  "fields": {
    "topics": {
      "assign": {
        "cells": {
            "search": 0.5,
            "machine learning": 0.1
        }
      }
    }
  }
}
'
```

Copyright Verizon Media

Update Document

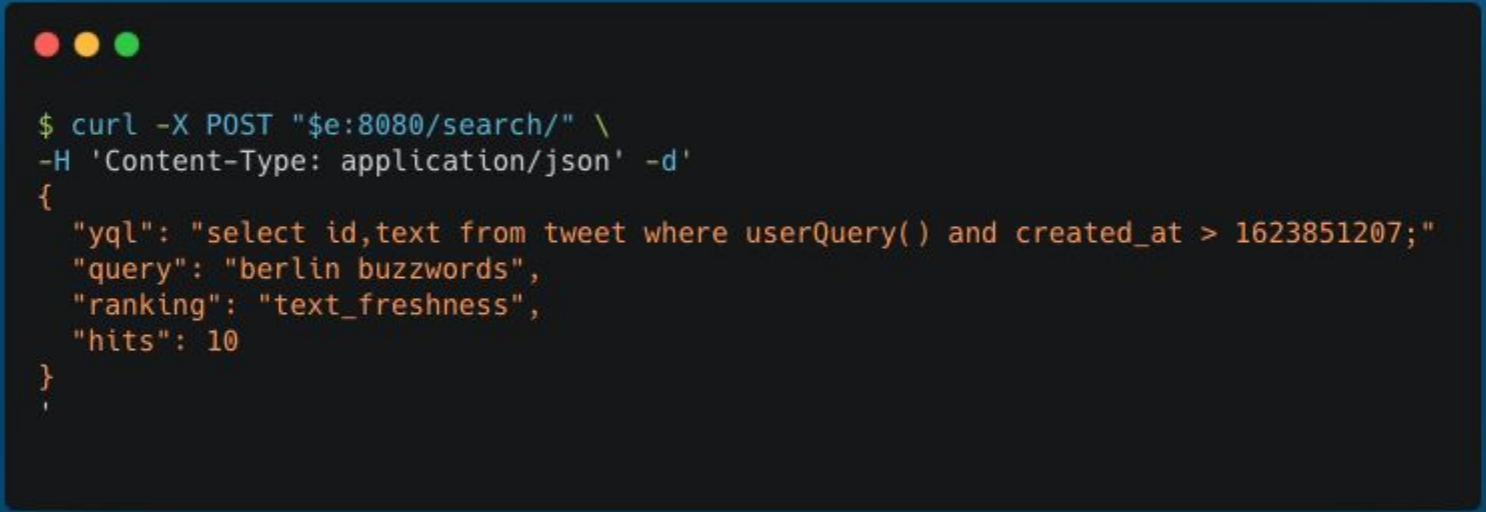
Update of string index field causes read/write pattern



```
$ curl -X PUT "$e:8080/document/v1/stream/tweet/docid/14561" \
-H 'Content-Type: application/json' -d'
{
  "fields": {
    "text": {
      "assign": "Berlin buzzwords - come join us at @berlinbuzzwords"
    }
  }
}
'
```

vespa

Query the tweet index



```
$ curl -X POST "$e:8080/search/" \
-H 'Content-Type: application/json' -d'
{
  "yql": "select id,text from tweet where userQuery() and created_at > 1623851207;"
  "query": "berlin buzzwords",
  "ranking": "text_freshness",
  "hits": 10
}
'
```

Copyright Verizon Media

Performance

Partial updates of numeric (int) attribute fields

- Single digit ms latency
- 50K updates/s per node (8-cpu 16Gb RAM, with 20% util for feed)

Note:

Append to transaction log requires high IO write capacity or transaction log synch false (Default synch operations to storage for durability)

Put against memory index (from 1K to 8K depending on size of text, token distribution and concurrency settings)

Copyright Verizon Media

Thank you

Vespa open source	https://vespa.ai/
Vespa slack space	http://slack.vespa.ai/
Vespa cloud free trial	https://cloud.vespa.ai/pricing#free-trial
Twitter	https://twitter.com/vespaengine
Github (Apache 2.0)	https://github.com/vespa-engine/

Debate: Which Search Engine? Tomorrow at Berlin Buzzwords

Copyright Verizon Media