Power, Area and Thermal Prediction in 3D Network-on-Chip using Machine Learning

ABHIJITH C
Roll Number: 242CS003
ANAND M K
Roll Number: 242CS008

Department of Computer Science and Engineering
National Institute of Technology Karnataka (NITK)
Surathkal, India

Introduction

- Managing power, area, and thermal aspects in 3D Network-on-Chip (NoC) systems is a major challenge affecting system performance and reliability.
- Traditional methods often fail to capture the complex relationships between these factors.
- Machine learning offers a powerful solution by analyzing large datasets and identifying patterns to predict system behavior.
- This study aims to develop a machine learning-based framework to predict power, area, and thermal characteristics using key NoC parameters.

Literature Survey : Q-Thermal: A Q-Learning-Based Thermal-Aware Routing Algorithm for 3-D Network On-Chips

Authors: Narges Shahabinejad, Hakem Beitollahi (**2020**)
Approach: A Q-learning-based routing algorithm utilizing thermal information in a Q-table to manage routing decisions, optimizing thermal distribution by selecting paths with lower temperatures.
Performance Metrics: Standard deviation of thermal distribution, Average latency, Number of thermal hotspots.
Results:

- 28% and 13% improvement in thermal distribution standard deviation compared to TAAR and PTB3R.
- 32% improvement in average latency compared to existing methods.
- 38% reduction in thermal hotspots compared to TAAR and 54% compared to PTB3R.

Observations: Optimizes thermal distribution and reduces hotspots in 3-D NoCs.
Limitations: Slight increase in area and power consumption compared to

TTQR: A Traffic- and Thermal-Aware Q-Routing for 3D Network-on-Chip

Authors: Hanyan Liu, Xiaowen Chen, Yunping Zhao, Chen Li, Jianzhuang Lu (**2022**)

Approach: A Q-learning-based routing algorithm using two Q-tables: one for local traffic status and one for global thermal information.

Performance Metrics: Average latency, Throughput, Statistical Traffic Load Distribution, Temperature distribution.

Results:

- 63.6% improvement in latency compared to TAAR (Topology-aware Adaptive Routing).
- 41.4% improvement in throughput compared to TAAR.

Observations:

- Provides more uniform temperature distribution across layers than TAAR.
- More balanced traffic load distribution across layers.

Limitations: Slightly higher average temperature compared to TAAR.

A Nearest-Neighbor-Based Thermal Sensor Allocation and Temperature Reconstruction Method for 3-D NoC-Based Multicore Systems

Authors: Menghao Guo, Tong Cheng, Xinyi Li, Li Li, Yuxiang Fu (**2022**)
Approach: Uses a nearest-neighbor-based initialization algorithm to allocate thermal sensors, followed by a Genetic Algorithm (GA) for optimization. Artificial neural networks (ANN) are employed for estimating the temperature of non-sensor-allocated nodes.
Performance Metrics: Average temperature error, Maximum temperature error.
Results:

- Average temperature error reduced by 17.60%–88.63%.
- Maximum temperature error reduced by 26.97%–85.92%.

Observations:

- Effectively allocates thermal sensors based on spatial thermal correlation.
- Uses ANN to estimate temperatures in non-sensor-allocated nodes.

Limitations: Assumes spatial thermal correlations remain constant across applications, which may affect temperature reconstruction accuracy in some cases.

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

Authors: Xinyi Li, Wenjie Fan, Heng Zhang, Jinlun Ji, Tong Cheng, Shiping Li, Li Li, Yuxiang Fu (**2024**)
Approach: A neural network-based mapping technique that optimizes temperature distribution by strategically mapping NN layers based on their computational loads.
Performance Metrics: Average Temperature, Temperature Variance, Maximum Temperature, Average Packet Latency.
Results: Average Temperature: 68.3°C, Maximum Temperature: 77.2°C, Temperature Variance: 5.9°C², Packet Latency: 13.6 cycles.
Observations:

- Reduces average temperature, temperature variance, and maximum temperature.
- Results in a more uniform temperature distribution across the NoC, improving thermal management.

Limitations:

- Primarily focused on offline inference scenarios, with limited consideration for dynamic or runtime scenarios.

NoCeption: A Fast PPA Prediction Framework for Network-on-Chips Using Graph Neural Network

Authors: Fuping Li, Ying Wang, Cheng Liu, Huawei Li, Xiaowei Li (**2022**)
Approach: A Graph Neural Network framework designed to predict power, performance, and area (PPA) of Network-on-Chips (NoCs).
Performance Metrics: Power prediction accuracy, Area prediction accuracy.
Results:

- Power prediction accuracy = 97.36%.
- Area prediction accuracy = 97.83%.

Observations:

- The framework offers fast and accurate PPA predictions, which aid in NoC design optimization.

Limitations:

- Performance and efficiency may degrade for larger systems, as the experiments are effective only up to a certain number of cores.

