

# Tugas MPI

## 1. Tugas 1: point-to-point communication

Akan dilakukan komunikasi point-to-point (p2p) antar proses menggunakan MPI. P2P adalah komunikasi dari sebuah proses ke sebuah proses lainnya. Pada program `mpi_p2p_terkecil.py`, proses dengan rank=0 akan mengirimkan pesan ke proses rank 1, 2, ..., n. Pada program `mpi_p2p_terbesar.py` proses dengan rank terbesar akan mengirimkan pesan ke proses rank 0, 1,...,n-1.

- Download skeleton source yang diberikan!
- Lengkapi source code sehingga program bisa berjalan!
- Jalankan program pada komputer.
- Kumpulkan source code yang dibuat serta screenshot hasil program yang berjalan!

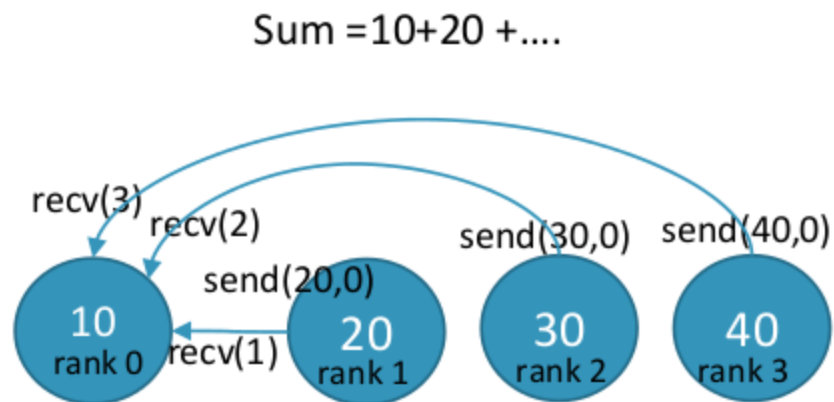
## 2. Tugas 2: broadcast communication

Dibuat komunikasi broadcast pada MPI. Broadcast adalah mengirimkan pesan kepada semua proses yang ada. Pada tugas ini, proses dengan rank=0 akan mengirimkan pesan ke semua proses yang ada.

- Download skeleton source yang diberikan!
- Lengkapi source code sehingga program bisa berjalan!
- Jalankan program pada komputer.
- Kumpulkan source code yang dibuat serta screenshot hasil program yang berjalan!

## 3. Tugas 3: penjumlahan MPI teknik p2p

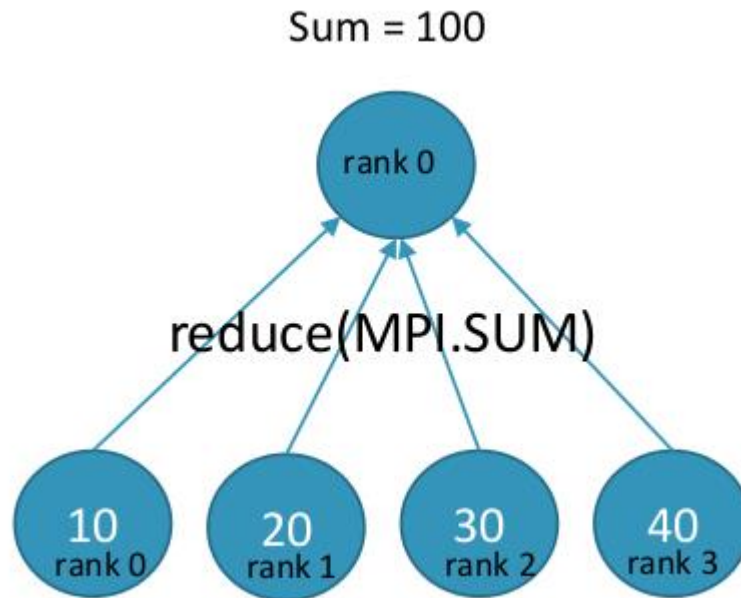
Pada tugas ini akan dibuat penjumlahan menggunakan teknik p2p. Pertama, pada setiap proses akan diberikan nilai secara random. Nilai-nilai dari setiap proses tersebut akan dijumlahkan menggunakan MPI. Kedua, proses dengan rank=0 akan bertindak mengumpulkan (menerima) nilai-nilai dari proses lain menggunakan komunikasi p2p. Ketiga, setelah nilai-nilai terkumpul maka proses dengan rank=0 akan menjumlahkan semua nilai dan menampilkannya.



