Project1: Search for Intelligent Puzzles

Start Assignment

Due Mar 5 by 11:59pm **Points** 100 **Submitting** a file upload
Available Feb 16 at 12am - Mar 7 at 11:59pm

**** Search for Intelligent Puzzles ****

This project is designed to explore the bag-of-tasks model of parallel programming. One commonly known example of this model is the SETI at Home program where a SETI server send data samples to client machines all over the world. The client processors perform data processing operations on the data in order to determine if the signals have high information content that may be indicative of intelligent life on another planet. Since the amount of data involved in the transactions is relatively small compared to the computational requirements, it is possible to obtain extremely high performance using diverse anddistributed systems. Since the order of task execution is not important for correct operation (this is the bag part of bag-of-tasks), it is possible to get good processor utilization even if workloads vary significantly. Since new tasks are assigned to processors as they finish their currently assigned tasks, a slow processor is naturally assigned fewer tasks to compensate.

In this problem, we will solve a related bag-of-tasks problem. In this problem we are searching for intelligent puzzles. These puzzles are in the general class of peg hopping games that are played on a 5x5 game board. In each position on the 5x5 grid may hold a peg, a hole that a peg can fit in, or an unused space (no hole or peg). The game is played by applying the following rule:

if a hole is position next to two pegs, then the peg on the opposite side can jump over the middle peg to the hole. The middle peg is removed. For example, if 0 is a hole and X is a peg, then this is illustrated as

0XX -> X00

This rule can be applied in either a horizontal or vertical direction. A game is solved by multiple applications of the rule. A puzzle has a solution if there are a sequence of moves such that all but one peg is removed. A puzzle is determined to be intelligent if a solution exists using the given rules. A puzzle can be determined to be intelligent by performing a depth first search of all possible moves and returning on the first solution of the puzzle. This produces a task with high workload variability, high computational intensity, and relatively low communication requirements. The task is to develop a bag-of-tasks parallel implementation of the provided serial program that tests puzzles.

The starter project files that includes a detailed description of the project and information on the computational cluster usage can be found on titan.hpc.msstate.edu under the directory /scratch/CSE4163/Project1/project1.tar.

The student will turn in the program (should be just the main.cc file) and a technical report that documents the implementation, the theoretical analysis, and the measured performance results. The rubric used for grading is given below:

project_checksheet.pdf (https://canvas.msstate.edu/courses/100858/files/7913391?wrap=1) **(https://canvas.msstate.edu/courses/100858/files/7913391/download?download_frd=1)**

The readme file that describes how to log into the cluster and get started is linked below.

README.txt (https://canvas.msstate.edu/courses/100858/files/7913423?wrap=1) **(https://canvas.msstate.edu/courses/100858/files/7913423/download?download_frd=1)**