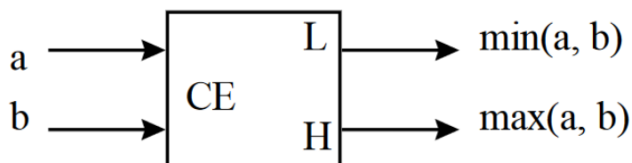
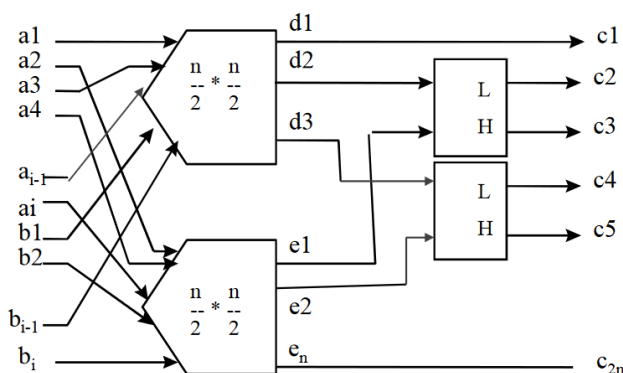


Odd-even merge sort

Jedná se paralelní řadící algoritmus využívající speciální síť procesorů. Základní jednotkou této sítě je element CE, která dokáže vzít dvě hodnoty a na jeden ze svých výstupů vrátit menší z nich a na druhý větší. Tato samotná jednotka tvoří síť 1x1.



Tyto elementární jednotky se skládají do větších sítí 2x2, 4x4, 8x8 (a další násobky dvou), které když se zapojí za sebou do úrovní, tak umožňují seřadit číselnou posloupnost libovolné velikosti - ideálně však velikosti násobku dvou. Větší síť se tvoří dle následujícího schématu:

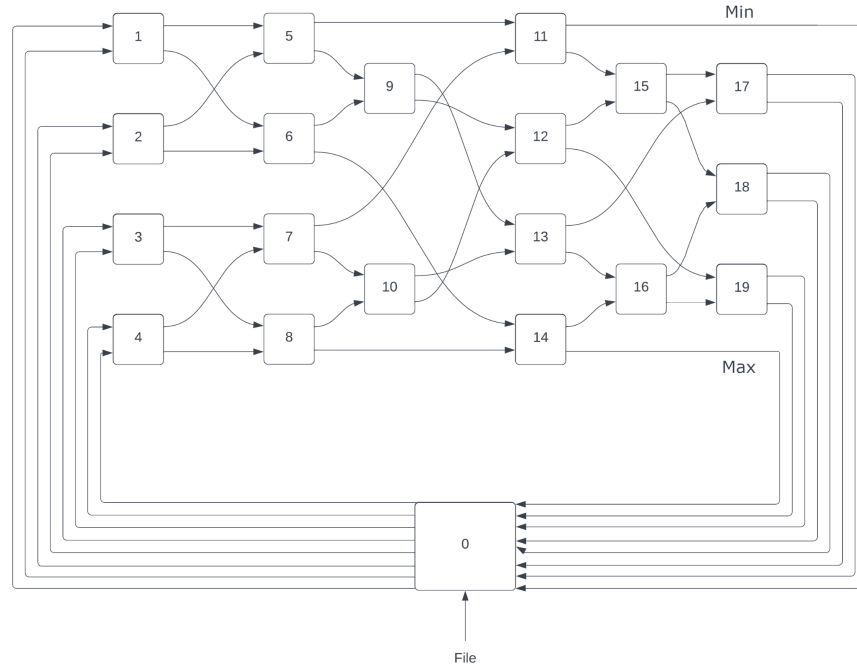


Podle velikosti posloupnosti k seřazení n lze určit množství potřebných úrovní m následovně: $n = 2^m$. Každá úroveň využívá jiný počet procesorů zapojených do sítě odpovídající velikosti.

- 1. úroveň využívá 2^{m-1} sítí 1x1.
- 2. úroveň využívá 2^{m-2} sítí 2x2, kde každá má 3 procesory.
- 3. úroveň využívá 2^{m-3} sítí 4x4, kde každá má 9 procesorů.
- 4. úroveň využívá 2^{m-4} sítí 8x8, kde každá má 25 procesorů
- atd.

Kvůli řazení do úrovní je časová složitost $t(n) = O(m^2) = O(\log^2 n)$ a tedy cena tohoto algoritmu je $c(n) = O(n \cdot \log^4 n)$. Dá se tedy říct, že tento algoritmus není optimální

Implementace



Protože je seřazováno osm čísel, bylo třeba vytvořit síť o třech úrovních.

- 1. úroveň - procesory 1, 2, 3, 4 tvořící čtyři sítě 1x1
- 2. úroveň - procesory 5, 6, 7, 8, 9, 10 tvořící dvě sítě 2x2
- 3. úroveň - procesory 11, 12, 13, 14, 15, 16, 17, 18, 19 tvořící jednu síť 4x4

Celkem 19 procesorů pro třídění. Dále jsem použil procesor 0 pro načtení čísel ze souboru, rozslání vstupních hodnot počátečním procesorům a sesbírání výsledné seřazené posloupnosti od konečných procesorů. Tedy celkem v mé implementaci využívám 20 procesorů. Schéma komunikace/propojení procesorů je znázorněno na diagramu. Samotná čísla v síti zasílám pomocí MPI_Send a přijímám s MPI_Recv.

Jednotlivé procesory nejříve odešlou menší ze vstupních čísel a až potom větší. Toto napomáhá procesoru 0 správně určit pořadí čísel jdoucích od jednoho konečného procesoru

Závěr

Tento algoritmus extrémně plýtvá procesory pro seřazení jedné řady čísel. Pokud by se tento algoritmus měl někdy doopravdy využít, navrhol bych několik procesů sloučit do jednoho, protože při řazení osmi čísel nikdy nebudou pracovat více jak čtyři procesory. Např. sloučení

- 1. procesor z 1, 5, 9, 11
- 2. procesor z 2, 6, 12, 15, 17
- 3. procesor z 3, 7, 13, 18, 19
- 4. procesor z 4, 8, 10, 14

Pro zefektivnění by také bylo možné posílat k řazení více řad čísel za sebou, aby všechny úrovně pracovali zároveň a nedocházelo k situaci, kdy každý procesor většinu času průběhu algoritmu nic nedělá.