electronics

MDPI

Article

TB-NUCA: A Temperature-Balanced 3D NUCA Based on Bayesian Optimization

Hanyan Liu , Yunping Zhao , Xiaowen Chen *, Chen Li and Jianzhuang Lu

The College of Computer Science, National University of Defense Technology, Changsha 410073, China
* Correspondence: xwchen@nudt.edu.cn; Tel.: +86-1897-310-4983

Abstract: Three-dimensional network-on-chip (NoC) is the primary interconnection method for 3D-stacked multicore processors due to their excellent scalability and interconnect flexibility. With the support of 3D NoC, 3D non-uniform cache architecture (NUCA) is commonly used to organize the last-level cache (LLC) due to its high capacity and fast access latency. However, owing to the layered structure that leads to longer heat dissipation paths and variable inter-layer cooling efficiency, 3D NoC experiences a severe thermal problem that has a big impact on the reliability and performance of the chip. In traditional memory-to-LLC mapping in 3D NUCA, the traffic load in each node is inconsistent with its heat dissipation capability, causing thermal hotspots. To solve the above problem, we propose a temperature-balanced NUCA mapping mechanism named TB-NUCA. First, the Bayesian optimization algorithm is used to calculate the probability distribution of cache blocks in each node in order to equalize the node temperature. Secondly, the structure of TB-NUCA is designed. Finally, comparative experiments were conducted under random, transpose-2, and shuffle traffic patterns. The experimental results reveal that, compared with the classical NUCA mapping mechanism (S-NUCA), TB-NUCA can increase the mean-time-to-failure (MTTF) of routers by up to 28.13% while reducing the maximum temperature, average temperature, and standard deviation of temperature by a maximum of 4.92%, 4.48%, and 20.46%, respectively.

Keywords: 3D network-on-chip; non-uniform cache architecture; memory mapping; thermal balance; Bayesian optimization

check for updates

Citation: Liu, H.; Zhao, Y.; Chen, X.; Li, C.; Lu, J. TB-NUCA: A Temperature-Balanced 3D NUCA Based on Bayesian Optimization. *Electronics* **2022**, *11*, 2910. https://doi.org/10.3390/electronics11182910

Academic Editor: Jim Harkin

Received: 12 August 2022
Accepted: 7 September 2022
Published: 14 September 2022

1. Introduction

As on-chip systems continue to expand in size and the amount of processing cores integrated on chips rises, the growing volume of communication between processing units causes bottlenecks. Due to its excellent scalability and interconnect flexibility, Network-on-Chip (NoC) is evolving into the fundamental infrastructure used in Chip Multiprocessors (CMP) [1–4]. Systems, on the other hand, have higher requirements for on-chip cache in terms of capacity and speed. Therefore, CMPs are typically arranged into a non-uniform cache architecture (NUCA) in order to efficiently utilize the capacity of the last-level cache (LLC) and permit fast access latency. It gives the system non-uniform cache access latency and distributes a sizable number of the shared cache to each node in the network [5]. However, the NoC performance improvement is constrained by the rapid drop in packet latency brought on by the increased physical distance between nodes. Stacking NoC utilizing vertical links such as through-silicon via (TSV), 3D NoC is the integration of NoC and 3D stacking technology. Compared to NoC, 3D NoC can achieve a shorter communication distance, greater bandwidths, and more flexible routing, resulting in lower latency and higher performance [6].

Despite the evident benefits mentioned above, 3D NoCs encounter significant thermal problems [7,8]. Since the chips are stacked vertically, the power density becomes larger, and the thermal conduction path becomes longer. In addition, unbalanced thermal distribution in the chip is caused by various heat dissipation efficiencies between layers [9].

For instance, the top-level processor nodes that are the furthest from the heat sink encounter serious thermal issues. The temperature of the 3D NoC is a common result of the processing elements and the connecting network. In multicore system architectures that integrate a high number of cores, routers are one of the drivers of thermal issues, and the power consumption created by communication has a substantial effect on the total chip power density [10]. These thermal issues in 3D NoCs lead to degraded performance and reliability and increased package costs [11]. In addition, peak temperatures that are too high and wide temperature variations shorten the lifetime of 3D processors by hastening component aging.

Current cooling techniques can be categorized into technical approaches and architectural/algorithmic approaches [12]. In the former case, it works efficiently to eliminate hotspots by integrating extra equipment into the chip. Its drawbacks include high area and manufacturing costs. To maintain the system temperature under a predetermined thermal limit, the latter suggests using a runtime thermal management technique [13,14]. In comparison to technical methods, they can control the system temperature with less circuit/device overhead. However, they need more sophisticated technology for routing and transporting thermal information.

Reducing the overall traffic on the NoC is essential for NUCA-based systems to achieve excellent performance and low power consumption. NoC designs attribute a sizeable portion of power consumption to the energy consumed by the router, which depends on how many hops it requires for a packet to get to its target node [15]. Three-dimensional NoC can be effectively cooled down, lowering the frequency of remote accesses and the hops count those remote accesses travel across. Several hardware-based strategies have been proposed in previous studies to solve these problems, such as data replication [16], data migration, and complementary software-based approaches backed by compilers [17–19]. However, most previous compiler-based work has focused on parallelization, compute-to-core mapping, and loop transformations (modifying the execution order of loop iterations). Few studies have focused on changing the layout of the cache in node banks. In classical NUCA mapping mechanism (S-NUCA), the cache system applies block-interleaving addressing to evenly distribute the cache blocks throughout LLC banks. In other words, each bank has an equal number of cache blocks. However, due to the asymmetry of the 3D mesh structure, the traffic load of the central node is much larger than that of the peripheral nodes. As a result, the hotspots of the central node in the upper layer of the network become the main factor limiting the performance. To mitigate the thermal problem of the 3D NoC, we consider changing the NUCA uniform mapping into non-uniform mapping.

In this paper, we observe the network's traffic load distribution and temperature distribution characteristics under the conventional mapping scheme of S-NUCA and propose a temperature-balanced NUCA mapping scheme based on Bayesian optimization. To the author's knowledge, this is the first time Bayesian optimization has been employed to select 3D NUCA mapping probabilities. According to simulation data, the temperature-balanced NUCA mapping mechanism (TB-NUCA) can improve network availability and longevity by lowering the network's maximum, mean, and standard deviation of temperature compared to the traditional mapping scheme of S-NUCA. Our analysis shows that our approach can effectively balance the heat distribution among the layers of 3D NoC and enhance the network performance while keeping the throughput almost constant.

The contributions of this paper are as follows:

- We analyze the performance of 3D NoC with a uniform 3D NUCA mapping scheme and experimentally show that the topmost node's temperature is the whole structure's bottleneck;
- We design a non-uniform 3D NUCA mapping scheme that achieves better balanced thermal distribution in 3D chips;
- We propose an optimization objective with the normalized product of temperature and delay and introduce the Bayesian optimization algorithm to achieve this objective.

The remainder of this essay is structured as follows: Sections 2 and 3 describe related work, as well as the preliminary versions of NUCA and the Bayesian optimization algorithm. Section 4 presents our proposed NUCA mapping scheme. The details of the experiments and results evaluation can be found in Section 5. The last section of this essay is a conclusion.

2. Related Work

We will review the development of NUCA and some thermal management techniques in this section.

2.1. 3D NUCA

NUCA is an NoC-based system designed to address line latency issues [20–22]. This system allows fast access to cache banks adjacent to the core, thus helping to lessen the latency brought by lengthy wires in large on-chip caches. Despite the fact that all nodes share the second-level cache, the cache access latency is determined by the Manhattan distance between the source and destination nodes. When a core accesses data from an L2 bank in the same node, it is said to be local access. Otherwise, it is referred to as remote access. Figure 1 shows the average access distance of a 3 × 3 × 3 grid network with cache banks in different network nodes. It can be observed that the central cache bank of the network has a shorter average access distance compared to the peripheral cache banks.

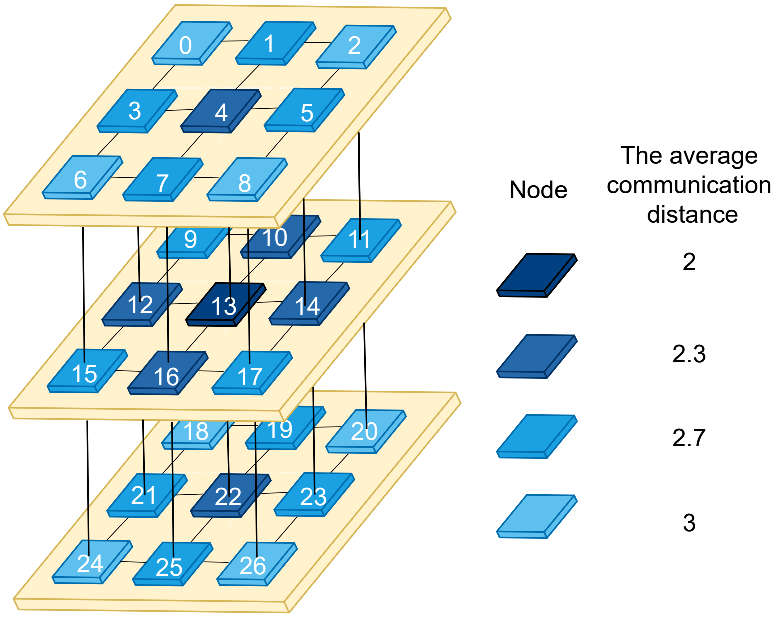


Figure 1. Average distance between nodes in a 3 × 3 × 3 mesh network.

Depending on whether a specific cache block in the memory is mapped to a distinct bank, Kim et al. proposed two NUCA schemes. On the basis of their indices in the physical address, S-NUCA maps cache blocks to certain banks. In Dynamic NUCA (D-NUCA), cache blocks are related to a group of cache banks. Cache blocks that are often accessed can be dynamically moved to the desired core to decrease cache access latency [23]. D-NUCA requires a complex search mechanism to locate cache blocks because of the flexibility of block migration, which indicates a large increase in hardware cost compared to S-NUCA. Moreover, D-NUCA is not always superior to S-NUCA. Therefore, S-NUCA is more widely used in modern CMP architectures than D-NUCA. Based on S-NUCA, our design aims to enhance it.

Prior NUCA research is centered on lowering packet hop counts between the source and destination to cut down on power consumption and cache access latency. Previous research has proposed several hardware-based strategies and complementary software-based schemes supported by compilers. Lira et al. [24] proposed HK-NUCA based on D-NUCA, which implements an efficient, low-overhead cache block search mechanism that effectively reduces failure latency and network contention. Vanapalli et al. [25] further proposed HKState-NUCA based on HK-NUCA, a structure designed with a search mechanism that reduces the number of searching messages and provides faster cache accesses and lower lapse rates than HK-NUCA. Hu and Marculescu [26] proposed using simulated annealing and genetic algorithms for communication mapping. Hung et al. [27] proposed a thermal-aware mapping and layout algorithm for 2D NoC design. Cong et al. [28] proposed a generic thermal-aware layout planning framework based on simulated annealing in response to the thermal issues in the 3D IC. Few studies have focused on changing the layout of the cache in node banks. In this research, we primarily discuss the shortcomings of the NUCA structure, i.e., uniform cache mapping under the S-NUCA structure.

2.2. Three-Dimensional Thermal Management Techniques

Three-dimensional NoCs encounter more severe thermal issues than 2D NoCs due to longer thermal routes and higher power density. We introduce the work aimed at reducing the thermal impact on improving 3D system performance from memory management and task scheduling perspectives.

2.2.1. Thermal-Aware Memory Management

Beigi and Memik [29] proposed an adaptive data placement algorithm to reduce the max temperature of a hybrid cache consisting of STT-RAM and SRAM. With an on-chip thermal sensor, the algorithm first determines each bank's temperature. The temperature data allows for the division of the banks into two categories: hot and normal. The algorithm carries out intra-bank data migration for banks that are operating normally. Cross-bank data migration is carried out by the algorithm for banks that are in a hot state. Jiang et al. [30] proposed an adaptive routing scheme for 3D NOC. According to this scheme, messages that need to be transferred are first routed using a two-dimensional intra-layer routing technique at the horizontal layer until they reach an intermediate router, after which they proceed vertically to a target router at a different layer. The lowest layer's routers are not thought to be overheating, according to the authors. If the intermediate routers are throttled, packets are transmitted to the next layer until a workable intermediate router is located. Yao et al. [31] developed a thermal-aware cell-based routing solution to obtain the power consumption of routing paths by modeling the shortest path and selecting the routing path with the least thermal power loss as the best routing path. A dynamic buffer allocation technique (DBA) was put out by Chou et al. [32] to alleviate traffic congestion in 3D NoC. The primary idea of DBA is to prevent traffic congestion by extending the input buffer and reducing the length of the output buffer of the nearly overheated router.

2.2.2. Thermal-Aware Task Scheduling

To balance the power consumption amongst the cores, Tsai and Chen [33] devised an online task scheduling technique in which workloads with high power are assigned to cores near the heatsink. Li et al. [34] proposed a greedy algorithm-based task scheduling to reduce the peak temperature of the 3D NoC. Focusing on lowering the peak temperature, Chaturvedi et al. [35] suggested a two-phase strategy with design-time optimization and runtime tweaking. Zhao et al. [36] investigated four schemes for migrating threads between cores to lessen the peak temperature and mitigate temperature variation of 3D NoC. A core-memory co-scheduling technique for real-time workloads was developed by Chaparro-Baquero et al. [37]. The technique was built on a model that guarantees resource availability while proactively and sporadically suspending request services.

3. Preliminary

3.1. Baseline 3D NUCA Mapping Scheme

Figure 2 presents our baseline architecture, which is frequently employed in research on 3D multicore systems [38–40]. Processing elements (PE) are included in the architecture as computing resources, while caches are included as on-chip storage resources. A 3D mesh network connects the 64 (4 × 4 × 4) tiles, which are dispersed over four layers. Each tile consists of a core, a private level-one (L1) instruction/data cache, a shared level-two (L2) cache bank, and a network interface that connects it with a router. Figure 2 shows the traditional mapping of memory cache blocks to the LLC bank on a 4 × 4 × 4 3D mesh network. Each memory block is statically mapped to a shared L2 cache bank according to the bank ID in the block address field. From the cache perspective, the entire physical address space is split into cells and into cache blocks. The primary memory physical address can be divided into the cache block address and the byte address. The cache block address is the index of a cache block in the primary memory within the entire physical address space, while the byte address represents the byte index within a cache block. The ID of the bank to which the cache block should be mapped is indicated by the bottom bit (6 bits) of the cache block address. Thus, cache blocks in memory are sequentially mapped to NoC.

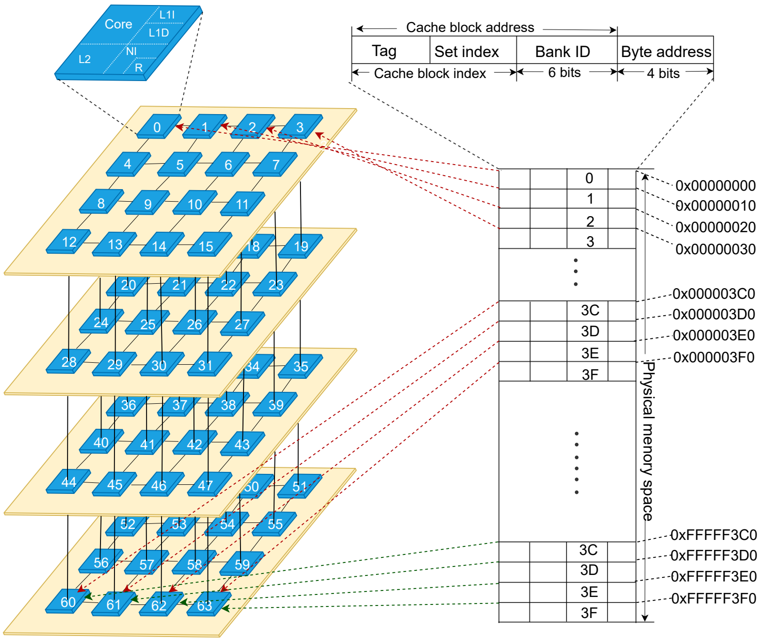


Figure 2. Baseline architecture.

Balanced application mapping mechanisms have been commonly employed to avoid network hotspots [41]. This paper will consider NUCA mapping oriented to balance temperature distribution. The cache is evenly divided over each bank as a result of the uniform mapping approach. However, the network's traffic distribution is uneven due to the asymmetry of the 3D mesh topology. As a result, the temperature of the topmost central node becomes the thermal bottleneck of the whole structure. Therefore, the algorithm in the next section is a NUCA mapping mechanism oriented to the thermal problem.

3.2. Bayesian Optimization Algorithm

Black-box optimization refers to optimization problems in which the specific expression of the optimization objective and its gradient information are unknown. The optimal global solution cannot be obtained by using the properties of the optimization target itself, and the gradient information of the parameters cannot be used directly. We can only continuously input data into the black-box function and then use the output value to guess the structural information of the black-box function. Bayesian optimization [42] is currently the most advanced black-box optimization algorithm for black-box functions with expensive validation, which can find the optimal global solution in less validation time. Many introductory machine learning courses mention grid and random search, but these two are exhaustive enumeration-like approaches. In these two searches, a random set of possible parameters is selected. The objective values are calculated separately, and finally, the best solution is obtained by comparing all the results. It can be seen that this kind of solution is very inefficient. Therefore, to solve such problems, an effective algorithm must be created in order to locate a workable solution in a short time. Bayesian optimization uses the previously observed historical information (prior knowledge) for the next optimization at each iteration, which can improve the quality of the results and the speed of the search.

Algorithm 1 shows a basic framework for basic Bayesian optimization, Sequential Model-based Global Optimization. A series of algorithms under this framework are distinguished mainly by the chosen surrogate model and the acquisition function. An optimization method based on this framework, Tree-structured Parzen Estimator (TPE) [42], is presented below.

Algorithm 1 Sequential Model-based Global Optimization.

```
Input: Expensive Function($f$), Domain($X$), Acquisition Function($S$), Surrogate Model($M$)
Output: Data Set ($D$)
$D \leftarrow$ Iintsamples($f, X$)                         ▷ Initialize the data set
for $i \leftarrow |D|$ to $T$ do                            ▷ Set the number of parameter selection to T
    $p(y|x, D) \leftarrow$ Fitmodel $(M, D)$                ▷ Fitting current data set to obtain the predictive distribution
    $x_i \leftarrow argmax_{x \in X} S(x, p(y|x, D))$       ▷ According to predictive distribution, find the extreme value point of S
    Observe $y_i \leftarrow f(x_i)$                         ▷ Expensive step
    $D \leftarrow D \cup (x_i, y_i)$                        ▷ Update the date set
end for
```

TPE uses a categorical approach to build the surrogate model. Instead of directly estimating the output of each sample point, the categorical surrogate model fuzzily slices the sample points into two categories, excellent and poor, and models them separately using Kernel Density Estimation. First, Bayes is introduced, $p(x|y)$, which is the conditional probability that the hyperparameter is x when the model loss is y. In the first step, we select the value at a certain quantile γ of y^i as the threshold y^*, i.e., $p(y < y^*) = \gamma$, based on the available data. Two probability densities, $l(x)$ and $g(x)$, are learned for samples that are above the threshold and below the threshold, respectively. That is, the TPE constructs different distributions for observations on either side of the threshold, which can be considered as a hyperparametric probability distribution for good grades, and a hyperparametric probability distribution for bad grades. These two probability densities are surrogate models.

$$f(x|y) = \begin{cases} l(x) & \text{if } y < y^* \\ g(x) & \text{if } y \geq y^* \end{cases} \tag{1}$$

where $l(x)$ represents the density created using the observations $\left\{x^{(i)}\right\}$ such that the corresponding loss $f\left(x^{(i)}\right)$ is less than y^*, and $g(x)$ represents the density created using the remaining observations.

The acquisition function chosen for this method is expected improvement (EI), which is used to select the point with the highest expected improvement (maximum expected utility).

$$EI_{y^*}(x) = \int_{-\infty}^{\infty} \max(y^* - y, 0) p(x|y) dy \tag{2}$$

According to Formula (2) for EI, which can be approximately equal to the following Equation (3), it can be found that EI can be maximized by minimizing $g(x)/l(x)$.

$$EI_{y^*}(x) = \int_{-\infty}^{y^*} \max(y^* - y, 0)p(x|y)dy \tag{3}$$

$$= \frac{\int_{-\infty}^{y^*} \max(y^* - y, 0)p(x|y)p(y)dy}{p(x)} \tag{4}$$

$$= \frac{l(x)\int_{-\infty}^{y^*} \max(y^* - y, 0)p(y)dy}{p(x)} \tag{5}$$

$$= \frac{l(x)y^*\gamma - l(x)\int_{-\infty}^{y^*} p(y)dy}{\gamma l(x) + (1-\gamma)g(x)} \alpha \left(\gamma + \frac{g(x)}{l(x)}(1-\gamma)\right)^{-1} \tag{6}$$

Therefore, we can obtain a new x^* based on the minimum $g(x)/l(x)$, then put x^* back into the data and refit $g(x)$ and $l(x)$ again, minimizing $g(x)/l(x)$ until we reach a predetermined number T.

Additionally, because TPE is a binary tree, its time complexity is $O(logN)$. The computation time C, which reflects the computation time necessary to determine the posterior parameters based on the observation cost function and the prior, is considered to be a finite constant because the computational steps of the Bayesian optimization process are fixed. Thus, the time complexity of the Bayesian search is $O(ClogN)$. It is less complicated than grid and random search complexity, $O(N)$.

When the search space is too large, the Bayesian algorithm performs faster than others. Since there are a large number of parameters for cache distribution optimization, the Bayesian optimization algorithm is the best choice.

4. Thermal-Balance Oriented Mapping Scheme

In classical 3D NUCA, cache blocks are mapped uniformly to each bank. To match the traffic load with the heat dissipation efficiency of the nodes, we want to modify the cache mapping mechanism. The distribution of the memory-to-LLC mapping is computed using a Bayesian optimization algorithm in the following discussion, and the hardware implementation is then covered.

4.1. Bayesian Optimization Algorithm Design to Calculate the Cache Distribution of Each Bank

The traffic load of a node is affected by the probability of the node being accessed and the routing frequency of the router. We adjust the node traffic load by changing the former to equal the network temperature. In the following discussion, the temperature-balanced memory-to-LLC mapping scheme will be calculated.

Since the probability distribution of the NUCA mapping and the function of temperature distribution is unknown, the gradient information of the function cannot be directly obtained. Therefore, this paper uses a black-box optimization algorithm to solve the meritocracy problem. It is assumed that the percentage of cache blocks mapped to bank i from the main memory is p_i. p_i can also be interpreted as the access possibility of bank i from the perspective of cache access. p_i is larger to indicate that more cache blocks are mapped to bank i from the main memory, thus making the bank more likely to be accessed and the traffic load of its node router higher. To be fair to the access pattern of each processing core, consider that in M cache blocks with large storage space, all processing cores will issue one request for one cache block (i.e., each processing core issues M cache access requests). Knowing that the hotspots are concentrated in the center of the layer, a temperature-balanced cache mapping will undoubtedly tend to map to the peripheral banks, which will lead to increased access latency and, consequently, lower network throughput.

To better balance performance and temperature, we set the objective function of Bayesian optimization as the product of the normalized average latency and the standard deviation of temperature. The optimization direction is to take the minimum value of the objective function. It is observed that the intra-layer distribution of the traffic of the 3D NoC shows centrosymmetric when NUCA is uniformly mapped (as shown in Figure 3a). Considering the traffic distribution and the average access distance of nodes, we divide the nodes of 4 × 4 × 4 3D NoC into 12 regions according to their symmetry (as displayed in Figure 3b). Suppose that the proportion of cache blocks mapped to a single bank in the jth region is p_j.

$$4(p_1 + p_2 + p_4 + p_5 + p_7 + p_8 + p_{10} + p_{11}) + 8(p_0 + p_3 + p_6 + p_9) = 1 \quad (7)$$

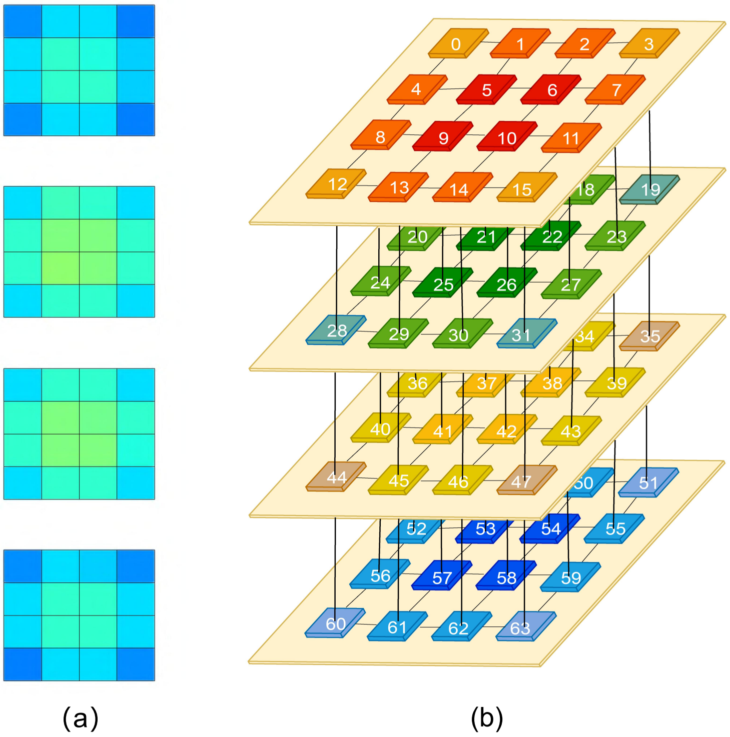


Figure 3. (**a**) Static traffic load distribution under random traffic pattern. (**b**) Architecture after chunking.

We randomly initialize a set of parameters $p_j\{p_{0j}, p_{1j}, \cdots, p_{11j}\}$ as the input and output the value of the objective function using the simulator. The input parameters and output value are integrated into the dataset as a priori information for the later selection of parameters. The loop continues until it reaches $10,000$ instances. Finally, we obtain a set of probabilities $p^*\{p_0^*, p_1^*, \cdots, p_{11}^*\}$ that minimizes the objective function. The cache mapping probability of all banks is shown in Equations (8)–(11), where P_i represents the ith layer of tensor **P**. In order to get the distribution of the cache blocks in each bank, we multiply the probability distribution matrix P by the number of blocks in the provided interval (512 blocks). We use the tensor **B** to represent the distribution of the cache blocks, as illustrated in Equations (12)–(15), where B_i represents the ith layer of **B**.

$$\mathbf{P}_0 = \begin{bmatrix} 0.0075 & 0.001 & 0.001 & 0.0075 \\ 0.001 & 0.0015 & 0.0015 & 0.001 \\ 0.001 & 0.0015 & 0.0015 & 0.001 \\ 0.0075 & 0.001 & 0.001 & 0.0075 \end{bmatrix} \tag{8}$$

$$\mathbf{P}_1 = \begin{bmatrix} 0.0155 & 0.0185 & 0.0185 & 0.0155 \\ 0.0185 & 0.005 & 0.005 & 0.0185 \\ 0.0185 & 0.005 & 0.005 & 0.0185 \\ 0.0155 & 0.0185 & 0.0185 & 0.0155 \end{bmatrix} \tag{9}$$

$$\mathbf{P}_2 = \begin{bmatrix} 0.0016 & 0.0016 & 0.0016 & 0.0016 \\ 0.0016 & 0.0175 & 0.0175 & 0.0016 \\ 0.0016 & 0.0175 & 0.0175 & 0.0016 \\ 0.0016 & 0.0016 & 0.0016 & 0.0016 \end{bmatrix} \tag{10}$$

$$\mathbf{P}_3 = \begin{bmatrix} 0.0215 & 0.0275 & 0.0275 & 0.0215 \\ 0.0275 & 0.0395 & 0.0395 & 0.0275 \\ 0.0275 & 0.0395 & 0.0395 & 0.0275 \\ 0.0215 & 0.0275 & 0.0275 & 0.0215 \end{bmatrix} \tag{11}$$

$$\mathbf{B}_0 = \mathbf{P}_0 \cdot 2^9 = \begin{bmatrix} 4 & 1 & 1 & 4 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 4 & 1 & 1 & 4 \end{bmatrix} \tag{12}$$

$$\mathbf{B}_1 = \mathbf{P}_1 \cdot 2^9 = \begin{bmatrix} 8 & 9 & 9 & 8 \\ 9 & 3 & 3 & 9 \\ 9 & 3 & 3 & 9 \\ 8 & 9 & 9 & 8 \end{bmatrix} \tag{13}$$

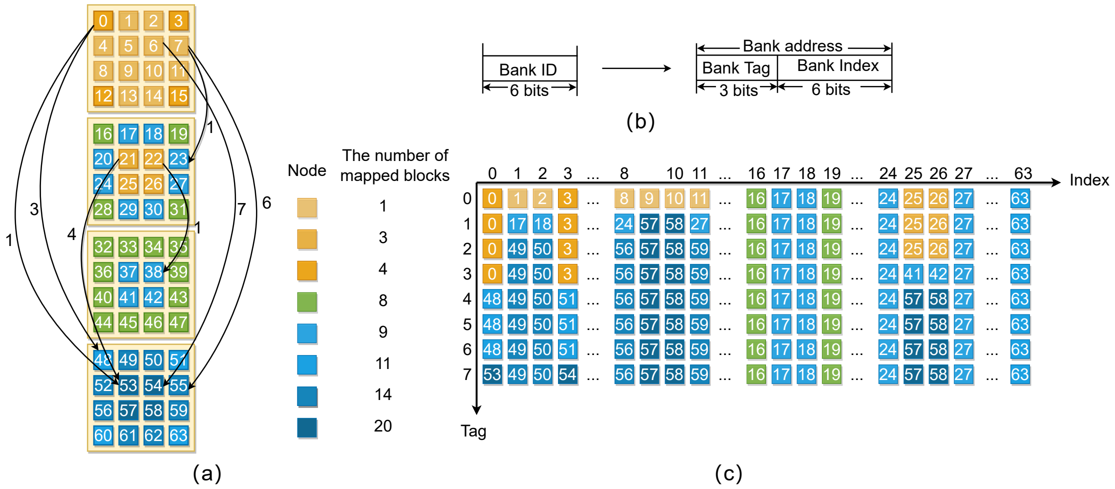
$$\mathbf{B}_2 = \mathbf{P}_2 \cdot 2^9 = \begin{bmatrix} 8 & 8 & 8 & 8 \\ 8 & 9 & 9 & 8 \\ 8 & 9 & 9 & 8 \\ 8 & 8 & 8 & 8 \end{bmatrix} \tag{14}$$

$$\mathbf{B}_3 = \mathbf{P}_3 \cdot 2^9 = \begin{bmatrix} 11 & 14 & 14 & 11 \\ 14 & 20 & 20 & 14 \\ 14 & 20 & 20 & 14 \\ 11 & 14 & 14 & 11 \end{bmatrix} \tag{15}$$

4.2. Hardware Design

Figure 4a depicts the block distribution of 3D NoC and the remapping scheme. Those arrow lines indicate that blocks mapped to the nodes in S-NUCA need to be remapped to new nodes. Since each bank is mapped with eight blocks traditionally, the green bank remains unchanged, while the blue bank should be mapped with more blocks, and the yellow bank should be mapped with fewer blocks. As a result, we transferred a few blocks from the yellow banks to the blue banks.

