

## Projekt do předmětu PRL

Jméno: Petr Polanský

Login: xpolan07

### Analýza algoritmu Enumeration sort

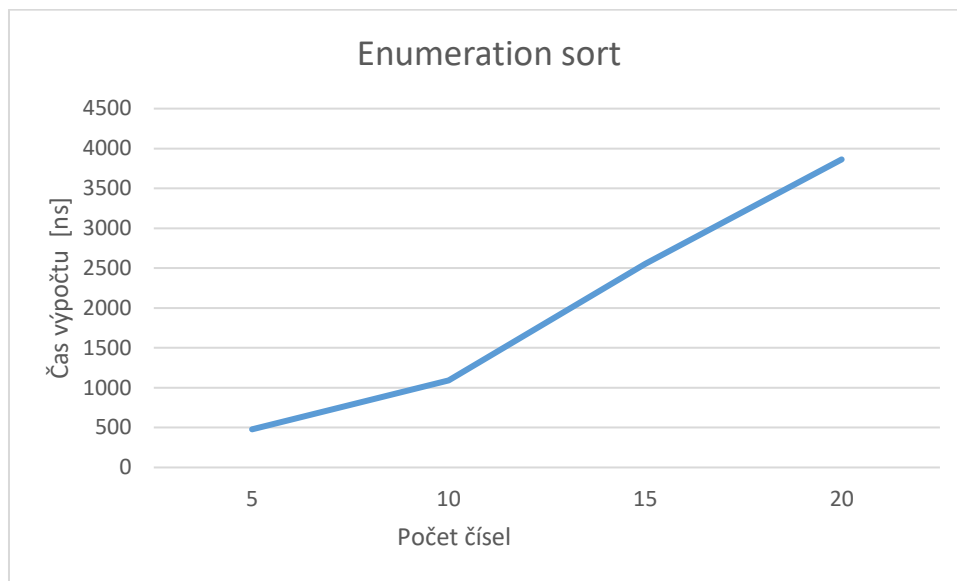
Provádění nastavení hodnoty registru C na hodnotu 1, probíhá v konstantním čase. Řazení jednotlivých čísel posloupnosti trvá  $2 * \text{počet\_procesorů}$  cyklů a následné vypisování seřazených hodnot probíhá v lineární čase závislém na počtu procesů. Uvažuje se, že přenos hodnoty sběrnici trvá konstantní dobu, bez ohledu na fyzickou vzdálenost procesorů. Počet procesorů odpovídá počtu řazených čísel ( $n$ ). Výsledná cena se tedy rovná časové složitosti ( $n$ ) \* počtu procesorů ( $n$ ), což je  $n^2$ .

