

Bachelor-/Masterarbeit

Exploring highly-scalable Breadth-First-Search (BFS)

Beschreibung

Im High Performance Computing (HPC) werden Anwendungen auf verteilten Systemen mit Tausenden oder Millionen von Prozessorelementen (PEs) ausgeführt. Solche Supercomputer werden vor allem für numerische Simulationen wie bspw. Wetter- oder Klimamodelle genutzt. Diese Anwendungen besitzen sehr gleichförmige Datenzugriffs- und Kommunikationsmuster, welche algorithmisch relativ einfach zu handhaben sind.

Seit einigen Jahren werden jedoch Anwendungen mit irregulären Kommunikationsmustern, beispielsweise aus den Bereichen Machine Learning, Data Mining und Materialwissenschaften zunehmend relevant. Gemein ist diesen, dass sich die unterliegenden Probleme häufig auf (grundlegende) Graphprobleme verallgemeinern lassen.

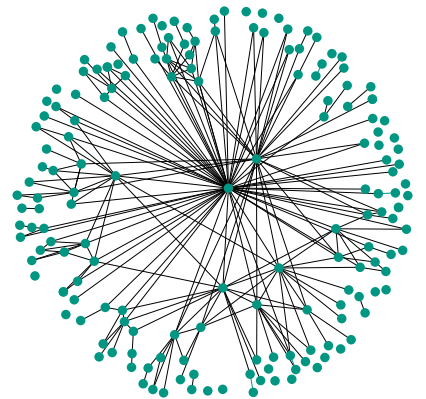
Um die wissenschaftliche Arbeit auf diesem zunehmend wichtigen Feld voranzutreiben, wurde daher der graph500-Benchmark ins Leben gerufen. Dieser sucht nach den schnellsten Breadth-First-Search (BFS) Algorithmen auf R-MAT-Graphen. Der Benchmark hat die Forschung zu verteilten BFS-Algorithmen intensiviert und zu vielen wichtigen und interessanten Ergebnissen geführt, wodurch es mittlerweile möglich ist R-MAT Graphen mit Billionen von Kanten in kurzer Zeit zu traversieren. Ein Nachteil dieses Benchmarks ist jedoch die Einschränkung auf eine einzige Instanzfamilie. So werden die Algorithmen gezielt auf die Eigenschaften dieser Graphen hin optimiert. Dieses Problem soll in der vorliegende Abschlussarbeit in Angriff genommen werden.



Thema der Arbeit

Erster Bestandteil der Arbeit ist eine umfassende Sichtung der vorhandenen Literatur zu effizienten verteilten BFS-Algorithmen. Die wichtigsten Startpaper werden von uns bereitgestellt. Die besten Algorithmen sollen dann in C++ (oder Rust) unter Verwendung von MPI (nach-)implementiert werden. Die Implementierungen sollen dann auf einer Vielzahl verschiedener Graphfamilien evaluiert werden. Zur Generierung der Graphen kann unser verteilter Graphgenerator KaGen <https://github.com/sebalamm/KaGen> genutzt werden.

Auf Basis dessen soll ein effizienter skalierbarer BFS-Algorithmus für allgemeine Graphfamilien entwickelt werden.



Voraussetzungen

- Interesse an parallelen Algorithmen und High Performance Computing
- Gute Programmierkenntnisse in (modernem) C++ (oder Rust)
- Grundlegende Kenntnisse in Parallelprogrammierung mit MPI

¹ <https://upload.wikimedia.org/wikipedia/commons/8/89/SuperMUC.jpg>