Virtual M&M Sorting

Course Level:

CS0/CS1/CS2

PDC Concepts Covered:

PDC Concept	Bloom Level
Serial vs Parallel (Speed UP)	K
Decomposition	K
Parallel Overhead	K
Sequential Dependency	K
Concurrency	K

Prerequisites:

None

Tools Required:

- Access to virtual M&M Sorting PowerPoint slides on shared Google Drive.
- Access to virtual chatroom service with sub-room capabilities, eg. Zoom breakout rooms

Introduction:

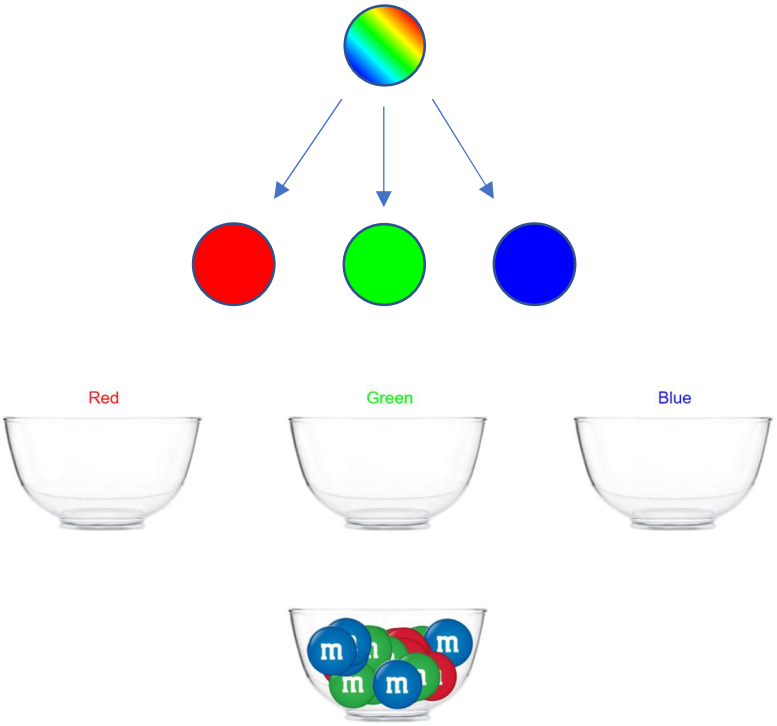
The main goal of this module is to introduce the basic concept that a task may be done in parallel in order to complete it faster. This introduces new steps and additional work: decomposition, where the work must be split up and results combined. As this is not needed in the sequential version, it is an example of parallel overhead.

Note:

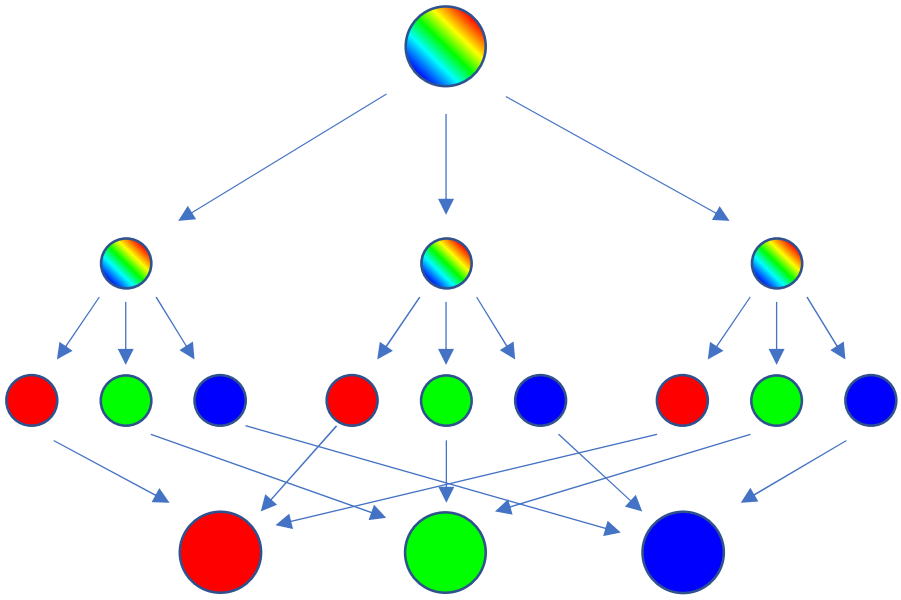
The most efficient way to perform this activity repeatedly is to duplicate the original slides (ctrl+d) before starting the activity, and then deleting them afterwards.

