Projects

Permuting data on parallel disks

Disk access times $\approx 10^{-2}$ sec
Data transfer rate $\approx 10^{-6}$ words/sec

"Want to do as few disk accesses as possible.

Convenient engineering assumption:

Disk is broken into large fixed-size blocks, e.g., of 1000 words.

\[ \text{Tracks} \]

\[ \frac{N}{PB} \]

\[ \text{Disks} \]

\[ \text{P disks} \]

\[ \text{N data records} \]

Block $= B$ data records

If parallel I/O's to read all data $= N/PB$.

Computer memory holds $M$ data records total.
Assume $M \gg PB$.

Permuting disk blocks

- Offline (perm fixed in advance)

\[ \begin{array}{cccc}
0 & 1 & 2 & 3 \\
3,2 & 1,2 & 3,1 & 2,3 \\
A & E & E & E \\
1,0 & 1,4 & 3,0 & 2,2 \\
B & A & D & B \\
2,4 & 1,3 & 0,0 & 1,0 \\
C & D & A & C \\
2,1 & 3,3 & 3,4 & 2,0 \\
D & C & B & A \\
0,1 & 0,3 & 0,4 & 0,2 \\
E & B & C & D
\end{array} \]

- $B = 1$

Theorem, can permute with $O(N/P)$ parallel I/O's
(not in place)
Conflict graph

Source disk

Dest disk

All degrees
\[ n = N/p. \]

Fact: Any d-regular bipartite multigraph can be edge-colored with d colors. (Color = step at which block is moved.)


How do we know perfect matching exists?

Hall's Thm.
For \( A \subseteq V_1 \), let \( N(A) \subseteq V_2 \) be the set of neighbors of \( A \). Then, a perfect matching exists if \( |N(A)| \geq |A| \) in \( G \).

Proof:
Let \( \text{f} \) be max flow \( = c(s,t) \) for some cut \( (s, t) \) by maxflow-mincut thm.
Let \( A = S \cap V_1 \). Since edges from \( V \) to \( V_2 \) have 0 capacity, \( N(A) \leq S \). Also, \( N(V_1 - A) \leq T \).

\[ c(s,t) \geq |V_1 - A| + |N(A)| \geq |V_1 - A| + |A| \]
Sorting (Vitter et al.)

\[ O\left(\frac{N}{PB}, \frac{\log(N/M)}{\log(M/B)}\right) \] IO's

Idea: Internal sort \( M \) records at a time into \( N/M \) runs. Merge runs.

Would like to merge \( M/B \) runs at a time.

\[ \frac{N}{PB} \rightarrow \frac{M}{B} \rightarrow \frac{N}{PB} \rightarrow \frac{M}{B} \rightarrow \log_{M/B}(N/M) \rightarrow \frac{N}{M} \text{ leaves} \]

Total \( \frac{N}{PB}, \frac{\log(N/M)}{\log(M/B)} \)

Problem: Can only read 1 block/run
- All of one run may be smaller than others.

Solution:
- Merge \( \sqrt{M/B} \) runs at a time (Depth of rec. doubled).
- Keep track of which blocks to read next in table.
- "Sloppy" merge. Clean up with \( O(N/PB) \) IO's.