

Wiki předmětu

14. dubna 2020

01:49:10

Login xharmi00, **Harmim Dominik**, 1. ročník NVER, prezenční, FIT
Ak. rok 2019/2020 studium č.2: změna zařazení

[Zpět na termíny.](#)

[[PRL Home]]

Navigace

- [Hlavní stránka](#)
- [Poslední změny](#)
- [Nápověda](#)

Stránka

Nástroje

- [Odkazuje sem](#)
- [Nahrát soubor](#)
- [Seznam souborů](#)
- [Seznam stránek](#)
- [Historie stránky](#)

Stránka

Implementace algoritmu "Odd-even transposition sort"

Pomocí knihovny Open MPI implementujte v jazyce C/C++ algoritmus Odd-even transposition sort tak, jak byl uveden na přednášce PRL.

Obsah

- 1 [Deadline](#)
- 2 [Vstup](#)
 - 2.1 [Soubor numbers](#)
- 3 [Výstup](#)
 - 3.1 [Příklad výstupu](#)
- 4 [Postup](#)
- 5 [Implementace](#)
 - 5.1 [Doplňkové informace](#)
- 6 [Dokumentace](#)
- 7 [Odevzdání](#)
- 8 [Hodnocení](#)

Deadline

27. 3. 2020

Vstup

Vstupem je posloupnost náhodných čísel uložená v souboru.

Soubor numbers

Soubor *numbers* obsahující čísla velikosti 1 byte, která jdou bez mezery za sebou. Pro příklad vytvoření tohoto souboru prostudujte soubor *test* (sekce [ukázkové zdrojové kódy](#)), ve kterém je ukázáno vytvoření takovéto posloupnosti náhodných čísel a její uložení do souboru pomocí utility *dd*. Tato utilita generuje náhodná čísla v rozsahu určeném velikostí bloku. Při bloku 1B jsou hodnoty v rozsahu 0-255. Vygenerovaná čísla jsou pak přesměrována do souboru. Vznikne tedy další soubor s náhodnými znaky jdoucími bez mezery za sebou. Po otevření v libovolném textovém editoru se hodnoty tváří jako náhodné ascii znaky, které by však měly být chápány jako celá čísla. Soubor je v tomto případě chápán jako binární.

Výstup

Výstup na *stdout* se skládá ze dvou částí:

1. Jednotlivé načtené neseřazené hodnoty v jednom řádku oddělené mezerou (vypsat po načtení prvním procesorem).
2. Jednotlivé seřazené hodnoty oddělené novým řádkem (od nejmenšího po největší).

Příklad výstupu

```
4 68 1 54
1
4
54
68
```

Postup

Vytvořte testovací skript *test*, který bude řídit testování. Tento skript bude mít tyto vlastnosti:

- Bude pojmenován *test* nebo *test.sh*.
- Bude přijímat pouze jeden parametr *pocet_hodnot*.

Skript vytvoří podle velikosti parametru *pocet_hodnot* soubor *numbers* s náhodnými čísly a následně spustí program se správným počtem procesorů. Skript nakonec smaže vytvořenou binárku a soubor *numbers*. Vzhledem ke strojové kontrole výsledků se v odevzdané verzi kódu **nebudou vyskytovat žádné jiné výstupy** než uvedené a ze stejných důvodů je třeba dodržet výše uvedené body týkající se testovacího skriptu. Za nedodržení těchto požadavků budou strhávány body.

Implementace

Algoritmus implementujte v jazyce C/C++ pomocí knihovny Open MPI.

Doplňkové informace

Nezapomeňte ošetřit všechny možné případy při výpočtu počtu procesorů.

Dokumentace

Součástí řešení je dokumentace, která bude o rozsahu **maximálně 3 strany** (rozumné a odůvodněné překročení limitu stran není důvod k bodové srážce) funkčního textu.

Do dokumentace **nedávejte**:

- Úvodní stranu, obsah, popis zadání.

V dokumentaci **popište**:

- Rozbor a analýzu algoritmu, odvodte jeho teoretickou složitost (časovou a prostorovou složitost, celkovou cenu). Uvedené vzorce slovně popište, včetně jejich proměnných. Uvedte také, proč cena tohoto algoritmu není optimální.
- Implementaci.
- Experimenty s různě velkými vstupy pro ověření časové složitosti (očekává se graf, nikoliv tabulka), nikoliv měření počtu kroků algoritmu.
- Komunikační protokol, jak si "procesy" zasílají zprávy. Pro vizualizaci použijte sekvenční diagram (http://en.wikipedia.org/wiki/Sequence_diagram). Nezapomeňte, že protokol musí být obecný, tedy pro n procesů.
- Závěr, ve kterém zhodnotíte dosažené výsledky, zamyšlení, zdali experimenty potvrdily teoretickou složitost, případně vysvětlení, proč tato teoretická složitost nebyla dodržena.
- Pozor, hodnotí se i to, jak dokumentace působí! (Zejména vzhled, správná čeština/slovenština/angličtina.)

Knihovna Open MPI je navržena na praktické použití, proto neobsahuje žádné implicitní metody pro měření složitosti algoritmů. Je tedy třeba vymyslet nějaký explicitní způsob jejího měření a ten pak co nejdětailněji popsat v dokumentaci!

Odevzdání

Do wisu se odevzdává jeden archiv `xlogin00.{tar|tgz|zip}`, který bude velký do 1MB, a který obsahuje:

- zdrojový kód- `ots.{c|cpp}`,
- hlavička- `ots.h` (pokud ji využijete),
- testovací shellový skript- `{test|test.sh}`,
- dokumentaci- `xlogin00.pdf`,
- nic jiného...

Platí, že kvalita každé z částí vzhledem k požadavkům má vliv na bodové ohodnocení . Počítejte s tím, že veškerá uvedená jména souborů jsou *case sensitive*.

Hodnocení

Hodnocení konzultujte emailem se svým opravujícím. V případě nutnosti je možné domluvit osobní konzultaci.