WASEDA University

27th Reconfigurable Architectures Workshop

IPDPS 2020 New Orleans

An Interval-based Mapping Algorithm for Multi-shape Tasks on Heterogeneous Reconfigurable FPGAs

Authors: Tingyu Zhou*, Tieyuan Pan, Michael Meyer, Yiping Dong, and Takahiro Watanabe
*Email: shu-yu@asagi.waseda.jp

Outline

1. Introduction
2. Related Work
3. Problem Formulation
4. Interval-based Mapping Algorithm
   - ❖ Search and Update of Available Ranges (ARSU)
   - ❖ Intersection of Available Ranges (ARI)
5. Evaluation
6. Conclusion and Future Work

1 Introduction

Reconfigurable Hardware Device

- The functionality of logic gates be customizable at run-time.
- ✓ Higher performance, efficiency and flexibility than CPU and ASIC.
  - ► Field Programmable Gate Logic (FPGA), etc.

Dynamic Partial Reconfiguration

- Configure one part of device circuits without interrupting the execution of the rest parts.
- ✓ Multiple tasks can be executed on a single device simultaneously.

Main Function Unit

- Configurable logic block (CLB)

System Overview

Task mapping problem.

- How to decide where to place each task on the FPGA for execution?
- How to manage unoccupied programmable resources?

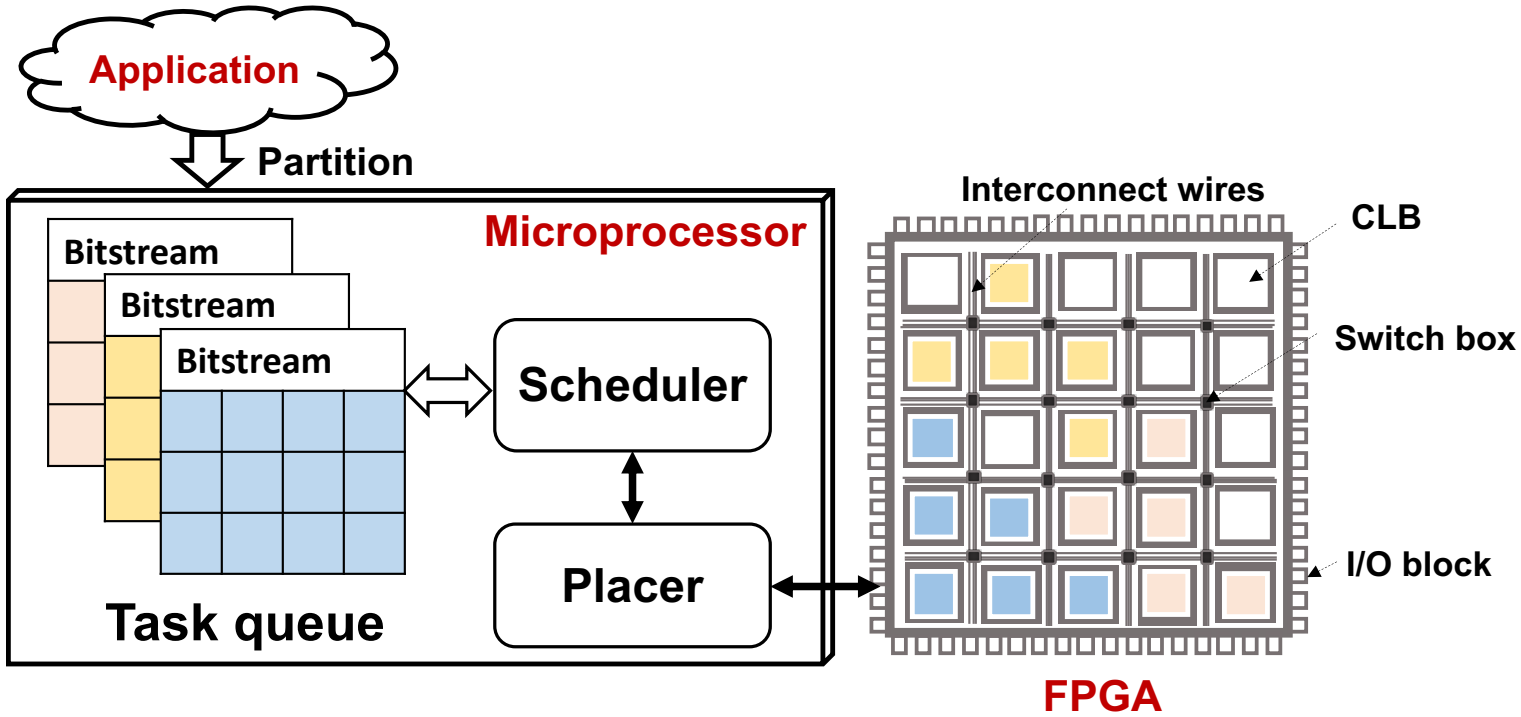


Fig.1 Overview of task execution process on the FPGA.

