DL Accelerator Hardware-Dataflow co-design Using MAESTRO

Puya Fard, Ruchi Jagdish Patel, Sauryadeep Pal, Mengting Yang

Assistant Prof. Hyoukjun Kwon
University of California - Irvine

1 Introduction

Deep learning (DL) accelerators are critical for optimizing computational workloads across various applications. The co-design space of DL accelerators comprises three interconnected domains: the deep learning model, the dataflow, and the hardware architecture. Each of these components introduces unique constraints and opportunities for optimization.

This project aims to co-optimize dataflow and hardware architecture for given DL models `vgg16` and `UNet` using the MAESTRO cost model. The objectives are to minimize latency and energy consumption while adhering to predefined hardware and dataflow constraints. This report outlines the optimization strategies employed, decisions taken during the design process, and results obtained.

2 Methodology

The hardware optimization process was guided by specific goals related to latency, energy, and design constraints. High-level flowchart is given below Figure 1 and the following key objectives were established:

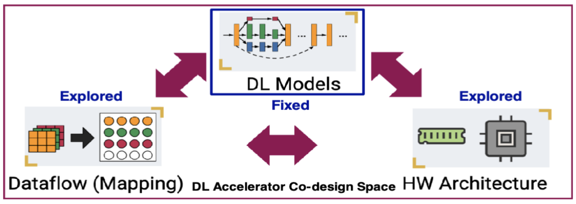


Figure 1: High-level flowchart

- **Latency and Energy Constraints:**
  - VGG16: Latency ≤ 20ms, Energy ≤ 25mJ
  - UNet: Latency ≤ 200ms, Energy ≤ 240mJ
- **Hardware Design Constraints:**
  - Processing Elements (PEs)≤ 4096
  - Network-on-Chip (NoC) bandwidth: ≤ 256 GB/s
- **Dataflow Constraints:**
  - Predefined template restriction.

```
Your Directive 1(?, ?) ?;
Your Directive 2(?, ?) ?;
Your Directive 3(?, ?) ?;
Your Directive 4(?, ?) ?;
TemporalMap(Sz(R), Sz(R)) R;
TemporalMap(Sz(S), Sz(S)) S;
Cluster(Your SZ, P);
SpatialMap(?,?) ?;
```

2.1 Accelerator Design

An iterative approach was adopted to configure the hardware parameters, which were evaluated for their impact on performance metrics such as latency and energy. This process involved adjusting the following key parameters in the `my_accelerator.m` file:

- **Number of Processing Elements (`num_pes`):** Configurations of PEs were explored at 128, 256, 512, 1024, and 2048 count.
- **L1 Cache Size (`l1_size_cstr`):** 4 KB (4096 Bytes).
- **L2 Cache Size (`l2_size_cstr`):** 100 KB (102400 Bytes).
- **Network-on-Chip Bandwidth (`noc_bw_cstr`):** 256 Bytes per cycle, the maximum allowed.
- **Off-Chip Bandwidth (`offchip_bw_cstr `):** 256 Bytes per cycle.

2.2 Dataflow Design

The design process involved iterative testing with MAESTRO to evaluate the effect of different configurations on latency and energy on `vgg16` and `UNet` DL Models. The given template for dataflow has been utilized. The parameter choices for `TemporalMap`, `SpatialMap`, and `Cluster` were informed by:

- **Data Dependency:** Ensuring minimal off-chip memory access by maximizing data reuse within clusters and caches.
- **Parallelism:** Balancing the workload across PEs to achieve high throughput without exceeding hardware constraints.
- **Hardware Constraints:** Aligning with the maximum PE count and NoC bandwidth set by `my_accelerator.m`.

