

## Individuali užduotis sesijos metu

### Užduotis

Programos duomenų failai tokio pat formato, kaip L1, L2 ir L3 metu, duomenų kiekis — bent 300 įrašų. Sistema susideda iš dviejų programų: pagrindinės (realizuojama **C++ OpenCL**) ir pagalbinės (realizuojama **Python multiprocessing: Process ir Queue**) (išsiaiškinti naudojimo principus savarankiškai). Programos komunikuoja per tinklą. Rezultatų failas panašus, kaip L1 ir L2 darbų metu, tačiau reikalingos dvi sudėtingos funkcijos bei du filtravimo kriterijai. Rezultatai – tekstinio tipo. Rezultatų faile turi būti pradiniai duomenys ir abiejų apskaičiuotų funkcijų rezultatai. Paruošti tokius duomenų failus:

1. Visi įrašai atitinka abi filtro sąlygas (rezultatuose visi įrašai);
2. Visi įrašai neatitinka pirmosios filtro sąlygos, bet bent dalis atitinka antrą filtro sąlygą (rezultatuose 0 įrašų);
3. Visi įrašai neatitinka antrosios filtro sąlygos, bet bent dalis atitinka pirmą filtro sąlygą (rezultatuose 0 įrašų);
4. Dalis įrašų atitinka abi filtro sąlygas (rezultatuose įrašų yra, bet ne visi).

Pagrindinė C++ programa turi tris lygiagrečiai veikiančias gijas:

- OpenCL vykdytojas (realizuojama visa su OpenCL susijusi logika);
- Python siuntėjas (siunčia pradinį duomenį per tinklą į Python dalį);
- Python gavėjas (priima rezultatus iš Python).

Gavus rezultatus iš Python ir iš OpenCL jie apjungiami ir išvedami lentele į rezultatų failą.

Reikalavimai OpenCL daliai:

- OpenCL apskaičiuoti rezultatai surašomi be tarpų (jeigu neatitinka filtro sąlygos, į OpenCL branduolio rezultatą elementas nerašomas visai).
- Priklausomai nuo naudojamo kompiuterio įrangos parenkamas tinkamas OpenCL vykdytojo variantas (pvz., jeigu turite NVidia ar AMD vaizdo plokštę, ji ir turi būti naudojama; jeigu ne, naudokite integruotą vaizdo procesorių; jeigu jis nepalaikomas, vykdykite skaičiavimus CPU).
- Programoje turi būti atspausdintas pasirinktas OpenCL įrenginys (platforma, įrenginio pavadinimas).

Python programa vykdo tokius procesus:

- Gavėjas, kuris gauna duomenis iš C++ dalies;
- Darbininkai, kurie skaičiuoja pasirinktą funkciją;
- Siuntėjas, kuris siunčia rezultatus į C++ dalį.

Reikalavimai Python daliai:

- Procesai tarpusavyje duomenimis apsieičia naudodami eiles (`multiprocessing.Queue`).
- Į C++ tinklu siunčiamas minimalus duomenų kiekis – įrašai, kurie netenkina filtro sąlygos, nesiunčiami, taip pat nesiunčiami struktūros laukai, kurių C++ pusėje atsakyme nėra ieškoma.

Komunikaciją per tinklą galima realizuoti bet kokiomis priemonėmis, pvz., *socketais*. Gali prireikti prisijungimų į abi puses, tą galima spręsti bet kokiais būdais, pvz., C++ gavėjas paleidžiamas iškart, o siuntėjas — tik kai naudotojas paspaudžia mygtuką; per tą laiką paleidžiama Python programa. Galima bandyti jungtis ir tikrinti, ar pavyko; jeigu ne, palaukti ir bandyti iš naujo. Galima spręsti ir kitais būdais.

Darbininkų kiekis parenkamas maksimalus pagal turimą CPU, vieną branduolį paliekant laisvą C++ skaičiavimams. Jeigu OpenCL skaičiavimus vykdo CPU, tai naudokite pusę branduolių.

Jei įrašas atitinka abu pasirinktus kriterijus, jis turėtų atlikti tokia „kelionę“: failas -> C++ pagrindinis; iš ten siunčiama dviem kryptimis: C++ pagrindinis -> C++ OpenCL vykdymo gija (perduodama parametru) -> C++ OpenCL branduolys -> C++ OpenCL vykdymo gija; C++ pagrindinis -> C++ siuntėjas -> tinklas -> Python gavėjas -> Python darbininkas -> Python siuntėjas -> tinklas -> C++ gavėjas. Pasibaigus OpenCL ir gavėjo gijoms rezultatai apjungiami ir galutinai nufiltruojami C++ pagrindinėje gijoje ir išspausdinami faile.

Darbo pabaigoje visos C++ gijos ir Python procesai, jei reikia, gauna darbo pabaigos signalus ir patys užsibaigia.

Papildomi reikalavimai programai:

- Python procesai turi būti sukuriama po vieną kartą, o ne kaskart priimant žinutę.
- Funkcijos skaičiavimas visam duomenų failui su vienu darbininku tiek C++ dalyje, tiek Python dalyje turėtų trukti bent po 6 sekundes (pvz., galima daug kartų perskaičiuoti maišos funkciją kiekvieną kartą ją skaičiuojant nuo reikšmės iš praeitos iteracijos).
- Reikia išmatuoti ir programos komentaruose pateikti, kiek laiko programa vykdoma naudojant vieną OpenCL darbo elementą ir vieną Python darbininką ir kiek laiko vykdoma naudojant maksimaliai apkraunant turimą aparatūrinę įrangą (tikslas – taip subalansuoti OpenCL ir Python darbininkų kiekius, kad bendras programos vykdymo laikas būtų kuo trumpesnis).
- Abiejų programų kodui naudoti kodo kokybės įrankius ir įsitikinti, kad jie nerodo įspėjimų. Rekomenduojami įrankiai: CLion ir Pycharm aplinkų standartiniai įrankiai, clang-tidy, cpplint, pylint, flake8, mypy, ReSharper papildinį Visual Studio aplinkai (pasirinkti po vieną C++ ir Python). Galima naudoti ir studento pasirinktus įrankius.

- Programos veikimas ir realizacija — 20 taškų.
- Atsakymai į dėstytojo klausimus apie programą — 5 taškai.
- Programos apiforminimas (komentarai, kodo tvarkingumas) — 5 taškai.
- Atsakymas žodžiu į dėstytojo pasirinktą klausimą iš egzamino medžiagos — 5 taškai.