Why Parallel Programming?

Goal

- Here, we are going to look at the higher level concepts associated with parallel programming
- Look at how to know when to use it
- Refresh: What is parallel programming?

Introduction

- Suppose we want to catch 100 fish. You are the only person available so you head out to catch fish one by one. In computing, what type of computing is this?
- What if we send a second person with you to catch 100 fish? In theory, it should take half the time to catch the fish because there are twice as many people working **in parallel** to catch the fish
- What if we send 4 people instead of 1? How long, in theory, should it take to catch 100 fish?
- What if we send 101 people to catch 100 fish? 100 people will catch 1 fish each, but 1 person will be left doing nothing
- What if we send 500 people to catch 100 fish? Most of the people will not be able to contribute anything in this case

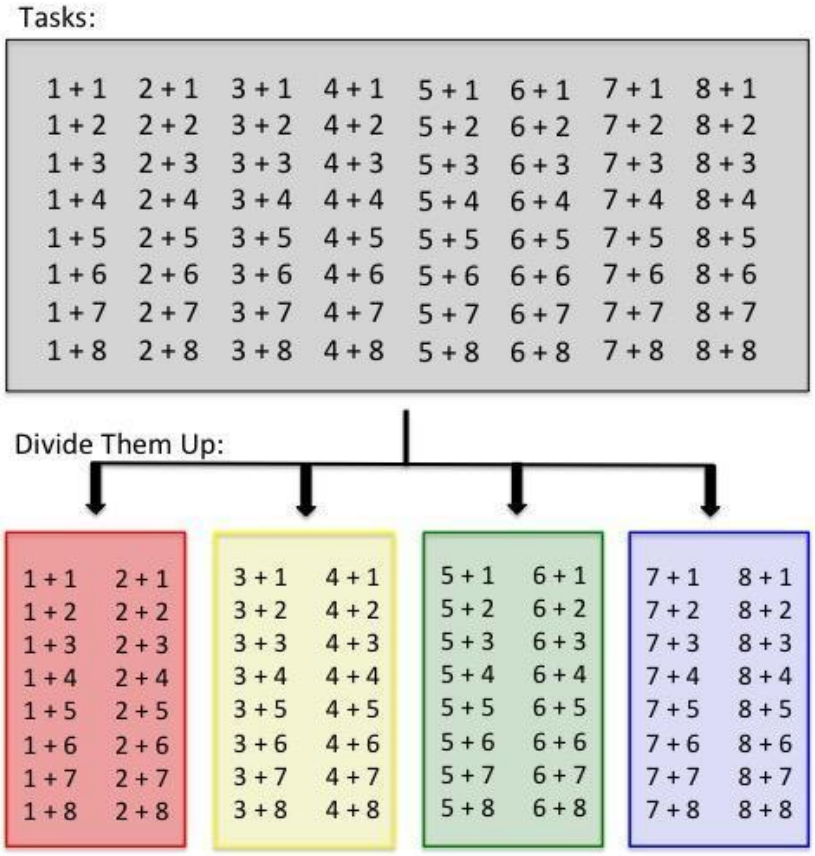
Refresh: Serial (sequential) and Parallel Programming

- What is a serial, or sequential, program?
- Traditionally, a serial program follows these steps:
  - A problem is broken into a discrete series of instructions
  - Instructions are executed sequentially one after another
  - Everything is executed on 1 processor and only 1 instruction is executed at a time
- A parallel program generally follows these steps:
  - A problem is broken into discrete parts that can be solved at the same time
  - Each part is further broken down into a series of instructions
  - Instructions from each part execute all at the same time on multiple different processors

Parallel Program

- Programs need to be designed specifically to be run in parallel
- Sometimes parts of a program can only be run in serial, so the portions of the program that CAN be run in parallel need to be designed to do so
- Example times to divide parts of the program into parallel include:
  - Large amounts of data that all require the same instructions
  - Iterations of a loop doing calculations over and over
  - CONTINUE THIS LIST

Example of Parallelism



QUESTION: Is dividing up the work here and parallelizing it beneficial? What do you gain by doing so?

