

CSCM603154 - Jaringan Komputer

Gasal 2020/2021

Tugas 3 (Kelompok) - Socket Programming - Job Management Systems

Release: Jumat, 9 Oktober 2020
Deadline: Minggu, 25 Oktober 2020

Note:

Mahasiswa diminta untuk membentuk kelompok secara mandiri beranggotakan 3 orang. Silahkan daftarkan nama kelompok dan nama anggota kelompok pada google sheet yang telah disediakan. Pendaftaran ini dilakukan untuk mempermudah asisten untuk pembagian tugas grading.

Intro

Meroketnya trend Data Science dan Big Data membuat infrastruktur Cloud Computing semakin sibuk dan banyak digunakan. Salah satu teknologi yang menopang kinerja Cloud adalah job management system yang mengatur penjadwalan jobs yang di submit untuk dieksekusi di lingkungan Cloud. Beberapa teknologi yang cukup populer diantaranya adalah (silahkan dipelajari mandiri):

1. YARN
2. Mesos
3. Oracle Grid Engine
4. HTCondor
5. PBS

Tugas job management system adalah menerima jobs dari user kemudian mencari alokasi sumber daya komputasi yang sesuai untuk mengeksekusi jobs tersebut. Tugas lainnya yang tidak kalah penting adalah memonitor jalannya eksekusi dan juga availabilitas sumber daya komputasi.

Deskripsi

Pada tugas kali ini, mahasiswa diminta untuk membuat job management system sederhana dengan spesifikasi sebagai berikut:

A. Arsitektur

Silahkan pelajari paradigma komputasi Master-Worker atau pola Map-Reduce yang biasa digunakan dalam komputasi Big Data. Mahasiswa diminta untuk mengimplementasikan paradigma ini dalam membuat program.

Sistem yang akan dibuat akan terdiri dari satu master node dan beberapa worker node(s). Master node berfungsi sebagai scheduler yang menerima jobs dari user kemudian mendistribusikannya ke worker node(s) yang tersedia.

Komunikasi hanya dapat dilakukan antara master node dengan worker node(s). Diantara worker node(s) tidak perlu saling berkomunikasi. *Silahkan cocokkan dengan arsitektur yang sudah dipelajari di kelas.*

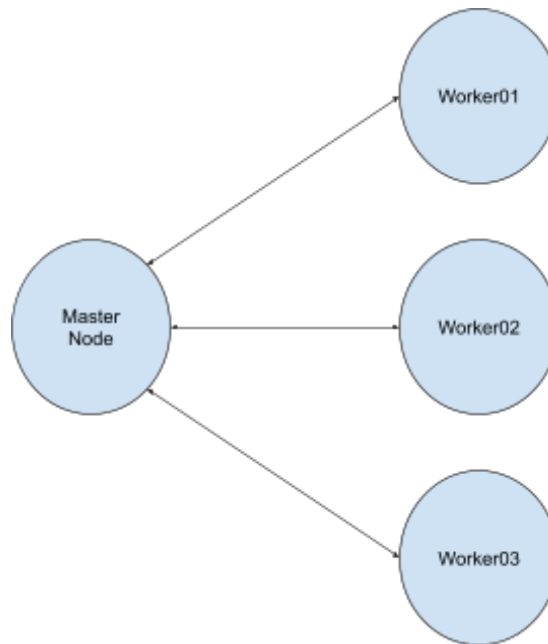


Fig: Master-Worker Paradigm

Skenario sederhana dari sistem yang akan dibuat memungkinkan master node menerima jobs dari users kemudian mengirimkan perintah kepada worker yang available untuk mengeksekusi job tersebut dan worker akan mengirimkan balik hasil eksekusi dari job tersebut kepada master.

Mahasiswa dibebaskan untuk menggunakan bahasa pemrograman apapun yang dapat mengimplementasikan fitur network communication seperti *sockets* dan *threads* dengan mudah (*rekomendasi: C, Java, Python*).

B. Komponen Penilaian

Berikut ini adalah fitur-fitur yang harus diimplementasikan berikut komponen penilaiannya

a. Fitur Dasar (wajib dikerjakan semua, total 60 poin)

- **[20 poin]** Adanya satu master node yang dapat menerima job dari user. Laptop/PC mahasiswa difungsikan sebagai Master node dan job dapat langsung di submit dari sini.

- **[20 poin]** Adanya (minimal) satu worker node yang dapat mengeksekusi job dari master. Virtual machine (EC2) difungsikan sebagai worker node(s).
- **[20 poin]** Adanya komunikasi antara master dan worker node(s). Master dapat mengirimkan perintah dan job kepada worker dan worker dapat mengirimkan balik hasil eksekusi kepada master.

*b. Fitur Elective (tidak perlu dikerjakan semua, **max 40 poin**)*

- **[10 poin]** Memonitor availabilitas dari setiap worker node(s) (contoh: active, dead, running, busy etc).
- **[10 poin]** Memonitor status jobs yang sedang dieksekusi di worker node(s) (contoh: failed, finished, running etc)
- **[10 poin]** Memonitor di worker node(s) mana suatu job dieksekusi.
- **[10 poin]** Membatalkan job yang sedang dieksekusi di suatu worker node(s).
- **[10 poin]** Adanya variasi jenis jobs berdasarkan lamanya waktu eksekusi (contoh: sangat singkat (milis), singkat (s), lama (minutes)).
- **[10 poin]** Adanya antrian jobs di master node berdasarkan algoritma FCFS.
- **[10 poin]** Adanya antrian jobs di worker node(s) berdasarkan algoritma FCFS.
- **[10 poin]** Implementasi algoritma penjadwalan selain FCFS di antrian master node.
- **[10 poin]** Implementasi algoritma penjadwalan selain FCFS di antrian worker node.

*c. Fitur Advanced (bersifat bonus, **max 50 poin**)*

- **[25 poin]** Master node berfungsi sebagai portal dan di set up di EC2. Ada program client yang bisa di install di local computer yang dapat men-submit jobs ke portal.
- **[25 poin]** Implementasi load balancer--tanpa menggunakan teknologi yang sudah ada--yang dapat membagi beban setiap worker node(s).
- **[25 poin]** Implementasi fitur auto-upscaling--tanpa menggunakan teknologi yang sudah ada--sehingga sistem dapat menambah jumlah worker node(s)/meng-create VM baru secara otomatis ketika sangat sibuk.
- **[25 poin]** Implementasi fitur auto-downscaling--tanpa menggunakan teknologi yang sudah ada--sehingga sistem dapat mengurangi jumlah worker node(s)/men-terminate VM yang ada secara otomatis ketika sangat sepi.
- **[25 poin]** Implementasi fitur security sehingga komunikasi antara master dan worker node(s) perlu di autentikasi sebelum mengirimkan jobs.