2 Related Work

❖ Maximal Empty Rectangle [1-2]

- ✢ Propose a maximal empty rectangle (MER) list to manage programmable resources.
- ✢ MER is an empty rectangle that can not be fully covered by other empty rectangles.
- ✕ Hardware task is simplified as a rectangular shape.
  - ▶ Internal unused area waste.

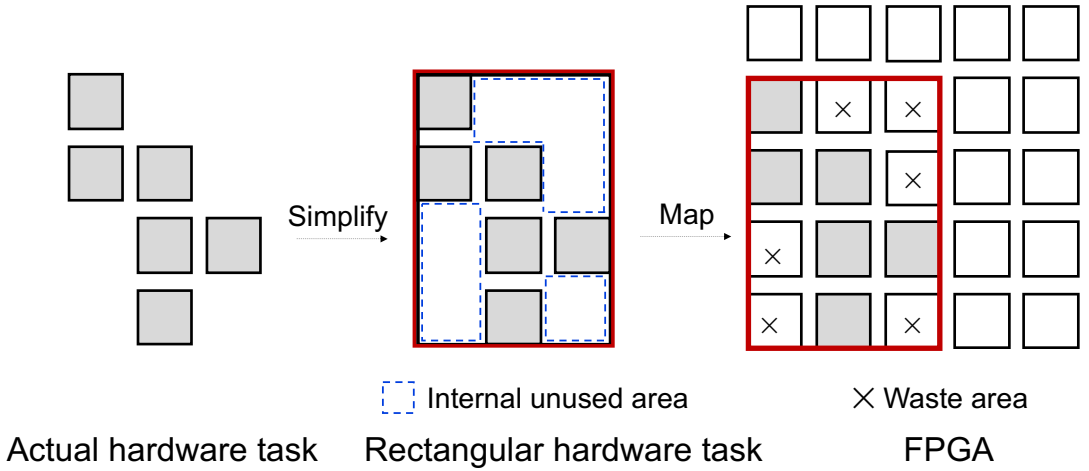


Fig.2 Waste areas due to the assumption of rectangular hardware tasks.

[1] X. Iturbe, K. Benkrid, T. Arslan, C. Hong, and I. Martinez, "Empty resource compaction algorithms for real-time hardware tasks placement on partially reconfigurable fpgas subject to fault ocurrence," in 2011 International Conference on Reconfigurable Computing and FPGAs. IEEE, 2011, pp. 27–34.
[2] T. Pan, L. Zeng, Y. Takashima, and T. Watanabe, "A fast mer enumeration algorithm for online task placement on reconfigurable fpgas," IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences, vol. 99, no. 12, pp. 2412–2424, 2016.

Best-Fit transformation (BFT) strategy [3]

- An IP core is regarded as the basic unit.
- The shape of a non-rectangular task is converted by changing the relative position between IP cores to obtain better mapping results.
- Traverse all FPGA matrix to find feasible position - low efficiency.
- The shape transform requires an additional redundant time cost.

