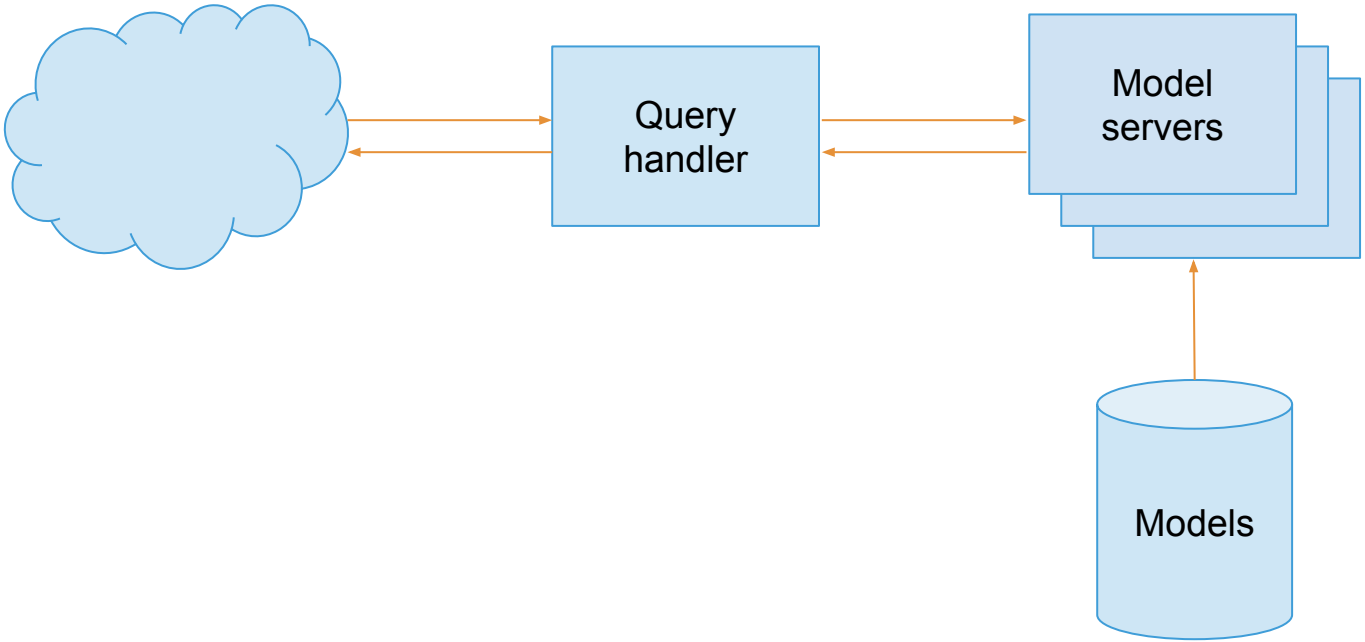
vespa.ai

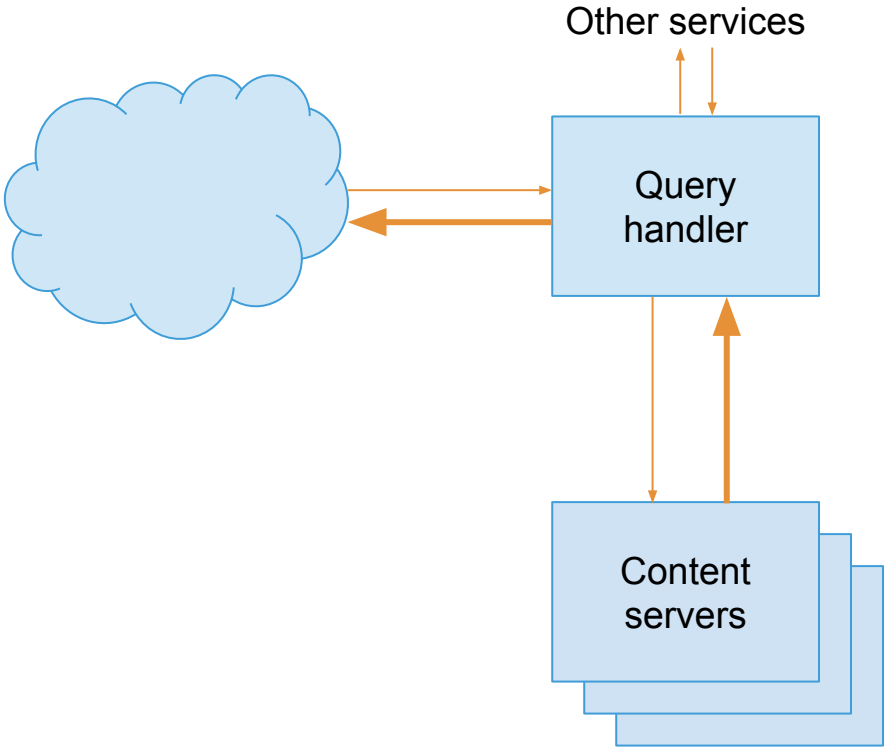
Scaling ONNX, TensorFlow and XGBoost model evaluation in search

Lester Solbakken | June 17th 2019

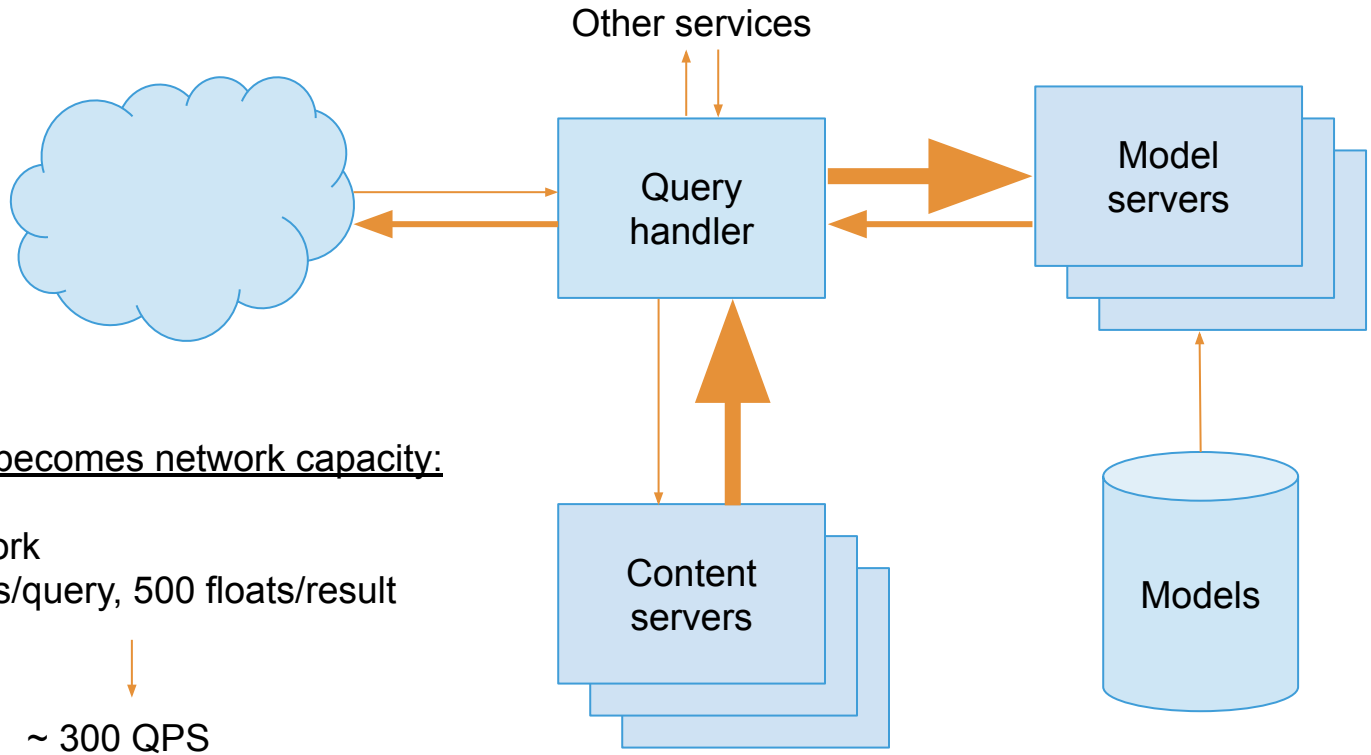
ML models in classification, transformation, ++



