

POLITECHNIKA POZNAŃSKA

WYDZIAŁ INFORMATYKI

PRZETWARZANIE RÓWNOLEGŁE

Równoległe sumowanie komórek pamięci za pomocą wielu wątków przetwarzania

Autorzy:

Adam SZCZEPAŃSKI
Mateusz CZAJKA

Prowadzący:

dr Rafał WALKOWIAK



18 stycznia 2014

Spis treści

1	Informacje o projekcie	2
1.1	Dane autorów	2
1.2	Historia projektu	2
2	Wstęp	3
2.1	Opis problemu	3
2.2	Zastosowane algorytmy	3
2.2.1	Podstawowy algorytm sekwencyjny	3
2.2.2	Algorytm sekwencyjny - sekcynność pamięci	3
2.2.3	Algorytm zrównoleglony - wpierw wiersze	3
2.2.4	Algorytm zrównoleglony - wpierw kolumny	3
	Spis rysunków	3
	Spis tablic	3

1 Informacje o projekcie

1.1 Dane autorów

Mateusz Czajka 106596

Adam Szczepański 106593

1.2 Historia projektu

1. Jest to pierwsza wersja projektu. Dokumentacja elektroniczna została przesłana w dniu 20 stycznia 2013.

2 Wstęp

2.1 Opis problemu

Głównym założeniem projektu było zapoznanie się biblioteką OpenMP na podstawie równoległego sumowania komórek tablicy. W ramach projektu zrealizowaliśmy 2 algorytmy sekwencyjne oraz 2 algorytmy zrównoleglone. Celem dwóch różnych algorytmów sekwencyjnych było zbadanie wpływu sekwencyjności pamięci podręcznej na czas realizacji zadania. W przypadku algorytmów zrównoleglonych badaliśmy wpływ kolejności uszeregowania pętli na końcowy rezultat.

2.2 Zastosowane algorytmy

2.2.1 Podstawowy algorytm sekwencyjny

```
el_type sums() {
    el_type sum = 0;

    for (size_t i = 0; i < TAB_SIZE; i++) {
        size_t& k = i;
        sum += tab[k];
    }

    return sum;
}
```

2.2.2 Algorytm sekwencyjny - sekwencyjność pamięci

```
el_type sums2() {
    el_type sum = 0;

    for (int j = 0; j < 32; j++) {
        for (size_t i=0; i<TAB_SIZE; i+=32) {
            int k = i + j;
            sum += tab[k];
        }
    }

    return sum;
}
```

2.2.3 Algorytm zrównoleglony - wpierw wiersze

```
el_type sum_omp_reduction() {
    int i;
    el_type sum = 0;
```

```

#pragma omp parallel for default(none) shared(tab) private(i) reduction(+:sum) // schedule(1)
    for (i=0; i<TAB_SIZE; i++) {
        sum += tab[i];
    }

    return sum;
}

```

2.2.4 Algorytm zrównoleglony - wpierw kolumny

```

el_type sum_omp_reduction2() {
    int j;
    el_type sum = 0;

#pragma omp parallel for default(none) shared(tab) private(j) reduction(+:sum)
    for (j = 0; j < 32; j++) {
        for (size_t i=0; i<TAB_SIZE; i+=32) {
            int k = i + j;
            sum += tab[k];
        }
    }

    return sum;
}

```

Spis rysunków

Spis tablic