МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе №4

по дисциплине «Параллельные алгоритмы»

Тема: Параллельное умножение матриц

Студент гр. 9303 _____ Халилов Ш.А.

Преподаватель _____ Сергеева Е.И.

Санкт-Петербург

2022

Цель работы.

Реализовать параллельный алгоритм умножения матриц и параллельный алгоритм “быстрого” умножения.

Задание.

Реализовать параллельный алгоритм умножения матриц.

4.1 Реализовать параллельный алгоритм умножения матриц. Исследовать масштабируемость выполненной реализации.

4.2 Реализовать параллельный алгоритм “быстрого” умножения матриц (Штрассена или его модификации).

- Проверить, что результаты вычислений реализаций 4.1 и 4.2 совпадают.

- Сравнить производительность с реализацией 4.1 на больших размерностях данных (порядка 10^4 – 10^6)

Выполнение работы.

Для параллельного умножения матриц с помощью нескольких потоков выполнена в файле utils.cpp. В этом файле реализована функция parallelSimpleMultiply, которая принимает ссылки на 2 вектора, первые два вектора будут умножатся. при перемножении матриц создается очередь задач, куда записывается номер строки исходных матриц. и для каждого потока передается функция multplyByRow и 4 аргумента этой функции, первые 2 это матрицы которые нужно перемножить, 3-й это исходная матрица для получения результата, 4-й аргумент это ссылка на очередь задач.

Реализация алгоритма Штрассена.

Для реализации алгоритма Штрассена была написана функция, которая:

1. Если количество размерность в матрице меньше 64, то происходит обычное умножение матриц.

2. Если предыдущие условия не выполняются, то функция вызывается рекурсивно в новом потоке.

Внутренняя функция реализует алгоритм штрассена. Схема алгоритма представлена на рисунке 1.