3 Findings

3.1 Hardware Accelerator Design

The evaluation of various hardware configurations revealed insights into the trade-offs between parallelism, latency, and resource utilization. By analyzing the impact of different processing element (PE) counts, the design process identified configurations that optimized performance:

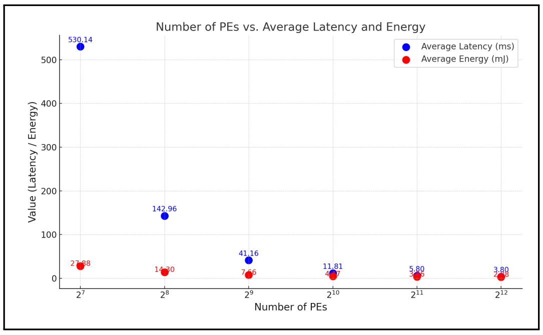


Figure 2: UNet Efficiency Analysis

After careful observation of the trend provided in Figure 2, From 2^6 to 2^{12} PEs, the total latency and energy is reduced significantly. Although 2^{12} PEs achieves the lowest latency and energy, the rate of improvement diminishes beyond 2^{11} PEs. Therefore, the optimal configuration is identified at 2^{11} PEs, providing an effective balance between performance and number of PEs utilized.

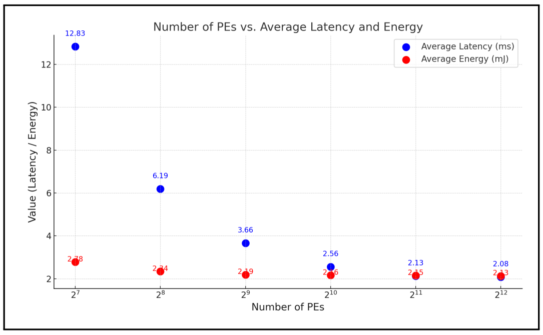


Figure 3: VGG16 Efficiency Analysis

Similarly, on Figure 3 latency and energy consumption decreases as the number of PEs increases, but the rate of reduction becomes negligible beyond 2^{10} PEs. However, considering that this hardware accelerator model is utilized for both dataflow models, the most optimal configuration for both architectures is achieved at 2^{10} PEs. Therefore the `my_accelerator.m` file will have the following parameters:

Hardware Accelerator Configuration

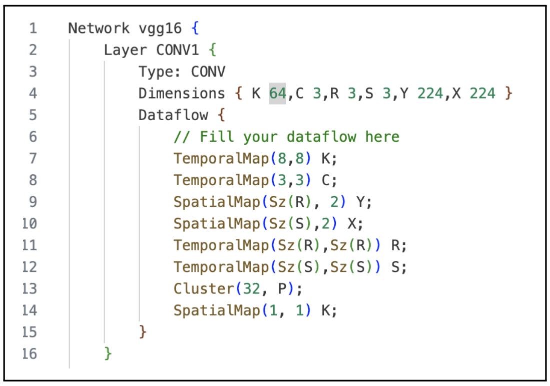
num_pes: 1024
l1_size_cstr: 4096
l2_size_cstr: 102400
noc_bw_cstr: 256
offchip_bw_cstr: 256

3.2 Dataflow Design for DL Model

In this section, we will outline the logic used to design the dataflow for each convolution layer in both DL architectures. While the underlying logic remains consistent, its application varies as it is optimized for each layer based on the specific dimensions of that convolution layer.

- Identifying the layer dimensions (K, C, R, S, Y, X) for convolutional and transposed convolutional layers.

- Optimizing `TemporalMap` for dimensions `K`, `C` and `SpatialMap` for dimensions `Y`, `X`. Identify parameter for `Cluster` and `SpecialMap` that lays below it.



```
Network vgg16 {
    Layer CONV1 {
        Type: CONV
        Dimensions { K 64,C 3,R 3,S 3,Y 224,X 224 }
        Dataflow {
            // Fill your dataflow here
            TemporalMap(8,8) K;
            TemporalMap(3,3) C;
            SpatialMap(Sz(R), 2) Y;
            SpatialMap(Sz(S),2) X;
            TemporalMap(Sz(R),Sz(R)) R;
            TemporalMap(Sz(S),Sz(S)) S;
            Cluster(32, P);
            SpatialMap(1, 1) K;
        }
    }
```

Figure 4: Snapshot of CONV1

- **TemporalMap(8,8) K and TemporalMap(3,3) C**
  These parameters divide the output channels (K=64) and input channels (C=3) into manageable chunks. Processing 4 output channels and 3 input channels at a time align well with the hardware's computational capacity.This mapping avoids overloading the L1 and L2 buffers while maintaining high throughput.

