Task-to-Thread Mapping in OpenMP Using Fuzzy Decision Making

Presenter:

Mohammad Samadi

Co-authors:

Tiago Carvalho, Luis Miguel Pinho, Sara Royuela

Polytechnic Institute of Porto & INESC TEC, Porto, Portugal

Barcelona Supercomputing Center, Barcelona, Spain

Universitat Politècnica de Catalunya, Barcelona, Spain

June 2024

Introduction

- Use OpenMP to leverage HPC capabilities in CPSs.
- There is a challenge with the predictability (e.g., real-time analysis) in parallel systems.
- Perform task-to-thread mapping in OpenMP using centralized or distributed queues.
- There exists an uncertainty under unforeseen conditions with current elementary or heuristic based mappings.
- Study AI-based task-to-thread mapping in OpenMP.

Background

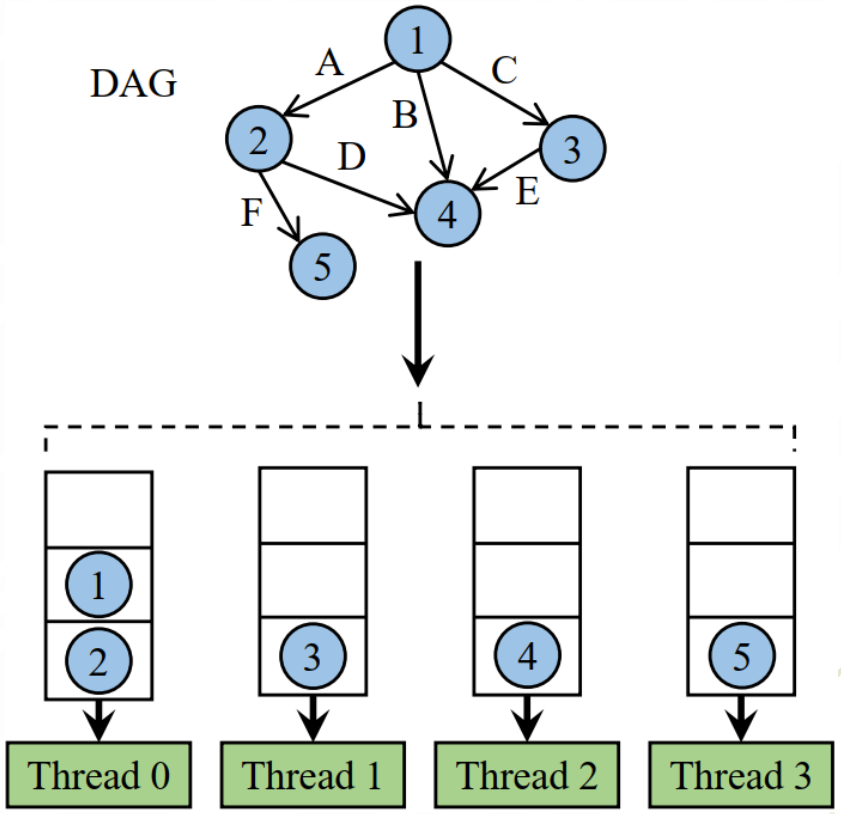
- Task-to-thread mapping
  - Single-queue (e.g., GCC) or multi-queue (e.g., LLVM) system
  - Elementary techniques, such as breadth-first scheduler (BFS)
  - Heuristic techniques, such as minimum execution time (MET)
- Artificial intelligence
  - Predefined information (i.e., training set) or intelligent procedures (e.g., heuristic path finding)
  - Fuzzy logic as a knowledge-based system in unforeseen conditions