Why Parallelism?

- There are 3 reasons that parallelism can be useful for an application
- These reasons follow the motto “Sooner, Better, More”
- Sooner Speedup
- Better Accuracy
- More Scaling

Sooner Speedup

- In theory, programs will run faster if they are parallelized compared to executing the same program in serial
- This means there should be some factor of “speedup” when running in parallel
- QUESTION: What is an example where parallelization would provide speedup?
- QUESTION: What is an example where parallelization would NOT provide speedup?

Better Accuracy

- The goal with accuracy is to form a better solution to a problem while obtaining the same level of accuracy (or more accuracy) by running in parallel
- If more processors are assigned to a task, there is more computing power available to do tasks such as comprehensive error checking, etc., leading to an accurate result
- NOTE: In order to make a program more accurate, speedup may need to be sacrificed
- QUESTION: What is an example where parallelization would provide more accuracy?
- QUESTION: What is an example where parallelization would NOT provide more accuracy?

More Scaling

- More processors can be used to module a bigger problem in the same amount of time as fewer processors can model a smaller problem
- More science can be done in the same amount of time with parallelism

Fishing Analogy, Scaling

- Fishing analogy from earlier is example of scaling add more resources to do same amount of work
- If 100 fishermen each catch 1 fish, then they will reach the goal of 100 fish, in theory 100 times faster

Fishing Analogy, Scaling

- More is not always better!!!
- If we send 500 fishermen to catch 100 fish, 400 of those fishermen will not have work to do and will be wasted resources

Problems with Parallelism

- Two overarching problems with parallelism that must be kept in mind when deciding whether or not to use parallelism for a problem include:
  - Communication overhead
  - Speedup limited by serial regions

Communication Overhead

- Communication overhead is when time is lost when workers have to wait for communications among themselves before and after calculations
- Falls under “more is not always better”
- Need a balance between communication done among workers and the computation done by each worker
- If too much communication takes place compared to number of computations, it can actually slow the problem down and can run slower than when running in serial

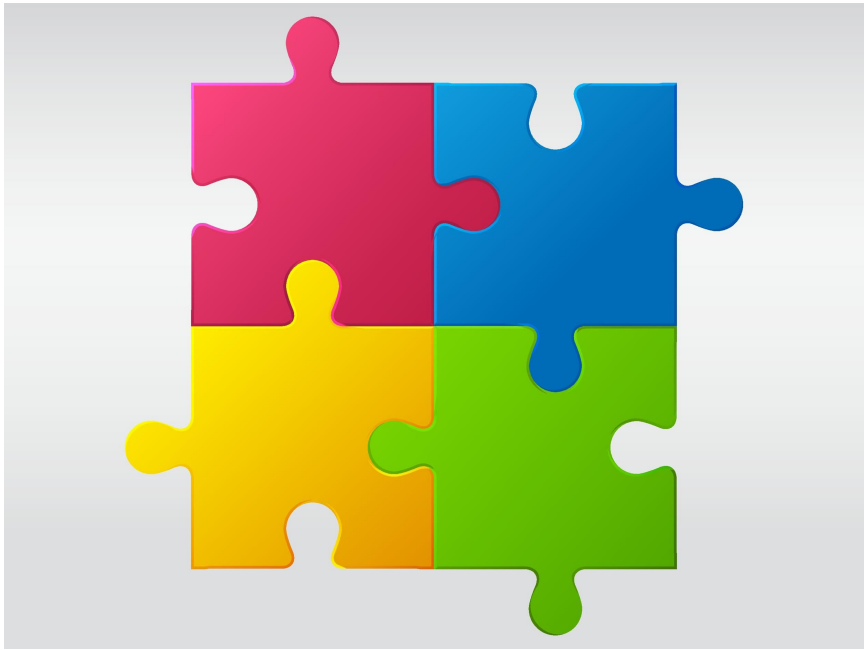
Speedup Limited by Serial Regions

- There are times when regions cannot be executed in parallel
- Referred to as Amdahl's Law
- If there is any serialized region in the code whatsoever, the program will not perfectly scale because of overhead from serialized region(s)

