



AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE
Wydział Fizyki i Informatyki Stosowanej

Systemy Równoległe i Rozproszone
Monte Carlo - Przydział pokoi metodą
simulated annealing

Piotr Libucha, Kamil Kaproń 09.05.2023r.

1. Wstęp

Celem projektu było stworzenie współbieżnego programu do rozwiązywania problemu przydziału pokoi z użyciem simulated annealing.

1.1. Opis problemu

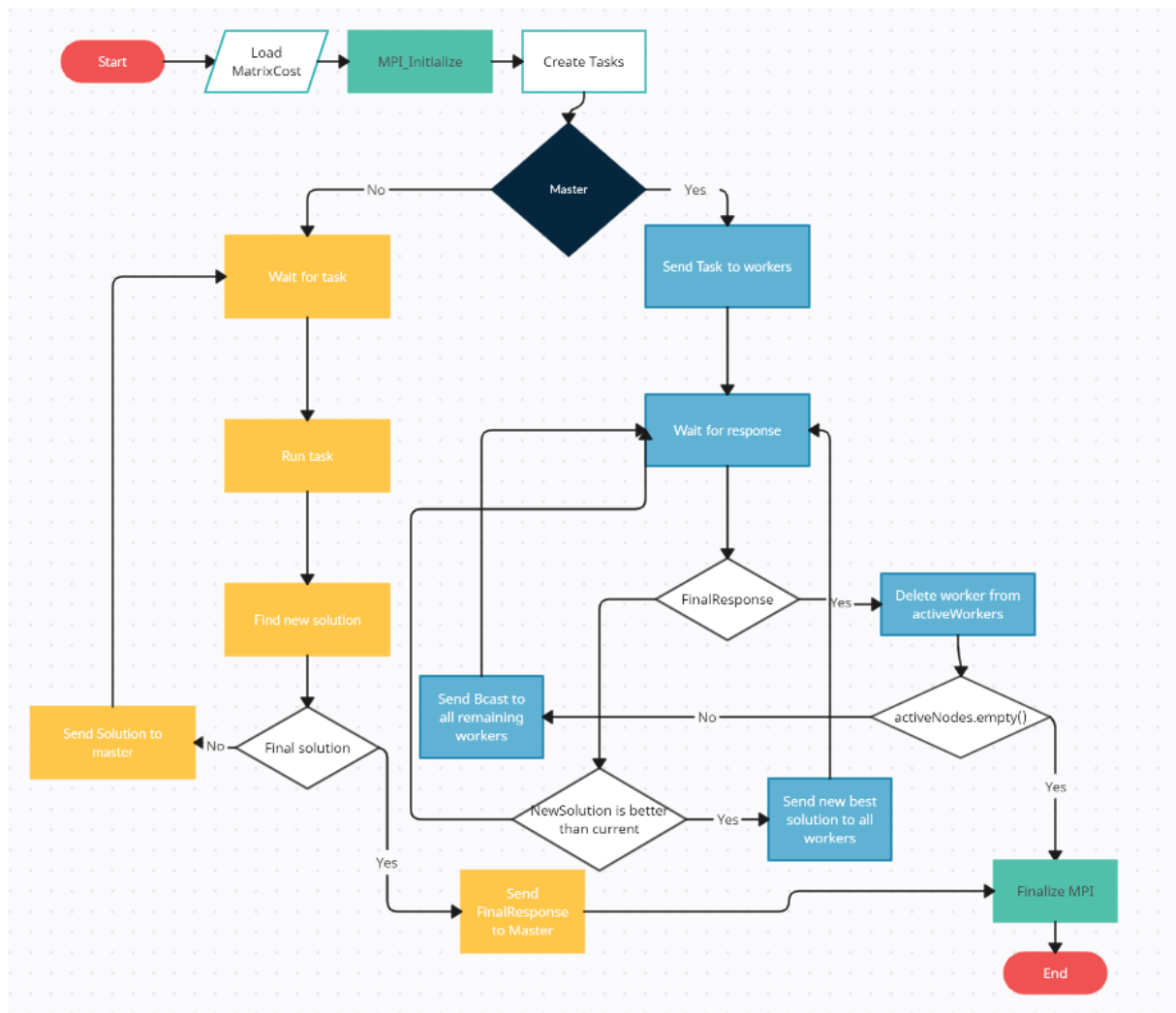
Problem przydziału pokoi polega na przydzieleniu określonej liczby studentów do określonej liczby pokoi, spełniając określone wymagania. Metoda simulated annealing to technika optymalizacyjna, która może być wykorzystana do rozwiązania tego problemu poprzez symulowanie procesu zmiany temperatury. Algorytm polega na losowym wyborze studenta i przeniesieniu go do innego pokoju z pewnym prawdopodobieństwem, zależnym od różnicy między wartością funkcji celu dla obu rozwiązań oraz aktualnej temperatury. Proces symulacji temperatury polega na stopniowym zmniejszaniu temperatury, co powoduje, że prawdopodobieństwo przyjęcia gorszego rozwiązania maleje, a algorytm skupia się na szukaniu rozwiązań o coraz mniejszej wartości funkcji kosztu.

