Záměr týmového projektu

Studijní program: **Softwarové a datové inženýrství**
Typ projektu: **softwarový**
Studenti-řešitelé: **Jiří Mayer, Jiří Klepl, Petr Šimůnek**
Supervizor: **RNDr. Martin Kruliš, Ph.D.**
Konzultant: **RNDr. Miroslav Kratochvíl**
Název a téma projektu: **Framework pro optimalizované datové výpočty na GPU**

Hlavním cílem je navrhnout framework, který by usnadnil práci při vývoji vysoce optimalizovaných GPU algoritmů pro datovou analýzu a jejich následné použití v obecně dostupných knihovnách. Tento vývojový proces je zpravidla poměrně zdlouhavý, neboť vyžaduje experimentální vědeckou práci (hledání vhodné implementace daného algoritmu, jeho optimalizace a tuning spouštěcích parametrů) a následnou softwarově-inženýrskou práci, kdy je hotová implementace (v podobě GPU kernelu) nabídnuta ve formě knihovny a bindingů do jazyků a prostředí, které zpřístupní výsledné řešení širší (především vědecké) komunitě.

Dílčí cíle je možné shrnout do tří hlavních bodů:

- Implementace často se opakujících fragmentů zdrojových kódů ve formě šablonových tříd. Tato část pokrývá zejména inicializaci GPU (prostřednictvím zvolené technologie), správu paměti a dat, a vytváření exekučních plánů spouštění jednotlivých kernelů.
- Podpora pro detailní experimentální měření výkonu implementovaných GPU algoritmů a pro systematické prohledávání prostoru konfiguračních parametrů (počty spuštěných vláken, velikost práce alokovaná na vlákno, velikost účelně cachovaných dat, ⋯) pomocí metod globální optimalizace.
- Předpřipravené bindingy a deployovací proces pro finální vytvoření knihovny výstupů experimentálního procesu (pro jazyky C++, Python a R).

Projekt bude řešen výhradně v jazyce C++ s použitím technologie CUDA. Součástí projektu je navrhnout vhodná rozhraní, aby bylo možné jej rozšířit na další známé GPU technologie a platformy (zejména Vulkan a OpenCL). Při řešení bude zváženo použití podpůrných knihoven a technologií (např. SYCL, Hip).

Součástí projektu bude adaptace existujícího prototypu GPU algoritmu (CUDA k-means https://github.com/krulis-martin/cuda-kmeans), což poslouží jako demonstrace použitelnosti daného řešení.

Orientační harmonogram prací:

- Do konce roku 2020: analýza, rešerše, seznámení týmu s problematikou, návrh kritických softwarových rozhraní, řešení otázek přenositelnosti mezi GPU platformami (implementace demonstračních prototypů v rámci studie proveditelnosti)
- Leden: Dokončení návrhu interfaces klíčových částí, implementace knihoven pro správu paměti, strukturovaných dat, a přenosu dat mezi hostem-GPU
- Únor: Studie použitelnosti datových knihoven, dokončení návrhu všech rozhraní (včetně vnějších)
- Březen: implementace abstrakce pro spouštění kernelů, automatická realizace datových transferů v závislosti na exekučním plánu
- Duben: implementace podpory pro výkonnostní testování a autotuning
- Květen: příprava deployment scénářů, implementace globálních rozhraní do C++, Pythonu a R
- Červen: začátek implementace knihovny CUDA k-means s použitím vytvořeného frameworku
- Červenec-srpen: testování, ladění, finalizace, dokumentace

Vývoj bude interně probíhat přibližně ve 14 denních iteračních cyklech, na konci každého cyklu proběhne revize stavu vedoucím nebo konzultantem a případná korekce plánu na další cyklus.