AGH Akademia Górniczo-Hutnicza
im. Stanisława Staszica w Krakowie

Faktoryzacja Choleskiego

19 maja 2024

Jakub Szaredko

1 Architektura procesora

Zadanie dostosowywałem pod urządzenie z wbudowanym procesorem Apple M1 Pro. Specyfikacja techniczna tego procesora nie jest publicznie dostępna, bazowałem na Wikipedii, gdzie zostały zapisane podstawowe parametry jednostki. Niestety, nie mogłem się doszukać liczby operacji zmiennoprzecinkowych na sekundę, jedyne informacje jakie znalazłem dotyczyły zintegrowanej karty graficznej.

Producent	Apple
Model	M1 Pro
Mikroarchitektura	Firestorm i Icestorm
Architektura instrukcji	ARMv8.5-A
Technologia	5nm
Liczba rdzeni	8
Taktowanie maksymalne	3.22 GHz
Cache L1	320 / 192 KB
Pamięć RAM	16 GB

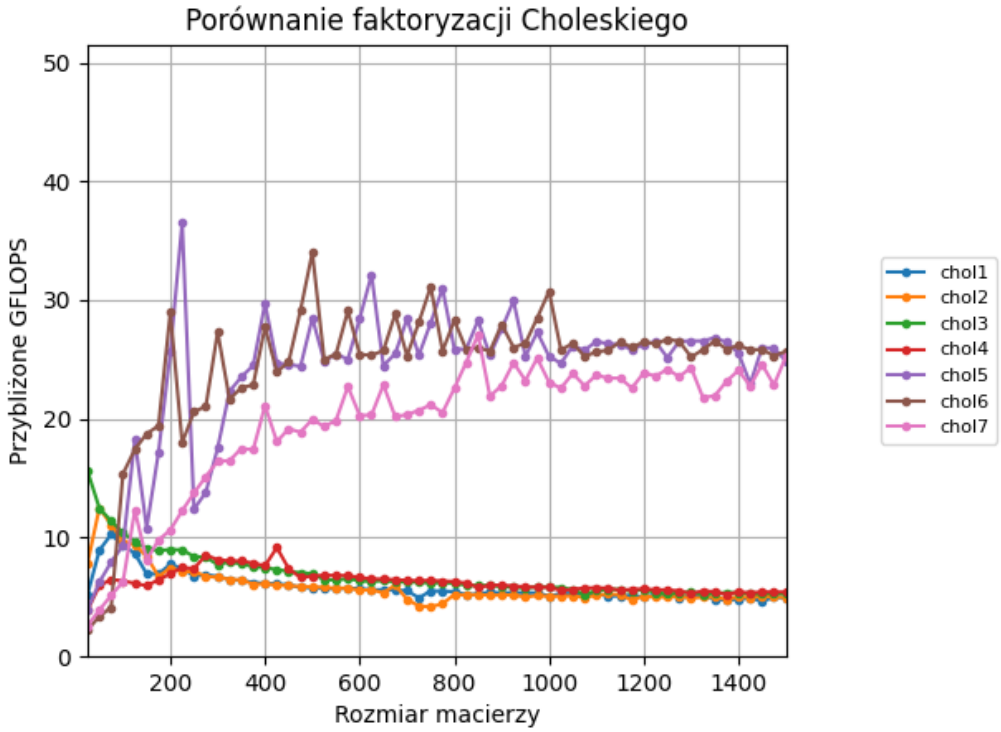
2 Dostosowanie kompilacji

1. Dodanie parametru `-march armv8.5-a`.
2. Wykorzystanie forsownie kompilatora GCC za pomocą komendy `gcc-13`, **macOS linkuje Clang ze standardową komendą** `gcc`.

3 Zastosowane optymalizacje

1. `chol1`: Bazowa funkcja bez żadnych dodatkowych optymalizacji.
2. `chol2`: Przeniesienie pętli dwuwymiarowej do oddzielnej funkcji.
3. `chol3`: Cofnięcie zmian z `chol2`, zamienienie najbardziej wykorzystywanych zmiennych na zmienne rejestrowe.
4. `chol4`: Dodanie operacji odejmowania w bloku o rozmiarze 8.
5. `chol5`: Dodanie zmiennych wektorowych, bez jawnego zasugerowania, że to zmienne rejestrowe.
6. `chol6`: Zasygnalizowanie, że zmienne wektorowe powinny trafić do rejestru.
7. `chol7`: Blok o rozmiarze 16.

4 Wyniki



Wykres 1: Wyniki poszczególnych wersji programu w zależności od wielkości macierzy

5 Podsumowanie

1. Najefektywniejszymi rozwiązaniami są te, które wykorzystały instrukcje wektorowe w blokach o rozmiarze 8.

2. Zwiększenie rozmiaru bloku do 16 spowodowało zauważalne zmniejszenie się wydajności.

3. Słowo kluczowe `register` dla zmiennych typu wektorowego nie ma znaczenia, zmienne automatycznie zostały potraktowane jako rejestrowe lub wręcz przeciwnie.

4. Zmienne rejestrowe dla rozwiązania `chol3` najbardziej były wpływowe dla małych rozmiarów macierzy wejściowych, dla macierzy nie mniejszych niż 300 wydajność jest taka sama. Możliwe jest to spowodowanym tym, że dla większych wielkości kompilator automatycznie przeniósł zmienne do rejestru.