Review of *The Input/Output Complexity of Sorting and Related Problems* (Aggarwal and Vitter, 1988)

Andrew Feldman

MIT 6.827 Spring 2022

Background

- Aggarwal and Vitter – acceptable runtime for sorting and sorting related tasks
- I/O bound scenario
  - Magnetic memory
- Approach 1 – system or hardware architecture
- **Approach 2 – algorithms => theoretical bounds**
  - Previously attempted by Floyd
- Data parallelism
  - Read data in blocks
  - Read multiple blocks

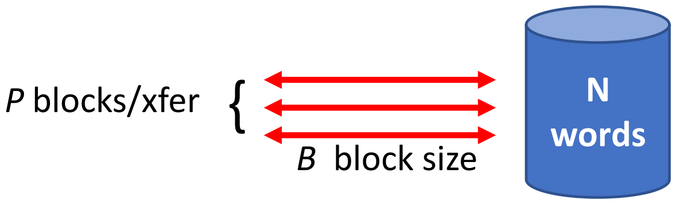
Background

- Scaling input size
  - Bank scenario – sort 2 million records overnight
  - Scaling would make this unattainable

Main results

	Lower- and upper-bound
Sorting	$\theta(\frac{N}{PB}\frac{\log(1+N/B)}{\log(1+M/B)})$
FFT	
Permutation network	$\theta(\min\{\frac{N}{P},\frac{N}{PB}\frac{\log(1+N/B)}{\log(1+M/B)}\})$
Matrix transposition	$\theta(\frac{N}{PB}\frac{\log\min\{M, 1+\ \min\{p,q\}, 1+N/B\}}{\log(1+M/B)})$

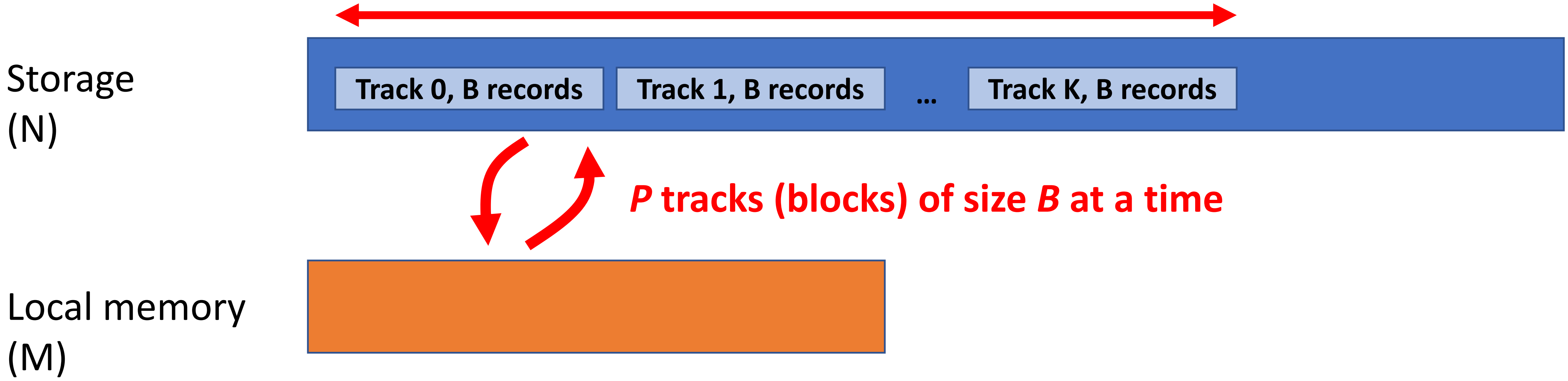
External storage model:



Lumped memory model:

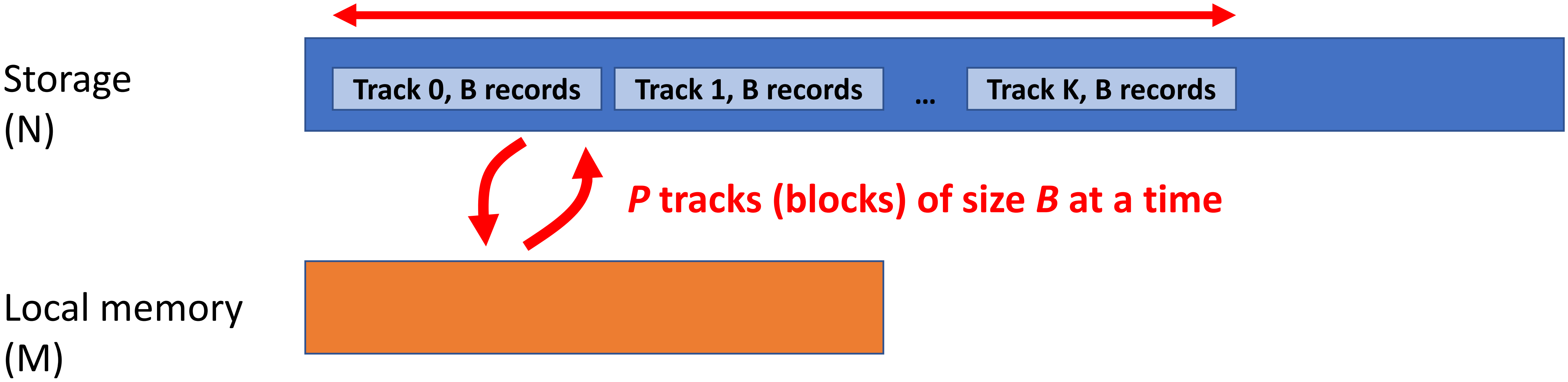
- Asymptotic I/O complexity of sorting, FFT, permutation, and matrix transposition
- Tight (same constant for lower, upper bounds) when **P = 1**

Memory model



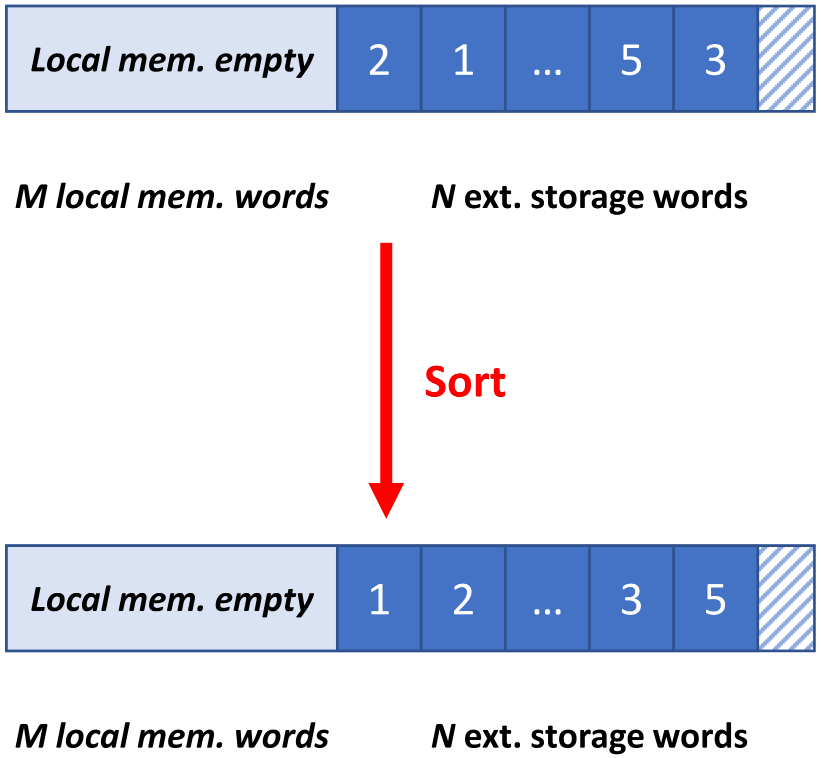
- N words of storage
- M words of local memory
- Tracks – contiguous blocks of k records in storage
- *Simple* - given record is only in one location at any given time

