



**ČESKÉ
VYSOKÉ
UČENÍ
TECHNICKÉ
V PRAZE**

České Vysoké Učení Technické v Praze
Fakulta Elektrotechnická

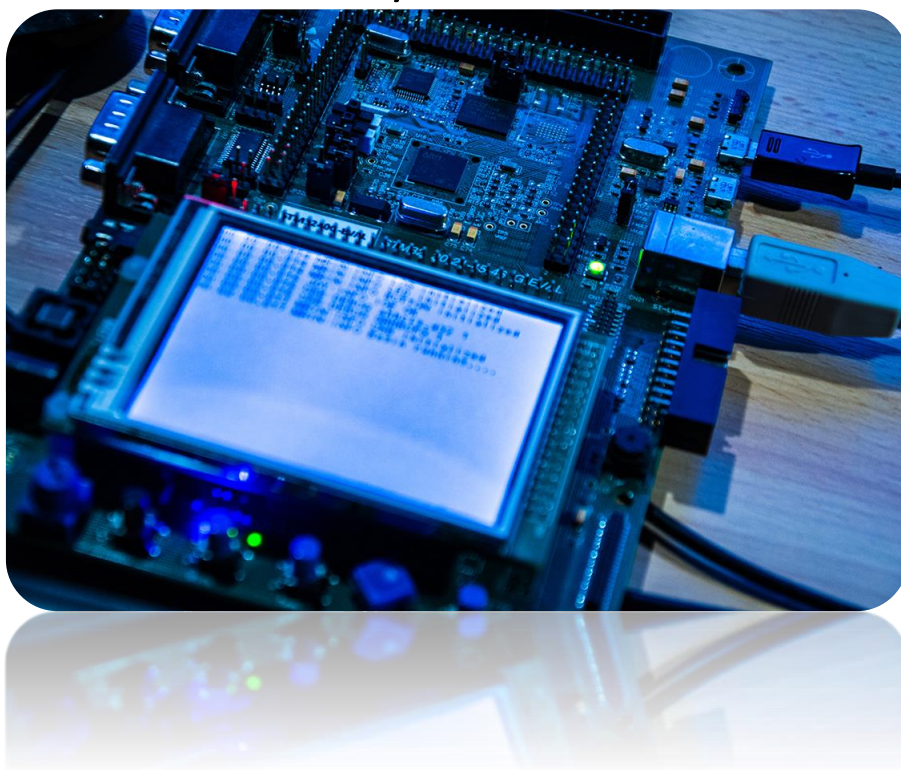
Počítačové inženýrství

PAP – Pokročilé architektury počítačů

Semestrální projekt 02

Závěrečná zpráva

Tvorba programu pro paralelní výpočetní systém



Zadání

Napište paralelní program pro Vámi vybraný problém. Vybraný problém slovně popište, uveďte program v jazyce C/C++ řešící tento problém, následně program paralelizujte. Výsledky Vašich experimentálních pozorování zpracujte, jak v kontextu k počtu vláken, tak v kontextu počtu procesorů.

Problém

Vybraný problém se nazývá enumeration sort. Je to algoritmus, který neseřazené vstupní pole seřadí od nejmenšího po největší prvek na základě počtu prvků, které jsou menší než daný prvek.

Pseudokod Enumeration sort

```
enumSort(_sorted[], _arr[], _size ){
    sum[_size];
    idx[_size][_size];
    //=== Indikuj vetsi-mensi
    for( i = 0; i < _size; i++){
        for( j = 0; j < _size; j++){
            if( _arr[i] < _arr[j] || (_arr[i] == _arr[j] && i<j)){
                idx[i][j] = 1;
            }
        }
    }

    //=== SUMARIZACE
    for(i = 0; i < _size; i++){
        for( j = 0; j < _size; j++){
            sum[i] += idx[i][j];
        }
    }
    ///=== Prohazeni prvku dle mensi X vetsi
    for( i = 0; i < _size; i++){
        _sorted[ sum[i] ] = _arr[i];
    }
}
```