- **SpatialMap(Sz(R),2) Y and SpatialMap(Sz(S),2) X**
  These parameters distribute the spatial dimensions (Y=224, X=224) across 2 PEs for parallel processing. Diving the spatial dimensions into chunks ensures balanced workload distribution while minimizing buffer usage for shared memory. The divisibility of Y and X by 8 ensures compatibility and avoids unnecessary overhead in the memory.

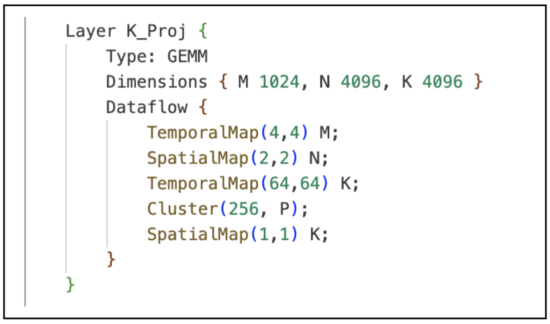
- **Cluster(32, P):** Grouping 32 PEs provides an optimal trade-off between memory sharing and computational parallelism. Parameters used across the model is either 8, 16, 32, or 64.

In conclusion, the most critical aspect of this process is ensuring that the chosen parameter values are divisible by the corresponding dimensions. This strategy minimizes buffer usage and ensures that the hardware accelerator does not exceed the L1 and L2 constraints. Additionally, keeping parameter values as low as possible is essential to avoid over-utilization of resources.

Iterative testing and validation of these parameters throughout the design process have been key to successfully determining the complete dataflow model.

3.3 Optimizing llama3_variant Dataflow

Similarly to the UNet and VGG16 dataflow models, this dataflow also follows the same logic to achieve optimized latency and energy consumption.



```
Layer K_Proj {
    Type: GEMM
    Dimensions { M 1024, N 4096, K 4096 }
    Dataflow {
        TemporalMap(4,4) M;
        SpatialMap(2,2) N;
        TemporalMap(64,64) K;
        Cluster(256, P);
        SpatialMap(1,1) K;
    }
}
```

Figure 5: Snapshot of K_Proj

As could be observed from Figure 5 above, the parameters for `TemporalMap M`, `SpatialMap N`, and `Cluster` has been modified. Previously the parameters were:

- TemporalMap (1,1) M → TemoralMap (4,4) M
- SpatialMap (1,1) N → SpatialMap (2,2) N
- Cluster (64, P) → Cluster (256, P)

This increases the degree of temporal parallelism, reducing the number of iterations required to process the entire dimension. Therefore, the total computation time decreases. Increasing the `SpatialMap` size from (1,1) to (2,2) allows two elements of the N dimension to be processed spatially across PEs simultaneously. Therefore, minimizing idle resources and ensuring efficient computation. Increasing the cluster size from 64 PEs to 256 PEs enables more PEs to work together on a single task.These changes have been applied across all the "GEMM" layers through the dataflow model.

The hardware accelerator used for this dl model has the following parameters:

Hardware Accelerator Configuration

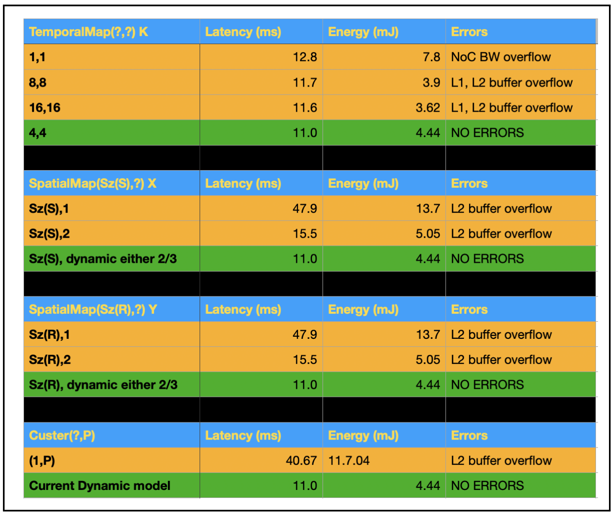
num_pes: 4096
l1_size_cstr: 4096
l2_size_cstr: 102400
noc_bw_cstr: 256
offchip_bw_cstr: 256

- **llama3_variant dataflow model**
  - Total Latency (ms): 89.13 ms
    (70.54% improved comparison to original model)
  - Total Energy (mJ): 2.12E+02 mJ
    (54.11% improved comparison to original model)

Therefore, this concludes the analysis for llama3_variant_dataflow model, indicating over 70% improvement in latency and 54% improvement in energy consuption.

3.4 Quantitative Analysis

This section focuses on evaluating the performance of dataflow model in terms of latency and energy efficiency based on `my_accelerator.m`. In order to further analyze why the current dataflow is optimal with the given hardware accelerator model, detailed analysis has been conducted to understand how each parameter value play a role in the total latency and energy.



TemporalMap(?,?) K	Latency (ms)	Energy (mJ)	Errors
1,1	12.8	7.8	NoC BW overflow
8,8	11.7	3.9	L1, L2 buffer overflow
16,16	11.6	3.62	L1, L2 buffer overflow
4,4	11.0	4.44	NO ERRORS
SpatialMap(Sz(S),?) X	Latency (ms)	Energy (mJ)	Errors
Sz(S),1	47.9	13.7	L2 buffer overflow
Sz(S),2	15.5	5.05	L2 buffer overflow
Sz(S), dynamic either 2/3	11.0	4.44	NO ERRORS
SpatialMap(Sz(R),?) Y	Latency (ms)	Energy (mJ)	Errors
Sz(R),1	47.9	13.7	L2 buffer overflow
Sz(R),2	15.5	5.05	L2 buffer overflow
Sz(R), dynamic either 2/3	11.0	4.44	NO ERRORS
Custer(?,P)	Latency (ms)	Energy (mJ)	Errors
(1,P)	40.67	11.7.04	L2 buffer overflow
Current Dynamic model	11.0	4.44	NO ERRORS

Figure 6: Analysis on UNet parameters

As could be observed above, the best parameters for our design is `TemporalMap(4,4) K`. The other parameters either cause L1, and L2 buffer overflow or not as good in terms of latency and energy.

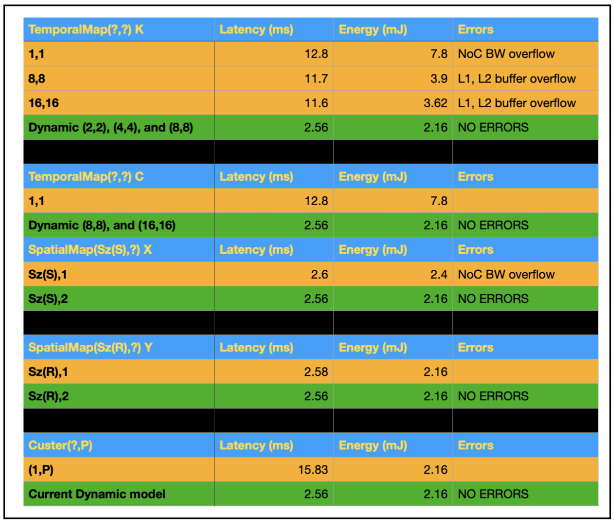
In order to optimize X and Y spatially (`SpatialMap (Sz(S), n) X; SpatialMap (Sz(R), m) Y;` its important that n <= R or S and the same for m and Y. If they're too big then more energy is used to engage additional PEs than we save by parallelization. The most important factor on deciding the parameter is **divisibility**.

These experimental parameters shows us that the current dataflow diagram has been fully optimized for the given hardware constraints. Thefore concluding that the dataflow design for DL model `UNet_dataflow.m` has the most optimal latency and energy performance with the hardware accelerator being utilized. The final optimized results in terms of latency and energy are:

- Total Latency (ms): 11.00 ms
- Total Energy (mJ): 4.44 mJ

Therefore, this concludes the analysis for UNet_dataflow model.

Now the same procedure has been applied for `vgg16_dataflow.m` dl model to understand how the parameters affect the outcome of latency and energy performance.



