Posudek oponenta závěrečné práce

České vysoké učení technické v Praze **Fakulta informačních technologií**

Student: Rudolf Talácko
Oponent práce: Ing. Daniel Langr, Ph.D.
Název práce: Pokročilé metody řazení ve vícevláknovém prostředí
Obor: Teoretická informatika

Datum vytvoření: 8. 6. 2017

Hodnotící kritérium:	*Způsob hodnocení - následující škálou 1 až 5:*
1. Náročnost a další komentář k zadání	*1=mimořádně náročné zadání,* ***2=náročnější zadání,*** *3=průměrně náročné zadání,* *4=lehčí, ale ještě dostatečně náročné zadání,* *5=nedostatečně náročné zadání*

Popis kritéria:
Podrobněji charakterizujte diplomovou (bakalářskou) práci a její případné návaznosti na předchozí nebo běžící projekty. Dále posuďte, čím je zadání této ZP náročné. (U obtížnější ZP lze dále tolerovat některé nedostatky, které by u ZP standardní obtížnosti tolerovány nebyly; a naopak u jednoduché ZP mohou být zjištěné nedostatky hodnoceny přísněji.)

Komentář:
Práce se zabývá návrhem hybridního paralelního algoritmu pro řazení dat a jeho porovnáním se stávajícími konkurenčními algoritmy. Problematika paralelního řazení je obecně velmi komplikovaná a přidává velkou míru algoritmické a implementační složitosti oproti sekvenčnímu řazení.

Hodnotící kritérium:	*Způsob hodnocení - následující škálou 1 až 4:*
2. Splnění zadání	***1=zadání splněno,*** *2=zadání splněno s menšími výhradami,* *3=zadání splněno s většími výhradami,* *4=zadání nesplněno*

Popis kritéria:
Posuďte, zda předložená ZP splňuje zadání. V komentáři uveďte body zadání, které nebyly zcela splněny, případně rozšíření ZP oproti původnímu zadání. Nebylo-li zadání zcela splněno, pokuste se posoudit závažnost, dopady a případně i příčiny jednotlivých nedostatků.

Komentář:
Zadání bylo splněno bez výhrad

Hodnotící kritérium:	*Způsob hodnocení - následující škálou 1 až 4:*
3. Rozsah písemné zprávy	***1=splňuje požadavky,*** *2=splňuje požadavky s menšími výhradami,* *3=splňuje požadavky s většími výhradami,* *4=nesplňuje požadavky*

Popis kritéria:
Zhodnoťte přiměřenost rozsahu předložené ZP vzhledem k obsahu, tj. zda všechny části ZP jsou informačně bohaté a ZP neobsahuje zbytečné části.

Komentář:
Rozsah práce je postačující. Vlastní text má 61 stran, z toho 23 stran se věnuje představení stávajících řadících algoritmů a optimalizačních technik, které s prací souvisejí. Zbytek je již věnován vlastní práci, tj. především popisu dosažených výsledků.

Hodnotící kritérium:	*Způsob hodnocení - bodové hodnocení 0 až 100 bodů (známka A až F):*
4. Věcná a logická úroveň práce	*85 (B)*

Popis kritéria:
Posuďte, zda předložená ZP je po věcné stránce v pořádku, případně vyskytují-li se v práci věcné chyby nebo nepřesnosti. Zhodnoťte dále logickou strukturu ZP, návaznosti jednotlivých kapitol a pochopitelnost textu pro čtenáře.

