MI-SPOL-12 (MI-PDP.16) (přednáška (1) a 2)

# Programový model nad sdílenou pamětí: OpenMP (paralelní regiony, paralelní vlákna, datový a funkční paralelismus, paměťový model, synchronizační nástroje)

[ paralelní systém se sdílenou pamětí (Symmetric multiprocessing - SMP), UMA
- komunikace pomocí R/W operací
paralelní systém s distribuovanou pamětí (NUMA)
- komunikace posíláním zpráv ]

OpenMP - knihovna pro programování nad sdílenou pamětí (nemusí distribuovanou)

- vybrané sekce určené pro paralelizaci → paralelní regiony
  - pomocí fork-join mechanismu jsou vždy vytvářena, prováděna a ukončena paralelní vlákna
  - mimo explicitní použití masken vlákno (může využívat pool pro nižší režie)
  - programátor je odpovědný za thread-safe program

Model volnější konzistence
- vlákna mohou držet některé hodnoty lokálně a nemusí hned zapisovat
- explicitní operace flush()

- paralelní region definovaný direktivou parallel
  - na konci implicitní bariéra

if (cond) - podmínka paralelizace
num_threads - počet vláken v par. regionu
vlastnosti (var) - vlastnosti proměnných
- shared - sdílená mezi vlákny
- private - lokální ve vlákně (undef)
- firstprivate - lokální ve vlákně s hodnotou, kterou měla v hlavním vlákně
- lastprivate - (cyklus) - hodnota ze staticky poslední iterace ven z par. regionu
- default - určuje defaultní vlastnost všech proměnných
- reduction - reduction(operator: variable)
  - skalární proměnné a nepřetížený operátor
  - lineární a logaritmická

threadprivate - globální platnost v rámci všech paralelních regionů

## Datový paralelizmus

- direktiva for → na konci cyklu je implicitní bariéra
- direktiva schedule:
  - static – rozdělí rovnoměrně mezi vlákna, s chunk-size rozdělí po kus velkých kusech (nejmenší řetěz, ideální pro stejně velké)
    - rozdělování bloky jsou po sobě jdoucí (stejné u všech)
  - dynamic – dynamické rozdělování o velikosti chunk-size nebo 1 (volitelná iterace)
  - guided – dynamicky přidělují max(⌈#dosud nepřidělených/p⌉, chunk-size) (rostoucí složitost)
  - auto – záleží na kompilátoru a OS

\+ collapse(), nowait

## Funkční paralelizmus

- direktiva task pro vytváření úloh – vhodné pro D&C algoritmy
- vlákna jsou producenti i konzumenti úloh

taskwait – čeká na dokončení úloh z daného par. regionu
task-if – podmínka vytvoření úlohy
single / master – kód je proveden pouze jedním nebo masterem vláknem

## Synchronizační nástroje

bariér – všechna sem musí dorazit
master, single
critical – vytvoření kritické sekce provádění pouze jedním vláknem (může být pojmenovaná)
atomic – atomická paměťová operace

read, write
update – read, write and modify
capture – rozšiřuje update o možnost získání dané proměnné (před nebo po)

{ my_ptr = ptr; ptr + BLOCK_SIZE }

cancel – ukončení paralelního regionu a skok na vyčleněnou bariéru
– ostatní pozdější ho sami