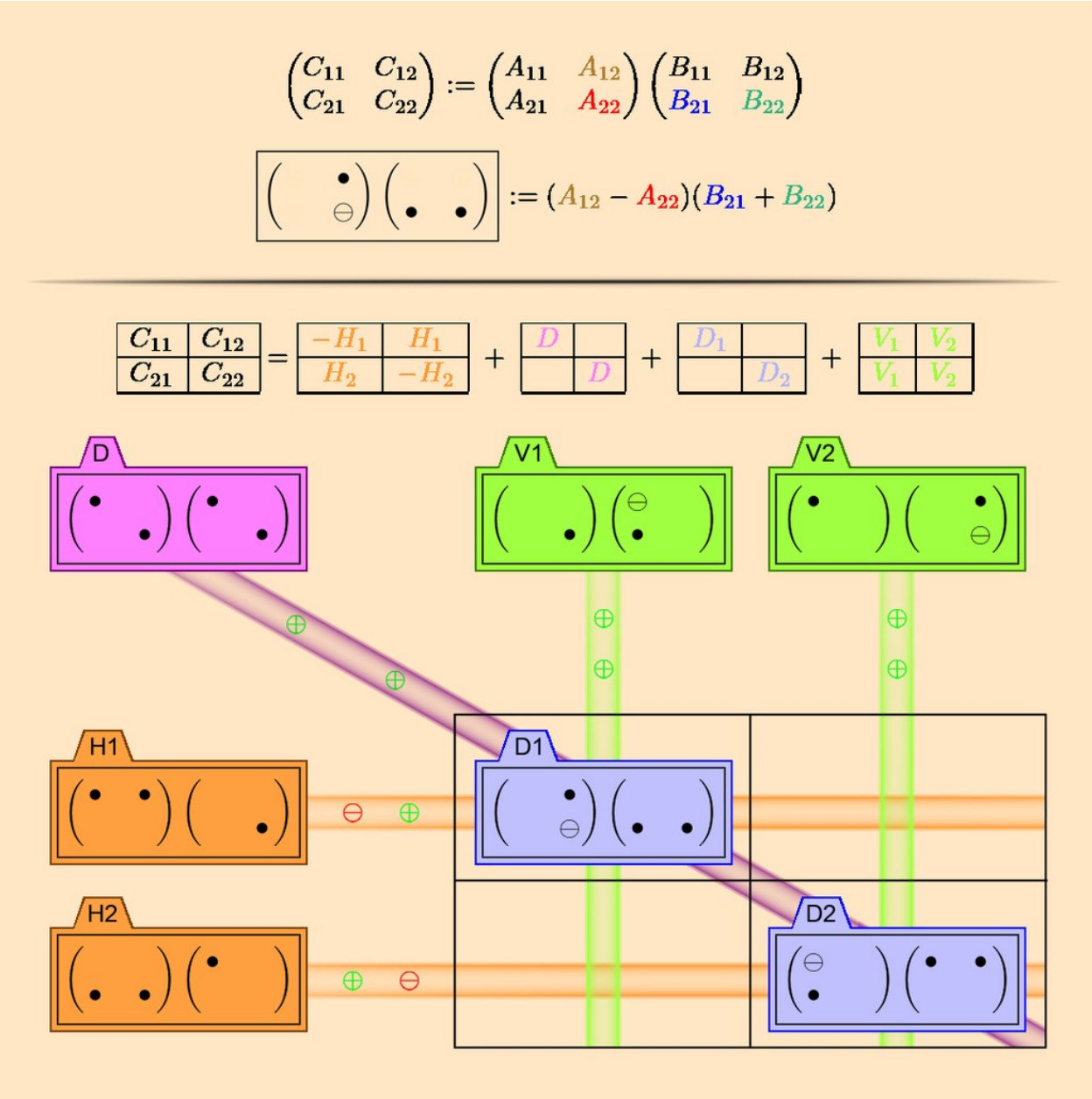

Рисунок 1 - Алгоритм Штрассена

Проверить, что результаты вычислений реализаций

Для проверки правильности алгоритма было сделано умножение матриц (представлено ниже) обоими алгоритмами

$$A = \begin{matrix} 0 & 4 & 6 & 4 & 7 & 6 & 9 & 1 & 5 & 9 & 5 & 9 & 4 & 1 & 2 & 3 \\ 0 & 9 & 0 & 2 & 4 & 2 & 2 & 9 & 6 & 5 & 6 & 7 & 6 & 1 & 4 & 2 \\ 9 & 0 & 5 & 2 & 0 & 4 & 3 & 8 & 1 & 7 & 8 & 0 & 3 & 3 & 8 & 9 \\ 4 & 8 & 7 & 0 & 5 & 4 & 4 & 3 & 9 & 1 & 9 & 2 & 3 & 3 & 2 & 8 \\ 9 & 4 & 8 & 6 & 5 & 2 & 4 & 9 & 3 & 0 & 9 & 0 & 0 & 5 & 5 & 0 \\ 6 & 4 & 9 & 0 & 0 & 8 & 5 & 5 & 0 & 0 & 1 & 4 & 9 & 8 & 7 & 7 \\ 1 & 5 & 7 & 2 & 2 & 4 & 9 & 3 & 3 & 2 & 3 & 2 & 9 & 5 & 8 & 4 \\ 0 & 1 & 4 & 0 & 4 & 0 & 2 & 1 & 2 & 8 & 4 & 2 & 4 & 3 & 4 & 8 \\ 6 & 3 & 9 & 6 & 8 & 9 & 9 & 6 & 5 & 2 & 4 & 6 & 9 & 5 & 5 & 9 \\ 8 & 6 & 6 & 1 & 3 & 1 & 2 & 0 & 2 & 1 & 8 & 7 & 6 & 0 & 2 & 0 \\ 8 & 7 & 1 & 4 & 1 & 5 & 7 & 7 & 4 & 5 & 7 & 2 & 1 & 4 & 1 & 4 \\ 5 & 2 & 7 & 5 & 5 & 6 & 0 & 4 & 1 & 8 & 5 & 0 & 7 & 4 & 5 & 6 \\ 5 & 9 & 5 & 7 & 9 & 8 & 6 & 5 & 1 & 2 & 8 & 7 & 0 & 7 & 5 & 4 \\ 7 & 3 & 3 & 5 & 3 & 3 & 7 & 1 & 0 & 7 & 4 & 7 & 3 & 4 & 2 & 3 \\ 2 & 1 & 2 & 6 & 7 & 1 & 3 & 2 & 7 & 5 & 0 & 8 & 8 & 9 & 7 & 2 \\ 5 & 1 & 2 & 3 & 3 & 6 & 4 & 5 & 2 & 7 & 9 & 4 & 1 & 3 & 6 & 6 \end{matrix}$$

$$B = \begin{matrix} 0 & 4 & 0 & 9 & 8 & 0 & 1 & 0 & 3 & 5 & 4 & 1 & 1 & 9 & 9 & 4 \\ 2 & 7 & 0 & 5 & 3 & 8 & 4 & 1 & 7 & 8 & 6 & 6 & 1 & 9 & 1 & 5 \\ 1 & 9 & 2 & 0 & 5 & 6 & 4 & 6 & 3 & 8 & 9 & 9 & 7 & 9 & 3 & 5 \\ 5 & 7 & 2 & 5 & 0 & 2 & 8 & 8 & 6 & 6 & 3 & 6 & 8 & 0 & 0 & 5 \\ 2 & 6 & 2 & 0 & 5 & 5 & 3 & 9 & 7 & 3 & 5 & 8 & 9 & 1 & 5 & 9 \\ 9 & 5 & 2 & 7 & 8 & 6 & 0 & 5 & 9 & 2 & 5 & 0 & 4 & 2 & 7 & 1 \\ 0 & 5 & 1 & 0 & 5 & 9 & 4 & 2 & 6 & 9 & 7 & 3 & 2 & 4 & 6 & 8 \\ 7 & 5 & 3 & 7 & 5 & 4 & 8 & 2 & 1 & 7 & 3 & 4 & 6 & 3 & 6 & 1 \\ 1 & 5 & 8 & 6 & 3 & 1 & 2 & 2 & 6 & 3 & 5 & 0 & 8 & 1 & 2 & 8 \\ 3 & 9 & 4 & 3 & 0 & 7 & 5 & 1 & 9 & 0 & 2 & 9 & 5 & 7 & 0 & 4 \\ 3 & 3 & 2 & 8 & 9 & 7 & 1 & 9 & 5 & 5 & 1 & 1 & 2 & 3 & 3 & 2 \\ 7 & 1 & 3 & 3 & 3 & 4 & 4 & 8 & 8 & 7 & 6 & 8 & 5 & 9 & 1 & 1 \\ 5 & 5 & 8 & 1 & 2 & 4 & 0 & 1 & 9 & 8 & 7 & 5 & 9 & 1 & 3 & 9 \\ 2 & 5 & 0 & 5 & 5 & 8 & 0 & 6 & 2 & 6 & 3 & 4 & 1 & 5 & 3 & 4 \\ 5 & 7 & 4 & 8 & 6 & 7 & 5 & 1 & 2 & 1 & 6 & 2 & 0 & 2 & 6 & 8 \\ 8 & 5 & 3 & 3 & 1 & 2 & 2 & 4 & 0 & 1 & 2 & 1 & 2 & 4 & 5 & 0 \end{matrix}$$

```
275 416 220 247 290 415 249 345 478 371 372 378 372 337 235 363
273 339 216 301 254 337 239 244 373 338 292 291 306 273 201 295
286 393 184 375 337 333 224 234 274 290 275 230 237 314 327 262
238 387 201 318 338 356 185 300 348 358 335 252 299 324 277 326
202 385 135 363 375 342 248 303 284 384 309 268 279 306 304 316
307 399 189 311 347 385 184 251 328 397 387 277 270 352 336 317
247 398 208 257 297 399 206 239 352 381 373 283 285 282 272 376
183 281 155 161 161 252 140 189 230 179 198 224 204 207 154 216
401 547 284 391 440 489 301 428 516 520 502 410 474 407 422 476
156 262 137 236 268 256 136 227 307 316 275 241 225 299 195 254
235 359 146 353 318 343 220 241 340 351 277 235 241 321 278 270
284 424 202 305 292 344 212 282 357 301 306 310 326 288 269 314
334 470 164 400 421 481 289 433 455 454 400 380 344 398 340 374
213 330 136 253 257 321 199 259 350 320 281 292 245 320 226 272
258 379 239 270 247 339 221 299 388 338 339 336 356 261 222 391
281 355 171 341 319 345 212 285 329 276 262 242 246 284 276 255

275 416 220 247 290 415 249 345 478 371 372 378 372 337 235 363
273 339 216 301 254 337 239 244 373 338 292 291 306 273 201 295
286 393 184 375 337 333 224 234 274 290 275 230 237 314 327 262
238 387 201 318 338 356 185 300 348 358 335 252 299 324 277 326
202 385 135 363 375 342 248 303 284 384 309 268 279 306 304 316
307 399 189 311 347 385 184 251 328 397 387 277 270 352 336 317
247 398 208 257 297 399 206 239 352 381 373 283 285 282 272 376
183 281 155 161 161 252 140 189 230 179 198 224 204 207 154 216
401 547 284 391 440 489 301 428 516 520 502 410 474 407 422 476
156 262 137 236 268 256 136 227 307 316 275 241 225 299 195 254
235 359 146 353 318 343 220 241 340 351 277 235 241 321 278 270
284 424 202 305 292 344 212 282 357 301 306 310 326 288 269 314
334 470 164 400 421 481 289 433 455 454 400 380 344 398 340 374
213 330 136 253 257 321 199 259 350 320 281 292 245 320 226 272
258 379 239 270 247 339 221 299 388 338 339 336 356 261 222 391
281 355 171 341 319 345 212 285 329 276 262 242 246 284 276 255
```

Рисунок 2 - результаты расчета.

Сравнить производительность с реализацией 4.1 на больших размерностях данных.

Результаты сравнения производительности Алгоритм Штрассена и обычное умножение матриц. представлена в табл. 1.

Размер матрицы	**Простой алгоритм умножения**	**Алгоритма Штрассена**
128	2.13762s	0.245102s
256	17.208s	1.78459s
512	138.514s	12.8118s
1024	-	101.89s
2048		783.581s

Таблица 1 — производительность времени умножения от размера матриц.

Как видно, с увеличением размера матрицы алгоритм Штрассена работает быстрее.

Выводы.

В ходе выполнения лабораторной работы было написана программа на языке программирования C++ для параллельного умножения потока матриц. Также был реализован алгоритм Штрассена.