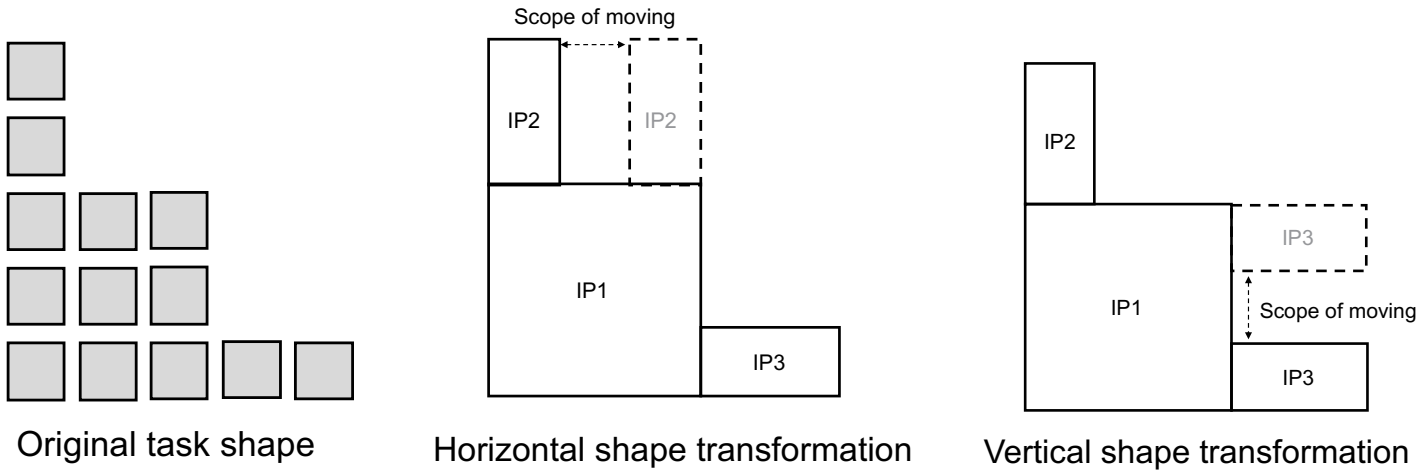


Fig.3 Non-rectangle task shape adjustment strategy.

[1] C. Wang, W. Wu, S. Nie, and D. Qian, “Bft: a placement algorithm for non-rectangle task model in reconfigurable computing system,” IET Computers & Digital Techniques, vol. 10, no. 3, pp. 128–137, 2016.

- Target Architecture: FPGA
  - Two-dimensional (2D) architecture, where CLBs are distributed in rows and columns.
- Input： Application
  - Represented as several multi-shape tasks.
  - Each task is cut into several continuous blocks $s_{(k,j)} = [b_{(k,j)}, u_{(k,j)}]$.

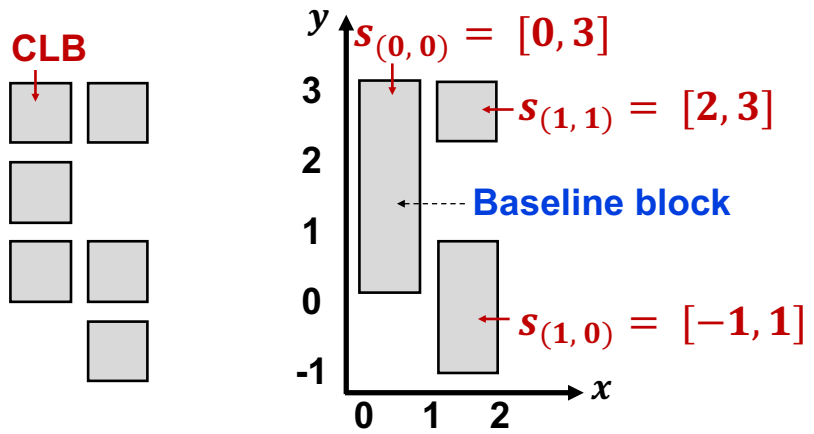


Fig.4 Multi-shape Task t and Continuous Block List S.

Interval List

- The largest continuous rectangles in each column to manage the unoccupied resources.

$$I_{(k,j)} = [b_{(k,j)}, u_{(k,j)}].$$

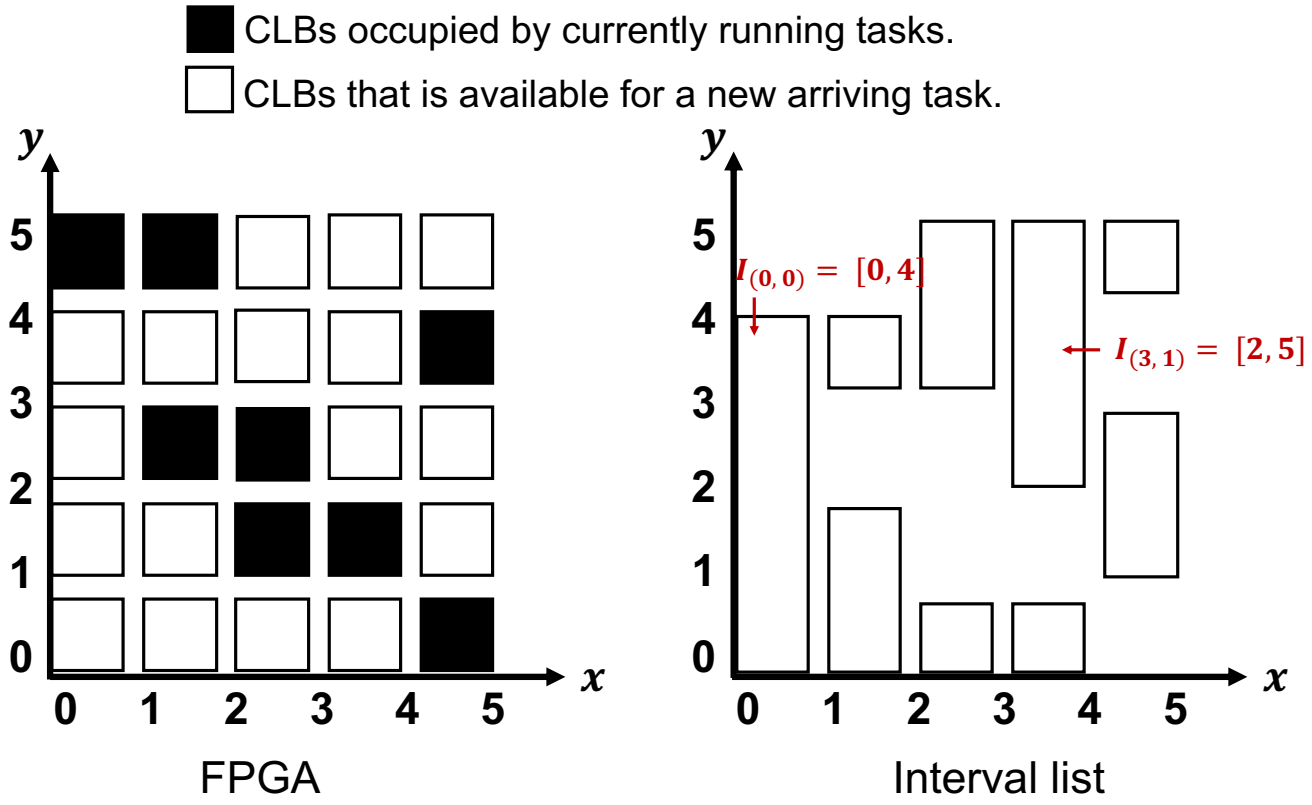
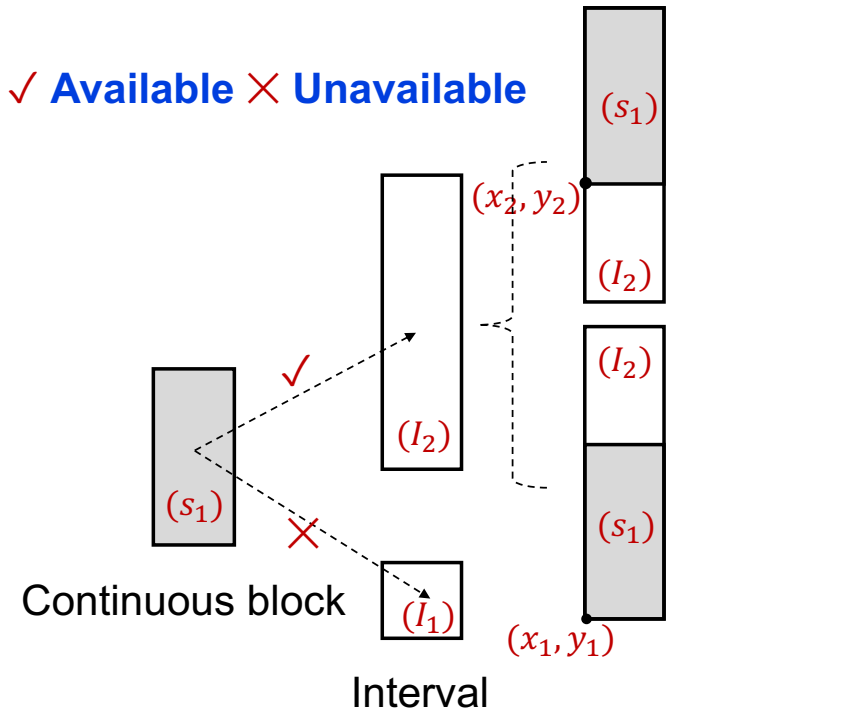


Fig.5 5 × 5 FPGA and interval list.

Table I. Intervals in each column.

Column	Interval $I_{(c,n)} = [b,t]$	
0	$I_{(0,0)} = [0,4]$	
1	$I_{(1,0)} = [0,2]$	$I_{(1,1)} = [3,4]$
2	$I_{(2,0)} = [0,1]$	$I_{(2,1)} = [3,5]$
3	$I_{(3,0)} = [0,1]$	$I_{(3,1)} = [2,5]$
4	$I_{(4,0)} = [1,3]$	$I_{(4,1)} = [4,5]$

- Search and Update of Available Ranges (ARSU): $AR_{(c,n,i)} = \{r_1, r_2, \dots, r_j\}$
  - The set of available ranges (AR) for a continuous block $s_{(c,n)}$ in the $i\text{-}th$ FPGA column, where $r_k = [y_{min}, y_{max}]$ is the range in one interval.



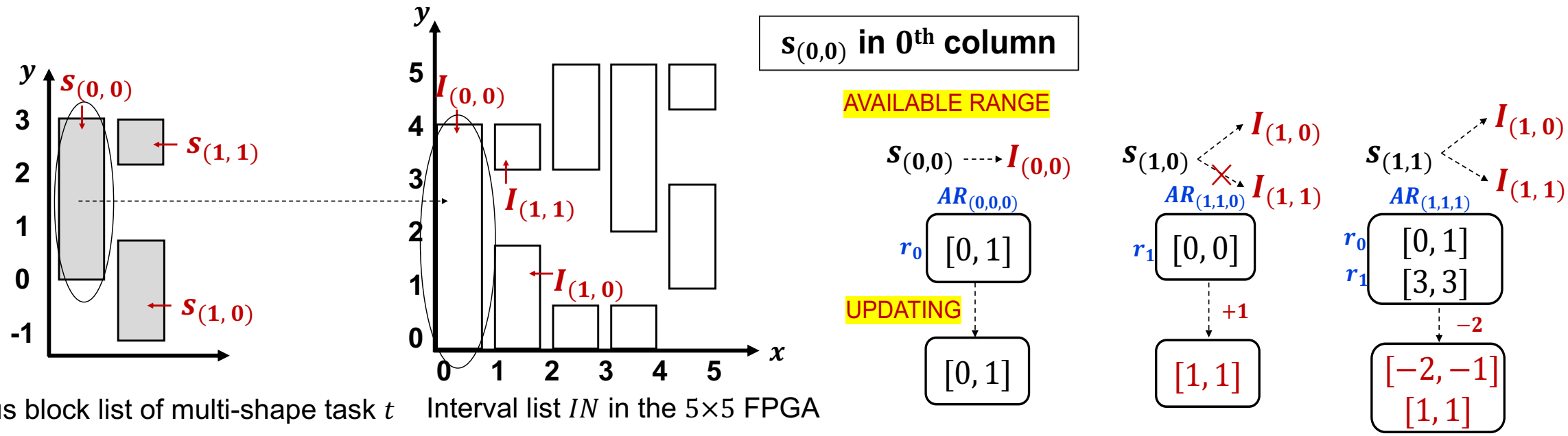
I_1 cannot accommodate s_1. ✕
I_2 can accommodate s_1. ✓ ⇢ Available range: $[y_1, y_2]$

Fig.6 Principle to search available ranges.

3 Interval-based Mapping Algorithm

❖ Search and Update of Available Ranges (ARSU)

- ✢ Search AR for the multi-shape task t from the 0-th column of the interval list.
- ✢ Update AR for each continues block based on its relative position between the baseline block $s_{(0,0)}$.



Continuous block list of multi-shape task t Interval list IN in the 5×5 FPGA

Fig.7 Map multi-shape task t in the 5×5 FPGA based on interval list IN.

Intersection of Available Ranges (ARI)

- Obtain the intersection value of all updated available ranges.

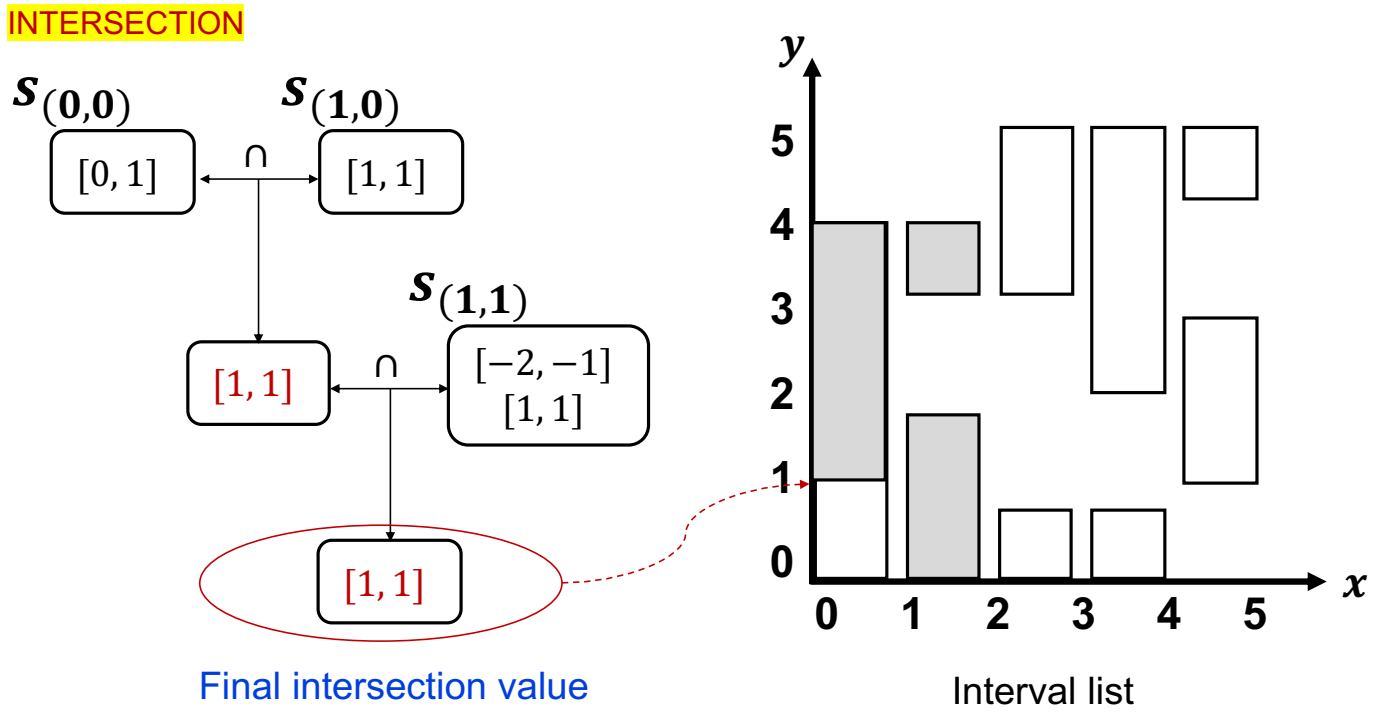


Fig.8 Intersection value of updated available ranges.

4 Evaluation

Simulation Setup

- FPGA size: 50 × 50 CLBs.
- Platform: MacOS 10.15.1, GCC 4.8 on 1.4 GHz Quad-Core Intel Core i5 Processor with 8 GB Memory.
- Task sets: TS1[1-5], TS2[5-10], TS3[5- 15] and TS4[1-15].
- Compared algorithms:
  - **Matrix-FF**: Matrix-based mapping algorithm with First-Fit selection strategy
  - **Matrix-BF**: Matrix-based mapping algorithm with Best-Fit selection strategy
  - **IMAL-FF**: Interval-based mapping algorithm with First-Fit selection strategy
  - **IMAL-BF**: Interval-based mapping algorithm with Best-Fit selection strategy
- Evaluation indicators: Mapping time, Acceptance ratio and Resource utilization ratio.

Mapping time (MT)

- Average CPU time elapsed for searching for available mapping positions for an arriving task.

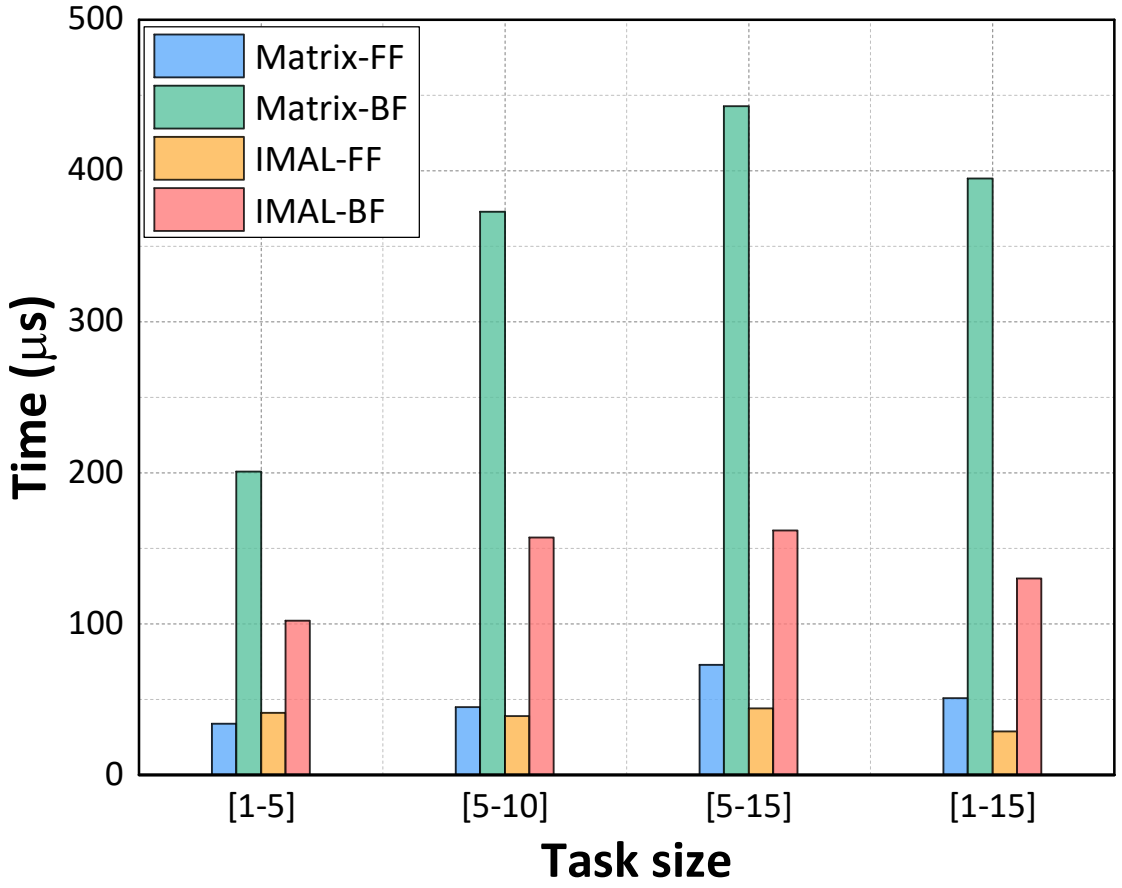


Fig.9 Mapping time evaluation.

Acceptance ratio

- The ratio of the number of tasks successfully executed on the FPGA to the total tasks.

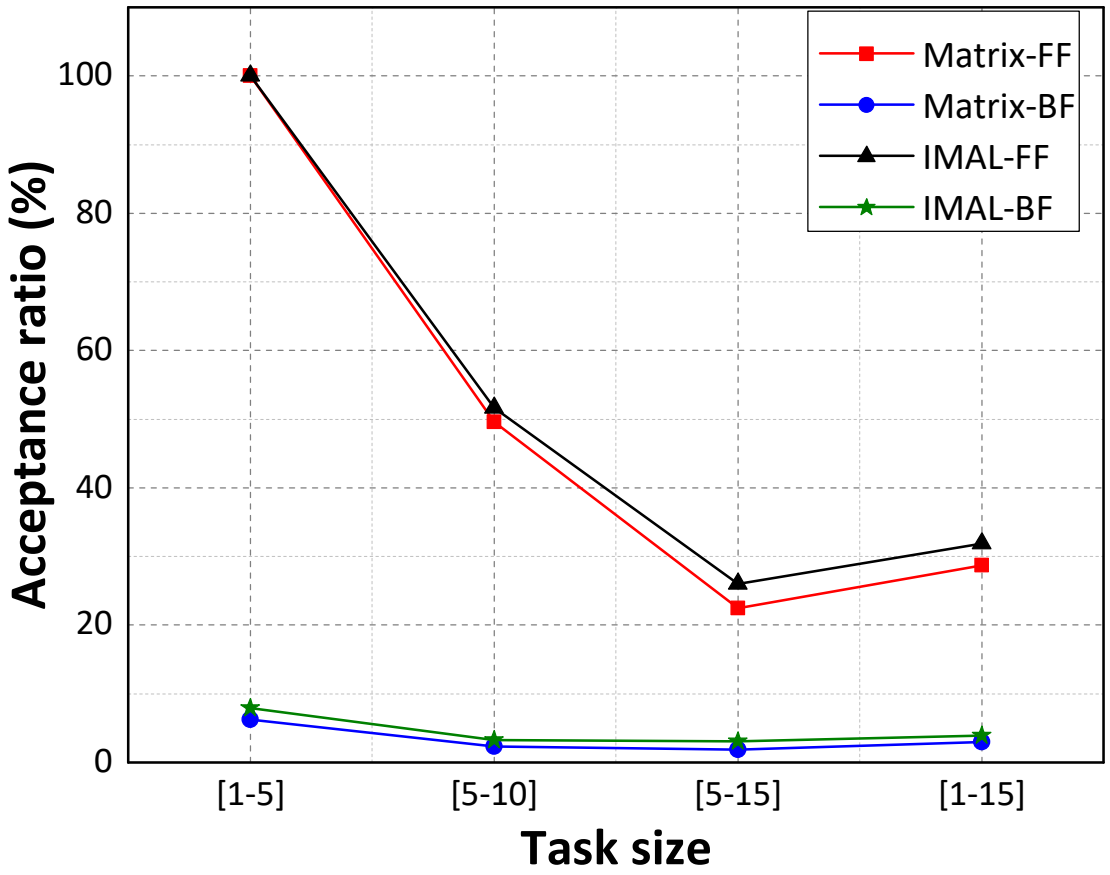


Fig.10 Acceptance ratio evaluation.

- Resource utilization ratio (RUR): $RUR = \frac{\sum_{\forall T_i \in Accept} SIZE_i \times e_i}{W \times H \times TH}$
  - $SIZE_i$ and e_i are the number of the CLBs and execution time of the task T_i.
  - W and H are the FPGA width and height.
  - ET is the elapsed time from the first task arriving at the FPGA to the completion of the last task.

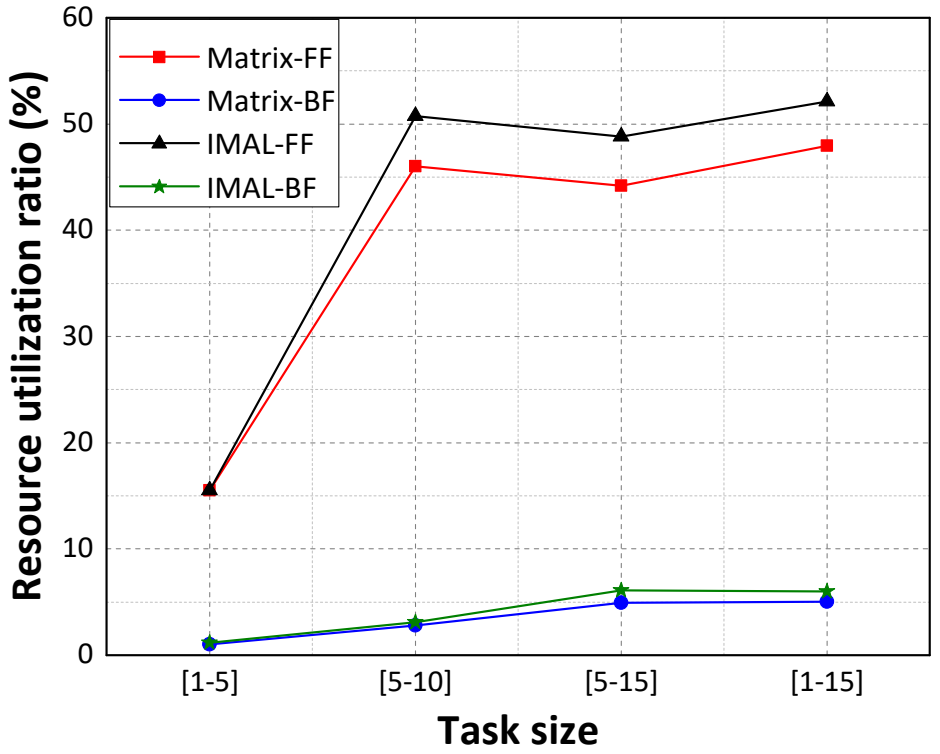


Fig.11 Resource utilization ratio evaluation.

5 Conclusion and Future Work

Conclusion

- Proposed an interval-based mapping Algorithm for multi-shape tasks on heterogeneous reconfigurable FPGA.
- The proposed approach consists of two parts: mapping and placement strategy.
- The simulation results demonstrate that the interval-based mapping algorithm is better than the traditional Matrix-based mapping algorithm in both FPGA utilization and running speed.
- Specially, the FPGA utilization ratio is improved by at least 10.3% compared with existing algorithms.

Future Work

- Do a hardware implementation for the proposed interval-based mapping algorithm on an FPGA board and evaluate it.

WASEDA University

27th Reconfigurable Architectures Workshop

IPDPS 2020 New Orleans

That's all.
Thank you for your attention!

Contact: TINGYU ZHOU <shu-yu@asagi.waseda.jp>