

Cel: Opanowanie technik określania skalowalności w sensie silnym i w sensie słabym

Kroki:

1. W celu rozszerzonego badania skalowalności algorytmu i programu mnożenia macierz-macierz dla pojedynczego węzła obliczeniowego, dokonaj studium zależności skalowalności w sensie silnym od rozmiaru zadania
2. Utwórz katalog *lab_12* i skopiuj do niego paczkę: *mat_mul_scalability_gcc.tgz*.
3. Rozpakuj paczkę i przejdź do katalogu *matrix_multiplication_gcc*.
4. W domyślnym ustawieniu, program w pliku *mat_mul_driver.c* alokuje tablice z danymi macierzowymi i dokonuje obliczenia iloczynu macierz-macierz za pomocą dwóch funkcji:
 - *dgemm_* z biblioteki MKL firmy Intel
 - *mat_mul_avx_1* z pliku *mat_mul_avx_1.c*Rozmiar macierzy definiowany jest przez parametr `DEFAULT_SIZE`, którego wartość ustalana jest dyrektywą `#define`, np.:
`#define DEFAULT_SIZE 1344`
5. Ćwiczenie można przeprowadzać dla domyślnego ustawienia kodu, można także, co może być ciekawsze poznawczo, wstawić w miejsce rozpakowanego pliku *mat_mul_par.c* najlepszą swoją implementację mnożenia macierz-macierz, a następnie zamienić wywołanie *mat_mul_avx_1* na *mat_mul_par*.
6. W pliku *sizes.h* zdefiniowane są parametry odpowiednie do badania skalowalności:
`#define BLOCK_SIZE_S 4`
`#define BLOCK_SIZE_L (BLOCK_SIZE_S)`
`#define ALIGNMENT 64`
7. W pierwszym kroku należy uruchomić program mnożenia macierz-macierz z domyślnymi parametrami za pomocą *make* i sprawdź poprawność działania (jak zwykle krokiem wstępnym jest utworzenie biblioteki interfejsu z PAPI: *make recreate_papi_lib_mth*). W pliku Makefile zadane jest automatyczne uruchomienie programu w wersji ze zliczaniem zdarzeń sprzętowych i przypisaniem wszystkich wątków do jednego rdzenia.
8. Właściwe badanie skalowalności polega na uruchamianiu kodu bez zliczania zdarzeń sprzętowych
./mat_mul_driver.exe
9. dla tablic o rozmiarze zadanym przez parametr `DEFAULT_SIZE`, z różną liczbą wątków określaną przez zadaną wartość zmiennej środowiskowej `OMP_NUM_THREADS`:
export OMP_NUM_THREADS=...
10. Badanie skalowalności w sensie silnym należy przeprowadzić dla co najmniej dwóch rozmiarów macierzy z sekwencji zapisanych w pliku *mat_mul_driver.c* (linie 60-75). Pierwsza z sekwencji dotyczy relatywnie małych zadań (rozmiar od 448 do 1152), takich dla których powinno być możliwe uzyskanie wiarygodnych wyników podczas zajęć, kiedy wiele osób korzysta z serwera. Druga z sekwencji dotyczy zadań większych (rozmiar od 1344 do 3840), o czasach wykonania znacząco dłuższych (badając tę sekwencję można ograniczyć się do wywołania tylko *dgemm_*, które pracuje najszybciej, wykomentowując resztę kodu).
[dla badania skalowalności obliczeń w funkcji *mat_mul_avx_1* najlepiej wybrać, jako oddające charakterystykę zjawiska lepszej skalowalności większych zadań, rozmiary 448 i 1344]
Należy uwzględnić liczbę rdzeni 1, 2, 4, 8, 16 i zapewnić ich odpowiednie przypisanie do rdzeni korzystając ze zmiennej środowiskowej `KMP_AFFINITY`:
export KMP_AFFINITY=scatter,1
(raz zadanym parametrów nie trzeba ustawiać przed każdym uruchomieniem)
11. Wyniki pomiarów czasu wykonania zapisz w tabeli (jak zwykle należy uruchomić program kilkakrotnie i wybrać najkrótsze czasy wykonania – przy stosowaniu "przyypinania" wątków,

wyniki powinny cechować się mniejszym rozrzutem). Utwórz osobne tabele dla różnych wersji implementacji – co najmniej 2 tabele: jedną dla *mat_mul_avx_1* lub implementacji własnej i drugą dla funkcji *dgemv* z biblioteki MKL (czasy podawane dla opcji LAPACK w *mat_mul_driver.exe*)

