Pomiary efektywności dla komputerów z procesorami AMD 10 h

Na podstawie dokumentacji AMD
opracował: Rafał Walkowiak

listopad 2015, zmiany listopad 2016

Liczniki zdarzeń a program profilujący

- Procesory AMD 10h wyposażone są w 4 ***liczniki wydajności*** przeznaczone do zliczania zdarzeń (w badanym okresie) spowodowanych przez aplikacje użytkownika i system operacyjny: liczba cykli CPU, liczba zatwierdzonych instrukcji, braków trafień do pp i podobnych zdarzeń.
- Program profilujący konfiguruje liczniki - określa jakie **zdarzenie** oraz przy jakich **warunkach dodatkowych** ma być zliczane i badane.

Metody pomiaru wydajności

- **Podejście klamrowe (ang. caliper mode)** – odczyt wartości licznika zdarzeń przed wejściem i po zakończeniu przetwarzania w krytycznym efektywnościowo fragmencie kodu. W wyniku odjęcia wartość zmierzonej „po” od wartości zmierzonej „przed” uzyskujemy liczbę zdarzeń za których wystąpienie odpowiedzialny jest testowany kod. W podejściu tym:
  - nie ma możliwości pomiaru dystrybucji mierzonych zdarzeń w badanym obszarze,
  - nie ma możliwości powiązania wystąpienia zdarzenia z powodującą je instrukcją – NIE WIADOMO JAKA PRZYCZYNA ZDARZENIA
- **Podejście próbkowania wg licznika wydajności** – licznik wydajności ładowany jest wartością limitu lub progu. Zliczenie określonej w ten sposób liczby zdarzeń powoduje wywołanie procedury przerwania dla obsługi zdarzenia, która zapisuje:
  - typ zdarzenia, ID procesu, ID wątku i IP – ZNANY LICZNIK INSTRUKCJI W OKOLICY instrukcji BĘDĄCEJ PRZYCZYNĄ ZDARZENIA.

Podejście próbkowania wg licznika wydajności

- Na podstawie zebranych próbek narzędzia budują histogram występowania poszczególnych zdarzeń w kodzie.
- Ze względu na wagę statystyczną uzyskanych pomiarów oraz działania uboczne procedury zbierania próbek ważne jest właściwe określenie progu zliczania. Wielkość ta jest uwarunkowana z jednej strony minimalizacją narzutu czasu zbierania próbek i ingerencji w zasoby systemu (pp, TLB, historię rozgałęzień kodu), a z drugiej strony wymaganą rozdzielczością pomiaru i wagą statystyczną wyniku pomiaru. Przykładowe wartości progów to 50 tyś/500tyś w zależności od typu zdarzenia - częstości występowania zdarzeń.
- Pomiar jest obarczony „poślizgiem" – różnica między rozkazem powodującym raportowane zdarzenie, a rozkazem, którego adres jest zapisywany – pierwszym wykonywanym po powrocie z procedury obsługi przerwania. Ze względu na dynamiczne wykonywanie rozkazów informacja o lokalizacji przyczyny wystąpienia zdarzenia jest tylko przybliżona. Informacja o liczbie zdarzeń jest zatem bardziej adekwatna dla zbioru, niż pojedynczej instrukcji.

Zdarzenia i wskaźniki

- Liczby zdarzeń raportowane przez program nie uwzględniają progów zliczania. Wskaźniki – uwzględniają progi (uwaga – możliwe błędy).
- W tej prezentacji: dla miar jakości podano koncepcyjne wzory – progi zliczania nie są uwzględniane; w niektórych przypadkach dla uzyskania właściwej wartości parametru próg jest podany we wzorze.
- Waga zjawiska może zostać oceniona na podstawie porównania liczby jego wystąpień z liczbą zrealizowanych w określonym czasie instrukcji – wyrażona stosunkiem tych wartości (wskaźniki – ang. rate).
- Wskaźniki pozwalają na oceną skali zjawiska:
  - **tysiąc braków trafień na 1mln instrukcji** nie jest niepokojące,
  - natomiast **tysiąc braków trafień na 10 tysięcy instrukcji** stanowi duży problem efektywnościowy.

Efektywność wykorzystania procesora

- Instrukcje na cykl IPC / odwrotność - cykle na instrukcję CPI
- IPC jest miarą równoległości dla poziomu przetwarzania instrukcji
- Wyznaczenie wymaga określenia liczby:
  - CPU Clocks Not Halted - CPU_clocks
  - Retired Instructions - Ret_instructions
- IPC = Ret_instructions/ CPU_clocks
- CPI = CPU_clocks/ Ret_instructions
- Niska wartość IPC wskazuje na obecność czynników zmniejszających efektywność: słaba przestrzenna lub czasowa lokalność dostępów, błędy predykcji dla rozgałęzień kodu, wyjątki FPU
- Informacje o dystrybucji w regionach kodu wielkości CPU_cloks i Ret_instructions pozwala na określenie kodu, który wymaga dużej ilości czasu lub często wykonywanego kodu

Efektywność dostępu do pamięci

Przepustowość odczytu, zapisu, dostępu do DRAM

Mierzone zdarzenia:

System_read, System_write, DRAM_accesses, (L3 miss)

Wyznaczane wskaźniki:

- Ilość danych czytanych z RAM = System_read * próg*64 [bajty]
- Ilość danych zapisywanych do RAM = System_write * próg*16 [bajty]
- Wielkość transferu procesor RAM (dwukierunkowego) = DRAM_access * próg*64
- Data bandwidth (B/s)= bytes_transfered/seconds
  - rdzenie współdzielą dostępną przepustowość DRAM,
  - procesor Phenom posiada dwa kontrolery pamięci
  - testy specyficzne dla architektury wyznaczają maksymalną dostępną przepustowość pamięci

Efektywność dostępu do pamięci – pamięć podręczna danych L1

- liczba braków trafień do pp danych
  - DC_misses = DC_refills_L2 + DC_refills_sys
  - Żądania obsłużone przez L2 i przez pamięć systemową
- wskaźnik żądań dostępu:
  - Data cache request rate = DC_accesses / Ret_instructions
- wskaźnik braku trafień
  - Data cache miss rate = DC_Misses / Ret_instructions
- stosunek braku trafień:
  - Data cache miss ratio = DC_Misses / DC_accessess
- ważna miara: DC_refills_sys / DC_misses
  - ze względu na >90 cykli dostęp z pamięci operacyjnej i 12 cykli z L2 cache .

Efektywność dostępu do pamięci – pamięć podręczna danych

Kategorie braku trafień do pp

- Braki trafień pierwszego dostępu (compulsory misses)– pierwsze odwołanie do jednostki danych, poprawa – wyprzedzające pobranie danych.
- Braki trafień wynikające z pojemności pp (capacity misses) – poprawa - zmniejszenie wykorzystywanej przestrzeni danych (zagęszczenie danych).
- Braki trafień wynikające z konfliktów (conflict misses – odwołanie do linii danych po jej unieważnieniu lub usunięciu), poprawa - przesunięcie pozycji danych do pozycji nie powodującej konfliktu dostępu – usunięcie false sharing lub niekorzystnego odstępu równego wielkości sekcji wielosekcyjnej pp.

Efektywność dostępu do pamięci – pamięć podręczna L2/L3

- wskaźnik żądań dostępu do pp L2
  - L2 request rate = (L2_requests + L2_fill_write) / Ret_instructions
  - L2_fill_write dostęp wtórny – zapis danych w L2 powodowany usunięciem ich z pp L1
- Wskaźnik/stosunek braku trafień do pp L2
  - L2 miss rate = L2_misses / Ret_instructions
  - L2 miss ratio = L2_misses / (L2_requests + L2_fill_write)
- Analogicznie L3_requests, L3_misses
- Pamięci (L1i L2) pracują w trybie wyłącznym lub prawie wyłącznym (L3)
- Pamięci są zunifikowane – pamięci danych i instrukcji

Dostęp do pamięci – translacja adresu danych

- wskaźnik żądań dostępu do L1 DTLB
  - L1 DTLB request rate = DC_accesses / Ret_instructions
  - Każdy dostęp (odczyt, zapis) do pp to dostęp do TLB
- stosunek braku trafień do L1 DTLB
  - L1 DTLB miss ratio = (DTLB_L1M_L2H + DTLB_L1M_L2M) / DC_accesses
  - Uwzględnia trafienia lub brak trafienia do DTLB L2
- wskaźnik żądań dostępu do L2 DTLB
  - L2 DTLB request rate = L1 DTLB miss rate
  - Brak trafienia do DTLB L1 powoduje odwołanie do DTLB L2
- stosunek braku trafień do L2 DTLB
  - L2 DTLB miss ratio = DTLB_L1M_L2M / (DTLB_L1M_L2H + DTLB_L1M_L2M)
- W przypadku strony wirtualnej o rozmiarze 4KB jej przeglądanie sekwencyjne powoduje wskaźnik 1/L2 DTLB miss rate na poziomie 4 tyś zatem przykładowo wartość wskaźnika 100 świadczy o problemie efektywnościowym w lokalności przestrzennej dostępu do danych.
- Analogicznie miary dla translacji adresu instrukcji.

Inne miary efektywności przetwarzania

- Miary dotyczące przekazywania sterowania:
  - rozgałęzień, wywołań procedur: liczba, predykcje
- Przypadki szczególne: dostępy niewyrównane do rozmiaru linii pp, operacje i wyjątki FPU

Analiza i optymalizacja kodu za pomocą analizatora wydajności CODE XL

Analizator wydajności AMD CodeXL

- Dostępny bezpłatnie dla Windows i Linux
- Pozwala na 4 tryby pracy – oceny wydajności:
  - Bazujące na upływie czasu – próbkowanie aplikacji ze stałą częstością, zapisywanie licznika rozkazów i określanie statystycznie najbardziej czasochłonnych części kodu w postaci histogramu (time-based profiling)
  - Bazujące na zdarzeniach – określenie zdarzeń, których (orientacyjne) miejsce występowania ma być mierzone, określenie progu zliczania dla każdego badanego zdarzenia (tylko procesory AMD) (event-based profiling)

CodeXL cd

- Próbkowanie pobrań kodu i raportowanie powiązanych zdarzeń (tylko procesory AMD) (instruction-base fetch sampling)
- Próbkowanie operacji procesora pod względem wywoływanych zdarzeń (tylko procesory AMD) (instruction-base operation sampling)

Działanie analizatora wydajności

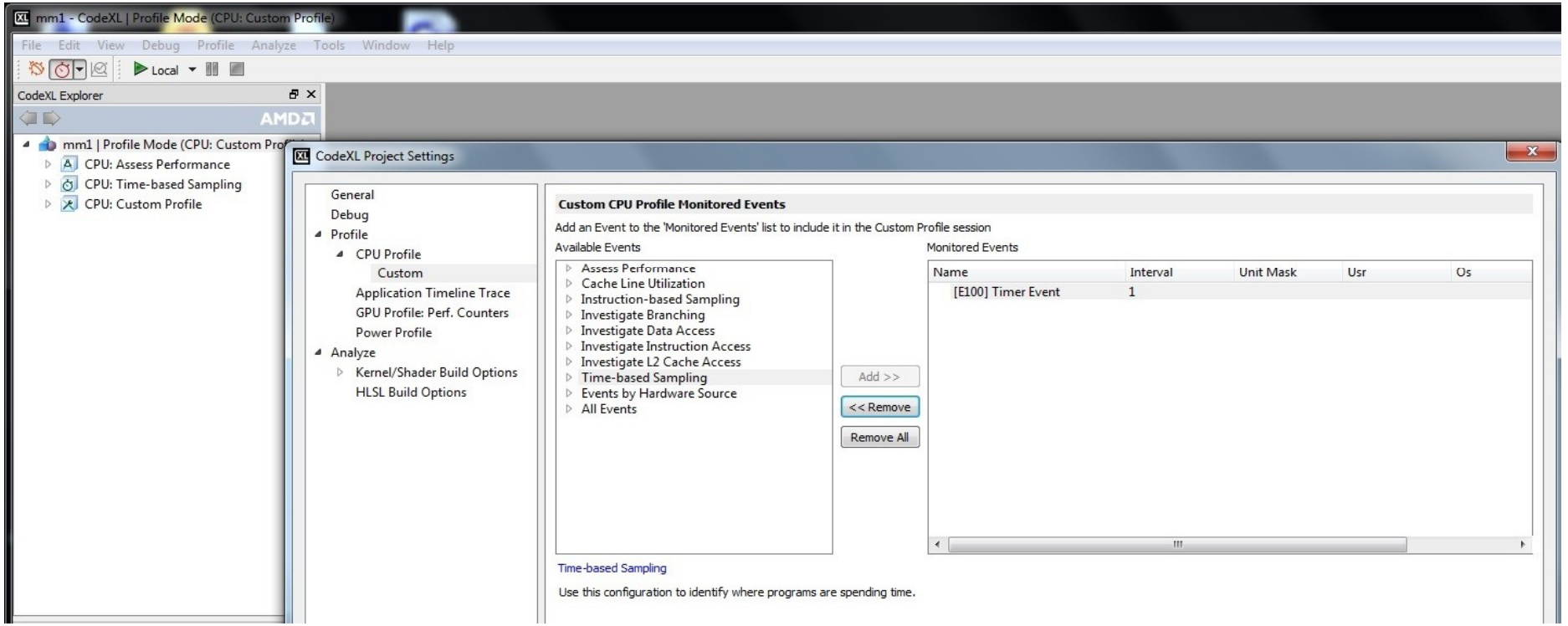
- Program oceny wydajności pracuje w ramach projektów składających się z sesji.
- Dla każdej sesji należy określić:
  - badaną aplikację i katalog roboczy
  - rodzaj realizowanej oceny wydajności,
  - czas rozpoczęcia i zakończenia zbierania danych,
  - częstotliwość próbkowania lub inne parametry w zależności od trybu pracy
  - maskę powinowactwa wątków.
- Przygotowana sesja oceny wydajności może zostać uruchomiona. Efektem uruchomionia jest dodanie nowej sesji w oknie zarządzania projektem. Wybór tej sesji powoduje wyświetlenie wyników sesji profilowania.
- Wyniki prezentowane są w grupach informacji charakterystycznych do realizowanego typu sesji. Można obserwować zbierane wyniki w wartościach bezwzględnych lub w udziale procentowym dla poszczególnych modułów oprogramowania, procesów i rdzeni. Dostępne wyniki oceny można filtrować i prezentować wyrywkowo – sposób prezentacji określa się w oknie dialogowym zarządcy widoku. Dane zebrane dla poszczegnych modułów kodu można analizować z rozbiciem na informacje dla linii kodu źródłowego i kodu asemblera.

Ocena bazująca na upływie czasu

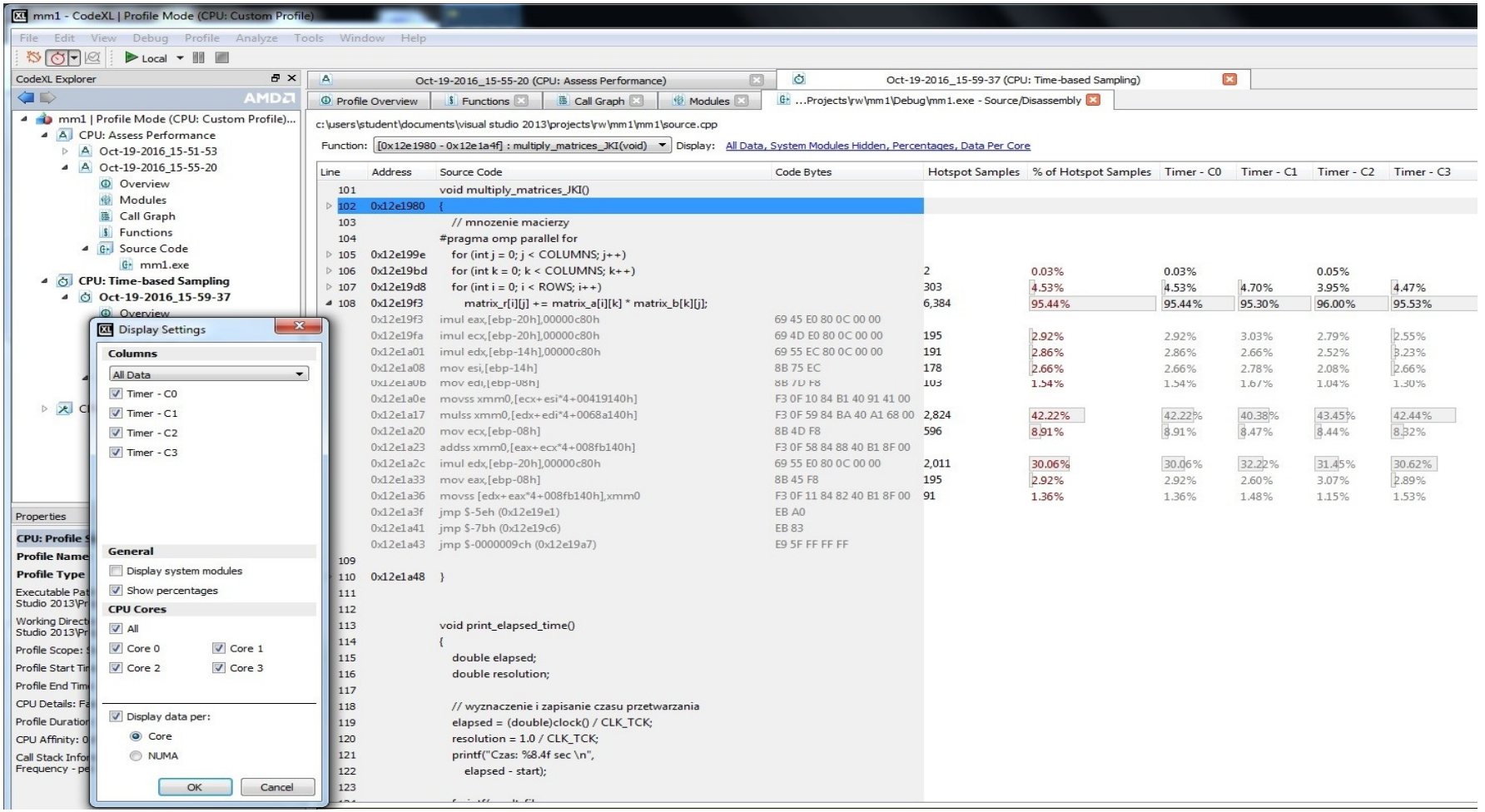
- Program oceny wydajności zbiera więcej próbek w obszarach kodu, w których program spędza więcej czasu gdyż pobranie próbki w tym obszarze jest bardziej prawdopodobne. Wynikowy histogram określa rozkład próbek, a wysokie wartości wskazują na czasochłonne fragmenty kodu.
- Standardowy okres próbkowania przetwarzania w CodeXL wynosi 1 ms i może zostać zmieniony poprzez:

Profile/profile settings/CPU profile/custom.

Ocena efektywności na upływie czasu konfiguracja



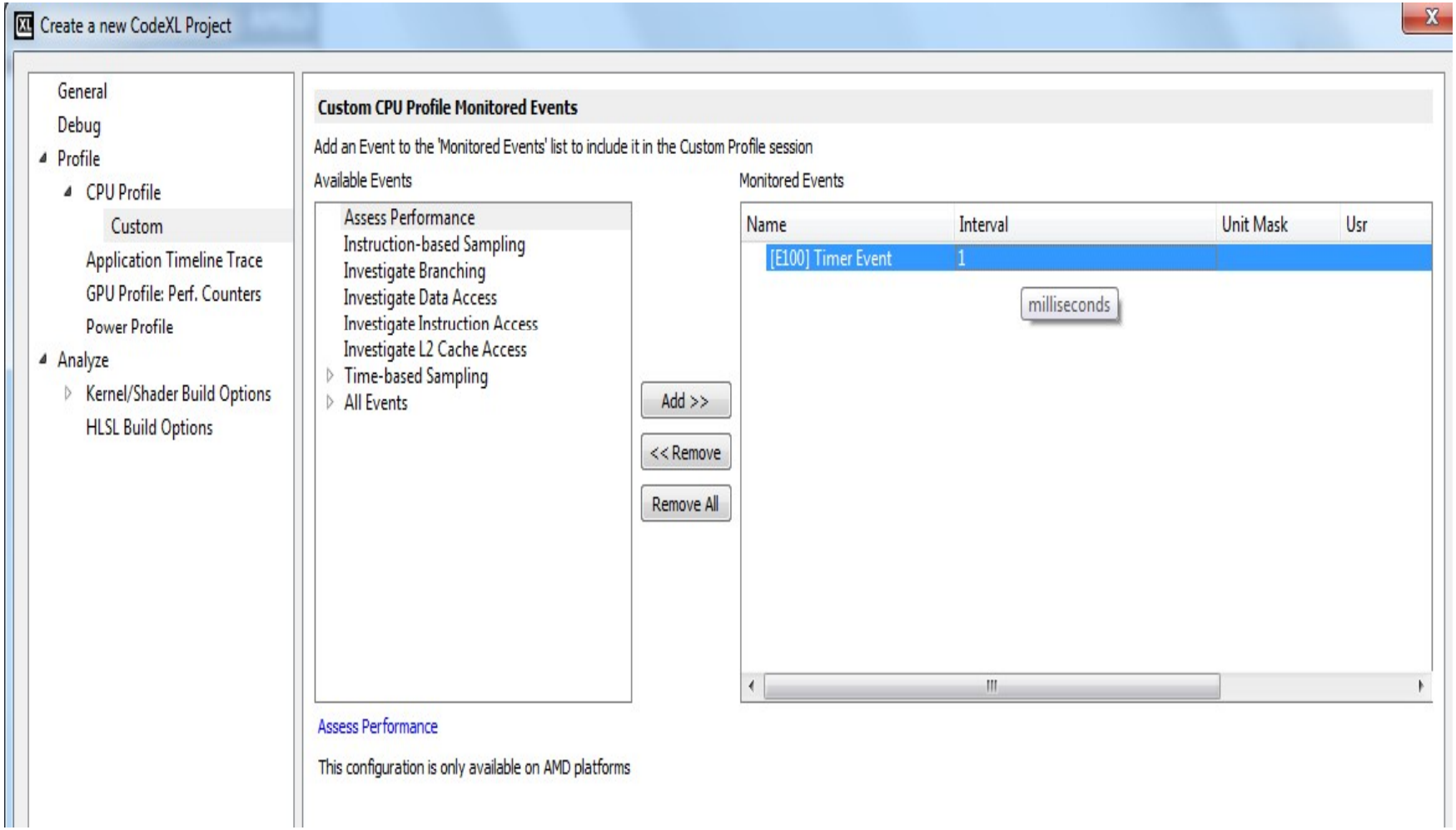
Ocena efektywności oparta na upływie czasu wyniki pomiaru



Prezentacja liczby próbek (miara upływu czasu) zebranych w ramach poszczególnych rdzeni dla przetwarzania modułów kodu .

Profile/profile settings/CPU profile/custom

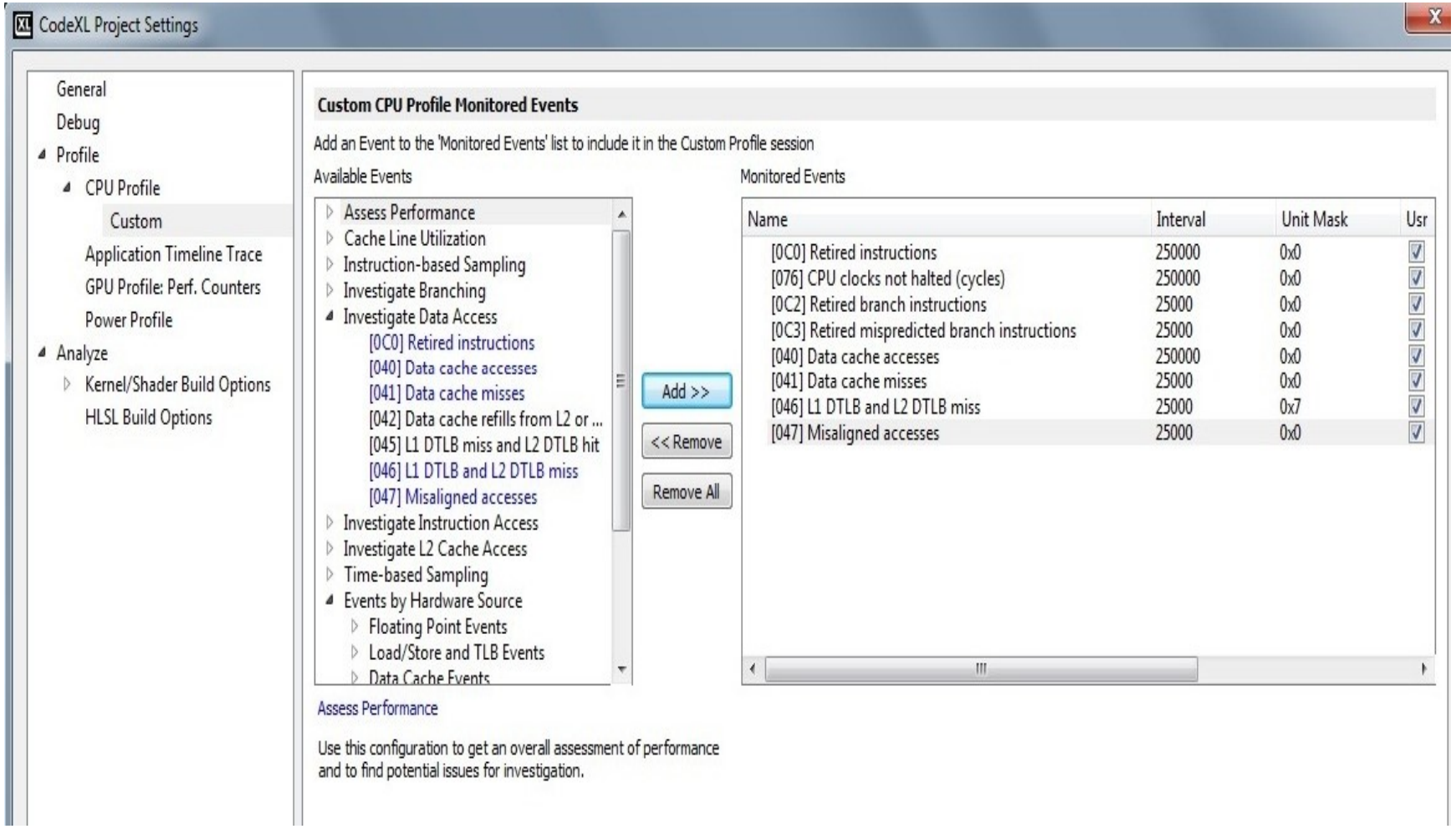
określanie eksperymentu,
poznawanie parametrów predefiniowanego eksperymentu



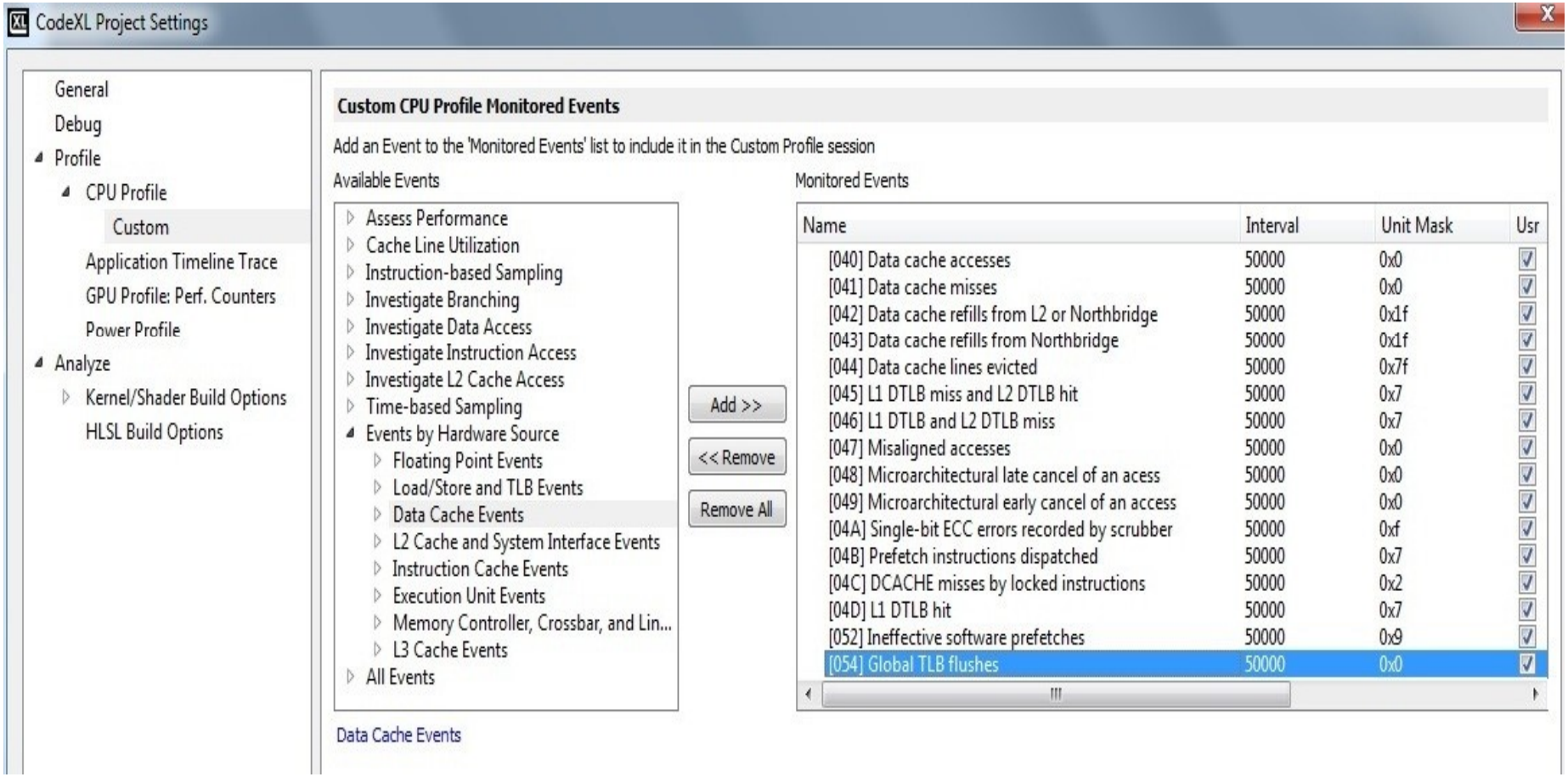
Ocena bazująca na zdarzeniach

- Możliwy jest wybór konfiguracji standardowej projektu oceny lub określenie własnego zestawu badanych zdarzeń (custom profile).
- Predefiniowane konfiguracje dotyczą:
  - Assess performance - Wydajność dostępu do kodu i realizacji rozgałęzień,
  - Investigate data access – analiza dostępu do pp danych i DTLB,
  - Investigate L2 cache access – analiza dostepu do pp L2,
  - Investigate instruction access – analiza dostępu do pp instrukcji i ITLB,
  - Investigate branching – analiza realizacji kodu w rozgałęzieniach wraz z predykcją
- Predefiniowana konfiguracja zawiera wybór zliczanych zdarzeń wraz z krotnością ich wystąpienia powodującą próbkowanie przetwarzania.
- Niektóre zdarzenia umożliwiają określenie warunków dodatkowych związanych z ich kwalifikacją do zliczenia – wyboru warunków dodatkowych można dokonać również w oknie edycji konfiguracji zdarzeń.
- Możliwa jest prezentacja wyników oceny na poziomie modułów kodu, linii kodu źródłowego i asemblera. Możliwe jest tematyczne filtrowanie zebranych wyników i wyświetlanie parametrów skonsolidowanych.

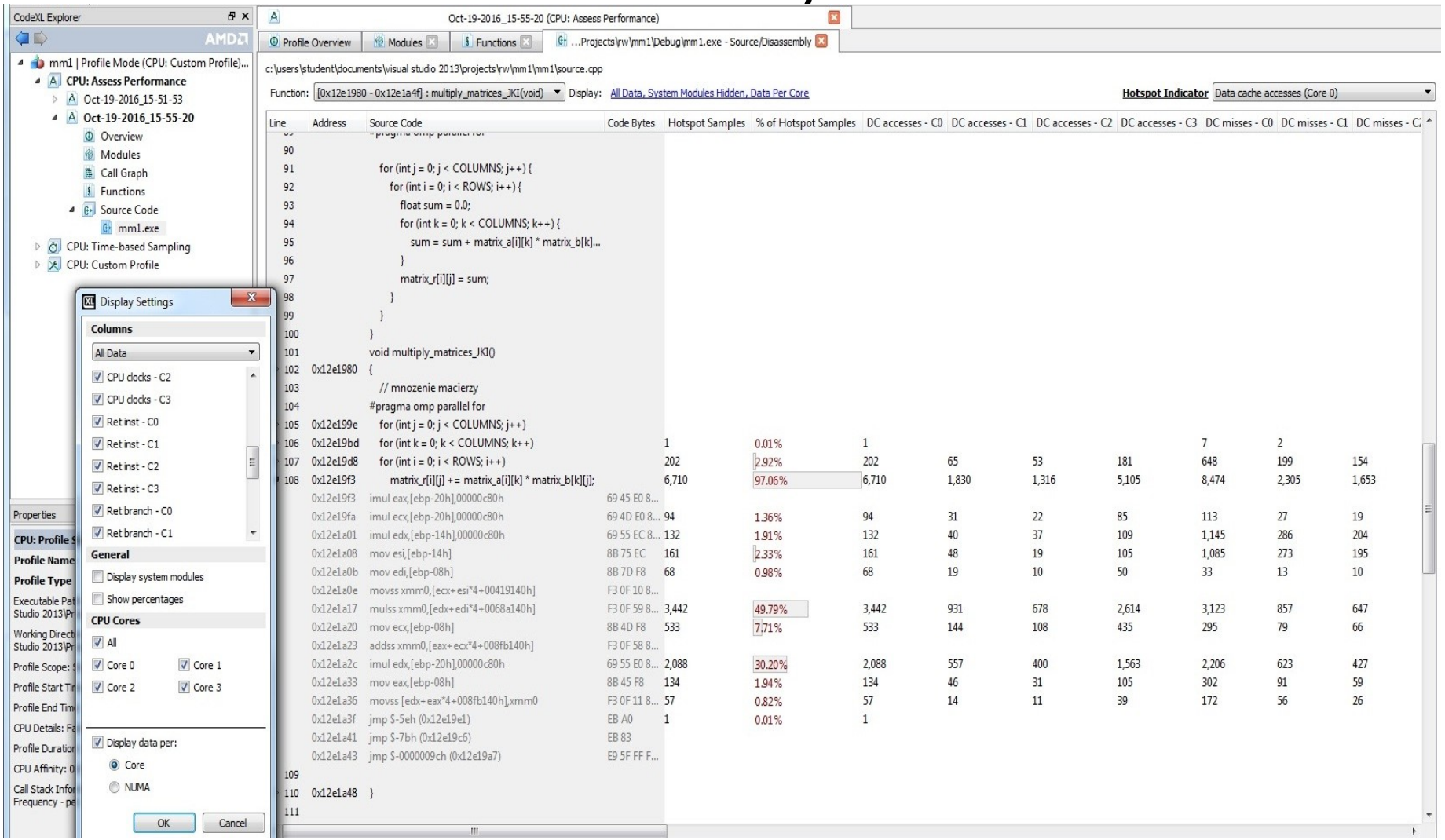
Ocena efektywności bazująca na zliczaniu zdarzeń – event based profiling



Ocena efektywności bazująca na zliczaniu zdarzeń – event based profiling



Ocena efektywności bazująca na zliczaniu zdarzeń –wyniki



Próbkowanie pobrań kodu (fetch sampling - instruction-base samp.)

Parametrem sesji oceny wydajności jest okres próbkowania.

Osiągnięcie określonego progu zliczania pobrań instrukcji powoduje monitorowanie aktualnej operacji pobrania.

Zakończenie lub przerwanie pobrania powoduje zapisanie próbki na temat zrealizowanej operacji pobrania. Zapisywane a następnie raportowane informacje dotyczą:

- identyfikatora procesu,
- adresu wirtualnego pobranej instrukcji procesu,
- trafienia do ITLB L1 i L2,
- trafienie do pp IC,
- opóźnienie pobrania.

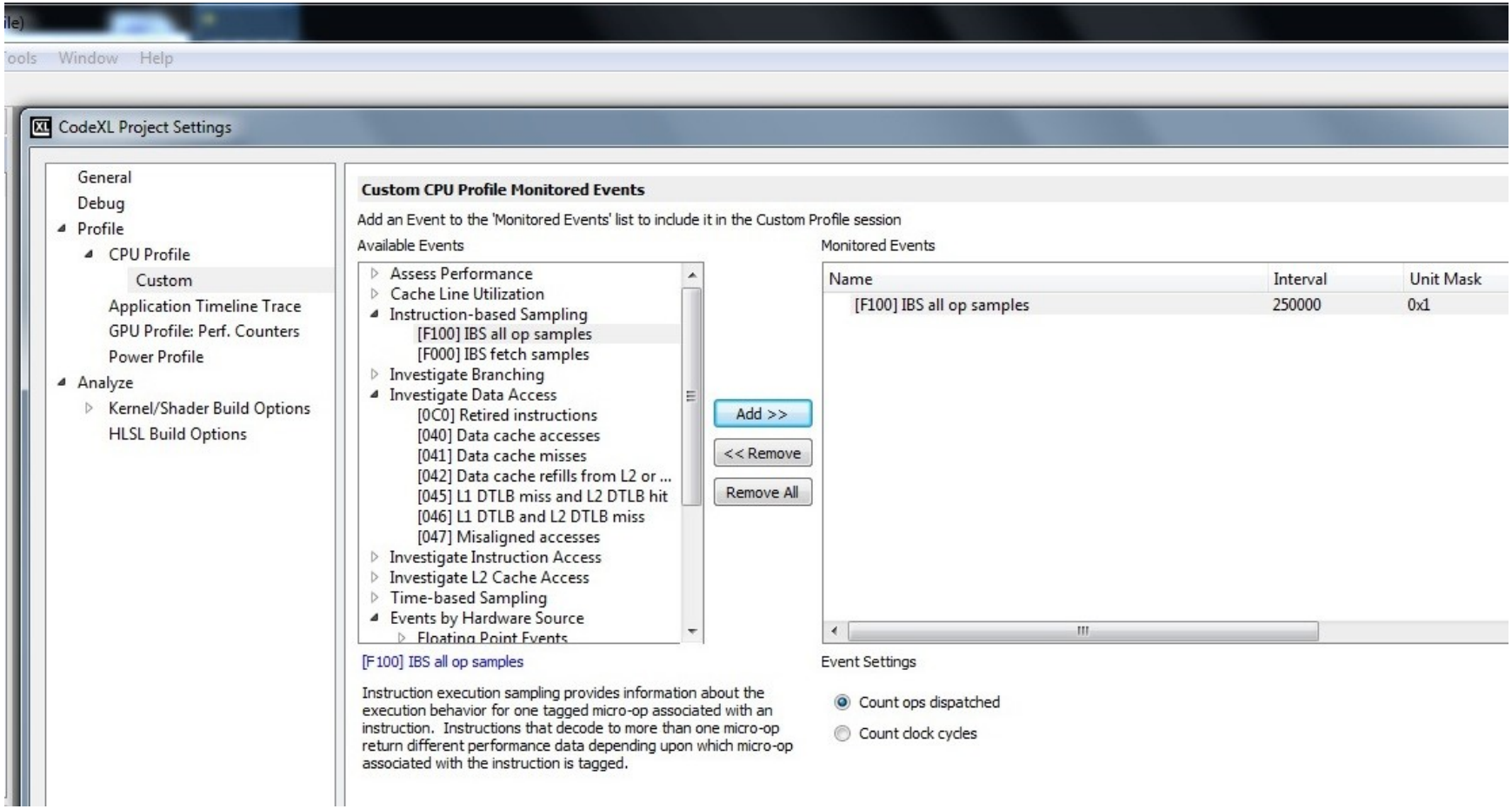
Próbkowanie operacji procesora (operation based - instruction-base samp.)

Parametrem sesji oceny wydajności jest okres próbkowania.

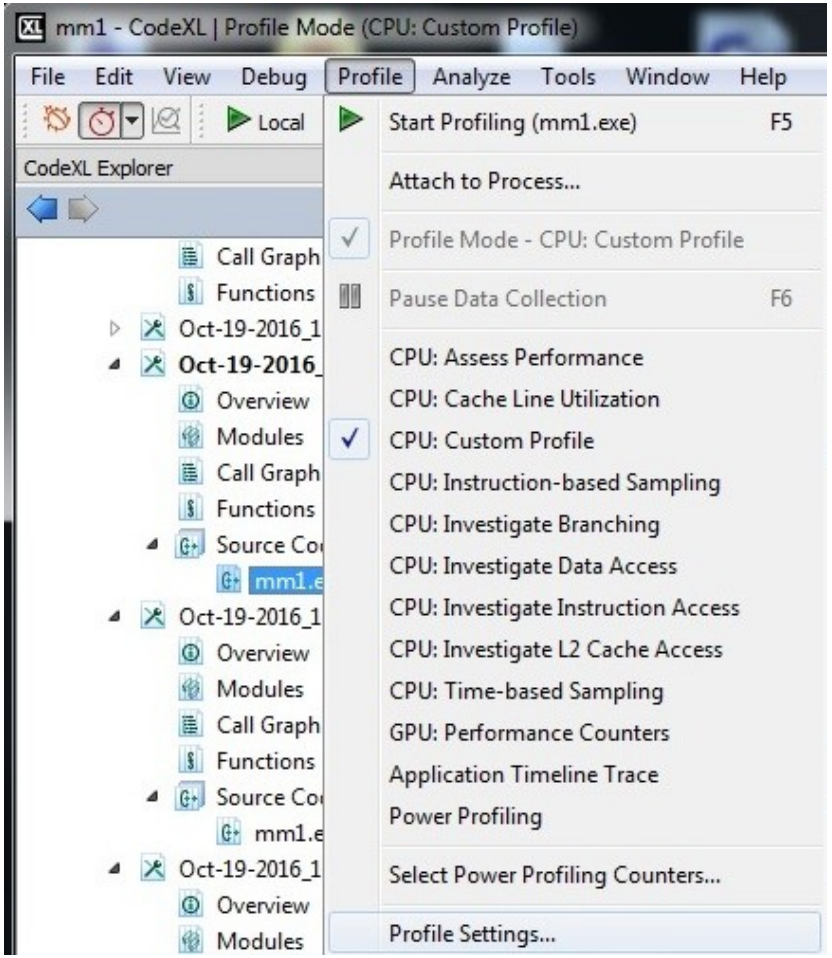
Osiągnięcie określonego progu zliczania cykli procesora powoduje monitorowanie aktualnej makrooperacji do momentu jej zrealizowania lub usunięcia. Dla zakończonych makrooperacji zapisywane, a następnie raportowane informacje dotyczą:

- identyfikatora procesu,
- adresu wirtualnego instrukcji do której należy makrooperacja,
- czas realizacji makrooperacji do zakończenia
- czas realizacji makrooperacji do zatwierdzenia
- właściwej predykcji / realizacji rozgałęzienia,
- typu operacji – odczyt, zapis,
- trafienie do DTLB, trafienie do DC
- opóźnienie w przypadku braku trafienia do DC
- lokalnego lub zdalnego dostępu, faktycznego źródła danych w przypadku korzystania ze sterownika pamięci.

Ocena efektywności bazująca na zliczaniu zdarzeń – instruction based sampling - konfiguracja



Tryby pracy programu Code XL – ocena kodu procesora



Przygotowano na podstawie:

- Basic Performance Measurements for AMD Athlon, Opteron, Phenom Processors (P.Drongowski AMD)
- An introduction to analysis and optimization with AMD CodeAnalyst

SYSTEMY OBLICZENIOWE W LABORATORIUM

- Komputery znajdujące się w ***Laboratorium Systemów Równoległych sala 2.7.6*** posiadają po jednym procesorze AMD typu PHENOM II X4 945
- System składa się z 4 procesorów logicznych – 4 rdzeni w ramach jednego procesora. System SMP.

PROCESOR PHENOM

Zgodność 32 bitowa X86 IA

wspomaganie SSE, SSE2, SSE3, SSE4a, ABM, MMX™, 3DNow!™

Technologia AMD64

rozszerzenia AMD64 technology instruction-set

Adresowanie 48-bitowe

16 rejestrów 64-bit dla integer

16 rejestrów 128-bit SSE/SSE2/SSE3/SSE4a

Architektura wielordzeniowa

opcje: Triple-core, **quad-core** lub six-core

AMD Balanced Smart Cache

oddzielne pp L1 i L2 dla każdego rdzenia

współdzielona L3

Struktura procesora

superskalarny 3 drożny (dekodowanie, wykonanie integer i FP, generacja adresu)

Struktura pp

64-Kbyte 2 drożna dzielona asocjacyjna pp danych L1

dwa dostępy 64-bit na cykl, 3 cyklowe opóźnienie

64-Kbyte 2 drożna dzielona asocjacyjna pp kodu L1

32 bajtowe pobrania

512-Kbyte 16 drożna dzielona asocjacyjna pp L2

Zarządzanie pamięcią na zasadzie wyłączności przechowywania danych L1 i L2

6-Mbyte Maximum, maksymalnie 64 drożna dzielona asocjacyjna pp L3 współdzielona

Technologia 45 nm

Złącze HyperTransport™

Procesor zintegrowany ze sterownikiem pamięci

PROCESOR PHENOM – PP KODU L1

- Układ dynamicznego wykonania instrukcji posiada 64KB pp kodu L1
- Dane w przypadku braku trafienia są pobierane do pp kodu L1 z L2, z L3 lub z pamięci systemowej w ilości 64 bajtów (pobranie) oraz kolejne 64 bajty (wstępne pobranie), po pobraniu realizowane jest wstępne dekodowanie instrukcji dla określenia granic między instrukcjami (zmiennej długości), usuwanie linii z pp jest realizowane zgodnie z algorytmem LRU (ang. least recently used)

PROCESOR PHENOM – PP DANYCH L1

- 64 kB dwu-sekcyjna, dwa porty 128 bitowe
- Strategia zapisu: Write-allocate cache – zapis realizowany do pp (przeciwna strategia do nowrite alllocation)
- Writeback cache – zapis poza pp realizowany w przypadku braku miejsca lub na skutek zlecenia zapisu stanu w pamięci głównej
- Algorytm LRU dla usuwania danych i protokół zapewnienia spójności MOESI

PROCESOR PHENOM – PP L2 I L3

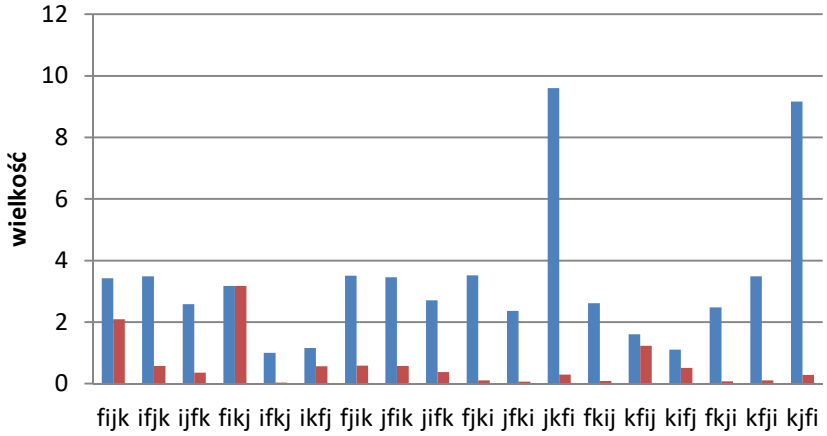
- PP L2 - victim i copy-back cache – zapisuje dane usunięte z pp L1, dane w pp są w trybie wyłącznym w L1 lub w L2
- PP L3 – victim i copy-back cache dla pp L2, głównie non-inclusive cache w przypadku, gdy dane żądane są przez jeden z rdzeni i jest mało prawdopodobne, że będą potrzebne innym, lecz możliwe powielenie.

Uwagi do wyników obliczeń

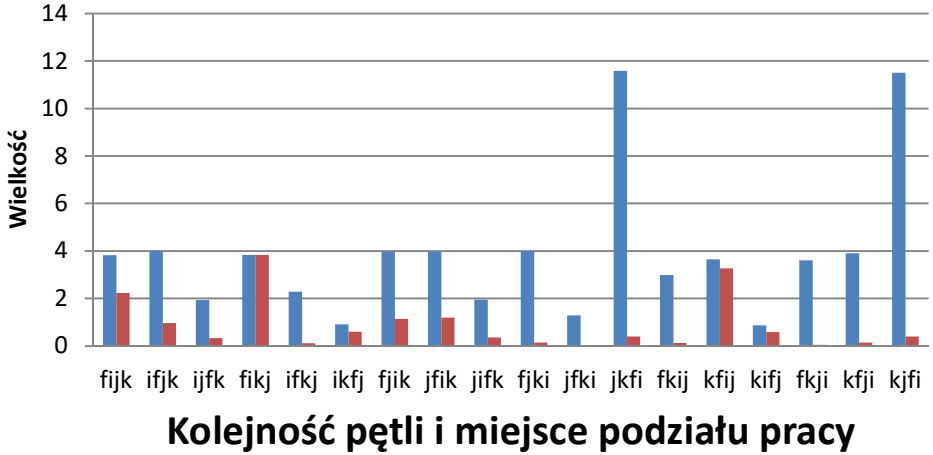
- Przyspieszenie to iloraz czasu obliczeń sekwencyjnych najlepszą z wykorzystywanych metod i czasu obliczeń równoległych badaną metodą.
- Skrócenie czasu obliczeń to iloraz czasu obliczeń sekwencyjnych i równoległych tą samą metodą.
- Obliczenia równoległe wykorzystują wszystkie procesory systemu, liczba wątków jest równa liczbie procesorów, zastosowano optymalizację kodu (Windows wersja Release, Linux O3)
- Tworzenie wątków odbywa się przed pętlami. Miejsce podziału pracy określa położenie dyrektywy #pragma omp for. W przypadku kodu z niebezpieczeństwem wyścigu umieszczono dyrektywę #pragma omp atomic lub zastosowano zmienną lokalną, gdy wątki równocześnie wyznaczały jeden wynik tablicy wyjściowej.

Przyspieszenie i skrócenie czasu obliczeń równoległych dla mnożenia macierzy

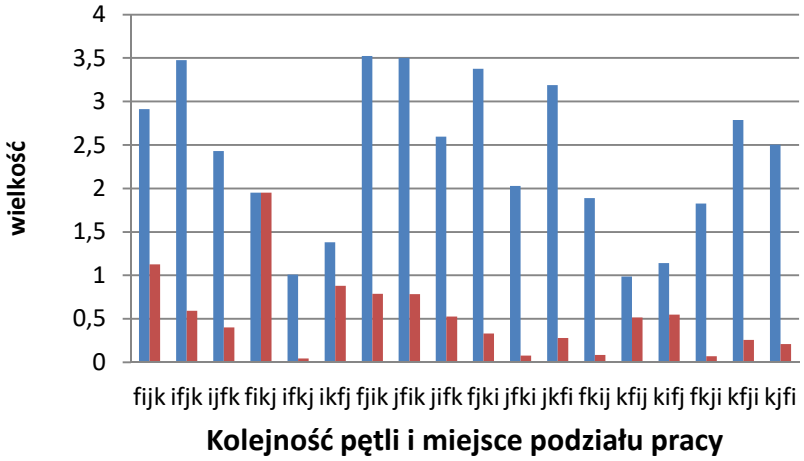
1000x1000 Intel Core i5 8MB



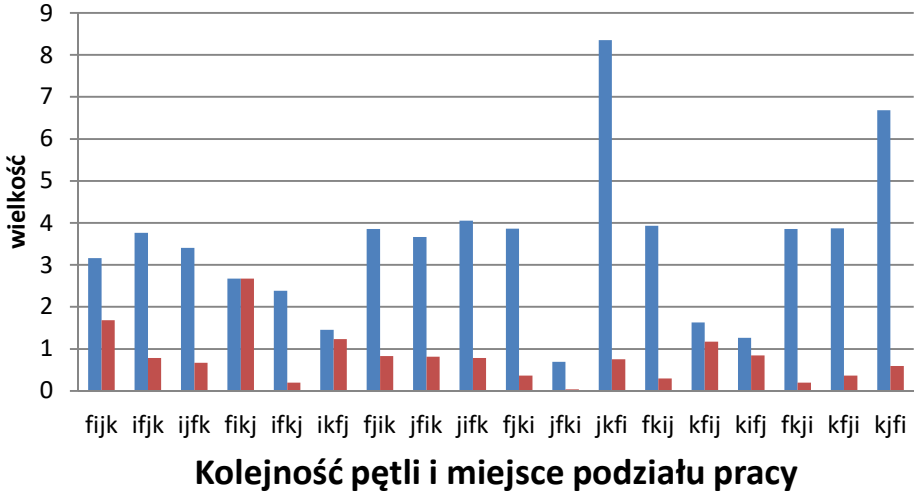
1000x1000 AMD PHENOM II X4 945 6MB



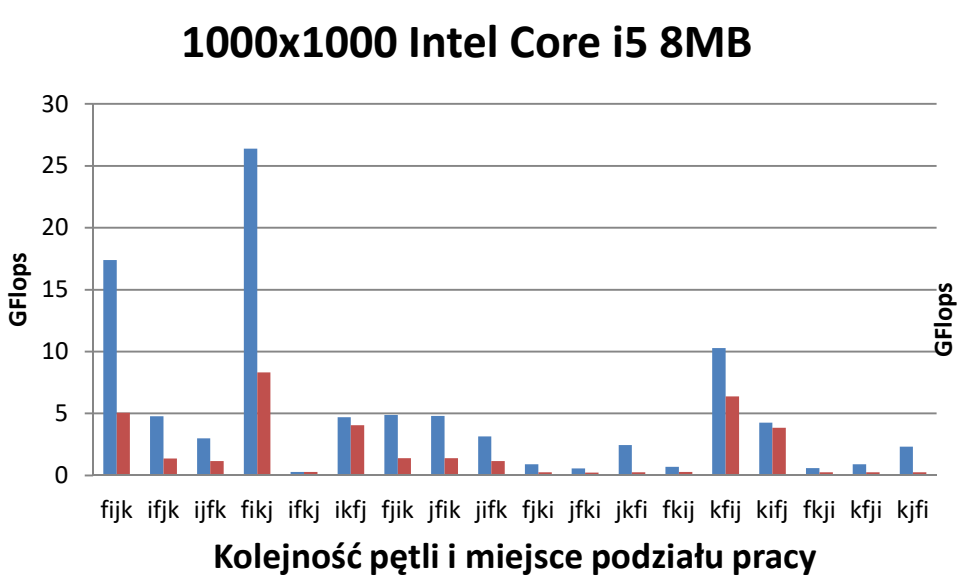
2000x2000 Intel Core i5 8MB

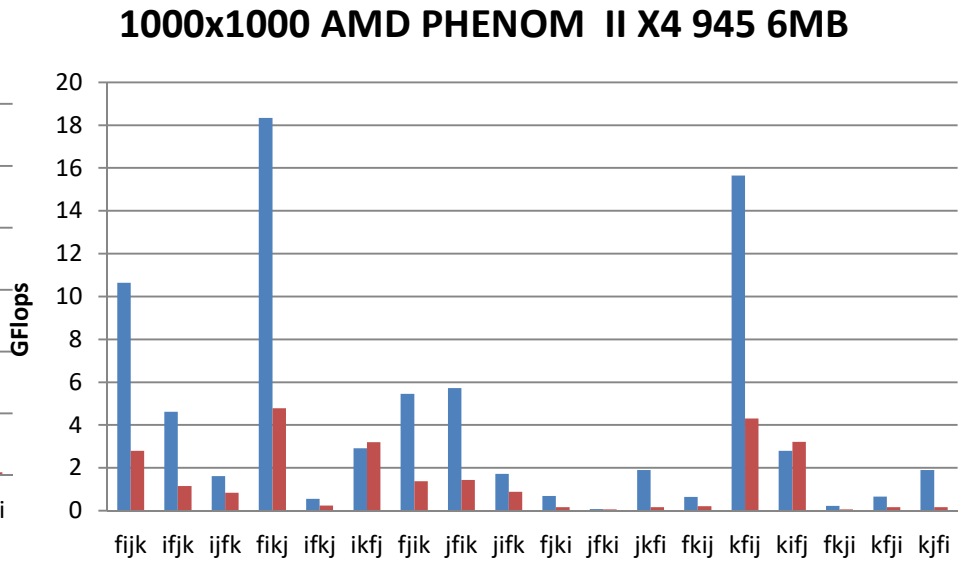


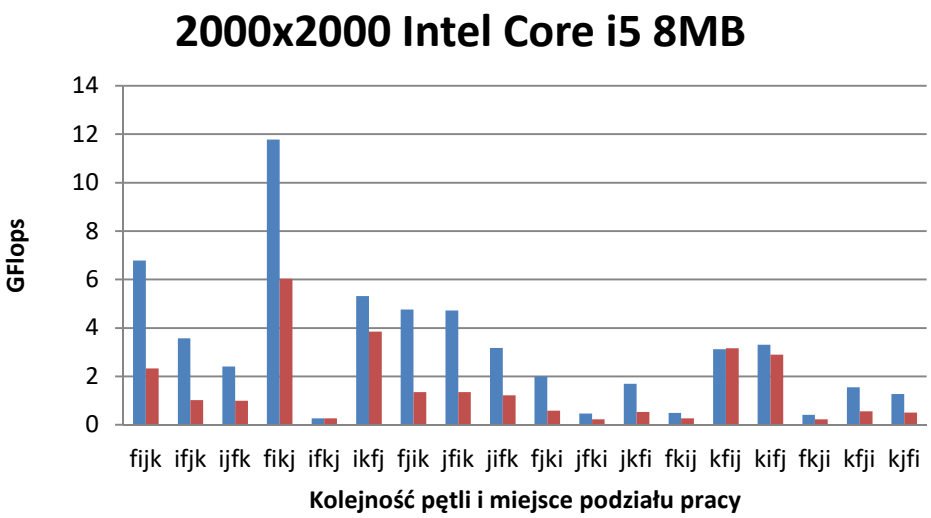
2000x2000 AMD PHENOM II X4 945 6MB

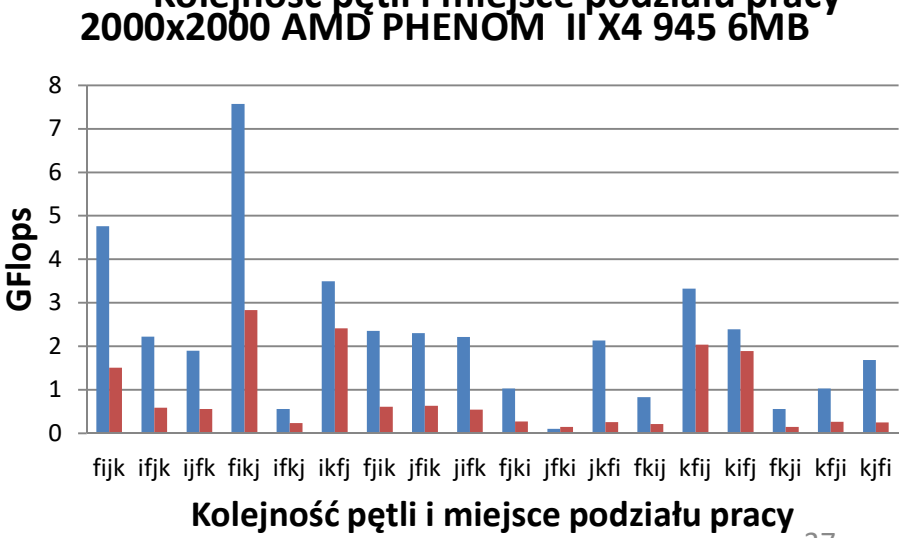


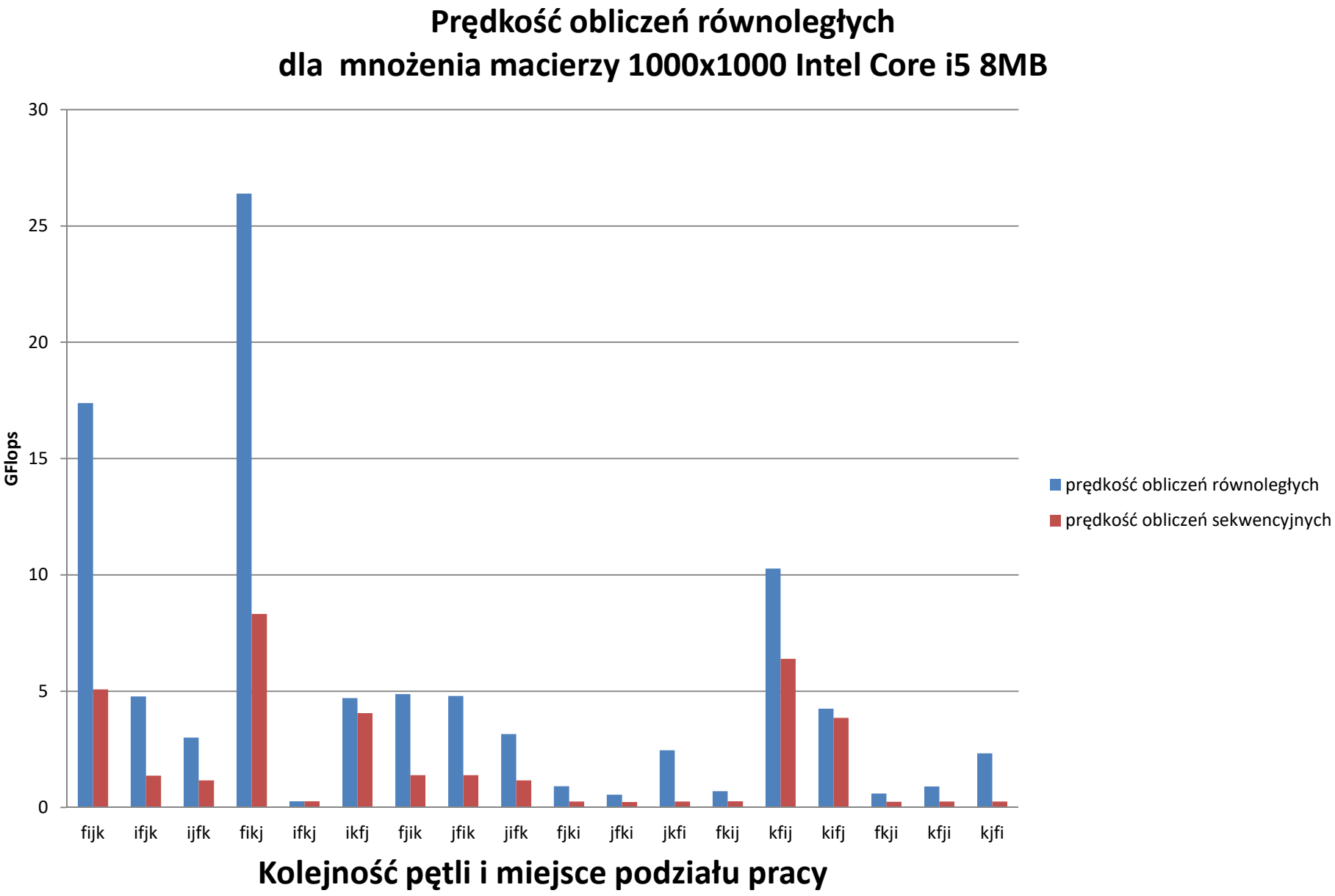
Prędkość obliczeń równoległych i sekwencyjnych dla mnożenia macierzy

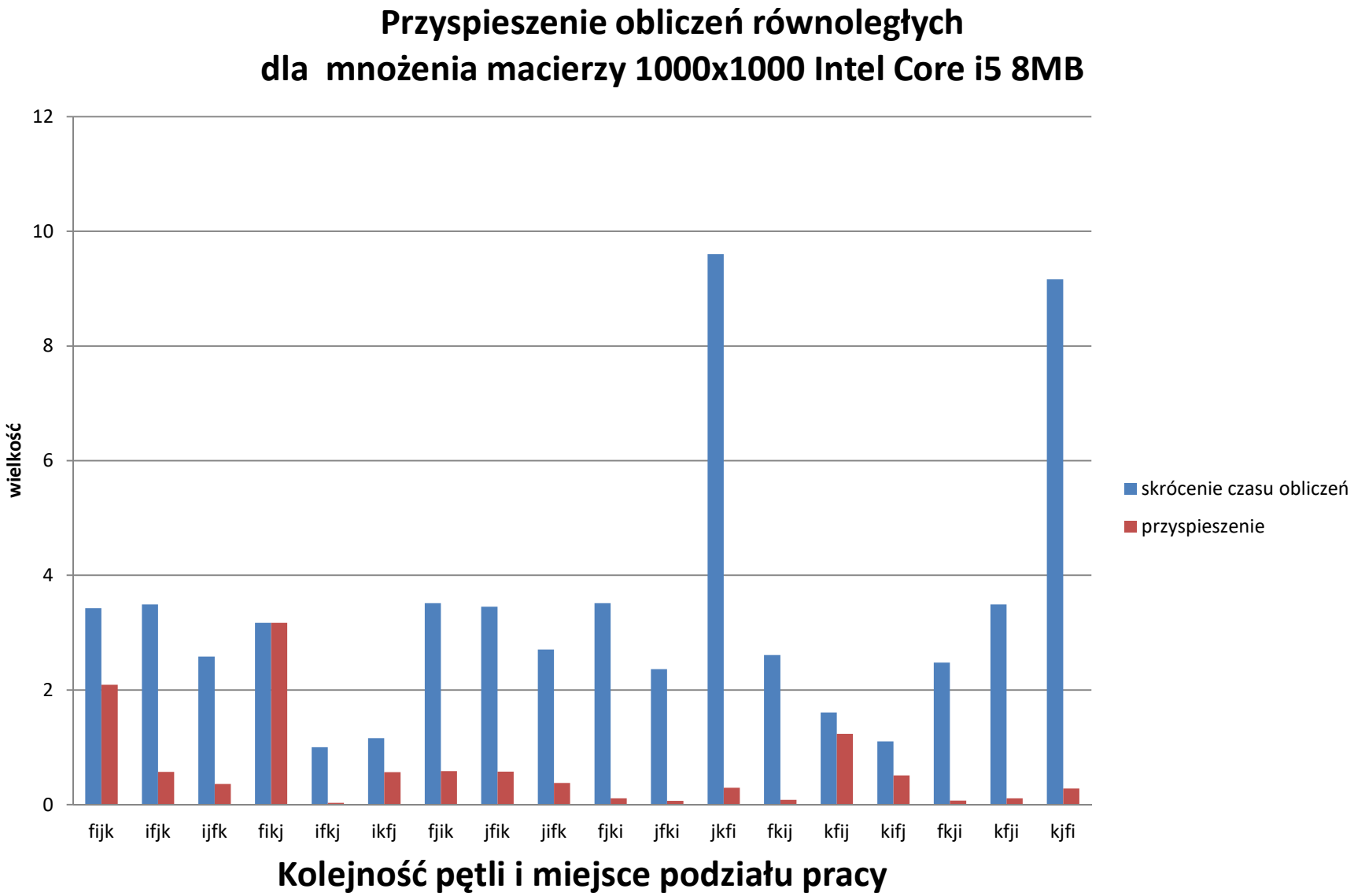


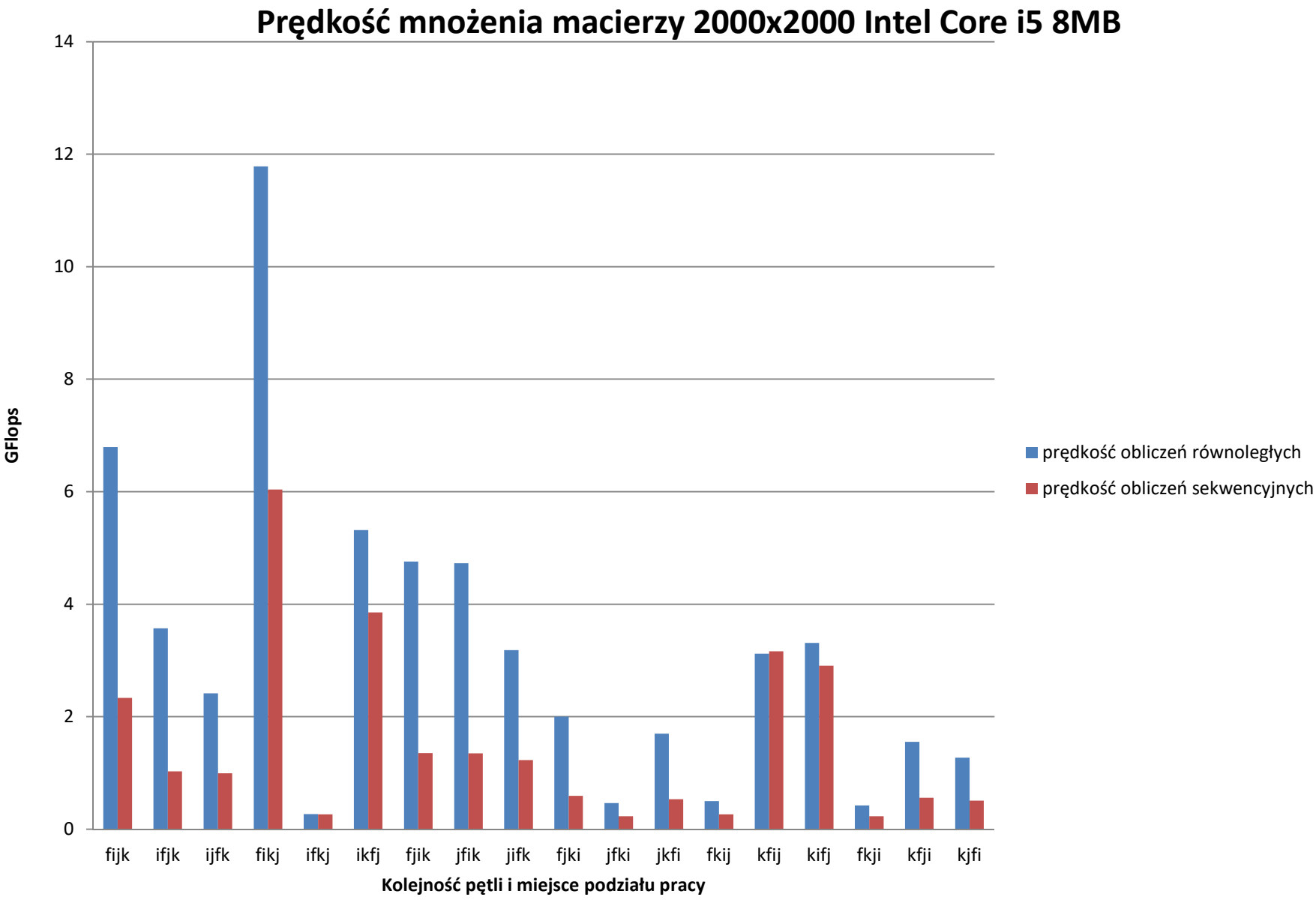


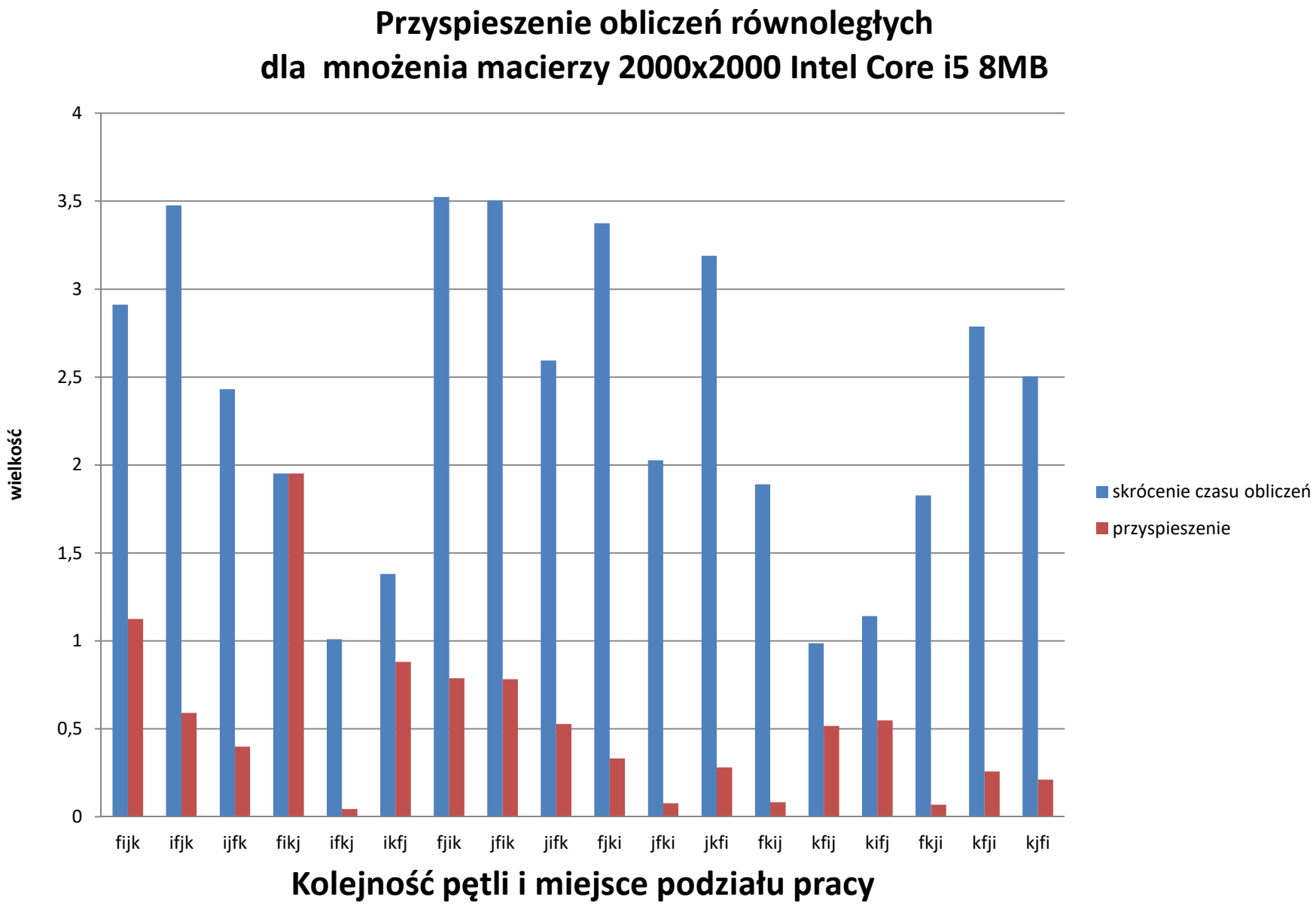




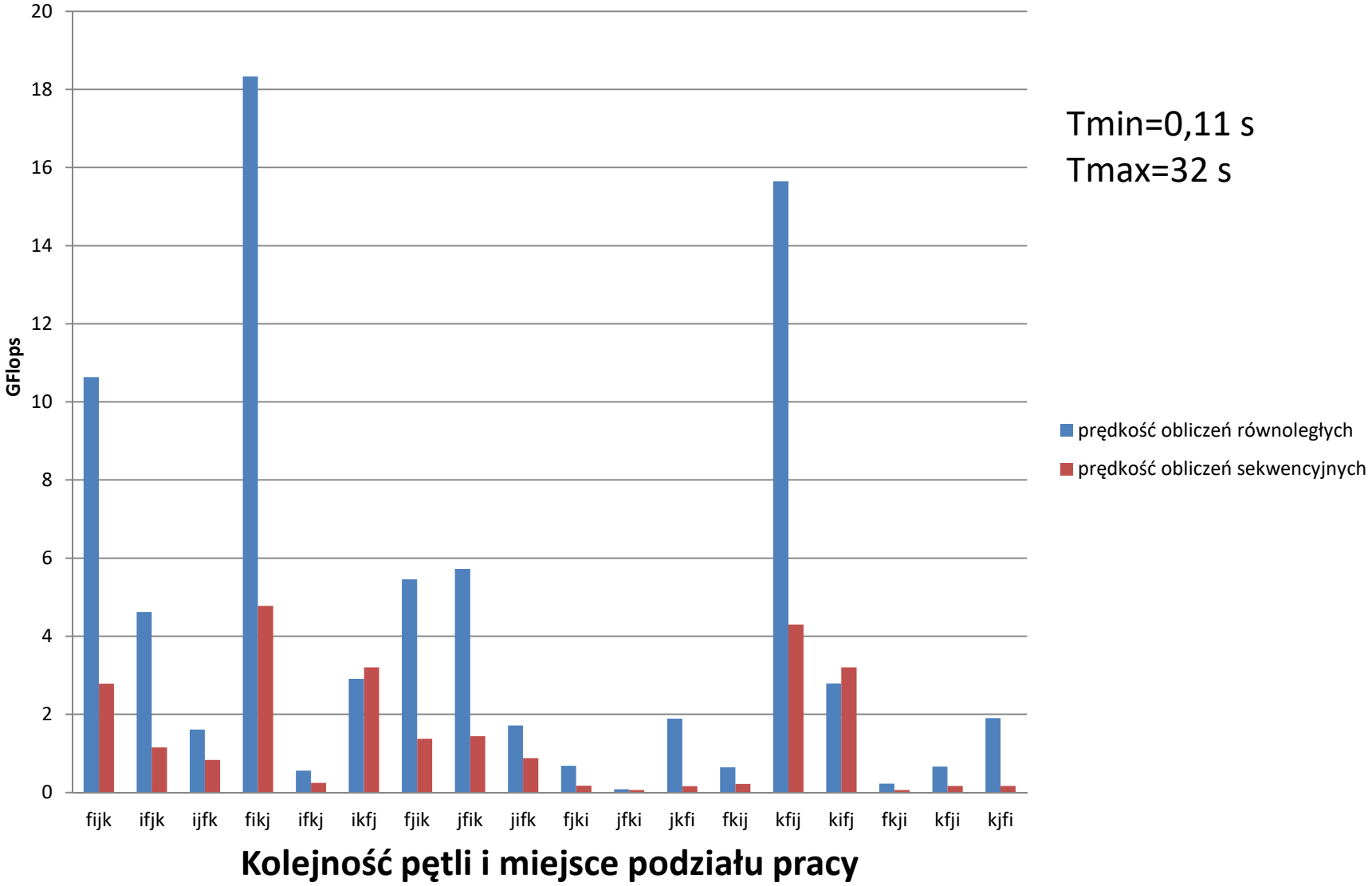

Prędkość obliczeń równoległych
dla mnożenia macierzy 1000x1000 Intel Core i5 8MB
GFlops
30
25
20
15
10
5
0
fijk
ifjk
ijfk
fikj
ifkj
ikfj
fjik
jfik
jifk
fjki
jfki
jkfi
fkij
kfij
kifj
fkji
kfji
kjfi
Kolejność pętli i miejsce podziału pracy
prędkość obliczeń równoległych
prędkość obliczeń sekwencyjnych


Przyspieszenie obliczeń równoległych
dla mnożenia macierzy 1000x1000 Intel Core i5 8MB
wielkość
12
10
8
6
4
2
0
fijk
ifjk
ijfk
fikj
ifkj
ikfj
fjik
jfik
jifk
fjki
jfki
jkfi
fkij
kfij
kifj
fkji
kfji
kjfi
skrócenie czasu obliczeń
przyspieszenie
Kolejność pętli i miejsce podziału pracy

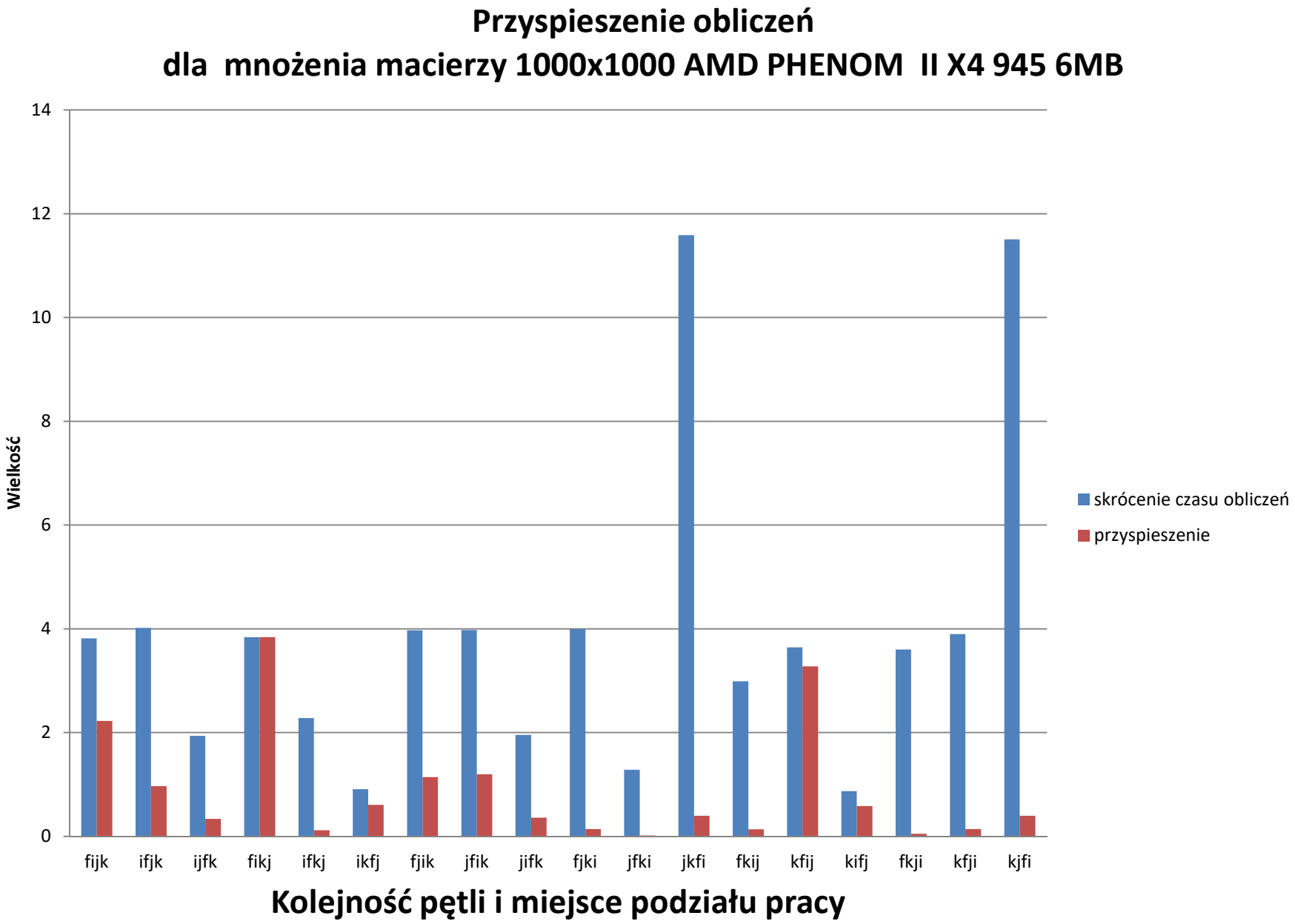

Prędkość mnożenia macierzy 2000x2000 Intel Core i5 8MB
GFlops
14
12
10
8
6
4
2
0
fijk
ifjk
ijfk
fikj
ifkj
ikfj
fjik
jfik
jifk
fjki
jfki
jkfi
fkij
kfij
kifj
fkji
kfji
kjfi
Kolejność pętli i miejsce podziału pracy
prędkość obliczeń równoległych
prędkość obliczeń sekwencyjnych

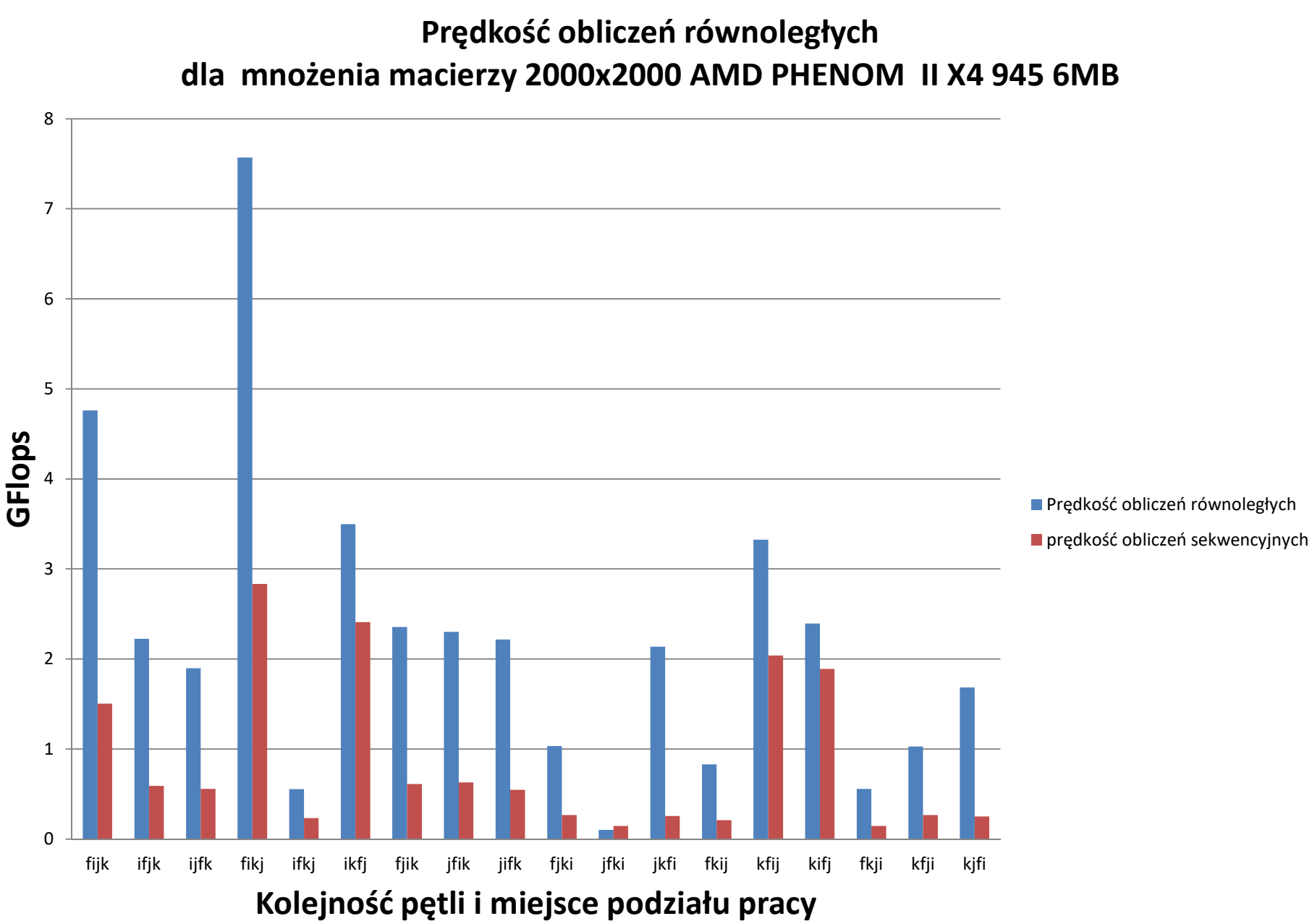

Przyspieszenie obliczeń równoległych
dla mnożenia macierzy 2000x2000 Intel Core i5 8MB
wielkość
4
3,5
3
2,5
2
1,5
1
0,5
0
fijk
ifjk
ijfk
fikj
ifkj
ikfj
fjik
jfik
jifk
fjki
jfki
jkfi
fkij
kfij
kifj
fkji
kfji
kjfi
skrócenie czasu obliczeń
przyspieszenie
Kolejność pętli i miejsce podziału pracy

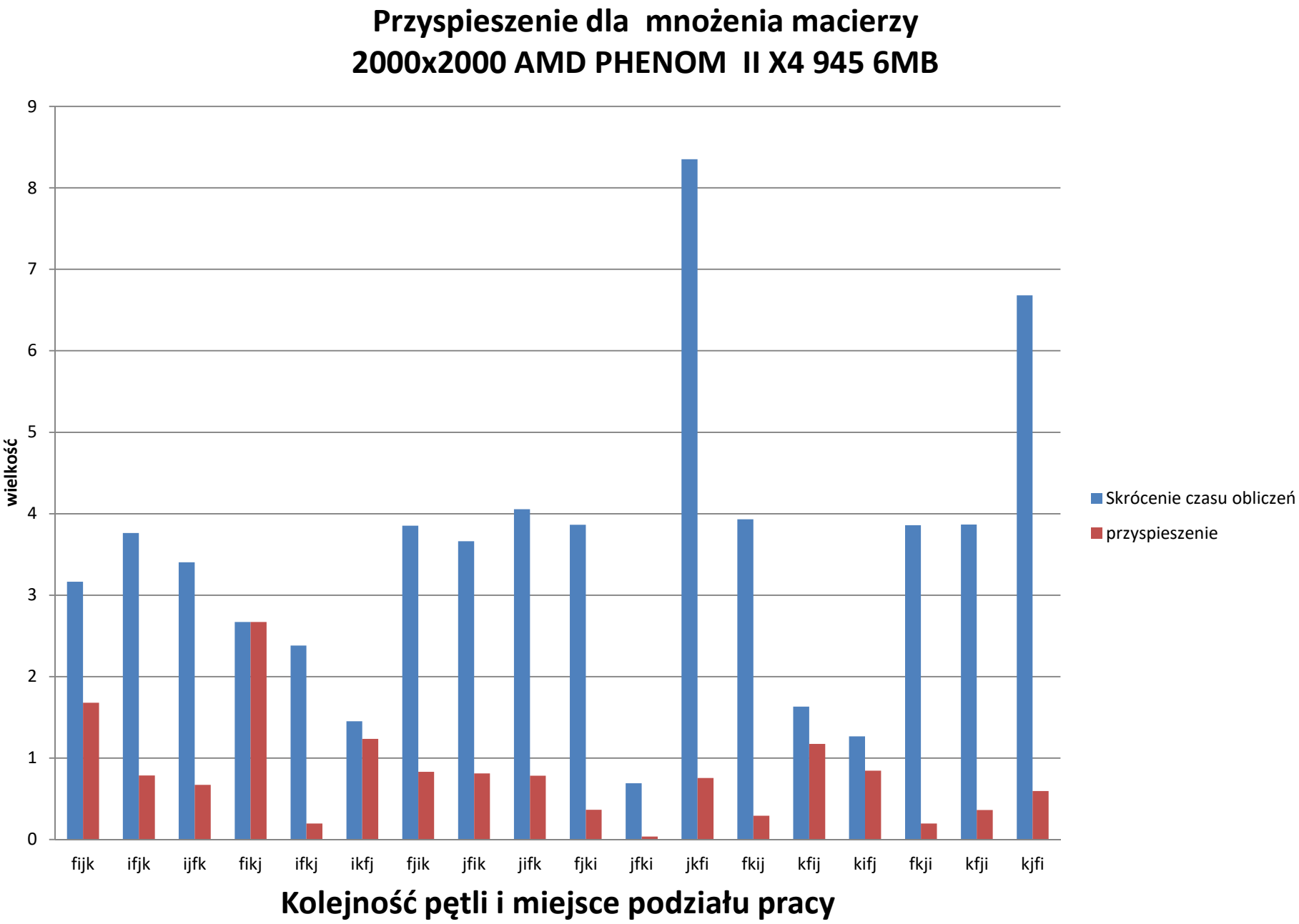
Prędkość obliczeń równoległych dla mnożenia macierzy 1000x1000 AMD PHENOM II X4 945 6MB



Tmin=0,11 s
Tmax=32 s


Przyspieszenie obliczeń
dla mnożenia macierzy 1000x1000 AMD PHENOM II X4 945 6MB
Wielkość
14
12
10
8
6
4
2
0
fijk
ifjk
ijfk
fikj
ifkj
ikfj
fjik
jfik
jifk
fjki
jfki
jkfi
fkij
kfij
kifj
fkji
kfji
kjfi
skrócenie czasu obliczeń
przyspieszenie
Kolejność pętli i miejsce podziału pracy


Prędkość obliczeń równoległych
dla mnożenia macierzy 2000x2000 AMD PHENOM II X4 945 6MB
GFlops
0
1
2
3
4
5
6
7
8
fijk
ifjk
ijfk
fikj
ifkj
ikfj
fjik
jfik
jifk
fjki
jfki
jkfi
fkij
kfij
kifj
fkji
kfji
kjfi
Kolejność pętli i miejsce podziału pracy
Prędkość obliczeń równoległych
prędkość obliczeń sekwencyjnych


Przyspieszenie dla mnożenia macierzy
2000x2000 AMD PHENOM II X4 945 6MB
wielkość
0
1
2
3
4
5
6
7
8
9
fijk
ifjk
ijfk
fikj
ifkj
ikfj
fjik
jfik
jifk
fjki
jfki
jkfi
fkij
kfij
kifj
fkji
kfji
kjfi
Skrócenie czasu obliczeń
przyspieszenie
Kolejność pętli i miejsce podziału pracy

Prędkość obliczeń równoległych dla mnożenia macierzy 1000x1000 Intel Core i5 8MB

fijk	Czas obliczen:	0,115	sec predkosc	17,3877	Gflops
ifjk	Czas obliczen:	0,4192	sec predkosc	4,7711	Gflops
ijfk	Czas obliczen:	0,6663	sec predkosc	3,0015	Gflops
fikj	Czas obliczen:	0,0758	sec predkosc	26,396	Gflops
ifkj	Czas obliczen:	7,4889	sec predkosc	0,2671	Gflops
ikfj	Czas obliczen:	0,4254	sec predkosc	4,7015	Gflops
fjik	Czas obliczen:	0,4106	sec predkosc	4,8712	Gflops
jfik	Czas obliczen:	0,4172	sec predkosc	4,794	Gflops
jifk	Czas obliczen:	0,633	sec predkosc	3,1594	Gflops
fjki	Czas obliczen:	2,1942	sec predkosc	0,9115	Gflops
jfki	Czas obliczen:	3,6198	sec predkosc	0,5525	Gflops
jkfi	Czas obliczen:	0,8142	sec predkosc	2,4565	Gflops
fkij	Czas obliczen:	2,8723	sec predkosc	0,6963	Gflops
kfij	Czas obliczen:	0,1947	sec predkosc	10,2727	Gflops
kifj	Czas obliczen:	0,4705	sec predkosc	4,2507	Gflops
fkji	Czas obliczen:	3,3385	sec predkosc	0,5991	Gflops
kfji	Czas obliczen:	2,2168	sec predkosc	0,9022	Gflops
kjfi	Czas obliczen:	0,8582	sec predkosc	2,3305	Gflops