Memory model



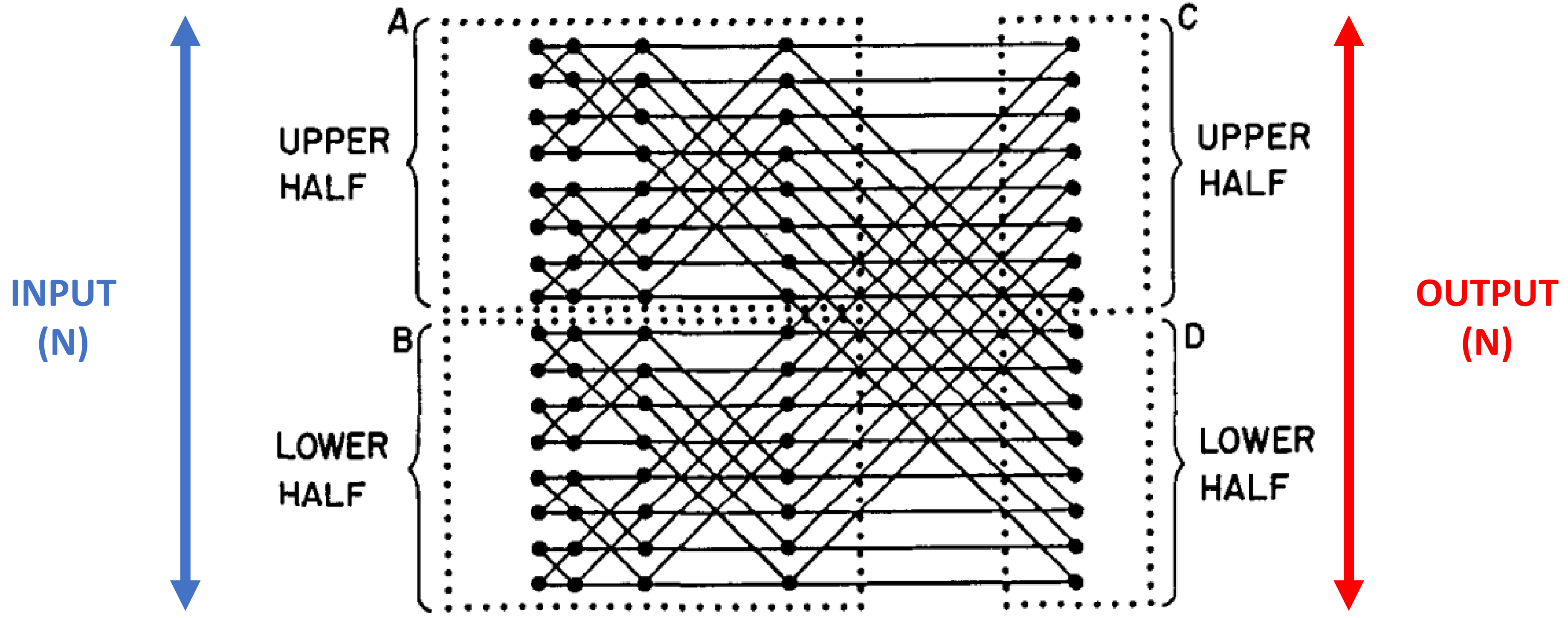
- After first input, records are indivisible

