

# *Smarter Parallel Prefix*

Jason Carver  
18.337J/6.338J

Optimizing Communication with uneven data distribution



# *Agenda*

- Parallel Prefix Review
- Attempted Optimizations
- Results
- Next Steps



# *Parallel Prefix Review*

- Seemingly difficult to parallelize
  - Example: Cumulative Sum
  - Basic idea (requires associativity):
    - Perform local actions
    - Receive previous data / Pass on data
    - Update local actions
- 
-

# *Attempted Optimizations*

- Do the total calculation first, not incremental
  - -O3 (d'oh!)
  - Send new total before doing new incremental
  - Uneven Distribution
- 
-

# Results

- Total first: none (after O3)
- -O3: 5x
- Actually be parallel: 42x over O3
- Uneven Distribution:
  - 22% incremental improvement
  - Total speedup: 54x
  - Time saved: 9s vs theoretical: 18.9s

# Caveats

- Theoretical total benefit:
  - $(\text{operations/comm}) * p * (p + 1) / 2$
- Needs many processors
- Needs good estimate of Ops/comm
- Needs much data per processor
  - Because of noise from false start processes
  - Note this works against having many processors

# *Calculating Ops/Comm*

- Calculate Comm time with:
  - Run Parallel Prefix
  - $(\text{time} - \text{last node step 2} - \text{first node step 1}) / (p-1)$
- Calculating ops is straightforward
- Example:  $(197\text{ms} - 4\text{ms} - 5\text{ms}) / 199 = 0.94\text{ms}$
- Ops took .038ns, so ratio of  $.94\text{ms} / .038\text{ns} \approx 25\text{k}$

# *Next Steps*

- Deal with process startup noise by aggressively calculating outside bounds
- Find best ratio of triangle to flat data

