

Paralelisasi Komputasi Penghitungan Jumlah Kendaraan Menggunakan GPU untuk Mendapatkan Data Penghitungan Secara Real Time = Parallelization of Vehicle Count Computation Using GPUs to Obtain Real Time Vehicle Data Count

Alan Novaldi, author

Deskripsi Lengkap: <https://lib.ui.ac.id/detail?id=9999920534141&lokasi=lokal>

Abstrak

Sistem lampu lalu lintas cerdas merupakan sistem yang dapat melakukan pengaturan lampu lalu lintas secara adaptif berdasarkan kondisi kepadatan lalu lintas. Salah satu cara untuk mendapatkan kondisi kepadatan lalu lintas adalah melakukan komputasi penghitungan jumlah kendaraan dari video CCTV yang terpasang pada persimpangan. Pada penelitian ini dilakukan paralelisasi program penghitungan jumlah kendaraan menggunakan modul Multiprocessing pada python untuk mendapatkan data penghitungan kendaraan dari setiap jalan di persimpangan. Selanjutnya utilisasi GPU dilakukan untuk mendapatkan data secara real time dari suatu komputasi berat video processing. Pada penelitian ini, utilisasi GPU dilakukan dengan menggunakan CUDA sebagai platform yang dapat menghubungkan program dengan GPU pada low-level. Pengelolaan utilisasi GPU pada high-level dilakukan menggunakan TensorFlow yang sudah terintegrasi dengan CUDA. Uji coba eksekusi program dilakukan untuk mendapatkan runtime terbaik dari eksekusi program. Komputasi secara paralel menghasilkan runtime eksekusi komputasi 1.6 kali lebih cepat jika dibandingkan dengan komputasi secara sekuensial. Pada tingkat utilisasi GPU yang lebih optimal, runtime eksekusi komputasi dapat ditingkatkan hingga 2 kali lebih cepat dari komputasi normal. Utilisasi GPU juga terbukti meningkatkan runtime eksekusi program karena komputasi utama video processing tidak lagi dijalankan menggunakan CPU. Hasil uji eksekusi komputasi digunakan untuk membuat visualisasi data penghitung jumlah kendaraan. Visualisasi ini dilakukan agar data yang penghitungan dapat diproses lebih lanjut untuk sistem pengatur lampu lalu lintas. Pada akhir penelitian dilakukan profiling performa GPU menggunakan Nvprof dan NVIDIA Visual Profiler sebagai tools yang disediakan oleh CUDA. Hasil profiling menunjukkan analisis yang menyatakan bahwa tingkat penggunaan GPU untuk komputasi masih belum secara maksimal dilakukan. Hal ini terbukti dari rendahnya angka compute utilization, average throughput dan kernel concurency dari eksekusi program. Sehingga diperlukan adanya optimisasi program penghitungan kendaraan agar utilisasi GPU lebih optimal.

.....Traffic light intelligence system is an adaptive system which able to control traffic flow on road intersection based on traffic condition. Traffic density information can be obtained from vehicle counting computation using deep learning methodology on CCTV record video data of a road intersection. This study performed parallelization of the vehicle counting computation using the Multiprocessing module in python to get the number of vehicles approaching the intersection. GPU Utilization is performed to obtain vehicle counting data in real time from a heavy computation like video-processing. GPU utilization is carried out using CUDA as a platform that can connect programs with GPUs at low-level architecture. GPU utilization management at high-level is done using TensorFlow which has been integrated with CUDA. Some experiments are performed to get the best runtime from program execution. Parallel computation produces runtime execution 0.6 times faster compared to sequential computation. On more GPU compute utilization optimization, parallel computation can produce runtime 2 times more compared to normal computation.

GPU utilization has also been proven to increase the program execution runtime because the main computational video processing is no longer run on the CPU. The experiment result on vehicle detection used to create data visualization about vehicle counting on a road intersection. Data visualization is done so that the vehicle data can be further processed for the traffic light control system. At the end of the study GPU performance profiling was done using Nvprof and NVIDIA Visual Profiler as tools provided by CUDA. Profiling results show that analysis states that the level of GPU usage for computing is still not maximally done. This analysis is shown from the low number of compute utilization, average throughput and kernel concurrency of program execution. GPU utilization need to be optimized in order the program can run optimally on GPU.