

Przetwarzanie równoległe PROJEKT 1 OMP studia dzienne kierunek Informatyka rok ak. 2019/ 2020  
Ogłoszony 30.03.2020 w ramach nauczania zdalnego podczas pandemii koronawirusa.

#### Przygotowanie projektu - grupy

Studenci wykonują projekt w grupach 2 osobowych (można przygotować projekt indywidualnie). Pracę nad projektem należy zorganizować jako pracę zdalną w celu ograniczenia rozprzestrzeniania się pandemii. Proszę o zgłoszenia grup dwuosobowych w terminie do 5.04.2020. Tworząc grupę proszę uwzględnić dostęp do własnego sprzętu i oprogramowania (system obliczeniowy **którą grupa projektowa dysponuje**). **Dwuosobowe grupy wykonujące ten projekt nie muszą składać się ze studentów jednej grupy dziekańskiej lub jednej grupy laboratoryjnej.**

#### Zakres projektu:

Analiza efektywności przetwarzania równoległego realizowanego w komputerze równoległym z procesorem wielordzeniowym z pamięcią współdzieloną. Przykładem takiego systemu jest komputer z procesorem 6 rdzeniowym umożliwiający jednocześnie uruchomienie 12 wątków dostępny w laboratorium 2.7.6 lub dowolny inny komputer pozwalający jednocześnie przetwarzanie co najmniej 2 wątków (ciekawsze efekty można zaobserwować przy użyciu procesorów z większą liczbą rdzeni).

#### Problem do analizy:

Problem znajdowania liczb pierwszych w podanym jako parametr przedziału  $<M..N>$ . W odpowiedzi na parametry  $<10,49>$  powinny zostać wykonane niezbędne obliczenia potrzebne do wyznaczenia liczb 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47.

#### Algorytmy do implementacji

Należy przygotować różne wersje algorytmów: sekwencyjne i równoległe.

#### Wersje podejść - koncepcyjne:

1. Testowanie każdej liczby z przedziału i sprawdzanie czy jest podzielna przez liczby mniejsze. Zakres liczb mniejszych używanych do sprawdzenia można ograniczyć do pierwszej w kolejności liczby, która jest dzielnikiem badanej liczby i liczb mniejszych od pierwiastka kwadratowego z badanej liczby np. aby sprawdzić liczbę 49 jest ona dzielona przez 2,3,5,7, a liczba 50 będzie dzielona tylko przez 2. Jako potencjalne dzielniki można wykorzystywać liczby pierwsze (jeśli są dostępne lub jeśli ich wyznaczenie i dostęp do nich nie spowodują wzrostu kosztu obliczeń (potrzebny test efektywności)).
2. Usuwanie ze zbioru liczb do zbadania liczb będących liczbami złożonymi (metoda Sita, które służy do odsiania nieinteresujących nas liczb złożonych – pozostają liczby pierwsze). Aby wyznaczyć i usunąć ze zbioru liczby złożone wyznaczane są wielokrotności liczb. Analogicznie do podejścia 1 interesują nas wielokrotności liczb będących liczbami w zakresie od 2 do pierwiastka kwadratowego z wartości górnego zakresu badanych liczb. Aby zmniejszyć ilość pracy – liczbę wykreśleń, do usunięcia liczb złożonych wystarczy kolejno używać wielokrotności, które są niezbędne: wielokrotności liczby 2 (2,4,6,8...) , wielokrotności liczby 3 począwszy od  $3*3, 3*5, 3*7, 3*9, 3*11...$  Wystarczy używać wielokrotności liczb pierwszych, jeśli są dostępne lub okaże się że ich wyznaczenie jest rozwiązaniem efektywnym. Może się zatem okazać, że aby wyznaczyć liczby pierwsze z przedziału  $<M..N>$  potrzebne są liczby pierwsze z przedziału  $<2,..., \text{pierwiastek}(N)>$  używane do usuwania wielokrotności.

