Range Reporting

Peter Gabrielsen 20114179
Christoffer Hansen 20114637

February 5, 2016

Abstract

blabla

Contents

1 Introduction

In the early days of computers disks were faster than processors. Since then processor technology has advanced at an incredible rate achieving annual speedups of 40 to 60 percent [RW94]. Although this is also true for disk capacity an entirely different story can be told for the speedup of disk performance. The disparity between processor, internal memory, and external memory speeds have grown larger for each year and the gap is widening. This disparity presents serious problems when designing algorithms on large data sets, and with more and more applications of Big Data in social media, banking, etc., these problems become increasingly relevant.

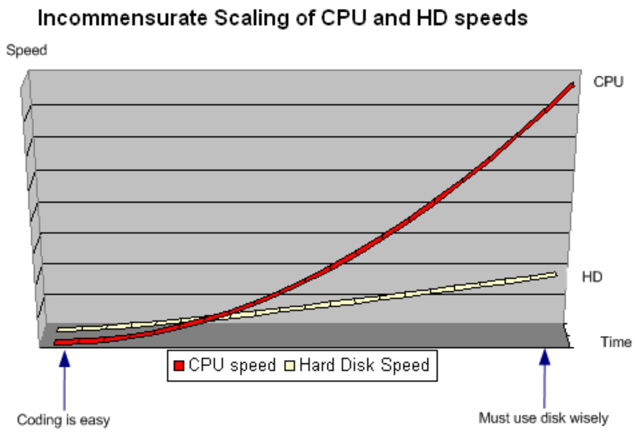

Figure 1: Growth of CPU compared to HD speeds over time.

Source: `http://read.cs.ucla.edu/111/2006fall/notes/lec15#incommensurate-scaling`

While the database community has always been involved in development of practically efficient external memory data structures, most algorithms research has focused on worst-case efficient internal memory data structures. With the advent of Big Data and problems concerning large data sets the algorithms community has been increasingly involved in developing worst-case efficient external memory data structures for solving these problems [Arg05].

To some large internet companies the bottleneck that disk-based systems presents has become to great of an obstacle and in an attempt to close the gap they have moved towards developing internal memory big data processing. This move has been allowed by growing main memory capacities but it comes with the price of issues such as fault-tolerance and consistency which are inherently more challenging to handle in volatile memory [ZCO$^+$15].

Another price of this move to internal memory is the actual cost of running server farms and the cost of internal memory compared to external memory. The extra costs and increased complexity suggests that external memory data structures have some well defined advantages.

In disk-based systems algorithms and data structures cannot be measured in the traditional models used for internal memory algorithms since the performance is bound by disk latency. In Section 2 we investigate a model that encapsulates performance of the I/O bottleneck.

In Section 3 we give a preliminary overview of some of the techniques used in developing external memory efficient data structures.

2 Model of computation

We will argue the results of this thesis in terms of the external memory model of Aggarwal and Vitter [AV88]. The external memory model (or I/O model) measures the efficiency of an algorithm by counting the total number of reads and writes performed. In detail the model consists of two levels of memory; a bounded internal memory of size M and an unbounded external memory. For a total of N records we define an *IO* operation to the process of transferring B consecutive records between the two levels of memory as depicted in figure 2. We restrict all computations on records to be done in internal memory. Throughout the thesis we will let K denote the total number of records in the output.

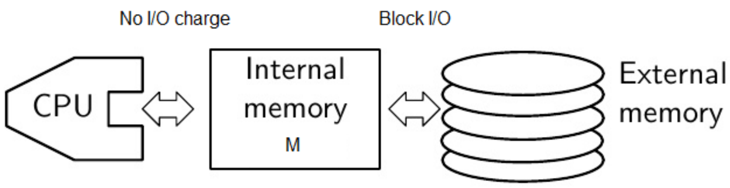

Figure 2: The IO Model. Only reads/writes between internal and external memory is charged.

The fundamental bounds in the external memory model is that scanning can be done in $\mathcal{O}(\text{Scan}) = \mathcal{O}(N/B)$, sorting in $\mathcal{O}(\text{Sort}) = \mathcal{O}(N/B \log_{M/B} N/B)$ and searching in $\mathcal{O}(\log_B N)$. We denote $\mathcal{O}(N/B)$ as being linear in terms of IOs. Note that the B factor is very important as $N/B < \mathcal{O}(N/B \log_{M/B} N/B) \ll N$.

For convenience we will assume $M > B^2$. This assumption is known as the *tall-cache assumption* in the cache-oblivious model and basically states that the number of blocks M/B is larger than the size of each block B [Pro99].

3 Preliminaries

3.1 Amortization

3.2 Global rebuilding

3.3 Filtering

4 Sublogarithmic Updates

5 External Memory Priority Search Tree

Bibliography

[Arg05] Lars Arge. External memory geometric data structures(lecture note). 2005.

[AV88] Alok Aggarwal and S. Vitter, Jeffrey. The input/output complexity of sorting and related problems. *Commun. ACM*, 31(9):1116–1127, September 1988.

[Pro99] Harald Prokop. Cache-oblivious algorithms, 1999.

[RW94] C. Ruemmler and J. Wilkes. An introduction to disk drive modeling. *Computer*, 27(3):17–28, 1994.

[ZCO+15] Hao Zhang, Gang Chen, Beng Chin Ooi, Kian-Lee Tan, and Meihui Zhang. In-memory big data management and processing: A survey. *Knowledge and Data Engineering, IEEE Transactions on*, 27(7):1920–1948, July 2015.