
Programowanie systemów z pamięcią wspólną - specyfikacja OpenMP

OpenMP

- Przenośność oprogramowania
- Model SPMD
- Szczegółowe wersje (*bindings*) dla różnych języków programowania
- Elementy składowe:
 - dyrektywy dla kompilatorów
 - funkcje biblioteczne
 - zmienne środowiskowe

OpenMP – składnia dyrektyw

- format (dla powiązania z językami C i C++):
`#pragma omp nazwa_dyrektywy lista_klauzul znak_nowej_linii`
- najważniejszymi z dyrektyw są dyrektywy podziału pracy (*work sharing constructs*), występujące w obszarze równoległym i stosowane do rozdzielenia poleceń realizowanych przez poszczególne procesory
- najważniejsze klauzule określają sposób traktowania zmiennych przez wątki w obszarze równoległym
- każda dyrektywa posiada swój własny zestaw dopuszczalnych klauzul

OpenMP – składnia dyrektyw

→ **parallel**

```
#pragma omp parallel lista_klauzul  
{ /* obszar równoległy */ }
```

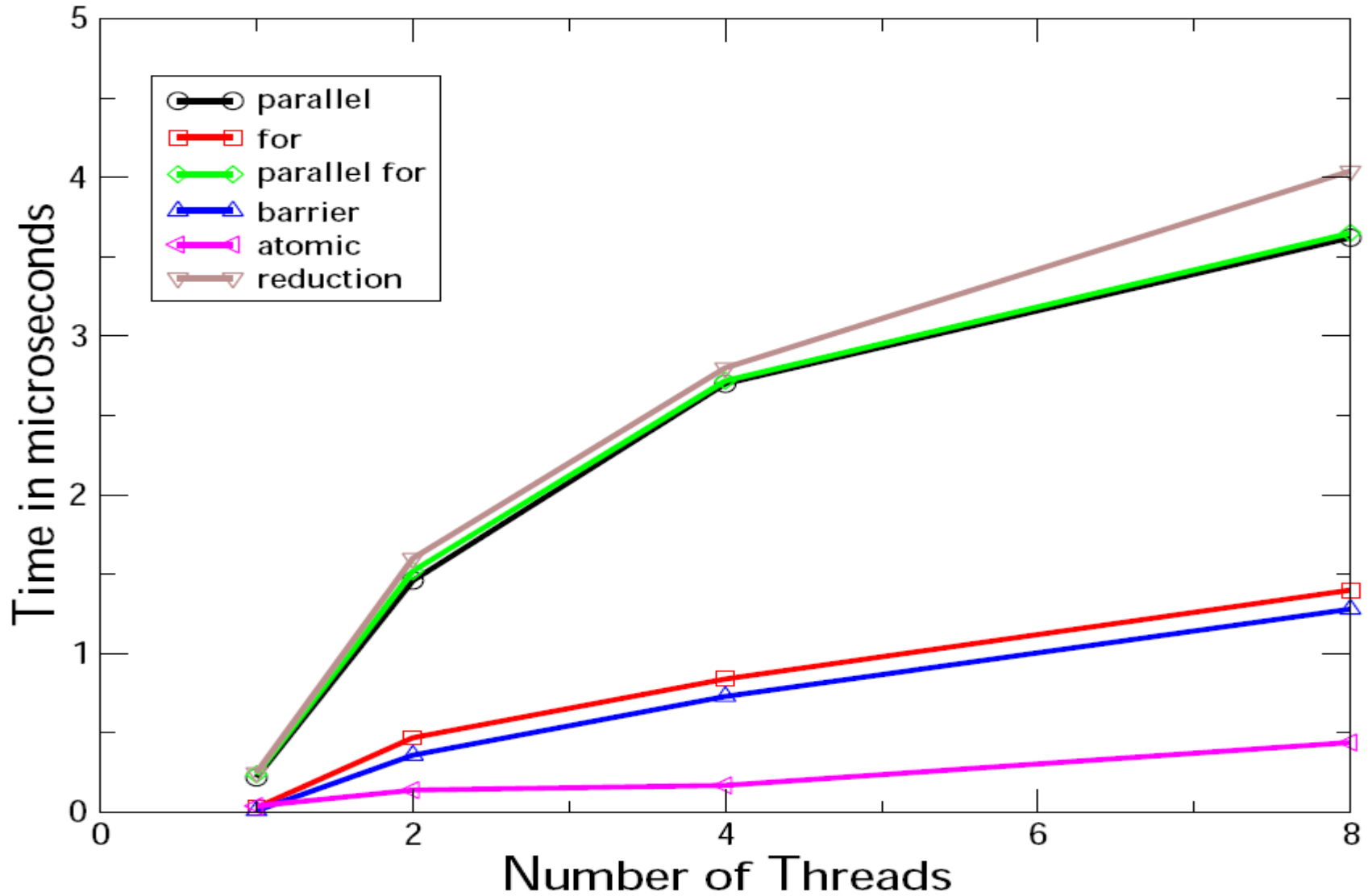
→ *lista_klauzul* (pusta lub dowolna kombinacja poniższych):

- **if(*warunek*)**
- **num_threads (*liczba*)**
- *klauzule_zmiennych* (**private, firstprivate, shared, reduction**) - *za chwilę*
- inne

OpenMP – liczba wątków

- liczbę wątków można próbować określić jawnie poprzez:
 - użycie klauzuli `num_threads` w dyrektywie `parallel`, np.:
`#pragma omp parallel num_threads(10)`
 - wywołanie procedury `omp_set_num_threads`, np.:
`omp_set_num_threads(10);`
 - ustalenie zmiennej środowiskowej `OMP_NUM_THREADS`, np.:
`$ export OMP_NUM_THREADS=10`
- w pozostałych przypadkach liczba wątków jest ustalana przez implementację OpenMP w danym systemie
- istnieje możliwość dynamicznego ustalania liczby wątków (aby np. umożliwić działanie dla systemów, które nie dysponują liczbą określoną poprzez `num_threads`)

Narzut OpenMP



OpenMP – traktowanie zmiennych

- klauzule współdzielenia zmiennych:
 - **shared** – zmienna wspólna wątków
 - **private** – zmienna lokalna wątków
 - **firstprivate** – zmienna lokalna wątków z kopiowaną wartością początkową
 - **lastprivate** – zmienna lokalna wątków z wartością końcową równą wartości jaka byłaby przy wykonaniu sekwencyjnym
 - inne
- dyrektywa **threadprivate** (zasięg ważności deklaracji jest taki jak zasięg ważności deklarowanych nazw: dla zmiennych statycznych plik, dla globalnych cały kod, zmienne automatyczne nie mogą być objęte dyrektywą)

#pragma omp threadprivate (lista_zmiennych) znak_nowej_linii

OpenMP – traktowanie zmiennych

- **zmienna jest wspólna** (dostępna wszystkim wątkom) jeśli:
 - istnieje przed wejściem do obszaru równoległego i nie występuje w dyrektywach i klauzulach czyniących ją prywatną
 - została zdefiniowana wewnątrz obszaru równoległego jako zmienna statyczna
- **zmienna jest prywatna** (lokalna dla wątku) jeśli
 - została zadeklarowana dyrektywą **threadprivate**
 - została umieszczona w klauzuli **private** lub podobnej (**firstprivate, lastprivate, reduction**)
 - została zdefiniowana wewnątrz obszaru równoległego jako zmienna automatyczna
 - jest zmienną sterującą równoległej pętli **for**

OpenMP – pozostałe dyrektywy

→ **critical**

`#pragma omp critical nazwa znak_nowej_linii {...}`

→ **atomic**

`#pragma omp atomic znak_nowej_linii`

→ **flush**

`#pragma omp flush lista_zmiennych znak_nowej_linii`

→ **barrier**

`#pragma omp barrier znak_nowej_linii`

→ **ordered**