Albrecht Fortenbacher
Angewandte Informatik

Hochschule für Technik
und Wirtschaft Berlin
University of Applied Sciences

Belegaufgabe:
Benchmark für das zeilenweise Sortieren von Texten

Sommersemester 2011
Distributed Systems and Parallel Processing

Belegaufgabe

Sortieren Sie ein vorgegebenes Textfile zeilenweise durch ein paralleles Programm (MPI). Führen Sie dazu im Labor 638 einen Benchmark durch. In der Belegpräsentation stellen Sie Ihren parallelen Algorithmus (Quicksort, Mergesort, …) vor und analysieren die Ergebnisse des Benchmarks (z.B. des erzielten Speedups).

Textfile

Das zu sortierende Textfile besteht aus dem Werk „G.J. Caesar: de bello gallico“. Der Text kommt erst unverschlüsselt und dann in verschiedenen Codierungen vor. Sie finden das Buch unter http://wikisource.org.

Das Textfile besteht aus 18603144 Zeilen, welche eine maximale Länge von 126 Zeichen haben (Zeilenende nicht eingerechnet). Insgesamt ist es knapp 2 GB groß. Sie finden es unter /usr/local/sortMe.txt auf allen Rechnern des Labors 638 (d.h. Sie können es direkt von der Festplatte lesen, ohne Netzzugriffe).

Vergleichsverfahren

Das File sortMe.txt soll zeilenweise sortiert werden (aufsteigend). Dabei kommt folgendes Vergleichsverfahren zum Einsatz:

- ein String s1 kommt vor einem String s2 (s1 ist kleiner als s2), falls
  - bis zum k'ten Buchstaben im Alphabet (Beispiel: q) beide Strings in der Anzahl der Klein- und Großbuchstaben übereinstimmen (also: s1 hat gleich viele a's, b's, …, q's, A's, B's, …, Q's wie s2
  - beim nächsten Buchstaben (r) ist die Summe der Klein- und Großbuchstaben (alle r's und R's) bei s1 größer
  - oder: s1 und s2 haben die gleiche Summe von r's und R's, aber s1 besitzt mehr r's
- falls sich s1 und s2 in keinem Buchstaben des Alphabets unterscheiden, dann werden s1 und s2 nicht geordnet, sie können in beliebiger Reihenfolge vorkommen.

Programmierung

Verwenden Sie für das Sortieren ein parallelisierbares Sortierverfahren wie Quicksort oder

Mergesort.

Programmiert werden sollte der Algorithmus mit C und MPI. Für ein effizientes Rechnen sollten die Daten möglichst zusammenhängend gespeichtert werden (Cache!). Da der Vergleich zweier Zeilen relativ aufwändig ist, kann es sinnvoll sein, eine Zeile durch ihr Histogramm (wie oft kommt jeder Buchstabe des Alphabets vor) zu speichern. Wenn Sie es geschickt machen (Datentyp unsigned char verwenden), dann können Sie eine Zeile durch ein Array von 52 Bytes darstellen.

Benchmark

Der Benchmark wird auf dem Cluster des Labors 638 durchgeführt:

- Messen Sie das Sortieren mit 1, 4, 8, 16, 24, 32 und 64 Parallelknoten, wobei jeweils alle 4 Cores eines Rechners angesprochen werden (außer bei einem Parallelknoten). Für einen Parallelknoten soll das gleiche (parallelisierte) Programm verwendet werden.
- Zur Kontrolle geben Sie die Zeilen mit den Nummern 10, 100, 1000, 10000, 100000, 1000000 und 10000000 aus.
- Zeitmessung geschieht mit MPI_Wtime, sie beginnt mit dem Einlesen des Textes und endet, wenn der sortierte Text geschrieben wurde.
- Mit dem Tool Jumpshot kann das Kommunikationsverhalten visualisiert werden. Achtung: bitte nicht messen, wenn die Daten für Jumpshot aufgezeichnet werden.

Präsentation

Bei der Präsentation stellen Sie Ihren parallelen Algorithmus, die Umsetzung des Algorithmus und die Ergebnisse des Benchmarks vor. Analysieren Sie sowohl das theoretische Verhalten des Algorithmus (Komplexität, Amdahl'sches Gesetz, Verhältnis Rechnen/Kommunikation, …) als auch die Ergebnisse des Benchmarks (Speedup, …). Ideal ist es, wenn man erklären kann, weshalb die Ergebnisse so oder so sind.

Bitte beginnen Sie Ihre Präsentation mit einer Übersetzung des Satzes (soweit er aus einer Zeile erschlossen werden kann), der in der sortierten Datei an Position 545146 (Zählung ab 1) steht.