Дослідження алгоритму обернення матриці Adjoint Determinant

Виконав: Федюченко Михайло

**Параметри:**

Data type = R (Big Decimal) ;

Matrix size: 128 – 4048 ;

Matrix Density = 100%, 10%

Accuracy = 64

**Порівняти час виконання на різній кількості ядер:**

#Core = 1/2/4

**Matrix size = 128, matrix density = 100%, #core=1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| leaf density/ leaf size | 0.50 | 0.10 | 0.03 | 0.01 |
| 16 | 389 | 435 | 380 | 428 |
| 32 | 326 | 332 | 379 | 373 |
| 64 | 313 | 348 | 331 | 315 |

**Matrix size = 128, matrix density = 10%, #core=1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| leaf density/ leaf size | 0.50 | 0.10 | 0.03 | 0.01 |
| 16 | 307 | 324 | 335 | 311 |
| 32 | 259 | 260 | 287 | 324 |
| 64 | 265 | 242 | 278 | 244 |

**Matrix size = 128, matrix density = 100%, #core=2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| leaf density/ leaf size | 0.50 | 0.10 | 0.03 | 0.01 |
| 16 | 801 | 837 | 930 | 790 |
| 32 | 784 | 921 | 771 | 808 |
| 64 | 831 | 534 | 954 | 860 |

**Matrix size = 128, matrix density = 10%, #core=2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| leaf density/ leaf size | 0.50 | 0.10 | 0.03 | 0.01 |
| 16 | 720 | 745 | 655 | 690 |
| 32 | 752 | 699 | 586 | 860 |
| 64 | 262 | 257 | 729 | 749 |

**Matrix size = 128, matrix density = 100%, #core=4**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| leaf density/ leaf size | 0.50 | 0.10 | 0.03 | 0.01 |
| 16 | 2155 | 1823 | 2056 | 2365 |
| 32 | 1727 | 1883 | 2004 | 1913 |
| 64 | 1128 | 1453 | 1341 | 1315 |

**Matrix size = 128, matrix density = 10%, #core=4**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| leaf density/ leaf size | 0.50 | 0.10 | 0.03 | 0.01 |
| 16 | 1563 | 1525 | 1478 | 1794 |
| 32 | 1230 | 1178 | 1358 | 1356 |
| 64 | 692 | 581 | 950 | 1072 |

**Висновки:** Для такого розміру матриці пришвидшення не відбувається на більшій кількості процесорів.

Оптимальні налаштування:

Core: **1**

Matrix Density: **10%**

Leaf density: **0.10**

Leaf Size: **64**

**Matrix size = 256 matrix density = 100%, #core=1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| leaf density/ leaf size | 0.50 | 0.10 | 0.01 | 0.005 |
| 16 | 1705 | 2021 | 2080 | 2265 |
| 64 | 1443 | 1237 | 1561 | 1313 |
| 128 | 1400 | 1369 | 1394 | 1421 |

**Matrix size = 256 matrix density = 10%, #core=1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| leaf density/ leaf size | 0.50 | 0.10 | 0.01 | 0.005 |
| 16 | 1169 | 1184 | 1202 | 1128 |
| 64 | 824 | 840 | 835 | 791 |
| 128 | 1004 | 1064 | 1106 | 1171 |

**Matrix size = 256 matrix density = 100%, #core=2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| leaf density/ leaf size | 0.50 | 0.10 | 0.01 | 0.005 |
| 16 | 2357 | 2547 | 2749 | 2346 |
| 64 | 3145 | 2124 | 2332 | 2146 |
| 128 | 2309 | 2190 | 2296 | 2376 |

**Matrix size = 256 matrix density = 10%, #core=2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| leaf density/ leaf size | 0.50 | 0.10 | 0.01 | 0.005 |
| 16 | 2050 | 1870 | 1792 | 2144 |
| 64 | 1567 | 1217 | 1898 | 1810 |
| 128 | 1160 | 1152 | 1583 | 1704 |

