



Jméno diplomanta: Bc. Milan Klášterka

Garantující katedra: KKY

Název diplomové práce: Akcelerace klasifikačních algoritmů založených na SVM pomocí grafických karet

	Předmět hodnocení	Nadprůměrné	Průměrné	Podprůměrné
1	Jazyková a grafická úprava	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2	Formální a obsahová stránka práce	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	Vhodnost použitých metod	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	Způsob zpracování a vyhodnocení	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5	Správnost získaných výsledků	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6	Vlastní přínos	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Doplnění hodnocení, připomínky, dotazy:

Cílem práce bylo implementovat SVM klasifikátor pro grafické karty (GPU) v jazyce CUDA C a výsledky a rychlost trénování i rozpoznávání porovnat s existujícími implementacemi.

Text práce o délce 52 stran je kvalitně strukturován, obsahuje menší počet překlepů. V první teoretické části je uveden úvod do problematiky výpočtů na GPU, do problematiky klasifikace, je detailně popsán klasifikátor SVM a trénovací algoritmus SMO (Sequential Minimal Optimization). Z popisu je zřejmé, že student tuto netriviální problematiku kvalitně teoreticky nastudoval.

Druhá část práce se věnuje detailní analýze existujících implementací, kde jedna je implementována na CPU a 3 na GPU. Student tyto implementace jednak analyzoval po teoretické stránce, tak i upravoval jejich kód pro vlastní potřeby.

Přestože je v textu práce vlastní implementace popsána pouze na necelých dvou stránkách, tak z přílohy na DVD, která obsahuje zdrojové kódy vlastní implementace, je zřejmé, že student tento netriviální algoritmus dokázal implementovat v paralelním výpočtu v jazyce CUDA C.

Z provedených experimentů je patrné, že výsledky výpočtů jsou stejné jako v cizích implementacích, čímž se ověřila správnost vlastní implementace, tak že i rychlost často překonává cizí implementace.

Je škoda, že student neprovedl více experimentů a více srovnání, na kterých by se detailně vyhodnotily přednosti vlastní implementace. Zřejmě z časových důvodů se to již nestihlo, čemuž odpovídá i stručnost kapitoly "experimenty", která je popsána pouze na čtyřech stránkách, což hodnotím jako největší nedostatek práce.

Přednosti práce jsou nastudování netriviálního algoritmu a jeho převedení do paralelní GPU implementace, srovnání a pochopení funkčnosti cizích implementací a správnost získaných výsledků.

Dotazy:

1) V práci jsou analyzovány 4 cizí implementace SVM klasifikátoru, libSVM, cuSVM, gpuSVM, GPU-accelerated libSVM. Proč byly vybrány právě tyto 4 implementace? Existují i jiné?

2) Ve výsledcích (tabulka 8.4 a 8.5) pro trénování vaší implementace cudaSVM uvádíte výborné výsledky v rychlosti, překonávající většinou cizí implementace, ve zhodnocení výsledků ale již není uvedeno, v jakých případech ostatní implementace překonáváte a v jakých ne. Dokážete ze získaných výsledků vysledovat, kdy je vaše implementace rychlejší, případně i zdůvodnit proč?

Splnění bodů zadání	<input checked="" type="checkbox"/> úplně	<input type="checkbox"/> částečně	<input type="checkbox"/> nesplněno	
Doporučení práce k obhajobě	<input checked="" type="checkbox"/> ano		<input type="checkbox"/> ne	
Celkové hodnocení práce	<input checked="" type="checkbox"/> výborně	<input type="checkbox"/> velmi dobře	<input type="checkbox"/> dobře	<input type="checkbox"/> nevyhověl
Jméno, příjmení, titul oponenta: Ing. Pavel Campr, Ph.D.				
Pracoviště oponenta: ZČU NTIS				

12. 6. 2014

Datum

Podpis

**SOUHLASÍ
S ORIGINÁLEM**