2. Budowa programu

Program został stworzony zgodnie z koncepcją Master – Worker. Mechanizm Master-Worker to architektura przetwarzania równoległego, w której jednostka centralna, zwana Masterem, koordynuje pracę wielu jednostek obliczeniowych, zwanych Workerami.

W opisywanym programie master jest odpowiedzialny za przydzielenie zadań do workerów, koordynację pracy między workerami oraz zbieranie wyników z powrotem. Workerzy natomiast wykonują przydzielone im zadania i wysyłają wyniki do Mastera.

Na starcie master wysyła stworzonego taska do wszystkich nodów. Na każdym z nich tworzony jest worker, zadanie jest odbierane, przypisywane do workerów, następnie jest uruchamiane. Każde znalezienie lepszego rozwiązania przez danego workera jest wysyłane do mastera, gdzie jest sprawdzane czy nowo znalezione rozwiązanie jest lepsze od aktualnie najlepszego rozwiązania. Jeśli tak master wysyła do wszystkich aktywnych nodów nowe najlepsze rozwiązanie. Cały proces trwa do momentu znalezienia przez jednego workera najlepszego rozwiązania. Wysyła on wtedy odpowiednią wiadomość do mastera, który kończy pracę wszystkich nodów. Przydział pokoi oraz jego koszt jest wyświetlany na konsoli.



3. Uruchomienie programu

Do projektu został załączony plik makefile, który sam kompiluje pliki. Należy przedtem pamiętać o zrobieniu source na odpowiednią wersję. W przypadku programu był on tworzony na wersji 411.

Komenda do source: `source /opt/nfs/config/source_mpich411.sh`

Po skompilowaniu programu make należy upewnić się, że w pliku nodes znajdują się dostępne instancje. Zapisanie do pliku nodes instancji odbywa się poprzez komendę:

`/opt/nfs/config/station204_name_list.sh 1 16 > nodes`

Wywołanie programu odbywa się poprzez wykonanie komendy:

`mpiexec -f nodes -n 2 ./out`

4. Dane wejściowe

Do generowania danych wejściowych został stworzony skrypt w pythonie, zwracający losową macierz kosztów. Dane generowane są do pliku „MatrixCost.dat”.

Przykładowe dane:

```
{{0, 79, 50, 52, 31, 76, 44, 88, 18, 89, 95},  
{79, 0, 98, 37, 29, 46, 19, 23, 46, 20, 29},  
{50, 98, 0, 16, 47, 99, 25, 28, 93, 1, 83},  
{52, 37, 16, 0, 1, 65, 77, 56, 73, 86, 36},  
{31, 29, 47, 1, 0, 78, 33, 95, 34, 55, 27},  
{76, 46, 99, 65, 78, 0, 9, 62, 98, 65, 17},  
{44, 19, 25, 77, 33, 9, 0, 51, 78, 45, 37},  
{88, 23, 28, 56, 95, 62, 51, 0, 79, 34, 25},  
{18, 46, 93, 73, 34, 98, 78, 79, 0, 92, 28},  
{89, 20, 1, 86, 55, 65, 45, 34, 92, 0, 56},  
{95, 29, 83, 36, 27, 17, 37, 25, 28, 56, 0}};
```

5. Dane wyjściowe

Przykładowy output programu:

```
{numRooms: 7,  
rooms: [[7, 9],  
[0, 3],  
[6, 1],  
[8, 0],  
[2, 0],  
[4, 0],  
[10, 5]],  
cost: 221}  
Process: 1 Finished Job!  
MASTER wait for messages  
MASTER Recieved FINAL_SOLUTION_FOUND form process: 1  
Process: 2 Waiting for message  
{numRooms: 7,  
rooms: [[7, 9],  
[0, 3],  
[6, 1],  
[8, 0],  
[2, 0],  
[4, 0],  
[10, 5]],  
cost: 221}  
Process: 2 Finished Job!  
MASTER wait for messages  
MASTER Recieved FINAL_SOLUTION_FOUND form process: 2  
Process: 0 Finished Job!
```