

# OpenMP

kevin Lopez Chavez

November 22, 2024

## 1 Report deliverables

- The file is named `report.pdf`.
- An explanation of why I think this application is a good choice for OpenMP (Include the link to the exact location where the original program is referenced from) in Sections 2.
- An explanation of the code, the program's functionality, flow, and OpenMP acceleration in Section 3.
- Explained the estimated speed up in Section 4.
- Proof of achieved speedup with expectations in Section 5.

## 2 Explanation why data normalization is a good choice of program for OpenMP

OpenMP can achieve a high speed over independent for-loop interactions, and it's the idea of why it can achieve a high speedup on z-score normalization (or any normalization strategy, also *batch normalization*). In the case of z-score normalization, the total(sum) must be calculated, which can be parallelized. Calculate the difference to the mean, which can also be parallelized, and update all the current data, which can also be normalized. With that being said, it's possible to parallelize all the different operations in z-score normalization.

### 2.1 Links to Exact Locations

I referred to the following articles to understand the implementation and usage of Z-Score Normalization.

**Explanation and example usage of data normalization:**

- [https://developers.google.com/machine-learning/crash-course/numerical-data/normalization#z-score\\_scaling](https://developers.google.com/machine-learning/crash-course/numerical-data/normalization#z-score_scaling)

**The code implementation of the Z-score normalization (Scikit-learn repository):**

- [https://github.com/scikit-learn/scikit-learn/blob/main/sklearn/preprocessing/\\_data.py](https://github.com/scikit-learn/scikit-learn/blob/main/sklearn/preprocessing/_data.py)

## 3 Code Explanation

Here, I will explain the original code implementation in Section 3.1 and the OpenMP code implementation in Section 3.2.

### 3.1 Original Algorithm

The original algorithm (not optimized) uses the basic functions of mean and standard deviation to update each value individually and one by one. Since this needs to be run a single thread or process at a time, it takes  $n$  times to finish executing the algorithm, where  $n$  is the number of data points. I have made functions to find the mean and standard deviation. And used in the main normalization function. Each function is defined as follows:

- The following formula is used to calculate the *Z-score normalization*:  $Z = \frac{\vec{X} - \mu}{\sigma}$ . where:  $\vec{X}$  is the original input,  $\mu$  is the mean of  $\vec{X}$ ,  $\sigma$  is the standard deviation of  $\vec{X}$ .
- The mean is calculated by  $\mu = \frac{\sum_{i=1}^n x_i}{n}$ ; where:  $x_i$  is the  $i$ th data point, and  $n$  is the total points.
- The standard deviation is calculated by:  $\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \mu)^2}{n}}$ .

### 3.2 OpenMP

It is possible to use OpenMP to handle multiple data points simultaneously. In the case of parallelizing z-score-normalization, it is possible to parallelize a lot of the work done on the data. Below are the different loops that can be parallelized and provide a high speed-up.

- Calculating the sum (needed for the mean ).
- Calculating the sum of squared deviations (needed for the standard deviation)
- Z-score Normalizing the actual data.

#### 3.2.1 Matrix Normalization

The following formula is used to calculate the *Z-score normalization*:  $Z = \frac{\vec{X} - \mu}{\sigma}$ . where:  $\vec{X}$  is the original input;  $\mu$  is the mean of  $\vec{X}$ ;  $\sigma$  is the standard deviation of  $\vec{X}$ ; In my case of normalizing data with *Z-score normalization*, I need to calculate the mean (Section 3.2.2) and Standard Deviation (Section 3.2.3 ), which is parallelized using OpenMP.

Once I have the mean and standard deviation, we must update every value based on the formula. To speed up the process of updating every value, it is also possible to use OpenMP, handling  $n$  data points at a time. Where  $n$  is the number of cores allowed to run parallel done by the function `omp_set_num_threads(n)`. It is possible to get the best speed up when using the maximum number of cores in the PC by using the function `omp_get_max_threads()`

To actually normalize the data, it is possible to use the basic formula from the original function. Since this function does not have any dependencies, it is possible to optimize it using OpenMP basic for loop `#pragma omp parallel for`. Using this OpenMP macro allows the compiler to distribute the work between the different processors configured and achieve higher speed up compared to the single-threaded program.

#### 3.2.2 Mean

The mean is calculated by  $\mu = \frac{\sum_{i=1}^n x_i}{n}$ ; where:  $x_i$  is the  $i$ th data point, and  $n$  is the total number of data points.

To speed up the mean calculation, it is possible to calculate the sum in parallel; the only issue is that the sum variable is a value shared across all the running processes. OpenMP does allow to speed this up even with the dependency by using the **reduction** keyword along with the variable shared. The complete macro to speed up the sum calculation is `#pragma omp parallel for reduction(+:sum)` where the sum is the shared variable(each process would have its own copy of it where they accumulate the sum). In the end, OpenMP reduces all the sum variables copied into the processes into one global variable.

### 3.2.3 Standard Deviation

The standard deviation is calculated by:  $\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \mu)^2}{n}}$ ; where:  $x_i$  is the  $i$ th data point,  $\mu$  is the mean,  $n$  is the total points.

With the mean of the whole data, it is possible to get the deviation of every data point and sum to the variable *sum*. Again, since it is necessary to write to the same variable name, this would cause race conditions; in this case, we again use the same method **reduction** from openMP to handle it. The full line to optimize all the operations is **#pragma omp parallel for reduction(+:sum)** where *sum* variable is the sum of squared deviations. In the end, to get the variance, we just divide it by the number of elements.