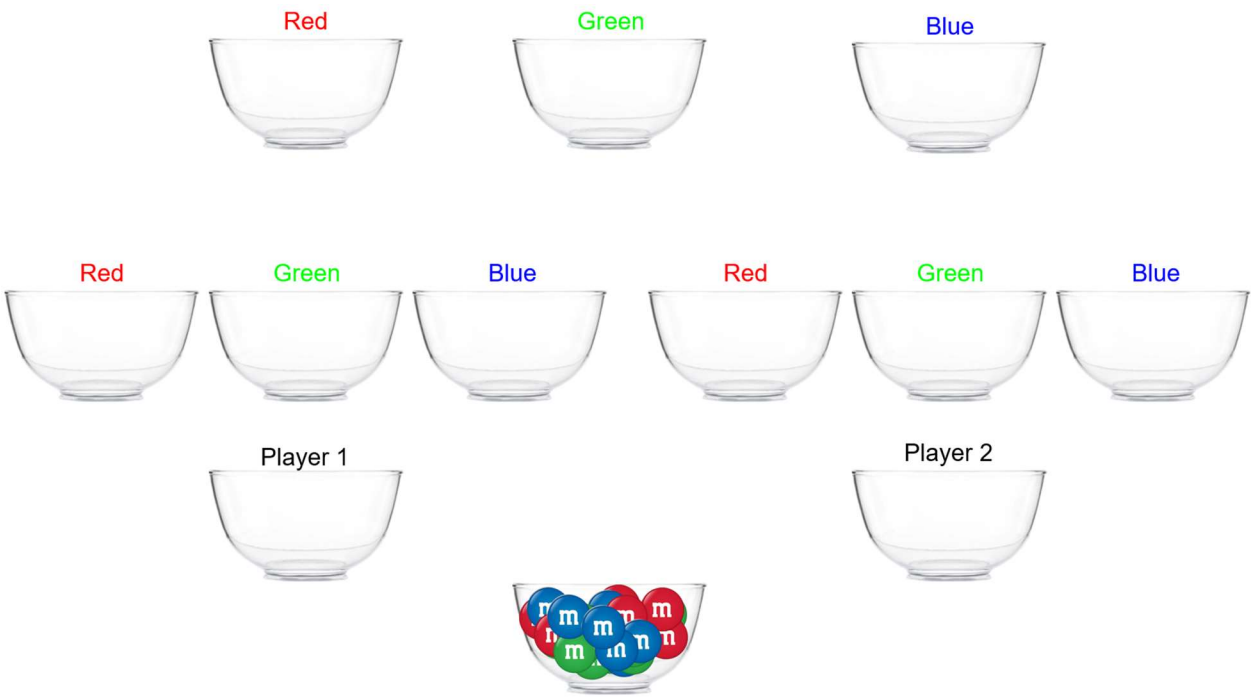
Activity Description:

Have all the M&Ms in one bowl. For simplicity and time only use 2-3 colors. Have one student sort the M&Ms by color into separate bowls, by clicking and dragging each M&M into the appropriately labelled bowl. Have another student time it and record it in the virtual chat.



Now two students will sort the M&Ms. However, this requires some organization. Each student has a bowl for unsorted M&Ms as well as one bowl for each color. Each student should go up to the main unsorted bowl and get about one half of the M&Ms (this can be done dragging multiple M&Ms at once). Then they should return to their bowls and sort them. After they are done sorting, they should empty each of their sorted bowls into the main sorted bowls. Once again, time how long it takes with three people and compare it to single person time.





Important Notes:

Several parallel concepts can be illustrated using this activity. It is up to instructor how many concepts they cover and in how depth they want to cover them.

1. Speed up: Point out that the parallel time is not one half of serial time. The extra work in the parallel version keeps this from happening.
2. Decomposition: Parallel implementations require dividing a task into subtasks. In this case the M&Ms must be distributed to the sorters and the sorted M&Ms must be recollected.
3. Synchronization: Coordination is needed between the sorters. They must take turns when getting the initial unsorted M&Ms.
4. Concurrency: There is a limit to the number of useful sorters. If there are more sorters than there are M&Ms then there is no work for some.
5. Load Imbalance: If the sorters get uneven amounts of M&Ms or if some sorters are faster than others then some people will get done early and have nothing to do.

Extensions:

1. Sequential Dependency: Add the additional task of counting the number of each color M&M. This must be done after the sorting is finished.
2. Partitioning and Load Imbalance: Let the students come up with other ways to distribute the M&Ms and deal with some sorters being faster than others.