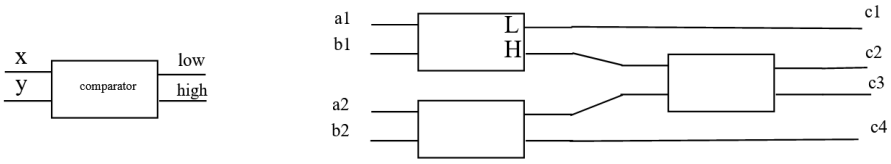
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Odd-Even Merge Sort

Bc. Vít Barták (xbarta47)

I. Rozbor a analýza algoritmu Odd-Even Merge Sort

Algoritmus *Odd-Even Merge Sort* je ze třídy paralelních řadících algoritmů, který pro svou činnost využívá speciální řadící síť procesorů. Každý procesor v síti funguje jako komparátor, který má dva vstupy a dva výstupy (low a high). Procesor umí porovnat vstupy a větší z nich dá na výstup high a menší z nich pak na výstup low (znázorněno na obrázku 1). Tyto výstupy slouží jako vstupy dalším procesorům v síti. Pro příklad: seřazené posloupnosti A={a1, a2} a B={b1, b2} jsou sítí 2x2 seřazeny v posloupnost C={c1, c2, c3, c4}. Pro více čísel se sítě rekurzivně skládají v kaskádu sítí N/2 x N/2. Informace o analýze algoritmu jsem čerpal ze čtvrté kapitoly z [1].



Obrázek 1 – Řadící síť 1x1 a 2x2

i) Časová složitost algoritmu: Vzhledem k tomu, že se velikost spojené posloupnosti zdvojnásobuje při průchodu každým stupněm sítě, máme celkem log n stupňů. Označme $s(2^i)$ jako čas, který potřebuje i-tá fáze pro spojení dvou posloupností, kde každá obsahuje 2^{i-1} prvků. Poté máme pro výpočet času předpisy:

$$s(2) = 1 \qquad pro\ i = 1$$
$$s(2^i) = s(2^{i-1}) + 1 \quad pro\ i > 1$$

Řešením je $s(2^i) = i$. Tudíž čas potřebný pro běh algoritmu (setřídění posloupnosti o n prvcích) je:

$$t(n) = \sum_{i=1}^{\log n} s(2^i) = O(\log^2 n)$$

Což je několikrát rychlejší, než O(n log n) sekvenčního Quicksortu.

ii) Prostorová složitost: Označme $q(2^i)$ jako počet komparátorů potřebných v i-té fázi pro spojení dvou posloupností, kde každá obsahuje 2^{i-1} prvků. Pro výpočet máme předpisy:

$$q(2) = 1 \qquad pro\ i = 1$$
$$q(2^i) = (i-1)2^{i-1} + 1 \quad pro\ i > 1$$

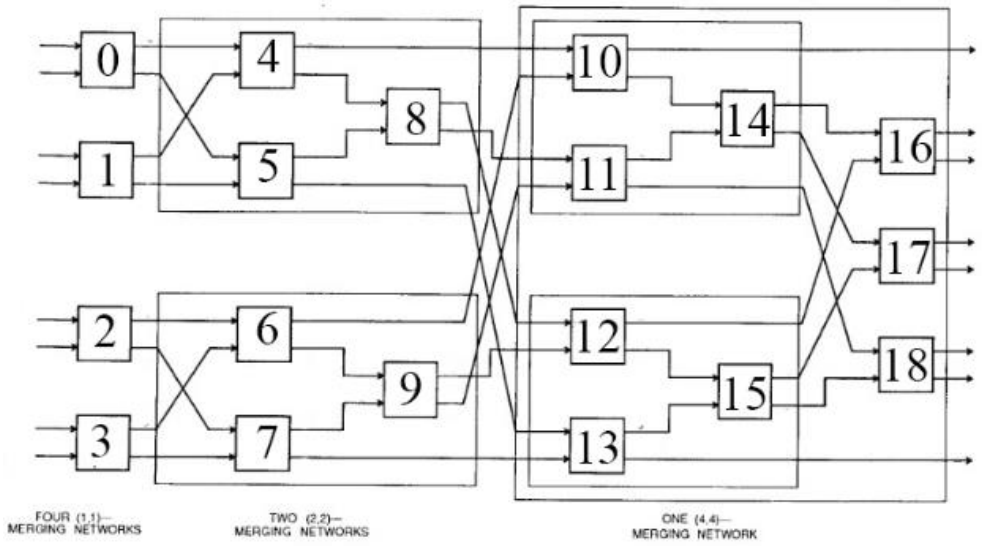
Kde řešením je $q(2^i) = (i-1)2^{i-1} + 1$. Tudíž počet komparátorů (procesorů) potřebných pro běh algoritmu je

$$p(n) = \sum_{i=1}^{\log n} 2^{(\log n)-i} q(2^i) = O(n \log^2 n)$$

iii) Cena: $c(n) = t(n) * p(n) = O(n \log^4 n)$ Odd-Even Merge Sort tudíž nemá optimální cenu, jelikož je větší než cena sekvenčního Quicksortu. OEMS je velice rychlý, ale vyžaduje velké množství komparátorů.

II. Implementace

Přiložený C++ program *oems.cpp* implementuje Odd-Even Merge Sort zásadně podle schématu na obrázku 2 pro řazení osmi náhodných čísel. Čísly 0 až 18 je označeno 19 komparátorů, kterých je potřeba pro řazení. Síť sestává z čtyř 1x1 sítí napojených na dvě 2x2 sítě napojených na jednu 4x4 řadící síť.



Obrázek 2 - Schéma zapojení sítě pro 8 čísel [1]

Ve funkci *main()* je ze všeho nejdřív inicializována knihovna OpenMPI. Následně proces s číslem 0 (master) načte funkcí *readNumbersFile*() 8 náhodných čísel v rozmezí 0-255 do vektoru *in_numbers* a rozešle pomocí *MPI_Isend()* procesům 0 až 3 každému dvě čísla (X a Y). Všechny procesy 0 až 3 přijmou čísla od mastera, porovnají je (uloží si je jako hodnoty L a H), následně je pošlou dále na výstup do sítí 2x2. Tento proces se opakuje pro každý komparátor – přijme 2 hodnoty, porovná je a odešle svému následníkovi.

Na základě hodnoty ranku každého procesu se rozhoduje, kdo je jeho předchůdcem (sourceX, sourceY) v síti a kdo je příjemcem jeho L a H hodnot (receiverL, receiverH). Tímto způsobem simuluji síť znázorněnou na obrázku 2. Za použití *MPI_Isend()* a *MPI_Irecv()* takto prostupuji všemi fázemi sítě a na konci jsou sesbírány seřazené hodnoty do master procesoru, který je poté vytiskne na stdout.

III. Závěr

Algoritmus Odd-Even Merge Sort byl úspěšně naimplementován za pomocí knihovny OpenMPI. V průměru tato implementace běží 6 μs, což bylo zjištěno opakovaným testováním. Teoretickou časovou složitost není možno pro takto nízký počet (8) čísel efektivně ověřit.

IV. Zdroje

[1] AKL, Selim G. *The design and analysis of parallel algorithms*. New Jersey: Prentice-Hall, 1989. ISBN 01-320-0056-3.