**TTNNM: Thermal- and Traffic-Aware Neural Network Mapping on 3D-NoC-based Accelerator**

The proposed Thermal-and Traffic-aware Neural Network Mapping (TTNNM) method aims to optimize temperature distribution and latency in 3D-NoC-based NN accelerators by strategically mapping NN layers based on their computational loads.

The introduction of your paper presents the context for the development of \*\*TTNNM\*\*, a mapping method tailored for 3D-NoC-based neural network accelerators. Key points include:

D Network on Chips (3D-NoCs) offer extensive on-chip wiring resources and high bandwidth capabilities. However, they face challenges such as hotspots and elevated temperature gradients resulting from increased integration and power density. These thermal issues can lead to device failure and jeopardize system stability.

In this paper, we introduce a thermal- and traffic-aware mapping method for 3D-NoC-based neural network accelerators. Firstly, we determine the mapping sequences and suitable dies for different neural network layers based on their average loads. Secondly, we allocate groups to appropriate nodes to minimize delay and alleviate hotspot temperatures.

1. \*\*Rise in Neural Network Applications\*\*: Neural networks (NNs) are widely used in fields like facial recognition and semantic analysis. As NNs grow deeper and more complex, specialized hardware, like \*\*3D Network-on-Chips (3D-NoCs)\*\*, is needed to handle the increased computational demand【6†source】【10†source】.

2. \*\*Challenges of 3D-NoCs\*\*: While 3D-NoCs offer advantages like high bandwidth and scalability, they also suffer from thermal issues due to higher integration and power density. Uneven heat dissipation and higher temperature gradients can cause hotspots, leading to system instability and potentially shorter device lifespans【10†source】.

3. \*\*Need for Effective Temperature Control\*\*: Traditional methods like task scheduling and voltage/frequency scaling help manage temperature but may introduce trade-offs such as increased latency. Mapping methods, which optimize how neural network layers are distributed across the chip, can offer a more balanced approach, improving both thermal management and performance【10†source】【8†source】.

4. \*\*TTNNM Approach\*\*:

- \*\*Load-based Layer Mapping\*\*: The method determines mapping sequences based on each NN layer’s computational load, prioritizing layers with the heaviest loads and mapping them closest to the heat sink.

- \*\*Node Allocation\*\*: TTNNM also optimizes node allocation to minimize packet latency while managing heat distribution, preventing traffic congestion and heat buildup.

\*\*Contributions\*\*:

- TTNNM improves temperature distribution and reduces traffic loads by balancing heat dissipation and communication paths. It achieves notable reductions in average temperature, temperature variance, maximum temperature, and packet latency compared to previous methods【10†source】.

In this section, the paper outlines the detailed \*\*mapping strategy\*\* of the TTNNM approach, which is designed to balance performance and heat distribution in 3D-NoC-based neural network accelerators. The strategy is divided into three fundamental steps:

### 1. \*\*Even Distribution of Computational Tasks\*\*

- The first step involves dividing the computational tasks of each neural network (NN) layer into multiple groups. This step ensures that tasks are evenly distributed across the chip, avoiding the artificial creation of \*\*hotspots\*\*, which could cause uneven heat distribution. This even distribution helps prevent areas of the chip from becoming overloaded with processing tasks, which would result in high local temperatures.

### 2. \*\*Ranking and Mapping NN Layers\*\*

- In this step, the neural network layers are \*\*ranked based on their workload\*\*. Specifically, each layer is evaluated for its computational, memory, and communication load. The layer with the highest load is prioritized for mapping onto dies closest to the \*\*heat sink\*\*. This is done because the proximity to the heat sink allows for better heat dissipation, helping to manage thermal hotspots.

- Once the heaviest layer is mapped, the next step is to choose a neighboring layer (referred to as an adjacent layer) from the current set of mapped layers (\*\*SaI\*\*). Among these, the next layer selected is the one with the highest remaining workload. This process continues iteratively until all NN layers have been mapped across the available dies in the 3D-NoC.

### 3. \*\*Fine-Grained Node Allocation\*\*

- After the layers have been assigned to the appropriate dies, TTNNM further refines the mapping by allocating the groups of each NN layer to specific \*\*nodes\*\* within those dies. This final step aims to minimize \*\*packet transmission delay\*\* and further reduce the occurrence of temperature hotspots, ensuring that traffic congestion and thermal bottlenecks are avoided.

### Key Considerations:

- \*\*Heterogeneous Thermal Transfer\*\*: TTNNM accounts for the varying thermal properties across the 3D-NoC layers. Since different layers of the chip may have distinct heat dissipation characteristics, this mapping strategy ensures that thermal transfer is optimized to reduce overall temperature gradients.

- \*\*Workload Differentiation\*\*: TTNNM takes into account the diverse computational, memory, and communication demands of different NN layers, ensuring that layers with heavier loads are handled in a way that prevents thermal imbalances and bottlenecks in data communication.

In the next subsections of the paper, the focus shifts to detailing the \*\*second and third steps\*\* (ranking the layers based on workload and node allocation), elaborating on how these steps ensure effective \*\*traffic and thermal management\*\* in the system.

**Determining the Mapping Order and Suitable Dies**

* **Disparity in Computational Load**: Different NN layers have varying computational complexities. For example, **convolutional layers** generally have a higher computational demand than pooling layers. This means the layers with greater computing, memory access, and communication needs will generate more heat.
* **Energy Consumption & Heat Accumulation**: Layers with heavier loads consume more energy, leading to **faster heat accumulation** in the processing elements (PEs) mapped to those layers. These PEs handle computations, memory accesses, and data packet transmissions.
* **Proximity to Heat Sink**: Dies that are closer to the **heat sink** can dissipate heat more efficiently. Therefore, TTNNM prioritizes mapping layers with higher workloads to these cooler areas, which helps in spreading the heat more evenly across the chip.
* **Mapping Iteration**: The process is iterative. The algorithm maps NN layers based on their load, always selecting the next layer adjacent to the previously mapped ones, prioritizing layers with the **highest remaining workload**. This continues until all layers are assigned to appropriate dies.

**2. Allocating Groups to Appropriate Nodes**

Once the layers are mapped to the dies, the next step involves assigning the groups within each layer to **specific nodes** on the 3D-NoC. This is where the algorithm optimizes both for **communication latency** and **heat distribution**:

**Step 1: Optimizing for Low Communication Hop Counts**

* Communication latency between nodes is a crucial factor. To minimize this, TTNNM selects **mapping sequences** that have lower communication hop counts. The hops between nodes are calculated using a dimensional-order routing algorithm (𝑥𝑦𝑧 routing), which determines the number of hops based on the distance in the x, y, and z directions.
* Formula (1) calculates the hop count between a source node (S) and a destination node (D), and formula (2) evaluates the total number of hops between two adjacent NN layers. By selecting mapping sequences with **hop counts below a certain threshold**, the system ensures lower communication latency between layers.

**Step 2: Restricting Traffic Load**

* After selecting mapping sequences with low hop counts, TTNNM further optimizes by limiting the **traffic load** on certain links. This step prevents network bottlenecks and ensures that data transmission is evenly distributed across the available communication links.

**Step 3: Evaluating Heat Distribution**

* Finally, the mapping sequence is chosen by evaluating the **thermal dissipation and conduction** properties of the tiles. This ensures that not only is the network optimized for low latency, but the heat generated by the PEs is also efficiently managed. This step is crucial for preventing hotspots and ensuring a uniform temperature distribution across the 3D-NoC.

**Key Takeaways:**

* The strategy prioritizes layers with heavier workloads and assigns them to cooler dies, closer to the heat sink, to **balance heat distribution**.
* It reduces communication delays by **minimizing hop counts** and carefully manages the traffic load across the network.
* Ultimately, TTNNM achieves both **low latency** and **even thermal distribution**, ensuring better performance and system stability in 3D-NoC-based neural network accelerators.

**Temperature Evaluation:**

* Each tile's temperature is evaluated based on its **energy consumption** during tasks such as computation, memory access, and packet transmission.
* The method also considers **thermal dissipation**, accounting for both heat transfer to the air and between adjacent tiles. This thermal influence diminishes exponentially with distance.
* The temperature value (TViTV\_iTVi​) of each tile is calculated based on the sum of energy consumption and the heat dissipation effects relative to the distance from the heat sink and neighboring nodes. The mapping sequence with the **minimum cumulative temperature** is selected for optimal performance.

### \*\*Evaluation\*\*

#### 4.1 Experimental Setup

The experiments were conducted using a \*\*cycle-accurate traffic-thermal co-simulation platform\*\*, called \*\*AccessNoxim\*\*. This platform integrates the \*\*Noxim NoC simulator\*\* and the \*\*Hotspot thermal model\*\*, providing detailed simulation of traffic and thermal dynamics in a 3D NoC (Network-on-Chip) environment. Table 1 provides key configuration parameters for the NoC, including:

- \*\*Mesh size\*\*: 8 × 8 × 4

- \*\*Buffer depth\*\*: 4 flits

- \*\*Packet length\*\*: 8 flits

- \*\*Routing algorithm\*\*: XYZ

- \*\*Initial temperature\*\*: 65°C

- \*\*Simulation time\*\*: 3 million cycles

The \*\*traffic traces\*\* of four different neural networks (AlexNet, VGG11, VGG13, and VGG16) were mapped to this platform to test the proposed \*\*Temperature- and Traffic-aware Neural Network Mapping (TTNNM)\*\* strategy.

#### 4.2 Experiment Results

##### 4.2.1 Temperature and Latency Overview

The performance of the proposed TTNNM was compared to three other mapping strategies: \*\*random\*\*, \*\*NN-aware\*\*, and \*\*cost model-based mapping\*\*.

\*\*Key findings\*\*:

- \*\*AlexNet\*\*: TTNNM reduced average temperature by up to 2.4°C and packet latency by 33.3% compared to random mapping.

- \*\*VGG11\*\*: TTNNM achieved a maximum temperature reduction of \*\*11°C\*\* and a 30.7% latency reduction compared to random mapping.

- \*\*VGG13\*\*: TTNNM lowered the maximum temperature by \*\*15.1°C\*\* and reduced latency by \*\*34.0%\*\*.

- \*\*VGG16\*\*: Compared to other methods, TTNNM resulted in a \*\*28.6% latency reduction\*\* and decreased maximum temperature by \*\*4.3°C\*\*.

\*\*Summary of Results\*\*:

- On average, TTNNM reduced:

- \*\*Temperature\*\* by 3.0°C compared to random, 2.2°C compared to NN-aware, and 2.4°C compared to cost model-based mapping.

- \*\*Temperature variance\*\* by \*\*58.4%, 64.8%, and 73.0%\*\*, respectively.

- \*\*Packet latency\*\* by \*\*31.7%, 17.2%, and 25.1%\*\* compared to the three strategies.

TTNNM outperformed other mapping strategies, especially in \*\*temperature reduction\*\* and \*\*latency\*\*. While NN-aware mapping optimized latency, its suboptimal performance in thermal control was resolved by TTNNM, which balances traffic, memory access, and computational loads across NoC.

##### 4.2.2 Temperature Changes Over Time

In \*\*VGG16\*\*, TTNNM consistently achieved the lowest average temperature and temperature variance throughout the simulation time. As the simulation progressed, it also minimized the maximum temperature, demonstrating sustained efficiency in thermal control.

##### 4.2.3 Temperature Distribution Across NoC

The \*\*temperature distribution\*\* across the NoC for various strategies showed that TTNNM provided a more \*\*even temperature distribution\*\*, effectively avoiding hot spots, which could degrade performance and reduce the lifespan of the NoC.

Overall, TTNNM offers significant advantages in maintaining \*\*lower and more stable temperatures\*\*, while also reducing \*\*latency\*\*, making it a robust solution for \*\*traffic-thermal-aware neural network mapping\*\*.

Thermal management is a critical concern in 3D-NoC (Three-Dimensional Network-on-Chip) architectures due to the compact and layered structure, which makes efficient heat dissipation difficult. The \*\*mapping strategy\*\* has a profound effect on both \*\*communication latency\*\* and \*\*temperature distribution\*\* across different layers in a 3D-NoC.

In the \*\*Temperature- and Traffic-aware Neural Network Mapping (TTNNM)\*\* strategy introduced here, special attention is given to how heat transfers between layers in a 3D-NoC and to the distinct computational, memory, and traffic loads imposed by different neural network (NN) layers. The \*\*TTNNM\*\* significantly improves thermal control and communication efficiency compared to conventional methods.

#### Key Results:

- \*\*Average temperature reduction\*\*:

- TTNNM achieves a temperature reduction of \*\*3.0°C\*\*, \*\*2.2°C\*\*, and \*\*2.4°C\*\* compared to random, NN-aware, and cost model-based mapping, respectively.

- \*\*Temperature variance decrease\*\*:

- TTNNM lowers temperature variance by \*\*58.4%\*\*, \*\*64.8%\*\*, and \*\*73.0%\*\*, leading to a more even heat distribution across the NoC, reducing the chances of hot spots.

- \*\*Maximum temperature reduction\*\*:

- The strategy reduces the maximum temperature by \*\*9.3°C\*\*, \*\*7.9°C\*\*, and \*\*12.0°C\*\*, improving the overall thermal stability of the chip.

- \*\*Latency improvement\*\*:

- Packet latency, a key performance factor, is reduced by \*\*31.7%\*\*, \*\*17.2%\*\*, and \*\*25.1%\*\*, demonstrating the TTNNM’s capability to optimize both thermal management and communication efficiency.

**temperature distribution**

The temperature distribution across the NoC (Network-on-Chip) is a critical factor in performance and system stability, particularly in 3D-NoCs, where heat dissipation paths are longer. As described in your summary of Fig. 4 and Fig. 5, **Thermal- and Traffic-aware Neural Network Mapping (TTNNM)** provides a more balanced thermal profile across different layers of the NoC compared to the **NN-aware mapping**.

In the TTNNM method:

* The temperature distribution is more uniform across layers, mitigating the risk of hotspots.
* Fewer tiles reach higher temperature ranges, indicating that the load distribution and traffic-aware approach lead to better thermal management.

This is crucial in reducing **thermal stress** on the chip, extending its lifespan, and maintaining **system performance** by preventing overheating. This method effectively manages heat by considering both **thermal transfer and traffic load**.

### Future Work

The current TTNNM approach could be further enhanced by employing \*\*heuristic algorithms\*\* to streamline the process of finding the optimal mapping, thereby reducing the \*\*search space\*\* and the \*\*search time\*\* required to arrive at the best thermal and traffic management strategy for 3D-NoCs.

In summary, TTNNM presents a significant step forward in addressing the thermal and latency challenges inherent to 3D-NoC-based neural network accelerators.

Limitatios:

**Heuristic Algorithm Exploration:** The paper does not thoroughly explore heuristic algorithms, which could be used to narrow down the search space and optimize the mapping further. This could reduce the time complexity of finding the optimal mapping solution.

 **Limited Focus on Dynamic Scenarios:** The proposed TTNNM approach primarily focuses on offline inference scenarios. The paper lacks consideration for runtime or dynamic scenarios, where tasks and thermal conditions can change over time. Dynamic mapping solutions could further optimize the system's adaptability to varying workloads and temperature conditions.

 **No Comprehensive Power Analysis:** While thermal distribution is addressed, power consumption (a closely related issue) is not extensively analyzed. Power and thermal optimization should ideally be considered in tandem, as reducing power consumption can further help manage thermal hotspots.

 **Scalability and Overhead Concerns:** The paper does not provide detailed discussions about the scalability of the mapping method in larger or more complex neural networks, nor does it delve into the overhead introduced by the mapping algorithm itself.