Komentář:
Práce je po věcné stránce v pořádku, poukázal bych jen na pár nedostatků:
— Tabulka 2.1: Přídavná paměť QuickSortu je sice opravdu O(n), ale pomocí jednoduchého triku (který používá každá efektivní implementace), konkrétně eliminace koncové rekurze, lze tyto nároky snížit na O(n log n) [Sedgewick 1978]. Tento trik dělá QuickSort prakticky in-place řadícím algoritmem (oproti např. MergeSortu), což by bylo vhodné při porovnání algoritmů uvést.
— Sekce 3.3.2.1.: "Pokud chceme využívat možnosti OpenMP, je nutné do progarmu přidat direktivu #include <omp.h>." To není pravda, OpenMP lze používat primárně pomocí direktiv #pragma omp, které uvedený hlavičkový soubor nepotřebují. Ten je potřeba pouze pro volání knihovních funkcí OpenMP (např. zjištění čísla aktuálního vlákna).
— Str. 21: "Mezi nejběžnější metody..." (myšleno pro nastavení počtu vláken). Troufám si tvrdit, že v praxi se nejčastěji používá proměnná prostředí OMP_NUM_THREADS, která umožňuje počet vláken nastavit nezávisle na zdrojovém kódu až před spuštěním programu.
— Str. 22: "...nový TASK je vložen do threadpool...". Threadpool je "zásobník" vláken. Task je spíše vložen do "taskpool", neboli "zásobník" OpenMP úloh.
— Str. 23: "Obvykle se setkáváme se třístupňovou cache pamětí" Otázkou je, co znamená "obvykle", autor pravděpodobně myslel obvykle u architektury x86_64. U jiných architektur to až tak obvyklé není (výpočetní koprocesory, mobilní telefony, procesory IBM Power, atd.).
— Str. 29: "...u náhodné posloupnosti musíme udělat výjimku" Nemusíme, softwarové pseudo-náhodné generátory generují stejnou posloupnost čísel, pokud je inicializujeme stejnou "seed" hodnotou. Toto je velmi snadný způsob, jak porovnat algoritmy pro "náhodná", ale zároveň stejná data.
— Str. 31: Pro indexy je ve zdrojových kódech napevno použit datový typ "int". Toto je poměrně nešťastné řešení, protože to limituje velikost řazených dat na cca. 2G. Především u serverů a HPC systémů, kde se více-vláknové řazení využije nejvíce, množství paměti běžně umožňuje pracovat s mnohem většími poli.
— Str. 58: "...pomocí kompilátoru GNU libstdc++": libstdc++ není kompilátor, ale GNU implementace standardní knihovny C++.

Hodnotící kritérium:	*Způsob hodnocení - bodové hodnocení 0 až 100 bodů (známka A až F):*
5. Formální úroveň práce	*90 (A)*

Popis kritéria:
Posuďte správnost používání formálních zápisů obsažených v práci. Posuďte typografickou a jazykovou stránku ZP, viz Směrnice děkana č. 14/2015, článek 3.

Komentář:
Pouze drobné připomínky:
— Např. Algorithm 1 a další: Pro logické AND je nevhodně použit symbol pro průnik množin (\cap). V pseudokódech algoritmů bych jednoznačně doporučil psát slovy "AND", což přispívá k čitelnosti a zabraňuje nejednoznačnostem. (Mimochodem, v českém textu by se spíše hodilo "Algoritmus" než "Algorithm"; předefinování je snadné.)
— Pro přiřazení je v pseudokódech někdy použita šipka a někdy znak "=" (viz. např. strana 9). Šipka je podle mého názoru lepší (také se používá např. ":=" apod.), rovnítko implikuje spíše rovnost hodnot.
— Nadužívání neproporcionálního fontu: Tento font se většinou používá k sazbě částí zdrojových kódů a k sazbě příkazů operačního systému, názvů souborů, atd. Např. není důvod tímto fontem sázet "C++" či "Galloping mode".
— Str. 21: Lepší psát "fork-join" než "join-fork", je to i logické (nejprve se jedno vlákno rozdělí na více vláken, která se pak opět sloučí).

Jinak se mi formální úroveň velmi zamlouvala, text je doplněn vhodnými ilustracemi popisovaných principů a grafy s výsledky jsou přehledné a vypovídající.

Hodnotící kritérium:	*Způsob hodnocení - bodové hodnocení 0 až 100 bodů (známka A až F):*
6. Práce se zdroji	*100 (A)*

Popis kritéria:
Vyjádřete se k aktivitě studenta při získávání a využívání studijních materiálů k řešení ZP. Charakterizujte výběr studijních pramenů. Posuďte, zda student využil všechny relevantní zdroje nebo zda se pokoušel řešit již vyřešené problémy. Ověřte, zda jsou všechny převzaté prvky řádně odlišeny od vlastních výsledků a úvah, zda nedošlo k porušení citační etiky a zda jsou bibliografické citace úplné a v souladu s citačními zvyklostmi a normami.

Komentář:
Nenašel jsem žádné nedostatky

Hodnotící kritérium:	*Způsob hodnocení - bodové hodnocení 0 až 100 bodů (známka A až F):*
7. Hodnocení výsledků, publikační výstupy a ocenění	*95 (A)*

Popis kritéria:
Vyjádřete se k úrovni dosažených hlavních výsledků ZP, např. k úrovni teoretických výsledků, nebo k úrovni a funkčnosti technického nebo programového vytvořeného řešení, apod. Případně také zhodnoťte, zda software nebo zdrojové texty, které nevytvořil sám student, byly v ZP použity v souladu s licenčními podmínkami a autorským právem. Popište případnou publikační činnost a získaná ocenění související s řešením této ZP.

Komentář:
Výsledky práce jsou velmi zajímavé a přínosné. Částečně vypovídají o kvalitě navrženého algoritmu. Pro jeho obecné ohodnocení bude ale potřeba v budoucnu provést více experimentů s různými typy dat a na různých hardwarových architekturách. Především by mě zajímal výkon pro data, která mají nekonstantní čas funkce porovnání dvou prvků (např. řazení řetězců). V práci byla použita data s různou velikostí, ale jako klíč bylo použito jedno celé číslo (viz str. 55). Toto není moc přínosný přístup, protože v praxi by při efektivním řazení nikdo nepřesouval velké datové struktury, ale pouze ukazatele na ně (jak je zmíněno dále v textu).

Hodnotící kritérium:	*Způsob hodnocení - nehodnotí se*
8. Komentář o využitelnosti výsledků	

Popis kritéria:
Uveďte, zda hlavní výsledky ZP rozšiřují již publikované známé výsledky a/nebo přinášející zcela nové poznatky. Uveďte možnosti využití výsledků ZP v praxi.

Komentář:

Využitelnost navrženého řadícího algoritmu nelze snadno posoudit. K tomu je potřeba především více experimentů provedených na různých architekturách a různých datech a především implementace ve formě obecně použitelné knihovny (pokud možno open-source). Zdrojové kódy byly vytvořeny za účelem měření prezentovaných výsledků a takovéto obecné využití nyní neumožňují. Každopádně má ale práce nakročeno "dobrých směrem".

Hodnotící kritérium:	*Způsob hodnocení - nehodnotí se*
9. Otázky k obhajobě	

Popis kritéria:
Uveďte případné dotazy, které by měl student zodpovědět při obhajobě ZP před komisí (body oddělte odrážkami).

Otázky:

— Str. 48: "Zde se objevil problém častých přístupů všech paralelních vláken do tabulky A_Parts,..." Proč je toto problém? Jestli jsem text správně pochopil, tak v této části se tabulka jednotlivými vlákny pouze čte, čili by tam nemělo docházet k výpadkům cache pamětí (false sharing).

— V textu je často použit pojem "transformace zdrojového kódu". Tento pojem mi nepřipadá úplně standardní, používá se poměrně málo a obecně v různých souvislostech (viz Google). V textu jsou sice tyto transformace uvedeny (sekce 3.2.1), ale pojem sám o sobě není definován. Prosím o vysvětlení, co si autor obecně pod tímto pojmem představuje. (Např. očividně se nejedná o transformaci kódu z jednoho jazyka do jiného.)

— Implementace používá pro generování náhodných čísel funkci rand() ze standardní C knihovny. Tato funkce je velmi zastaralá, obsahuje nekvalitní generátor a už delší dobu se jí vůbec nedoporučuje používat. Jaké jsou dnešní "moderní" možnosti pro generování náhodných čísel v jazyce C++?

Hodnotící kritérium:	*Způsob hodnocení - bodové hodnocení 0 až 100 bodů (známka A až F):*
10. Celkové hodnocení	*90 (A)*

Popis kritéria:
Shrňte stránky ZP studenta, které nejvíce ovlivnily Vaše celkové hodnocení. Celkové hodnocení **nemusí** být aritmetickým průměrem či jinou hodnotou vypočtenou z hodnocení v předchozích jednotlivých kritériích 1 až 9.

Text hodnocení:

Celkové se mi práce velmi líbila, především díky silné experimentální sekci. Téma práce bylo spíše náročnější a autor se s ním vypořádal velmi dobře.

Podpis oponenta práce: