How to Write Fast Code
18-645, spring 2008
26th Lecture, Apr. 21st

Instructor: Markus Püschel
TAs: Srinivas Chellappa (Vas) and Frédéric de Mesmay (Fred)
Course Evaluations

- Are open now
- Please fill it out
Research Project

- Project expectations
- Paper templates and instructions on the website
- Poster template soon

• Today
• Papers due (6 pm)
• Last class: poster session
  Scaife Hall
  5:30 – 8:30 pm
• Due:
  • Final papers
  • Final code
Today

- Sorting
  (Example of a non-numerical problem)
Sorting

- Fundamental problem in computer science
  - Extensively studied
  - Many different algorithms (Wikipedia)

- Comparison based algorithms
  - Complexity: $\Omega(n \log(n))$
  - Quicksort
  - Mergesort
  - Insertion sort
  - Sorting networks

- Other algorithms
  - Radix sort

- How to make sorting fast?
Performance Issues

- Many algorithms to choose from
- Usually not optimized for the memory hierarchy

Performance Issues

- Performance may depend on
  - the distribution of input data
  - the computing platform

Sorting Algorithms and Memory Hierarchy Optimizations

- Quicksort
- Mergesort
- Insertion sort
- Sorting networks
- Radix Sort

- Putting it together: adaptive sorting
Quicksort (Hoare 1961)

- Start on blackboard
- One partitioning step (inplace version)

function partition(array, left, right, pivotIndex)

pivotValue := array[pivotIndex]
swap array[pivotIndex] and array[right] // Move pivot to end
storeIndex := left

for i from left to right // left ≤ i < right
    if array[i] ≤ pivotValue
        swap array[i] and array[storeIndex]
        storeIndex := storeIndex + 1
    swap array[storeIndex] and array[right] // Move pivot to its final place

return storeIndex

Discussion: blackboard

**Mergesort** (von Neumann 1945)

- **Start on blackboard**
- **Merge function**

```python
function merge(left, right)
    var list result
    while length(left) > 0 and length(right) > 0
        if first(left) ≤ first(right)
            append first(left) to result
            left = rest(left)
        else
            append first(right) to result
            right = rest(right)
    if length(left) > 0
        append rest(left) to result
    if length(right) > 0
        append rest(right) to result
    return result
```

- **Discussion: blackboard**
John von Neumann (1903-1957)

- Hungarian (later American citizen) genius

- Among the first four selected for the Institute of Advanced Studies, Princeton (with Gödel and Einstein)

- Major contributions in: set theory, functional analysis, quantum mechanics, ergodic theory, continuous geometry, economics and game theory, computer science, numerical analysis, hydrodynamics, statistics
  - Founded game theory and applied it to economics
  - Von Neumann computer architecture
  - Manhattan project
Insertion Sort

- **Pseudocode**

```plaintext
function insertionSort(array A)
    for i = 1 to length[A]-1 do
        value = A[i]
        j = i-1
        while j >= 0 and A[j] > value do
            j = j-1
        A[j+1] = value
```

- **Discussion: blackboard**
Sorting Networks

- Start on blackboard
- Example: N = 4, 5 comparators

- Discussion: blackboard

Sorting Networks as Basic Blocks

Example

```
<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

network

cmp&swap (r0, r1)
cmp&swap (r2, r3)
cmp&swap (r4, r5)
cmp&swap (r6, r7)
```

unrolled code, scheduled for instruction level parallelism