ML models in search, personalization, ads ++



ML models in search, personalization, ads ++

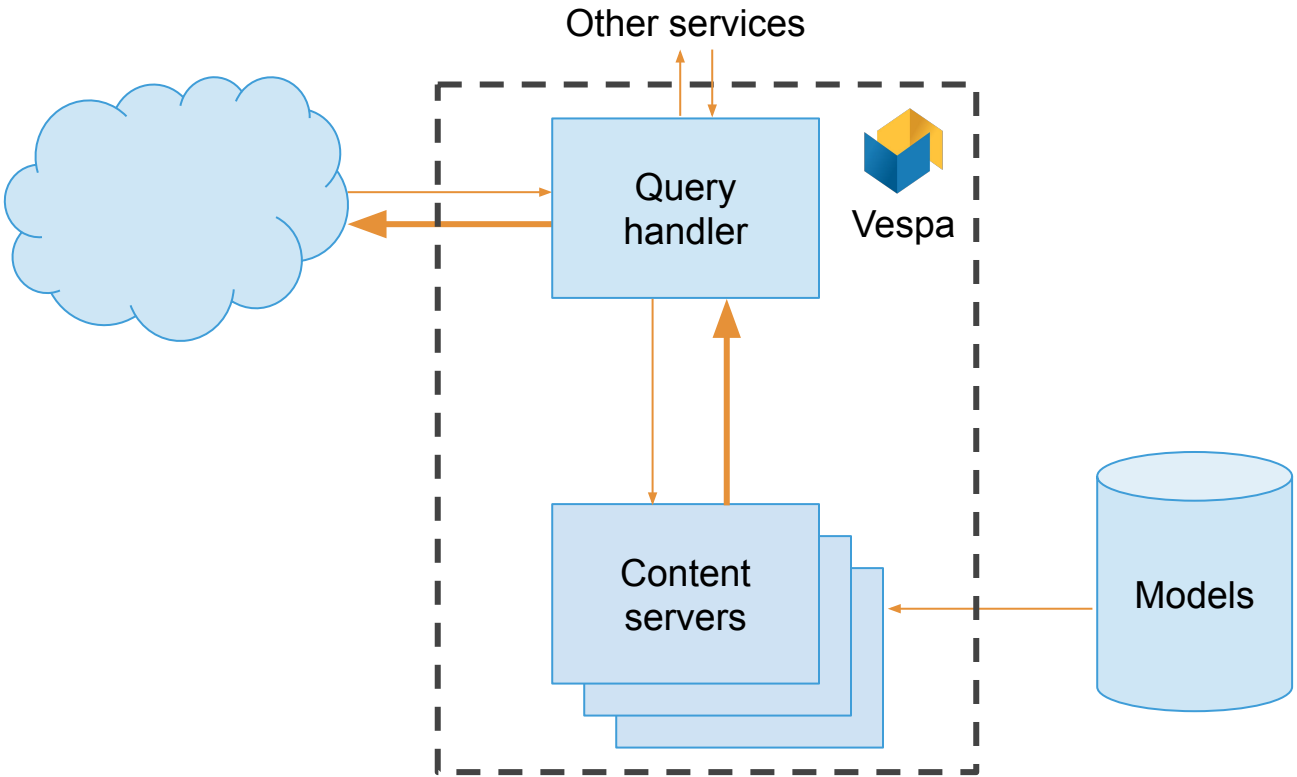


Bottleneck becomes network capacity:

10Gb network
1000 results/query, 500 floats/result

~ 300 QPS

ML models in search, personalization, ads ++



verizon media

Oath: A Verizon company

YAHOO!

