External Memory Algorithms

IT-C Course on Advanced Algorithms

Gerth Stølting Brodal

BRICS
Department of Computer Science
University of Aarhus

gerth@brics.dk

Lectures on External Memory Algorithms

March 20, 2001

- External memory model – parameters N, M, B, D
- Algorithms: scanning, merging, sorting, permutation
- Lower bounds: sorting, permutation

March 27, 2001

- B-trees
- Analysis of B-trees
- Oblivious B-trees

April 3, 2001

- Minimum spanning trees
- Functional approach
- Superphases and blocking values

Literature

Sorting

The Input/Output Complexity of Sorting and Related Problems
Alok Aggarwal and Jeffrey Scott Vitter
Communications of the ACM, 31(9):1116–1127, 1988

B-trees

Amortized Analysis of (a,b)-Trees
Gerth Stølting Brodal and Rolf Fagerberg
Note, 4 pages, 2000

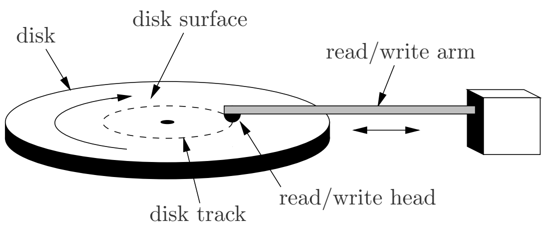
Cache-Oblivious B-Trees
Michael A. Bender, Erik Demain, and Martin Farach-Colton
In *Proc. 41th Annual Symposium on Foundations of Computer Science*, 399–409, 2000

MST

A Functional Approach to External Graph Algorithms
James Abello, Adam L. Buchsbaum, and Jeffery R. Westbrook
In *Proc. 6th Annual European Symposium on Algorithms*, LNCS 1461, 332–343, 1998

On External Memory MST, SSSP and Multi-way Planar Graph Separation
Lars Arge, Gerth Stølting Brodal, and Laura Toma
In *Proc. 7th Scandinavian Workshop on Algorithm Theory*, LNCS 1851, 433–447, 2000

Disk Drives

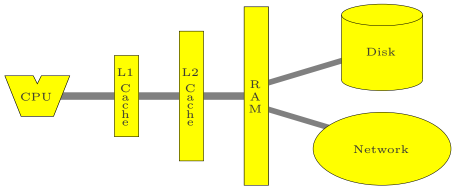


- Time for rotation ≈ Time for seek
- Amortize search time by large block transfer

$\boxed{\text{Time for rotation} \approx \text{Time for seek} \approx \text{Time to transfer data}}$

- Parallel disks

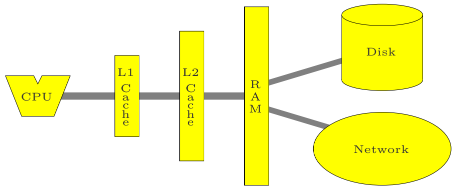
Trends in Implementation Technology



- Integrated circuit logic technology
  - Transistor count increases ≈ 60-80% per year
- Semiconductor DRAM
  - Density improves ≈ 60% per year
  - Cycle time improved ≈ 35% in 10 years
- Magnetic disk technology
  - Density improves ≈ 50% per year
  - Access time improved ≈ 35% in 10 years

Source: *Computer Architecture – A Quantitative Approach*, Hennessy & Patterson, 2nd. Ed. 1996

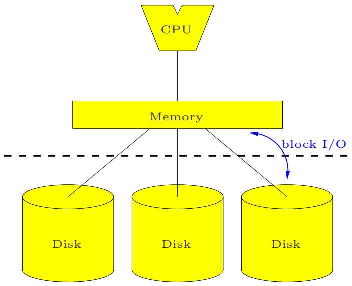
Trends in Implementation Technology



	L1 Cache	L2 Cache	Virtual memory
Block size (bytes)	4-32	32-256	4096-16384
Hit time (cycles)	1-2	6-15	10-100
Miss penalty (cycles)	8-66	30-200	700.000-6.000.000
Size	1-128KB	256KB-16MB	16-8192MB

Source: *Computer Architecture – A Quantitative Approach*, Hennessy & Patterson, 2nd. Ed. 1996

The I/O Model



N = problem size
M = memory size
B = size of disk block
D = number of disks

$$B \le M < N$$
$$D \le M/B$$

- One I/O moves B consecutive records from/to each disk
- Performance measures: # I/O steps, disk space, CPU time

Disk Striping

- Technique to make a one disk algorithm use multiple disks

  $f(N, B, M)$ I/Os for one disk $\Rightarrow$ $f(N, BD, M)$ I/Os for D disks
- Consider the external memory as one big (virtual) memory

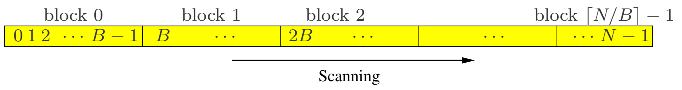
stripe	$\mathcal{D}_0$				$\mathcal{D}_1$				$\mathcal{D}_2$			
0	0	1	2	3	4	5	6	7	8	9	10	11
1	12	13	14	15	16	17	18	19	20	21	22	23
2	24	25	26	27	28	29	30	31	32	33	34	35
3	36	37	38	39	40	41	42	43	44	45	46	47
4	48	49	50	51	52	53	54	55	56	57	58	59
	...				...				...			

