OpenMP Tasky

AVS - Architektury výpočetních systémů
Cvičení 5, 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

Jdeme na to

- **Připojte se na Barboru**
  - 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 5. cvičení**

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

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

```
ml Advisor VTune intel
```

Step 1 – Hledání prvočísel

- **Paralelizace smyčky pomocí tasků (vyvažování zátěže)**
  - Doplňte pragmy pro generování tasků
  - Pozor na zápis do sílené proměnné
  - Zvolte vhodnou granularitu pro tasky
- **Ověřte správnost výpočtu**

```
int sieve()
{
 // 1. Use OpenMP tasks to distribute the work
 // 2. Find a suitable chunk size
 // 3. Be careful when updating nPrimes
  constexpr int chunkSize = 1;
  int nPrimes = 0;

  // Test all numbers
  for (int number = 1; number <= size; number += chunkSize)
  {
    nPrimes += getNumberOfPrimes(number, ((number + chunkSize) > size)
                                         ? size : number + chunkSize);
  }
  return nPrimes;
}
```

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

- **Zkompilujte a spusťte kód**

```
make && make run
```

Správný výstup

```
-------------------------------------------------
 Parallel Sieve of Eratosthenes
 Running on: r2i0n1.ib0.smc.salomon.it4i.cz
 Number of cores:   24
 Number of threads: 24
-------------------------------------------------
  - Testing pool:              1000
  - Number of repetitions:     1000
  - Number of primes found:     169
  - Time to compute primes:   0.076 ms
-------------------------------------------------
  - Testing pool:             10000
  - Number of repetitions:     1000
  - Number of primes found:    1230
  - Time to compute primes:   0.253 ms
-------------------------------------------------
  - Testing pool:            100000
  - Number of repetitions:      500
  - Number of primes found:    9593
  - Time to compute primes:   2.500 ms
-------------------------------------------------
  - Testing pool:          10000000
  - Number of repetitions:        3
  - Number of primes found:  664580
  - Time to compute primes: 241.000 ms
-------------------------------------------------
```

Step 1 – Hledání prvočísel – Škálování a efektivita

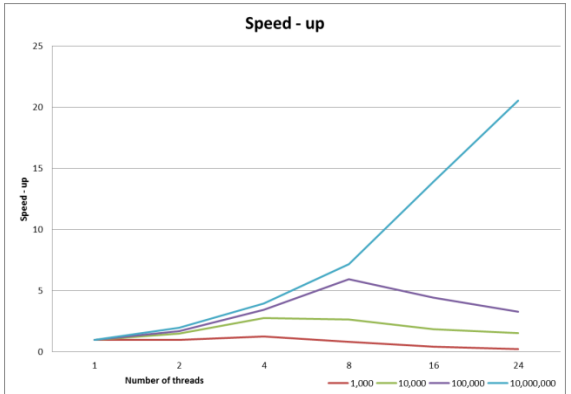
Vytvořte tabulky silného škálování

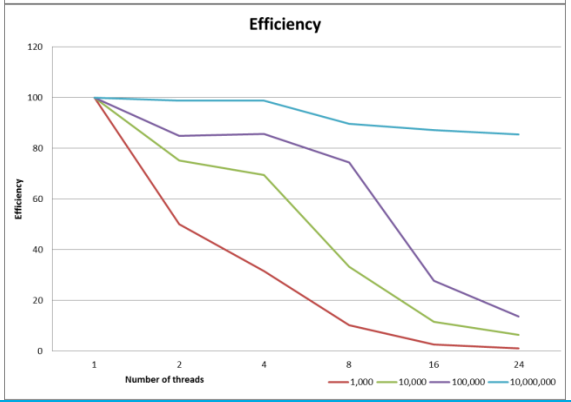
```
make bench
sbatch run.sl
```

1000 – 10M čísel

P	T	S	E
1			
2			
4			
8			
16			
24			
36			

Vytvořte grafy silného škálování pro zrychlení a efektivitu





Step 2 – Výpočet Fibonacciho čísla

Paralelizace rekurzivního algoritmu

- Doplňte pragmy pro generování tasků ve funkci `fibBenchmark`
- Doplňte pragmy do funkce `calculateFibNumber`
- Zvolte vhodnou granularitu pro tasky

```
/// Calculate Fibonacci number for a given number
size_t calculateFibNumber(size_t rank)
{
  // 2. Use OpenMP tasks parallelize recursive algorithm
  // 3. Find sensible seqTreshold
  size_t x, y;
  constexpr size_t seqTreshold = 16;

  if (rank < 2) return rank;

  x = calculateFibNumber(rank - 1);
  y = calculateFibNumber(rank - 2);

  return x + y;
}//---------------------------------------------------------------------

/// Fibonacci benchmark
template<size_t rank> size_t fibBenchmark()
{
  // 1. Use OpenMP tasks to distribute the work
  return calculateFibNumber(rank);
}//---------------------------------------------------------------------
```