TechCrunch

HUFFPOSTI

RYOT

MAKERS

tumblr.

#BUILTBYGIRLS

BUILD STUDIO

engadget

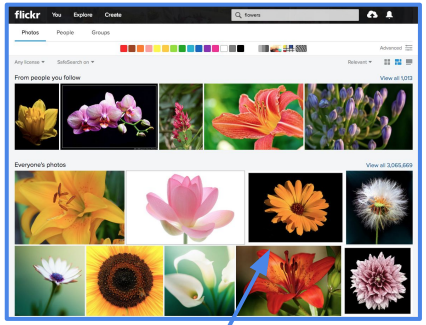
FLURRY

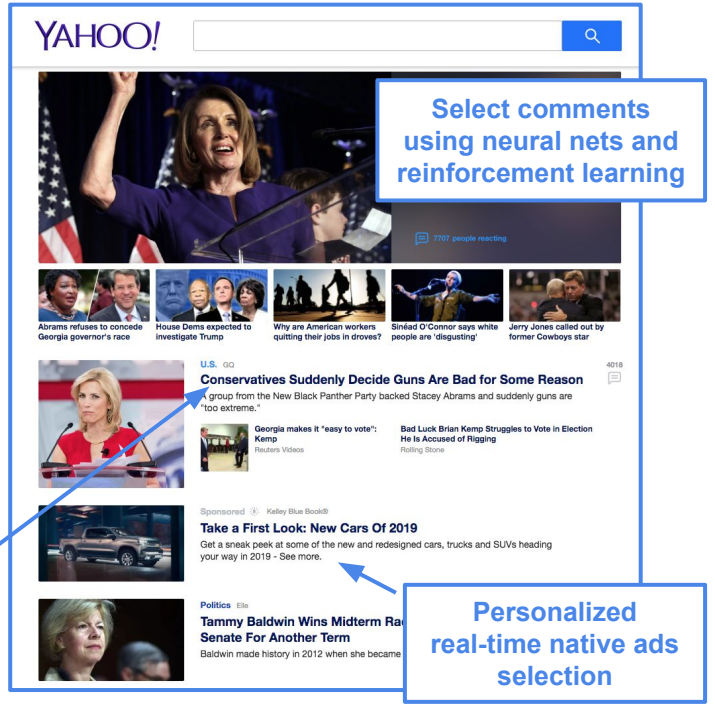
autoblog

Vespa at verizon media

Hundreds of Vespa applications

- serving over a billion users,
- hundreds of thousands of queries per second,
- billions of content items.





Vespa - core features

- Search and filter over structured and unstructured data
- Query time organization and aggregation of matching data
- Real-time writes
- Advanced relevance scoring
- Scaleable and fast
- Elastic and fault tolerant
- Pluggable
- Easy to operate

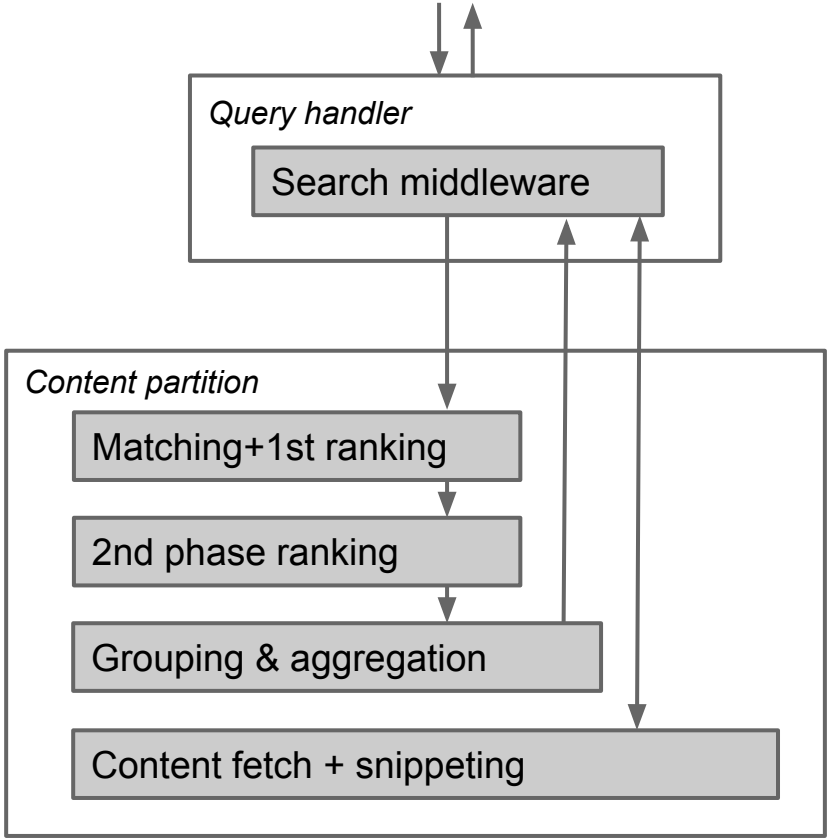
Performance at scale

Low latency computation over large data sets

… by parallelization over nodes and cores

... pushing execution to the data

... and preparing data structures in real time at write time



TensorFlow, ONNX and XGBoost integration

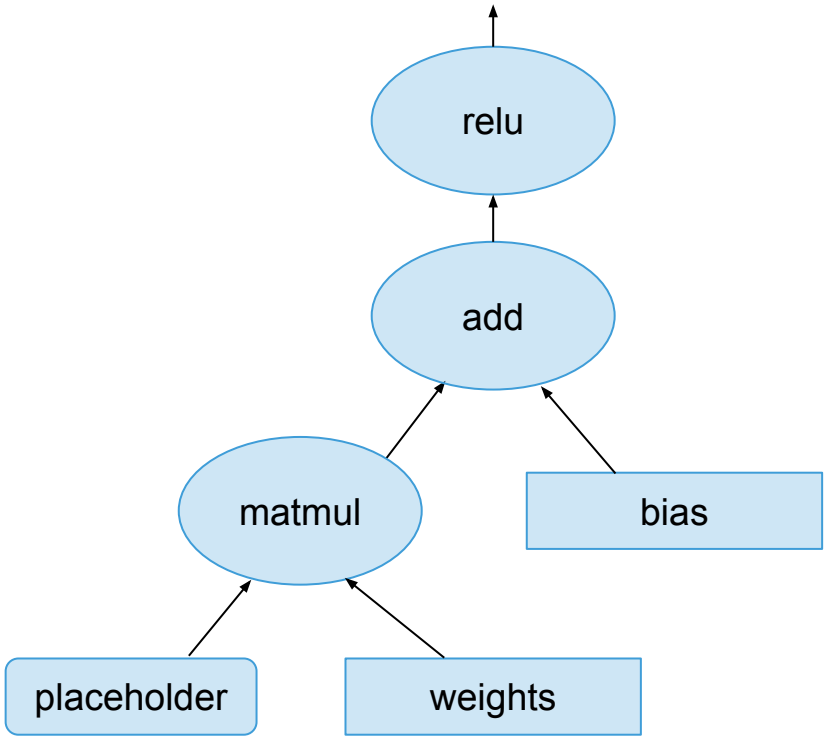
1. Save models directly to

```
<application package>/models/
```

