Finishing parallel divide & conquer, parallel models, loop manipulations

2/1/16
Announcements

• HW 3 due at beginning of class Wednesday

• Reading:
  – For Wednesday: Chapter 6 thru 6.1.7
  – For Monday: Rest of Chapter 6

• No class Friday in honor of the midterm
Parallel divide and conquer: from merge sort to quick sort
Recall: Reductions

```
2 1 4 3 1 3 0 2
```
Scans

- Instead of just getting overall value, also compute value for every prefix

<table>
<thead>
<tr>
<th>A</th>
<th>2</th>
<th>1</th>
<th>4</th>
<th>3</th>
<th>1</th>
<th>3</th>
<th>0</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>sum</td>
<td>2</td>
<td>3</td>
<td>7</td>
<td>10</td>
<td>11</td>
<td>14</td>
<td>14</td>
<td>16</td>
</tr>
</tbody>
</table>

\texttt{var sum = + scan A;}
RAM model

• “The RAM model consists of a single processor and a memory with sufficient capacity. Each memory location can be accessed in a random (direct) way. In each time step, the processor performs one instruction as specified by a sequential algorithm.” (page 202)
List ways in which the RAM model differs from actual sequential computers
List ways in which the RAM model differs from actual sequential computers

• Different kinds of memory have different access times (cache, RAM, VM on disk, ...)
  – More generally, instructions take different amounts of time

• Memory does not have infinite capacity

• I/O (user and network) is important for many programs
PRAM model

• “A PRAM consists of a bounded set of identical processors which are controlled by a global clock. Each processor is a RAM and can access common memory to read and write data. All processors execute the same program synchronously.” (page 204)

• (C/E)R(C/E)W
List possible issues w/ creating a parallel computer implementing the PRAM model
List possible issues w/ creating a parallel computer implementing the PRAM model

- Bounded number of processors in practice
- Maintaining a consistent memory image
- Keeping instructions executing in lock step
- Limitations on the number of simultaneous accesses to a piece of memory
How ok is it for a programming model to differ from real hardware?

A. Anything different from the real machine is unacceptable
B. Only minor deviations can be tolerated
C. The model only has to match the main features
D. A vague resemblance between the model and real hardware is nice
E. What’s a programming model?
Bulk synchronously parallel (BSP)

• Separate processors connected by a network and a synchronization unit
• Program is series of supersteps
  – During each, the processors do local computation and send point-to-point messages
  – Each ends with a barrier synchronization
Loop scheduling

• Static
• Dynamic
  – self-scheduling
  – chunk scheduling
  – guided self-scheduling
Loop manipulations

• Loop coalescing: Converting multi-dimensional space from nested loops into 1D

• Loop tiling: Converting single loop into nested loops (outer = over tiles, inner = within a tile)
Linearizing multi-dimensional arrays

- cudaMemcpy only transfers 1D arrays
- Need to represent 2D array:

in a 1D form: