I/O Algorithms

I/O-efficient merge-sort 2014 - group 6

Frederik Lønstrup Hansen
20096365

Randi Katrine Hillerøe
20103073

Andreas Fyrstelin Kristiansen
20092027

Rasmus Kampmann Olsen
20093692

Abstract

ANSVAR: ... rasmus? Har vi brug for et abstract? Der er jo som sådan intet nyt i det her, så tænker en introduction er nok

1. Introduction

In this report we describe our implementation, experiments and results we have achieved during project 1 for I/O algorithms 2014. As reading and writing to and from the disk is estimated at 10^6 times slower than using RAM or the cpu cache, creating efficient algorithms can mean the difference between getting a result within a reasonable time-frame or not. The usual memory model is made with a very small fast and local memory near cpu, a larger RAM memory, and finally the disk. Each level that is further away from the cpu becomes slower by a large factor. In our experiments and calculations we are only interested in calculating and experimenting with the time of the I/O operations. An I/O operation is defined by either a read or write to the main memory.

2. Merge sort with streams

This section describes our implementation of the 4 different required streams.

2.1. Single item stream

Single item streams...

2.2. Fread and FWrite

Fread and FWrite

2.3. Buffered

Buffered input and output

2.4. mmap/munmap

mmap/munmap

3. D-way merge sort

ANSVAR: rasmus...andreas? D-way merge sort

4. External Merge-sort

results: Andreas?
External merge sort

5. Experiments

results: Frederik er pisse god til det der med grafer har jeg hørt!

something...? Setup, thoughts about variables etc

6. Comparisons of results

Ansvar: Os all?? Compare the results...discuss stuff

6.1. Ours versus Heapsort/Quicksort

7. Conclusion

Ansvar: Os alle !?

Oh wow I/O is slow shit!! You better make sure you do it in a smart way!