FAKULTA INFORMAČNÍCH TECHNOLOGIÍ ČVUT V PRAZE

Posudek oponenta závěrečné práce

Oponent práce:	Ing. Daniel Langr, Ph.D.
Student:	Bc. Ondřej Voronecký
Název práce:	Efektivní paralelní vícecestný Quicksort algoritmus
Obor / specializace:	Počítačové systémy a sítě
Vytvořeno dne:	5. ledna 2024

Hodnotící kritéria

1. Splnění zadání

▸ [1] **zadání splněno**
[2] zadání splněno s menšími výhradami
[3] zadání splněno s většími výhradami
[4] zadání nesplněno

Zadání bylo splněno bez výhrady.

2. Písemná část práce 96 /100 (A)

Písemná část práce je velmi zdařilá, obsahuje všechny nezbytné části a ty na sebe logicky navazují. Text je dobře čitelný a srozumitelný, bez chyb a překlepů. Práce je zdařilá i po formální stránce. Mám zde pouze několik málo výhrad, často spíše zanedbatelného charakteru:

1) Nedá se říct, že quicksort je součástí standardní knihovny C++. Standard nepředepisuje pro implementaci žádný konkrétní algoritmus, to je záležitostí implementace.
2) Pojem "pivot" by dle mého názoru měl být neživotný. Tj. např. místo "jednoho pivota" by bylo vhodnější psát "jeden pivot", apod. na více místech v textu.
3) OpenMP není jen knihovna, alespoň ne v tom obvyklém smyslu. Spíše se jedná o nadstandardní rozšíření vybraných programovacích jazyků. Součástí tohoto rozšíření je i knihovna poskytující určité OpenMP funkce, ale to je jen jedna součást.
4) Ve vědecké literatuře obecně není zvykem dávat reference na literaturu až za konec věty (ale asi to může být věcí názoru). Místo "...pořadí. [1] [2]" bych doporučoval "...pořadí [1] [2]." (nebo ještě lépe "...pořadí [1,2].".
5) Na stránce 3 je předložka "s" na konci řádku.
6) Podstatná a relevantní výhoda algoritmu insertion sort (IS) je, že je velmi efektivní pro data, která jsou "skoro seřazená". V souvislosti s algoritmem quicksort je to důležité, protože místo aplikace IS na každou krátkou posloupnost dosaženou v rámci rekurze je možné IS aplikovat jen jednou na konci (tako to efektivní implementace dělají), případně

jednou pro každé vlákno.
7) Obecně se doporučuje vyhýbat se zlomkům v rámci textu odstavce, pokud je to možné. Např. polovinu lze bez zlomku napsat jako "1/2", což je v textu čitelnější a obecně to nezvyšuje nutnost zvyšovat mezeru mezi řádky. To samé platí o druhém zlomku na straně 11.
8) Pro bitový posun by byl lepší operátor "mnohem menší než" (v Latexu \ll) než "<<".
9) Ve vědecké literatuře bývá zvykem umisťovat plovoucí prostředí (obrázky, tabulky, algoritmy,...) v rámci stránky na její horní část. Umístění "v textu" někdy i působí rušivě, např. na stránce 44, kde díky tomu nad obrázkem jsou jen dva řádky textu.
10) "Počet jader procesorů je běžně právě mocninou dvou..." - Osobně bych si toto netroufal tvrdit, procesory s jinými počty jader jsou všudypřítomné.
11) Algoritmy GNU QS a BQS nejsou součástí standardní knihovny C++. Jsou součástí knihovny GNU libstdc++, ale využívají OpenMP, které samotný jazyk C++ nezná. Tato součást se nazývá "libstdc++ parallel mode" a byla popsána v článku, na který chybí reference (autoři Singler a Kosnik).

3. Nepísemná část, přílohy 100/100 (A)

Implementace algoritmu je dostupná ve formě zdrojového kódu v jazyce C++. Kód je velice dobře napsaný a strukturovaný. Navíc použití vyžaduje pouze vložení jediného hlavičkového souboru, což je pro uživatele prakticky ten nejjednodušší způsob. Kód je navíc dokumentovaný formou komentářů pro nástroj Doxygen. Líbilo se mi rovněž využití vhodných nástrojů Catch2 a Google Benchmark pro testování korektnosti algoritmu a měření jeho efektivity.

4. Hodnocení výsledků, jejich využitelnost 90/100 (A)

Autor provedl rozsáhlé experimentální vyhodnocení navrženého a implementovaného algoritmu. Oceňuji i rozsáhlé testování parametrů algoritmu. Ve všech případech ale autor používal pouze synteticky generovaná vstupní data. Určitě by bylo zajímavé vidět i výsledky pro data generovaná nějakou reálnou aplikací. (Vhodnou testovací sadou mohou být například volně dostupné velké řídké matice, kde je obecně potřeba řadit jejich prvky kvůli následnému uložení v určitém formátu. Navíc různé formáty vyžadují řazení podle různých kritérií, které mohou mít vliv na efektivitu jednotlivých řadících algoritmů.) U výsledků bych rovněž uvítal více porovnání s časy, které dosahuje jednovláknový std::sort. I pro něj by bylo vhodné vidět, jak se chová pro různá data a mít možnost vyhodnotit urychlení paralelních algoritmů.

Celkové hodnocení 96/100 (A)

Celkově je práce velmi zdařilá po všech jejích stránkách. Řešený problém byl poměrně složitý (již paralelní quicksort s jedním pivotem představuje sám o sobě netriviální problematiku) a autor se jeho řešení zhostil výborně. Navržené řešení je navíc použitelné v praxi a pro určité případy vstupních dat dokonce dosahuje nejrychlejších časů řazení.

Otázky k obhajobě

1) Byla při měření škálovatelnosti zohledněna NUMA architektura? Tj. při použití 1 až 10 vláken, byla tato vlákna mapována na jeden NUMA uzel (CPU)?

2) Částečně mi není jasný používaný pojem "mohutnost hodnot". Znamená např. mohutnost hodnot 1, že všechny řazené prvky mají stejnou hodnotu?

3) Implementace prozatím není veřejně dostupná. Plánuje autor její zveřejnění? (A případně i konferenční/časopiseckou publikaci? Ale to je dotaz asi spíše na vedoucího práce.)

Instrukce

Splnění zadání

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.

Písemná část práce

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. 52/2021, 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.

Nepísemná část, přílohy

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

Hodnocení výsledků, jejich využitelnost

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.

Celkové hodnocení

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.