Layout for $D = 3$ and $B = 4$

- One I/O can read/write one stripe of size BD

Scanning

- Scanning an input of size N on one disk requires N/B I/Os

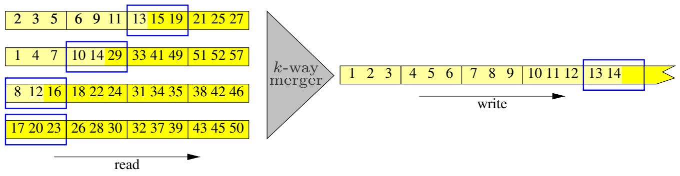


- Disk striping $\Rightarrow$ $N/(BD)$ I/Os

$$scan(N) = O\left(\frac{N}{BD}\right)$$

Merging

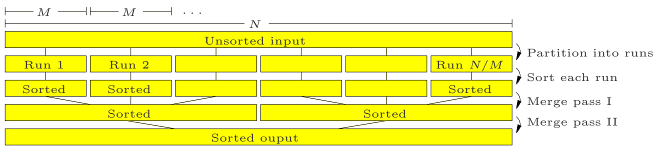
Merge $k \leq M/B - 1$ sorted lists with total size N



- Internal memory: one block of each input list and the output list
- Output one record at a time – read/write blocks on demand
- Total $2N/B$ I/Os for one disk
- Disk striping $\Rightarrow$ $2N/(BD)$ I/Os for D disks $k \leq M/(BD) - 1$

Merge Sort

- Form N/M runs each of size M
  - Read each run into internal memory
  - Sort run internally
  - Write sorted run to disk
- Merge $k = M/B - 1$ runs at a time



- Total number of I/Os for one disk

$$\#\,\text{merge passes} = \log_k \frac{N}{M} = \Theta(\log_{M/B} \frac{N}{M})$$

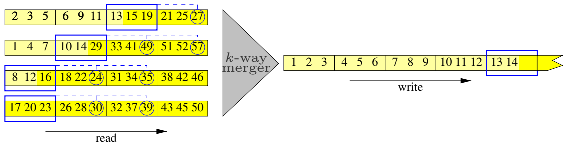
$$O(\frac{N}{B}(1 + \log_{M/B} \frac{N}{M})) = O(\frac{N}{B} \log_{M/B} \frac{N}{B}) \text{ I/Os}$$

- Disk striping $\Rightarrow$ $O(\frac{N}{BD} \log_{M/(BD)} \frac{N}{BD})$ I/Os for D disks $BD \rightarrow B$?

Merge Sort – Aggarwal & Vitter

- Stronger model: P parallel block reads/writes to one disk
- Merging: $P \leq B/2$, $k = M/B - 1$
  - Each block stores $B/2$ records and the maximum of each of the next P blocks in the run

  $\Rightarrow$ possible to read/write the next P blocks in a single I/O



$\Rightarrow$ $O(\frac{N}{BP})$ I/Os

- Merging: $P > B/2$, $B' \leftarrow 2\sqrt{BP}$, $P' \leftarrow \frac{1}{2}\sqrt{BP}$, $k = \Theta(M/B')$

  $\Rightarrow$ $O(\frac{N}{B'P'}) = O(\frac{N}{BP})$ I/Os
- Merge Sort: $O(\frac{N}{BP} \log_{M/B} \frac{N}{B})$ I/Os $P \rightarrow D$?

Merge Sort on Multiple Disks

- Randomized technique to convert algorithms for the Aggarwal & Vitter model to the D disk I/O model [Sanders et al. 2000]

  $\Rightarrow$ expected $O(\frac{N}{BD} \log_{M/B} \frac{N}{B})$ I/Os
- "Simple Randomized Merge Sort" [Barve et al. 1997] (not optimal for all parameter values)
- "Greed Sort" [Nodine & Vitter 1995] Deterministic $O(\frac{N}{BD} \log_{M/B} \frac{N}{B})$ I/Os
- ...

$$sort(N) = O\left(\frac{N}{BD} \log_{M/B} \frac{N}{B}\right)$$

Permutation

Sort N records with keys $1, 2, \ldots, N$ forming a permutation

Algorithm 1

Move the N records one-by-one $\Rightarrow O(N)$ I/Os on one disk

For D disks it is possible to achieve $O(\frac{N}{D})$ I/Os

Algorithm 2

Sorting $\Rightarrow O(\frac{N}{BD} \log_{M/B} \frac{N}{B})$ I/Os

$$permute(N) = O\left(\min\left\{\frac{N}{D}, \frac{N}{BD} \log_{M/B} \frac{N}{B}\right\}\right)$$

Use Algorithm 2 if $BD \geq \log_{M/B} \frac{N}{B}$

Summary – Upper Bounds

$$\begin{aligned} scan(N) &= O\left(\frac{N}{BD}\right) \\ sort(N) &= O\left(\frac{N}{BD} \log_{M/B} \frac{N}{B}\right) \\ permute(N) &= O\left(\min\left\{\frac{N}{D}, \frac{N}{BD} \log_{M/B} \frac{N}{B}\right\}\right) \end{aligned}$$

P A U S E

Lower Bound for Sorting

- Adversary argument for the comparison model
- Count # input permutations consistent with the comparisons done
- Invariant
  - I/Os read/write complete tracks (i.e. blocks of size B starting at a multiple of B in external memory)
  - The total order of the records in internal memory is known
  - The total order of the records in a block written to external memory is known
- Initially all $N!$ possible input permutations consistent
- Block write $\Rightarrow$ # consistent input permutations unchanged

Lower Bound for Sorting – II

- Block read of track not read before (at most N/B such I/Os)
  - $B!$ possible permutations among records read from track
  - $\binom{M}{B}$ possible ways to merge the content of the track read with the records in internal memory
  - $B!\binom{M}{B}$ possible outcomes of the comparisons performed
  - Adversary chooses the outcome that maximizes the number of consistent input permutations
- Block read of track previously written to external memory
  - Total order of records in the track read is already known
  - At most $\binom{M}{B}$ possible ways to merge the content of the track read with the records in internal memory
  - Adversary chooses the outcome that maximizes the number of consistent permutations
- # consistent input permutations after T I/Os is $\geq \dfrac{N!}{(B!)^{N/B} \cdot \binom{M}{B}^T}$

Sorting – Lower Bound Calculations

consistent input permutations after T I/Os is $\geq \dfrac{N!}{(B!)^{N/B} \cdot \binom{M}{B}^T}$

If the input permutation is uniquely identified then $\frac{N!}{(B!)^{N/B} \cdot \binom{M}{B}^T} < 2$

$$\boxed{\begin{array}{l} \log k! = k \log k - \frac{1}{\ln 2}k + O(\log k) \\ \log \binom{M}{B} \leq B \log \frac{M}{B} + \frac{1}{\ln 2}B \end{array}}$$

$$\begin{aligned} T &> \frac{\log N! - \frac{N}{B}\log B! - 1}{\log \binom{M}{B}} \\ &\geq \frac{N \log N - \frac{1}{\ln 2}N - \frac{N}{B}(B \log B - \frac{1}{\ln 2}B + O(\log B)) - 1}{B \log \frac{M}{B} + \frac{1}{\ln 2}B} \\ &= \frac{N \log \frac{N}{B} - \frac{N}{B}O(\log B) - 1}{B \log \frac{M}{B} + \frac{1}{\ln 2}B} \\ &= \Omega\left(\frac{N \log \frac{N}{B}}{B \log \frac{M}{B}}\right) \\ &= \Omega\left(\frac{N}{B}\log_{M/B}\frac{N}{B}\right) \quad \square \end{aligned}$$

log-Lemmas

Lemma $k \log k - \frac{1}{\ln 2}k \leq \log k! \leq k \log k - \frac{1}{\ln 2}\sum_{i=1}^{k}\frac{i-1}{i}$

Proof Proof by induction. Case $k = 1$ ok.

$$\boxed{\sum_{i=1}^{k}\frac{i-1}{i} = k - H(k)}$$

$$\begin{aligned} &(k+1)\log(k+1) - \tfrac{1}{\ln 2}(k+1) \\ &= (k+1)\log(k+1) - \tfrac{1}{\ln 2}k - k\tfrac{1}{k \ln 2} \\ &\leq (k+1)\log(k+1) - \tfrac{1}{\ln 2}k - k(\log(k+1) - \log k) \\ &= \log(k+1) + k\log k - \tfrac{1}{\ln 2}k \\ &\leq \log(k+1) + \log k! = \log(k+1)! \\ &\leq \log(k+1) + k \log k - \tfrac{1}{\ln 2}\textstyle\sum_{i=1}^{k}\frac{i-1}{i} \\ &= (k+1)\log(k+1) - \tfrac{1}{\ln 2}\textstyle\sum_{i=1}^{k}\frac{i-1}{i} - k(\log(k+1) - \log k) \\ &\leq (k+1)\log(k+1) - \tfrac{1}{\ln 2}\textstyle\sum_{i=1}^{k}\frac{i-1}{i} - k\frac{1}{(k+1)\ln 2} \\ &= (k+1)\log(k+1) - \tfrac{1}{\ln 2}\textstyle\sum_{i=1}^{k+1}\frac{i-1}{i} \quad \square \end{aligned}$$

