

Paralelní a distribuované algoritmy – dokumentace

Pipeline merge sort

Bc. Jaroslav Sendler, xsendl00
xsendl00@stud.fit.vutbr.cz

31. března 2012

Dokumentace k 1.projektu do předmětu Paralelní a distribuované algoritmy (PRL). Obsahuje popis zadání, rozbor a analýzu algoritmu Pipeline merge sort. V závěru dokumentu se nachází komunikační protokol mezi „procesory“ (způsob zasílání zpráv). Pro vizualizaci je využit sekvenční diagram.

Zadání

Pomocí knihovny Open MPI implementujte algoritmus **Pipeline merge sort**.

Vstup: Soubor „numbers“ obsahující čísla velikosti 1 byte, která jdou bez mezery za sebou.

Výstup: Výstup na *stdout* se skládá ze dvou částí:

- Jednotlivé načtené hodnoty v jednom řádku oddělené mezerou (vypsát po načtení prvním procesorem).
- Jednotlivé seřazené hodnoty oddělené novým řádkem (od nejmenšího po největší).

Postup: Vytvořte testovací skript *test*, který bude řídit testování. Skript bude mít následující vlastnosti:

- Bude pojmenován *test* nebo *test.sh*.
- Bude přijímat 1 parametr a to *pocet_hodnot*.

Skript vytvoří podle velikosti parametru *pocet_hodnot* soubor *numbers* s náhodnými čísly a následně spustí program s počtem procesorů $\log_2(\text{pocet_hodnot}) + 1$. Skript nakonec smaže vytvořený binární soubor 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.

Rozbor a analýza algoritmu

Algoritmus Pipeline merge sort pracuje s lineárním polem procesorů $p(n) = \log n + 1$, kde n je počet prvků určených k seřazení a $+1$ značí první procesor, který načítá vstupní data.

Základní body algoritmu: • Data nejsou uložena v procesorech, ale postupně do nich vstupují.

- Každý procesor spojuje dvě seřazené posloupnosti délky 2^{i-2} .
- Procesor P_i začne, když má na jednom vstupu posloupnost délky 2^{i-2} a na druhém 1, tedy začne $2^{i-2} + 1$ cyklů po procesoru P_{i-1} .
- P_i začne v cyklu

$$1 + \sum_{j=0}^{i-2} 2^j + 1 = 2^{i-1} + i - 1$$

- P_i skončí v cyklu $(n - 1) + 2^{i-1} + i - 1$.
- Celý algoritmus skončí po $2n + \log n - 1$ cyklech.

Slovní popis algoritmu

Teoretická složitost algoritmu

časová složitost: $2n + \log n - 1$ cyklů, kde n je počet prvků k seřazení
tedy $t(n) = O(n)$

cena: $t(n) \cdot p(n) = O(n) \cdot (\log n + 1)$, kde $p(n)$ je počet „procesorů“
tedy $c(n) = O(n * \log n)$, což je optimální

Naměřené hodnoty

V tabulce ?? jsou zobrazeny výsledky naměřené při experimentování s posloupnostmi různých délek.

Komunikační protokol

BITMAPOVÁ HLAVIČKA

	32 bit	Popis Pole
0		Typ souboru (0x4D42) - BM
1		Velikost souboru
2		Rezervováno 1 a 2 (0x00) (0x00)
3		Offset počátku bit. dat

INFORMAČNÍ HLAVIČKA

	32 bit	Popis Pole
0		Velikost hlavičky (0x28)
1		Šířka obrázku (px)
2		Výška obrázku (px)
3		Počet rovin (0x01), bity na pixel (0x08)
4		Kompresní metoda (0x00) - není použita
5		Velikost bitmapy
6		X pixel na metr (0x0B13)
7		Y pixel na metr (0x0B13)
8		Počet barev v obrázku (0x00)
9		Počet důležitých barev v obrázku (0x00)

Obrázek 1: Přehled hlaviček obrázku formátu BMP verze 3.