Nearly optimal oblivious sort

by Bingnan Chen
Division of Computer Science and Engineering, HKUST
The Hong Kong University of Science and Technology

Abstract

It is well known that sorting N elements in the external memory model requires at least $\text{sort}(N) = \frac{2N}{B}\left\lceil \frac{\log(N/B)}{\log(M/B)} \right\rceil$ I/Os, where M is the internal memory size and B is the block size, and this is attained precisely by external merge sort.

In recent years, due to growing concerns on privacy breaches on outsourced data, there is much interest in making algorithms *data-oblivious*, i.e., the access pattern to external memory should be independent of the input data. Unfortunately, merge sort is fundamentally difficult to be made oblivious, so prior work by Goodrich [29] chose to make distribution sort oblivious, but it incurs a large constant factor slowdown, with an I/O cost at least $40 \cdot \text{sort}(N)$. A recent algorithm called bucket sort [4, 50] took a different approach, reducing it to $3 \cdot \text{sort}(N)$. In this paper, we show that distribution sort can be made oblivious while attaining an I/O cost of $(1 + o(1)) \cdot \text{sort}(N)$, provided that M/B is greater than all polylogarithms of N, thus closing the gap between oblivious and non-oblivious external sort to within lower order terms. The algorithm is also simple and practical; our experimental results show that its I/O cost is indeed very close to the non-oblivious lower bound, especially when M is sufficiently large.