

Project

Mata Kuliah

Sistem Terdistribusi

Pemrosesan Video Terdistribusi

Berbasis Cloud untuk Deteksi Objek

Dibuat oleh:

Nama : Ansyarullah
NIM : 1217050017
Kelas : A

A. Tentang Project

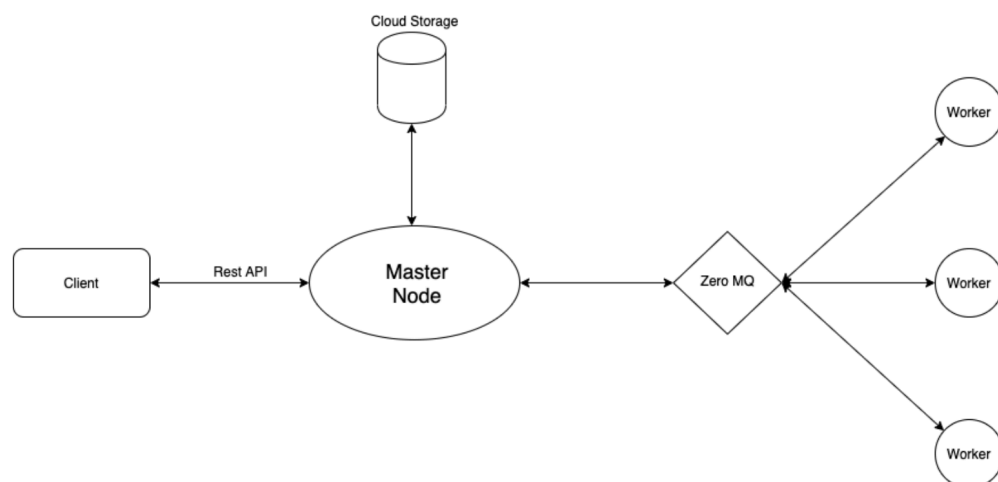
Proyek bertujuan untuk membuat pemrosesan terdistribusi frame video untuk deteksi objek menggunakan DNN yang telah dilatih sebelumnya. Dengan memasukkan video ke dalam sistem, pada dasarnya kita akan mendapatkan kembali video yang sama dengan objek yang berbeda yang disorot dan juga jumlah mereka dalam bingkai tertentu. Objek-objek ini bisa berupa mobil, orang, atau objek lain yang didukung oleh DNN yang telah dilatih sebelumnya. Hal ini akan terjadi dengan memecah video menjadi beberapa frame dan memproses masing-masing frame tersebut. Setelah pemrosesan dilakukan pada frame, mereka digabungkan kembali dengan cara agar urutan frame asli tetap terjaga. Dengan demikian, kita mendapatkan video yang diperlukan dengan semua objek dan jumlahnya disorot.

B. Komponen Perangkat Lunak

Berikut adalah beberapa komponen perangkat lunak yang digunakan.

1. Antarmuka REST API untuk melayani klien
2. Penyimpanan Objek Cloud untuk menyimpan video input dan output
3. Instance VM untuk node master dan *workers*
4. ZeroMQ untuk menangani pemrosesan frame video

Komponen perangkat lunak tersebut disajikan dalam Diagram Arsitektur berikut.



C. Deskripsi Proyek dan Interaksi antara Komponen yang berbeda

- a. Node klien mengirimkan permintaan untuk memproses video ke node master (Semua node di sini mengacu pada VM). Permintaan terdiri dari URL yang mengarah ke ember penyimpanan cloud yang menyimpan video. Permintaan ini diterima oleh node Master yang memecah video menjadi beberapa frame. Masing-masing frame dikirim ke node pekerja untuk diproses melalui ZeroMQ yang memiliki pembungkus khusus untuk menangani gambar secara khusus. Hingga node pekerja selesai memproses dan mengirimkan acknowledgement kembali, ZeroMQ tidak akan mengirimkan frame baru

untuk diproses ke node pekerja tersebut. ZeroMQ mengirimkan frame dalam batch 100 frame ke setiap VM. Dengan demikian, ZeroMQ membantu menangani pemrosesan beberapa frame secara bersamaan.

- b. Masing-masing node pekerja dan node master akan menjadi instance VM yang terpisah. Mereka akan digunakan untuk mengidentifikasi objek di setiap frame video menggunakan model Neural Network yang telah dilatih sebelumnya. Masing-masing node pekerja akan memproses frame video yang diberikan dan mengembalikan frame yang telah diproses kembali ke node Master.
- c. Node Master kemudian akan menserialisasi frame sesuai dengan nomor frame dan menyusunnya kembali untuk mendapatkan kembali video yang serupa dengan video input. Hal ini harus dilakukan agar kita mendapatkan kembali video dengan urutan yang sama dengan yang asli. Kemudian, video yang telah diproses ini disimpan dalam ember penyimpanan cloud. URL video baru yang memiliki objek yang disorot di dalamnya, akan dikirim kembali ke klien setelah video yang diproses berhasil disimpan di ember penyimpanan cloud.

D. Deskripsi setiap Komponen:

a. NolMQ

Tujuan: Dalam proyek ini, ZeroMQ digunakan untuk mengimplementasikan antrian pesan untuk mendapatkan gambaran tentang sistem broker seperti RabbitMQ. Oleh karena itu, di project ini ingin menjelajahi dunia sistem tanpa broker dan juga memilih ini untuk beberapa alasan khusus yang tercantum di bawah ini.

Keuntungan:

- i. Tidak seperti implementasi antrian pesan lainnya seperti RabbitMQ dan lainnya; ZeroMQ tidak membutuhkan broker pesan khusus. Tanpa broker, berbiaya rendah dan mudah diimplementasikan.
- ii. Kamera ini menggunakan pengakuan positif untuk mengirim frame, sehingga memastikan tidak ada frame video yang hilang.
- iii. Dalam proyek kita, kita harus mengirim gambar antara mesin virtual yang berbeda. Hal ini akan sangat mudah dilakukan dengan ZeroMQ, karena ada pembungkus yang disebut ImageZMQ yang dibuat khusus untuk tujuan ini.
- iv. Keuntungan terakhir adalah skala yang sangat baik untuk aplikasi terdistribusi seperti aplikasi ini.

Kekurangan:

Hubungan antara berbagai komponen perlu dikelola lebih banyak jika dibandingkan dengan sistem lain.

b. Contoh VM :

Tujuan : Proyek ini telah menggunakan beberapa instance VM, beberapa di antaranya bertindak sebagai node pekerja dan salah satunya bertindak sebagai node Master dan satu lagi sebagai klien.

Keuntungan :

- i. Karena Project ini telah menggunakan VM yang sangat berpengalaman dalam mengimplementasikannya dan memilihnya daripada Kubernetes.
- ii. Juga mudah untuk melakukan debug jika terjadi kesalahan pada VM karena kita dapat dengan mudah melakukan SSH ke dalam VM dan memeriksa log, yang sangat sulit dilakukan pada Kubernetes.
- iii. Juga sangat mudah untuk menggunakan VM di GCP dengan OS yang diperlukan. Dan juga sangat mudah untuk menginstal perangkat lunak lain di atasnya.

Kekurangan:

- iv. Karena kita telah menggunakan VM, kita harus menambah dan menghapus node pekerja secara manual tergantung pada beban. Hal ini dapat diotomatisasi jika kita menggunakan Kubernetes menggunakan sesuatu seperti Load Balancing.
- v. Kerugian kedua adalah penggunaan sumber daya yang tidak efisien dalam hal penggunaan VM; karena kita tidak membutuhkan banyak memori untuk menjalankan aplikasi sederhana.

c. Penyimpanan Objek Cloud :

Tujuan: Di sini, penyimpanan objek Cloud pada dasarnya digunakan untuk menyimpan video yang sudah diproses, dan juga untuk mengambil video pada awalnya.

Keuntungan:

- i. File video yang sangat besar dapat dengan mudah diunggah dan diunduh dari Cloud Object Storage.
- ii. Karena VM terletak di dalam ekosistem yang sama di Google Cloud, ada akses cepat ke video yang terletak di penyimpanan Cloud. Hal ini terutama ditingkatkan jika mereka berada di Wilayah atau Zona yang sama.

Kekurangan:

- iii. Jika ide proyek ini disediakan sebagai layanan; mungkin akan merepotkan pengguna. Karena mereka akan lebih memilih untuk mengunggah video langsung dari perangkat mereka daripada mengunggahnya ke Google Cloud dan kemudian mengaksesnya dari sana.

- iv. Meskipun file video yang sangat besar dapat disimpan di Google Cloud, namun secara teoritis ada batasnya. Dan untuk mendapatkan batas penyimpanan yang lebih besar, akan dikenakan biaya lebih banyak dan membuat layanan ini lebih mahal untuk dioperasikan.

d. REST API :

Tujuan: Project ini menggunakan REST API untuk menangani permintaan antara klien dan master node. Klien mengirimkan permintaan yang pada dasarnya adalah URL Cloud Object Storage tempat video disimpan, yang kemudian diproses oleh node master.

