More performance analysis, plus parallel divide and conquer

1/29/16
Announcements

• HW 3 out; due Wednesday morning
• Midterm
  – Multiday, open book takehome
  – Out Wednesday or Thursday
  – No class next Friday (2/5)
  – Covers everything we’ve done so far
• Reading for Monday: Section 4.5 (parallel models) and 4.6 (parallel loops)
From HW 2: What is wrong with the following?

public class ThreadOp implements Runnable {

    ... 

    public void run() {
        for(int i=0; i<timeSteps; i++) {
            if(!isRunning) {
                ... return;
            }
            d.diffuse(...);
            try {
                barrier.await();
                catch(...) {
                    ...
                }
            }
        }
    }
}
Which of the following is **NOT** an issue with the speedup metric?

A. Comparing between hardware generations due to technology changes
B. Comparing against a non-optimized serial program
C. Comparing against a parallel program running serially rather than a serial program
D. Comparing “cold start” and “warmed up” runs
E. All of these are issues with speedup
Which of the following is **NOT** an issue with the speedup metric?

A. Comparing between hardware generations due to technology changes
B. Comparing against a non-optimized serial program
C. Comparing against a parallel program running serially rather than a serial program
D. Comparing “cold start” and “warmed up” runs
E. All of these are issues with speedup
Rather than running the same size problem with more parallelism (fixed-size speedup), sometimes we compare the running time on a proportionally larger instance (scaled speedup). How is this justified?

A. Gives better speedup numbers
B. Hard to find appropriate small instances
C. Larger systems are used for larger instances
D. Easier to plot the results (~flat instead of ~exponentially increasing)
E. Historical reasons; this should not be done
Rather than running the same size problem with more parallelism (fixed-size speedup), sometimes we compare the running time on a proportionally larger instance (scaled speedup). How is this justified?

A. Gives better speedup numbers
B. Hard to find appropriate small instances
C. Larger systems are used for larger instances
D. Easier to plot the results (~flat instead of ~exponentially increasing)
E. Historical reasons; this should not be done
Parallel divide and conquer