Paralelizace

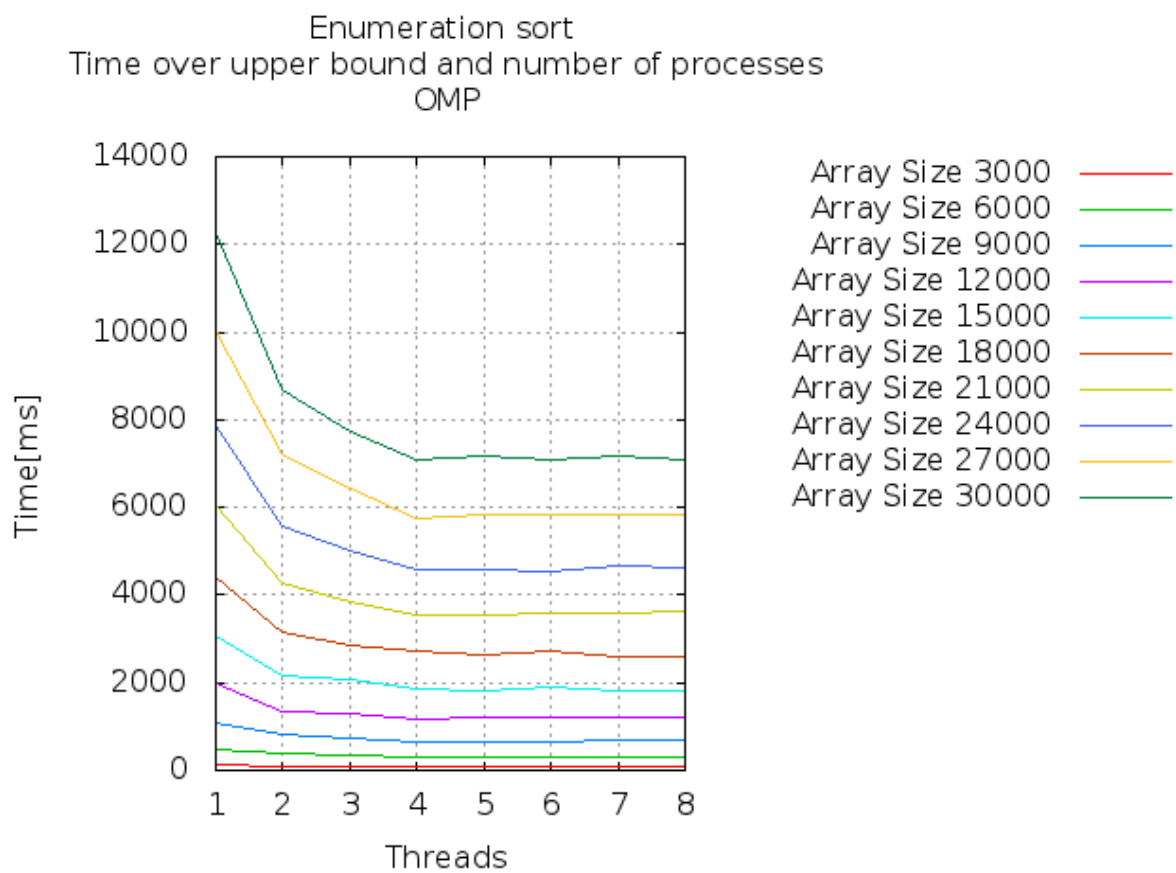
Paralelizaci daného problému jsem vyvíjel na Ubuntu 14.04 LTS 64bit, na virtuálním stroji se dvěma jádry a ke každému jádru byli přiřazeny dva thready a přiřazenými 4GB RAM paměti.

Fyzický stroj:

- Lenovo T530
- Intel i7 3630QM 2.40GHz
- 8GB RAM DDR3
- HDD 7200rpm

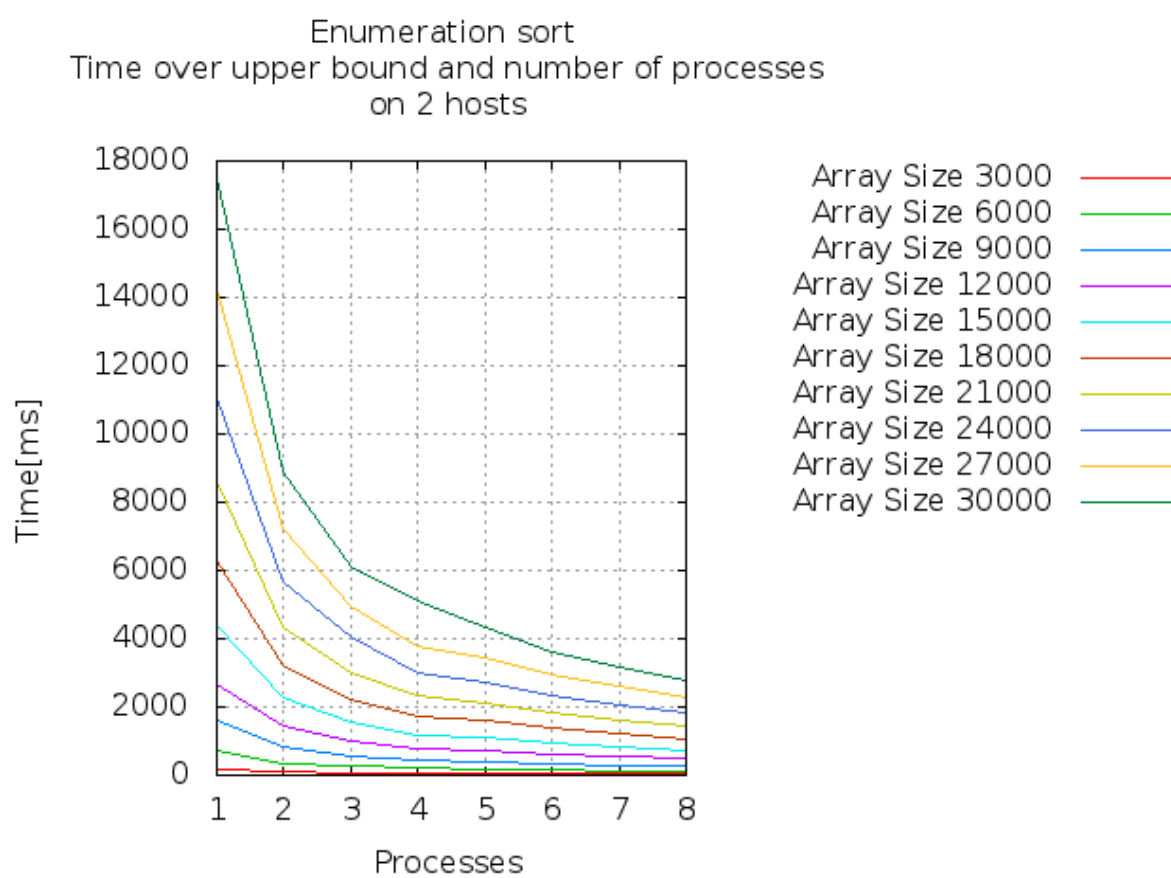
Výsledky však nebyly stabilní a pro jeden běh programu se stejnými vstupními parametry jsem získal velmi odlišné výsledky, takže konečné měření probíhalo na školních strojích v učebně K307 a to až na 4ech.

