Lecture 15: Secondary Memory Algorithms

Xiaotie Deng

AIMS Lab
Department of Computer Science
Shanghai Jiaotong University

November 27, 2017

1. Sorting Big Data

2. Memory I/O wrt Disk Access

Sorting Big Data

Sorting Big Data

- All different sorting algortihms
- The favorite quick sorting in the past
- Extension to partition sort
- Locality efficiency in computation.
- Sorting competition: http://sortbenchmark.org

Memory I/O wrt Disk Access

Disk Data Transfer to Memory

- B-tree model
  - RAM size M, transfer block size B, data szie N.
- Operations/complexity
  - Data transfer: N/B i/o steps
  - Inset and query: $\log_B N$
- Searching is $\log_2 B$ faster than binary search tree.

Ideal-Cache Model

- Ideal cache of size Z words.
- Cache lines (page size) of L words moved together
- Tall Memory: $Z = \Omega(L^2)$.
- Cache Complexity $Q(n)$: The number of cache misses.
- Work complexity $w(n)$: the total number of operations.

REFERENCE:Matteo Frigo Charles E. Leiserson Harald Prokop Sridhar Ramachandran, Cache-Oblivious Algorithms.

Ideal-Cache Model Illustration

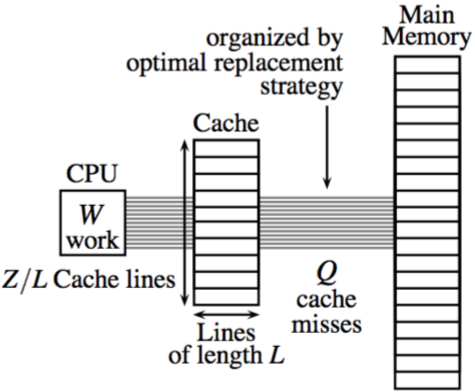

Figure 1: The ideal-cache model

Funnelsort

- Split input into $n^{1/3}$ contiguous arrays of size $n^{2/3}$, sort the them recursively.
- Merge $n^{1/3}$ sorted subsequence using a $n^{1/3}$-merger, described next.

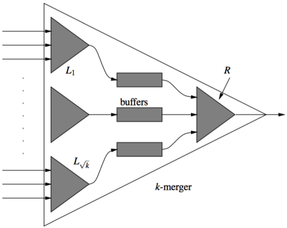

Figure 3: Illustration of a k-merger. A k-merger is built recursively out of $\sqrt{k}$ "left" $\sqrt{k}$-mergers $L_1, L_2, \ldots, L_{\sqrt{k}}$, a series of buffers, and one "right" $\sqrt{k}$-merger R.

k-Merger

- Recursively merging sorted sequences
- It suspends merging when output sequence becomes "long enough"
- resumes work on another merging subproblem
- See Figure 3, it has an invariant:
  - Each invocation of k-merger outputs the next k^3 elements of the sorted sequence obtained by merging the k input sequences, each of size k^2.

The Structure of k-Merger in Figure 3

- Input: $\sqrt{k}$ sets of $\sqrt{k}$ elements in left
- merged into $\sqrt{k}$ buffers of size $2k^{3/2}$ in the middle
- connecting to the $\sqrt{k}$ inputs of the $\sqrt{k}$ merger R in the right
- which is invoked $k^{3/2}$ times to output k^3 elements.
- Each left $\sqrt{k}$ merger outputs $k^{3/2}$ elements on each invocation

Properties

- k-merger occupies $O(k^2)$ contiguous memory locations
  - $S(k) \leq S(\sqrt{k}) * (1 + \sqrt{k}) + 2k^{3/2} * \sqrt{k} = O(k^2)$
- Performing r insert and remove operations on a circular queue causes in $O(1 + r/L)$ cache misses as long as two cache lines are available for the buffer.
  - The subsequence is already sorted. We will have no more miss for $L - 1$ inserts/deletes after we have had one miss.
- If $Z = \Omega(L^2)$ then $Q_M = O(1 + 2 + k^3/L + k^3 \log_2 k/L)$

Bounding Cache Misses

- Case 1: $k < \alpha\sqrt{Z}$.
  - a kmerger fits in the contiguous memory and can upload all $O(k^3)$ elements.
  - Tall cache assumption implies $L = O(\sqrt{Z})$ and The number of pages $Z/L = \Omega(k)$.
  - #misses on r_i operations is $\leq 1 + r_i/L$: r_i be # of elements extracted from i-th input queue,
  - Implying $\sum_{i=1}^{k} O(1 + r_i/L) = O(k + \frac{k^3}{L})$.
- Case 2: $k > \alpha\sqrt{Z}$.
  - Proving $Q_M(k) \leq ck^3 log_Z k/L - A(k)$ for $A(k) = k(1 + 2c\log_Z k/L) = o(k^3)$, by induction.

Bounding Cache Misses Case 2: $k > \alpha\sqrt{Z}$

- base case: $\alpha Z^{1/4} < k < \alpha\sqrt{Z}$.
  - As above, $Q_M(k) = O(1 + k + k^3/L) = O(k^3/L)$. The claim follows
- general case:$\alpha Z^{1/4} < \sqrt{k} < k$.
  - Claim holds for $\sqrt{k}$-mergers: $Q_M(\sqrt{k}) = O(1 + \sqrt{k} + k^{3/2}/L)$
  - Right merger R is invoked $k^{3/2}$ times.
  - Total number l of left merger invocation is no more than $l \leq k^{3/2} + 2\sqrt{k}$.
  - Each invocation of the right merger takes $\sqrt{k}$ checks to find out the emptiness of each buffer, which results in $\sqrt{k}$ misses each of the $k^{3/2}$ invocations, totaling k^2 misses.
  - $Q_M(k) \leq (2^{3/2} + 2\sqrt{k})Q_M(\sqrt{k}) + k^2$.

Solving Recursion

$$
\begin{aligned}
Q_{\mathrm{M}}(k) &\le \left(2k^{3/2}+2\sqrt{k}\right) Q_{\mathrm{M}}(\sqrt{k}) + k^2 \\
&\le 2\left(k^{3/2}+\sqrt{k}\right)\left[\frac{ck^{3/2}\log_Z k}{2L} - A(\sqrt{k})\right] + k^2 \\
&\le ck^3 \log_Z k/L + k^2\left(1 + c\log_Z k/L\right) \\
&\qquad - \left(2k^{3/2}+2\sqrt{k}\right) A(\sqrt{k}) \\
&\le ck^3 \log_Z k/L - A(k)\ .\square
\end{aligned}
$$

Exercise

- Compute the number of computation in the merge sort discussed in the above.
- Compute the number of misses in a quick sort solution in the secondary memory algorithm.
- Do the same for bulb sort.