Keuntungan:

- i. Sangat mudah untuk membuat kode REST API jika dibandingkan dengan sesuatu seperti gRPC.
- ii. Muatan permintaan sangat kecil, karena hanya berupa satu nama URL. Selain itu, dengan asumsi hanya sedikit permintaan yang dikirim dalam satu waktu, ini hampir sama cepatnya dengan gRPC. Ini
- iii. Ada fleksibilitas yang lebih besar yang disediakan karena REST dapat menangani beberapa jenis panggilan dan menangani format file yang berbeda dengan sangat mudah; yang diperlukan untuk proyek ini.

Kekurangan:

- iv. Karena REST membuat koneksi TCP baru untuk setiap permintaan, metode ini mungkin lambat jika dibandingkan dengan gRPC terutama jika klien dan master secara fisik berada pada jarak yang lebih jauh satu sama lain.

E. Debug/Pengujian dalam proyek:

- a. Project ini telah menggunakan log VM (yaitu SSH ke dalam VM dan memeriksa pemrosesannya) untuk men-debug kesalahan apa pun yang berkaitan dengan Master atau node pekerja. Dengan memeriksa log ini, kita akan dapat memastikan apakah frame video dipecah dengan benar atau tidak. Kita juga akan dapat melihat apakah objek terdeteksi dalam satu frame atau tidak.
- b. Proses yang sama juga digunakan untuk memastikan bahwa ZeroMQ bekerja dengan baik. Pada dasarnya kita dapat memeriksa log untuk ZeroMQ dengan melakukan SSH ke dalam VM dan memastikan apakah pesan-pesan sedang mengantri dengan benar dan juga kita dapat memeriksa apakah semua permintaan ditangani dengan benar atau tidak dengan mencetak permintaan.
- c. Project ini menyertakan bendera Verbose di dalam kode. Dengan mengaktifkannya, semua proses dan permintaan akan dicetak; dan ini akan membuat debugging atau pengujian menjadi lebih mudah.

- d. Langkah verifikasi terakhir dapat diperoleh dengan memeriksa video akhir yang memiliki objek yang disorot di dalamnya. Jika video baru cocok dengan video asli dalam hal frame, maka itu berarti pemisahan frame dan kemudian proses penggabungan frame dilakukan dengan benar oleh master node.
- e. Jika video baru yang diperoleh juga memiliki semua objek yang disorot di dalamnya, maka itu berarti setiap node pekerja telah berhasil menyelesaikan tugas mereka mengidentifikasi objek dalam satu frame.
- f. Model jaringan syaraf tiruan yang ingin kita gunakan sudah dilatih dan diuji sebelumnya. Oleh karena itu, kita bisa yakin dengan output yang dihasilkannya dan tidak perlu mengimplementasikan mekanisme pelatihan/pengujian terpisah untuk model tertentu. Satu-satunya hal yang dapat kita mainkan adalah berbagai parameter yang terkait dengan model jaringan saraf seperti ukuran kepercayaan yang pada akhirnya berdampak pada seberapa baik objek diidentifikasi dalam video.

F. Kemampuan Sistem

- a. Sistem ini dapat menangani video dalam berbagai ukuran dan resolusi. Meskipun demikian, menyediakan video dengan ukuran besar akan menghabiskan banyak waktu untuk pemrosesan dan untuk mendapatkan hasil akhir. (Sebagai contoh: Video 1080p berdurasi 15 menit akan membutuhkan waktu sekitar 20 menit untuk diproses).
- b. Sistem ini dapat mendeteksi berbagai macam objek yang berbeda yang didukung oleh Model Jaringan Syaraf Tiruan yang digunakan. Ini termasuk berbagai objek seperti mobil, sepeda, orang, dll. Dengan demikian kita bisa mendapatkan jumlah total mobil atau orang dalam video atau total gabungan dari semua jenis objek yang berbeda juga.
- c. Sistem yang sama, dengan desain arsitektur yang sama dapat diperluas untuk berbagai jenis aplikasi yang melibatkan pemrosesan video. Hanya kode untuk node pekerja yang mungkin berubah dalam hal ini, dan sisa stack tetap sama. Contohnya adalah Peningkatan Resolusi Video menggunakan Jaringan Syaraf.

G. Keterbatasan Sistem

- a. Bergantung pada beban atau ukuran video, kita tidak dapat secara otomatis meningkatkan node pekerja. Kita harus secara manual membuat VM baru dan menambahkannya sebagai simpul pekerja. Hal ini dapat dihindari dengan menggunakan Kubernetes dengan penyeimbangan beban otomatis yang diaktifkan.
- b. Sistem ini bergantung pada fakta bahwa semua VM tetap aktif dan melakukan tugasnya sampai akhir pemrosesan video. Misalkan, salah satu VM berhenti

bekerja dan macet; maka frame yang diproses pada saat macet akan hilang. Karena setiap VM menerima bundel 100 frame sekaligus. Dengan demikian, minimal 100 frame akan hilang tergantung pada jumlah VM yang crash.