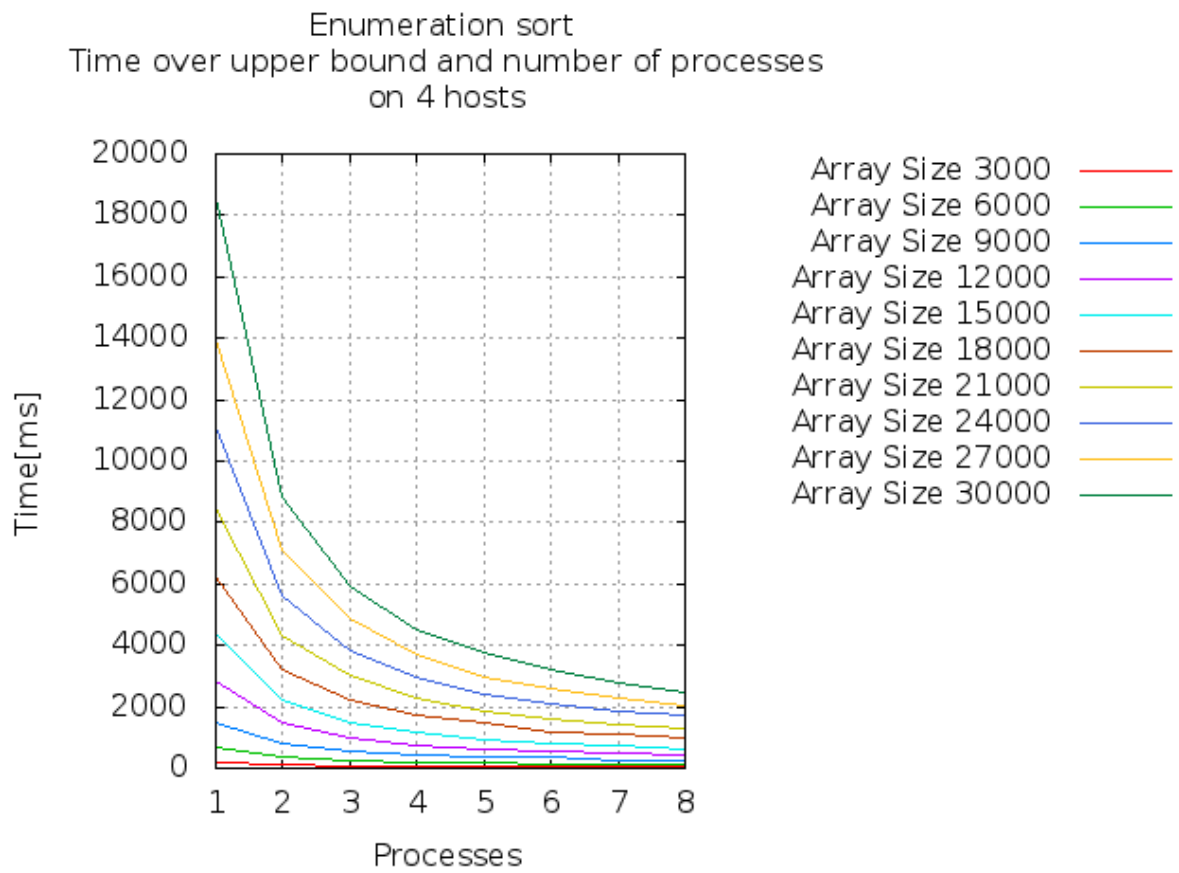
Parallelize v OMP



Paralelizace v MPI

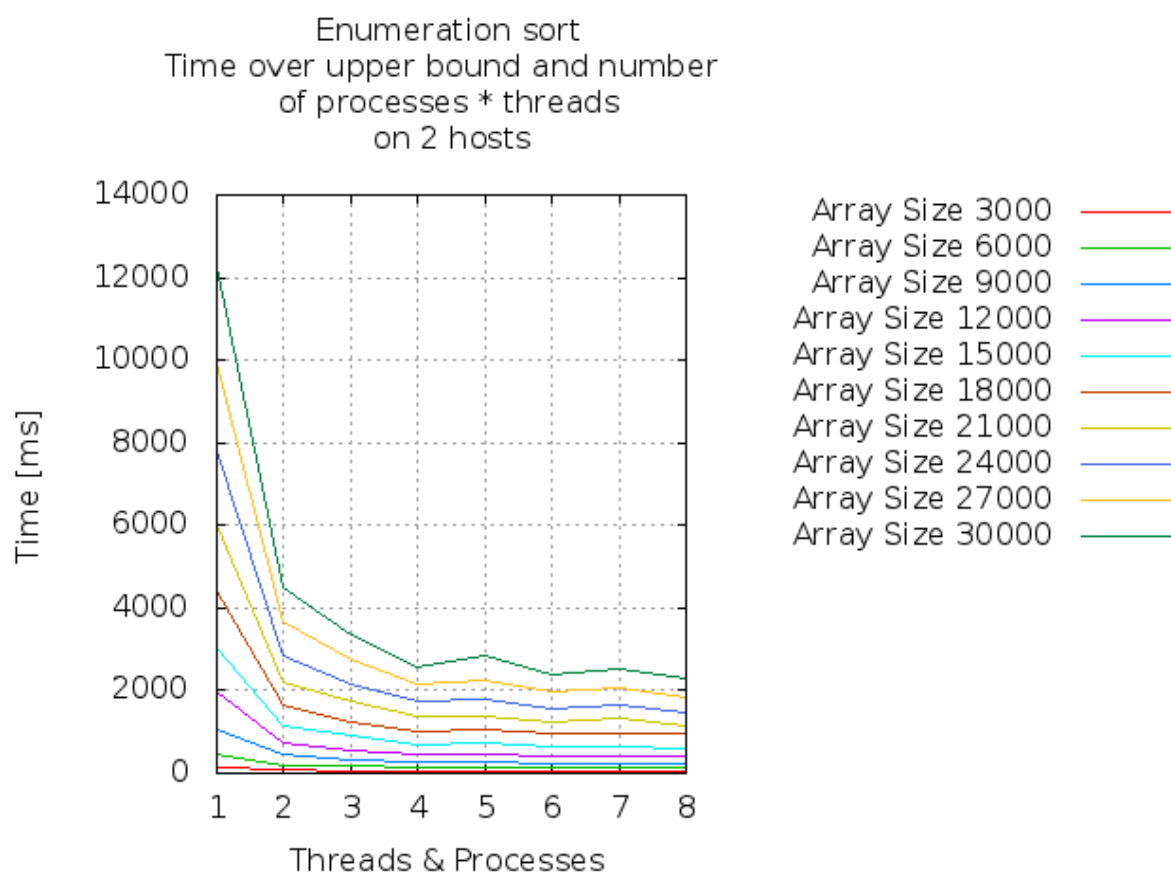
MPI na 2 hostech

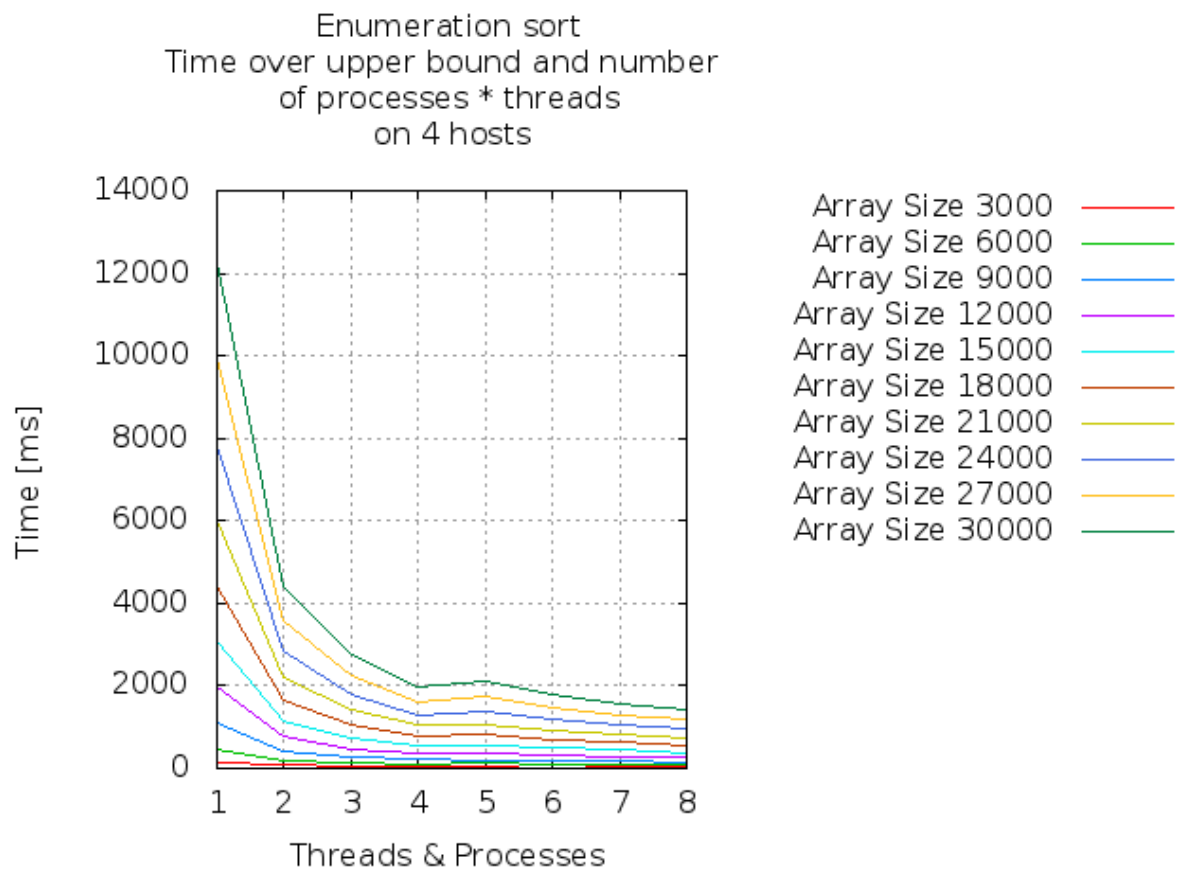


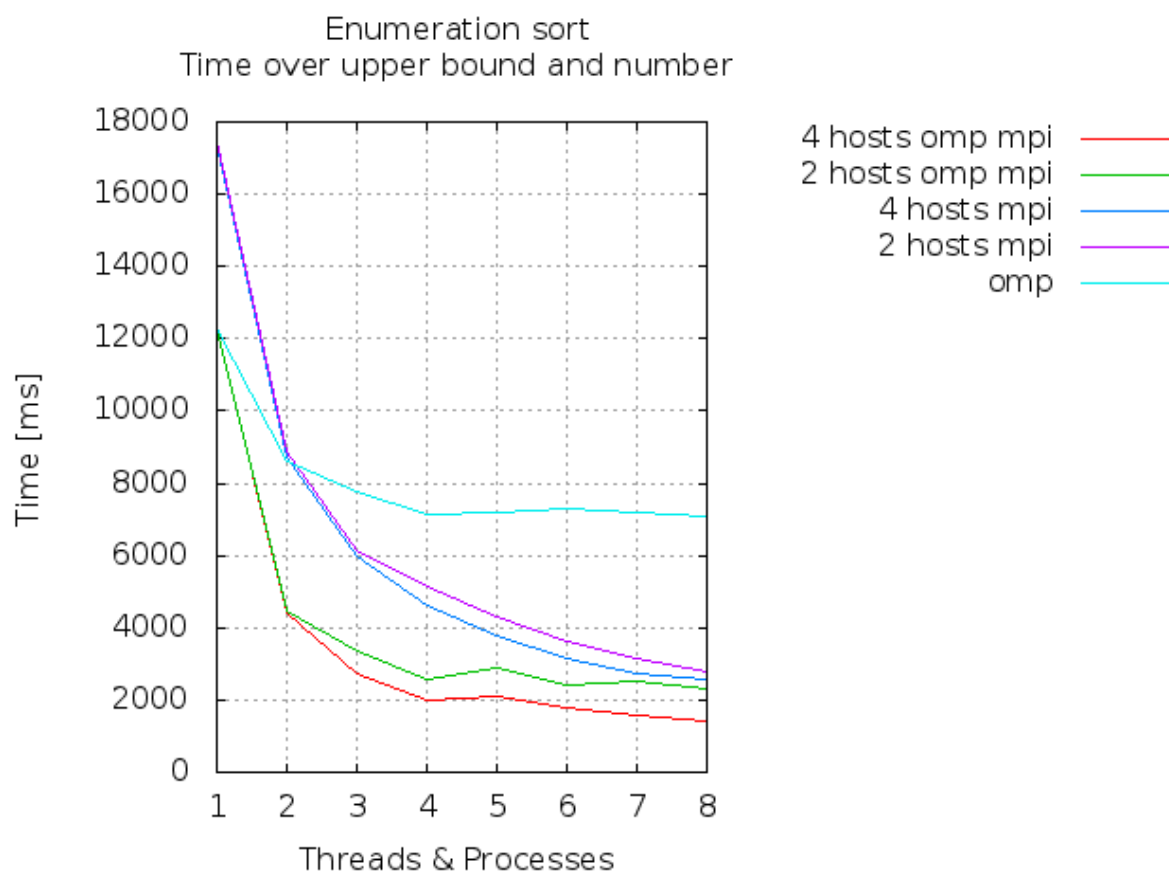


Paralelizace v OMP i MPI

OMP MPI on 2 hosts







Závěr

Při překročení fyzického počtu vláken a paralelizaci pomocí OMP bylo vidět, že se čas algoritmu zvýšil. Tento jev se při paralelizaci MPI neprojevoval, protože počet jader CPU a jejich vláken byl vyšší, než bylo pro testování potřeba. Pro lepší přehlednost by byly lepší grafy znázorňující zrychlení nikoliv čas, ale i tak mají jistou vypovídající hodnotu o zlepšení.

O paralelizaci problému pomocí OMP můžeme říct, že se běh programu zlepší, ale omezující podmínkou je počet vláken, které poskytuje HW aktuálního stroje, na kterém je SW spuštěn. U paralelizace pomocí MPI závisí i na ostatním strojích, kde je SW spuštěn a tím roste i možnost zvýšit počet procesů, na kterých SW běží.