Université Libre de Bruxelles

Real-Time Operating Systems

Mastermind solver

Authors:
ABOU ZAIDI, Ahmed
AZZOUZI, Samia
SEFU Kevin

December 13, 2018

1 Introduction

In this second project we had to implement a parallel Mastermind solver using MPI on HYDRA. The main goal of this project was to help us become familiar with parallel algorithms and to understand the benefits and drawbacks that parallel algorithm present.

2 Implementation choices

2.1 Division of the task between processes

To understand how we decide to split the solving task between the processes, let consider the following example :
Let consider a mastermind game with S spots and N colors.

Step 1 : Calculate the number of all possible guesses that could be generated:

$$\frac{N!}{(N-S)!}$$

Step 2 : Divide the number return in Step 1 by the number of processes -1 (we don't take into account the master node). The step 2 will return for all processes the maximum number of guesses that the process will generate in the worst case and the index from which this generation will start. Lets call those two numbers G and I

Step 3 : At each turn all the processes will send (starting from I), the next plausible guess until the process reach the number G.

3 MPI Protocal used

In our project we decide to us two MPI protocols, firstly the MPI Gather and secondly the MPI BCast.
We used the MPI Gather in order to get from the game master all the guess that players node generate therefore the master's receive buffer size at each turn was equal to the spot's number * the number of process.

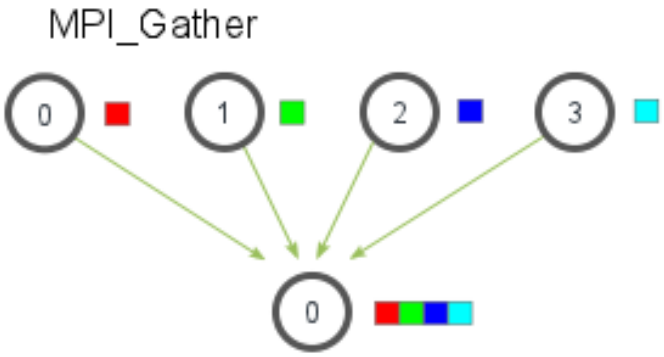

Figure 1: MPI Gather

Secondly we used MPI BCast in order to send a guess and its evaluation to all player's node.

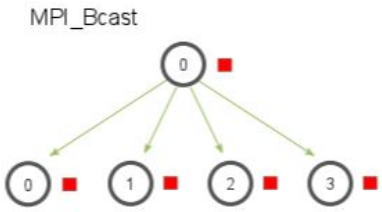

Figure 2: MPI BCast

4 Class diagram and code description

5 Performances and Limitations

About our code performances and limitations we observed after several running that, the solution was found in at most 6 turn for a sport range of 6 and color range of 15. We also observed that our algorithm end in at most 25 seconds for those range also.
One big limitation of our algorithm is the fact that, if the number of spots and

colors became huge the speed of our algorithm will decrease proportionally and resources needed will grows also.

6 Formula giving the number of possible guesses at start depending on spots and colors number

7 Difficulties during project implementation

7.1 Plausible Guess

During the project a big difficulty that we met was to understand, the concept of plausible guess and how to implement it in our source code. To overcome this problem we decide to download on our personal cell phones the mastermind game and to try to understand this notion.

7.2 Hydra message protocol

As hydra was something new to us, it wasn't something easy for us to understand directly how hydra's message protocols works. This problem were solved by simply reading the file available on UV and by practicing.

8 Personal critics on parallel algorithm

We discover that parallelization was a powerfull method because it allows us to split a task into several nodes and therefore to speed up the time needed for solving the initial problem but we observed also that parallelization has costs such as :

1. Sometimes not all processors are used, this can happen because of the intrinsic sequential nature of a given step or part of the problem.
2. Parallel algorithm are harder to design
3. Delays of the communication