**Lesson 2 – Every Bit of the Internet**

**\*\*Instructions:** Please change the text color of your responses to red text. Please organize the endings to each page.

**ACTIVITY 2.2.4 – Parallel and Distributed Computing**

Define the following terms:

1. threads

| Independent tasks within a program. |
| --- |

1. parallel processing

| Using multiple CPUs at the same time |
| --- |

1. Sequential processing

| Executing tasks one at a time, rather than in parallel. |
| --- |

1. efficiency

| The amount of computational resources used by an algorithm, compared to the size of the input. |
| --- |

1. Scalable

| A solution that can be enlarged without much change in how it works. |
| --- |

1. Fault-tolerant

| Functional even after components fail |
| --- |

1. Crowdsourcing

| Using many people to solve small problems that can then be combined to get meaningful results. |
| --- |

Review the sequential algorithm below for how to bake a cake. Note that this algorithm assumes there is one person doing one task at a time, similar to having one processor in a computer.

| Task Sequence | Task | Processing Time  (in minutes) |
| --- | --- | --- |
| 1 | preheat oven to 375 degrees | 0.25  (to just set the oven temperature) |
| 2 | get flour from pantry | 0.25 |
| 3 | get eggs from refrigerator | 0.25 |
| 4 | get milk from refrigerator | 0.25 |
| 5 | get sugar from pantry | 0.25 |
| 6 | get baking soda from pantry | 0.25 |
| 7 | get oil from pantry | 0.25 |
| 8 | get cocoa powder from pantry | 0.25 |
| 9 | get two bowls from cabinet #1 | 0.25 |
| 10 | get measuring cup from cabinet #2 | 0.25 |
| 11 | get measuring spoon from cabinet #2 | 0.25 |
| 12 | get fork from drawer #1 | 0.25 |
| 13 | get mixer from cabinet #3 | 0.25 |
| 14 | get baking pan from cabinet #4 | 0.25 |
| 15 | measure wet ingredients (milk, oil) and add to bowl #1 | 0.5 |
| 16 | crack eggs and add to bowl #1 | 0.5 |
| 17 | mix all wet ingredients with fork | 0.25 |
| 18 | measure all dry ingredients and add to bowl #2 | 1.0 |
| 19 | pour dry ingredients into bowl #1 (after wet ingredients are mixed) | 0.25 |
| 20 | use mixer to mix ingredients for 2 minutes | 2.0 |
| 21 | grease the pan with some oil | 0.5 |
| 22 | pour batter into pan | 0.5 |
| 23 | place pan in oven | 0.25 |
| 24 | bake for 45 minutes | 45 |
| 25 | remove from oven | 0.25 |
| 26 | let cake cool for 20 minutes | 20 |
| 27 | get frosting from grocery store | 30 |
| 28 | get frosting spatulas from drawer #1 | 0.25 |
| 29 | spread frosting on cake | 4 |
| Total Processing Time | | 108.75 minutes |

1. Given the constraints below, work with your partner to come up with the most efficient solution using parallel computing.

Thread 1: Get milk (0.25 min) → Get oil (0.25 min) → Get two bowls from cabinet #1 (0.25 min) → Get eggs from refrigerator (0.25 min) →Get fork from drawer #1 (0.25 min) →Get cocoa powder (0.25 min) →Get baking soda (0.25 min) →Get sugar (0.25 min) →Get flour (0.25 min) →Get mixer (0.25 min) → Wait for 1 min → Use mixer to mix ingredients for 2 minutes (2 min) →Get frosting from grocery store (30 min) → Get frosting spatulas from drawer #1 (0.25 min) → End

Thread 2: Preheat oven to 375 degrees (0.25 min) →Get measuring spoon from cabinet #2 (0.25 min) →Get measuring cup from cabinet #2 (0.25 min) →Measure wet ingredients (milk, oil) and add to bowl #1 (0.5 min) → Crack eggs and add to bowl #1 (0.5 min) → Mix all wet ingredients with fork (0.25 min) →Measure all dry ingredients and add to bowl #2 (1.0 min) → Pour dry ingredients into bowl #1 (0.25 min) → Grease the pan with some oil (0.25 min) →Wait for 1.75 min →Pour batter into pan (0.5 min) →Place pan in oven (0.25 min) →Bake for 45 minutes (45 min) →Remove from oven (0.25 min) → Let cake cool for 20 min (20 min) →Spread frosting on cake (4 min) → End

Run time: 75.25 minutes

Comparing the efficiency of solutions can be done by comparing the time it takes them to perform the same task. 

 Draw a chart or timeline for the parallel computing solution. Make sure the drawing includes the task descriptions and the processing times.

[Link to chart](https://docs.google.com/drawings/d/1u_MI6Oj1BuWzGx4E0aSdjdY617zR5rtUXwgT8QXNU4M/edit?usp=sharing)

 Calculate the solution's total processing time. 75.25 minutes

 Calculate the speedup of your parallel solution. 108.75 minutes / 75.25 minutes = 1.45

1. Work with your partner to research @home distributed computing projects. Find one that interests you and prepare a visual summary.

Summarize your findings, including:

* Name and goal of the project
* Organization(s) behind this project
* Start date and end date (if applicable)
* Number of total users
* Why this project is important

Climateprediction.net is an @home distributed computing project that is used for climate modeling. Oxford University has run it since 2003, and it has 307,096 total users and 659,535 total hosts, resulting in a processing power of 15.131 teraflops, although the average was 78.8 in 2020. This project is important because it allows climate models to be refined through comparing different ones against each other, which is both important for meteorology and environmentalism, as predicting the climate is a hard task that provides invaluable information about how to handle climate change.

[Visual Summary](https://docs.google.com/presentation/d/1njosjRysSsFYd-hw8NNkj3kjr_rBoAb5HoE9lbJn1C8/edit?usp=sharing)

1. In some cases, humans are still better at solving certain problems than computers. One particular type of solution relies on humans for ***crowdsourcing***, and then computers take over to combine and analyze data. Use the internet to research one crowdsourcing project and create a computational artifact to share with the class.

* Name and goal of the project
* Organization(s) behind this project
* Start date and end date (if applicable)
* Any findings or conclusions from this project

reCAPTCHA was a crowdsourced project with the goal of restricting internet access to bots and transcribing books. It was originally developed by a team of computer scientists at Carnegie Mellon University, who formed the company reCAPTCHA, and was later bought by Google in 2009. The project was originally released in 2007, began to have functionality limited in 2013, and was shut down in 2018. This project allowed the New York Times to digitize many old newspapers and Google to digitize many books on Google Books, which decreased the amount of storage necessary to store these, as plaintext has a much smaller file size than images. Additionally, reCAPTCHA was also used to make Google Maps more accurate by reading text on Street View.

[Visual Summary](https://docs.google.com/presentation/d/1S9l1vj6xzVN2Hksaf5OoVPcdTIxkn4xTdPCZLXlB5to/edit?usp=sharing)