2. Reference model outputs in ranking expressions:

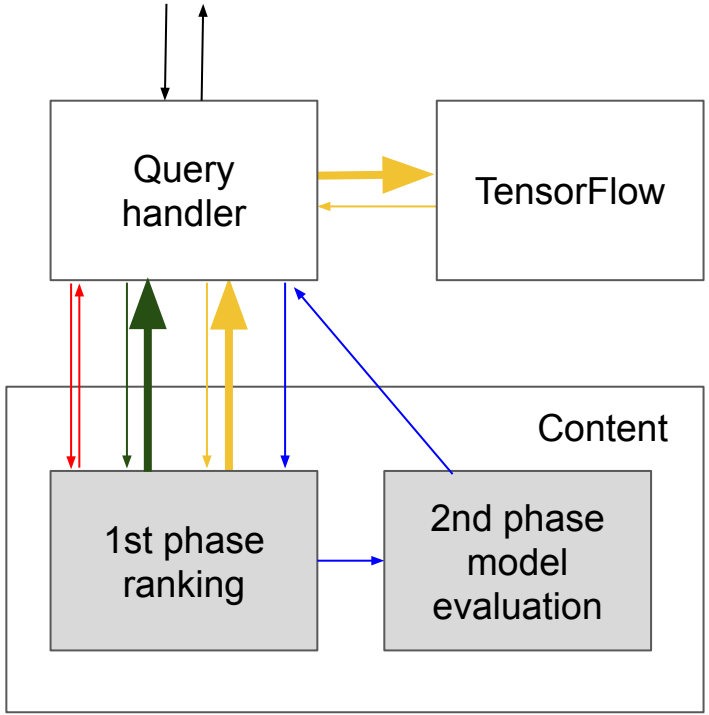
```
search music {
   rank-profile song inherits default {
      first-phase {
         expression {
            0.7 * nativeRank(artist,album,track) +
            0.1 * tensorflow(tf-model-dir) +
            0.1 * onnx(onnx-model-file, output) +
            0.1 * xgboost(xgboost-model-file)
         }
      }
   }
}
```

Converting computational graphs to Vespa




```
map(
    join(
        reduce(
            join(
                placeholder,
                weights,
                f(x,y)(x * y)
            ),
            sum,
            d1
        ),
        bias,
        f(x,y)(x + y)
    ),
    f(x)(max(0,x))
)
```

Benchmark - recommendation system



Baseline

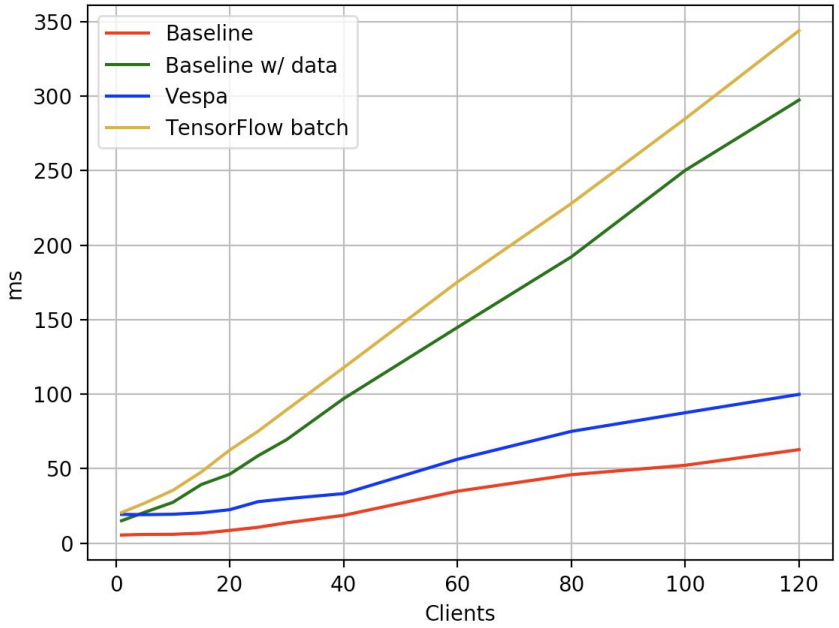
Baseline w/ data

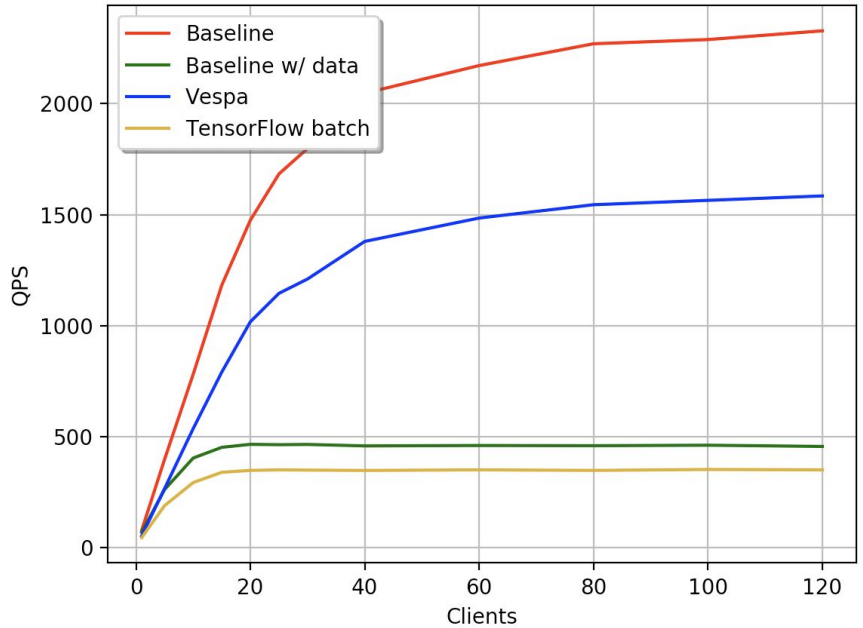
Model evaluation on Vespa

TensorFlow serving evaluation

3 layer neural network: *(128+128) x 512 x 128 x 1*
~ 200 K parameters

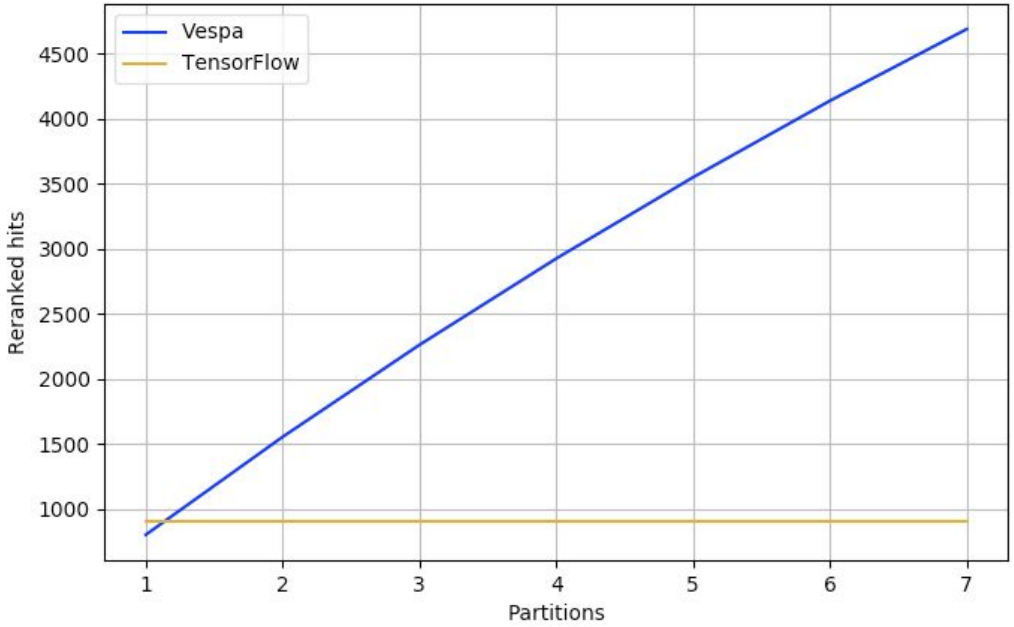
Scaling up model inference performance





http://blog.vespa.ai/post/173669458506/scaling-tensorflow-model-evaluation-with-vespa

Increasing number of evaluated results



SLA
Latency: 100ms @ 95%

Utilizing increased resources to potentially increase quality of returned results.

To conclude

- External model servers don’t scale well for ML in search
- Use additional content nodes for decreasing latency, increasing throughput and/or increasing results reranked
- Multi-phase ranking - you might be doing it wrong
- ML model support in Vespa is ongoing work

https://vespa.ai

https://github.com/vespa-engine/vespa

vespa.ai

Thank you!