

Hledání percentilu v souboru

závěrečná prezentace SP z KIV/PPR

Stanislav Král
(A20N0091P)

Západočeská univerzita v Plzni

Fakulta aplikovaných věd

7. prosince 2021

- oproti původním diagramům CF a DF pouze minimální nepodstatné změny
- navržený algoritmus využívající bitový posun pro výpočet indexu bucketu se osvědčil jako velmi efektivní
 - následný návrh efektivní paralelizace nebyl jednoduchý

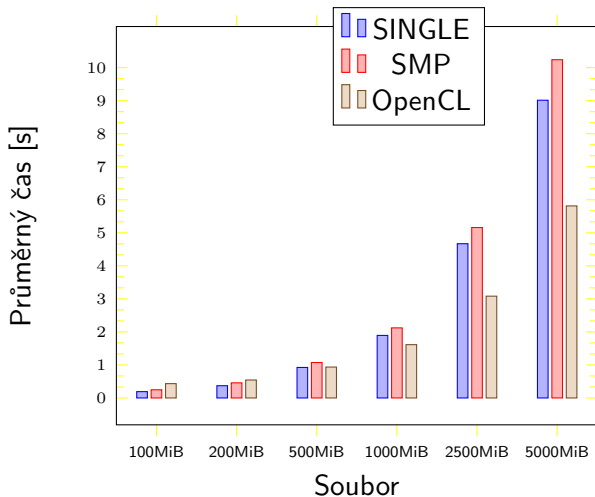
- v některých případech má použití klíčového slova `inline` velký (pozitivní) výkonostní dopad
 - hlavně pozorováno u klasifikace validity zpracovávaného čísla
 - vlastní implementace `std::fpclassify` rychlejší (z učiněných pozorování až o desítky procent)
- velikost bufferu při čtení ze souboru výrazně ovlivňuje rychlost celého výpočtu
 - optimální velikost se liší u jednotlivých variant výpočtu (`single`, `SMP` a `OpenCL`)
- vytvoření třídy pro čtení dat ze souboru s využitím bufferu za účelem sdílení kódu mezi varianty výpočtu vedlo k celkovému zpomalení výpočtů
 - z tohoto důvodu není kód čtení ze souboru sdílený a opakuje se
 - z učiněných pozorování až desítky procent

- SMP původně implementováno pomocí `parallel_for`, avšak pomalé, následně nahrazeno implementací pomocí `flow_graph`
- dokumentace knihovny TBB je dle mého názoru nedostatečná, což mi výrazně ztěžovalo implementaci SMP
 - nesrozumitelné chyby během překladu a nedostatek příkladů použití konstrukcí `flow_graph`
- u optimalizace OpenCL výpočtu hraje podstatnou roli počet zápisu/čtení do/z OpenCL bufferů
 - původní implementace výpočtu, kdy bylo OpenCL zařízení využíváno jen k realizaci bitových posunů, trvala jednou tolik času, než sériová implemetace
 - následná úprava výpočtu tak, aby byl celý histogram sestavován na OpenCL zařízení, výrazně zrychlila tento výpočet
 - dle naměřených výsledků je pak urychlení oproti sériovému výpočtu 1.32

- výpočet je mnohem rychlejší na OS Linux než na OS Microsoft Windows
 - na Linuxu dochází k jevu, kdy opakovaná spuštění programu nad jedním a tím samým souborem jsou rychlejší
 - na souboru s obrazem OS Ubuntu se jednalo o zrychlení i několika sekund
- během testování na Intel HD zjištěno, že některá volání vytváření OpenCL bufferu a zápisu do něj, která na zařízeních NVidia a AMD fungují v pořádku, na tomto zařízení nefungují zcela správně
 - bylo třeba provést drobné úpravy kódu na straně C++
 - samotný OpenCL kód zůstal stejný
- nesprávná práce s OpenCL zařízením má na OS Linux schopnost shodit celý systém

Zařízení	1.	2.	3.	4.	Průměr[s]
SINGLE	4.197	4.308	4.195	4.257	4.239
SMP	5.486	5.466	5.496	5.535	5.495
Ellesmere	3.294	3.069	3.374	3.105	3.211

Tabulka: Časy běhů programu při zpracování souboru s obrazem Ubuntu OS (2.7GiB).



Obrázek: Graf zobrazující data naměřená při spouštění programu pro soubory o různých velikostech.

- mým hlavním cílem bylo mít co nejrychlejší výpočet
 - avšak bohužel i někdy za cenu čitelnosti kódu
 - např. u SMP za účelem rychlosti výpočtu zvýšeny paměťové nároky (kopie histogramu pro každé vlákno namísto synchronizovaného přístupu k jednomu jedinému)
- bohužel, sériová verze je při použití všech optimalizací při překladu rychlejší než SMP
 - pravěpodobně způsobeno špatným návrhem paralelizace
 - možnost prozkoumání do budoucna, zdali neexistuje lepší způsob
- na středně výkonném GPU (AMD RX480) výpočet rychlejší než na 12-ti vláknovém procesoru (AMD Ryzen 1600 6C/12T)

Konec prezentace