#### Wersje podejść – warianty zrównoleglenia:

1. Uwzględnienie dwóch podejść do podziału pracy pomiędzy wątki: podejście podziału domenowe i funkcyjne. Różne podejścia podziału pracy (tworzenia zadań równoległych) dotyczą drugiego podejścia koncepcyjnego opisanego powyżej: Zadaniem realizowanym równoległe w **jednym trybie podziału** pracy może być wykreślenie wszystkich wymaganych wielokrotności **podzbioru liczb pierwszych**.
2. Zadaniem realizowanym równoległe w **drugim trybie podziału** pracy może być wykreślenie **wielokrotności mających ograniczony zakres wartości**, ale są to wielokrotności wszystkich wymaganych liczb.
3. Równoważenie pracy procesorów poprzez odpowiednie ustalenie liczby wątków OMP i odpowiedni sposób przydziału pracy dla wątków. Można zastosować wskazanie pracy dla dowolnego wątku (dyrektywy single, sections, section) oraz dyrektywę podziału pracy w pętli for w różnorodnych wariantach - statyczny blokowy, warianty statyczne cykliczne, statyczne dynamiczne. Równoważenie obciążenia powinno wziąć pod uwagę możliwość określenia i możliwy rozrzut kosztów obliczeń. Pomocne mogą tutaj się okazać eksperymenty obliczeniowe określające czas upływający podczas przetwarzania oraz stopień zrównoważenia pracą poszczególnych procesorów.

#### Przygotowanie systemu obliczeniowego

W celu wykonania projektu należy użyć komputera z procesorem wielordzeniowym z zainstalowanym oprogramowaniem:

Visual Studio (testowano oprogramowanie Visual Studio 2019 dostępne dla studentów Informatyki Politechniki Poznańskiej przez serwer oprogramowania Politechniki Poznańskiej) oraz oprogramowaniem

Intel Parallel Studio XE udostępniane jest ono studentom Politechniki Poznańskiej przez stronę:

<https://software.intel.com/en-us/parallel-studio-xe/choose-download/student-windows> po rejestracji

Przed instalacją proszę sprawdzić czy środowisko sprzętowo-programowe jest wystarczające do instalacji (Minimum System Requirements).

#### Wymagane elementy pakietu oprogramowania

W celu minimalizacji czasu i transferu danych proponuję pobranie pakietu oprogramowania udostępnianego studentom (dla nauczycieli jest to Intel® Parallel Studio XE Cluster Edition for Windows (wersja 2020 Initial Release)) ograniczonego do następujących elementów składowych (cały pakiet to ponad 3 GB) – z poniżej opisanego pakietu korzystałem:

Intel C++ Compiler 19.1

Intel Math Kernel Library 2020 for C/C++ Intel for IA-32

Intel Integrated Performance Primitives 2020 for IA-32

Intel Threading Building Blocks 2020

Intel VTune Profiler 2020

Intel Inspector 2020

Intel Advisor 2020

Intel Trace Analyzes and Collector 2020

Ze względu na brak możliwości występowania przy pobierania oprogramowania w roli studenta nie testowałem tej procedury. Z tego powodu proszę osoby, które przejdą przez instalację o pomoc tym, którzy będą mieli trudności. W poprzednich latach studenci, chcący korzystać z tego oprogramowania samodzielnie pozyskiwali je i nie zgłaszali trudności.

Trudności związane z dostępnością sprzętu i instalacją oprogramowania oraz jego użyciem proszę zgłaszać do prowadzącego zajęcia.

#### Wersje kodu

• Wymagane jest przygotowanie kilku wersji kodu zgodnie z wymaganiami zadania. Wersje algorytmów rozwiązujących program powinny powstawać częściowo jako wersje wstępne do wykonania wstępnych eksperymentów dotyczących;

- poprawności uzyskiwanych wyników
- pomiaru czasu upływającego podczas przetwarzania
- zrównoważenia pracy użytych procesorów.

Zbiór kodów należy uzupełnić kolejnymi wersjami procedury obliczającej, będącymi próbami ulepszenia wyników uzyskanych we wcześniejszych eksperymentach. Proszę realizując eksperyment kontrolować zależność pomiędzy numerem eksperymentu, trybem realizacji eksperymentu a testowanym kodem, aby właściwie móc opracować wyniki przetwarzania.

Wymagane jest aby wszystkie kody źródłowe używane w eksperymentach zostały umieszczone w sprawozdaniu w części wstępnej, gdzie zostaną scharakteryzowane pod względem użytego podejścia do organizacji przetwarzania – użyte struktury danych, dostęp do danych - współdzielenie, podział pracy i dominujące operacje arytmetyczne.

#### Uruchomienia przetwarzania

Uruchomienia przetwarzania proszę wykonywać w optymalnej dla efektywności konfiguracji pozwalającej wykonać eksperyment oceny efektywności (Release, x86)

Badane instancje powinny zapewnić (wyznaczenie parametry MAX):

- przetwarzanie trwające odpowiednio długo dla zapewnienia poprawności metody zbierania informacji o przebiegu przetwarzania przez program oceny jakości, lecz nie dłużej niż około 1 minuta (dla dowolnej liczby procesorów i wątków)
- możliwość zapisu wyników obliczeń w jednej strukturze: `byte matrix[ROWS]`;
- warianty bazujące na różnych podejściach koncepcyjnych ze względu na różne wydajności mogą być testowane dla różnych instancji
- Testowane instancje:
  - 2...MAX
  - MAX/2...MAX
  - 2...MAX/2
- Liczba użytych procesorów:
  - Przetwarzanie sekwencyjne – jeden procesor,
  - Przetwarzanie równoległe dla maksymalnej liczby dostępnych w systemie procesorów logicznych,
  - Przetwarzanie równoległe dla maksymalnej liczby procesorów fizycznych - jeśli jest inna niż liczba procesorów logicznych,
  - Przetwarzanie równoległe dla liczby procesorów równej  $\frac{1}{2}$  liczby procesorów fizycznych (jeśli jest to liczba inna niż użyta we wcześniejszych wykonanych eksperymentach).

Uwagi do postaci kodu:

Ze względu na minimalizację kosztów związanych z synchronizacją w OpenMP:

- Wprowadzamy w kodzie jedną dyrektywę tworzącą wątki `#pragma omp parallel` dla minimalizacji narzutu związanego z ponownym uruchomieniem zbioru wątków i wstrzymaniem-synchronizacją pracą wątków.
- Wszystkie niepotrzebne synchronizacje wbudowane po analizie poprawności przetwarzania (obecności wyścigu) usuwamy za pomocą klauzuli `nowait`
- Minimalizacja unieważnień linii pamięci podręcznej w ramach współdzielonych danych przez różne procesory poprzez minimalizację zbędnych zapisów będących przyczyną unieważnienia kopii linii pp.  
Na potrzeby testów poprawności opcja kodu pozwala na wyświetlenie liczb pierwszych na konsoli w wierszach po 10 liczb **oraz** wyświetla liczbę liczb pierwszych w badanym zakresie: np. w zakresie od 50 do 100 znaleziono 10 liczb pierwszych.

Problemy do podjęcia w trakcie przygotowania kodu (ich konsekwencje dla tworzonego kodu oraz przebiegu eksperymentu) i analizy wyników jakości przetwarzania:

- lokalność dostępu do danych: wynikająca z przebiegu algorytmu, przetwarzania równoległego (unieważnianie linii pamięci podręcznej procesora), podziału pracy na wątki,
- zrównoważenie obciążeń procesorów: uwzględnienie czasu wykonania i rozrzutu wartości czasu wykonania zadania równoległego oraz wynikających z tego faktu uzasadnień dla statycznego i dynamicznego podziału pracy, ocena jakości zrównoważenia na podstawie wyników pracy oprogramowania testującego.
- rodzaj wykonywanej operacji podstawowej podczas obliczeń: wariant 1 koncepcji - dzielenie, wariant 2 koncepcji – dodawanie.

Określenie parametrów uruchomień i zdarzeń procesora mierzonych podczas eksperymentu, które **zostaną użyte** do oceny prędkości i przyspieszenia oraz wyjaśnienia przyczyn obserwowanych wartości miar efektywności .