**Matrix size = 256 matrix density = 100%, #core=4**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| leaf density/ leaf size | 0.50 | 0.10 | 0.01 | 0.005 |
| 16 | 4905 | 5431 | 6312 | 5582 |
| 64 | 4622 | 4579 | 4319 | 4143 |
| 128 | 3949 | 3968 | 4176 | 4500 |

**Matrix size = 256 matrix density = 10%, #core=4**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| leaf density/ leaf size | 0.50 | 0.10 | 0.01 | 0.005 |
| 16 | 5092 | 5120 | 4567 | 4657 |
| 64 | 3255 | 2607 | 3649 | 3505 |
| 128 | 1587 | 1657 | 2852 | 3295 |

**Висновки:** Для такого розміру матриці пришвидшення не відбувається на більшій кількості процесорів. Для усіх налаштувань помітна кореляція

Оптимальні налаштування:

Core: **1**

Matrix Density: **10%**

Leaf density: **0.005**

Leaf Size: **64**

**Matrix size = 512** **matrix density = 100%, #core=1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| leaf density/ leaf size | 0.50 | 0.10 | 0.05 | 0.01 | 0.0025 |
| 64 | 7415 | 7856 | 7451 | 8168 | 7754 |
| 128 | 5693 | 5589 | 5797 | 5849 | 6076 |
| 256 | 5759 | 5604 | 5737 | 5741 | 5629 |

**Matrix size = 512** **matrix density = 10%, #core=1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| leaf density/ leaf size | 0.50 | 0.10 | 0.05 | 0.01 | 0.0025 |
| 64 | 5235 | 5356 | 5602 | 5695 | 5208 |
| 128 | 4202 | 4371 | 4412 | 4740 | 4352 |
| 256 | 4186 | 4038 | 3945 | 4472 | 4269 |

**Matrix size = 512** **matrix density = 100%, #core=2**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| leaf density/ leaf size | 0.50 | 0.10 | 0.05 | 0.01 | 0.0025 |
| 64 | 8741 | 8182 | 7833 | 8346 | 8201 |
| 128 | 6914 | 7509 | 7293 | 7600 | 7645 |
| 256 | 8385 | 7957 | 7724 | 7854 | 7224 |

**Matrix size = 512** **matrix density = 10%, #core=2**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| leaf density/ leaf size | 0.50 | 0.10 | 0.05 | 0.01 | 0.0025 |
| 64 | 6232 | 6161 | 6820 | 7187 | 6369 |
| 128 | 6133 | 5753 | 5678 | 5173 | 5592 |
| 256 | 4287 | 4164 | 6011 | 6134 | 5792 |

**Matrix size = 512** **matrix density = 100%, #core=4**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| leaf density/ leaf size | 0.50 | 0.10 | 0.05 | 0.01 | 0.0025 |
| 64 | 14220 | 15055 | 13272 | 12803 | 13709 |
| 128 | 12764 | 11823 | 13154 | 12295 | 11846 |
| 256 | 11114 | 11685 | 11524 | 11262 | 11005 |

**Matrix size = 512** **matrix density = 10%, #core=4**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| leaf density/ leaf size | 0.50 | 0.10 | 0.05 | 0.01 | 0.0025 |
| 64 | 11673 | 11943 | 12396 | 11458 | 11722 |
| 128 | 8933 | 9251 | 9520 | 8805 | 9705 |
| 256 | 5943 | 5877 | 8167 | 8601 | 7834 |

**Висновки:** Для такого розміру матриці пришвидшення не відбувається на більшій кількості ядер.

Оптимальні налаштування:

Core: **1**

Matrix Density: **10%**

Leaf density: **0.05**

Leaf Size: **256**

**Matrix size = 1024 matrix density = 100%, #core=1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| leaf density/ leaf size | 0.50 | 0.10 | 0.05 | 0.01 | 0.001 |
| 64 | 35126 | 35417 | 35542 | 35484 | 35369 |
| 256 | 34445 | 34130 | 34504 | 34673 | 34194 |
| 512 | 34327 | 33379 | 33932 | 34501 | 33740 |

