Algorithms and Architectures

Michael T. Goodrich
CS 165
Univ. of California, Irvine

Algorithm  Architecture
Computational Models

- We often skip this step, but an algorithm description requires a computational model.
- There are several computational models, which are based on various computational architectures.
Turing Machine

- Mathematical model of computation, 1936
- **Complexity-theoretic Church–Turing thesis:** Any polynomial-time computable function (for any computational model) has a polynomial-time algorithm on a Turing machine.

https://mediartinnovation.com/2014/05/26/alan-turing-turing-machine-1936/

Alan Turing

https://en.wikipedia.org/wiki/Alan_Turing
Turing Machine

- Not a real-world computer, but imitations exist.
- Any computational model/system that can simulate a Turing machine can compute any computable function: Turing-complete.

https://en.wikipedia.org/wiki/Turing_machine

not Alan Turing
ENIAC

- The first programmable, electronic, general-purpose digital computer, completed in 1945.
- It was Turing-complete

https://en.wikipedia.org/wiki/ENIAC

Presper Eckert and John Mauchly
The von Neumann Architecture

ENIAC gave rise to a computational model called the von Neumann Architecture.

Memory stores both data and instructions

https://en.wikipedia.org/wiki/Von_Neumann_architecture

John von Neumann
Random Access Machine (RAM)

- A refinement of the von Neumann architecture.

https://www.geeksforgeeks.org/what-is-random-access-machine/
Programming Languages

- Machine-independent languages written by humans and compiled into instructions for specific computer architectures.

The FLOW-MATIC programming language she created was later extended to create COBOL, a widely used high-level language for business applications.

Grace Hopper
RAM Programming Primitives

- Language primitives for the RAM model are based on high-level programming languages, which were defined for the von Neumann architecture.

<table>
<thead>
<tr>
<th>Operations</th>
<th>Number of steps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arithmetic operations: + - * /</td>
<td>1</td>
</tr>
<tr>
<td>Logical operations: AND, OR, NOT</td>
<td>1</td>
</tr>
<tr>
<td>Conditional:</td>
<td></td>
</tr>
<tr>
<td>Comparison: a &lt; b</td>
<td>1</td>
</tr>
<tr>
<td>Conditional branching: if</td>
<td></td>
</tr>
<tr>
<td>Subroutine calls: call, return</td>
<td>1</td>
</tr>
<tr>
<td>Loops</td>
<td>Depends on the number of loop iterations and loop condition</td>
</tr>
<tr>
<td>Subprogram</td>
<td>Depends on the nature of the subprogram</td>
</tr>
<tr>
<td>Memory access: Read, Write</td>
<td>1</td>
</tr>
</tbody>
</table>

https://medium.com/@exploreintellect/what-is-the-ram-model-of-computation-a5e4a7ce22b4
Moore’s “law” is the observation that the number of transistors in an integrated circuit doubles about every two years.
Moore’s Law Misquote

- Not clock speed

Grace Hopper gave out “nanoseconds” to prove this point.
Word RAM Model

- The **word RAM** (word random-access machine) model is a model of computation in which a random-access machine can do arithmetic and bitwise operations on a word of \( w \) bits in constant time.

- Examples: bitwise AND, OR, XOR, MSB
Example Word RAM Algorithm

- Given two arrays, A and B, of n bits, compute the first bit where A and B differ.
  1. Compute $C = A \text{ XOR } B$. Time: $O(n/w)$
  2. Repeatedly test each word of MSB for equality with 0. Time: $O(n/w)$.
  3. For the first word, $c$, in $C$ that is not 0, compute $\text{MSB}(c)$.
- Total running time: $O(n/w)$. 

Algorithms and Architectures
Understanding the Orders of Magnitude

Internal memory: 10 ns

External memory: 10 ms

1 million times slower!
The Memory Hierarchy

- The trade-off of size and speed

- Capacity: 1 TB, 4 GB

- Latency: 10 ms, 10 ns, 5 ns, 0.5 ns
Analogy: Cooking Eggs

- Suppose Anna is cooking eggs in Irvine and wants to add salt and pepper.

- Suppose it takes her 10 seconds to go to her pantry, get salt and pepper and add them to her eggs.
Analogy: Cooking Eggs

- If going to her pantry is like a computer going to internal memory, then what would be analogous to going to external memory?
Analogy: Cooking Eggs

If going to her pantry is like a computer going to internal memory, then what would be analogous to going to external memory?

Walking to Chicago, buying salt and pepper, and walking back
External Memory Model

- External memory identifies the frontier between the highest two layers in the memory hierarchy for a particular data set.
  - \( B \) = block size
  - \( M \) = “internal” memory size
    - \( m = M/B \) (number of internal blocks)
  - \( N \) = “external” input size
    - \( n = N/B \) (number of input blocks)

- The model only counts the number of reads and writes to external memory (I/Os). All other costs are ignored.
B-Trees

- A version of the \((a,b)\) tree data structure, which is the best-known method for maintaining a map in external memory, is a “B-tree.”
- A B-tree of order \(d\) is an \((a,b)\) tree with \(a = d/2\) and \(b = d\).
Proposition 15.2: A B-tree with \( n \) entries has I/O complexity \( O(\log_B n) \) for search or update operation, and uses \( O(n/B) \) blocks, where \( B \) is the size of a block.

- **Proof:**
  - Each time we access a node to perform a search or an update operation, we need only perform a single disk transfer.
  - Each search or update requires that we examine at most \( O(1) \) nodes for each level of the tree.
External-Memory Sorting

- Which of these sorting algorithms is good/bad in external memory?
  - Insertion-sort
  - Heapsort
  - Shellsort
  - Mergesort
  - Quicksort
Better External-Memory Sorting

- **Multi-way Merge-sort:**
  - Merge $M/B$ sorted subarrays instead of 2.
  
  Number of I/Os: $O((N/B) \log_{M/B} (N/B))$.
  - This is optimal.
Parallel Random Access Machine (PRAM)

- Synchronous shared memory model.
  - **Exclusive Read (ER):** \( p \) processors can simultaneously read the content of \( p \) distinct memory locations.
  - **Concurrent Read (CR):** \( p \) processors can simultaneously read the content of \( p' \) memory locations, where \( p' < p \).
  - **Exclusive Write (EW):** \( p \) processors can simultaneously write the content of \( p \) distinct memory locations.
  - **Concurrent Write (CW):** \( p \) processors can simultaneously write the content of \( p' \) memory locations, where \( p' < p \).
PRAM Algorithms

- CRCW PRAM can compute OR of n bits in $O(1)$ time with n processors.
- CRCW PRAM can compute Min of n numbers in $O(1)$ time with $n^2$ processors.
- Merge two sorted lists of n elements in $O(\log n)$ time with n processors in CREW PRAM.
- Sort n numbers in $O(\log^2 n)$ time with n processors by parallel merge-sort in CREW PRAM.
Parallel External Memory (PEM)

- In joint work, we defined a parallel external memory model and designed efficient algorithms for it.
PEM Sorting Result

- Sorting with optimal parallel I/O complexity:

\[ O \left( \frac{N}{PB} \log \frac{M}{B} \frac{N}{B} \right) \]

- \( N \): Input size
- \( P \): # of processors
- \( M \): memory (cache) size
- \( B \): block (cache line) size