

Jak vypracovat semestrální práci včas

- Je nanejvýš vhodné na ní začít pracovat včas, už po prvním semináři
- Jako následující příklad se uvažuje nalezení neznámých koeficientů k a l , rovnice $k \cdot y(t) + l \cdot y(t) = x(t)$.
 - Jsou zadány funkce $y(t)$ a $x(t)$ pomocí naměřených hodnot a interval t

1. fáze – pochopení zadání

- Vytvoří se referenční řešení pouze pro jedno vlákno
 - Čímž se získají body do dokumentace k měření urychlení
- Např. pomocí genetického algoritmu rulety
 - http://en.wikipedia.org/wiki/Fitness_proportionate_selection
 - <http://www.edc.ncl.ac.uk/highlight/rhjanuary2007g02.php/>
 - <http://www.kiv.zcu.cz/studies/predmety/uir/predn/P3/FThema3.pdf>
- fitness function je $ff = k \cdot y(t) + l \cdot y(t) - x(t)$
- Náročnost práce cca 2 dny
 - Naměřeno na studentovi, který předtím uvedený genetický algoritmus ani neznal

2. fáze – sdílená paměť

- Vezme se kód z 1. fáze a nechá se běžet paralelně v několika vláknech
 - Dokud neuplyne stanovený čas
 - Nebo se vyzkouší stanovený počet kombinací
 - Nebo hodnota ff klesne pod stanovené minimum
- Hlavní vlákno potom sesbírá mezivýsledky a vybere ten, pro který dala fitness function nejmenší hodnotu
- Náročnost práce cca 1 den
 - Prakticky jde jen o vytvoření vlákna, jeho pozastavení a ukončení

3. fáze – distribuovaná paměť

- Ten samý postup jako u 2. fáze, pouze se kód z 1. fáze nespustí ve vláknu, ale v procesu MPI
- Takže mezivýsledky sesbírá a vyhodnotí proces s ID==0
- Náročnost práce cca 2 dny
 - 1 den na seznámení s MPI a 1 den 2. fáze

4. fáze - dokumentace

- Náročnost práce – cca 4 dny
 - 1. úvodní stránka+obsah + 3 stránky/den
- Viz vzorové dokumentace na portále, kdo chce plný počet bodů

Volba jazyka

- Použijte jeden jazyk, např. pro MPI a PVM se píše v C
- Proto je lepší psát vše rovnou v C, abyste výpočet psali jenom jednou
 - A také abyste věděli, že výpočet funguje a že případná chyba je zřejmě někde v paralelizaci

Možná úskalí

- Zatím je celková náročnost práce cca 9 dnů
 - Odhad zohledňuje to, že student na tom nebude všech 9 dnů dřít od slunka do slunka, ale jenom několik hodin denně
- Je možné, že se v průběhu práce objeví problém, jehož řešení si vyžádá více času
 - Např. chyba synchronizace v paralelním programu
 - Nebo pointer v C, na který vás Java nepřipravila
- Nebo bude snaha o maximální počet bodů, případně navrhnout lepší řešení než uvedený genetický algoritmus
- Proto je lepší začít na práci dělat hned po prvním semináři
- Největším rizikem je, když
 - v zadání hledáte složitosti, které tam nejsou
 - čekáte, až úlohu vyřeší někdo jiný a poradí vám
 - neptáte se, když něco nevíte a nemůžete na to přijít
 - odkládáte vypracování s tím, že to stihnete, protože to tady vyšlo na cca 9 dní
 - řešíte to o zkouškovém, kdy se máte učit na zkoušky

Co když je sériový algoritmus rychlejší?

- Mohlo dojít k využití paralelizace v procesoru na úrovni instrukcí – viz přednáška o urychlení vlákna
- Urychlení některých úloh je závislé na počtu procesorů a objemu dat – viz uspořádané proměnné
- Data mohla být příliš malá
 - V takovém případě se dají vytvořit umělá data, která budou dostatečně velká a na nich zjistit, od jakého objemu dat dochází k urychlení
- Takže i takový výsledek může být správné řešení
 - PPR není jen o tom naučit se psát paralelní programy, ale i o pochopení, že ne vždy je paralelizace pomocí vláken vhodné řešení