- a. Download skeleton source yang diberikan!
- b. Lengkapi source code sehingga program bisa berjalan!
- c. Jalankan program pada komputer.
- d. Kumpulkan source code yang dibuat serta screenshot hasil program yang berjalan!

#### 4. Tugas 4: penjumlahan MPI teknik reduce

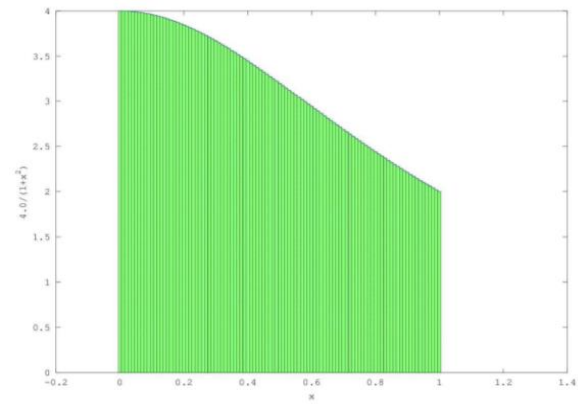
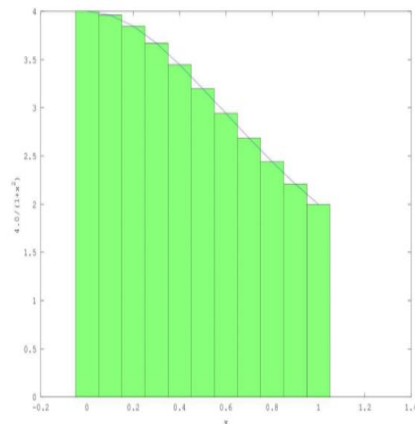
Sama seperti tugas 3 hanya teknik yang digunakan adalah teknik reduce yang ada pada MPI. Teknik ini lebih cepat dari teknik pada tugas 3. Pada tugas 3, hasil diperoleh secara sekuensial. Teknik reduce akan mempercepat komunikasi hasil.



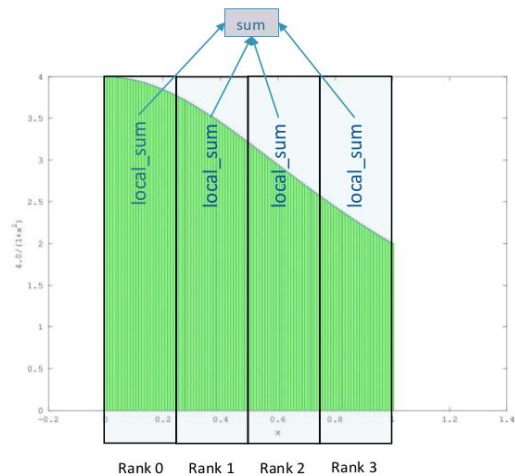
- a. Download skeleton source yang diberikan!
- b. Lengkapi source code sehingga program bisa berjalan!
- c. Jalankan program pada komputer.
- d. Kumpulkan source code yang dibuat serta screenshot hasil program yang berjalan!

#### 5. Tugas 5: menghitung nilai pi

Script `serial_pi.py` adalah script untuk menghitung pi secara serial menggunakan pendekatan  $4/(1+x^2)$ . Pada teknik ini, akan dihitung luas area di bawah kurva  $4/(1+x^2)$ . Menghitung luas kurva dapat didekati secara diskrit dengan menentukan banyaknya step yang digunakan. Semakin banyak step yang digunakan maka hasilnya akan lebih akurat tetapi proses akan sangat lama.



Script `mpi_pi.py` adalah bentuk parallel mpi dari versi `serial_pi.py`. Ide dari program ini adalah setiap proses akan mendapatkan bagian untuk menghitung hasil bagiannya masing-masing. Misalkan: -n 5 (ada 5 proses yang berjalan) dan terdapat 1000 step maka proses 0 bertugas mengerjakan bagian dari 0-199 step, proses 1 dari 200-399, dst



- Download skeleton      source      yang  
diberikan!
- Lengkapi source code sehingga program bisa berjalan!
- Jalankan program pada komputer.
- Kumpulkan source code yang dibuat serta screenshot hasil program yang berjalan!

**Happy coding!**