Classroom Activity 1 – Power of processors

Aim

The aim of this activity is to demonstrate how a supercomputer uses multiple processors at the same time to solve big problems more efficiently and that each problem has an optimal number of processors.

Preparation

- Split a deck of cards into its four suits, keeping the order of the cards random.
- Have a stopwatch or timer at hand.

Instructions

1. Split the class into groups of 1 student, 3 students, 5 students and 10 students. You can adjust these group sizes depending on student numbers, as long as there is a clear small, medium and big group.
2. At the word **Go**, the groups each need to arrange their suit of cards in order from Ace to King.
3. Record how long it takes for each group to finish, noting which group finishes first.

Outcome:

This activity will demonstrate that working simultaneously as a group on a problem can lead to a faster solve, but that the ideal group size depends on the complexity of the problem. The expected outcome is that one of the middle-sized groups is fastest, as they have a good balance of having lots of processors (brains) but not too many that it slows down the activity with people management. A supercomputer similarly uses multiple processors at the same time to solve big problems more efficiently. Each computational problem and software package has its optimal number of processors, which can vary depending on the research task.

Real world example:
When scientists are developing their software to run on a supercomputer, they need to make sure that it can efficiently distribute all of its computing tasks across the available processors. When running the code at large scales, it's important for them to know what the most efficient number of processors is. In this example from 4D-VAR, an important subsystem of commonly used weather forecasting models, efficiency increases when running the code across 128 to 384 processors, but then decreases significantly from there. The optimal number of processors depends on the code and how it works. Computer engineers and scientists can find ways to improve the efficiency of their software and make it run effectively on more processors at the same time. Each code is different: some codes are limited by memory constraints, while some are limited by data transfer speeds. There can also be inefficiencies in the way the code handles its calculations and data that can be fixed up to improve how the code scales up to large numbers of processors. Some codes

currently in use in science around the world are efficient all the way up to hundreds of thousands of processors.