## Sprawozdanie z projektu i eksperymentu obliczeniowego

Wstęp:

- Nazwa zaliczanego przedmiot; laboratorium z przetwarzania równoległego
- Imiona i nazwiska autorów sprawozdania, numery indeksów, numer grupy dziekańskiej i termin zajęć laboratoryjnych.
- Terminy: wymagany i termin rzeczywisty oddania sprawozdania.
- Informacja czy sprawozdanie jest wersją pierwszą czy też wersją poprawianą – w przypadku poprawiania sprawozdania wymagane jest zaznaczenie w sposób czytelny modyfikowanych fragmentów sprawozdania.
- Krótki opis treści realizowanego zadania
- Adres email kontaktowy do autorów sprawozdania.

### Punkt 1:

- Opis wykorzystanego systemu obliczeniowego:
  - oznaczenie używanego procesora i jego parametry: liczba procesorów fizycznych, liczba procesorów logicznych, liczba uruchamianych w systemie wątków, oznaczenie typu procesora, wielkość i organizacja pamięci podręcznych procesora, wielkość i organizacja bufora translacji adresów procesora (jeśli takie informacje są dostępne)
  - wersja systemu operacyjnego, wersja i nazwa oprogramowania użytego do przygotowania kodu wynikowego i przeprowadzenia testów.

### Punkt 2:

- Prezentacja przygotowanych wariantów kodów z wyjaśnieniem przewidywanego przebiegu przetwarzania. W sprawozdaniu powinny znaleźć się kluczowe elementy kodów charakterystyczne dla poszczególnych wersji kodu. Kolejne wersje kodu powinny zostać oznaczone w sposób skrótowy jednoznaczny i nawiązujący do wariantu użytego algorytmu i sposobu podziału pracy wątków. Oznaczenia zostaną użyte w dalszej części sprawozdania: w prezentacji wyników eksperymentów i wniosków związanych z poszczególnymi wariantami kodu. W opisie przebiegu przetwarzania (dla poszczególnych wariantów kodu) należy uwzględnić:
  - zagadnienie **podziału** pracy – wielkość zbioru zadań, który powstaje w przetwarzaniu równoległym,
  - sposób **przydziału** zadań do procesorów – kiedy i w jakiej liczbie zadania są przydzielane do procesorów
  - uzasadnić wybrany sposób podziału przetwarzania na zadania i sposób przydziału zadań w powiązaniu ze spodziewanym zrównoważeniem/brakiem zrównoważenia procesorów przetwarzaniem.
  - omówić dyrektywy i klauzule Open MP użyte w kodzie i ich znaczenie dla poprawności przebiegu obliczeń.
  - omówić występujące w kodzie potencjalne problemy poprawnościowe: wyścig (co to jest wyścig, dlaczego w kodzie nie występuje).
  - omówić występujące w kodzie potencjalne problemy efektywnościowe:
    - false sharing (co to jest false sharing i jak często (w stosunku do podstawowej operacji dodawania lub dzielenia) będzie występował podczas obliczeń).
    - synchronizacja – czy i gdzie w kodzie występuje i ja wpływa na czas obliczeń.

### Punkt 3:

Prezentacja wyników i omówienie przebiegu eksperymentu obliczeniowo-pomiarowego:

- a) omówienie jakie kodu (wymienione w sprawozdaniu) były testowane, jakich instancji przetwarzanie dotyczyło, dla jakich instancji (reprezentatywnych do porównań charakterystycznych wariantów przetwarzania) uruchomiono ocenę jakości przetwarzania i prezentowane są wyniki szczegółowe przetwarzania na poziomie miar jakości obliczonych na podstawie zdarzeń procesora.
- b) opisanie własnymi słowami sposób zbierania przez oprogramowanie Intel Parallel Studio XE informacji o efektywności przetwarzania, określenie znaczenia zebranych i prezentowanych w sprawozdaniu informacji; informacja o wykorzystanych trybach pracy narzędzi oprogramowania Intel Parallel Studio XE
- c) tabela z wynikami z czytelnym określeniem jakiego uruchomienia: testowany wariant kodu i parametrami instancji; (warto aby tabela była jedna – zbiorcza zorientowana poziomo na stronie dla lepszego wykorzystania miejsca) i w sposób czytelny prezentowała kluczowe parametry, które są z wynikami eksperymentu oraz miary wyliczone:
  - czas przetwarzania – czas upływający podczas przetwarzania od momentu rozpoczęcia obliczeń do gotowości tablicy zawierającej jednoznaczne informacje o zbiorze liczb pierwszych Tobl lub Elapsed time (wyznaczony w ramach eksperymentu Microarchitecture Exploration dla Intel Vtune Profiler)
  - liczba instrukcji kodu asemblera (instructions retired)
  - liczba cykli procesora w czasie wykonywania badanego kodu – (clockticks)
  - udział procentowy wykorzystanych zasobów procesora do przetwarzania kodu (retiring)
  - „ograniczenie wejścia” - udział procentowy w ograniczeniu efektywności przetwarzania części wejściowej procesora (front-end bound)
  - „Ograniczenie wyjścia” - udział procentowy w ograniczeniu efektywności przetwarzania części wyjściowej procesora (back-end bound)
  - „Ograniczenie systemu pamięci” - udział procentowy w ograniczeniu efektywności przetwarzania systemu pamięci (memory bound)
  - „Ograniczenie jednostek wykonawczych” - udział procentowy w ograniczeniu efektywności przetwarzania jednostek wykonawczych procesora (core bound)
  - Efektywne wykorzystanie rdzeni fizycznych procesora – „effective physical core utilization” – dodatkowe informacje jeśli występuje niezrównoważenie – na podstawie widoku Effective CPU Utilization Histogram lub widoku Bottom-up w ramach eksperymentu Microarchitecture Exploration dla Intel Vtune Profiler.
  - przyspieszenie przetwarzania równoległego dla badanego wariantu kodu równoległego – parametr ten jest ilorazem czasu przetwarzania najlepszego dostępnego przetwarzania równoległego (dowolnym algorytmem w tym samym systemie) oraz czasu przetwarzania równoległego, dla którego przyspieszenie jest wyznaczane.
  - prędkość przetwarzania liczona jako liczba przetestowanych liczb w jednostce czasu,
  - efektywność przetwarzania równoległego jako iloraz przyspieszenia przetwarzania równoległego i liczby użytych w przetwarzaniu procesorów fizycznych.
- d) wartości parametrów powinny być prezentowane w formacie pozwalającym na ich czytelne porównanie - wartości wraz z jednostkami prezentowanych wartości: przykładowo: czas przetwarzania 0,0053 s, prędkość obliczeń 1,23E+05 1/s (liczba zbadanych liczb na sekundę obliczeń).
- e) oprócz powyżej wymienionych miar efektywności obliczeń można przedstawić dodatkowe miary szczegółowe dostępne w wynikach przetwarzania, które w przypadku gdy wąskim gardłem ograniczającym wzrost prędkości, będą określały jaki element systemu pamięci powoduje opóźnienie przetwarzania: dostęp do poziomów pamięci podręcznej L1, L2, L3, dostęp do danych w pamięci operacyjnej, czy ograniczenia wynikające z dostępu do adresu wirtualnego w buforze translacji (DTLB lub STLB). Taka analiza choć nieobowiązkowa (umożliwia uzyskanie wyższej oceny), przy dobrym zrozumieniu miar i ich prezentacji znaczenia pozwala na głębsze spojrzenie w przyczyny braku pełnej wydajności przetwarzania kodu.

### Punkt 4

Wnioski. W wnioskach należy:

- porównać jakość rozwiązań problemu przy użyciu różnych zaproponowanych w badanych kodach podejść – w oparciu o prędkość przetwarzania lub czas obliczeń przy jednakowych instancjach problemu.
- podsumować czy użyte podejścia efektywnie wykorzystują struktury wewnętrzne procesora, jakie wąskie gardła pojawiają się w poszczególnych wariantach kodu i czy jest możliwość poprawy w tym zakresie lub znaleziono rozwiązanie, które pozwala na lepsze wykorzystanie jednostek wewnętrznych procesora, czy są warianty kodu, które w lepszy sposób niż pozostałe wykorzystują zasoby procesora i dostarczają przetwarzania przewyższającego efektywnością inne podejścia.