System Model

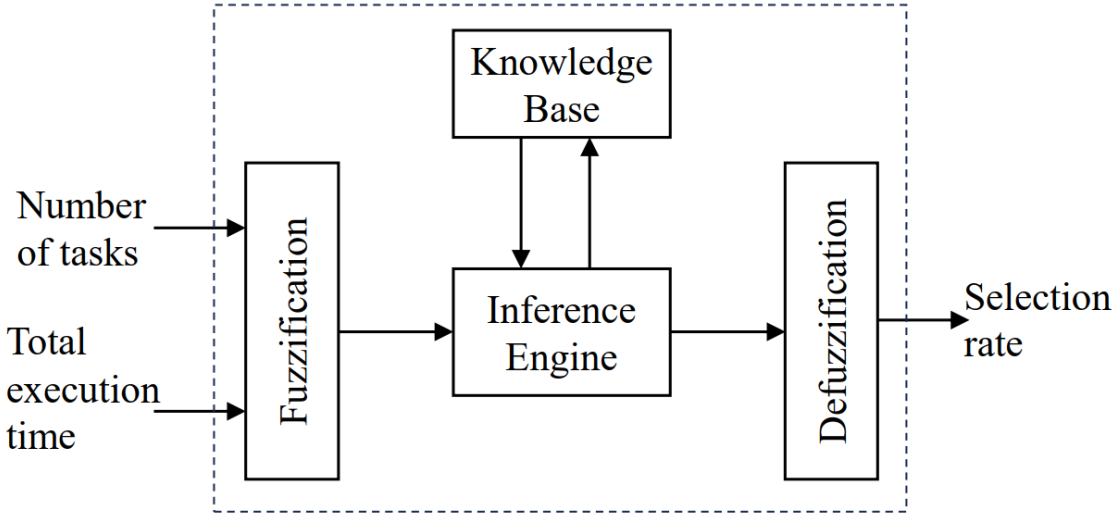
OpenMP program

```
1  #pragma omp parallel
2  #pragme omp single
3  {
4    #pragma omp task depend(out: A,B,C)
5    Task1();
6
7    #pragma omp task depend(in: A) depend(out: D,F)
8    Task2();
9
10   #pragma omp task depend(in: C) depend(out: E)
11   Task3();
12
13   #pragma omp task depend(in: B,D,E)
14   Task4();
15
16   #pragma omp task depend(in: F)
17   Task5();
18 }
```

Multi-queue tasking system



Fuzzy Controller



Rule #	Number of tasks	Total execution time	Selection rate
1	feeble	very small	very high
2	few	small	high
3	normal	mean	medium
4	many	large	low
5	lots	very large	very low

The number of (maximum) rules = $5^2 = 25$

Linguistic terms:

- Number of tasks: {feeble, few, normal, many, lots}
- Total execution time: {very small, small, mean, large, very large}
- Selection rate: {very low, low, medium, high, very high}

Data ranges for universal sets:

- Number of tasks: {0, 4}
- Total execution time: {0, 100,000,000} ns
- Selection rate: {0, 1}

