Tema de casă #2 – MPI

Responsabil de temă: Mihail Ionescu (mihai.ionescu@cs.pub.ro)

Data publicării: 07.11.2012
Data ultimei modificări a enunţului: 07.11.2012
Termenul de predare: 24.11.2012

Obiective

După realizarea acestei teme de casă studentul va fi capabil să:

- realizeze o aplicaţie bazată pe MPI.
- gestioneze eficient o structură de date mai complicata.

Cunoştinţele necesare rezolvării acestei teme de casă:

- Programare în C, structuri..

Enunţul problemei

Realizaţi o aplicaţie de tip *Map/Reduce* bazată pe MPI. Intr-o prezentare simplificata, Map/Reduce este o paradigma generala de programare in sisteme distribuite, in care operatia care trebuie executata sa imparte in doua faze. Faza 1, Map, in care un set de procese (numit mappers) imparte datele de intrare in parti egale. Fiecare mapper va trimite catre un set de procese numit reducers datele pentru care este repsonsabil. Fiecare proces reduce implementeaza semantica aplicatiei si returneaza datele procesului mapper care le-a apelat. Procesele mapper vor agrega datele si vor trimite inapoi catre procesul principal datele agregate. La randul lui, procesul principal agregheaza toate aceste date si prezinta rezultatul final.

Concret, aplicatia va trebui sa numere cuvintele dintr-un text mare si sa prezinte pentru fiecare cuvant de cate ori apare, ordonat dupa frecventa. Procesul principal va citi un fisier de configurare care va specifica numarul de procese mapper disponibile si pentru fiecare mapper, cate procese reduce. Apoi, fiecare proces map va lua partea corespunzatoare a fisierului de care este responsabil, va crea in mod dinamic procesele reduce si va trimite fiecarui proces reduce datele pentru care responsabil. Fiecare process reduce va calcula frecventa cuvintelor pe partea sa de date si va returna aceste date, procesului mapper parinte. Acesta agregheaza toate aceste date si le trimite inapoi la procesul parinte, care le agregheaza din nou, le sortateaza dupa freceventa si le prezinta ca output intr-un fisier.

Fisierul de input poate fi orice text. Testele vor trebuie facute cu un text mare, http://www.gutenberg.org/files/2600/2600.txt. Fisierul de output trebuie sa fie de forma:
cuvant tab frequency pe fiecare line, sortate dupa frequency.

Fisierul de configurare trebuie sa specifice numarul de mappers, numarul de reducers pentru fiecare mapper, numele fisierului de intrare si numele fisierului de iesire. Se presupune ca fiecare proces mapper are access la fisierul de intrare.

Toate testele vor fi facute pe un calculator Ubuntu 11.04 cu MPICH2. Tema va contine rezultatul masuratorilor (timpul de executie) variind numarul de mappers de la 2 la 4 din 1 in 1 si numarul de reducers de la 4 la 10 din 2 in 2.

IMPORTANT: Notarea se va face pe curba, unde programele care merg cel mai repede vor lua punctaj maxim, iar punctajul va scadea pe masura ce performantele scad. Timpii considerati vor fi aceia obtinuti pe masina de test.
In cazul in care programul nu merge (sau merge foarte prost) pe calculatorul de test, studentul poate arata ca merge pe calculator personal (laptop, etc). In acest caz, se vor scadea 10 puncte.
Tema va putea fi predata numai o saptamana dupa deadline, caz in care se vor scadea 5 puncte pe zi. Dupa o saptamana nu mai poate fi predata.

Fisierele care contribuie la rezolvarea temei trebuie **OBLIGATORIU** împachetate într-o arhiva de tip '**.zip**', cu numele 'Grupa_NumePrenume_TemaX.zip' (de exemplu, studenta Stan Sonia de la grupa 341C3 va trimite pentru tema 2 o arhiva cu numele 341C3_StanSonia_Tema1.zip). Aceasta arhiva trebuie OBLIGATORIU sa conţină şi un fişier text README în care se explică soluţia considerata în rezolvarea problemei.