## 3.3 Compilation Steps and Flags

I used the following commands to compile the code

- **Original version:**

- \$ g++ ./z\_score\_norm\_original.cpp -o original.exe

- **OpenMP version:**

- \$ g++ -fopenmp ./z\_score\_norm\_OpenMP.cpp -o OpenMP.exe

Where the flags mean the following:

- **g++:** Compiler used to compile the C++ code.
- **-fopenmp:** This flag enables OpenMP.
- **./z\_score\_norm\_original.cpp ./z\_score\_norm\_OpenMP.cpp:** The source file.
- **-o original.exe:** The -o: name of the output executable file.
- **&& ./original.exe:** Execute the program after compiling.

to run the code and get meaningful results the code was run 20 times in a row, for this I used the bash script below.

---

```
1 $ echo -e "-----\n\n\n\n\n ##### ORIGINAL #####"
2 g++ ./z_score_norm_original.cpp -o original.exe
3 for i in {1..20}
4 do
5     echo -e "-----\n\n\n\n\nRun #${i}"
6     ./original.exe
7
8 echo -e "-----\n\n\n\n\n ##### OpenMP #####"
9 g++ -fopenmp ./z_score_norm_OpenMP.cpp -o OpenMP.exe
10 for i in {1..20}
11 do
12     echo -e "-----\n\n\n\n\nRun #${i}"
13     ./OpenMP.exe
14 done
```

---

## 4 Estimated Speedup

Looking down from the big picture, we might assume that this should get a speed up of  $n$  where  $n$  is the number of processors, but there is one thing that we still need to take into account, and there are still some portions of the code that need to be handled sequentially. In the **reduce** method of OpenMP, the multiple processors need to create private locations of the sum variable that allow it to accumulate the sum value across different processors, and it would need to reduce it back to one single sum. This would take some time with the addition of the rest of the code that needs to run in a single core. For example, entering functions, making the last set of operations(divisions), and other operations.

## 5 Proof of Achieved Speedup

To prove the speed-up achieved, I have attached the logs and Figures 1,2. I posted the logs (run the programs 20 times) for the original version in section 5.1 and the oepnMP version in section 5.2.

The programs were executed 20 times, the numbers are shown in table 1:

Table 1: OpenMP and Original Comparison

Run_number	original_time	OpenMP_time
run #1	52.54	6.04
run #2	39.52	5.84
run #3	36.36	6.01
run #4	36.04	6.46
run #5	33.96	6.37
run #6	52.62	5.22
run #7	52.31	6.12
run #8	31.13	6.70
run #9	31.12	5.68
run #10	30.87	5.74
run #11	36.23	5.19
run #12	60.24	4.43
run #13	36.28	5.55
run #14	37.14	6.05
run #15	33.95	8.20
run #16	36.74	8.13
run #17	60.59	8.15
run #18	63.19	7.24
run #19	36.95	8.33
run #20	38.10	8.09
<b>average</b>	<b>41.79</b>	<b>6.48</b>
standard_deviation	10.71	1.16
speed_up	1.00	<b>6.45</b>

$$\text{Speed-up} = \frac{\text{Time for Original Version}}{\text{Time for OpenMP Version}} = \frac{41.79}{6.48} \approx 6.45$$

One can assume that the speed-up would be close to the number of parallel processes running. When looking in, we can see that there is more to why it won't run at the expected theoretical max. There are portions of the code that need to run in a single process, and also, the OpenMP instructions can add some overhead(The reduce method would need to make copies of the sum variable in each process and then reduce it back to one). Therefore, getting a speedup of  $\approx 6.45$  is a valid number for this application.



## 5.1 Logs for Original File Execution

```
1
2 kevin@Kevin-X16 MINGW64 /c/Repositories/School/Semester4/CECS_574/HW2/lopez_kevin_013378831 (main)
3 $ echo -e "-----\n\n\n\n\n ##### ORIGINAL #####"
4 g++ -fopenmp ./z_score_norm_original.cpp -o original.exe
5 for i in {1..20}
6 do
7     echo -e "-----\n\n\n\nRun #i"
8     ./original.exe
9
10 echo -e "-----\n\n\n\n\n ##### OPENMP #####"
11 g++ -fopenmp ./z_score_norm_openMP.cpp -o openMP.exe && ./openMP.exe
12 for i in {1..20}
13 do
14     echo -e "-----\n\n\n\nRun #i"
15     ./openMP.exe
16 done
17 -----
18
19
20
21
22 ##### ORIGINAL #####
23 -----
24
25
26
27 Run #1
28 Matrix before normalization
29 4.00 5.00 2.00 7.00 3.00 9.00 4.00 4.00 8.00 1.00
30 Mean of the first row: 5.00
31 Standard Deviation of the first row: 3.16
32 Mean of the last row: 5.00
33 Standard Deviation of the last row: 3.16
34
35 Matrix after normalization
36 -0.32 0.00 -0.95 0.63 -0.63 1.26 -0.32 -0.32 0.95 -1.26
37 Mean of the first row: 0.00
38 Standard Deviation of the first row: 1.00
39 Mean of the last row: 0.00
40 Standard Deviation of the last row: 1.00
41 Time taken for NON-OpenMP code: 52.54 seconds.
42 -----
43
44
45
46 Run #2
47 Matrix before normalization
48 3.00 7.00 1.00 4.00 5.00 7.00 10.00 0.00 5.00 7.00
49 Mean of the first row: 5.00
50 Standard Deviation of the first row: 3.16
51 Mean of the last row: 5.00
52 Standard Deviation of the last row: 3.16
53
54 Matrix after normalization
55 -0.63 0.63 -1.26 -0.32 0.00 0.63 1.58 -1.58 0.00 0.63
56 Mean of the first row: 0.00
57 Standard Deviation of the first row: 1.00
58 Mean of the last row: 0.00
59 Standard Deviation of the last row: 1.00
60 Time taken for NON-OpenMP code: 39.52 seconds.
61 -----
62
63
64
65 Run #3
66 Matrix before normalization
67 2.00 3.00 8.00 3.00 4.00 1.00 10.00 9.00 0.00 8.00
68 Mean of the first row: 5.00
69 Standard Deviation of the first row: 3.16
70 Mean of the last row: 5.00
71 Standard Deviation of the last row: 3.16
72
73 Matrix after normalization
74 -0.95 -0.63 0.95 -0.63 -0.32 -1.26 1.58 1.27 -1.58 0.95
75 Mean of the first row: 0.00
76 Standard Deviation of the first row: 1.00
77 Mean of the last row: 0.00
78 Standard Deviation of the last row: 1.00
```

```

79 Time taken for NON-OpenMP code: 36.36 seconds.
80 -----
81
82
83
84 Run #4
85 Matrix before normalization
86 0.00 6.00 7.00 1.00 4.00 3.00 7.00 2.00 7.00 2.00
87
88 Mean of the first row: 5.00
89 Standard Deviation of the first row: 3.16
90 Mean of the last row: 5.00
91 Standard Deviation of the last row: 3.16
92
93 Matrix after normalization
94 -1.58 0.32 0.63 -1.26 -0.32 -0.63 0.63 -0.95 0.63 -0.95
95 Mean of the first row: -0.00
96 Standard Deviation of the first row: 1.00
97 Mean of the last row: -0.00
98 Standard Deviation of the last row: 1.00
99 Time taken for NON-OpenMP code: 36.04 seconds.
100 -----
101
102
103
104 Run #5
105 Matrix before normalization
106 4.00 6.00 1.00 10.00 7.00 8.00 10.00 8.00 6.00 0.00
107 Mean of the first row: 5.00
108 Standard Deviation of the first row: 3.16
109 Mean of the last row: 5.00
110 Standard Deviation of the last row: 3.16
111
112 Matrix after normalization
113 -0.32 0.32 -1.26 1.58 0.63 0.95 1.58 0.95 0.32 -1.58
114 Mean of the first row: -0.00
115 Standard Deviation of the first row: 1.00
116 Mean of the last row: -0.00
117 Standard Deviation of the last row: 1.00
118 Time taken for NON-OpenMP code: 33.96 seconds.
119 -----
120
121
122
123 Run #6
124 Matrix before normalization
125 6.00 8.00 8.00 6.00 0.00 4.00 6.00 6.00 3.00 2.00
126
127 Mean of the first row: 5.00
128 Standard Deviation of the first row: 3.16
129 Mean of the last row: 5.00
130 Standard Deviation of the last row: 3.16
131
132 Matrix after normalization
133 0.32 0.95 0.95 0.32 -1.58 -0.32 0.32 0.32 -0.63 -0.95
134 Mean of the first row: -0.00
135 Standard Deviation of the first row: 1.00
136 Mean of the last row: -0.00
137 Standard Deviation of the last row: 1.00
138 Time taken for NON-OpenMP code: 52.62 seconds.
139 -----
140
141
142
143 Run #7
144 Matrix before normalization
145 8.00 8.00 7.00 2.00 5.00 8.00 8.00 2.00 2.00 8.00
146
147 Mean of the first row: 5.00
148 Standard Deviation of the first row: 3.16
149 Mean of the last row: 5.00
150 Standard Deviation of the last row: 3.16
151
152 Matrix after normalization
153 0.95 0.95 0.63 -0.95 0.00 0.95 0.95 -0.95 -0.95 0.95
154 Mean of the first row: 0.00
155 Standard Deviation of the first row: 1.00
156 Mean of the last row: 0.00
157 Standard Deviation of the last row: 1.00
158 Time taken for NON-OpenMP code: 52.31 seconds.
159 -----

```

```

160
161
162
163 Run #8
164 Matrix before normalization
165 8.00 8.00 5.00 1.00 3.00 1.00 5.00 4.00 4.00 3.00
166
167 Mean of the first row: 5.00
168 Standard Deviation of the first row: 3.16
169 Mean of the last row: 5.00
170 Standard Deviation of the last row: 3.16
171
172 Matrix after normalization
173 0.95 0.95 0.00 -1.26 -0.63 -1.26 0.00 -0.32 -0.32 -0.63
174 Mean of the first row: 0.00
175 Standard Deviation of the first row: 1.00
176 Mean of the last row: 0.00
177 Standard Deviation of the last row: 1.00
178 Time taken for NON-OpenMP code: 31.13 seconds.
179 -----
180
181
182
183 Run #9
184 Matrix before normalization
185 9.00 2.00 4.00 7.00 0.00 3.00 3.00 8.00 1.00 7.00
186
187 Mean of the first row: 5.00
188 Standard Deviation of the first row: 3.16
189 Mean of the last row: 5.00
190 Standard Deviation of the last row: 3.16
191
192 Matrix after normalization
193 1.27 -0.95 -0.32 0.63 -1.58 -0.63 -0.63 0.95 -1.26 0.63
194 Mean of the first row: -0.00
195 Standard Deviation of the first row: 1.00
196 Mean of the last row: -0.00
197 Standard Deviation of the last row: 1.00
198 Time taken for NON-OpenMP code: 31.12 seconds.
199 Standard Deviation of the last row: 3.16
200
201 Matrix after normalization
202 1.26 -0.95 -1.26 -0.32 -0.32 -1.58 0.00 -0.95 0.32 1.26
203 Mean of the first row: 0.00
204 Standard Deviation of the first row: 1.00
205 Mean of the last row: 0.00
206 Standard Deviation of the last row: 1.00
207 Time taken for NON-OpenMP code: 30.87 seconds.
208 -----
209
210
211
212 Run #11
213 Matrix before normalization
214 9.00 2.00 10.00 0.00 7.00 9.00 7.00 9.00 0.00 0.00
215 Mean of the first row: 5.00
216 Standard Deviation of the first row: 3.16
217 Mean of the last row: 5.00
218 Standard Deviation of the last row: 3.16
219
220 Matrix after normalization
221 1.26 -0.95 1.58 -1.58 0.63 1.26 0.63 1.26 -1.58 -1.58
222 Mean of the first row: -0.00
223 Standard Deviation of the first row: 1.00
224 Mean of the last row: -0.00
225 Standard Deviation of the last row: 1.00
226 Time taken for NON-OpenMP code: 36.23 seconds.
227 -----
228
229
230
231 Run #12
232 Matrix before normalization
233 0.00 6.00 1.00 3.00 1.00 0.00 8.00 5.00 7.00
234 Mean of the first row: 5.00
235 Standard Deviation of the first row: 3.16
236 Mean of the last row: 5.00
237 Standard Deviation of the last row: 3.16
238
239 Matrix after normalization
240 -1.58 0.32 -1.26 -0.63 -1.26 -0.95 -1.58 0.95 0.00 0.63

```



```

241 Mean of the first row: 0.00
242 Standard Deviation of the first row: 1.00
243 Mean of the last row: 0.00
244 Standard Deviation of the last row: 1.00
245 Time taken for NON-OpenMP code: 60.24 seconds.
246 -----
247
248
249
250 Run #13
251 Matrix before normalization
252 10.00 4.00 10.00 2.00 10.00 7.00 3.00 6.00 3.00 2.00
253 Mean of the first row: 5.00
254 Standard Deviation of the first row: 3.16
255 Mean of the last row: 5.00
256 Standard Deviation of the last row: 3.16
257
258 Matrix after normalization
259 1.58 -0.32 1.58 -0.95 1.58 0.63 -0.63 0.32 -0.63 -0.95
260 Mean of the first row: 0.00
261 Standard Deviation of the first row: 1.00
262 Mean of the last row: 0.00
263 Standard Deviation of the last row: 1.00
264 Time taken for NON-OpenMP code: 36.28 seconds.
265 -----
266
267
268
269 Run #14
270 Matrix before normalization
271 7.00 4.00 2.00 9.00 5.00 5.00 6.00 5.00 7.00 1.00
272 Mean of the first row: 5.00
273 Standard Deviation of the first row: 3.16
274 Mean of the last row: 5.00
275 Standard Deviation of the last row: 3.16
276
277 Matrix after normalization
278 0.63 -0.32 -0.95 1.27 0.00 0.00 0.32 0.00 0.63 -1.26
279 Mean of the first row: 0.00
280 Standard Deviation of the first row: 1.00
281 Mean of the last row: 0.00
282 Standard Deviation of the last row: 1.00
283 Time taken for NON-OpenMP code: 37.14 seconds.
284 -----
285
286
287
288 Run #15
289 Matrix before normalization
290 0.00 9.00 8.00 4.00 8.00 10.00 6.00 6.00 8.00 2.00
291 Mean of the first row: 5.00
292 Standard Deviation of the first row: 3.16
293 Mean of the last row: 5.00
294 Standard Deviation of the last row: 3.16
295
296 Matrix after normalization
297 -1.58 1.26 0.95 -0.32 0.95 1.58 0.32 0.32 0.95 -0.95
298 Mean of the first row: 0.00
299 Standard Deviation of the first row: 1.00
300 Mean of the last row: 0.00
301 Standard Deviation of the last row: 1.00
302 Time taken for NON-OpenMP code: 33.95 seconds.
303 -----
304
305
306
307 Run #16
308 Matrix before normalization
309 3.00 7.00 8.00 8.00 1.00 3.00 1.00 9.00 1.00 6.00
310 Mean of the first row: 5.00
311 Standard Deviation of the first row: 3.16
312 Mean of the last row: 5.00
313 Standard Deviation of the last row: 3.16
314
315 Matrix after normalization
316 -0.63 0.63 0.95 0.95 -1.26 -0.63 -1.26 1.26 -1.26 0.32
317 Mean of the first row: 0.00
318 Standard Deviation of the first row: 1.00
319 Mean of the last row: 0.00
320 Standard Deviation of the last row: 1.00
321 Time taken for NON-OpenMP code: 36.74 seconds.