Sorting



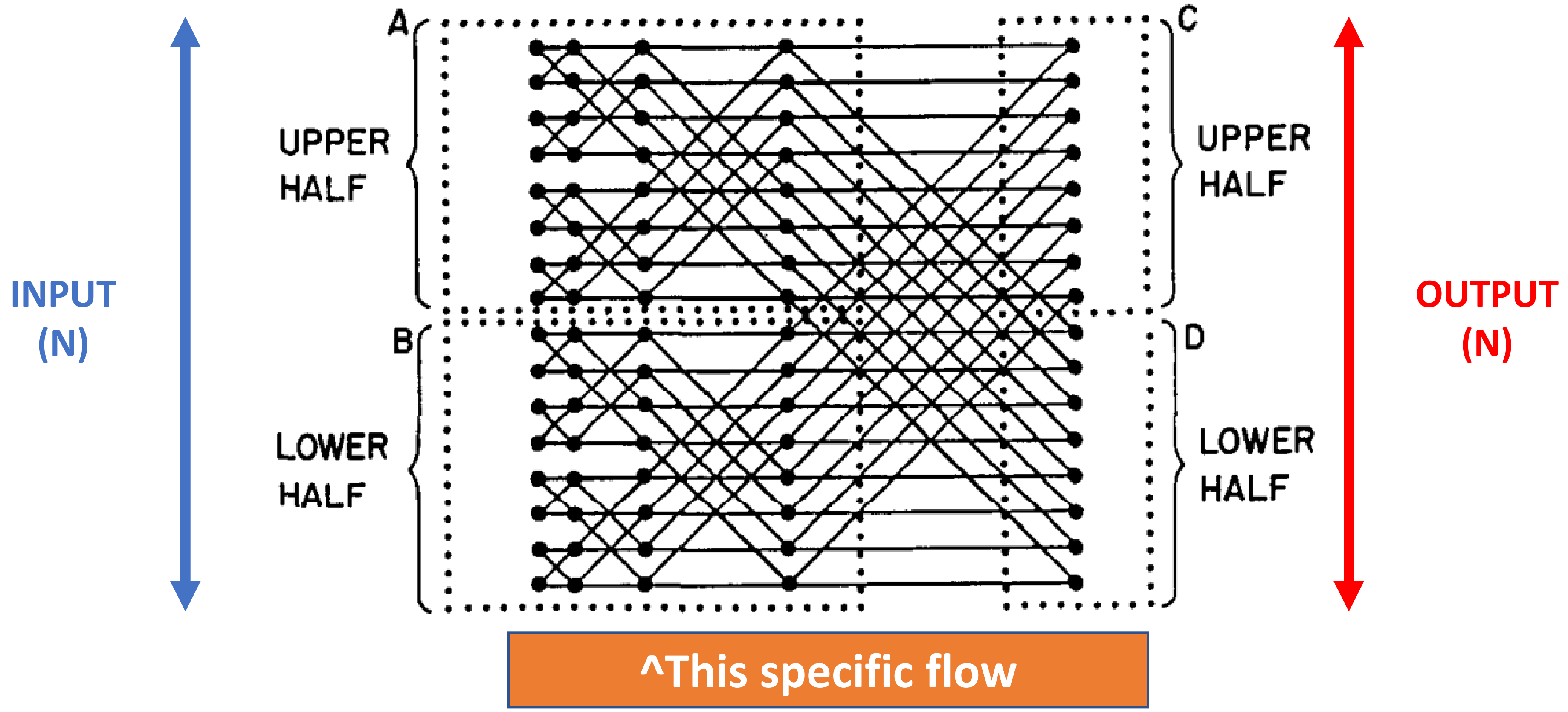
- Results must be ordered
- Results need not be contiguous

Permutation matrix digraph



- Diagraph w/ >= log N columns
- At each layer, pairs of nodes can be optionally swapped
- Pairs are known @ computation time

FFT digraph



- Digraph w/ log N columns
- Divide-and conquer

Matrix transposition

- Input: row-major matrix, # elements = N
- Output:
  - Row-major matrix transpose
  - Column-major input matrix

Approach

- Demonstrate worst-case, average lower-bound for permutation
- Use sorting to develop the w.c./average upp-bound for permutation
- Extend to permutation matrix (limited data movement possibilities)
- Extend to FFT (based on permutation matrix)
- Extend to matrix transpose & sort
- Show that for P=1, bounds are tight

Key lemma

- To implement N-element permutation…
- There exists a computation strategy employing **only** simple I/Os

Worst-/average-case **lower-bound**

Worst-/average-case **upper-bound**

$$\Theta\left(\min\left\{\frac{N}{P}, \frac{N}{PB}\frac{\log(1+N/B)}{\log(1+M/B)}\right\}\right).$$

- Permutation is a special case of sorting

Evaluation

- Strengths –
  - Identified shared relationship between permutations, sorting & other algoritms explored => generalized bound
  - Tight bound for P=1
- Weakness –
  - Not tested experimentally
- Originality –
  - Different from prior papers focused on architecture
  - Expands on the work of Floyd

Future work

- Challenge: remove assumption that records are indivisible
- Data movement within cache hierarchy