Odd-even transposition sort

Matej Karas

Rozbor

Algoritmus je navrhnutý pre N procesorov, z ktorých každý vlastní jedno číslo a v kroku Odd sa každý nepárny procesor p_i spojí s párnym p_{i+1} a porovnajú svoje hodnoty. Procesory si vymenia svoje hodnoty pokiaľ hodnota procesoru p_i je väčšia ako hodnota procesoru p_{i+1}. V kroku Even sa každý párny procesor p_i spojí s nepárnym p_{i+1} a porovnajú svoje hodnoty. Procesory si vymenia svoje hodnoty pokiaľ hodnota procesoru p_i je väčšia ako hodnota procesoru p_{i+1}.
Je zrejmé, že algoritmus bude efektívny, len pri veľkom množstve čísel a teda veľkom množstve procesorov, čo pre bežné použitie nie je vhodné o čom budú svedčiť aj namerané výsledky.

Teoretická zložitosť:

Časová	$O(n/2) + O(n/2) = O(n)$
Priestorová	$O(n)$
Celková cena	$O(n) * O(n) = O(n^2)$

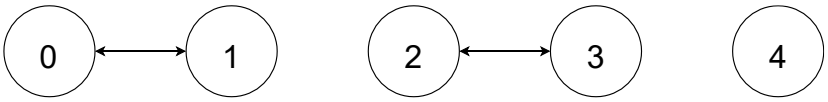
Časová zložitosť vyplýva z cyklu, v ktorom sa n/2 krát porovnávajú čísla pri Even kroku a n/2 krát pri Odd kroku. Z toho plynie, že časová zložitosť je $O(n)$.
Priestorová vyplýva z toho, že každý procesor uchováva len jedno číslo, a teda zložitosť je $O(n)$.
Celková cena je potom *priestorová * časová* zložitosť a teda $O(n^2)$. Táto zložitosť nieje optimálna pretože s narastajúcimi počtom prvkov stúpa kvadraticky a teda pre väčšie množstva nie je použiteľný.

Implementácia

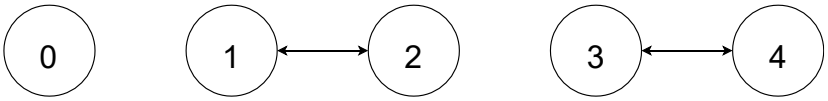
Úlohu som implemenoval dvomi spôsobmi – vytvorením virtuálnych topológií a klasickým Send/Recieve, pretože virtuálne topológie sa ukázali byť nevhodné pre danú úlohu a tvorba grafu zaberala príliš mnoho času.
Pri oboch variantách na začiatku najprv načíta, tzv. root proces (proces, ktorý má rank 0) súbor s číslami, ktoré následne rozošle ostatním rankom funkciou ***MPI_Scatter***. Následne prebieha radiaci algoritmus, tak ako bol popísaný v prednáškach. Po zoradení, sú čísla opäť zozbierané funkckiou ***MPI_Gather*** a následne vypísané na stdout.
V scripte *test.sh* je následne možné zvoliť, ktorá varianta sa má použiť alebo spustenie testovacej varianty programu, kedy sa budú vypisovať časi jednotlivých sekcií, pomocou ktorých boli uskutočnené experimenty.

Virtuálne topológie

V tejto variente je vytvorený ***MPI_Comm*** pre výmenu správ procesov pri predávaní správ pri Even a Odd kroku algoritmu. Topológia je vytvorená ako graf, ktorý je znázornený na obrázku. Komunikácia medzi procesmi bude popísaná v ďalšej kapitole.