### Implementace

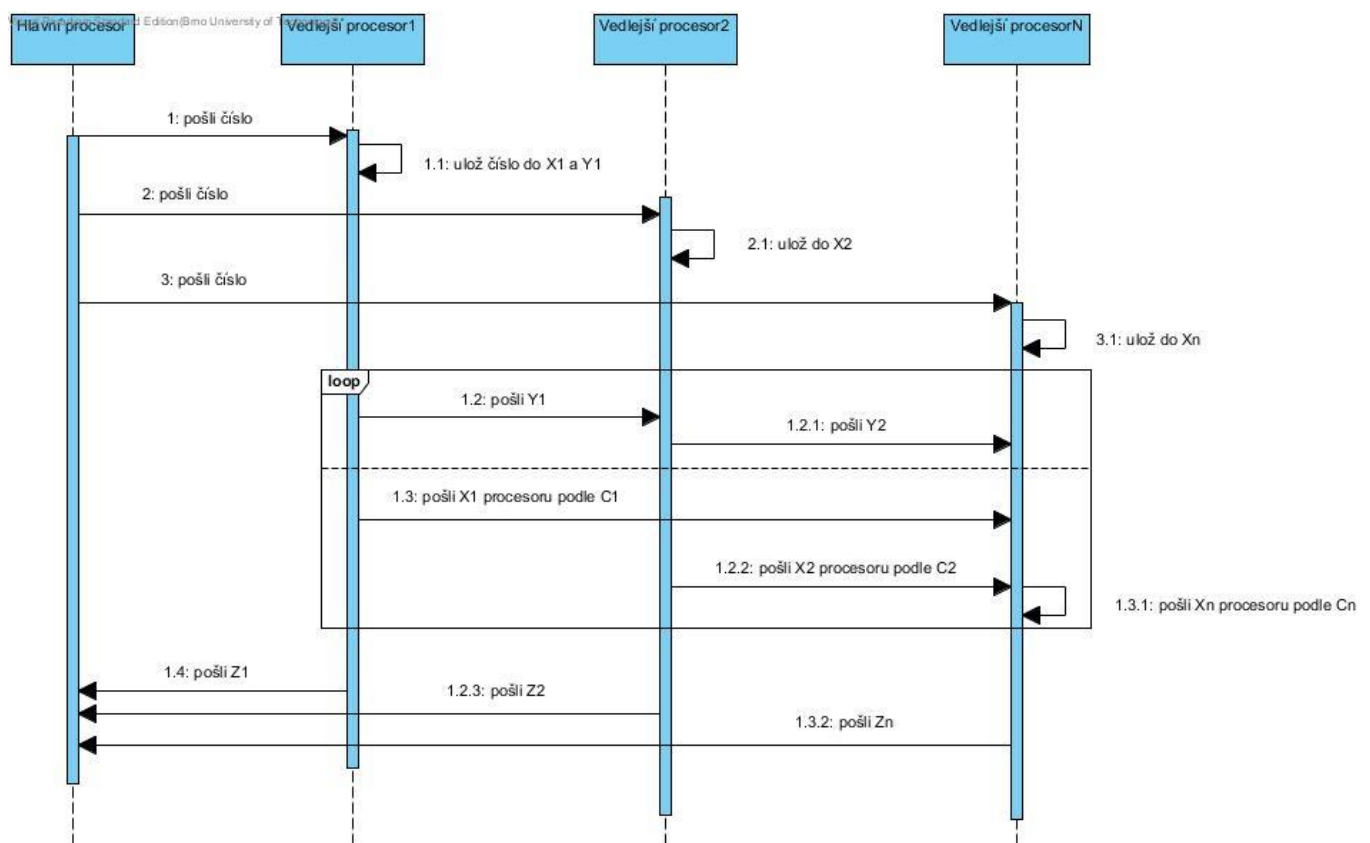
Pro vytvoření algoritmu Enumeration sort jsem použil knihovnu Open MPI. Jednotlivé procesory jsem vytvořil jako datovou strukturu s prvky C, X, Y, Z, které byly typu integer. Funkce `MPI_Comm_size` mi do proměnné `numprocs` uložila počet běžících procesorů. Funkce `MPI_Comm_rank` mi do proměnné `myid` uložila id právě obsluhovaného procesoru. Hlavní procesor získává náhodná čísla ze vstupu a posílá je společnou sběrnici ostatním procesorům do registrů X a prvnímu procesoru do registru Y. Pro posílací a přijímací funkce musí být nastaven příznak TAG, podle kterého každý procesor rozpozná jednotlivé zprávy pro různé registry. Každý procesor nejprve nastaví svůj registr C na hodnotu 1.

Poté v  $2 * \text{počet\_procesorů}$  cyklech probíhá řazení. Každý procesor očekává hodnotu registru Y od procesoru s nižším číslem. První procesor tedy očekává hodnotu od hlavního procesoru. Pokud nejsou registry X a Y prázdné, tak dojde k jejich porovnání, a pokud je  $X > Y$ , tak se inkrementuje registr C o 1. Aby nedošlo ke konfliktu při porovnání stejně velkých čísel, bylo nutné provést rozdělení tohoto porovnání pomocí algoritmu (Shuzo Yajima, 1982). Procesory dále pošlou svoji hodnotu registru Y registrům s vyšším číslem. Poté procesory pošlou svoje hodnoty registrů X registrům s pořadovým číslem, které odpovídá hodnotě v registru C. Tím dojde k seřazení čísel podle jejich hodnot. Nakonec procesory pošlou hodnoty z registrů Z řídicímu procesoru, který je vypíše.

## Graf časové složitosti



## Komunikační protokol



## Závěr

Výsledky měření neodpovídají předpokládané časové složitosti, z důvodu nedokonalé implementace algoritmu. Čísla ze souboru po přečtení jsou ihned posílána hlavním procesorem ostatním procesorům. Výsledná časová složitost je tedy ovlivněna přístupem do souboru.