Parallel programming patterns
Going forward

• Reading
  – Wednesday: Sections 2.3-2.4
  – Friday: Sections 3.5-3.6
  – Next Monday: Sections 3.7-3.8

• Heat diffusion homework

• Extra credit opportunity: MLK convocation next week
Which of the following is true?

A. It’s Monday
B. It’s too early in the morning
C. This is CS 309
D. David is taking attendance
E. Not exactly one of the above
Parallel programming patterns

- Fork-join
- Parbegin-parend
- SPMD and SIMD
- Master-worker
- Client-server
- Pipelining
- Task pools
- Producer-consumer
Parallel programming patterns

- Fork-join
- `Parbegin-parend`
- `SPMD` and `SIMD`
- Master-worker
- Client-server
- Pipelining
- Task pools
- Producer-consumer
- Parallel loops
- Reductions (and maybe scans)
Reduction: Summing values in an array

\[
\begin{array}{cccccccc}
2 & 1 & 4 & 3 & 1 & 3 & 0 & 2 \\
\end{array}
\]
Reduction: Summing values in an array

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

```
  
  10
  / 
 3  7
```

```
  16
 /   
10   6
```

```
  6
/  
4  2
```

```
  3
/  
2  1
```
Finding max of an array

```
  4
 /  \
2 4 3 2
```

2 1 4 3 1 3 0 2
Finding the maximum index

2 1 4 3 1 3 0 2
Finding the maximum index
Parts of a reduction

- Tally: Intermediate state of computation
- Combine: Combine 2 tallies
- Reduce-gen: Generate result from tally
Parts of a reduction

• Tally: Intermediate state of computation
  \((value, index)\)

• Combine: Combine 2 tallies
  take whichever pair has larger value

• Reduce-gen: Generate result from tally
  return index
Parts of a reduction

- **Tally**: Intermediate state of computation
  
  \((\text{value, index})\)

- **Combine**: Combine 2 tallies
  
  take whichever pair has larger value

- **Reduce-gen**: Generate result from tally
  
  return index

- **Init**: Create “empty” tally

- **Accumulate**: Add single value to tally
Parts of a reduction

• Tally: Intermediate state of computation
  \((value, index)\)
• Combine: Combine 2 tallies
  take whichever pair has larger value
• Reduce-gen: Generate result from tally
  return index
• Init: Create “empty” tally
  return \((\text{MIN\_INT}, -1)\)
• Accumulate: Add single value to tally
  \((\text{larger value, its index})\)
Defining reductions

• Tally: Intermediate state of computation
• Combine: Combine 2 tallies
• Reduce-gen: Generate result from tally
• Init: Create “empty” tally
• Accumulate: Add single value to tally

Sample problems: +
Defining reductions

• Tally: Intermediate state of computation

• Combine: Combine 2 tallies

• Reduce-gen: Generate result from tally

• Init: Create “empty” tally

• Accumulate: Add single value to tally

Sample problems: +, histogram
Defining reductions

- Tally: Intermediate state of computation
- Combine: Combine 2 tallies
- Reduce-gen: Generate result from tally
- Init: Create “empty” tally
- Accumulate: Add single value to tally

Sample problems: +, histogram, 2\textsuperscript{nd} largest