Even varianta



Odd varianta

Send/Recv

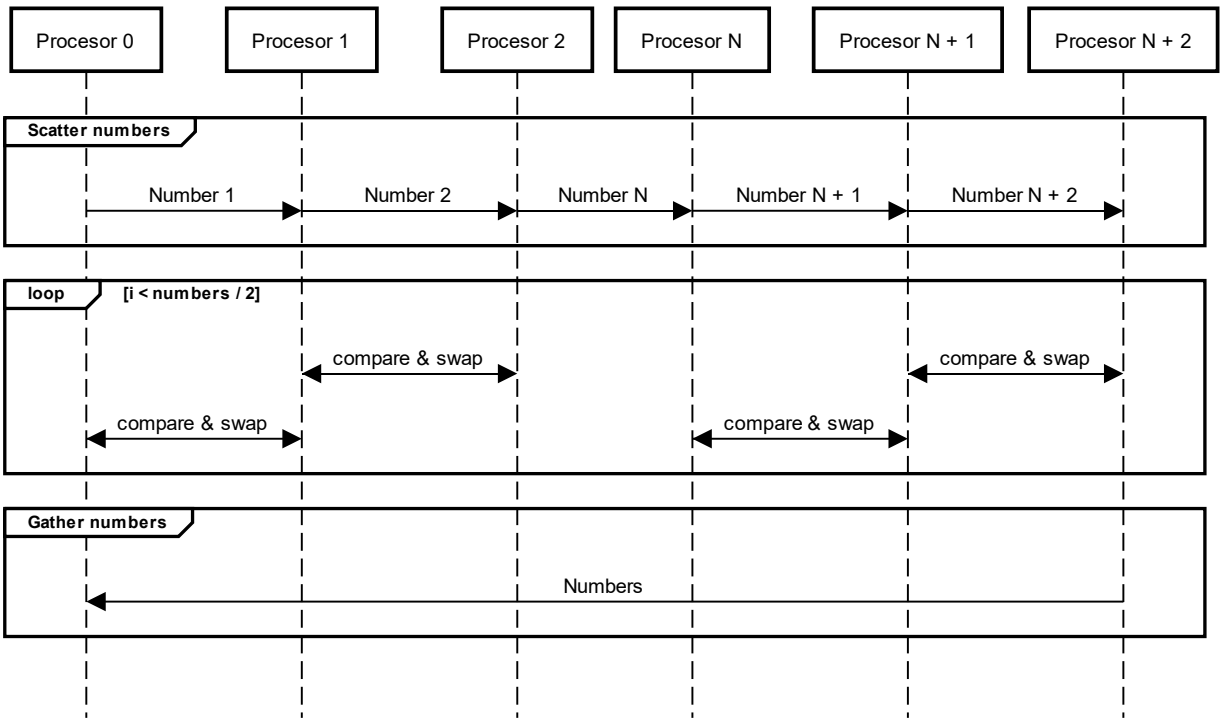
V tejto variante sú správi zasielané medzi procesmi funkciou ***MPI_Sendrecv***. Pokiaľ nejaký proces nezasiela žiadnu správu (napr. pri odd kroku je to proces 0), tak zašle správu cielu ***MPI_PROC_NULL***.

Experimenty

Experimenty boli uskutočnené na 4 typoch strojov, z toho 2 boli na Ostravskom superpočítači Salomon a Anselm. Namerané grafy sú priložené v sekcii Virtuálna topológia nebola vhodná z dôvodu, že jej vytvorenie je príliš časovo náročné pre danú úlohu. Ďalej z grafov virtuálnych topológií je vidieť, ako sa doba radenia extrémne zdivhne pri použití viacero procesov než je dostupných fyzických jadier. Je to z dôvodu vyplachovania cache procesoru a zároveň neustále prepínanie kontextu a réžia s ním spojená.

Grafy.

