

Dr. Varga Imre
Debreceni Egyetem, Informatikai Kar

OpenMP összefoglaló

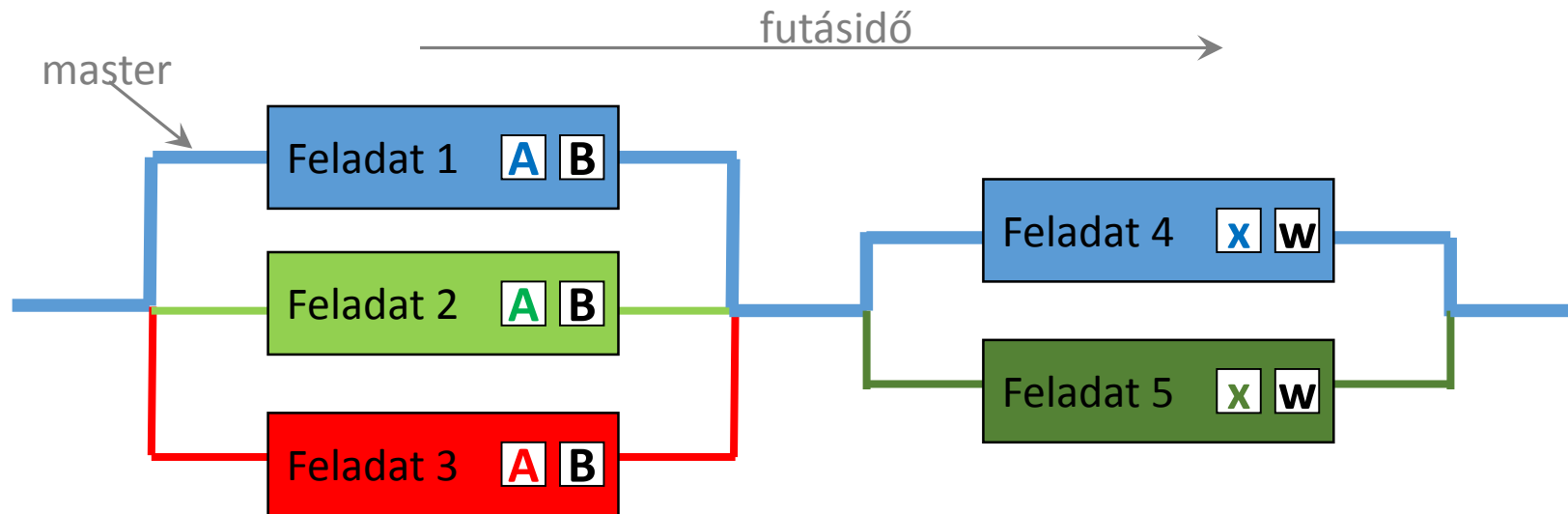
OpenMP

Open Multi-Processing API

- Osztott memória modell
- Folyamatos fejlesztés 1997 óta
- Támogatott nyelvek: C, C++, Fortran
- Támogatott operációs rendszer: Windows, Linux
- Tartalmaz:
 - Fordítási direktívákat
 - Könyvtári függvényeket
 - Környezeti változókat

OpenMP

- Folyamaton belül párhuzamosan futó szálak
 - Egyik szál kitüntetett: master
- Feladat- és adat párhuzamosság



OpenMP direktívák

```
#pragma omp parallel
```

- A következő utasítás fusson több szálon is!

```
#pragma omp parallel num_threads(4)
```

- A következő utasítás fusson 4 szálon egyszerre!

```
#pragma omp parallel private(x,y) shared(a,b)
```

- A következő blokkban deklarált változók közül
x és y legyen minden szálon egyedi
a és b legyen minden szálon közös

OpenMP direktívák

```
#pragma omp parallel for
```

- A következő for ciklus lépéseit oszd fel a szálak között egyenletesen!

```
#pragma omp parallel for schedule(guided)
```

- A következő for ciklus lépéseit oszd fel a szálak között terheléstől függően!

```
#pragma omp parallel for reduction(+:Sum)
```

- A következő for ciklus elosztott lefutása után az egyes privát Sum változó értékeket add össze!

OpenMP direktívák

`#pragma omp parallel sections`

- Hozz létre párhuzamosan futó szakaszokat!

`#pragma omp section`

- A következő blokk legyen az imént létrehozott egyik párhuzamosan futó szakasz!

`#pragma omp master`

- A következő utasítást csak a master futassa!

`#pragma omp single`

- A következő utasítást csak az egyik szál futassa!

OpenMP direktívák

```
#pragma omp barrier
```

- Hozz létre itt egy szinkronizációs pontot! Egyik szál sem mehet addig tovább amíg mindegyik el nem ért ideig.

```
#pragma omp critical
```

- A következő utasítás (blokk) alkotson kritikus szakaszt, azaz egyszerre csak egy szál futtathassa ezt a részt (a többiek csak azután hogy az előző befejezte). Kölcsönös kizárás.

OpenMP könyvtári függvények

```
int omp_get_num_procs()
```

- Visszaadja az elérhető processzormagok számát

```
int omp_get_num_threads()
```

- Visszaadja a szálak aktuális számát

```
void omp_set_num_threads()
```

- Beállítja hány szálon fussanak a párhuzamos szakaszok

```
int omp_get_thread_num()
```

- Visszaadja az adott szál azonosítóját

OpenMP példa

```
#include<stdio.h>
#include<omp.h>          // gcc openmp.c -fopenmp
int IsSame(int *A, int N){
    int i, Sum1=0, Sum2=0;
    #pragma omp parallel for
        for(i=0;i<N;i++){
//            #pragma omp critical
                Sum1=Sum1+A[i];
        }
    #pragma omp parallel for schedule(guided) reduction(+:Sum2)
        for(i=0;i<N;i++)
            Sum2=Sum2+A[i];
    return Sum1==Sum2;
}
```