FAKULTA INFORMAČNÍCH TECHNOLOGIÍ ČVUT V PRAZE

Posudek oponenta závěrečné práce

Student:	Dominik Šíba
Oponent práce:	Ing. Daniel Langr, Ph.D.
Název práce:	Pokročilé metody paralelního řazení dat
Obor:	Teoretická informatika

Datum vytvoření: 30. 5. 2018

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 optimalizací a především paralelizací algoritmů pro řazení dat, což je obecně netriviální úloha. Konkrétně se práce zaměřuje na algoritmy mergesort a radixsort. Autor navržené algoritmy implementoval a porovnal jejich výkonnost s existujícími implementacemi.

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.

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 písemné zprávy odpovídá zadání a požadavkům a neobsahuje žádné zbytečné části.

Hodnotící kritérium:	*Způsob hodnocení - bodové hodnocení 0 až 100 bodů (známka A až F):*
4. Věcná a logická úroveň práce	*90 (A)*

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:
Celkově je práce po věcné a logické stránce v pořádku. Obsahuje všechny potřebné části, tj. popis problematiky, analýzu řešení, popis implementace a provedených experimentů, a závěrečné zhodnocení. Zde uvádím několik poznámek/postřehů spíše minoritní důležitosti:
- Definice inplace (ip) a out-of-place (oop) nemusejí být tak jednoznačné. Autor uvádí, že oop algoritmy potřebují "pomocné pole" velikosti O(n). Např. quicksort žádné takové pole nepotřebuje, přesto ho autor zařazuje mezi oop algoritmy. Na druhou stranu i quicksort potřebuje pomocnou paměť v podobě místa na zásobníku pro rekurzivní volání. Tato paměť ale není O(n), alespoň při efektivní implementaci, a quicksort tudíž bývá zařazován mezi ip algoritmy (potřebná paměť je asymptoticky nižší než paměť oop algoritmů jako je mergesort).
- "V dnešní době se rychlost jednotlivých jader příliš nezvyšuje.": Je samozřejmě otázkou, co zde autor myslí rychlostí. Výkon měřený např. ve FLOPS se často zvyšuje např. díky pokročilejším SIMD jednotkám, příkladem budiž instrukční sada AVX512. Ohledně frekvence jader je to tak, že se v posledních letech spíše snižuje. Zatímco před pár lety měly procesory přes 4 i 5 GHz, dnes multi-core architektury s hodně jádry mají mezi 2 a 3 GHz, many-core architektury běžně kolem 1 GHz. Každopádně s poznámkou o potřebě paralelizmu nelze než souhlasit.
- OpenMP ve skutečnosti není knihovna. Jedná se o specifikaci nestandardních rozšíření jazyka C, C++ a Fortran o konstrukty podporující vícevláknové programování. Součástí této specifikace je i popis knihovního API (omp.h). Existuje více implementací OpenMP většinou vytvořených přímo výrobci překladačů (GNU, Intel, clang, IBM, PGI, ...).
- Str. 11.: "K využití tohoto standardu je potřeba programu přidat direktivu #include <omp.h>...". To není správně, OpenMP lze využívat bez načítání tohoto hlavičkového souboru např. jen přidáním direktiv #pragma omp v jazycích C a C++. Hlavičkový soubor je potřeba pouze pokud se volají knihovní funkce OpenMP, což pro paralelizaci programu obecně není potřeba.
- Str. 13: "Direktiva #pragma omp task slouží k vytvoření nové úlohy, kterou začne vykonávat nové vlákno." To rovněž není správně. OpenMP úlohy (task) jsou vytvářeny v rámci paralelního regionu, kde vlákna již existují. Nové úlohy jsou pouze mezi běžící vlákna distribuována plánovačem OpenMP.
- Osobně bych doporučoval se vyhnout použití funkce rand(), o které je známo, že často implementuje nekvalitní generátor náhodných čísel z malého rozsahu 0,...,32767. Při generování velkých souborů čísel se tudíž čísla mnohonásobně opakují, což může mít poměrně velký vliv na chování řadících algoritmů. C++11 zavádí přímo do Standardní knihovny vysoce kvalitní generátory náhodných čísel v rámci hlavičkového souboru <random>.

Hodnotící kritérium:	*Způsob hodnocení - bodové hodnocení 0 až 100 bodů (známka A až F):*
5. Formální úroveň práce	*95 (A)*

Popis kritéria:
Posuďte správnost používání formálních zápisů obsažených v práci. Posuďte typografickou a jazykovou stránku ZP, viz Směrnice děkana č. 26/2017, článek 3.

Komentář:
Práce je po formální stránce v pořádku, pouze autor z nějakého důvodu odstranil odsazování (indentation) prvních řádků v odstavcích, což působí velmi nevzhledně a ztěžuje to čitelnost. Každopádně to není standardní (toto odsazování má nějaký důvod a používá se prakticky v každém textu, alespoň pokud se mezi odstavci nedělá umělá mezera).
Jinak bych ještě velmi doporučil exportovat grafy (pravděpodobně z Excelu?) ve vektorovém formátu, např. v PDF, který lze snadno vkládat od (pdf)Latexu. Bitmapový formát zhoršuje čitelnost na obrazovce, speciálně s použitím nízkého rozlišení obrázků (ppi).

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:
Při práci se zdroji jsem neobjevil žádný problém.

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:
V rámci práce bylo provedeno mnoho experimentů, které umožňují velmi dobře ohodnotit navržené optimalizace a paralelizace algoritmů mergesort a radixsort. Velmi se mi líbí porovnání s paralelním multi-way mergesortem z libstdc++ Parallel mode, který je léty prověřený a obecně vykazuje velice dobrou škálovatelnost.

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:
Provedené experimenty umožňují čtenáři získat povědomí o výkonu a škálovatelnosti paralelních verzí algoritmů mergesort a radixsort. Autor se zaměřuje i na specifické případy vstupního uspořádání dat, kde mi ale trochu chybí dva případy, které se v praxi mohou vyskytovat (na rozdíl od vzestupně/sestupně seřazených posloupností, které se na vstupu řadících algoritmů vyskytují jen zřídka). Jedná se konkrétně o data s částečně seřazenými částmi, a poté data, která obsahují jen malý počet různých prvků.

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:
1) Autor implementuje funkci low_bound, která defakto provádí binární vyhledávání. Čím se tato funkce liší od standardní knihovní funkce C++ std::lower_bound?
2) Z jakého důvodu autor zařadil quicksort mezi out-of-place řadící algoritmy?

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:
Práce se mi celkově líbila, splnila zadání a veškeré moje připomínky jsou spíše drobnějšího významu.

Podpis oponenta práce: