Batch size effects on throughput and latency for ResNet50 on MN-Core 2

Kaito Yamane†, Hinata Hosoi†, Taiki Yokota†, and Takaaki Miyajima†
†Graduate School of Science and Technology, Meiji University (Japan), e-mail:takaaki.miyajima@cs.meiji.ac.jp

明治大学 MEIJI UNIVERSITY

Overview

As deep learning models continue to scale, improving inference efficiency has become increasingly important, driving the development of specialized accelerators for deep learning workloads. However, it remains unclear whether GPU-oriented optimization strategies, such as increasing batch size, are also effective on domain-specific accelerators like MN-Core 2. In this study, we evaluate the batch-size dependence of inference performance for ResNet50 on MN-Core 2 and compare it with publicly available NVIDIA V100 GPU benchmarks. Latency and throughput were measured for batch sizes ranging from 1 to 16. Our results show that **MN-Core 2 achieves nearly peak throughput at a batch size of 4, reaching approximately 90% of the maximum throughput observed at a batch size of 16,** while latency increases almost linearly. In contrast, the V100 exhibits substantial throughput gains with increasing batch size. These findings indicate that **GPU-oriented batch-size scaling strategies do not directly apply to MN-Core 2** and highlight the need for accelerator-specific inference optimization guidelines.

MN-Core 2

Key architectural characteristics:

- **Deep-learning-specific processor** optimized for computation-intensive workloads
  - High ratio of transistors allocated to **arithmetic units (7.4%) [1]**
- **Synchronous execution** across all **Processing Elements (PEs)**
  - **Single instruction stream** generated by the host CPU
  - **Efficient exploitation of data parallelism**
- **Hierarchical local-memory structure**
  - Frequently reused data kept **on chip**, reducing **external memory access**
- **MLSDK software stack**
  - Mapping computations, data placement, and instruction generation
  - **Performance strongly depends on software optimization**

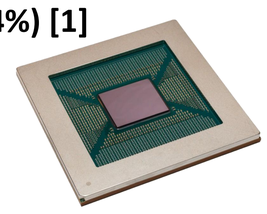

Figure 1: MN-Core 2 [2]

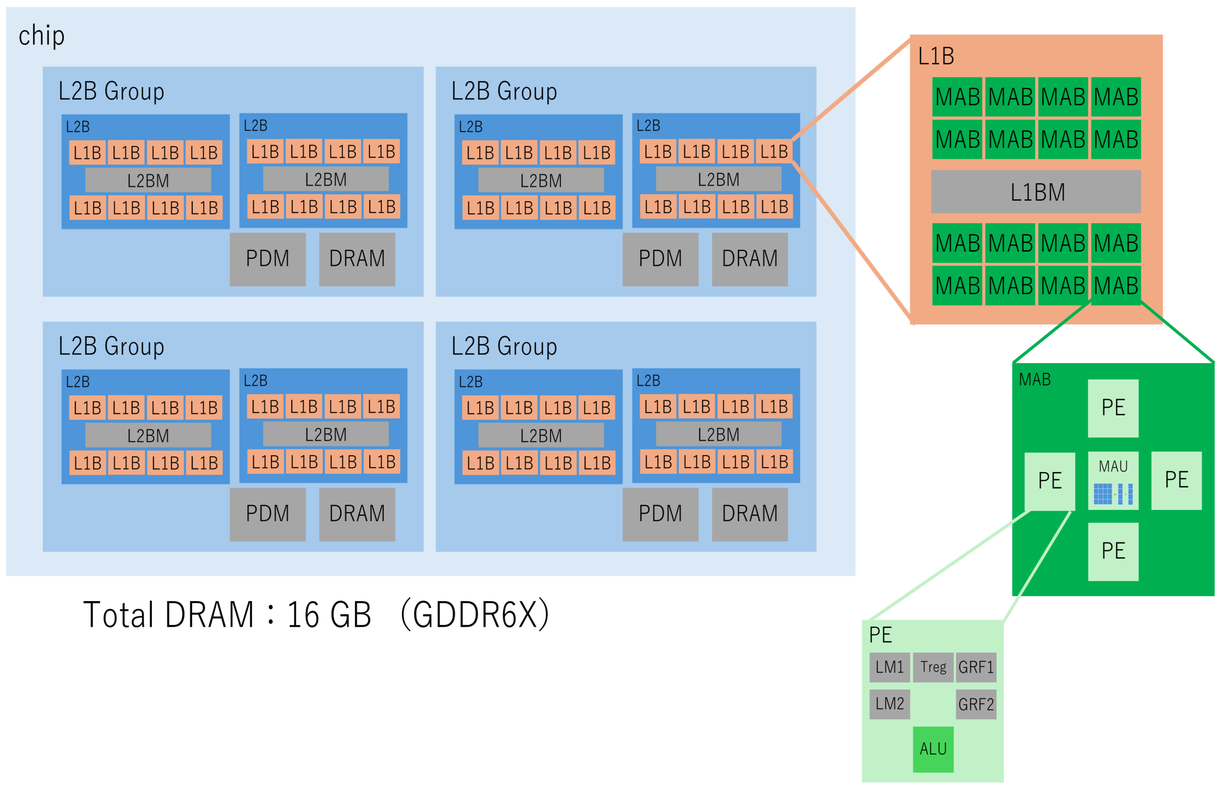

Figure 2: MN-Core 2 Architecture

Experimental Setup

- **Target model:** ResNet50 v1.5
- Compared platforms:
  - **MN-Core 2** (domain-specific AI accelerator)
  - **NVIDIA V100** (general-purpose GPU) [3]
- Batch sizes:
  - 1, 2, 4, 8, and 16
- Metrics:
  - **Latency** (processing time (sec) per batch)
  - **Throughput** (images/sec)
- MN-Core 2:
  - Precision: FP16 (AMP, automatically converted from FP32 by MLSDK v0.2)
  - Framework: PyTorch (resnet50.tv_in1k, timm)
- GPU reference:
  - NVIDIA V100 publicly available inference benchmark [3]
  - Precision: FP32
  - Framework: TensorFlow
- Note:
  - Due to differences in numerical precision and software frameworks between MN-Core 2 and the GPU benchmark, our comparison focuses on **batch-size scaling trends** rather than absolute performance values.

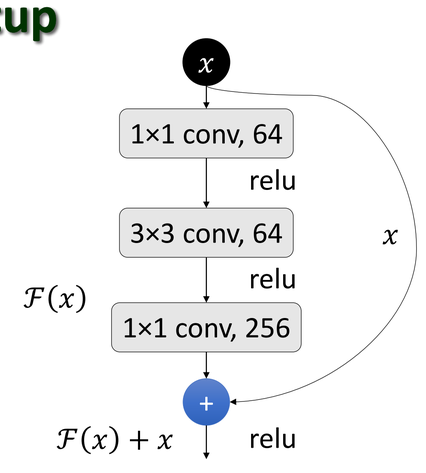

Figure 3: ResNet Architecture

Results: Latency & Throughput Scaling

Figures 4 and 5 show the latency and throughput of ResNet50 inference on MN-Core 2 and an NVIDIA V100 GPU as a function of batch size.

Key observations:

- MN-Core 2:
  - Latency increases almost linearly with batch size
  - Throughput shows only limited improvement as batch size increases
  - **At batch size 4, throughput already exceeds 90% of the maximum** observed at batch size 16
- NVIDIA V100 GPU:
  - Throughput increases substantially with batch size
  - Batch-size scaling is effective in improving throughput

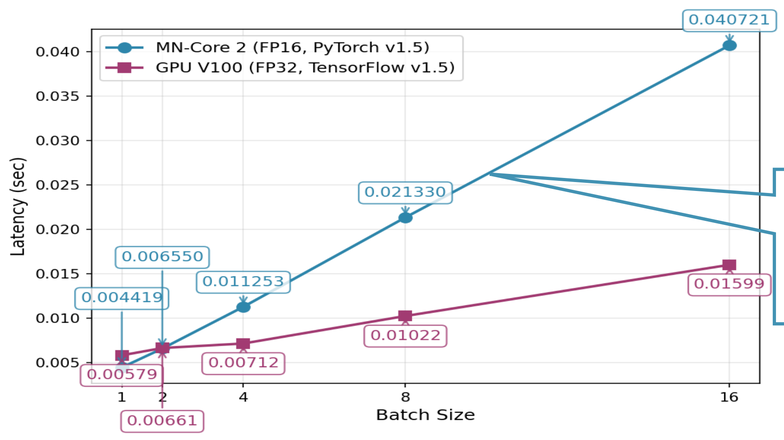

Figure 4: Latency scaling on MN-Core 2 (FP16) and NVIDIA V100 (FP32) for ResNet50 v1.5.

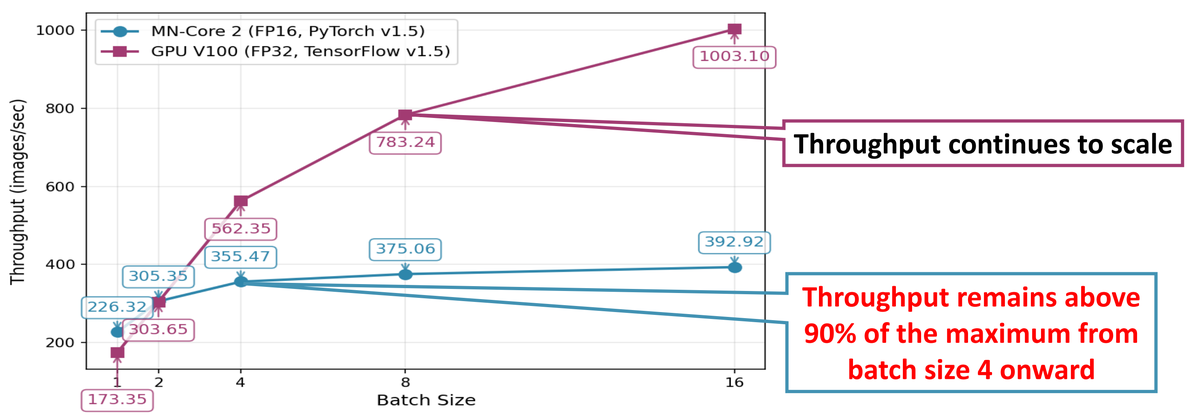

Figure 5: Throughput scaling on MN-Core 2 (FP16) and NVIDIA V100 (FP32) for ResNet50 v1.5.

Discussion & Interpretation

- The early saturation of throughput on MN-Core 2 is likely related to its synchronous execution model, in which all processing elements execute a single instruction stream.
- Unlike GPUs with hardware-level schedulers, instruction generation and data placement on MN-Core 2 are managed by the MLSDK software stack, which may limit the performance benefits of larger batch sizes.
- These architectural and software characteristics suggest that **GPU-oriented inference optimization strategies, particularly batch-size scaling, do not directly apply to MN-Core 2**, highlighting the need for accelerator-specific optimization guidelines.

Conclusion

- **MN-Core 2 achieves near-peak inference throughput at small batch sizes**, with throughput saturating early compared to GPU-based execution.
- **This result demonstrates that GPU-oriented inference optimization strategies, particularly batch-size scaling, do not directly apply to MN-Core 2**, highlighting the importance of accelerator-specific optimization approaches.

Future Work

- Further optimize the MLSDK software stack to better exploit MN-Core 2 hardware resources.
- Extend the evaluation to additional models and workloads to establish general accelerator-specific inference optimization guidelines.

References

[1] Preferred Networks, Inc. MN-Core™ 2 White Paper. Technical Report, 2023. https://projects.preferred.jp/mn-core/assets/MN-Core_2_whitepaper_en.pdf

[2] Preferred Networks. MN-Core™ Series. https://projects.preferred.jp/mn-core/ (accessed 2025)

[3] NVIDIA Corporation. ResNet-50 v1.5 for TensorFlow: Performance Benchmarks. https://catalog.ngc.nvidia.com/orgs/nvidia/resources/resnet_50_v1_5_for_tensorflow/performance