

Bachelor-/Masterarbeit

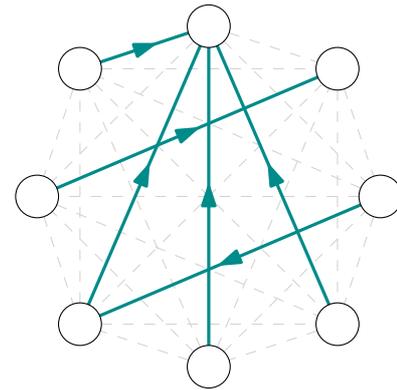
Speedup Techniques for Irregular Sparse All-to-All

Beschreibung

Im High Performance Computing (HPC) werden Anwendungen auf verteilten Systemen mit Tausenden oder Millionen von Prozessorelementen (PEs) ausgeführt. Diese Supercomputer kommunizieren über Hochgeschwindigkeitsnetzwerke wie bspw. InfiniBand.

Viele Anwendungen verbringen einen Großteil ihrer Ausführungszeit mit Kommunikation, in der Daten im Netzwerk verteilt werden. Reguläre Kommunikationsmuster wie All-To-All- und Stencil-Exchanges werden von aktuellen MPI-Implementierungen effizient unterstützt und skalieren bis zu einer hohen Anzahl PEs. Für den Fall, dass Daten unregelmäßig verteilt sind oder die Anzahl an Kommunikationspartnern einzelner PEs stark schwankt, skalieren bestehende Implementierungen von All-To-All, sowie explizite Punkt-zu-Punkt-Kommunikation schlecht. Selbst Metadaten für alle möglichen Ziel-PEs zu erstellen, würde die Kommunikationszeit dominieren.

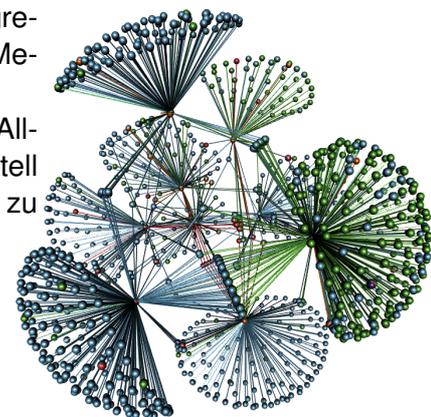
Dies stellt insbesondere ein Problem für massiv-parallele verteilte Graphalgorithmen dar, da es hier häufig nötig ist, Informationen zwischen Knoten des Graphen auszutauschen. Die Struktur großen Realwelt-Graphen, wie z.B. sozialen Netzen und Webgraphen, führt in der Regel zu irregulären sparsen Kommunikationsmustern.



Thema der Arbeit

Experimente haben gezeigt, dass naives Senden mittels einer hohen Anzahl asynchroner Nachrichten das Netzwerk "verstopfen" kann. Abhilfe können hier Aggregation und indirekte Kommunikation schaffen. Denkbar sind auch Ansätze die Mechanismen zur Flusskontrolle oder Graphfärbungen nutzen.

Ziel der Arbeit ist es, neue skalierbare Kommunikationsprimitive für irreguläre All-To-All-Exchanges für eine große Anzahl PEs zu entwickeln und diese experimentell mit bestehenden Ansätzen im Kontext von Graphalgorithmen zu vergleichen und zu evaluieren.



Vorraussetzungen

- Interesse an parallelen Algorithmen und HPC
- Gute Programmierkenntnisse in (modernem) C++
- Grundlegende Kenntnisse in Parallelprogrammierung mit MPI

