Uni Osnabrück / Theoretische Informatik — Prof. Dr. M. Chimani

Algorithm Engineering — Übungsblatt 2

Wintersemester 2015/16 — Ausgabe 2. Nov. — Besprechung 16. Nov.

Aufgabe 2.1: External Array-Heap
(Alle Teilnehmer)

Beweisen Sie die in der Vorlesung angesprochenen Lemmata C und D:

Lemma C. Sei $cM > 3B$. Nach N Operationen werden maximal $L \leq \log_\alpha(N/B)$ Level benutzt.

Lemma D. *Store*(i, S), *compact*(i) und *merge*$(i-1, S, S')$ benötigen maximal $3\ell_i/B$ I/Os.

Aufgabe 2.2: Externes Mergesort (Implementierung)
(Matthias Bultmann, Enno Lohmeier, Simon Veltel)

Implementieren Sie den in der Vorlesung besprochenen externen Mergesort-Algorithmus. Untersuchen Sie Ihre Implementierung in Bezug auf Laufzeit (absolut, pro Schlüssel,...) und die *tatsächliche* Cache-Größe für:

- uniform zufällig verteilte Schlüssel (z.B. 32-bit Integers) in Feldern interessanter Größe (z.B. 2^{12} bis 2^{25}... gerne auch mehr),
- unterschiedliche Elementgrößen,
- unterschiedliche Werte für M und B,
- unterschiedliche Rechner.

Anmerkung: Die beobachteten Unterschiede sollten also durch Caches vs. RAM entstehen, nicht durch RAM vs. HDD. Bei Nachfragen zur Aufgabenstellung bitte zeitnah melden!

Aufgabe 2.3: Funnelsort
(Waldemar Smirnow, Michael Stypa)

(Lazy) Funnelsort ist ein cache-oblivious Sortieralgorithmus. Wie und warum funktioniert er? Was sind seine Laufzeit- und I/O-Garantien? Was kann man über seine Praxistauglichkeit sagen?

Aufgabe 2.4: Externes Hashing
(Patrick Schulz, Felix Siebert)

Das Paper *„How Caching Affects Hashing“* von G.L. Heileman und W. Luo (Proc. 7th ALENEX, 2005) untersucht die Auswirkungen von Caches auf das Verhalten verschiedener Hashfunktionen. Stelle die wesentlichen Resultate vor.