12. Dla zebranych danych należy utworzyć odpowiednie wykresy przyspieszenia obliczeń. Na jednym wykresie powinny znaleźć się krzywe dla różnych (co najmniej 2) rozmiarów macierzy. Powinny powstać co najmniej dwa wykresy, jeden dla *mat_mul_avx_1* lub swojej najlepszej wersji mnożenia macierzy i drugi dla procedury *dgemv* z biblioteki MKL. **Na każdym z wykresów powinna pojawić się linia przyspieszenia idealnego.**
13. W celu zbadania skalowalności w sensie słabym należy obliczenia przeprowadzić dla wszystkich rozmiarów z wybranej sekwencji z pliku *mat_mul_driver.c*. Rozmiary w sekwencji są dobrane tak, że liczba operacji algorytmu zwiększa się w przybliżeniu dwukrotnie dla kolejnych wartości z sekwencji. Oznacza to, że jeśli zadania dla kolejnych rozmiarów będą uruchamiane dla liczby rdzeni zwiększającej się dwukrotnie (448 dla 1, 576 dla 2, 704 dla 4, 896 dla 8 i 1152 dla 16 rdzeni), w efekcie rozmiar na pojedynczy rdzeń pozostanie stały - tak jak wymaga tego badanie skalowalności w sensie słabym.
[Na zajęciach najlepiej wybrać pierwszą sekwencję, obliczenia dla drugiej sekwencji można przeprowadzić, kiedy maszyna nie będzie współdzielona z innymi użytkownikami.]
14. Dla każdego z rozmiarów należy przeprowadzić dwa pomiary czasu wykonania: jeden dla jednego rdzenia i drugi dla liczby odpowiadającej stałemu rozmiarowi na rdzeń. Wyniki należy umieścić w tabeli. Na podstawie wyników eksperymentów należy stworzyć co najmniej dwa wykresy przyspieszenia przeskalowanego odpowiadające badaniu skalowalności w sensie słabym: jeden dla *mat_mul_avx_1* lub swojej najlepszej procedury, drugi dla procedury *dgemv*. Dane dla skalowalności w sensie słabym należy odczytywać z tabeli czasów wzdłuż przekątnej głównej (dane dla skalowalności w sensie silnym to dane w pojedynczym wierszy lub kolumnie – dla stałego rozmiaru zadania). **Na każdym z wykresów powinna pojawić się linia przyspieszenia idealnego.**
15. Jako dodatkowy element badania skalowalności w sensie słabym skonstruuj wykres czasu wykonania zadania jako funkcji liczby wątków (rdzeni), przy założeniu stałego rozmiaru zadania na wątek (rdzeń) – (zaczynając do najmniejszego zadania dla jednego wątku, należy uwzględnić czasy zadania p-razy większego dla liczby wątków p).
16. Wyciągnij wnioski co do możliwości skalowania obliczeń w jednym węźle dla mnożenia macierz-macierz
17. Pobierz ze strony przedmiotu i rozpakuj paczkę *mat_vec_row_MPI.tgz*
18. Uruchom kod przez wykonanie poleceń *make* i *make run*. Zaobserwuj możliwość sterowania wykonaniem przez podanie w linii komend parametru *-np* określającego liczbę procesów MPI oraz opcji oznaczających przypisanie procesów do kolejnych rdzeni mikroprocesorów serwera: (*/usr/bin/mpirun -np x --bind-to core --map-by core mat_vec_row_MPI*)
19. Przeprowadź badanie **skalowalności w sensie słabym** dla funkcji mnożenia macierz-wektor. Rozważ sekwencję wymiarów (np. 3036, 4294, 6072, 9600, 13580, 19200 – linia 9 w pliku *mat_vec_test.c*) dobraną tak, aby praca (liczba operacji zmiennoprzecinkowych) mnożenia macierz-wektor rosła proporcjonalnie do liczby rdzeni w sekwencji 1,2,4,10,20,40. Wyniki pomiarów czasu zapisz w tabeli (dla każdej liczby wątków dwa pomiary – zgodnie z metodologią badania skalowalności w sensie słabym).
20. Utwórz odpowiedni wykres dla skalowalności w sensie słabym. **Na wykresie powinna pojawić się linia przyspieszenia idealnego.**
21. Jako dodatkowy element badania skalowalności w sensie słabym skonstruuj wykres czasu wykonania zadania jako funkcji liczby wątków (rdzeni), przy założeniu stałego rozmiaru zadania na wątek (rdzeń) – (zaczynając do najmniejszego zadania dla jednego wątku, należy uwzględnić czasy zadania p-razy większego dla liczby wątków p).
22. Wyciągnij wnioski co do możliwości skalowania obliczeń w jednym węźle dla mnożenia macierz-wektor. Wnioski dotyczą użytej platformy, w tym implementacji MPI.