On The Limits Of Cache Oblivious Matrix Transposition

Francesco Silvestri

francesco.silvestri@dei.unipd.it

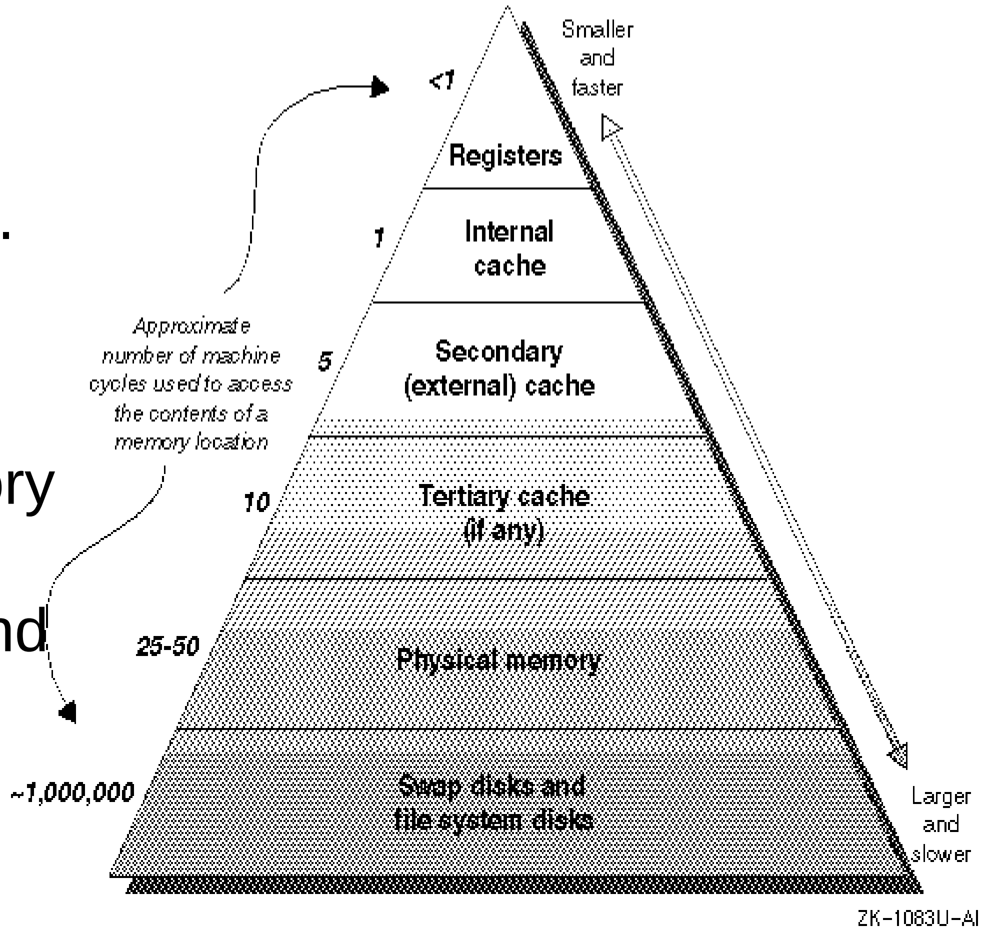
DEPARTMENT OF INFORMATION ENGINEERING
UNIVERSITY OF PADOVA

Overview

- Models for memory hierarchy;
- Statement of the problem;
- Negative result on optimal cache-oblivious matrix transposition;
- Conclusions.

The memory hierarchy

- Modern platforms feature a hierarchical cascade of memories.
- The farther the memory from the CPU, the bigger the capacity and the access time!



Locality

- In good algorithms:
  - the same data are frequently reused within a short time interval

Temporal Locality of Reference

  - data stored at consecutive addresses are involved in consecutive operations

Spatial Locality of Reference

- Many models account for these two properties.

Models

- External Memory (EM) *[Aggarwal, Vitter, 1988]*
  - Represents DISK-RAM hierarchy.
  - Arbitrarily large disk, RAM of M words.
  - Data are transferred in blocks of B consecutive words.
  - Block transfers are explicitly controlled by the program.
  - *I/O Complexity*: number of accesses to disk.

