



Posudek oponenta závěrečné práce

Student: Bc. Jaroslav Ryba
Oponent práce: Ing. Daniel Langr, Ph.D.
Název práce: Efektivní násobení řídkých matic
Obor: Webové a softwarové inženýrství

Datum vytvoření: 23. 5. 2019

Hodnotící kritérium:	Způsob hodnocení – následující škálou 1 až 4:
1. Splnění zadání	<u>1=zadání splněno,</u> 2=zadání splněno s menšími výhradami, 3=zadání splněno s většími výhradami, 4=zadání nesplněno
<i>Popis kritéria:</i> Posuďte, zda předložená ZP dostatečně a v souladu se zadáním obsahově vymezuje cíle, správně je formuluje a v dostatečné kvalitě naplňuje. V komentáři uveďte body zadání, které nebyly splněny, posuďte závažnost, dopady a případně i příčiny jednotlivých nedostatků. Pokud zadání svou náročností vybočuje ze standardů pro daný typ práce nebo student případně vypracoval ZP nad rámec zadání, popište, jak se to projevilo na požadované kvalitě splnění zadání a jakým způsobem toto ovlivnilo výsledné hodnocení.	
<i>Komentář:</i> Práce cíle zadání plně splňuje. Řešený problém je netriviální, ať už charakterem operace násobení řídkých matic, tak jejím vykonáváním na GPU, které jsou obecně vhodné spíše pro zpracování "hustých" dat.	
Hodnotící kritérium:	Způsob hodnocení – bodové hodnocení 0 až 100 bodů (známka A až F):
2. Písemná část práce	89 (B)
<i>Popis kritéria:</i> Zhodnoťte přiměřenost rozsahu předložené ZP vzhledem k obsahu, tj. zda všechny části ZP jsou informačně bohaté a ZP neobsahuje zbytečné části. Dále 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. Posuďte správnost používání formálních zápisů obsažených v práci. Posuďte typografickou a jazykovou stránku ZP, viz Směrnice děkana č. 26/2017, článek 3. Posuďte, zda student využil a správně citoval relevantní zdroje. Ověřte, zda jsou všechny převzaté prvky řádně odlišeny od vlastních výsledků, zda nedošlo k porušení citační etiky a zda jsou bibliografické citace úplné a v souladu s citačními zvyklostmi a normami. Zhodnoťte, zda převzatý software a jiná autorská díla, byly v ZP použity v souladu s licenčními podmínkami.	

Komentář:

Rozsah práce je dostatečný a práce je po věcné stránce velmi dobře zpracovaná. Text je logicky členěn, jednotlivé kapitoly a sekce na sebe obsahově navazují a obash je pro čtenáře pochopitelný. Některé výhrady k obsahu spíše drobného charakteru:

- str. 1 - "Správně implementovaný a optimalizovaný algoritmus tak může teoreticky získat přístup k řádově sto i tisícinásobným výpočetním prostředkům." - Toto tvrzení se mi zdá poněkdu nadsazené. Např. dnešní špičkové (GP)GPU karty dosahují výkonu do cca. 15 TFLOPS v jednoduché přesnosti, kdežto špičkové komoditní Intel procesory řádově jednotky TFLOPS. Rozdíl v teoretickém výkonu tak rozhodně není sto či tisíci násobný.
- str. 7 - "CSC, také nazývaný Harwell-Boeingův formát..." - Pokud vím, HB formát je speciální souborový formát pro uložení řídkých matic do souborů (podobně jako Matrix Market formát). CSC je formát uložení matic v paměti.
- str. 13 a další - "dostupných z [14]" - Existuje pravidlo, které říká, že věta by měla dávat smysl i po odstranění citační reference. Lepší je formulace např. "zkoumání kódů Sparse BLAS [14]" apod.
- str. 14 a další - "Interface" -> "Rozhraní" (Nejsem zastávce překladu termínu za každou cenu, ale "rozhraní" je zcela běžný pojem používaný v ČJ.)
- sekce 1.4. - Jádro problému násobení řídkých matic je právě v tom, že u nich nemáme přímý přístup k prvkům v konstantním čase (čtení/zápis $A[i][j]$). Autor toto zmiňuje v sekci 1.4.1.2 a 1.4.2.2, ale již nepopisuje, jak se s tímto problémem uvedené knihovny vypořádávají. Tj. píše, že jsou založené na základním algoritmu násobení matic, který používá přímou indexaci maticových prvků, ale již nezkoumá, jak knihovný tento přístup řeší pro řídké matice. To je podle mě poměrně důležité vzhledem k řešenému problému této práce.
- Sekce 2.3 - autor pro jednotlivé formáty uvádí algoritmy pro transformaci z řídké matice. Tyto algoritmy jsou ale pro řešený problém víceméně irelevantní. Naopak, algoritmy pro násobení uvedené nejsou a jsou jen slovně popsány (sekce 2.3.[1-4].4). Mnohem přínosnější by bylo uvést právě tyto algoritmy ve formě pseudokódu. Speciálně následně pro pochopení jejich paralelizace. Viz i začátek sekce 2.4 "Jak si lze snadno povšimnout z předchozí sekce, úpravou sekvenčních algoritmů můžeme získat paralelní verze..." - po pravdě si toho snadno povšimnout nelze, když sekvenční algoritmy nejsou uvedeny.
- str. 40 - "vypočítává průměrnou relativní odchylku" - Při počítání "chyby", tj. ověřování správnosti výsledku v tomto případě, je vždy lepší brát maximální odchylku, nikoliv průměrnou.
- str. 41 - "Na grafu 4.1 je vidět ... u všech formátů kubicky závislá..." - Toto na grafu 4.1 pro formáty CRS a ELL opravdu vidět není. Lepší by bylo v tomto případě logaritmické měřítko osy Y.
- str. 45 - "Pro tento formát byl omezen maximální počet nenulových diagonál..." - Pro jaký formát? Pro BCSR, o kterém se mluví před tím? Nebo pro DIA, který ale zmiňuje až následující odstavce. Z obsahu odstavce spíše dává smysl první varianta.

Po formální stránce je práce rovněž výborně zpracována. Opět jsem našel pouze několik drobných nedostatků:

- str. 4. - Definice 1.8. - Řádek za vzorcem by neměl začínat samostatnou tečkou. Ta by měla být součástí vzorce.
- str. 5. - "fsetů" -> "offsetů" (pravděpodobný překlep)
- str. 13 a další - "známým algoritmem.[12]" - Reference na citaci by měla být součástí věty.
- str. 14 - seznam (odrážky) je buď součástí jedné věty a nebo jednotlivé položky tvoří samostatné věty. V prvním případě položky seznamu začínají malým písmenem a končí čárkou, případně středníkem. V druhém případě položky začínají velkým písmenem a končí tečkou.
- str. 21 - Algoritmus 2 - Doporučoval bych vyhnout se zápisu "count++" v obecném pseudokódu algoritmu. Tento zápis je doménou (některých) programovacích jazyků, kdežto pseudokód algoritmu by měl být obecně čitelný při znalosti základní matematické syntaxe. Lepší např. přidat řádek "count <- count + 1".
- obrázek 2.1 je špatně čitelný, místo na zvětšení písma je na něm přítom dost.
- str. 36 - "neoptimálnější" -> "optimálnější"
- str. 36 - "...že bottleneck efektivitu..." -> "úzké hrdlo (bottleneck)" apod. (v neformálním textu je to v pořádku, ve formálním ne).
- str. 37 - "z jednoho floatu na celý blok" - dtto.

Relevantní zdroje byly citovány. V seznamu citací jsou drobné formální chyby - viz [22] a [23] (pravděpodobně tam chybí část názvu / jména autora).

Hodnotící kritérium:

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

3. Nepísemná část, přílohy

100 (A)

Popis kritéria:

Dle charakteru práce se případně vyjádřete k nepísemné části ZP. Například: SW dílo – kvalita vytvořeného programu a vhodnost a přiměřenost technologií, které byly využity od vývoje až po nasazení. HW – funkční vzorek – použité technologie a nástroje, Výzkumná a experimentální práce – opakovatelnost experimentů

Komentář:

Vytvořené zdrojové kódy mají dobrou strukturu, jsou čitelné, komentované a logicky rozdělené na příslušné jednotky.

Hodnotící kritérium:

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

4. Hodnocení výsledků, jejich využitelnost

75 (C)

Popis kritéria:

Dle charakteru práce zhodnoťte možnosti nasazení výsledků práce v praxi nebo uveďte, zda výsledky ZP rozšiřují již publikované známé výsledky nebo přinášející zcela nové poznatky.

Komentář:

Využitelnost výsledků práce je poměrně diskutabilní z jednoho zásadního důvodu, a sice, že autor uvažoval pro ukládání výsledné matice pouze hustý formát. To v první řadě činí porovnání výkonnosti navržených algoritmů s existujícím řešením z `cuSparse` poněkud nesmyslné, protože zde se ukládají výsledky do řídkých matic, což je mnohem komplikovanější záležitost. Hlavně to ale brání práci s maticemi takových rozměrů, které se v hustém formátu do paměti počítače/GPU karty nevejdou. A takové matice vznikají v praxi zcela běžně.

Na druhou stranu, jak autor sám píše, navržené řešení je spíše prototypem a v rámci omezení ukládání výsledku v hustém formátu byly získány velice zajímavé poznatky. Zda bude jejich extrapolace i pro výsledný řídký formát použitelná a přínosná ukáže budoucnost.

Hodnotící kritérium:

Způsob hodnocení – nehodnotí se

5. Otázky k obhajobě

Popis kritéria:

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

Otázky:

1) Pokud jsem text správně pochopil, autor předpokládá, že v příslušných formátech jsou v paměti uloženy všechny prvky všech matic, účastnících se operace násobení. Pro symetrické matice je ale zvykem ukládat pouze jejich horní či dolní trojúhelníkovou část. Uvažoval autor takovéto případy alespoň teoreticky? Jak složitá by byla modifikace algoritmů pro podporu symetrických matic?

2) Pro účely prototypu autor uvažuje reprezentaci hodnot maticových prvků v jednoduché přesnosti (float). Jak složitá by byla podpora např. i dvojité přesnosti, celých čísel, či Boolovských hodnot? Stačilo by příslušné třídy a funkce jednoduše parametrizovat pomocí C++ šablon, nebo by řešení vyžadovalo složitější přístup?

Hodnotící kritérium:

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

6. Celkové hodnocení

95 (A)

Popis kritéria:

Shrňte stránky ZP, které nejvíce ovlivnily Vaše celkové hodnocení. Celkové hodnocení nemusí být aritmetickým průměrem či jinou hodnotou vypočtenou z hodnocení v předchozích jednotlivých kritériích. Obecně platí, že bezvadně splněné zadání je hodnoceno klasifikačním stupněm A.

Text hodnocení:

Práce se mi celkově velmi líbila. Zabývá se netriviálním problémem, splňuje zadání, obsah je velmi dobře a čtivě podán a zdrojové kódy jsou kvalitní a vytvořené podle správných programátorských zásad jazyka C++. Dosažené výsledky jsou zajímavé a využitelné pro další výzkum problému násobení řídkých matic na grafických procesorech.

Podpis oponenta práce: