

## IF3230 Sistem Paralel dan Terdistribusi Praktikum 2 Penggunaan OpenMPI - Bucket Sort

### A. Persiapan Praktikum

Peserta dapat melakukan ssh ke salah satu dari 6 server berikut : 167.205.35.25 - 167.205.35.31 (kecuali 167.205.35.27).

Apabila peserta telah melakukan langkah-langkah praktikum 1 dengan baik, maka hanya perlu menjalankan file `.bashrc` dengan perintah :

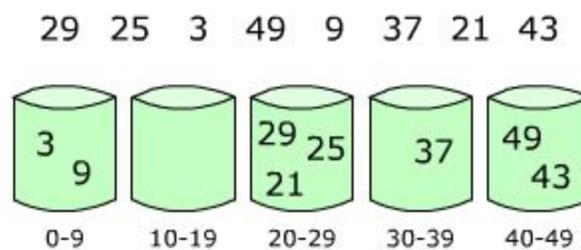
```
> . ~/.bashrc
```

Jika peserta belum melakukan langkah persiapan praktikum 1 dengan baik, maka dapat mengulangi langkah persiapan praktikum sesuai dengan file panduan praktikum 1.

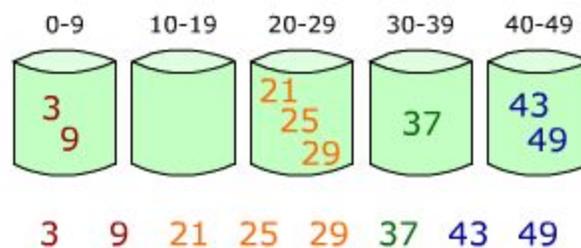
### B. Bucket Sort

Pada praktikum kali ini, setelah peserta mengerti mengenai fitur-fitur penting dari OpenMPI, peserta ditugaskan untuk mengimplementasikan algoritma sorting Bucket Sort. Algoritma ini mendistribusikan elemen-elemen dari suatu array yang sedang dilakukan proses pengurutan ke sejumlah *bucket-bucket*. Tujuan dari distribusi elemen-elemen ke tiap *bucket* adalah untuk mempercepat proses pengurutan karena tiap *bucket* akan melakukan pengurutan di dirinya sendiri sebelum digabungkan menjadi kembali menjadi satu array yang telah terurut.

Ilustrasi berikut menggambarkan bagaimana cara kerja Bucket Sort.



Gambar 1. Membagi Elemen-Elemen Dari Array Ke Bucket



Gambar 2. Tiap Bucket Mengembalikan Elemen Yang Telah Terurut

Dapat dilihat pada gambar 1 dan gambar 2 bahwa tiap-tiap *bucket* akan memperoleh bagiannya masing-masing sesuai dengan rentang nilai yang dimiliki per *bucket*. Oleh karena itu, harus ditentukan mekanisme untuk membagi elemen ke tiap-tiap *bucket*. Pada tiap-tiap *bucket* dapat dilakukan pengurutan seperti biasa (contoh: insertion sort) dan akhirnya tiap *bucket* mengembalikan nilai yang terurut. *Pseudocode* untuk Bucket Sort dapat dilihat dibawah ini.

```
// Copyright https://en.wikipedia.org/wiki/Bucket_sort
function bucketSort(array, n) is
  buckets ← new array of n empty lists
  for i = 0 to (length(array)-1) do
    insert array[i] into buckets[msbits(array[i], k)]
  for i = 0 to n - 1 do
    nextSort(buckets[i]);
  return the concatenation of buckets[0], ..., buckets[n-1]
```

Cara untuk membagi elemen array ke tiap-tiap *bucket* dan algoritma pengurutan per *bucket* dibebaskan kepada tiap-tiap peserta. Adapun referensi program untuk proses pengurutan pada tiap-tiap *bucket* dapat dilihat pada repository project.

### C. Pengumpulan & Deliverables

*Deliverables* yang harus dikumpulkan pada program ini adalah *source code* program anda serta laporan singkat yang berisi perbandingan waktu (*time performance*). Pengukuran performansi dilakukan pada bagian distribusi awal data ke tiap-tiap proses dan saat proses sorting dimulai hingga selesai. Adapun untuk kasus-kasus pengurutan adalah sebagai berikut (komputer yang digunakan 6 host):

1. N elemen pada array dengan jumlah proses yang digunakan M proses  
Nilai N: 50.000, 100.000, 200.000, 400.000  
Nilai M: 1, 4, 8, 16, 32

Mekanisme pengumpulan untuk tugas ini sama seperti praktikum 1, yaitu sebagai berikut:

- a. Lakukan fork pada repo di gitlab
- b. Lakukan git clone (boleh di server)
- c. Lakukan commit setelah selesai mengerjakan satu program
- d. Lakukan merge request dari repo hasil fork anda di akhir praktikum untuk penilaian
- e. Segala bentuk kecurangan akan mendapatkan konsekuensi