Models (cont'd)

- Ideal Cache (IC) *[Frigo et al., 1999]*
  - Represents RAM-CACHE hierarchy.
  - Arbitrarily large RAM, cache of M words.
  - Cache:
    - Organized into M/B blocks of B words,
    - Fully associative.
  - Data are transferred in blocks of B consecutive words.
  - Block transfers are automatically controlled by hardware:
    - Optimal offline strategy for block replacement.
  - *Cache Complexity*: number of accesses to RAM

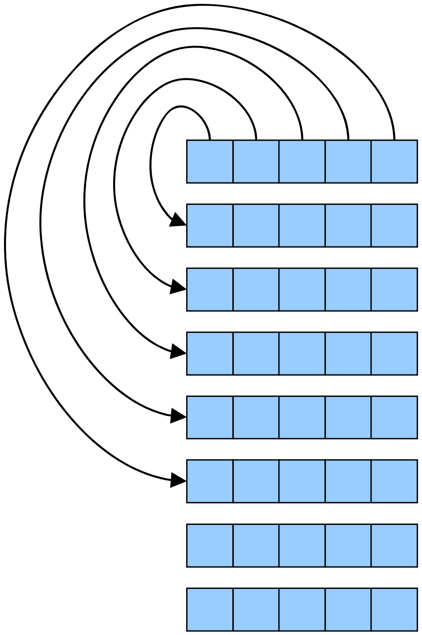
Cache-Obliviousness

- An algorithm for IC is *cache-oblivious* (CO) if its specification is independent of the two parameters M and B.
- An algorithm is *cache-aware* (CA) otherwise.
- **CO algorithms adapt automatically to the actual platform in which they run:**
  - **Desirable in the overlay computing scenario**
- Optimal CO algorithm: best cache complexity on each $\mathrm{IC}(M,B)$ model.

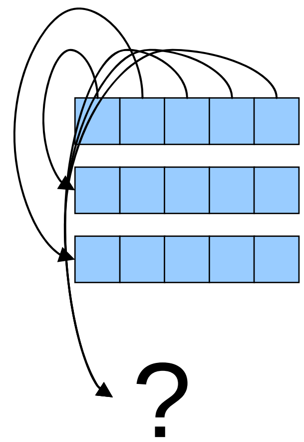
Tall cache

- Many optimal CO algorithms in literature. Most of them require the *tall cache assumption* (TCA):

$$M \geq B^2 \Longleftrightarrow \frac{M}{B} \geq B$$



Tall cache:
$M/B \geq B$



Short cache:
$M/B < B$

Question

Does every problem admit an
optimal CO algorithm
which does not require the TCA?

Known results

- *[Brodal, Fagerberg, 2003]*:
  - There is no optimal CO algorithm for the sorting problem without the TCA;
  - There is no optimal CO algorithm for general permutations even with the TCA.
- *[Bilardi, Peserico, 2001]*:
  - Similar results in a model without spatial locality (HMM), in the context of DAG computations.

Our result

<u>There is no optimal cache-oblivious algorithm for matrix transposition without the TCA</u>

Main ingredient: exploit $\mathcal{EM}$ lower bound arguments through a simulation technique.

Matrix Transposition (MT)

- Input: an $N^{1/2} \times N^{1/2}$ matrix A in row major
- Output: A^T in row major
- Cache (I/O) complexity:

$$\Omega\left(\frac{N \log M}{B \log\left(1 + \frac{M}{B}\right)}\right)$$

- There is an optimal CA algorithm $\forall$ IC(M,B) *[Aggarwal, Vitter, 1988]*.
- There is an optimal CO algorithm $\forall$ IC(M,B) that satisfies the TCA *[Frigo et al., 1999]*.

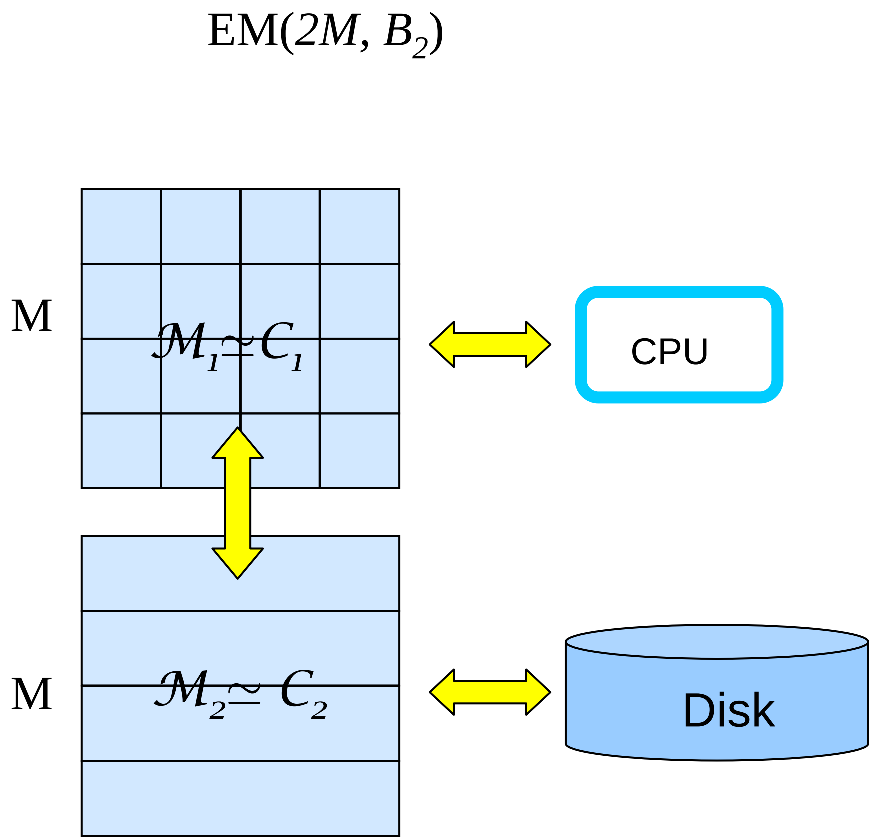
Matrix transposition (cont'd)

- $\mathcal{A}$ is an <u>optimal</u> CO algorithm for MT without TCA.
- C_1: tall $\mathrm{IC}(M,B_1)$, C_2: short $\mathrm{IC}(M,B_2)$.
  - t_1, t_2: (optimal) cache complexities of $\mathcal{A}$ on C_1 and C_2, respectively.
  - Sequence of operations does not change.
  - Sequence of I/Os could be different.
  - **When an operation requires the word x, the B_1-block and B_2-block containing x must be in C_1 and C_2, respectively.**

The simulation technique

- $\mathcal{A}'$: new MT algorithm for $\mathrm{EM}(2M, B_2)$.
- $\mathcal{A}'$ simulates the executions of $\mathcal{A}$ on both C_1 and C_2 at the same time:
  - Divide the memory in two segments $\mathcal{M}_1$ and $\mathcal{M}_2$:
    - $\mathcal{M}_1$ represents C_1, $\mathcal{M}_2$ represents C_2.
  - Operations on $\mathcal{M}_1$.
  - I/Os on $\mathcal{M}_2$.

The simulation technique (cont'd)



- I/Os of B_2-blocks in C_2: I/Os between $\mathcal{M}_2$ and the disk.
- I/Os of B_1-blocks in C_1: words exchanged between $\mathcal{M}_2$ and $\mathcal{M}_1$.

The new algorithm

- The I/O complexity of $\mathcal{A}$' is $T=\Theta(t_2)$.
- $K = \frac{t_1 B_1}{B_2}$
- **Crux: change $\mathcal{A}$' in such a way that O(K) words are exchanged between $\mathcal{M}_1$ and a B_2-block in $\mathcal{M}_2$, before this block is removed from $\mathcal{M}_2$.**

Potential function

- Adaptation of the argument used in *[Aggarwal,Vitter,1988]* to lower bound the I/Os of $\mathcal{A}$'.
- Potential function after t I/Os ($POT(t)$)
  - It measures the progress of an MT algorithm for EM(2M, B_2).
  - $POT(0) = 0;\ POT(T) = N \log B_2$
- The rate of POT is limited by the amount of words exchanged between $\mathcal{M}_1$ and $\mathcal{M}_2$:

$$\nabla POT = POT(t) - POT(t-1) = O\left(K \log \frac{M}{K}\right)$$

The grand finale

- $T=\Theta(t_2)$, then ($\exists$ a constant $0 < \gamma < 1$):

$$T \cdot \nabla POT \geq POT(T) \Longrightarrow t_2 \in \Omega\left(N\frac{B_2^\gamma}{M}\right)$$

- If $B_2=\Theta(M)$:

$$t_2 \in \Omega\left(\frac{N}{M^{1-\gamma}}\right) \in \omega\left(N\frac{\log M}{M}\right)$$

Lower bound

A contradiction:

There is no optimal cache-oblivious algorithm for MT without the TCA

Conclusions

- The TCA is a reasonable assumption, and we do need it for CO optimality for certain problems:
  - Sorting $\rightarrow$ Brodal and Fagerberg, 2003
  - Matrix Transposition $\rightarrow$ us
  - Matrix Multiplication $\rightarrow$?
  - FFT $\rightarrow$?

- The simulation technique can be applied to other problems.
- We still don't fully understand WHY the TCA is needed!

Any questions?

attention!