```

```

322 -----
323
324
325
326 Run #17
327 Matrix before normalization
328 1.00 4.00 0.00 2.00 10.00 9.00 4.00 6.00 7.00 3.00
329 Mean of the first row: 5.00
330 Standard Deviation of the first row: 3.16
331 Mean of the last row: 5.00
332 Standard Deviation of the last row: 3.16
333
334 Matrix after normalization
335 -1.26 -0.32 -1.58 -0.95 1.58 1.27 -0.32 0.32 0.63 -0.63
336 Mean of the first row: 0.00
337 Standard Deviation of the first row: 1.00
338 Mean of the last row: 0.00
339 Standard Deviation of the last row: 1.00
340 Time taken for NON-OpenMP code: 60.59 seconds.
341 -----
342
343
344
345 Run #18
346 Matrix before normalization
347 0.00 0.00 3.00 9.00 2.00 8.00 1.00 8.00 2.00 3.00
348 Mean of the first row: 5.00
349 Standard Deviation of the first row: 3.16
350 Mean of the last row: 5.00
351 Standard Deviation of the last row: 3.16
352
353 Matrix after normalization
354 -1.58 -1.58 -0.63 1.26 -0.95 0.95 -1.26 0.95 -0.95 -0.63
355 Mean of the first row: 0.00
356 Standard Deviation of the first row: 1.00
357 Mean of the last row: 0.00
358 Standard Deviation of the last row: 1.00
359 Time taken for NON-OpenMP code: 63.19 seconds.
360 -----
361
362
363
364 Run #19
365 Matrix before normalization
366 7.00 2.00 6.00 7.00 10.00 5.00 2.00 1.00 2.00 5.00
367 Mean of the first row: 5.00
368 Standard Deviation of the first row: 3.16
369 Mean of the last row: 5.00
370 Standard Deviation of the last row: 3.16
371
372 Matrix after normalization
373 0.63 -0.95 0.32 0.63 1.58 0.00 -0.95 -1.26 -0.95 0.00
374 Mean of the first row: 0.00
375 Standard Deviation of the first row: 1.00
376 Mean of the last row: 0.00
377 Standard Deviation of the last row: 1.00
378 Time taken for NON-OpenMP code: 36.95 seconds.
379 -----
380
381
382
383 Run #20
384 Matrix before normalization
385 2.00 9.00 2.00 0.00 1.00 9.00 2.00 0.00 6.00 6.00
386 Mean of the first row: 5.00
387 Standard Deviation of the first row: 3.16
388 Mean of the last row: 5.00
389 Standard Deviation of the last row: 3.16
390
391 Matrix after normalization
392 -0.95 1.26 -0.95 -1.58 -1.26 1.26 -0.95 -1.58 0.32 0.32
393 Mean of the first row: 0.00
394 Standard Deviation of the first row: 1.00
395 Mean of the last row: 0.00
396 Standard Deviation of the last row: 1.00
397 Time taken for NON-OpenMP code: 38.10 seconds.
398 -----
399
400
401

```

---

## 5.2 Logs for OpenMP File Execution

```
1
2 ##### OPENMP #####
3 Run #1
4 Matrix before normalization
5 7.00 8.00 5.00 0.00 10.00 7.00 6.00 0.00 8.00 5.00
6 Maximum number of threads available: 20
7 Mean of the first row: 5.00
8 Maximum number of threads available: 20
9 Standard Deviation of the first row: 3.16
10 Maximum number of threads available: 20
11 Mean of the last row: 5.00
12 Maximum number of threads available: 20
13 Standard Deviation of the last row: 3.16
14 Maximum number of threads available: 20
15
16 Matrix after normalization
17 0.63 0.95 0.00 -1.58 1.58 0.63 0.32 -1.58 0.95 0.00
18 Maximum number of threads available: 20
19 Mean of the first row: -0.00
20 Maximum number of threads available: 20
21 Standard Deviation of the first row: 1.00
22 Maximum number of threads available: 20
23 Mean of the last row: -0.00
24 Maximum number of threads available: 20
25 Standard Deviation of the last row: 1.00
26 Time taken for OpenMP code: 6.04 seconds.
27 -----
28
29
30
31 Run #2
32 Matrix before normalization
33 6.00 0.00 0.00 6.00 7.00 1.00 10.00 0.00 9.00 1.00
34 Maximum number of threads available: 20
35 Mean of the first row: 5.00
36 Maximum number of threads available: 20
37 Standard Deviation of the first row: 3.16
38 Maximum number of threads available: 20
39 Mean of the last row: 5.00
40 Maximum number of threads available: 20
41 Standard Deviation of the last row: 3.16
42 Maximum number of threads available: 20
43
44 Matrix after normalization
45 0.32 -1.58 -1.58 0.32 0.63 -1.26 1.58 -1.58 1.27 -1.26
46 Maximum number of threads available: 20
47 Mean of the first row: 0.00
48 Maximum number of threads available: 20
49 Standard Deviation of the first row: 1.00
50 Maximum number of threads available: 20
51 Mean of the last row: 0.00
52 Maximum number of threads available: 20
53 Standard Deviation of the last row: 1.00
54 Time taken for OpenMP code: 5.84 seconds.
55 -----
56
57
58
59 Run #3
60 Matrix before normalization
61 5.00 3.00 7.00 2.00 3.00 6.00 3.00 1.00 8.00 9.00
62 Maximum number of threads available: 20
63 Mean of the first row: 5.00
64 Maximum number of threads available: 20
65 Standard Deviation of the first row: 3.16
66 Maximum number of threads available: 20
67 Mean of the last row: 5.00
68 Maximum number of threads available: 20
69 Standard Deviation of the last row: 3.16
70 Maximum number of threads available: 20
71
72 Matrix after normalization
73 0.00 -0.63 0.63 -0.95 -0.63 0.32 -0.63 -1.26 0.95 1.26
74 Maximum number of threads available: 20
75 Mean of the first row: -0.00
76 Maximum number of threads available: 20
77 Standard Deviation of the first row: 1.00
78 Maximum number of threads available: 20
```

```

79 Mean of the last row: -0.00
80 Maximum number of threads available: 20
81 Standard Deviation of the last row: 1.00
82 Time taken for OpenMP code: 6.01 seconds.
83 -----
84
85
86
87 Run #4
88 Matrix before normalization
89 10.00 10.00 4.00 1.00 5.00 0.00 6.00 10.00 10.00 7.00
90 Maximum number of threads available: 20
91 Mean of the first row: 5.00
92 Maximum number of threads available: 20
93 Standard Deviation of the first row: 3.16
94 Maximum number of threads available: 20
95 Mean of the last row: 5.00
96 Maximum number of threads available: 20
97 Standard Deviation of the last row: 3.16
98 Maximum number of threads available: 20
99
100 Matrix after normalization
101 1.58 1.58 -0.32 -1.26 0.00 -1.58 0.32 1.58 1.58 0.63
102 Maximum number of threads available: 20
103 Mean of the first row: -0.00
104 Maximum number of threads available: 20
105 Standard Deviation of the first row: 1.00
106 Maximum number of threads available: 20
107 Mean of the last row: -0.00
108 Maximum number of threads available: 20
109 Standard Deviation of the last row: 1.00
110 Time taken for OpenMP code: 6.46 seconds.
111 -----
112
113
114
115 Run #5
116 Matrix before normalization
117 8.00 3.00 0.00 8.00 1.00 5.00 9.00 10.00 10.00 4.00
118 Maximum number of threads available: 20
119 Mean of the first row: 5.00
120 Maximum number of threads available: 20
121 Standard Deviation of the first row: 3.16
122 Maximum number of threads available: 20
123 Mean of the last row: 5.00
124 Maximum number of threads available: 20
125 Standard Deviation of the last row: 3.16
126 Maximum number of threads available: 20
127
128 Matrix after normalization
129 0.95 -0.63 -1.58 0.95 -1.26 0.00 1.26 1.58 1.58 -0.32
130 Maximum number of threads available: 20
131 Mean of the first row: -0.00
132 Maximum number of threads available: 20
133 Standard Deviation of the first row: 1.00
134 Maximum number of threads available: 20
135 Mean of the last row: -0.00
136 Maximum number of threads available: 20
137 Standard Deviation of the last row: 1.00
138 Time taken for OpenMP code: 6.37 seconds.
139 -----
140
141
142
143 Run #6
144 Matrix before normalization
145 4.00 4.00 6.00 8.00 0.00 10.00 3.00 1.00 9.00 0.00
146 Maximum number of threads available: 20
147 Mean of the first row: 5.00
148 Maximum number of threads available: 20
149 Standard Deviation of the first row: 3.16
150 Maximum number of threads available: 20
151 Mean of the last row: 5.00
152 Maximum number of threads available: 20
153 Standard Deviation of the last row: 3.16
154 Maximum number of threads available: 20
155
156 Matrix after normalization
157 -0.32 -0.32 0.32 0.95 -1.58 1.58 -0.63 -1.26 1.27 -1.58
158 Maximum number of threads available: 20
159 Mean of the first row: -0.00

```

```

160 Maximum number of threads available: 20
161 Standard Deviation of the first row: 1.00
162 Maximum number of threads available: 20
163 Mean of the last row: -0.00
164 Maximum number of threads available: 20
165 Standard Deviation of the last row: 1.00
166 Time taken for OpenMP code: 5.22 seconds.
167 -----
168
169
170
171 Run #7
172 Matrix before normalization
173 10.00 5.00 1.00 8.00 0.00 4.00 7.00 3.00 7.00 7.00
174 Maximum number of threads available: 20
175 Mean of the first row: 5.00
176 Maximum number of threads available: 20
177 Standard Deviation of the first row: 3.16
178 Maximum number of threads available: 20
179 Mean of the last row: 5.00
180 Maximum number of threads available: 20
181 Standard Deviation of the last row: 3.16
182 Maximum number of threads available: 20
183
184 Matrix after normalization
185 1.58 0.00 -1.26 0.95 -1.58 -0.32 0.63 -0.63 0.63 0.63
186 Maximum number of threads available: 20
187 Mean of the first row: -0.00
188 Maximum number of threads available: 20
189 Standard Deviation of the first row: 1.00
190 Maximum number of threads available: 20
191 Mean of the last row: -0.00
192 Maximum number of threads available: 20
193 Standard Deviation of the last row: 1.00
194 Time taken for OpenMP code: 6.12 seconds.
195 -----
196
197
198
199 Run #8
200 Matrix before normalization
201 2.00 6.00 6.00 1.00 3.00 9.00 0.00 6.00 5.00 2.00
202 Maximum number of threads available: 20
203 Mean of the first row: 5.00
204 Maximum number of threads available: 20
205 Standard Deviation of the first row: 3.16
206 Maximum number of threads available: 20
207 Mean of the last row: 5.00
208 Maximum number of threads available: 20
209 Standard Deviation of the last row: 3.16
210 Maximum number of threads available: 20
211
212 Matrix after normalization
213 -0.95 0.32 0.32 -1.26 -0.63 1.26 -1.58 0.32 -0.00 -0.95
214 Maximum number of threads available: 20
215 Mean of the first row: 0.00
216 Maximum number of threads available: 20
217 Standard Deviation of the first row: 1.00
218 Maximum number of threads available: 20
219 Mean of the last row: 0.00
220 Maximum number of threads available: 20
221 Standard Deviation of the last row: 1.00
222 Time taken for OpenMP code: 6.70 seconds.
223 -----
224
225
226
227 Run #9
228 Matrix before normalization
229 5.00 5.00 1.00 5.00 6.00 3.00 6.00 9.00 3.00 8.00
230 Maximum number of threads available: 20
231 Mean of the first row: 5.00
232 Maximum number of threads available: 20
233 Standard Deviation of the first row: 3.16
234 Maximum number of threads available: 20
235 Mean of the last row: 5.00
236 Maximum number of threads available: 20
237 Standard Deviation of the last row: 3.16
238 Maximum number of threads available: 20
239
240 Matrix after normalization

```

```

241 0.00 0.00 -1.26 0.00 0.32 -0.63 0.32 1.27 -0.63 0.95
242 Maximum number of threads available: 20
243 Mean of the first row: -0.00
244 Maximum number of threads available: 20
245 Standard Deviation of the first row: 1.00
246 Maximum number of threads available: 20
247 Mean of the last row: -0.00
248 Maximum number of threads available: 20
249 Standard Deviation of the last row: 1.00
250 Time taken for OpenMP code: 5.68 seconds.
251 -----
252
253
254
255 Run #10
256 Matrix before normalization
257 3.00 8.00 2.00 6.00 2.00 10.00 1.00 8.00 5.00 0.00
258 Maximum number of threads available: 20
259 Mean of the first row: 5.00
260 Maximum number of threads available: 20
261 Standard Deviation of the first row: 3.16
262 Maximum number of threads available: 20
263 Mean of the last row: 5.00
264 Maximum number of threads available: 20
265 Standard Deviation of the last row: 3.16
266 Maximum number of threads available: 20
267
268 Matrix after normalization
269 -0.63 0.95 -0.95 0.32 -0.95 1.58 -1.26 0.95 0.00 -1.58
270 Maximum number of threads available: 20
271 Mean of the first row: 0.00
272 Maximum number of threads available: 20
273 Standard Deviation of the first row: 1.00
274 Maximum number of threads available: 20
275 Mean of the last row: 0.00
276 Maximum number of threads available: 20
277 Standard Deviation of the last row: 1.00
278 Time taken for OpenMP code: 5.74 seconds.
279 -----
280
281
282
283 Run #11
284 Matrix before normalization
285 1.00 10.00 4.00 8.00 9.00 5.00 8.00 7.00 9.00 3.00
286 Maximum number of threads available: 20
287 Mean of the first row: 5.00
288 Maximum number of threads available: 20
289 Standard Deviation of the first row: 3.16
290 Maximum number of threads available: 20
291 Mean of the last row: 5.00
292 Maximum number of threads available: 20
293 Standard Deviation of the last row: 3.16
294 Maximum number of threads available: 20
295
296 Matrix after normalization
297 -1.26 1.58 -0.32 0.95 1.26 0.00 0.95 0.63 1.26 -0.63
298 Maximum number of threads available: 20
299 Mean of the first row: 0.00
300 Maximum number of threads available: 20
301 Standard Deviation of the first row: 1.00
302 Maximum number of threads available: 20
303 Mean of the last row: 0.00
304 Maximum number of threads available: 20
305 Standard Deviation of the last row: 1.00
306 Time taken for OpenMP code: 5.19 seconds.
307 -----
308
309
310
311 Run #12
312 Matrix before normalization
313 2.00 3.00 6.00 5.00 3.00 10.00 4.00 5.00 2.00 6.00
314 Maximum number of threads available: 20
315 Mean of the first row: 5.00
316 Maximum number of threads available: 20
317 Standard Deviation of the first row: 3.16
318 Maximum number of threads available: 20
319 Mean of the last row: 5.00
320 Maximum number of threads available: 20
321 Standard Deviation of the last row: 3.16

```

```

322 Maximum number of threads available: 20
323
324 Matrix after normalization
325 -0.95 -0.63 0.32 0.00 -0.63 1.58 -0.32 0.00 -0.95 0.32
326 Maximum number of threads available: 20
327 Mean of the first row: -0.00
328 Maximum number of threads available: 20
329 Standard Deviation of the first row: 1.00
330 Maximum number of threads available: 20
331 Mean of the last row: -0.00
332 Maximum number of threads available: 20
333 Standard Deviation of the last row: 1.00
334 Time taken for OpenMP code: 4.43 seconds.
335 -----
336
337
338
339 Run #13
340 Matrix before normalization
341 7.00 4.00 7.00 0.00 2.00 6.00 1.00 5.00 3.00 8.00
342 Maximum number of threads available: 20
343 Mean of the first row: 5.00
344 Maximum number of threads available: 20
345 Standard Deviation of the first row: 3.16
346 Maximum number of threads available: 20
347 Mean of the last row: 5.00
348 Maximum number of threads available: 20
349 Standard Deviation of the last row: 3.16
350 Maximum number of threads available: 20
351
352 Matrix after normalization
353 0.63 -0.32 0.63 -1.58 -0.95 0.32 -1.26 0.00 -0.63 0.95
354 Maximum number of threads available: 20
355 Mean of the first row: -0.00
356 Maximum number of threads available: 20
357 Standard Deviation of the first row: 1.00
358 Maximum number of threads available: 20
359 Mean of the last row: -0.00
360 Maximum number of threads available: 20
361 Standard Deviation of the last row: 1.00
362 Time taken for OpenMP code: 5.55 seconds.
363 -----
364
365
366
367 Run #14
368 Matrix before normalization
369 4.00 10.00 0.00 1.00 1.00 0.00 6.00 0.00 10.00 2.00
370 Maximum number of threads available: 20
371 Mean of the first row: 5.00
372 Maximum number of threads available: 20
373 Standard Deviation of the first row: 3.16
374 Maximum number of threads available: 20
375 Mean of the last row: 5.00
376 Maximum number of threads available: 20
377 Standard Deviation of the last row: 3.16
378 Maximum number of threads available: 20
379
380 Matrix after normalization
381 -0.32 1.58 -1.58 -1.26 -1.26 -1.58 0.32 -1.58 1.58 -0.95
382
383 Maximum number of threads available: 20
384 Mean of the first row: 0.00
385 Maximum number of threads available: 20
386 Standard Deviation of the first row: 1.00
387 Maximum number of threads available: 20
388 Mean of the last row: 0.00
389 Maximum number of threads available: 20
390 Standard Deviation of the last row: 1.00
391 Time taken for OpenMP code: 6.05 seconds.
392 -----
393
394
395
396 Run #15
397 Matrix before normalization
398 2.00 4.00 6.00 7.00 8.00 5.00 10.00 0.00 10.00 9.00
399 Maximum number of threads available: 20
400 Mean of the first row: 5.00
401 Maximum number of threads available: 20
402 Standard Deviation of the first row: 3.16

```

```

403 Maximum number of threads available: 20
404 Mean of the last row: 5.00
405 Maximum number of threads available: 20
406 Standard Deviation of the last row: 3.16
407 Maximum number of threads available: 20
408
409 Matrix after normalization
410 -0.95 -0.32 0.32 0.63 0.95 0.00 1.58 -1.58 1.58 1.26
411 Maximum number of threads available: 20
412 Mean of the first row: 0.00
413 Maximum number of threads available: 20
414 Standard Deviation of the first row: 1.00
415 Maximum number of threads available: 20
416 Mean of the last row: 0.00
417 Maximum number of threads available: 20
418 Standard Deviation of the last row: 1.00
419 Time taken for OpenMP code: 8.20 seconds.
420 -----
421
422
423
424 Run #16
425 Matrix before normalization
426 5.00 9.00 8.00 5.00 1.00 9.00 10.00 1.00 8.00 1.00
427 Maximum number of threads available: 20
428 Mean of the first row: 5.00
429 Maximum number of threads available: 20
430 Standard Deviation of the first row: 3.16
431 Maximum number of threads available: 20
432 Mean of the last row: 5.00
433 Maximum number of threads available: 20
434 Standard Deviation of the last row: 3.16
435 Maximum number of threads available: 20
436
437 Matrix after normalization
438 0.00 1.27 0.95 0.00 -1.26 1.27 1.58 -1.26 0.95 -1.26
439 Maximum number of threads available: 20
440 Mean of the first row: -0.00
441 Maximum number of threads available: 20
442 Standard Deviation of the first row: 1.00
443 Maximum number of threads available: 20
444 Mean of the last row: -0.00
445 Maximum number of threads available: 20
446 Standard Deviation of the last row: 1.00
447 Time taken for OpenMP code: 8.13 seconds.
448 -----
449
450
451
452 Run #17
453 Matrix before normalization
454 9.00 10.00 7.00 3.00 4.00 2.00 1.00 5.00 1.00 2.00
455 Maximum number of threads available: 20
456 Mean of the first row: 5.00
457 Maximum number of threads available: 20
458 Standard Deviation of the first row: 3.16
459 Maximum number of threads available: 20
460 Mean of the last row: 5.00
461 Maximum number of threads available: 20
462 Standard Deviation of the last row: 3.16
463 Maximum number of threads available: 20
464
465 Matrix after normalization
466 1.26 1.58 0.63 -0.63 -0.32 -0.95 -1.26 0.00 -1.26 -0.95
467 Maximum number of threads available: 20
468 Mean of the first row: -0.00
469 Maximum number of threads available: 20
470 Standard Deviation of the first row: 1.00
471 Maximum number of threads available: 20
472 Mean of the last row: -0.00
473 Maximum number of threads available: 20
474 Standard Deviation of the last row: 1.00
475 Time taken for OpenMP code: 8.15 seconds.
476 -----
477
478
479
480 Run #18
481 Matrix before normalization
482 1.00 6.00 9.00 0.00 10.00 6.00 1.00 6.00 0.00 5.00
483 Maximum number of threads available: 20

```



```

484 Mean of the first row: 5.00
485 Maximum number of threads available: 20
486 Standard Deviation of the first row: 3.16
487 Maximum number of threads available: 20
488 Mean of the last row: 5.00
489 Maximum number of threads available: 20
490 Standard Deviation of the last row: 3.16
491 Maximum number of threads available: 20
492
493 Matrix after normalization
494 -1.26 0.32 1.26 -1.58 1.58 0.32 -1.26 0.32 -1.58 0.00
495 Maximum number of threads available: 20
496 Mean of the first row: 0.00
497 Maximum number of threads available: 20
498 Standard Deviation of the first row: 1.00
499 Maximum number of threads available: 20
500 Mean of the last row: 0.00
501 Maximum number of threads available: 20
502 Standard Deviation of the last row: 1.00
503 Time taken for OpenMP code: 7.24 seconds.
504 -----
505
506
507
508 Run #19
509 Matrix before normalization
510 0.00 10.00 10.00 1.00 6.00 9.00 2.00 7.00 6.00 6.00
511 Maximum number of threads available: 20
512 Mean of the first row: 5.00
513 Maximum number of threads available: 20
514 Standard Deviation of the first row: 3.16
515 Maximum number of threads available: 20
516 Mean of the last row: 5.00
517 Maximum number of threads available: 20
518 Standard Deviation of the last row: 3.16
519 Maximum number of threads available: 20
520
521 Matrix after normalization
522 -1.58 1.58 1.58 -1.26 0.32 1.26 -0.95 0.63 0.32 0.32
523 Maximum number of threads available: 20
524 Mean of the first row: -0.00
525 Maximum number of threads available: 20
526 Standard Deviation of the first row: 1.00
527 Maximum number of threads available: 20
528 Mean of the last row: -0.00
529 Maximum number of threads available: 20
530 Standard Deviation of the last row: 1.00
531 Time taken for OpenMP code: 8.33 seconds.
532 -----
533
534
535
536 Run #20
537 Matrix before normalization
538 5.00 0.00 8.00 10.00 8.00 3.00 3.00 1.00 1.00 7.00
539 Maximum number of threads available: 20
540 Mean of the first row: 5.00
541 Maximum number of threads available: 20
542 Standard Deviation of the first row: 3.16
543 Maximum number of threads available: 20
544 Mean of the last row: 5.00
545 Maximum number of threads available: 20
546 Standard Deviation of the last row: 3.16
547 Maximum number of threads available: 20
548
549 Matrix after normalization
550 0.00 -1.58 0.95 1.58 0.95 -0.63 -0.63 -1.26 -1.26 0.63
551 Maximum number of threads available: 20
552 Mean of the first row: 0.00
553 Maximum number of threads available: 20
554 Standard Deviation of the first row: 1.00
555 Maximum number of threads available: 20
556 Mean of the last row: 0.00
557 Maximum number of threads available: 20
558 Standard Deviation of the last row: 1.00
559 Time taken for OpenMP code: 8.09 seconds.

```

---