

# Paralelizace smyček v OpenMP

AVS - Architektury výpočetních systémů  
Cvičení 4, 2023 / 2024

**Jirka Jaroš**

Vysoké učení technické v Brně, Fakulta informačních technologií  
Božetěchova 1/2, 612 66 Brno - Královo Pole  
jarosjir@fit.vutbr.cz



# PŘIHLÁŠENÍ NA BARBORU

- **Připojte se na Barbora**

- Otevřete si dva terminály (jeden pro tooly, jeden pro kompilaci, atd.)
- Kdo chce pracovat na domácím PC a má intel tooly, může.

```
ssh barbora
```

- **Vaše PC: Připojte si disk z Barbory a nakopírujte tam obsah 4. cvičení**

```
mkdir /tmp/barbora  
sshfs barbora: /tmp/barbora
```

- **Natáhněte modul kompilátoru, Advisoru a VTune**

```
ml Advisor VTune intel
```

- **Doplňte zdrojový kód**

- Zjistěte id vlákna a počet vláken v sekvenční i paralelní části.
- Doplňte vhodnou omp pragmu
- **Nedefinujte žádné další proměnné**

```
void parallelHelloWorld(int numOfWorkers)
{
    // 1. Find out the thread id and number of threads in the seq. region
    int threadId = -1;
    int threadNum = 0;

    // 2. Run this block by a given number of threads (numOfWorkers)
    {
        // 3. Find out the thread id and number of threads in the par. region
        threadId = -1;
        threadNum = 0;

        printf(" - Hi from parallel region. I'm %d of %d.\n", threadId, threadNum);
    }

    // Back in the sequential region
    fflush(stdout);
    printf("-----\n");
    printf(" - Hi from serial region. I'm %d of %d.\n", threadId, threadNum);
    // seq part
} // end of parallelHelloWorld
```

- **Zkompilujte a spusťte kód**

```
make && make run
```

## Správný výstup

```
./lab4
-----
Parallelization with OpenMP
Running on: sc-gpu1
Number of cores: 12
-----
- Test 1: Parallel Hello world
- Hi from parallel region. I'm 0 of 20.
- Hi from parallel region. I'm 12 of 20.
- Hi from parallel region. I'm 3 of 20.
- Hi from parallel region. I'm 2 of 20.
- Hi from parallel region. I'm 11 of 20.
- Hi from parallel region. I'm 13 of 20.
- Hi from parallel region. I'm 1 of 20.
- Hi from parallel region. I'm 10 of 20.
- Hi from parallel region. I'm 9 of 20.
-----
- Hi from serial region. I'm 0 of 1.
-----
```

<https://www.openmp.org/wp-content/uploads/OpenMP-4.5-1115-CPP-web.pdf>

- **Doplňte zdrojový kód**

- Doplněte dvě omp pragmy pro paralelizaci smyček

```
void matrixMul(float* c, const float* a, const float* b)
{
    // 1. Use omp pragma to clear the c matrix
    for (size_t i = 0; i < N * P; i++)
    {
        c[i] = 0.0f;
    }

    // 2. Use omp pragma parallelize matrix multiplication
    // Go over all rows in the result matrix
    #pragma omp parallel for
    for (size_t i = 0; i < N; i++)
    {
        // Go over all cols in the result matrix
        for (size_t k = 0; k < M; k++)
        {
            // Calculate vector dot product of a given row and col
            #pragma omp simd
            for (size_t j = 0; j < P; j++)
            {
                c[i * P + j] += a[i * M + k] * b[k * P + j];
            }
        }
    }
}
} // end of matrixMul
```

- **Zkompilujte a spustěte kód**

```
make && make run
```

- **Změření doby výpočtu s různým počtem vláken**

```
OMP_NUM_THREADS=8 ./lab4
```

- **Otázky**

- Produkuje kód správný výsledek?
- Kolik GFLOP jste dosáhli?
- Jaké je zrychlení vůči sekvenční verzi?
- Jaké je efektivita?

<https://www.openmp.org/wp-content/uploads/OpenMP-4.5-1115-CPP-web.pdf>

- **Doplňte správné pragmy**

- Zjistěte, která smyčka je ideální pro paralelizaci.
- Vhodně omezte režii startování vláken a barrier.
- Zvolte vhodné velikosti pro `blockSizeL3` a `blockSizeL2`

- **Analýza s VTune**

- Odstartujte job s Vtune

```
sbatch vtune.sl  
vtune-gui VTune/lab4/thr/thr.vtune
```

- Ověřte že všechna vlákna mají co dělat (je v paralelizované smyčce dost iterací?)
  - Pokud ne, zamyslete se, jak získat více iterací k paralelizaci.

- **Zkompilujte a spusťte kód**

```
make && make run
```

- **Změření doby výpočtu s různým počtem vláken (upravte job)**

```
sbatch run.sl
```

- **Otázky**

- Produkuje kód správný výsledek?
- Kolik GFLOP jste dosáhli?
- Jaké je zrychlení vůči sekvenční verzi?
- Jaké je efektivita?

<https://www.openmp.org/wp-content/uploads/OpenMP-4.5-1115-CPP-web.pdf>

- **Doplňte správné pragmy**

- Přidejte správnou pragmu pro paralelizaci.
  - ověřte, že kód počítá správně (všechny běhy poskytují stejné výsledky).
- Vyzkoušejte různé varianty plánování smyček:
  - static
  - dynamic
  - guided
- Vyberte vodné velikosti chunku.

- **Zkompilujte a spusťte kód**

```
make && make run
```

- **Změřte výkon kódu**

```
sbatch run.sl
```

- **Otázky**

- Které plánování poskytuje nejlepší výsledky
- Jaký je vhodný chunk size?

<https://www.openmp.org/wp-content/uploads/OpenMP-4.5-1115-CPP-web.pdf>

- **Doplňte správné pragmy**

- Přidejte správnou pragmu pro paralelizaci.
- Přidejte pragmu pro přerušení výpočtu, pokud je klíč nalezen.
- Ověřte, že kód počítá správně (všechny běhy poskytují stejné výsledky).

<https://www.openmp.org/wp-content/uploads/OpenMP-4.5-1115-CPP-web.pdf>

- **Zkompilujte a spustěte kód**

```
make && make run
```

- **Změřte výkon kódu**

```
sbatch run.sl
```

- **Otázky**

- Jakého zrychlení jste dosáhli?
- Proč je dosažená efektivita nízká?



**Pokračování příště**