- podsumować użyte podejścia pod kątem jakości zrównoleglenia przetwarzania. W tej analizie należy wykorzystać przede wszystkim przyspieszenie, efektywność oraz dodatkowo „efektywne wykorzystanie rdzeni fizycznych procesora”; ostatni parametr jest tutaj miarą pomocniczą gdyż efektywne - zrównoważone wykorzystanie rdzeni fizycznych nie zawsze doprowadza do obliczeń o najkrótszym czasie.
- określić, które podejście okazało się najlepsze/najgorsze pod względem prędkości przetwarzania, czy ta przewaga jest niezależna od liczby użytych procesorów i rozmiaru testowanej instancji. Jakie są przyczyny tych wyników.
- Jakie podejście równoległe jest najbardziej efektywne, czy efektywność skaluje się ze wzrostem liczby procesorów (jest stała, w jaki sposób się zmienia), jak wygląda przegląd efektywności użytych podejść. Jakie są przyczyny tych wyników.
- Jakie ograniczenia efektywnościowe dominują w poszczególnych metodach rozwiązania problemu.
- (proszę używać do porównania miar względnych – np. przetwarzane trwało 2 razy krócej, a nie o 2 sekundy krócej, prędkość wzrosła o 100%, a nie o 10000 zbadanych liczb na sekundę),

W próbie uzasadnienia przyczyn wysokiej lub niskiej efektywności poszczególnych podejść do realizacji kodu proszę odwoływać się w sposób jednoznaczny do **poszczególnych wartości** miar w tablicy wyników, poszczególnych **linii kodu** omawianej wersji kodu (znaczenia występujących w tych liniach konstrukcji językowych), rysunków zawierających efekty pracy oprogramowania Intel Vtune Profiler (zrzuty ekranu) oraz wykresów. Proszę unikać sformułowań ogólnych typu szybki, wolny, lepszy, gorszy oraz uwzględnić znaczenie wartości parametru: jeżeli stosunek trafień do pamięci ma wartość 0,5 lub 0,00001 to jego dwukrotny wzrost ma diametralnie różne konsekwencje na czas przetwarzania, podobnie różne znaczenie dwukrotny wzrost wartości przyspieszenia przetwarzania równoległego wynoszącego 5 lub 0,3.

#### Przygotowanie sprawozdania.

Forma: sprawozdanie w formie elektronicznej wymagane jest w terminie do 27.04. W sprawozdaniu zawierającym powyżej wymienione elementy składowe:

- powinny być ponumerowane strony,
- każdy obiekt sprawozdania poza tekstem ciągłym: listing kodu, rysunek, wzór, tabela powinien posiadać numer kolejny i podpis określający w sposób jednoznaczny i pełny przedmiotową w obiekcie zawartość.

#### Podsumowanie i ocena realizacji projektu

Po sprawdzeniu merytorycznej zawartości zostanie wystawiona ocena, możliwe jest skierowanie sprawozdania nie spełniającego powyżej opisanych wymagań do uzupełnienia i/lub rozmowa z autorami na temat przebiegu eksperymentu i zawartości sprawozdania. Ewentualne uzupełnienia sprawozdania powinny zawierać zaznaczone dodane elementy (sprawozdanie należy uzupełnić w terminie 7 dni).

**Nieuzasadnione opóźnienie** w oddaniu sprawozdania obniża ocenę w stosunku do oceny przyznanej za jakość sprawozdania. Obniżają o ½ stopnia ocenę następujące opóźnienia - po terminie, po tygodniu, po 2 tygodniach, po 4 tygodniach). Opóźnienie większe niż 6 tygodni powoduje wystawienie oceny ndst z pierwszego terminu zaliczenia przedmiotu.

#### Konsultacje:

Trudności praktyczne i koncepcyjne związane z realizacją projektu mogą Państwo zgłaszać przez email. Potrzebę spotkania online dla wyjaśnienia zagadnień proszę zgłaszać również przez email. W przypadku większej liczby zgłoszeń potrzeby konsultacji będą się one odbywały elektronicznie – online regularnie w sposób ogłoszony przez email.

#### Literatura:

Wykłady i literatura przedmiotu  
 Opracowania: Vtune\_koncepcja, praktyczne\_wprowadzenie\_VtuneProfiler  
 Dokumentacja Intel Vtune Profiler

Opis przygotowany: 29.03.2020, **zmiany 31.03.2020 zaznaczone na zielono.**