

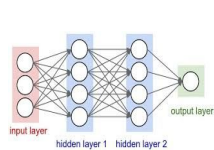
Imagine

Yi Chen
Data Engineering

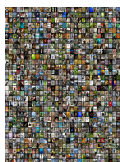


Match Framework

Inputs:



Model



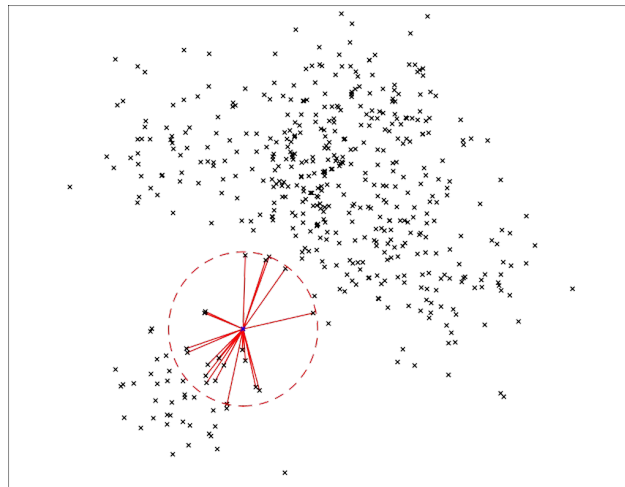
Match Space



Item



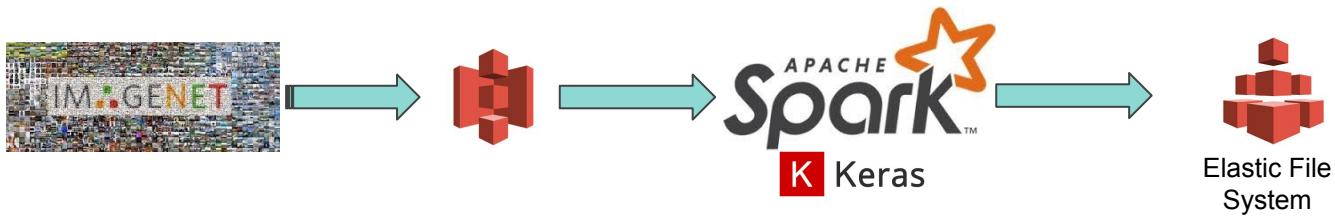
Match Framework



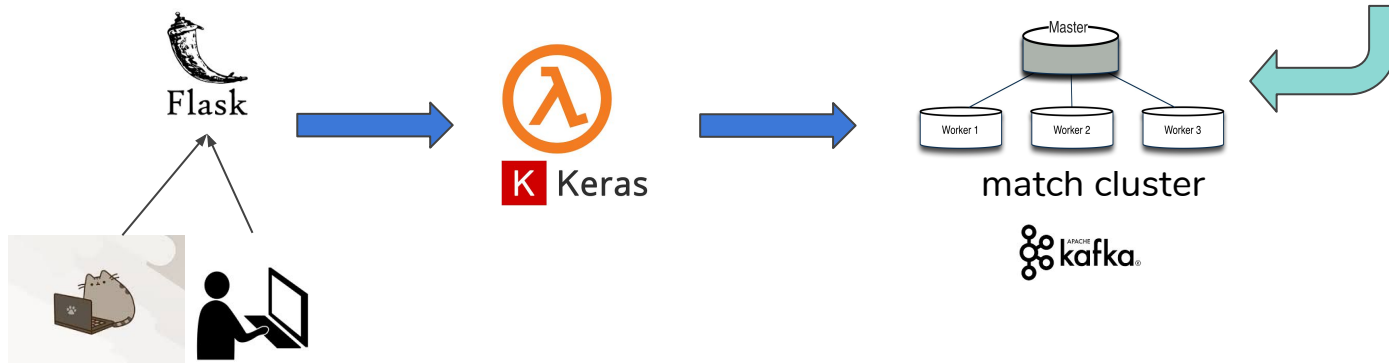
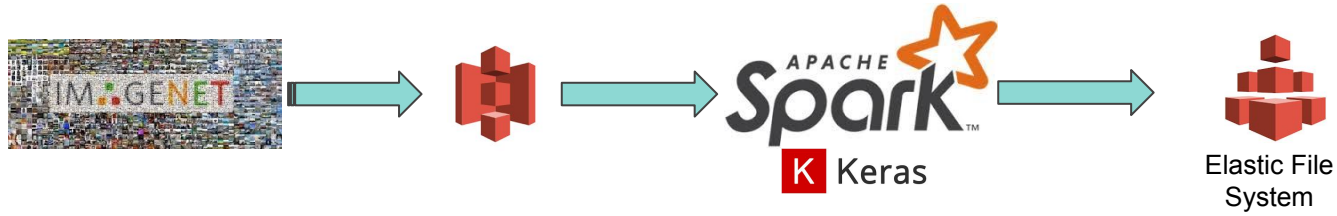
Output:



Tech Stack



Tech Stack



Demo: Nearest Images

input



output





Welcome to Your Network Storage (Neighbor)!

Please upload your images

[Upload File](#) | [2 files](#)

[Get](#)

[Search Network](#) | [10/10](#)

Downloaded from [Screencastify](#)

Downloaded from [Screencastify](#)

Downloaded from [Screencastify](#) | [10/10](#)

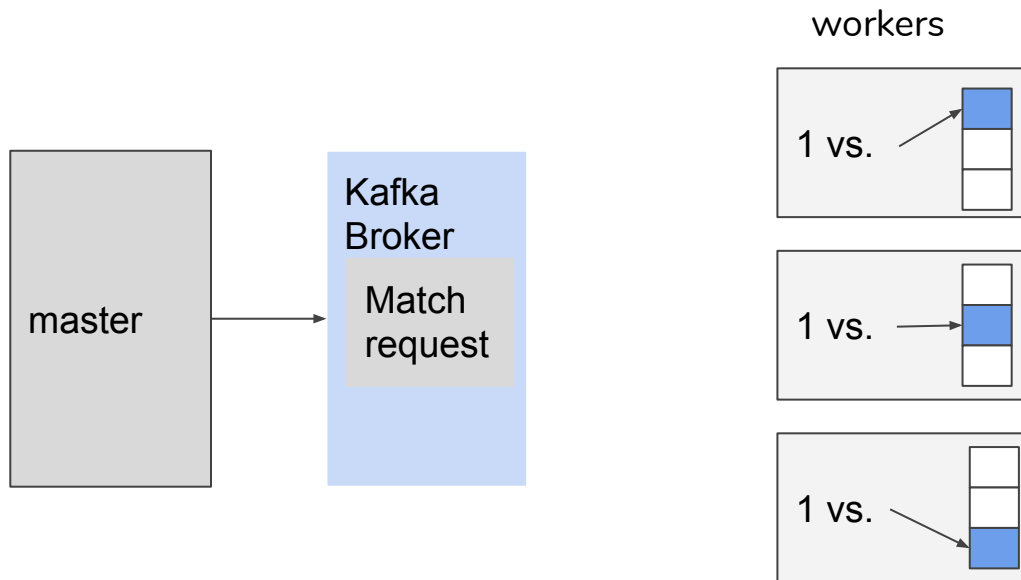
[New sharing](#)

[File](#)



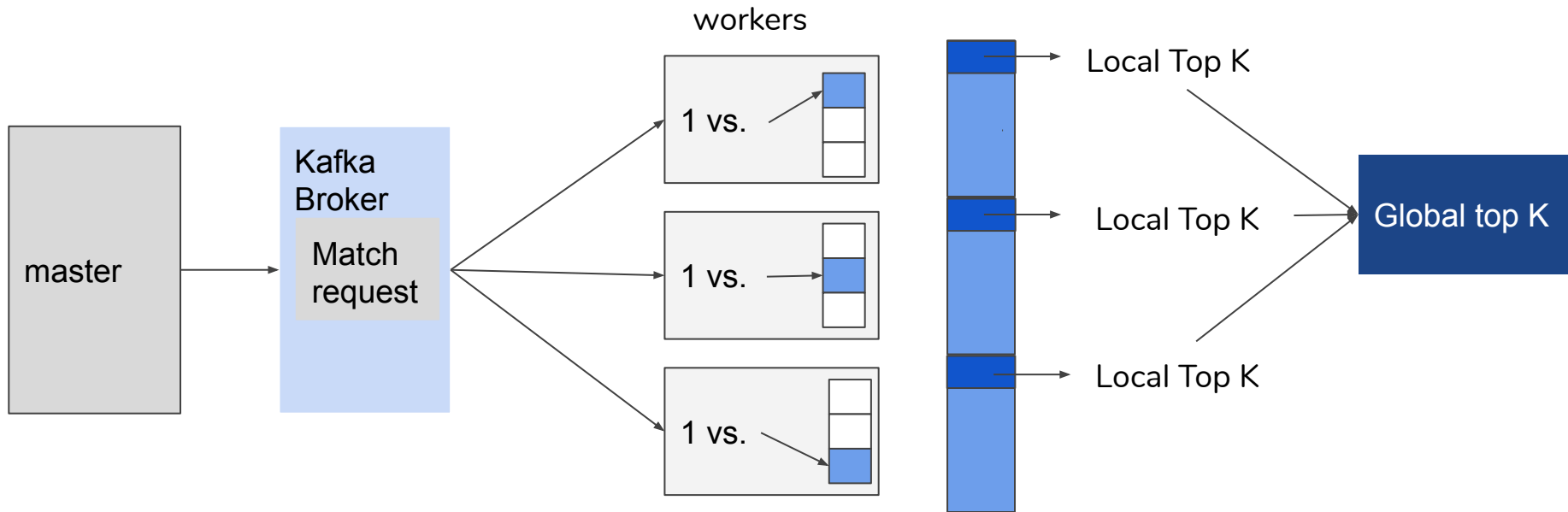


Parallelized the work

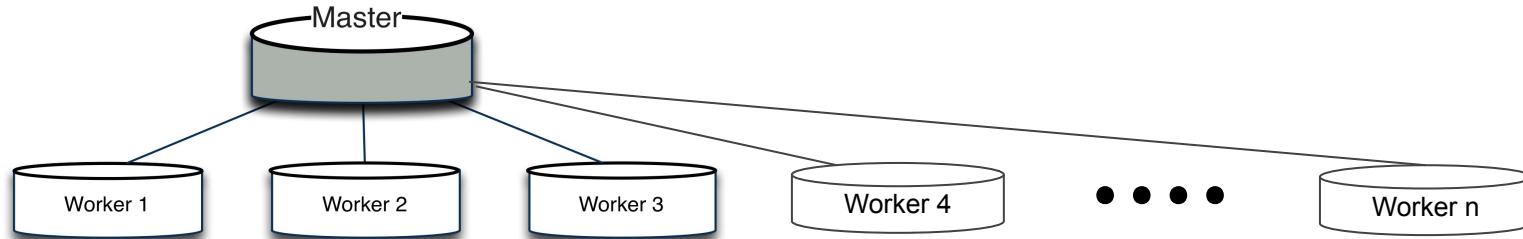




Parallelized the work



Scalability - match space increase



of workers = match space size/match ability per worker

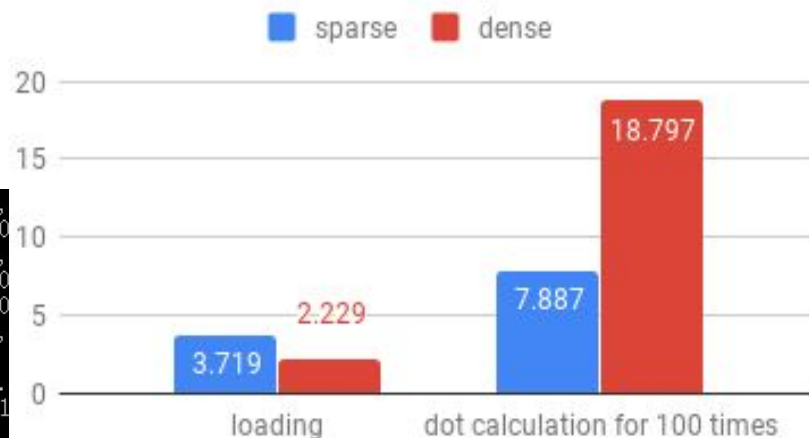


Detailed algorithm-Sparse vs. Dense

- Loading is only required when worker starts
- Doc product on sparse is more than **2X faster** than dense

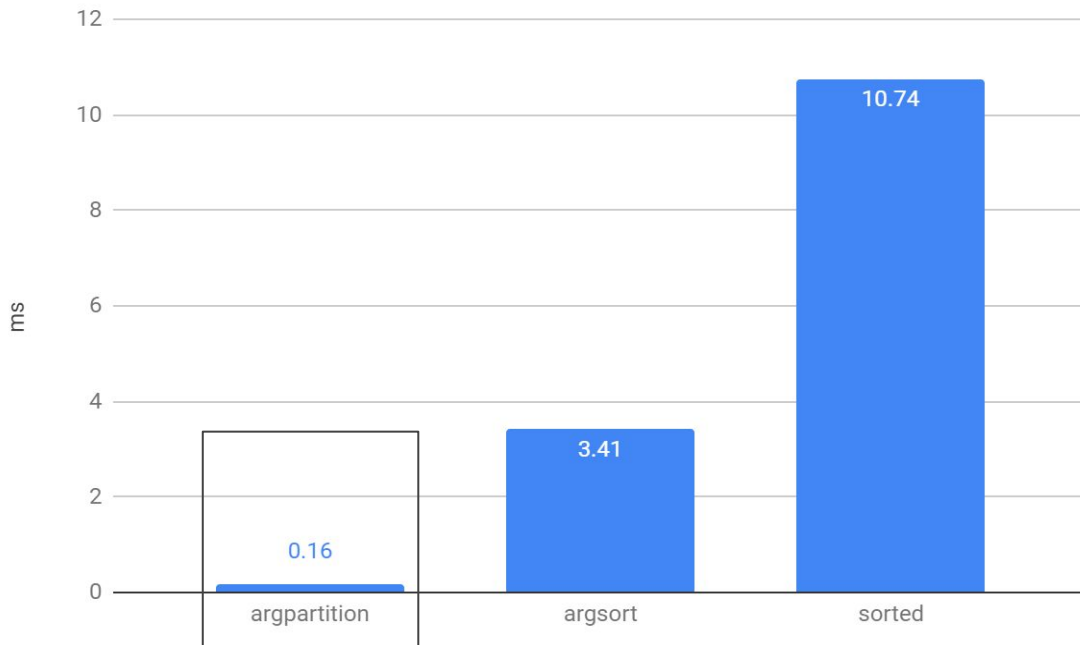
```
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.6590303182601929,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 38.64180374145508, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 5.812925338745117, 0.0,
0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 14.590415000915527, 0.0, 21.297393798828125, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 17.016456604003906, 0.0, 0.0, 0.0, 0.0, 0.0,
12695, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 10.1
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.4152735471725464, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
```

Sparse vs. Dense





Detailed algorithm-Sorting vs. Argpartition for 20,000 images





Yi Chen

B.S in Engineering, Xi'an JiaoTong University

M.S in Economics, Concordia University

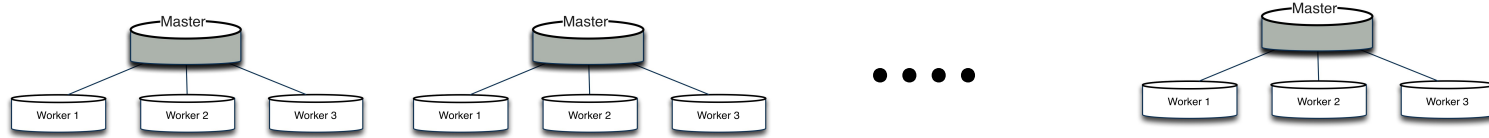
Data analyst and Data predictive Modeller

juliettyichen@gmail.com



Q&A Backup

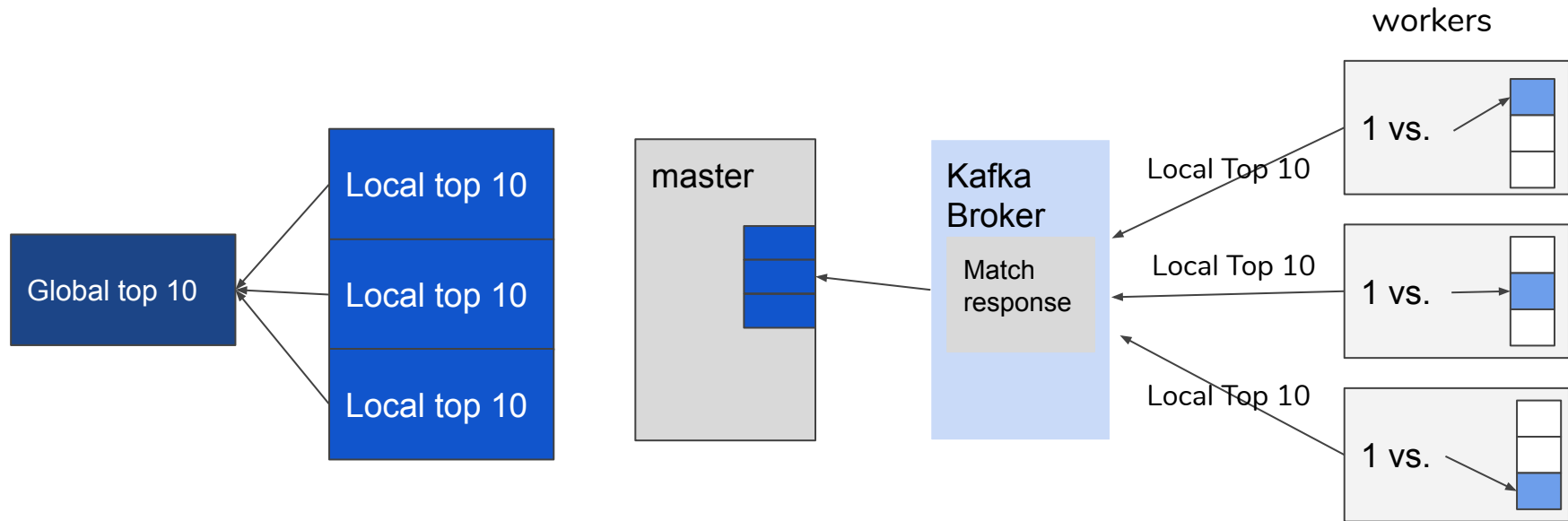
Scalability-search queries increase



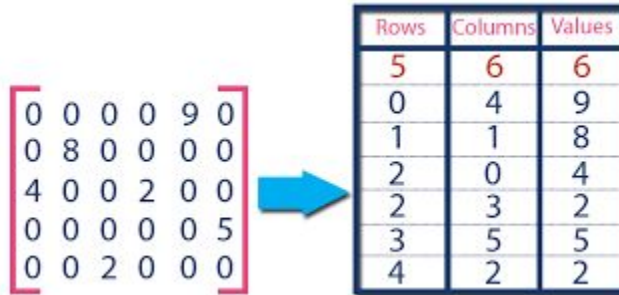
of master = $\text{qps} / \text{qps per cluster}$ (qps: query per second)



Parallelized the work



How sparse matrix stored



0	0	0	0	9	0
0	8	0	0	0	0
4	0	0	2	0	0
0	0	0	0	0	5
0	0	2	0	0	0

Rows	Columns	Values
5	6	6
0	4	9
1	1	8
2	0	4
2	3	2
3	5	5
4	2	2