TemporalMap(?,?) K	Latency (ms)	Energy (mJ)	Errors
1,1	12.8	7.8	NoC BW overflow
8,8	11.7	3.9	L1, L2 buffer overflow
16,16	11.6	3.62	L1, L2 buffer overflow
Dynamic (2,2), (4,4), and (8,8)	2.56	2.16	NO ERRORS
TemporalMap(?,?) C	Latency (ms)	Energy (mJ)	Errors
1,1	12.8	7.8	
Dynamic (8,8), and (16,16)	2.56	2.16	NO ERRORS
SpatialMap(Sz(S),?) X	Latency (ms)	Energy (mJ)	Errors
Sz(S),1	2.6	2.4	NoC BW overflow
Sz(S),2	2.56	2.16	NO ERRORS
SpatialMap(Sz(R),?) Y	Latency (ms)	Energy (mJ)	Errors
Sz(R),1	2.58	2.16	
Sz(R),2	2.56	2.16	NO ERRORS
Custer(?,P)	Latency (ms)	Energy (mJ)	Errors
(1,P)	15.83	2.16	
Current Dynamic model	2.56	2.16	NO ERRORS

Figure 7: Analysis on VGG16 parameters

As we could observe on Figure 7 above, `TemporalMap K and C` parameters play a major role in terms of what the total latency and energy consumption will be. Not every parameter will work in terms of hardware constraints. It is important to make sure there won't be any L1, and L2 buffer overflows when deciding on the parameters. Table above is the analysis to conclude the fact that the current dataflow model provides the best performance in terms of latency and energy consumption considering the given hardware accelerator constraints.

- Total Latency (ms): 2.56 ms
- Total Energy (mJ): 2.16 mJ

`my_accelerator.m` with DL models works, there will be comparison between other parameters that could be used in order to understand **why this is the most optimal version for the given hardware accelerator.**

4 Conclusion

This report evaluated the design and optimization of dataflows for UNet, VGG16, and llama3_variant DL architectures using a hardware accelerator. The following key conclusions were drawn:

The evaluation of hardware configurations (Figure 2 and Figure 3) revealed significant reductions in latency and energy consumption as the number of PEs increased. For both UNet and VGG16, the optimal configuration was identified at 2^{10} PEs. The hardware accelerator parameters were finalized as:

Hardware Accelerator Configuration

num_pes: 1024
l1_size_cstr: 4096
l2_size_cstr: 102400
noc_bw_cstr: 256
offchip_bw_cstr: 256

The dataflow design focused on aligning `TemporalMap`, `SpatialMap`, and `Cluster` parameters with the layer dimensions for convolutional and transposed convolutional layers. Iterative testing ensured that chosen parameters like `TemporalMap(2,2), (4,4), or (8,8) K, TemporalMap(1,1),(4,4), (8,8), (16,16) C;` (depending on the given dimensions for that specific convolution layer) and `SpatialMap(Sz(R),m) Y; SpatialMap(Sz(S),n) X;` with `n` and `m <= R` or `S` leveraged parallelism while maintaining L1 and L2 memory constraints (Figure 4). Similarly, logical adjustments to llama3_variant dataflow (Figure 5), such as increasing TemporalMap M and N to (4,4) and (2,2) respectively, `Cluster(64, P)` to `Cluster(256, P)`, significantly enhanced temporal and spatial parallelism, resulting in a 70.54% improvement in latency and 54.11% improvement in energy efficiency.

The minimum latency (ms) and energy (mJ) achieved achieved has been quantified with strong analysis demonstrated in Findings section of the report. The end optimization results for all three DL models are:

- **VGG16**: Achieved a total latency of 2.56 ms and energy consumption of 2.16 mJ.
- **UNet**: Achieved a total latency of 11.0 ms and energy consumption of 4.44 mJ.
- **llama3_variant**: Achieved a total latency of 89.13 ms (70.54% improvement) and energy consumption of 2.12 mJ (54.11% improvement), with its corresponding Hardware Accelerator.

In conclusion, These results highlight the importance of hardware configurations and dataflow parameters to the specific requirements of each deep learning architecture.