

1 Testreihen

1.1 Umgebung

Um das verteilte Sortieren sinnvoll Testen zu könne, ist es notwendig die Ausführung auf mehrere Rechner zu verteilen. Eine rein lokale Ausführung würde zum einen keinen realistischen Kommunikationsoverhead bedeuten, zum anderen wäre die Ausführung an die Leistungsfähigkeit der lokalen CPU gebunden.

Die Messreihen wurden daher auf virtuellen Maschinen in der Amazon Elastic Compute Cloud (Amazon EC2)¹ ausgeführt. Die Ausführung erfolgte dabei jeweils für Server und jeden Client getrennt auf einer eigenen „Micro“ Instanz. Zur Durchführung der Testreihen wurde mit Hilfe von Python, Fabric² und boto³ automatisiert und ausgeführt.

1.2 Durchführung der Messung

Die Performancemessung wird ausschließlich auf dem Server durchgeführt. Der Server misst dabei, wie viele Millisekunden er für das einlesen der zu sortierenden Daten, zum verteilen der Daten an die Clients sowie zum Interieren über die Sortierten Ergebnisse benötigt. Außerdem wird noch die gesamte Ausführungszeit gemessen. Da es sich bei der Umgebung, in der das Sortieren durchgeführt wird, um virtuelle Maschinen handelt, ist die Ausführungszeit natürlich abhängig von der Auslastung der physikalischen Hosts. Es ist jedoch davon auszugehen, dass aufgrund der Tatsache, das Amazon die Instanzen über viele Rechner verteilt, ein brauchbarer Mittelwert entsteht.

1.3 Parameter der Messungen

Die Messreihen wurden jeweils mit 2, 4, 8 und 16 Sortierclients und einem Server durchgeführt. Dabei wurden jeweils die Blockgrößen 10, 100, 1000 und 10000 getestet. Für die lokale Sortierung gelten diese Parameter natürlich nicht, hierbei wurde nur die Sortierung mittels `Collections.sort()` ausgeführt. Als Daten die sortiert werden sollen wurden Goethes Faust I und II ausgewählt, die dem Projekt Gutenberg als Textdatei entnommen wurden. Die Testdatei hat dabei eine Dateigröße von 556 Kilobyte und wird vom Server vor der Verteilung an jedem Whitespace aufgetrennt. Dabei entstehen 81333 einzelne Strings, die dann Sortiert wurden.

¹<http://aws.amazon.com/ec2/>

²<http://fabfile.org>

³<http://code.google.com/p/boto/>