Next, we will consider how to implement the remapping mechanism [5]. For a 4 × 4 × 4 mesh network, using the traditional mapping mechanism, the bank address field needs 6 ($\log_2 64$) bits so as to directly map the block to the NoC. We extend the width of the bank address field to remap the cache to NoC. The 512 blocks within the interval can be sequentially separated into eight groups, each of which includes 64 blocks. We extend the bank address by 3 bits and use it as the bank tag, while the original 6 bits are used as the bank index, as depicted in Figure 4b. The former represents the group number, and the latter represents the initial bank ID in the S-NUCA. To be consistent with the new mapping mechanism, we remapped some blocks from the upper-layer bank (yellow) to the lower-layer bank (blue). Figure 4c depicts the elaborate remapping scheme throughout the interval.



	0	1	2	3	...	8		10	11	...	16	17	18	19	...	24	25	26	27	...	63
0	0	1	2	3	...	8	9	10	11	...	16	17	18	19	...	24	25	26	27	...	63
1	0	17	18	3	...	24	57	58	27	...	16	17	18	19	...	24	25	26	27	...	63
2	0	49	50	3	...	56	57	58	59	...	16	17	18	19	...	24	25	26	27	...	63
3	0	49	50	3	...	56	57	58	59	...	16	17	18	19	...	24	41	42	27	...	63
4	48	49	50	51	...	56	57	58	59	...	16	17	18	19	...	24	57	58	27	...	63
5	48	49	50	51	...	56	57	58	59	...	16	17	18	19	...	24	57	58	27	...	63
6	48	49	50	51	...	56	57	58	59	...	16	17	18	19	...	24	57	58	27	...	63
7	53	49	50	54	...	56	57	58	59	...	16	17	18	19	...	24	57	58	27	...	63

Figure 4. (**a**) The memory-to-LLC mapping scheme of TB-NUCA. (**b**) The bank address field under the remapping scheme. (**c**) The detailed result of the mapping scheme of TB-NUCA.

It is impossible to obtain the bank ID straight from the bank address field because we altered the classical mapping scheme. Therefore, each L1 cache controller needs to be equipped with an address-mapping device (addr_map), which is in charge of translating the bank address into the associated bank ID. Once the L1 cache controller submits a coherency request, the device determines the bank ID of the destination cache block in order to deliver the packet to the correct node. The device is a simple logic unit that can be implemented quickly and inexpensively with a lookup table. Addr_map is a straightforward logic unit that can be quickly and cheaply implemented via a lookup table.

Even though the analysis and findings above pertain to a mesh network of 4 × 4 × 4 nodes, similar conclusions and analogous probability distributions **P** and block distributions **B** are possible for other scale networks. Overall, the distribution trend is more blocks are mapped to the bottom layer while fewer blocks are mapped to the top layer.

5. Simulated Results

5.1. Simulated Setup

We tested the proposed mapping scheme on a Ubuntu 20.04 system with Intel(R) Core(TM) i7-10700F CPU, and SystemC [43] used is the 2.3.3 version. Simulations were performed through a cycle-accurate traffic-thermal co-simulation platform called Access-Noxim [10], which integrates NoC simulator Noxim and architecture-level thermal model hotspot. In addition, we coupled AccessNoxim with an open-source hyperparameter optimization (HPO) framework called Optuna [44]. Furthermore, we tested our proposed method based on the coupled system and auto-searched it for the optimal mapping scheme.

The experiments include two steps: first, we use Optuna to generate a set of parameters and deliver them to AccessNoxim, and we use AccessNoxim to run simulations. Next, Optuna evaluates parameters and simulation results and provides parameters for the next simulation. Steps 1 and 2 are repeated until the number of times reaches the preset value.

The simulation parameters for the network's co-simulation are shown in Table 1. The 3D NoC is a 4 × 4 × 4 mesh structured network containing 64 blocks. The flit injection rate is adjusted to 0.08 in order to raise the maximum temperature without saturating the network. Despite the absence of a memory-to-LLC mapping mechanism in synthetic traffic simulation, we may construct a specific traffic distribution corresponding to banks through a probability distribution P for accesses instead of a non-uniform mapping mechanism. The number of study trials is 1000. The experimental constraint is that the node probability is greater than 0 and the sum of the probabilities of all nodes is 1, and the probability selection step is 0.002. We evaluated the throughput, temperature, and traffic load distribution of our proposed method and compared them with the classical mapping scheme.

Table 1. Specification of parameters for simulation.

Parameter	Value
Packet size	8 flits
Buffer size	4 flits
Simulation time	1×10^5 cycles
Warm-up time	1×10^4 cycles
Mesh size	4 × 4 × 4
Traffic pattern	random, transpose-2, shuffle
Injection rate	0.08 flits/cycle/node
Ambient temperature	45 °C
Routing algorithm	XYZ

5.2. *Analysis of Thermal Distribution and Mean-Time-To-Failure (MTTF)*

Figure 5 shows the temperature distribution under random, transpose-2, and shuffle traffic patterns. The detailed maximum, mean, and standard deviation of the temperature distribution are presented in Table 2. The TB-NUCA mapping scheme exhibited a more uniform temperature distribution among the layers than S-NUCA. Since the cache distribution in the middle area of the upper two layers is reduced during cache mapping, the temperature of the critical region is decreased. Our analysis shows that, compared with S-NUCA, regarding the thermal distribution of TB-NUCA in the three traffic patterns, the maximum temperature decreases by 4.92%, 3.03%, and 4.8%, the average temperature decreases by 4.48%, 3.99%, and 3.14%, and the standard deviation of temperature decreased by 20.46%, 1.38%, and 2.43%, respectively.

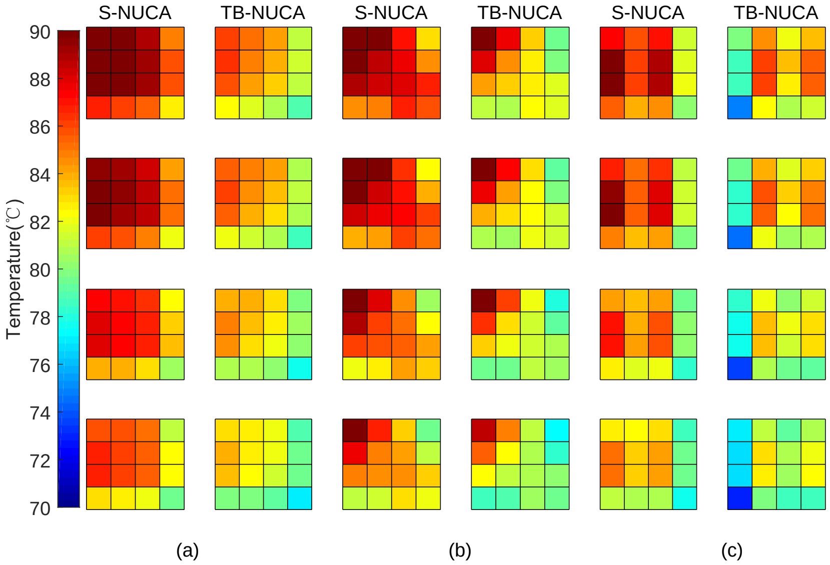


Figure 5. Temperature distribution comparison under different traffic conditions. (**a**) Random. (**b**) Transpose-2. (**c**) Shuffle.

Table 2. Comparison of temperature distribution between TB-NUCA and S-NUCA.

	Uniform-Random		**Transpose-2**		**Shuffle**	
NUCA	S-NUCA	TB-NUCA	S-NUCA	TB-NUCA	S-NUCA	TB-NUCA
Max	90.8444	86.3749	95.4394	92.5412	90.5907	86.2438
Avg.	86.1727	82.3084	85.9377	82.505	83.9708	81.0254
S.D.	2.7985	2.226	3.2783	3.233	3.1297	3.0536

Next, we count the number of hotspots for S-NUCA and TB-NUCA on 64 routers in the experimental platform under three traffic patterns. Routers with temperatures above 85°C are considered hotspots. Figure 6 shows the ratio of hot routers to total routers under three traffic patterns. As was shown, TB-NUCA reduces hotspots by 84.4%, 68.6%, and 72% on average compared to S-NUCA while enhancing the network availability by 60%, 37.5%, and 28.2%, respectively.

