Parallel Puzzle-Solving

Joel C. Adams

Parallel Puzzle-Solving

- Unplugged activity that can be used in any course
- Simple version can be used to introduce shared-memory concepts
- More complex versions can be used for distributed-memory concepts
- Data-visualization can be used to explain Amdahl's & Gustafson's Laws

Setup: Children’s Puzzles (from Thrift Stores)

Puzzles’ sizes differ by factors of 10

Shared-Memory Parallel Exercise

1. Divide students into several "processor groups"
   - → Single core, dual-core, quad-core, etc.
   - → Number of groups and their sizes vary, depending on course enrollment
2. Give each group a puzzle
3. Repeat:
   a. *Ready-set-go!* Start all groups and a timer simultaneously.
   b. Time how long it takes each group to solve their puzzle (e.g., online-stopwatch.com); record that time in a spreadsheet
   c. Disassemble puzzles, rotate them among the groups, reset the timer

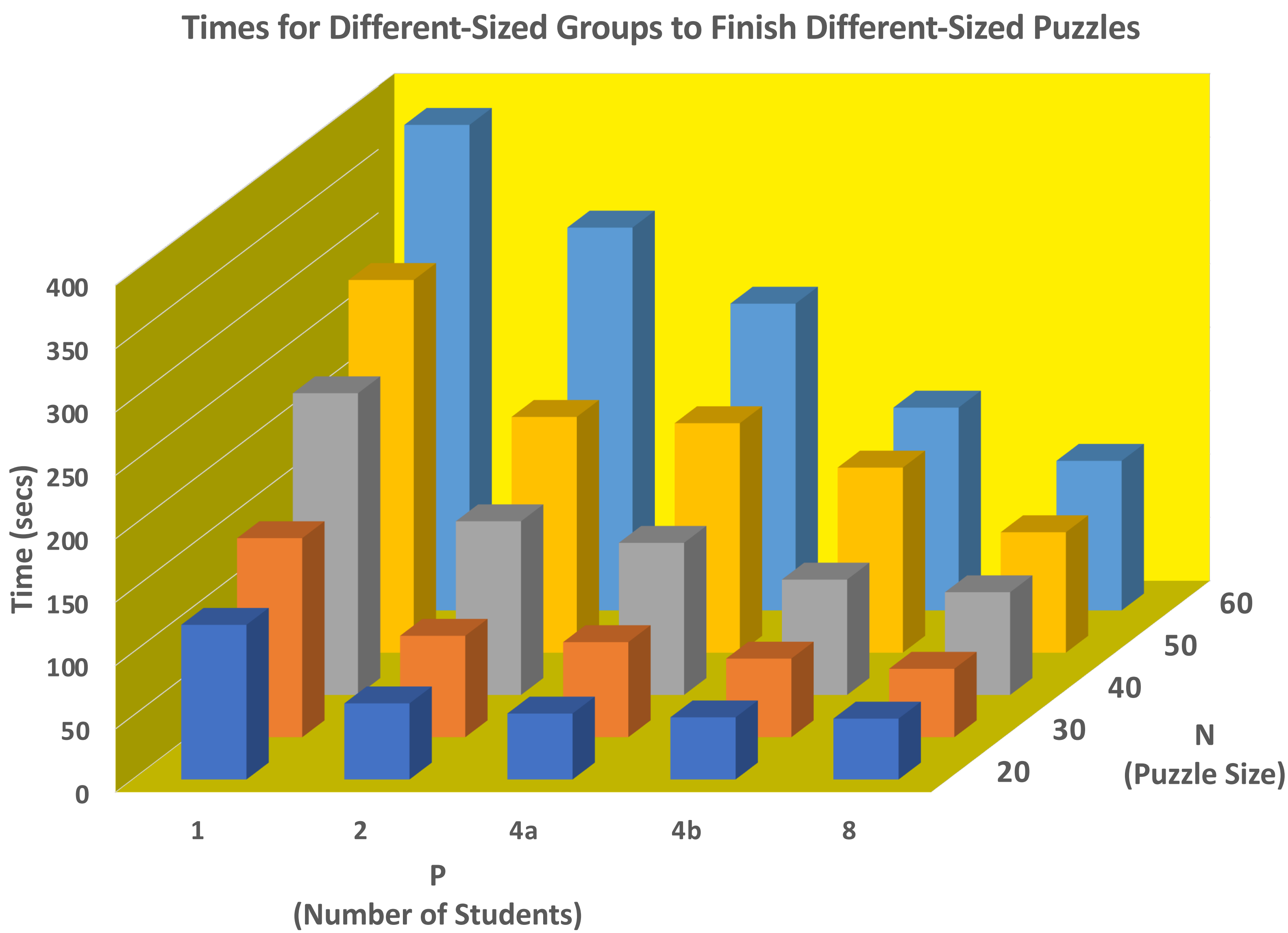
   Until each group has solved each puzzle
4. Lead discussion of students' experiences & observations

Unicore
processor

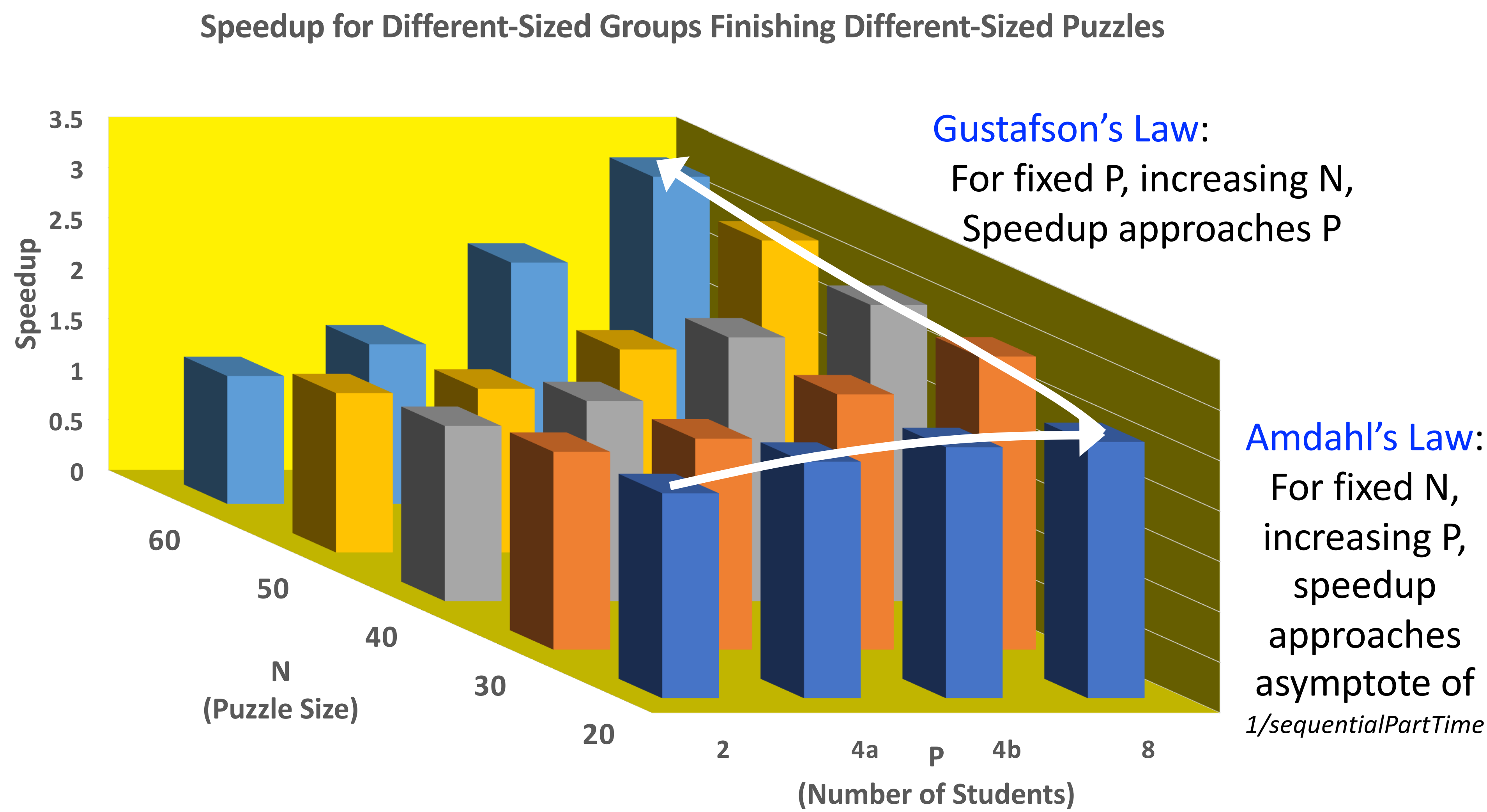
Dual-core processor

Quad-core
processor

Octa-core processor (minus two students so we can see the puzzle)



Times for Different-Sized Groups to Finish Different-Sized Puzzles
Time (secs)
400
350
300
250
200
150
100
50
0
1
2
4a
4b
8
P
(Number of Students)
20
30
40
50
60
N
(Puzzle Size)


Speedup for Different-Sized Groups Finishing Different-Sized Puzzles
Speedup
3.5
3
2.5
2
1.5
1
0.5
0
60
50
40
30
20
N
(Puzzle Size)
2
4a
P
4b
8
(Number of Students)
Gustafson's Law:
For fixed P, increasing N,
Speedup approaches P
Amdahl's Law:
For fixed N,
increasing P,
speedup
approaches
asymptote of
1/sequentialPartTime

Parallel "Laws"

Definition: $Speedup_P = Time_1 / Time_P$

- Amdahl's Law: Let $Time_1 = 1$. For a problem of size N and increasing P:

$$Speedup_P(N) = \frac{1}{\frac{parallelPartTime}{P} + sequentialPartTime}$$

As $P \to \infty$, $Speedup_P(N) \to \frac{1}{sequentialPartTime}$

- Gustafson's Law: As N increases:

$$Speedup_P(N) = P + sequentialPartTime * (1 - P)$$

If $sequentialPartTime \to 0$ as $N \to \infty$, then $Speedup_P \to P$