

Posudek oponenta závěrečné práce

České vysoké učení technické v Praze

Fakulta informačních technologií

Student: Michal Čermák
Oponent práce: Ing. Daniel Langr, Ph.D.
Název práce: Paralelní metody stabilního řazení
Obor: Teoretická informatika

Datum vytvoření: 23. 1. 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 out-of-place and in-place variantami algoritmu MergeSort pro obecné řazení. Základní out-of-place MergeSort je poměrně triviální, totéž se ale nedá říct o jeho optimalizovaných variantách a efektivních implementacích, které jsou předmětem zkoumání práce. Ta se dále zabývá in-place variantami MergeSortu, které obecně šetří dodatečnou paměť algoritmu za cenu vyšší časové náročnosti řazení. V neposlední řadě se práce zaměřuje na dnes všudypřítomné vícevláknové prostředí, které do problematiky řazení přidává další úroveň obtížnosti.	
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: Ke splnění zadání mám následující výhradu. Bod 4. zadání požaduje cituji "Navrhněte paralelní hybridní algoritmus využívající kombinaci in-place, out-place MergeSortu a InsertionSortu". Kapitola 5, která navržený hybridní algoritmus představuje, ale o InsertionSortu nehovoří, tj. vypadá to, že navržený algoritmus pouze kombinuje out-of-place a in-place MergeSort. Přitom Sekce 3.2.1.3 a Tabulka 3.4 dokládají, že zakončení rekurze InsertionSortem má pro MergeSort přínos.	
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: Práce obsahuje všechny nezbytné části a ty jsou informačně velmi bohaté.	
Hodnotící kritérium:	Způsob hodnocení - bodové hodnocení 0 až 100 bodů (známka A až F):
4. Věcná a logická úroveň práce	95 (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:

Práce je po věcné stránce takřka zcela v pořádku, nemám žádné vážné výhrady. Text na sebe logicky navazuje, prezentovaná problematika je dobře a úplně vysvětlena a nenašel jsem žádné věcné ani logické chyby, pouze drobné nepřesnosti:

- 1.2.1.4 "...pro algoritmy, používající pouze porovnání a prohození prvku, které jsou zaměřením této práce...". Neřekl bych, že MergeSort používá pouze operace porovnání a prohození prvku. To je případ např. QuickSortu nebo InsertionSortu. MergeSort ale potřebuje navíc přesouvat prvky mezi poli (polem a bufferem, apod.), což je další operace, kterou je potřeba s prvky provádět.

- 3.2.1 "Základní algoritmus... ..Časy dosažené tímto algoritmem..." Pokud jsem text správně pochopil, jedná se o základní MergeSort algoritmus implementovaný autorem. To z textu není přímo patrné (zda se jedná o autorovu nebo nějakou cizí referenční implementaci). Pokud jde o měřené časy, mluvil bych o implementaci nikoli o algoritmu samotném (např. algoritmus může mít různé implementace, třeba v různých programovacích jazycích, a jejich časy mohou být v praxi velmi rozdílné).

- Kapitola 4: U výsledků, které prezentují časy při měření paralelních implementací algoritmů, bych doporučoval uvádět spíše zrychlení vůči referenčnímu sekvenčnímu řešení. Takto se pro přínos paralelizace musí zároveň čtenář dívat zpět na tabulku sekvenčních časů a případně si počítat jejich poměry.

Hodnotící kritérium:

Způsob hodnocení - bodové hodnocení 0 až 100 bodů (známka A až F):

5. Formální úroveň práce

95 (A)

Popis kritéria:

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

Komentář:

Formální úroveň je na velmi vysoké úrovni. Mám jen drobné výhrady:

- Zelená barva v tabulkách (pravděpodobně 0,255,0 v RGB) je poměrně špatně čitelná, volil bych tmavší zelenou.

- Používání zlomků v matematických vzorcích uvnitř textu není příliš vhodné, protože vede na násilné zvětšování odsazování přilehlých řádků, viz např. strana 28. Místo zlomků doporučuji používat lomítko pro vyjádření poměru, tedy např. " $n/2$ " místo " $\frac{n}{2}$ ".

Hodnotící kritérium:

Způsob hodnocení - bodové hodnocení 0 až 100 bodů (známka A až F):

6. Práce se zdroji

100 (A)

Popis kritéria:

Vyjáďte se k aktivitě studenta při získávání a využívání studijních materiálů k řešení ZP. Charakterizujte výběr studijních pramenů. Posudte, 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:

Autor pracuje se zdroji na velmi dobré úrovni, nemám žádné výhrady.