Zkompilujte a spusťte kód

```
make && make run
```

Správný výstup

```
-------------------------------------------------
 Parallel Fibonacci number calculation.............
 Running on: r2i0n1.ib0.smc.salomon.it4i.cz
 Number of cores:   24
 Number of threads: 24
-------------------------------------------------
  - Testing pool:                    10.
  - Number of repetitions:         1000
  - Fib number (id  10):             55
  - Time to compute primes:       0.062 ms
-------------------------------------------------
  - Testing pool:                    20.
  - Number of repetitions:          100
  - Fib number (id  20):            6765
  - Time to compute primes:       0.300 ms
-------------------------------------------------
  - Testing pool:                    30.
  - Number of repetitions:           10
  - Fib number (id  30):         832040
  - Time to compute primes:      11.900 ms
-------------------------------------------------
  - Testing pool:                    40.
  - Number of repetitions:            1
  - Fib number (id  40):      102334155
  - Time to compute primes:    1367.000 ms
-------------------------------------------------
```

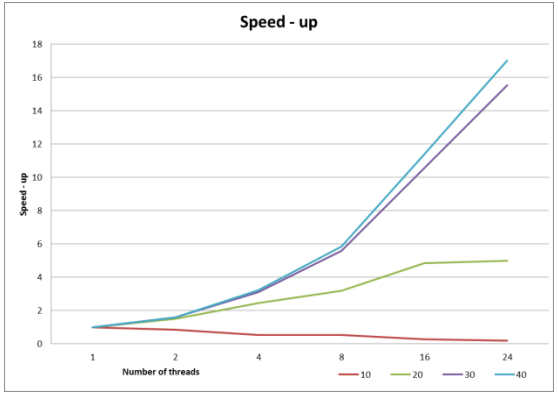
Vytvořte tabulky silného škálování

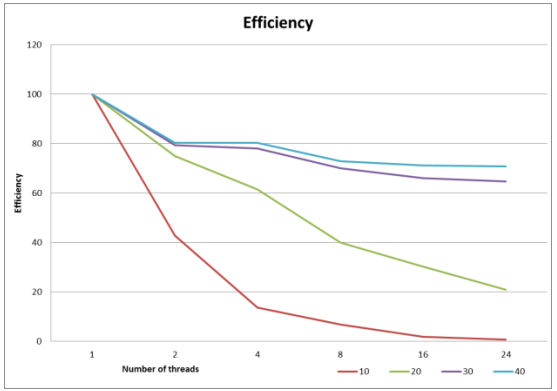
```
make bench
sbatch run.sl
```

Fibonacci rank 10, 20, 30, 40

P	T	S	E
1			
2			
4			
8			
16			
24			
36			

Vytvořte grafy silného škálování pro zrychlení a efektivitu





Step 3 – Paralelní vyhledávání v seřazeném poli

Paralelizace rekurzivního algoritmu

- Doplňte pragmy pro generování tasků funkce `search`
- Doplňte pragmy do funkce `binarySearch`
  - Využijte pragmu `taskgroup` pro vymezení oblasti, kde musí všechny tasky dokončit
  - Využijte pragmu `cancel` pro ukončení ostatních tasků, pokud již někdo našel
  - Pozor na souběhy při zápisech do proměnné `pos`
  - Pozor na správnou synchronizaci tasků
  - Najděte vhodnou granularitu pro tasky

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

Zkompilujte a spusťte kód

```
make && make run
```

Správný výstup

```
-------------------------------------------------
 Parallel Search in sorted array
 Running on: sc-gpu1
 Number of cores: 12
-------------------------------------------------
  - Testing pool:                 1000
  - Key:                           650
  - Number of repetitions:      100000
  - Position of the key:           650
  - Time to find the key:       16.070 us
-------------------------------------------------
  - Testing pool:              1000000
  - Key:                        758231
  - Number of repetitions:      100000
  - Position of the key:        758231
  - Time to find the key:       28.390 us
-------------------------------------------------
  - Testing pool:             10000000
  - Key:                       4212157
  - Number of repetitions:      100000
  - Position of the key:       4212157
  - Time to find the key:       29.860 us
-------------------------------------------------
  - Testing pool:            999999999
  - Key:                     351233241
  - Number of repetitions:       10000
  - Position of the key:     351233241
  - Time to find the key:       40.100 us
-------------------------------------------------
```

Pokračování příště