Milestone III

21 May 2021

CS-449

Systems for Data Science

Submitted By:
Vignoud Julien

Sciper
282 142

3. Optimizing

1. The MAE for $k = 100$ is 0.7563 and for $k = 200$ is 0.7485.
2. For computing the k-nearest neighbours, the minimum time is $444221\mu s$, the maximum time is $1155015\mu s$, the average is $631405\mu s$ and the standard deviation is $264087\mu s$.
3. For computing the predictions, the minimum time is $2796943\mu s$, the maximum time is $4355840\mu s$, the average is $3607420\mu s$ and the standard deviation is $614019\mu s$.
4. Running the previous implementation of Milestone 2 on the cluster takes 18s for all knn. The ratio is 18/0.63, which is around 29. Using the Breeze library makes the implementation faster due to its matrix algebra optimization and underlying routines.

4. Scaling

1. The MAE for $k = 200$ of the ml-1m test set is 0.7346
2. The computation times are reported in Table 2. The average time is 77.4s for computing the knn and 107s for the predictions. The optimized version spent on average 18.5s to compute the knn and 56s to compute the predictions, which is much faster than the Spark version. Computing the knn using Spark takes 58.9 more seconds for knn and the predictions takes 51 more seconds than the optimized version.

knn	predictions
77	111
74	111
82	106
76	99
78	108

Table 1: Computation time in seconds using local[1] master

3. The computation times are reported in Table 3. The statistics are reported in Table 3, containing the minimum, maximum and average runtime for the knn and prediction parts using different number of executors. We can observe a speedup while increasing the number of executors. This is coherent since increasing the parallelization decreases the total time. However the speedup is not linear. In fact it decreases with the number of executors: using 2 executors instead of 1 almost reduces the runtime by half however using 16

executors instead of 8 doesn't half the runtime anymore. Increasing the number of executors is less and less rewarding.

nb executors	knn	predictions
1	23	121
1	18	99
1	20	119
2	12	56
2	13	61
2	11	50
4	9.1	30
4	8.6	27
4	8.1	26
8	7.0	15
8	7.0	16
8	6.7	15
16	6.3	8.4
16	6.1	8.2
16	6.3	8.6

Table 2: Computation time in seconds using different number of executors

5. Economics

1. The cluster costs 35000.- and costs 20.40 .-/day for rent. Renting for more than 1716 days, i.e., 4 years and 276 days costs more than buying the cluster.
2. The IC cluster contains $2 * 14 = 28$ cores and has $24 * 64 = 1536$GB of RAM. The daily cost of a container with similar characteristics would be $28 * 0.092 + 1536 * 0.012 = 21$.-/day. The ratio is $20.4/21 = 0.97$, the difference is negligible.
3. At max power, a RPi costs 0.216 .-/day while at min power it costs 0.043.-/day.

nb executors	knn			predictions		
	min	max	avg	min	max	avg
1	18	23	20.3	99	121	113
2	11	13	12	50	61	55.7
4	8.1	9.1	8.6	26	30	27.6
8	6.7	7.0	6.9	15	16	15.3
16	6.1	6.3	6.37	8.2	8.6	8.4

Table 3: Statistics on the computation time when increasing the number of executors

Four RPi's sum up to 4 Cortex-A72, which is equivalent to 1 Intel CPU core. It also sums to $4 * 8 = 32$GB of RAM. The daily cost of a container with such specifications would cost $0.092 + 32 * 0.012 = 0.476$.-/day. The ratio at max power is $4*0.054/0.476 = 0.45$ and at min power it is $4*0.0108/0.476 = 0.09$. It is cheaper to use 4 RPi's than renting a container, even when used at maximum power.

4. Let's solve the inequality $4 * (94.83 + 0.054x) < 0.476x$ for x, the number of days of renting the container such that the cost is higher than buying and using the RPi's (at max power). The inequality gives us $\frac{4 * 94.83}{0.476 - 4 * 0.054} < x$ The left hand side is equal to 1458.9, i.e., renting a container for at least 1459 days is more expensive than buying the RPi's. Similarly, at min power, the minimum number of days would be 877.

5. Given that $35000/94.83 = 369$, the IC cluster costs as much as 369 RPi's. Such a number of RPi's would accumulate $369 * 8 = 2952$GB of RAM, which is more than the cluster's RAM: $2952/1536 = 1.92$. Assuming perfect scaling along with a equivalent throughput between an Intel core and 4 Cortex-A72 (i.e. 4 RPi's), the throughput ratio is $369/(4 * 28) = 3.29$.

6. Let's solve the equation $x * 200 * 32/8 + 100x = 10^9/2$ for x, the number of users that we can fit in 1GB. We get $x = 10^9/(2*900) = 555,555$ users/GB. A RPi has 8GB of RAM, thus can fit $8x = 4,444,440$ users while the IC cluster has 1536GB of RAM and can fit $1536x = 853,332,480$ users in memory.

7. As always, the preferred option heavily depends on the usage. We can categorize the requirements in two dimensions: performance and time. The preferred options are summarized in Table 7.

   If the use case is for a long-term task, independently of the performances needed, it may be worth to buy the hardware, as we have seen in the previous questions that it is usually cheaper than renting. Conversely, a short-term usage can be more convenient to perform by renting, as the initial cost of buying the material won't be compensated before the end of the task.

   Regarding the performances, if a small amount or RAM and CPU throughput is required, it may be best to rent a container that fits our need if the task is short, or buy the adequate amount of RPi's for a long term task. Indeed both option can be adapted to the granularity of our needs while the cluster package comes for entire days with pre-determined specifications that are too high for a low performance task.
   However, for a high performance requirement, renting the cluster or a container for a short task may be the best option to fit the performance need without the

initial cost of buying (we saw that cost the same daily price), also depending on whether the task is distributed or not. For a long term high-performance task, we can argue between using multiple RPi's and buying a cluster. Indeed, we can achieve similar performance with both options but the RPi's may come out cheaper at the cost of more maintenance of the infrastructure.

Preferred option		Time	
		Short	Long
Performance	Low	Container	RPi
	High	Container/Cluster	RPi's/Cluster

Table 4: Preferred option depending on the requirements