Technische Universität München | Institut für Informatik | I13
Boltzmannstr. 3 | D- 85748 Garching | Germany

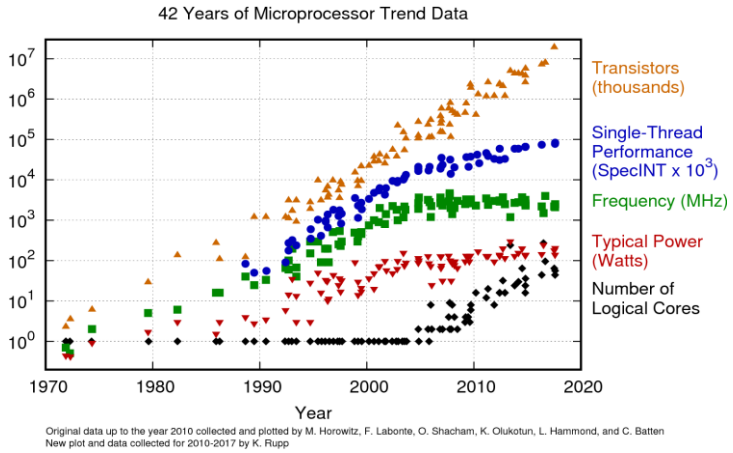
Master's Thesis

7th Aug. 2018

Massive Graph Partitioning on a GPU

Looking for a challenging but rewarding master's thesis? Use your master's thesis to set the course for your future in a seminal field. Hardware is at an inflection point and software has to adapt. Acquire the skills required for designing and developing tomorrow's systems while doing research on algorithms targeting massively parallel hardware (GPUs).

Background



For nearly 40 years, ever increasing clock rates allowed software systems to naturally benefit from hardware advancements and to cope with ever growing workloads. Software systems could remain unmodified and simply benefit from faster sequential execution due to higher clock rates. However, in the mid 2000's, this trend came to a halt. Since then, processors have seen only moderate improvements in sequential execution. In order to sustain the trend of exponentially growing computational throughput, manufacturers have therefore progressively turned towards scaling hardware parallelism, such as increasing the number of cores and extending SIMD capabilities. Today, parallelism is the new dimension of scale.

Other than before, this requires algorithms to be redesigned from the ground up to scale with increasing hardware parallelism. As hardware parallelism keeps increasing, it amortizes for more and more algorithms to take a constant factor increase in the total amount of work being performed in order to achieve linear scalability. The following thesis targets GPUs as a representative for this ongoing trend.

Your Task

You work on the challenging problem of ***graph partitioning***. Large graph processing systems such as Pregel, Giraph and GraphX can perform data analytics on massive real-world graphs such as friendship graphs from social networks or web graphs. A prominent example is the *PageRank* algorithm that is used by Google to prioritize web search results. The graphs to be analyzed are too large to fit on a single machine; hence, they are partitioned (i.e., cut into a number of equally-sized pieces) and distributed onto multiple machines. In doing so, the quality of partitioning (i.e., the cut size) must be kept minimal in order to reduce communication overhead when analyzing the distributed graph.

Current graph partitioning algorithms are designed for being executed on CPUs and cannot exploit the opportunities of massively parallel GPUs. Your task is to develop and implement a graph partitioning algorithm on GPU.

Prerequisites

- Fundamental understanding of parallel algorithms
- **Prior experience with C++ and CUDA**
- Ability to research and work independently

Supervisor: Prof. Dr. Hans-Arno Jacobsen
Advisors: Dr. Ruben Mayer, Elias Stehle

Interested? Send your inquiry, preferably with projects, references, CV, motivational letter and/or study transcript to: ruben.mayer@tum.de