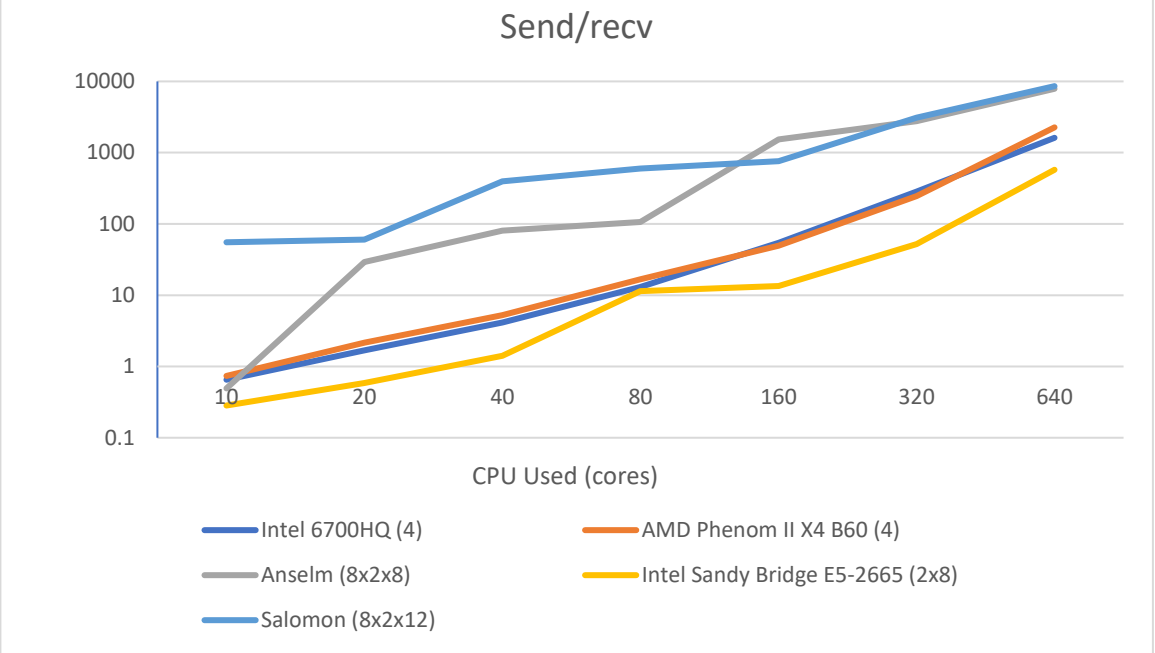
Komunikačný protokol

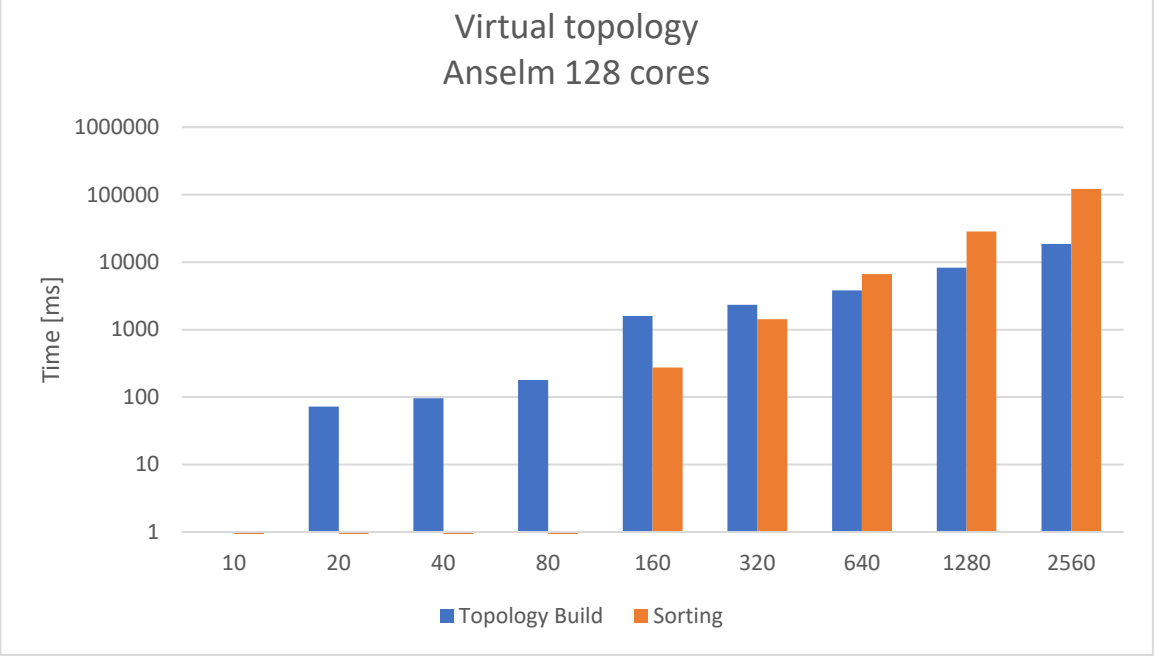


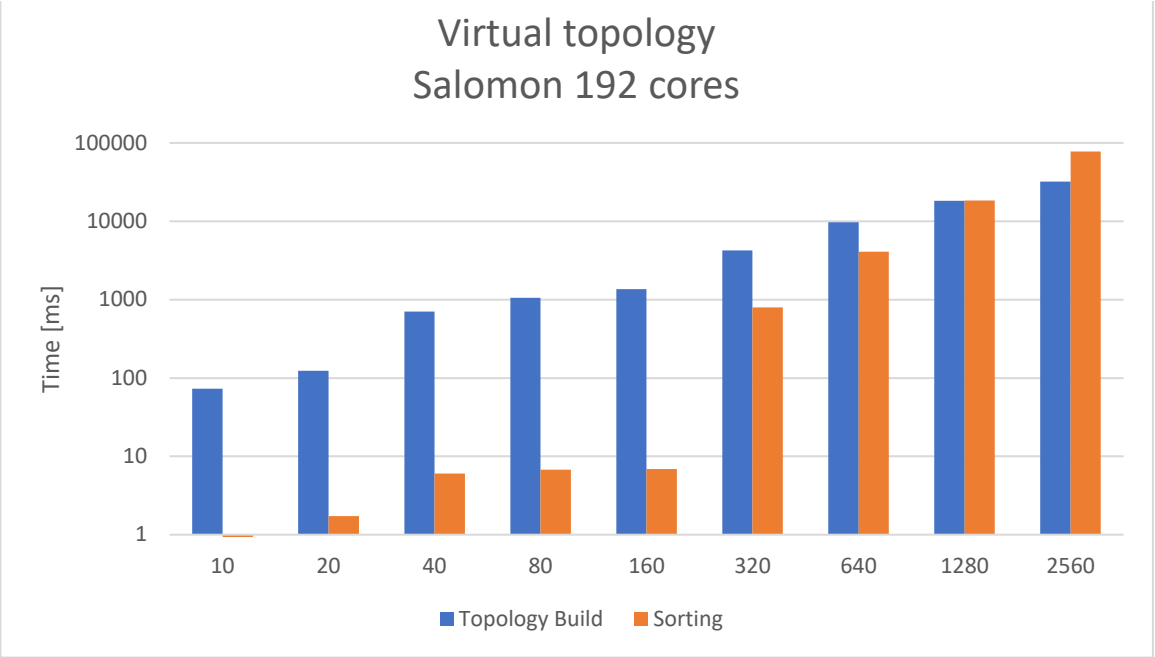
Záver

Teoretická zložitosť sa nedosiahla, pretože v nej nie je zarátaná réžia komunikácie medzi procesormi. Algoritmus je neefektívny na multiprocesorovom systéme, resp. clustri, pretože réžia zasielania správ je príliš drahá. Algoritmus by mal byť efektívny len pri veľkom množstve procesorov a nepreťažovaní daného systému (vytvárania viac procesov, ako je dostupných fyzických jadier).
Virtuálna topológia nebola vhodná z dôvodu, že jej vytvorenie je príliš časovo náročné pre danú úlohu. Ďalej z grafov virtuálnych topológií je vidieť, ako sa doba radenia extrémne zdivhne pri použití viacero procesov než je dostupných fyzických jadier. Je to z dôvodu vyplachovania cache procesoru a zároveň neustále prepínanie kontextu a réžia s ním spojená.

Grafy


Send/recv
10000
1000
100
10
1
0.1
10
20
40
80
160
320
640
CPU Used (cores)
Intel 6700HQ (4)
AMD Phenom II X4 B60 (4)
Anselm (8x2x8)
Intel Sandy Bridge E5-2665 (2x8)
Salomon (8x2x12)


Virtual topology
Anselm 128 cores
1000000
100000
10000
1000
100
10
1
Time [ms]
10
20
40
80
160
320
640
1280
2560
Topology Build
Sorting


Virtual topology
Salomon 192 cores
Time [ms]
100000
10000
1000
100
10
1
10
20
40
80
160
320
640
1280
2560
Topology Build
Sorting