**Matrix size = 1024 matrix density = 10%, #core=1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| leaf density/ leaf size | 0.50 | 0.10 | 0.05 | 0.01 | 0.001 |
| 64 | 23275 | 22978 | 23427 | 24097 | 23646 |
| 256 | 24097 | 23896 | 23670 | 23680 | 23355 |
| 512 | 23476 | 23810 | 23873 | 24130 | 23992 |

**Matrix size = 1024 matrix density = 100%, #core=2**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| leaf density/ leaf size | 0.50 | 0.10 | 0.05 | 0.01 | 0.001 |
| 64 | 31962 | 30946 | 32767 | 32845 | 35786 |
| 256 | 34251 | 36349 | 35615 | 35419 | 34657 |
| 512 | 38339 | 34186 | 34240 | 35502 | 34250 |

**Matrix size = 1024 matrix density = 10%, #core=2**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| leaf density/ leaf size | 0.50 | 0.10 | 0.05 | 0.01 | 0.001 |
| 64 | 24602 | 25784 | 24600 | 23721 | 24433 |
| 256 | 26183 | 24665 | 26541 | 24040 | 24099 |
| 512 | 24677 | 24371 | 25979 | 24413 | 24448 |

**Matrix size = 1024 matrix density = 100%, #core=4**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| leaf density/ leaf size | 0.50 | 0.10 | 0.05 | 0.01 | 0.001 |
| 64 | 48211 | 48724 | 49855 | 46234 | 46582 |
| 256 | 50132 | 48853 | 48094 | 49019 | 47457 |
| 512 | 66002 | 56304 | 62238 | 64935 | 59512 |

**Matrix size = 1024** **matrix density = 10%, #core=4**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| leaf density/ leaf size | 0.50 | 0.10 | 0.05 | 0.01 | 0.001 |
| 64 | 37922 | 41969 | 35776 | 38511 | 38336 |
| 256 | 39176 | 39002 | 39155 | 37922 | 39176 |
| 512 | 44022 | 42094 | 39136 | 38387 | 40181 |

**Висновки:** Частково помітно збільшення ефективності на двох ядрах на **13%** та з Matrix Density 10%

Оптимальні налаштування:

Core: **1**

Matrix Density: **10%**

Leaf density: **0.05**

Leaf Size: **256**

**Matrix size = 2048 matrix density = 100%, #core=1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| leaf density/ leaf size | 0.50 | 0.10 | 0.05 | 0.01 | 0.001 |
| 64 | 271994 | 270045 | 269032 | 270618 | 272198 |
| 512 | 291347 | 290911 | 288394 | 287446 | 285584 |
| 1024 | 266317 | 261663 | 269559 | 268944 | 267124 |

**Matrix size = 2048 matrix density = 10%, #core=1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| leaf density/ leaf size | 0.50 | 0.10 | 0.05 | 0.01 | 0.001 |
| 64 | 174350 | 176685 | 179245 | 176942 | 176064 |
| 512 | 181515 | 182526 | 178371 | 179178 | 178862 |
| 1024 | 173876 | 175388 | 172950 | 172651 | 172636 |

**Matrix size = 2048 matrix density = 100%, #core=2**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| leaf density/ leaf size | 0.50 | 0.10 | 0.05 | 0.01 | 0.001 |
| 64 | 183374 | 193883 | 184083 | 185426 | 181825 |
| 512 | 203310 | 195496 | 199031 | 197609 | 193184 |
| 1024 | 222317 | 223398 | 224277 | 223694 | 222324 |

**Matrix size = 2048 matrix density = 10%, #core=2**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| leaf density/ leaf size | 0.50 | 0.10 | 0.05 | 0.01 | 0.001 |
| 64 | 134456 | 133430 | 134277 | 133788 | 134767 |
| 512 | 147200 | 150593 | 140429 | 139229 | 141937 |
| 1024 | 178723 | 174700 | 151282 | 150329 | 153128 |