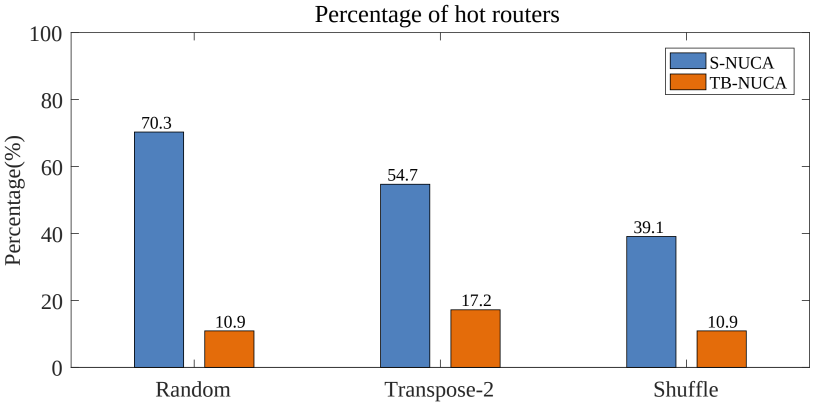


Figure 6. Percentage of hotspots for TB-NUCA and S-NUCA.

Additionally, we analyze how the remapping scheme increases the MTTF of the routers in the critical region of the NoC. Since the total number of communications before and after remapping is constant, the remapping scheme increases the utilization of some nodes with low utilization or high thermal dissipation efficiency and decreases the utilization of nodes in thermally critical areas. In other words, remapping makes the utilization of nodes match their heat dissipation efficiency. Figure 7 shows the MTTF distribution of all the routers before and after applying the remapping scheme. To visualize the effect of the remapping scheme, we have plotted the percentage of routers having a particular range of MTTF before and after remapping. The MTTF is normalized concerning the lowest MTTF before remapping. From Figure 7, we observe that the MTTF distribution of all traffic models shifts to the right region. Most importantly, the number of routers with low MTTF values decreases. Shifting the lower MTTF values towards the higher MTTF region implies that TB-NUCA improves the lifetime of the most failure-prone routers, thereby increasing the lifetime of the whole chip.

5.3. Analysis of Traffic Load Distribution and Performances

Figure 8 shows the traffic load distribution in random, transpose-2, and shuffled traffic modes. As expected, the traffic load distribution of TB-NUCA is closer to the lower layer than that of S-NUCA. In the TB-NUCA, the traffic of the lower layer is significantly higher than that of the upper layer because the cache is mapped toward the lower layer. From the experimental data, the standard deviation of flow increases by 1.3% to 4.7% on average.

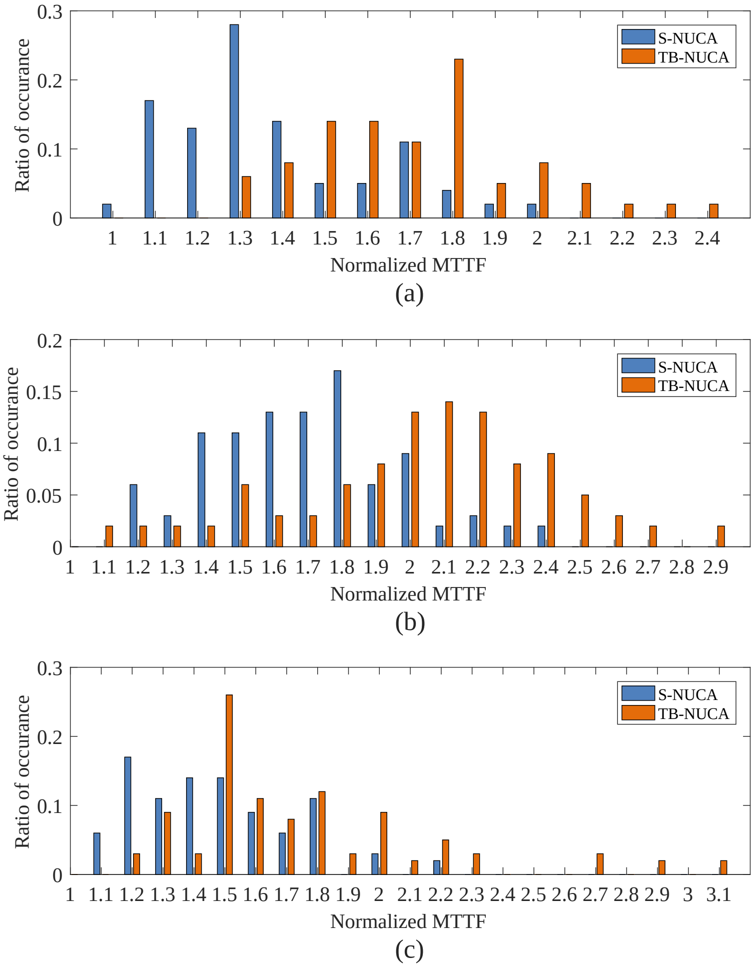


Figure 7. Effect of mapping scheme on MTTF distribution under different synthetic traffic patterns. (**a**) Random, (**b**) Transpose-2. (**c**) Shuffle.

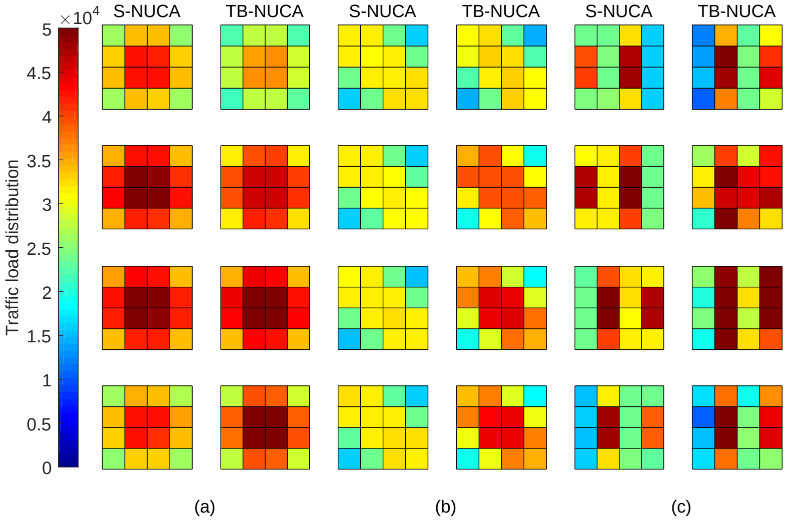


Figure 8. Traffic load distribution comparison under different traffic conditions: (**a**) Random. (**b**) Transpose-2. (**c**) Shuffle.

In TB-NUCA mapping, the traffic load of the lower layer increases, which leads to serious traffic congestion in the lower layer. Throughput can be affected by traffic congestion.

Table 3 shows the throughput, average latency, and power of three traffic patterns. Knowing that the hotspots are concentrated in the center of the layer, a temperature-balanced cache mapping will undoubtedly tend to map to the peripheral banks, which will lead to an increase in latency and fluctuation in network throughput. Although the average latency of TB-NUCA was slightly larger than that of S-NUCA, the variations in throughput and total power were within 1%. All in all, the TB-NUCA mapping scheme did not have much impact on the performance overhead.

Table 3. Throughput comparison under different conditions for TB-NUCA and S-NUCA.

	Uniform-Random		**Transpose-2**		**Shuffle**	
NUCA	S-NUCA	TB-NUCA	S-NUCA	TB-NUCA	S-NUCA	TB-NUCA
Throughput (flits/cycle/node)	0.0789	0.0794	0.0794	0.0788	0.0793	0.0789
Average Latency (cycles)	9.669	9.8734	7.773	8.449	8.221	8.669
Total Power (J)	0.0244	0.0242	0.0239	0.0238	0.0212	0.0213

5.4. Simulation on Benchmark

Furthermore, to demonstrate the effectiveness of the proposed algorithm, we select a program from the list of PARSEC 2.1 [45] benchmarks to perform the experiments. The simulation is performed under NoximAccess with the trace of the program, which is acquired through the trace simulator. The other experimental conditions were kept consistent with those of the synthetic traffic. Experimental results are shown in Table 4.

As shown in Table 4, compared with S-NUCA, TB-NUCA achieved a reduction in maximum temperature, average temperature, and standard deviation of temperature by 41%, 16.9%, and 58.4%, respectively. In addition, TB-NUCA reduced the percentage of hotspots from 16% to 0%, which greatly improved the availability of the network.

Table 4. Performance of blackscoles under TB-NUCA and S-NUCA mapping scheme.

	Blackscholes			
	Max Temp.	Avg. Temp.	S.D. Temp.	Hotspots pct.
S-NUCA	133.636	70.0672	16.8398	14%
TB-NUCA	78.8563	58.2297	7.00026	0%

6. Discussions

The proposed method, TB-NUCA, essentially changes the traffic distribution of the network by changing the cache distribution, allowing more traffic to be distributed in the lower layers. As a result, the traffic matched the ability of the heat dissipation capacity of nodes and then balanced the temperature of the on-chip network. The simulated results showed that compared to the classical schemes, TB-NUCA had a more balanced thermal distribution, a lower standard deviation of the thermal distribution, a longer MTTF, and fewer hotspots. The method proposed in this paper requires offline search, which consumes certain computing resources. Additionally, the variations in throughput and total power were within 1%, and the average latency of TB-NUCA was slightly larger than that of S-NUCA. In the future, we will take optimizing latency performance into account.

7. Conclusions

Huge power and thermal densities lead to bottlenecks in the system performance of the 3D chip-stacked NoCs. In this work, we proposed a temperature-balanced memory mapping scheme named TB-NUCA to alleviate the thermal problem of 3D NoCs that is usually caused by huge power density or imbalanced traffic. In order to solve the thermal problem, our proposed TB-NUCA mapping scheme was searched offline by the Bayesian optimization algorithm. We designed an objective function that was the product of the average packet transmit delay of the whole chip and the maximum temperature of the nodes. Optimizing the constructed objective function to obtain the best cache distribution of the mapping scheme has successfully alleviated the thermal problem. In summary, our method provides a new solution to the thermal problem of 3D NoCs, which is also the basic guarantee for our long-term development of it.

Author Contributions: Conceptualization, H.L. and X.C.; methodology, H.L.; software, H.L.; validation, H.L. and Y.Z.; formal analysis, H.L.; investigation, H.L.; resources, J.L.; data curation, H.L.; writing—original draft preparation, H.L.; writing—review and editing, Y.Z., X.C., C.L. and J.L.; visualization, H.L.; supervision, X.C.; project administration, H.L.; funding acquisition, X.C. All authors have read and agreed to the published version of the manuscript.

Funding: This paper and the research results were supported by the Scientific and Technological Innovation Talents Project. The specific grant number is 22-TDRCJH-02-018. H.L., Y.Z., C.L., J.L. and X.C. all received this grant.

Data Availability Statement: The data presented in this study are available on request from corresponding authors.

Acknowledgments: The authors thank the support from the Scientific and Technological Innovation Talents Project under grant 22-TDRCJH-02-018.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Cui, Y.; Prabhakar, S.; Zhao, H.; Mohanty, S.; Fang, J. A Low-cost Conflict-free NoC architecture for Heterogeneous Multicore Systems. In Proceedings of the 2020 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), IEEE, Limassol, Cyprus, 6–8 July 2020; pp. 300–305.
2. Ma, S.; Jerger, N.E.; Wang, Z. DBAR: An efficient routing algorithm to support multiple concurrent applications in networks-on-chip. In Proceedings of the 2011 38th Annual International Symposium on Computer Architecture (ISCA), San Jose, CA, USA, 4–8 June 2011; pp. 413–424.
3. Zheng, H.; Wang, K.; Louri, A. Adapt-noc: A flexible network-on-chip design for heterogeneous manycore architectures. In Proceedings of the 2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA) IEEE, Seoul, Korea, 27 February–3 March 2021; pp. 723–735.
4. Indragandhi, K.; Jawahar, P. Core Performance Based Packet Priority Router for NoC-Based Heterogeneous Multicore Processor. In *Intelligent System Design*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 389–397.
5. Wang, Z.; Chen, X.; Lu, Z.; Guo, Y. Cache access fairness in 3d mesh-based nuca. *IEEE Access* **2018**, *6*, 42984–42996. [CrossRef]
6. Momeni, M.; Pozveh, A.J. An adaptive approximation method for traffic reduction in network on chip. In Proceedings of the 2020 6th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS) IEEE, Mashhad, Iran, 23–24 December 2020; pp. 1–5.
7. Black, B.; Annavaram, M.; Brekelbaum, N.; DeVale, J.; Jiang, L.; Loh, G.H.; McCaule, D.; Morrow, P.; Nelson, D.W.; Pantuso, D.; et al. Die stacking (3D) microarchitecture. In Proceedings of the 2006 39th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO'06) IEEE, Orlando, FL, USA, 9–13 December 2006; pp. 469–479.
8. Qian, Y.; Lu, Z.; Dou, W. From 2D to 3D NoCs: A case study on worst-case communication performance. In Proceedings of the 2009 IEEE/ACM International Conference on Computer-Aided Design-Digest of Technical Papers IEEE, San Jose, CA, USA, 2–5 November 2009; pp. 555–562.
9. Jiang, X.; Lei, X.; Zeng, L.; Watanabe, T. Fully adaptive thermal–aware routing for runtime thermal management of 3D network–on–chip. In Proceedings of the International MultiConference of Engineers and Computer Scientists, Hong Kong, China, 16–18 March 2016; pp. 659–664.
10. Jheng, K.Y.; Chao, C.H.; Wang, H.Y.; Wu, A.Y. Traffic-thermal mutual-coupling co-simulation platform for three-dimensional network-on-chip. In Proceedings of the 2010 International Symposium on VLSI Design, Automation and Test IEEE, Hsin Chu, Taiwan, 26–29 April 2010; pp. 135–138.
11. Yeo, I.; Liu, C.C.; Kim, E.J. Predictive dynamic thermal management for multicore systems. In Proceedings of the 45th Annual Design Automation Conference, Anaheim, CA, USA, 9–13 June 2008; pp. 734–739.
12. Shahabinejad, N.; Beitollahi, H. Q-thermal: A Q-learning-based thermal-aware routing algorithm for 3-D network on-chips. *IEEE Trans. Components Packag. Manuf. Technol.* **2020**, *10*, 1482–1490. [CrossRef]
13. Lee, S.C.; Han, T.H. Q-function-based traffic-and thermal-aware adaptive routing for 3D network-on-chip. *Electronics* **2020**, *9*, 392. [CrossRef]

14. Momeni, M.; Shahhoseini, H. Energy optimization in 3D networks-on-chip through dynamic voltage scaling technique. In Proceedings of the 2020 28th Iranian Conference on Electrical Engineering (ICEE) IEEE, Tabriz, Iran, 4–6 August 2020; pp. 1–4.
15. Wang, H.; Peh, L.S.; Malik, S. Power-driven design of router microarchitectures in on-chip networks. In Proceedings of the 36th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO-36), San Diego, CA, USA, 3–5 December 2003; IEEE: Piscataway, NJ, USA, 2003; pp. 105–116.
16. Hardavellas, N.; Ferdman, M.; Falsafi, B.; Ailamaki, A. Reactive NUCA: Near-optimal block placement and replication in distributed caches. In Proceedings of the 36th Annual International Symposium on Computer Architecture, Austin, TX, USA, 20–24 June 2009; pp. 184–195.
17. Chen, G.; Li, F.; Son, S.W.; Kandemir, M. Application mapping for chip multiprocessors. In Proceedings of the 45th Annual Design Automation Conference, Anaheim, CA, USA, 9–13 June 2008; pp. 620–625.
18. Wolf, M.E.; Lam, M.S. A data locality optimizing algorithm. In Proceedings of the ACM SIGPLAN 1991 Conference on Programming Language Design and Implementation, Toronto, ON, Canada, 24–28 June 1991; pp. 30–44.
19. Bondhugula, U.; Baskaran, M.; Hartono, A.; Krishnamoorthy, S.; Ramanujam, J.; Rountev, A.; Sadayappan, P. Towards effective automatic parallelization for multicore systems. In Proceedings of the 2008 IEEE International Symposium on Parallel and Distributed Processing, Sydney, Australia, 14–18 April 2008; pp. 1–5.
20. Kim, C.; Burger, D.; Keckler, S.W. An adaptive, non-uniform cache structure for wire-delay dominated on-chip caches. In Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems, San Jose, CA, USA, 5–9 October 2002; pp. 211–222.
21. Chishti, Z.; Powell, M.D.; Vijaykumar, T. Distance associativity for high-performance energy-efficient non-uniform cache architectures. In Proceedings of the 36th Annual IEEE/ACM International Symposium on Microarchitecture, San Diego, CA, USA, 3–5 December 2003; MICRO-36; IEEE: Piscataway, NJ, USA, 2003; pp. 55–66.
22. Beckmann, B.M.; Wood, D.A. Managing wire delay in large chip-multiprocessor caches. In Proceedings of the 37th International Symposium on Microarchitecture (MICRO-37'04) IEEE, Portland, OR, USA, 4–8 December 2004; pp. 319–330.
23. Arora, A.; Harne, M.; Sultan, H.; Bagaria, A.; Sarangi, S.R. Fp-nuca: A fast noc layer for implementing large nuca caches. *IEEE Trans. Parallel Distrib. Syst.* **2014**, *26*, 2465–2478. [CrossRef]
24. Lira, J.; Molina, C.; Gonz, A. Hk-nuca: Boosting data searches in dynamic non-uniform cache architectures for chip multiprocessors. In Proceedings of the 2011 IEEE International Parallel & Distributed Processing Symposium IEEE, Anchorage, AK, USA, 16–20 May 2011; pp. 419–430.
25. Vanapalli, K.; Kapoor, H.K.; Das, S. An efficient searching mechanism for dynamic NUCA in chip multiprocessors. In Proceedings of the 2015 19th International Symposium on VLSI Design and Test IEEE, Ahmedabad, India, 26–29 June 2015; pp. 1–5.
26. Hu, J.; Marculescu, R. Energy-aware mapping for tile-based NoC architectures under performance constraints. In Proceedings of the 2003 Asia and South Pacific Design Automation Conference, Kitakyushu, Japan, 21–24 January 2003; pp. 233–239.
27. Hung, W.; Addo-Quaye, C.; Theocharides, T.; Xie, Y.; Vijakrishnan, N.; Irwin, M.J. Thermal-aware IP virtualization and placement for networks-on-chip architecture. In Proceedings of the IEEE International Conference on Computer Design: VLSI in Computers and Processors 2004 ICCD, San Jose, CA, USA, 11–13 October 2004; pp. 430–437.
28. Cong, J.; Wei, J.; Zhang, Y. A thermal-driven floorplanning algorithm for 3D ICs. In Proceedings of the IEEE/ACM International Conference on Computer Aided Design ICCAD-2004, San Jose, CA, USA, 7–11 November 2004; pp. 306–313.
29. Beigi, M.V.; Memik, G. TAPAS: Temperature-aware adaptive placement for 3D stacked hybrid caches. In Proceedings of the Second International Symposium on Memory Systems, Alexandria, VA, USA, 3–6 October 2016; pp. 415–426.
30. Jiang, X.; Lei, X.; Zeng, L.; Watanabe, T. High performance virtual channel based fully adaptive thermal-aware routing for 3D NoC. In Proceedings of the 2017 18th International Symposium on Quality Electronic Design (ISQED), Santa Clara, CA, USA, 14–15 March 2017; pp. 289–295.
31. Yao, K.; Ye, Y.; Pasricha, S.; Xu, J. Thermal-sensitive design and power optimization for a 3D torus-based optical NoC. In Proceedings of the 2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), Irvine, CA, USA, 13–16 November 2017; pp. 827–834.
32. Chou, C.T.; Lin, Y.P.; Chiang, K.Y.; Chen, K.C. Dynamic buffer allocation for thermal-aware 3D network-on-chip systems. In Proceedings of the 2017 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), Taipei, Taiwan, 12–14 June 2017; pp. 65–66.
33. Tsai, T.H.; Chen, Y.S. Thermal-aware real-time task scheduling for three-dimensional multicore chip. In Proceedings of the 27th Annual ACM Symposium on Applied Computing, Trento, Italy, 26–30 March 2012; pp. 1618–1624.
34. Li, J.; Qiu, M.; Niu, J.W.; Yang, L.T.; Zhu, Y.; Ming, Z. Thermal-aware task scheduling in 3D chip multiprocessor with real-time constrained workloads. *ACM Trans. Embed. Comput. Syst. (TECS)* **2013**, *12*, 1–22. [CrossRef]
35. Chaturvedi, V.; Singh, A.K.; Zhang, W.; Srikanthan, T. Thermal-aware task scheduling for peak temperature minimization under periodic constraint for 3D-MPSoCs. In Proceedings of the 2014 25nd IEEE International Symposium on Rapid System Prototyping, New Delhi, India, 16–17 October 2014; pp. 107–113.
36. Zhao, D.; Homayoun, H.; Veidenbaum, A.V. Temperature aware thread migration in 3D architecture with stacked DRAM. In Proceedings of the International Symposium on Quality Electronic Design (ISQED), Santa Clara, CA, USA, 4–6 March 2013; pp. 80–87.

37. Chaparro-Baquero, G.A.; Sha, S.; Homsi, S.; Wen, W.; Quan, G. Thermal-aware joint CPU and memory scheduling for hard real-time tasks on multicore 3D platforms. In Proceedings of the 2017 Eighth International Green and Sustainable Computing Conference (IGSC), Orlando, FL, USA, 23–25 October 2017; pp. 1–8.
38. Kim, D.H.; Athikulwongse, K.; Healy, M.B.; Hossain, M.M.; Jung, M.; Khorosh, I.; Kumar, G.; Lee, Y.J.; Lewis, D.L.; Lin, T.W.; et al. Design and analysis of 3D-MAPS (3D massively parallel processor with stacked memory). *IEEE Trans. Comput.* **2013**, *64*, 112–125. [CrossRef]
39. Wordeman, M.; Silberman, J.; Maier, G.; Scheuermann, M. A 3D system prototype of an eDRAM cache stacked over processor-like logic using through-silicon vias. In Proceedings of the 2012 IEEE International Solid-State Circuits Conference, San Francisco, CA, USA, 19–23 February 2012; pp. 186–187.
40. Dreslinski, R.G.; Fick, D.; Giridhar, B.; Kim, G.; Seo, S.; Fojtik, M.; Satpathy, S.; Lee, Y.; Kim, D.; Liu, N.; et al. Centip3de: A 64-core, 3d stacked near-threshold system. *IEEE Micro* **2013**, *33*, 8–16. [CrossRef]
41. Sahu, P.K.; Chattopadhyay, S. A survey on application mapping strategies for network-on-chip design. *J. Syst. Archit.* **2013**, *59*, 60–76. [CrossRef]
42. Bergstra, J.; Bardenet, R.; Bengio, Y.; Kégl, B. Algorithms for hyper-parameter optimization. *Adv. Neural Inf. Process. Syst.* **2011**, *24*.
43. Accellera Systems Initiative. SystemC, Version 2.3.3. Available online: https://github.com/accellera-official/systemc/releases/tag/2.3.3 (accessed on 4 September 2022).
44. Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; Koyama, M. Optuna: A next-generation hyperparameter optimization framework. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019; pp. 2623–2631.
45. Bienia, C.; Kumar, S.; Singh, J.P.; Li, K. The PARSEC benchmark suite: Characterization and architectural implications. In Proceedings of the 17th International Conference on Parallel Architectures and Compilation Techniques, Toronto, ON, Canada, 25–29 October 2008; pp. 72–81.