Corollary $\log k! = k \log k - \frac{1}{\ln 2}k + O(\log k)$

Lemma $\log \binom{M}{B} \leq B \log \frac{M}{B} + \frac{1}{\ln 2}B$

Proof $\log \binom{M}{B} = \log \frac{M(M-1)\cdots(M-B+1)}{B!} \leq \log \frac{M^B}{B!} = B \log M - \log B!$

$\leq B \log M - B \log B + \frac{1}{\ln 2}B = B \log \frac{M}{B} + \frac{1}{\ln 2}B$ $\square$

Sorting

- Comparison based external memory sorting requires

$$sort(N) = \Omega\left(\frac{N}{B}\log_{M/B}\frac{N}{B}\right) \text{ I/Os}$$

- For D disks divide the above single disk lower bound by D

$$sort(N) = \Omega\left(\frac{N}{BD}\log_{M/B}\frac{N}{B}\right) \text{ I/Os}$$

Lower Bound for Permutation

Simple I/O model

- I/O moves a track of records (possible **nil** records) between internal and external memory
  - Destination of a move must contain **nil** before the move
  - Source contains **nil** after the move
- Exactly one copy of each record
- Records indivisible
- View internal and external memory as one big extended memory

Lemma

If a permutation can be achieved with k I/Os in the I/O model then the permutation can also be obtained with $O(k)$ simple I/Os

Lower Bound for Permutation – II

- Count maximal # permutations achievable in extended memory with T simple I/Os (one permutation for $T = 0$ and $T \leq 4N$)
- Reading a track for the first time (at most such reads) increases # permutations at most with a factor

$$\left(\frac{N}{B} + T\right) B! \binom{M}{B} \leq 5N \cdot \binom{M}{B}$$

- Reading a track previously written to external memory increases # permutations at most with a factor

$$\left(\frac{N}{B} + T\right) \binom{M}{B} \leq 5N \cdot \binom{M}{B}$$

- The tth track write increases # permutations with at most a factor

$$N/B + t \leq 5N$$

Permutation – Lower Bound Calculations

To be able to generate all $N!$ possible permutations we require

$$(B!)^{N/B} \left(5N \binom{M}{B}\right)^T \geq N!$$

implying

$$\begin{aligned}
T &\geq \frac{\log N! - \frac{N}{B}\log(B!)}{\log(5N) + \log\binom{M}{B}} \\
&\geq \frac{N\log N + \frac{1}{\ln 2}N - \frac{N}{B}\left(B\log B - \frac{1}{\ln 2}B + O(\log B)\right)}{\log(5N) + \log\binom{M}{B}} \\
&= \frac{N\log\frac{N}{B} - O(\log B)}{\log(5N) + \log\binom{M}{B}} \\
&= \Omega\left(\min\left\{\frac{N\log\frac{N}{B}}{\log N}, \frac{N\log\frac{N}{B}}{\log\binom{M}{B}}\right\}\right) \\
&= \Omega\left(\min\left\{N, \frac{N\log\frac{N}{B}}{B\log\frac{M}{B}}\right\}\right) \\
&= \Omega\left(\min\left\{N, \frac{N}{B}\log_{M/B}\frac{N}{B}\right\}\right)
\end{aligned}$$

□

Permutation

- Constructing permutations in external memory where records are indivisible requires

$$\mathit{permute}(N) = \Omega\left(\min\left\{N, \frac{N}{B}\log_{M/B}\frac{N}{B}\right\}\right) \text{ I/Os}$$

- For D disks divide the above single disk lower bound by D

$$\mathit{permute}(N) = \Omega\left(\min\left\{\frac{N}{D}, \frac{N}{BD}\log_{M/B}\frac{N}{B}\right\}\right) \text{ I/Os}$$

- This is also a lower bound for non-comparison based sorting

Conclusion

$$
\begin{aligned}
scan(N) &= \Theta\left(\frac{N}{BD}\right) \\
sort(N) &= \Theta\left(\frac{N}{BD}\log_{M/B}\frac{N}{B}\right) \\
permute(N) &= \Theta\left(\min\left\{\frac{N}{D}, \frac{N}{BD}\log_{M/B}\frac{N}{B}\right\}\right)
\end{aligned}
$$

THE END