**Matrix size = 2048 matrix density = 100%, #core=4**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| leaf density/ leaf size | 0.50 | 0.10 | 0.05 | 0.01 | 0.001 |
| 64 | 247662 | 263790 | 241865 | 256313 | 230614 |
| 512 | 244575 | 244656 | 243863 | 249958 | 257293 |
| 1024 | 369420 | 422431 | 372376 | 427596 | 443398 |

**Matrix size = 2048 matrix density = 10%, #core=4**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| leaf density/ leaf size | 0.50 | 0.10 | 0.05 | 0.01 | 0.001 |
| 64 | 188115 | 215320 | 200727 | 176063 | 180769 |
| 512 | 200346 | 203460 | 187722 | 184816 | 189135 |
| 1024 | 309504 | 281812 | 323421 | 314439 | 259439 |

**Висновки:** Помітно збільшення ефективності на **25%** на двох ядрах в порівнянні з одним та на **9%** для чотирьох

Оптимальні налаштування:

Core: **2**

Matrix Density: **10%**

Leaf density: **0.10**

Leaf Size: **64**

**Matrix size = 4096 matrix density = 100%, #core=1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| leaf density/ leaf size | 0.10 | 0.01 | 0.001 | 0.0005 |
| 64 | 2746951 | 2734798 | 2718052 | 2723446 |
| 1024 | 2675234 | 2797223 | 2815957 | 2770616 |
| 2048 | 2653358 | 2662315 | 2620597 | 2673155 |

**Matrix size = 4096 matrix density = 10%, #core=1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| leaf density/ leaf size | 0.10 | 0.01 | 0.001 | 0.0005 |
| 64 | 1723775 | 1714891 | 1742904 | 1754202 |
| 1024 | 1702472 | 1685477 | 1699887 | 1659156 |
| 2048 | 1606597 | 1638479 | 2206822 | 1610034 |

**Matrix size = 4096 matrix density = 100%, #core=2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| leaf density/ leaf size | 0.10 | 0.01 | 0.001 | 0.0005 |
| 64 | 1575721 | 1578776 | 2967862 | 1569646 |
| 1024 | 1670614 | 1648416 | 1735891 | 1668763 |
| 2048 | 2384423 | 2389041 | 2413610 | - |

**Висновки:** Було протестовано не до кінця, через брак пам’яті, але з отриманих даних для даного розміру матриці відбулося величезне збільшення на **48%** ефективності при використанні двох ядер

Оптимальні налаштування:

Core: **2**

Matrix Density: **100%**

Leaf density: **0.0005**

Leaf Size: **64**

**—----------------**

**Загальні висновки:** Використання паралельного обчислення на декількох ядрах буде ефективним у разі великої за розміром матриці. У разі обчислення невеликих матриць для даного завдання - не варто запускати більше ніж на одному ядрі. Також для великих матриць швидкість виконання при 10% Matrix density були швидшим. Тестування алгоритму на чотирьох процесорах з різними розмірами матриць виконувалося близько трьох днів на AWS Workspace Linux. На жаль не вдалося до кінця протестувати для матриці 4096, але попередні результати вже передбачають що час виконання буде зменшуватися при використання більшої кількості ядер (2-4).

**Таблиця оптимальних налаштувань**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Matrix size/Matrix Density | 128 | 256 | 512 | 1024 | 2048 | 4096 |
| 100% | Core: **1**  L density: **0.50**  L Size: **64** | Core: **1**  L density: **0.10**  L Size: **64** | Core: **1**  L density: **0.10**  L Size: **128** | Core: **2**  L density: **0.10**  L Size: **64** | Core: **2**  L density: **0.001**  L Size: **64** | Core: **2**  L density: **0.0005**  L Size: **64** |
| 10% | Core: **1**  L density: **0.10**  L Size: **64** | Core: **1**  L density: **0.005**  L Size: **64** | Core: **1**  L density: **0.05**  L Size: **256** | Core: **1**  L density: **0.05**  L Size: **256** | Core: **2**  L density: **0.10**  L Size: **64** | Core: **1**  L density: **0.10**  L Size: **2048** |