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:

Vyjáďte 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 zajímavé a určitě relevantní k současné problematice paralelního stabilního řazení s nižšími paměťovými nároky, než má klasický out-of-place MergeSort. Hardwarová architektura použitá k měření byla vhodně zvolena, pouze je potřeba uvědomit si NUMA uspořádání paměti a případně by bylo zajímavé jeho vliv na výkon algoritmů změřit. Verze GNU C++ překladače 4.8.5 je trochu starší, dnes se běžně používají verze řady 5 až 7, které by mohly mít o něco pokročilejší optimalizační schopnosti, ale to není jisté. Rovněž by bylo zajímavé porovnání výsledků v případě použití Intel překladače, který je obecně považován za výkonnější, ale předpokládám, že nebyl autorovi k dispozici.

Hodnotící kritérium:

Způsob hodnocení - nehodnotí se

8. Komentář o využitelnosti výsledků

Popis kritéria:

Uvedte, zda hlavní výsledek ZP rozšiřují již publikované známé výsledky a/nebo přinášející zcela nové poznatky. Uvedte možnosti využití výsledků ZP v praxi.

Komentář:

Výsledky získané autorem práce jsou určitě využitelné při dalším rozvoji studované oblasti, tedy vícevláknového stabilního řazení s nízkými paměťovými nároky. Sekce 3.1.4 a 3.1.5 popisují data použitá k experimentům a jejich počáteční uspořádání. Použitých kombinací je poměrně hodně, přesto by ale podle mého názoru práci velmi obohatilo zahrnutí těchto případů:

a) Veškeré typy vstupních dat vyžadují konstantní čas pro porovnání dvou prvků. Bylo by zajímavé studovat chování algoritmu i na opačných případech (např. při řazení řetězců, kde doba porovnání závisí na tom, na kolikátém znaku s řetězce liší, a tudíž není konstantní).

b) V praxi se často pracuje s daty, která nejsou ani uspořádána náhodně ani seřazena na vstupu. Takováto data jsou často simulována částečně seřazenými daty. Jejich využití pro měření by rovněž bylo pro čtenáře zajímavé.

Hodnotící kritérium:

Způsob hodnocení - nehodnotí se

9. Otázky k obhajobě

Popis kritéria:

Uvedte případné dotazy, které by měl student zodpovědět při obhajobě ZP před komisí (body oddělte odřádkami).

Otázky:

1) Cituji ze sekce 3.1.1 "Zvolen byl jazyk C++, protože oproti jazyku C obsahuje template funkce, které umožňují překladači místo volání funkcí na porovnání (prohození) dvou prvku vložit instrukce těchto funkcí přímo do kódu řadicí funkce (inline)." Znamená to snad, že programovací jazyk C neumožňuje uvedenou techniku vkládání "inline" kódu funkcí na místo volání na úrovni strojového kódu?

2) V Sekci 3.2.1.4 se píše cituji "První cyklus ve funkci Merge je netradičního tvaru a kompilátor ho nedokáže sám automaticky rozbalit, musí se tedy rozbalit explicitně.". Zajímalo by mě, zda jde pouze o názor autora nebo zda je podložený nějakými fakty (např. studiem "disassemblovaného" strojového kódu výsledného programu apod.). Tabulka 3.5 sama svědčí o tom, že ruční rozbalení cyklů nemá pro výsledné časy přínos, spíše naopak.

3) Jak byla alokována pole dat pro měření? Byla data nějakým způsobem zarovnána, např. na 64-bytové adresy, které odpovídají velikosti cache řádkům? Mohlo by takovéto zarovnání mít vliv na výkon, např. v blokových verzích algoritmů?

4) Proč nebyl v hybridním algoritmu použit také InsertionSort, když to bylo požadováno v zadání práce?

Hodnotící kritérium:

Způsob hodnocení - bodové hodnocení 0 až 100 bodů (známka A až F):

10. Celkové hodnocení

89 (B)

Popis kritéria:

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

Text hodnocení:

Celkově je práce na velmi vysoké úrovni, problematika je nastudována do podrobností a dobře představena a vysvětlena čtenáři. Autor změřil efekt velkého množství variant uvažovaných algoritmů, navrhl nový algoritmus a změřil jeho chování v dnes všudypřítomném vícejádrovém prostředí. Za hlavní nedostatek považuji absenci InsertionSortu ve výsledném hybridním algoritmu, zvláště, když autor ukazuje, že kombinace s InsertionSortem má pro MergeSort obecně dobrý vliv z hlediska minimalizace času řazení.

Podpis oponenta práce: