

## Przetwarzanie równoległe PROJEKT OMP

Temat projektu dotyczy analizy efektywności przetwarzania równoległego realizowanego w komputerze równoległym z procesorem wielordzeniowym z pamięcią współdzieloną.

Przykładem jest komputer z procesorem 4 rdzeniowym dostępny w laboratorium 2.7.6. W skład projektu wchodzi:

- 1) Przygotowanie kilku wersji kodu zgodnie z wymaganiami wersji zadania,
- 2) Analiza poprawności kodu równoległego przygotowanych wersji (sprawdzenie zmiennych współdzielonych pod kątem możliwości wystąpienia wyścigu), usunięcie stwierdzonych błędów poprzez zapewnienie niepodzielnego uaktualnienia współdzielonych zmiennych
- 3) Analiza efektywności kodu –
  - a) analiza warunków lokalnego czasowo dostępu do danych w pamięci podręcznej (uwzględnienie wielkości pamięci podręcznej, wyodrębnienie etapów przetwarzania w oparciu o różne bloki danych dla przetwarzania sekwencyjnego i równoległego (oddzielnie jeśli różne), parametrem analizy jest wielkość pamięci podręcznej procesora L3 i tablicy wynikowej.
  - b) analiza warunków lokalnego przestrzennie dostępu do danych, w kontekście wielkości bufora translacji adresów;
    - określenie zależności między wielkością macierzy, a wielkością bufora translacji zapewniającego jednokrotne uzupełnianie odwzorowania strony na ramkę pamięci (zakładamy wielkość pamięci podręcznej adresów stron wirtualnych danych – DTLB równą ok. 550 par wpisów par adresów stron wirtualnej i ramki pamięci o rozmiarze 4 kB),
    - znaczenie braku trafienia przy braku lokalności przestrzennej – jak często brak trafienia występuje (na ile instrukcji kodu)
  - c) analiza podziału pracy na wątki, proszę określić zadania realizowane przez poszczególne wątki i obszary danych wejściowych i wyjściowych przetwarzanych przez jeden wątek ,
  - d) ocena kodu pod kątem unieważniania danych w pamięciach podręcznych przez zapisy innych wątków - false-sharing, ocena znaczenia występującego false-sharing na efektywność przetwarzania.
- 4) Określenie **charakterystycznych wielkości instancji** problemu na podstawie analizy efektywności kodu. Charakterystyczna wielkość instancji określa taką instancję która charakteryzuje się potencjalnie wyższą lub niższą prędkością przetwarzania za względu na lokalność czasową lub lokalność przestrzenną dostępu do danych dla poszczególnych wersji kodu, współbieżnych lub sekwencyjnych (liczba i wielkość instancji jest zależna od wariantu zadania i **uzasadniona analizą kodu** – użytych w przetwarzaniu struktur danych przez poszczególne wątki oraz sposobem i kolejnością dostępu do danych - średnio około 6 różnych instancji uwzględniając wielkości macierzy N i parametru R metody 3 pętlowej). Dla każdej instancji spodziewamy się określonego analizą kodu poziomu efektywności przetwarzania (zapewnienie zrównoważenia procesorów obliczeniami, rodzaj lokalności (lub braku lokalności) dostępu do danych.
- 5) Określenie zakresu zbieranych miar efektywności, które **zostaną użyte** do oceny prędkości, efektywności oraz wyjaśnienia znaczenia miar i przyczyn wyboru miar.

- 6) Proszę opisać własnymi słowami sposób zbierania miar i ich znaczenie (progi zliczania zdarzeń) - (program Code XL określić przyczynę i wykorzystywane tryby pracy: time based/event based / instruction based). Podczas eksperymentu proszę sprawdzić czy prezentowane wartości są spójne z warunkami eksperymentu (logicznie poprawne – możliwe - np. liczba zdarzeń ujemna, podzbiór większy niż domena), czy wielkości mierzone zmieniają się zgodnie z oczekiwaniami przy zmianie wielkości instancji i kodu. Proponowane miary jakości przetwarzania (mierzone i wyliczane) dla obszaru kodu w którym realizowane są badane obliczenia (bez zakresu kodu przygotowania danych):
- a) czas obliczeń (od momentu rozpoczęcia pracy nad mnożeniem macierzy pierwszego z wykonujących obliczenia procesorów do zakończenia pracy przez ostatni procesor)  $T_{obl}$ ,
  - b) prędkość obliczeń wg wzoru:  $Z/T_{obl}$  w MFLOPS – milionach operacji zmiennoprzecinkowych na sekundę gdzie  $Z$  - złożoność w liczbie operacji zmiennoprzecinkowych =  $2 \cdot (n \exp 3)$ ;
  - c) liczba wykonanych instrukcji kodu asemblera
  - d) liczba cykli (zajętego obliczeniami badanego kodu) procesora (test zrównoważenia procesorów) -LCP
  - e) IPC dla jednego procesora i IPC szczytowe dla zbioru równolegle pracujących nad liczbą obliczeniami procesorów
  - f) liczba braków trafień do pp L3 = liczbie pobrań linii pamięci z pamięci operacyjnej – LBTL3
  - g) krotność pobierania danych do pamięci podręcznej =  $LBTL3 \cdot dlpp / (wtz \cdot 3 \cdot n \cdot n)$  gdzie wtz jest wielkością użytego typu zmiennej, dlpp jest długością linii pp
  - h) **wskaźniki braku** trafień  $<0,1>$  przy dostępie do:
    - pamięci podręcznej poziomu 3 L3 procesora – zdarzenie L3 miss świadczy o konieczności pobrania linii pp z pamięci operacyjnej
    - jakiegokolwiek poziomu pierwszego lub drugiego bufora translacji danych - DTLB L1 miss i DTLB L2 miss – zdarzenie świadczy o konieczności uruchomienia kosztownej czasowo procedury odtworzenia adresu ramki pamięci operacyjnej odpowiadającego podanemu adresowi obiektu żadanego przez procesor realizujący kod użytkownika (ang. page walk)
  - i) miara kosztu synchronizacji: wskaźnik czasu braku wykorzystania procesorów podczas obliczeń wg wzoru:  $(LUPF \cdot T_{obl} - LCP \cdot T_{clk}) / (LUPF \cdot T_{obl})$  gdzie LUPF to liczba użytych procesorów fizycznych (=często równa liczbie wątków OMP),  $T_{clk}$  – czas trwania cyklu taktowania procesora [s]
  - j) przyspieszenie i efektywność przetwarzania równoległego (wg standardowych wzorów), jako punkt odniesienia we wzorach należy przyjąć uzyskany dla jednakowej instancji problemu:
    - czas **przetwarzania sekwencyjnego dla metody 3 pętlowej**, w kolejności zagnieżdżenia pętli  $ikj$ , gdy ciało pętli wewnętrzne zawiera kod:  $c[i,j] += a[i,k] \cdot b[k,j]$ ; lub
    - inny (jeśli jest krótszy) czas przetwarzania sekwencyjnego uzyskany w eksperymencie
- 7) Eksperyment proszę wykonać dla kodu kompilowanego w trybie release – optymalizacja generacji kodu przez procesor.

- 8) Do obliczeń proszę wykorzystać tablice zainicjowane liczbami losowymi (np. `matrix_a[i][j] = (float) rand() / RAND_MAX ;`)
- 9) Po wykonaniu wstępnych testów logiczności i poprawności mierzonych miar efektywności wykonanie eksperymentu zgodnie z przygotowanym planem.
- 10) Przygotowanie sprawozdania.

**Sprawozdanie zawierać powinno:**

- Prezentacja wykorzystanego systemu obliczeniowego jeśli jest inny od komputerów laboratoryjnych.
- Kluczowe fragmenty kodu sekwencyjnego i równoległego, krótko omówić dyrektywy Open MP i ich znaczenie dla przebiegu obliczeń, opis różnic w wersjach kodu wersji i ich znaczenia dla przebiegu obliczeń. Fragmenty kodu ponumerować na potrzeby odwołań w treści sprawozdania.
- Prezentacja wyników **analiz z przygotowania eksperymentu**: poprawność, efektywność – synchronizacja, false sharing, lokalność dostępu dodanych, przydział pracy do wątków – opis słowny i rysunki, **uzasadnienie wielkości instancji**, zależności określające warunki lokalności czasowej i przestrzennej przetwarzania
- Obliczenie, porównanie i prezentacja w sposób **czytelny** (zwięzły - tabela) wszystkich znaczących miar prędkości i efektywności przetwarzania.
- Porównać prędkości, za pomocą wartości miar jakości przetwarzania ocenić hipotezy z teoretycznych analiz efektywności obliczeń dla wersji kodów i rozmiarów instancji. Proszę dokonać ostatecznej, możliwie szerokiej i biorącej pod uwagę wszystkie znane okoliczności, podbudowanej eksperymentem **próby uzasadnienia** przyczyn niskiej/wysokiej:
  - prędkości przetwarzania,
  - przyspieszenia oraz efektywności przetwarzania równoległego
- dla poszczególnych wersji kodu i rozmiarów instancji, za pomocą zmierzonych miar efektywności. W próbie uzasadnienia należy się odwoływać czytelnie do ocenianych i komentowanych pozycji w tabelach, przebiegach w wykresach z określeniem nr tabeli/ wykresu, wiersza i kolumny w tabeli oraz przebiegu wykresu.

**Zasady prezentacji miar jakości przetwarzania:**

- wielkości miar proszę przedstawić zaokrąglone do 2 cyfr znaczących np.  $2.3 \cdot 10^5$  co oznacza 230000, niedopuszczalne są wartości miar typu: 234800000654433466,340530022
- proszę wyjaśnić w języku polskim znaczenie miary efektywności i ich jednostki (np. sekunda, MFLOPS, liczba trafień dla pamięci podręcznych L3, liczba trafień do L3 przypadająca na instrukcję kodu asemblera)
- prezentować miary efektywności za pomocą tabel (ewentualnie wykresu dla lepszej czytelności) – tabele i wykresy należy **podpisać** za pomocą **numeru** tabeli/wykresu, oraz **opisu**, opisać jednoznacznie zawartość pod (tabeli lub wykresu). W sposób jednoznaczny opisać znaczenie prezentowanych wartości (podpis powinien zawierać takie informacje jak: wersja kodu, wielkość instancji, liczba procesorów, liczba wątków, tryb pracy programu oceny efektywności), **nazwy** określającej zawartość proszę nie umieszczać tytułu nad obiektem

- o określenie zakresu prędkości dla rozmiarów instancji i wariantów badanego kodu, określenie najszybszych wariantów kodu i wielkości instancji, dla których prędkość jest maksymalna

Wymagana jest wersja drukowana (dwustronnie) sprawozdania oddana podczas zajęć (sprawdzanie poprawności formalnej - sprawozdania niezgodne z powyższymi wymogami będą oddawane do poprawy) i elektroniczna przesłana na adres e-mail prowadzącego zajęcia. Po sprawdzeniu merytorycznej zawartości zostanie wystawiona ocena, możliwe jest skierowanie sprawozdania do uzupełnienia i/lub rozmowa z autorami na temat przebiegu eksperymentu i sprawozdania.

Termin oddania projektu – 4-5 zajęcia. Nieuzasadnione opóźnienie w oddaniu projektu obniża automatycznie ocenę o 1 stopień na tydzień.

### Tematy projektu:

1. Mnożenie macierzy porównanie efektywności metod –
  - 3 pętle - kolejność pętli:  $ijk$ ,
  - 6 pętli - kolejność pętli: zewnętrznych  $ijk$ , wewnętrznych:  $ikj$ , podział pracy przed pętlą 1.
2. Mnożenie macierzy porównanie efektywności metod –
  - 3 pętle - kolejność pętli:  $ijk$ ,
  - 6 pętli - kolejność pętli: zewnętrznych  $ijk$ , wewnętrznych:  $ikj$ , podział pracy przed pętlą 1.
3. Mnożenie macierzy porównanie efektywności metod –
  - 3 pętle - kolejność pętli:  $jki$ ,
  - 6 pętli - kolejność pętli: zewnętrznych  $ijk$ , wewnętrznych:  $ikj$ , podział pracy przed pętlą 1.
4. Mnożenie macierzy porównanie efektywności metod –
  - 3 pętle - kolejność pętli:  $jik$ ,
  - 6 pętli - kolejność pętli: zewnętrznych  $ijk$ , wewnętrznych:  $ikj$ , podział pracy przed pętlą 1.
5. Mnożenie macierzy porównanie efektywności metod –
  - 3 pętle - kolejność pętli:  $kij$ ,
  - 6 pętli - kolejność pętli: zewnętrznych  $ijk$ , wewnętrznych:  $ikj$ , podział pracy przed pętlą 1.
6. Mnożenie macierzy porównanie efektywności metod –
  - 3 pętle - kolejność pętli:  $kji$ ,
  - 6 pętli - kolejność pętli: zewnętrznych  $ijk$ , wewnętrznych:  $ikj$ , podział pracy przed pętlą 1.
7. Mnożenie macierzy porównanie efektywności metod –
  - 3 pętle - kolejność pętli:  $ikj$ ,
  - 6 pętli - kolejność pętli: zewnętrznych  $ijk$ , wewnętrznych:  $ikj$ , podział pracy przed pętlą 4.
8. Mnożenie macierzy porównanie efektywności metod –
  - 3 pętle - kolejność pętli:  $ijk$ ,
  - 6 pętli - kolejność pętli: zewnętrznych  $ijk$ , wewnętrznych:  $ikj$ , podział pracy przed pętlą 4.
9. Mnożenie macierzy porównanie efektywności metod –
  - 3 pętle - kolejność pętli:  $jki$ ,
  - 6 pętli - kolejność pętli: zewnętrznych  $ijk$ , wewnętrznych:  $ikj$ , podział pracy przed pętlą 4.
10. Mnożenie macierzy porównanie efektywności metod –
  - 3 pętle - kolejność pętli:  $jik$ ,
  - 6 pętli - kolejność pętli: zewnętrznych  $ijk$ , wewnętrznych:  $ikj$ , podział pracy przed pętlą 4.
11. Mnożenie macierzy porównanie efektywności metod –
  - 3 pętle - kolejność pętli:  $kij$ ,
  - 6 pętli - kolejność pętli: zewnętrznych  $ijk$ , wewnętrznych:  $ikj$ , podział pracy przed pętlą 4.
12. Mnożenie macierzy porównanie efektywności metod –
  - 3 pętle - kolejność pętli:  $kji$ ,

- 6 pętli - kolejność pętli: zewnętrznych ijk, wewnętrznych: ikj, podział pracy przed pętlą 4
- 13. Mnożenie macierzy porównanie efektywności metod –
  - 6 pętli - kolejność pętli: zewnętrznych ijk, wewnętrznych: ikj, podział pracy przed pętlą 1.
  - 6 pętli - kolejność pętli: zewnętrznych ijk, wewnętrznych: ikj, podział pracy przed pętlą 4.
- 14. Mnożenie macierzy - wpływ lokalizacji dyrektywy #pragma omp for na poprawność i efektywność metod
  - 3 pętli - kolejność pętli: ijk, #pragma omp parallel przed pierwszą pętlą for
- 15. Mnożenie macierzy - wpływ lokalizacji dyrektywy #pragma omp for na poprawność i efektywność metod
  - 3 pętli - kolejność pętli: ikj, #pragma omp parallel przed pierwszą pętlą for
- 16. Mnożenie macierzy - wpływ lokalizacji dyrektywy #pragma omp for na poprawność i efektywność metod
  - 3 pętli - kolejność pętli: jik, #pragma omp parallel przed pierwszą pętlą for
- 17. Mnożenie macierzy - wpływ lokalizacji dyrektywy #pragma omp for na poprawność i efektywność metod
  - 3 pętli - kolejność pętli: jki, #pragma omp parallel przed pierwszą pętlą for
- 18. Mnożenie macierzy - wpływ lokalizacji dyrektywy #pragma omp for na poprawność i efektywność metod
  - 3 pętli - kolejność pętli: kij, #pragma omp parallel przed pierwszą pętlą for
- 19. Mnożenie macierzy - wpływ lokalizacji dyrektywy #pragma omp for na poprawność i efektywność metod
  - 3 pętli - kolejność pętli: kji, #pragma omp parallel przed pierwszą pętlą for

#### Literatura:

- Wykłady i literatura przedmiotu

Przygotowany: 12.03.2018

Zmiany: 24.04.2018