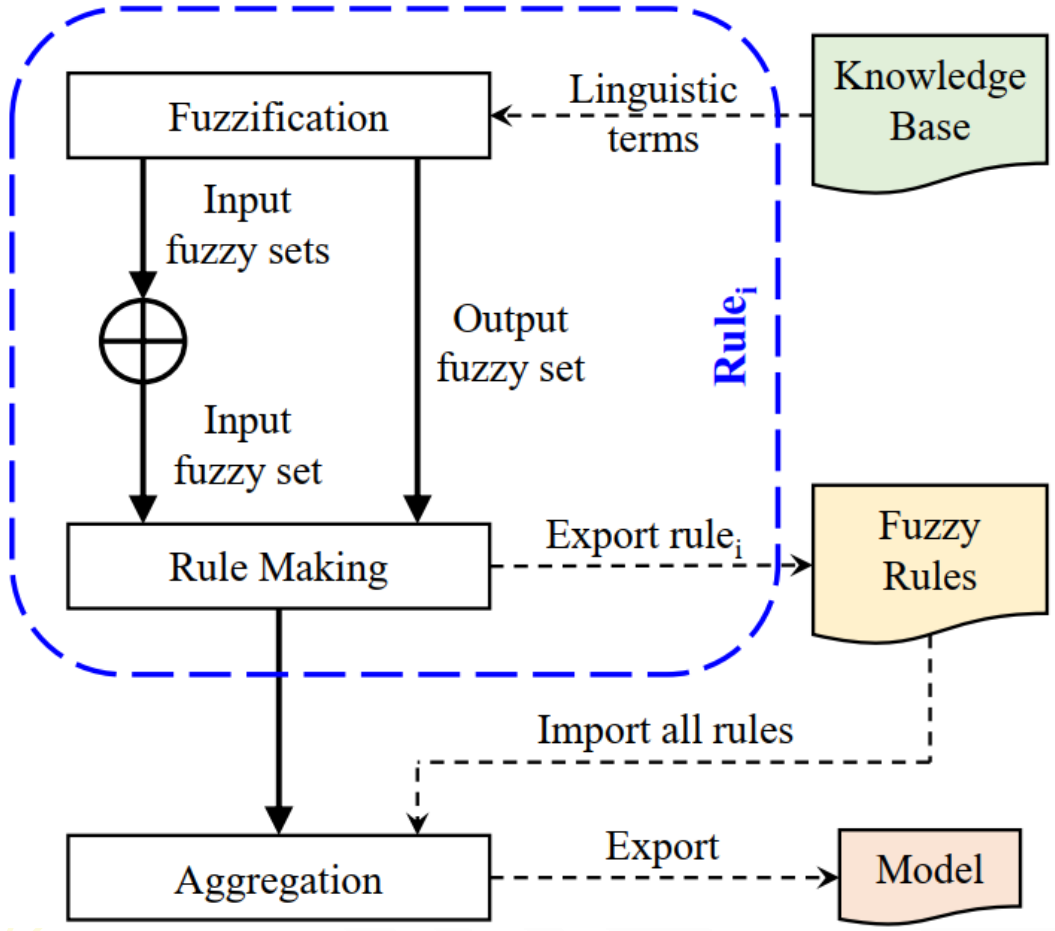
Fuzzy Controller

Components of the suggested fuzzy inference system (FIS):

- Fuzzification: Triangular membership for the input parameters and bell-shaped membership for the output parameter
- Rule making and inference engine: Mamdani rule-making type (i.e., the Maximum-Minimum function)
- Aggregation: Maximum function
- Defuzzification: Center of Gravity (also known as Center of Area or Centroid) method

Methodology

Building model using the controller



Intelligent task-to-thread mapping

```
1  N = Number of threads
2  NT[0..N-1] = Number of tasks
3  TET[0..N-1] = Total execution time
4  SR[0..N-1] = Selection rate
5
6  // Allocation phase
7  For each OpenMP task in the DAG do {
8    Update the information of queues
9    I = 0
10   While (I < N) {
11     NT_Fuzzy = Fuzzification (NT[I])
12     TET_Fuzzy = Fuzzification (TET[I])
13     Input_Fuzzy = Minimum (NT_Fuzzy, TET_Fuzzy)
14     SR_Fuzzy = Inference (Input_Fuzzy, Model)
15     SR[I] = Defuzzification (SR_Fuzzy)
16     I = I + 1
17   }
18   Select queue with highest selection rate
19   Allocate task to the queue
20 }
21
22 // Dispatching phase
23 For each thread do {
24   If thread is idle and its queue is not empty {
25     Select a ready task based on the FIFO
26     Dispatch task-to-thread and execute it
27   }
28 }
```

Future Works

- Evaluate the intelligent task-to-thread mapping using simulations and experiments under different configurations (i.e., number of tasks, types of TDGs, and number of threads, among others)
- Use real-world use cases (e.g., automotive driving) for evaluations

Thanks for your attention!

For more information:
mmasa@isep.ipp.pt
mohammad.samadi@upc.edu

isep Instituto Superior de Engenharia do Porto
BSC Barcelona Supercomputing Center Centro Nacional de Supercomputación
UPC UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH