Data Flow Graph Partitioning Method for CGRA Temporal Mapping Based on Bayesian Optimization

Yihan Hu, Jiangnan Li, Wenbo Yin, Lingli Wang, Wai-Shing Luk

State-Key Laboratory of ASIC & System,
Fudan University, Shanghai, China

Abstract

Coarse-Grained Reconfigurable Arrays (CGRAs) are attracting more and more attention for their high flexibility and energy efficiency. Due to the limited resources, mapping large data flow graphs (DFGs) that represent application kernels onto a CGRA is difficult, for which partitioning is employed. However, existing partitioning methods in the CGRA domain are unable to solve large kernels. In this work, we propose *BOPart*, an efficient DFG partitioning method based on Bayesian optimization. This enables the mapping of large DFGs that surpass the capacity of the target CGRA. Moreover, we design a graph coarsening method to reduce the complexity of the partitioning problem, which further improves the performance and convergence of *BOPart*. *BOPart* can handle benchmarks with up to 333 operations, surpassing the capability of state-of-the-art temporal mapping and partitioning method, which can only handle benchmarks with up to 94 operations.

Contributions

1) We propose *BOPart*, a DFG partitioning method based on Bayesian optimization, which enables the mapping of large DFGs that exceed the capability of target CGRA.
2) We design a graph coarsening method to efficiently reduce the complexity of the partitioning problem, which enables the *BOPart* to deal with larger DFGs.
3) Experiments shows that *BOPart* can partition large DFGs that contain up to 333 operations, surpassing the state-of-the-art temporal mapping and partitioning method (up to 94 operations) in the CGRA domain.

Background

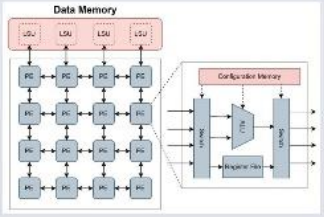

Fig. 1 A typical CGRA architeture

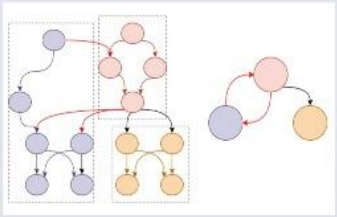

Fig. 2 Partitioning and Deadlock

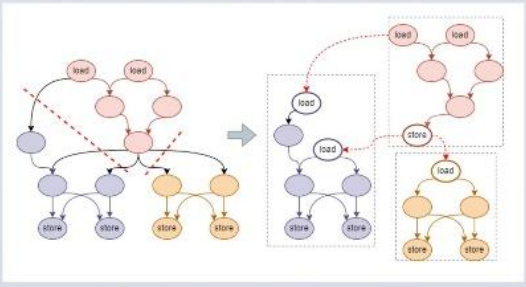

Fig. 3 Extra LSUs introduced by partitioning

It is hard to design an analytic and derivable function that describes the objective of the partitioning problem in CGRA. Therefore, bayesian optimization, a sequential model-based global optimization method for black-box functions, can be applied for CGRA partitioning.

Proposed Algorithm

We use Tree-structured Parzen Estimator (TPE), which utilizes mixture-of-Gaussian models to form the surrogate model and uses the expected improvement (EI) as the acquisition function, to model the solution space of partitioning problem.

Let A represents the adjacent matrix of the quotient graph generated by partitioning, n represents the number of operations in each subgraph. Considering the edge-cuts size, the balance of resources and edge-cuts between subgraphs, and the validity of partitioning, cost function is defined as:

$$cost(x) = \alpha \sum_{j=1}^{K} \sum_{k=1}^{K} A_{jk} + \beta(max(n) - min(n)) + \gamma(max(A_+) - min(A_+) + validity(A)$$

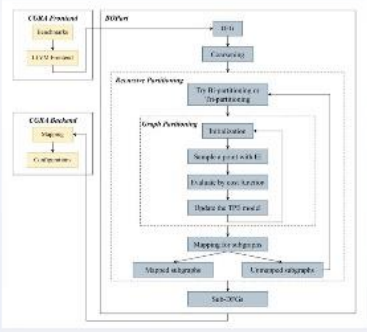

Fig. 4 Overview of *BOPart*

Algorithm 3: Graph Coarsening

Procedure: *Coarsening*(G);
Input: DFG G_S, number of subgraphs K
Output: Coarsened DFG G_C

$G_C = G$;
$TotalNum$ = Number of vertices in G ;
updated = True ;
while *updated* **do**
  updated = False ;
  vertices = *TopologicalSort*(G_C) ;
  foreach v_0 *in vertices* **do**
    if *size of* $v_0 \geq TotalNum/(K*2)$ **then**
      continue ;
    f_0^0 = the fanin of v_0 ;
    f_0^1 = the fanout of v_0 ;
    if $f_0^1 = 1$ **then**
      v_1 = the successor of v_0 ;
      f_1^0 = the fanin of v_1 ;
      f_1^1 = the fanout of v_1 ;
      if $f_1^0 \leq 1$ *or* $(f_0^1 \leq 2$ *and* $f_1^1 = 1)$ **then**
        Merge v_0 and v_1 in G_C ;
        updated = True ;
        break ;
return G_C ;

Algorithm 1: *GraphPartition*

Input: Original graph G, number of subgraphs K, maximum iterations $MaxIter$
Output: Best partition $\mathbf{x}^* = [x_1^*, x_2^*, ..., x_N^*]$

Generate random partition vectors X_0 to initialize the surrogate model ;
$\mathbf{x}^* = \arg\min_{\mathbf{x} \in X_0} cost(\mathbf{x})$;
foreach t *in* $[1, MaxIter]$ **do**
  $\mathbf{x}_t = \arg\max_{\mathbf{x}} EI(\mathbf{x}) = \arg\min_{\mathbf{x}} \frac{g(\mathbf{x})}{l(\mathbf{x})}$;
  Fit the mixture-of-Gaussian models $l(\mathbf{x})$ and $g(\mathbf{x})$ with $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_t\}$ and $\{cost(\mathbf{x}_1), cost(\mathbf{x}_2), ..., cost(\mathbf{x}_t)\}$;
  if $cost(\mathbf{x}_t) < cost(\mathbf{x}^*)$ **then**
    $\mathbf{x}^* = \mathbf{x}_t$;
return $\mathbf{x}^*$;

Algorithm 2: Recursive Partition

Procedure: *RecursivePartition*(G, $MaxIter$, $MaxII$);
Input: Original graph G, maximum iterations $MaxIter$, II constraint $MaxII$
Output: Subgraphs $\mathbf{G}^* = \{G_1, G_2, ...\}$

$\mathbf{x}^*$ = *GraphPartition*(G, 2, $MaxIter$) ;
Generate subgraph set $\mathbf{G}_0$ with $\mathbf{x}^*$;
$\mathbf{G}^* = \emptyset$;
foreach G_i *in* $\mathbf{G}_0$ **do**
  if *Map*(G_i) *succeeds and II of Map*(G_i) $\leq MaxII$ **then**
    $\mathbf{G}^* = \mathbf{G}^* \cup \{G_i\}$;
  else
    $\mathbf{G}_1$ = *RecursivePartition*(G_i, $MaxIter$, $MaxII$) ;
    $\mathbf{G}^* = \mathbf{G}^* \cup \mathbf{G}_1$;
return $\mathbf{G}^*$;

Experimental Results

We use ADRES-4x4 as our target CGRA. We use seven benchmarks from ExPRESS: *fir1*, *fir2*, *feedback*, *cosine1*, *cosine2*, *matmul*, *matinv* and one benchmark from MiBench: *susan*. We chose these benchmarks because all the operations are supported by our target CGRA architecture. The statistics of the benchmarks are presented in Table I. The experiments were measured on the Intel i7-10700 CPU(2.90GHz).

TABLE I. BENCHMARKS AND RESULTS

Benchmark	#Nodes	II_{org}	II_{part}	II_{est}	Run Time/s
feedback	24	4	4	0	0.554
fir2	36	5	5	0	0.218
cosine1	40	6	6	0	29.762
fir1	66	6	7	1	3.156
cosine2	82	10	10	0	30.431
susan	109	10	11	1	20.56
matmul	109	7	10	3	6.52
matinv	333	21	28	7	357

II_{org}, II_{part}: original and after-partition II after mapping

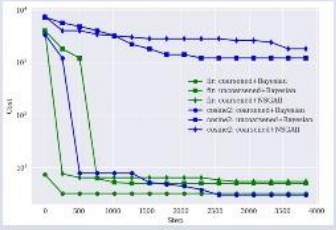

Fig. 5 Performance of different partitioning method

Conclusion

In this paper, we propose *BOPart*, a DFG partitioning method for CGRAs with temporal mapping. With Bayesian optimization and the proposed graph coarsening method, *BOPart* enables the mapping of large-scale benchmarks that contain up to 333 operations, which surpasses state-of-the-art temporal mapping and partitioning methods. Moreover, our experiments show that for DFG partitioning of CGRAs with temporal mapping, Bayesian optimization can achieve better performance than NSGA-II.

Reference

[1] G. Ansaloni, K. Tanimura, L. Pozzi, and N. Dutt, "Integrated kernel partitioning and scheduling for coarse-grained reconfigurable arrays," IEEE TCAD, vol. 31, no. 12, pp. 1803–1816, 2012.

[2] S. A. Chin and J. H. Anderson, "An architecture-agnostic integer linear programming approach to CGRA mapping," in 55th DAC, 2018.

[3] M. J. P. Walker and J. H. Anderson, "Generic connectivity-based CGRA mapping via integer linear programming," in IEEE 27th FCCM, 2019.

[4] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyperparameter optimization," NIPS, vol. 24, 2011.

[5] S. A. Chin, N. Sakamoto, A. Rui et al., "CGRA-ME: A unified framework for CGRA modelling and exploration," in IEEE 28th ASAP, 2017.

[6] T. Kojima, A. Ohwada, and H. Amano, "Mapping aware kernel partitioning method for CGRAs assisted by deep learning." IEEE TPDS 2022.