Exercises

Puzzle Problem

- Need volunteers to help put a puzzle together in parallel
- Do combination of increasing number of people, dividing the work in different locations, and communicating among each other to exchange pieces and put the puzzle together
- Was parallelism useful or a hindrance for this problem? What about other similar problems?



Mix and Match

- In this activity, we are going to look at example applications and programs and decide whether or not they would benefit from parallelism and why

Simulating Galaxy Formations

- Is it better to run this application in parallel or serial? Why?

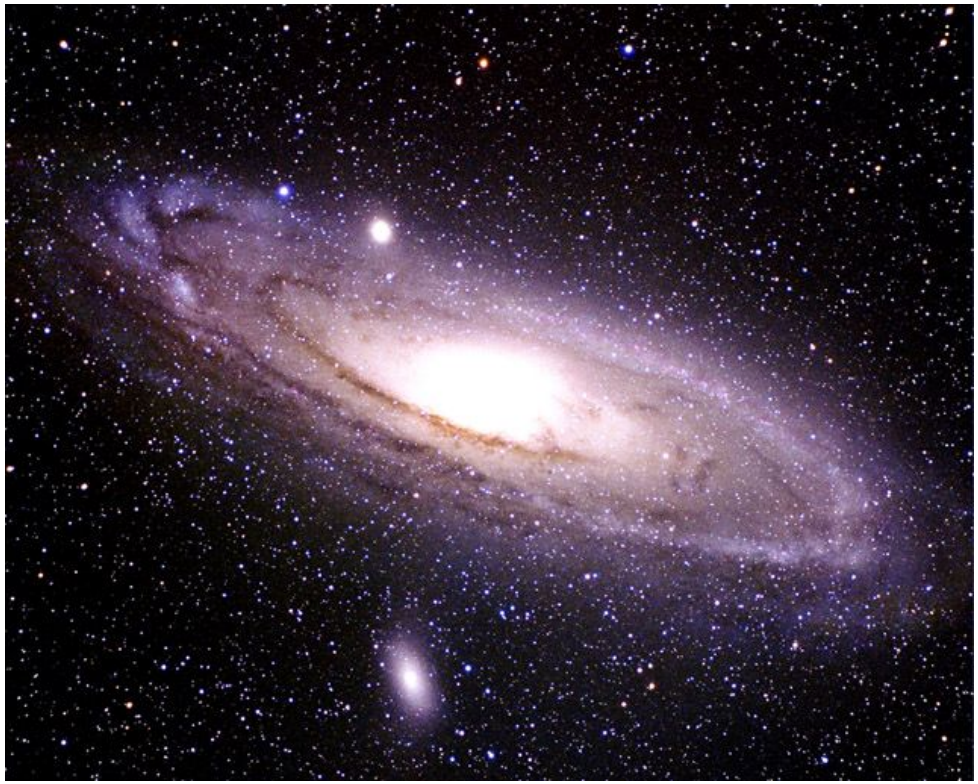
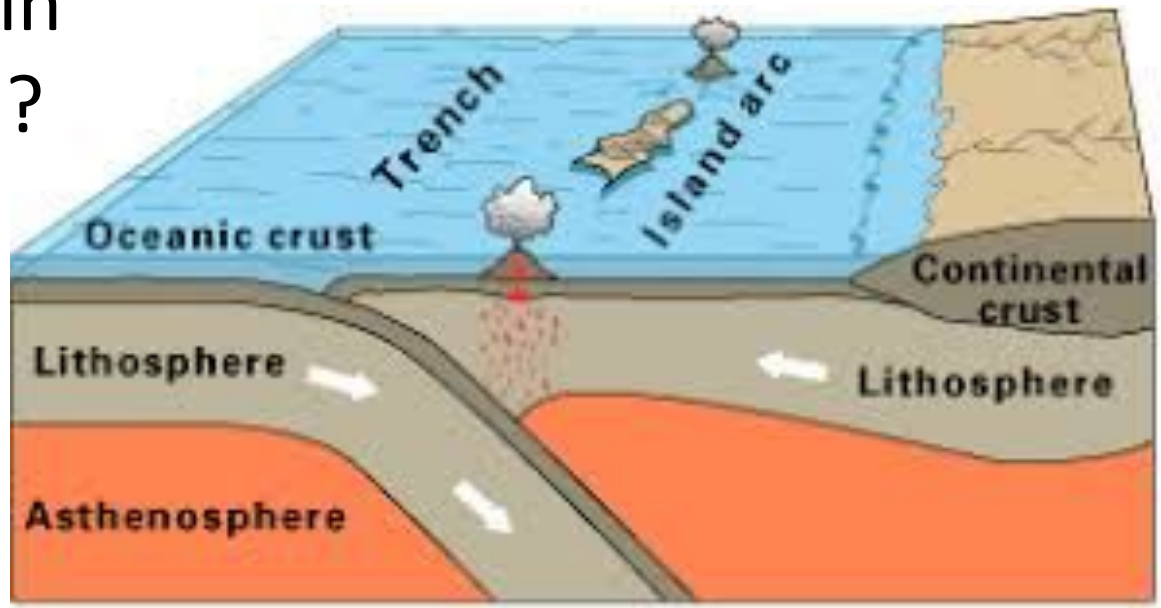


Plate Tectonic Movements

- Is it better to run this application in parallel or serial? Why?



Oceanic-oceanic convergence

Predicting Hurricane Paths

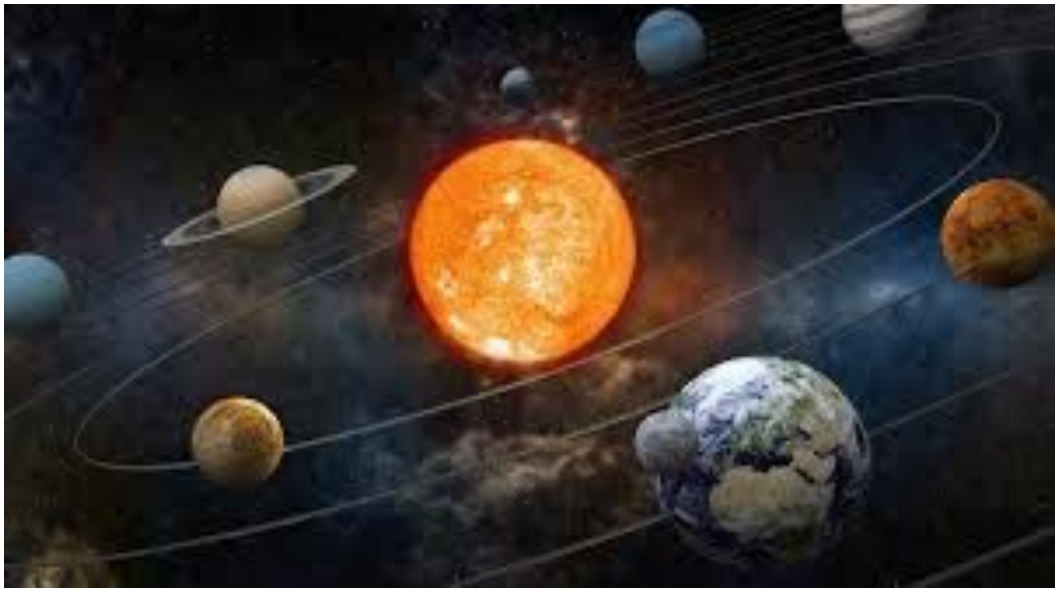
- Is it better to run this application in parallel or serial? Why?

Tip Calculator

- Is it better to run this application in parallel or serial? Why?

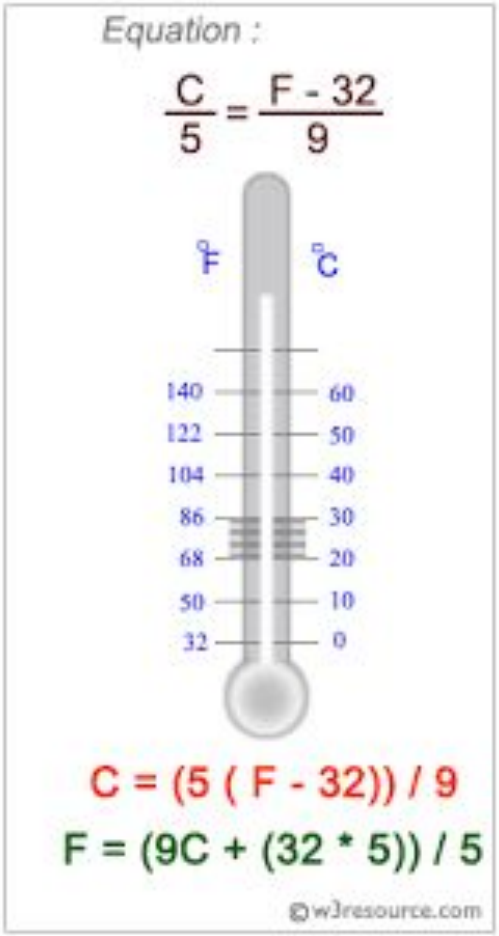
Simple Planetary Movements

- Is it better to run this application in parallel or serial? Why?



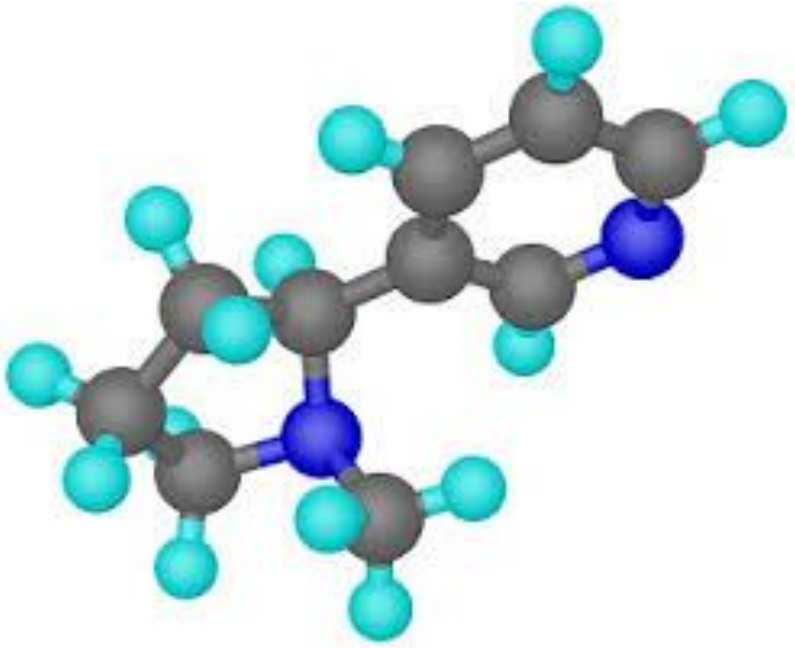
Convert Temp from C to F

- Is it better to run this application in parallel or serial? Why?



Molecular Dynamic Simulation

- Is it better to run this application in parallel or serial? Why?



Example Program: Time to Science with GalaxSee