LSTM-based Temperature Prediction and Hotspot Tracking for Thermal-aware 3D NoC System

- **Authors:** Tong Cheng, Haoyu Du, Li Li, Yuxiang Fu (2021)
- **Approach:** A Long Short-Term Memory (LSTM)-based model for temperature prediction that can work alongside Proactive Dynamic Thermal Management (PDTM).
- **Metrics Used:** Mean-Square-Error (MSE), Average Prediction Accuracy
- **Results:**
  - Mean-Square-Error = 0.411°C
  - Average Prediction Accuracy improved by 41.92% to 73.63%
- **Observations:** The proposed method improves temperature prediction accuracy compared to traditional ARMA (Autoregressive Moving Average) models. Additionally, the model can quickly locate new hotspots within 0.075 ms.
- **Limitation:** The study is conducted on an 8×8×4 3D NoC system, but it is unclear how well the model scales to larger systems.

Adaptive Machine Learning-Based Proactive Thermal Management for NoC Systems

Authors: Chen et al. **Year:** 2023

Approach: An Adaptive Single-Layer Perceptron (ASLP) is utilized for predicting temperature, while adaptive reinforcement learning dynamically adjusts the throttling ratio. A revised thermal-traffic co-simulator and MCSL benchmark were used to evaluate the proposed model in simulated traffic patterns within an 8x8 NoC system.

Results:

- Saturation throughput increased by up to 43%.
- Average temperature errors reduced by up to 78%, and maximum errors by up to 74%.
- Improved throughput and reduced latency under varying PIRs.

Limitations:

- Limited to the XY routing algorithm.
- Relies on accurate temperature readings from physical sensors or estimations.

Deep Reinforcement Learning for Self-Configurable NoC

Author: Reza et al. **Year:** 2020

Approach: Address power management in NoCs using reinforcement learning (RL), which configures network resources dynamically based on application needs and system utilization. Specifically, RL is applied to adjust NoC voltage levels, while a neural network (NN) is implemented to identify NoC performance patterns. A concentrated mesh NoC architecture is utilized to reduce the overhead of machine learning techniques.

Results:

- Throughput improved by 15% and 10% using COSMIC and E3S benchmarks, respectively.
- EDP improved by 45% and 110% using the same benchmarks.

Strength: Compared to non-ML solutions, it achieves significant improvements in throughput and EDP.

Limitation: Concentrated mesh NoC considered to reduce ML overheads.

Machine Learning Enabled Power-Aware Network-on-Chip Design

Authors: Dominic et al.
Year: 2017

Approach: Reduce both static and dynamic power consumption through power-gating techniques. The prediction of link utilization and traffic loads is implemented using a decision tree. The dataset used for training the predictor consists of historical data on link utilization, buffer utilization, and packet type.

Results:

- Dynamic power: 12.3 mW; static power: 16.6 mW.
- Latency improved by 14%; power savings: 31.7%–85.6%.
- Area reduced by 62.3%.
- Decision tree accuracy: +13.2% (traffic) and +13.8% (link utilization).

Limitations: Only the XY routing algorithm was used.

Machine Learning Enabled Solutions for Design and Optimization Challenges in NoCs

Authors: Farhadur et al.
Year: 2023

Approach: Address power issues in NoCs through the application of ML to configure the network. The input dataset includes node and link usage, computation and communication demands, thermal and power consumption, task deadline requirements, and execution time. Various design and optimization challenges are discussed in the study.

Results:

- Throughput and latency improved by up to 30%, with an average improvement of 15%.
- Energy consumption reduced by 6%.

Strength: Superior performance compared to traditional ML methods.

Limitations: Only the XY routing algorithm was used.

Adaptive Machine Learning-Based Temperature Prediction Scheme for Thermal-Aware NoC System

Authors: Kun-Chih Jimmy et al.
Year: 2020

Approach: Predictive Dynamic Thermal Management (PDTM) technique, which proactively manages node temperatures based on predicted thermal information. An artificial neural network (ANN) is employed for temperature prediction.

Results:

- Average temperature error reduced by 37.2% to 62.3%.
- System throughput improved by 9.16%.
- Area overhead reduced by 18.59% to 22.11%.

Strength: The model adapts dynamically to temperature behavior, improving overall system performance.

Limitations: Only the XY routing algorithm was used.

Predictive Thermal Management for Energy-Efficient Execution of Concurrent Applications on Heterogeneous Multicores

Authors: Eduardo Weber et al. **Year:** 2019

Approach: Runtime manager incorporating a temperature predictor. Various regression models were trained using this data, and the model with the best performance was integrated into the runtime manager.

Results:

- MAE: 1.13°C; max error: 16.91°C; std. dev.: 1.31; AIC: 33222.
- 10% improvement in energy and performance; doubled thermal cycling.

Strength: The predictor gives better error averages while maintaining a reasonable standard deviation.

Limitations: Increased thermal cycling and overhead for temperature prediction.

Challenges from Literature Survey

- **Availability of Datasets**: Most ML and DL models require a large volume of quality datasets for training. However, the availability of such datasets is limited in the area of PAT prediction for NoC.
- **Integrated Power, Thermal, and Area Prediction Models**: Most existing studies independently focus on power or thermal optimization. Studies on simultaneous analysis of power, area, and thermal are rare.
- **Scalability**: Many studies using ML or DL models have been tested in specific architectures or smaller systems, but their scalability to larger multi-core environments is unclear.
- **Limited Routing Algorithms Considered:** Only a few routing algorithms, such as Q-learning-based approaches, have been considered for optimizing thermal and power management in NoC systems.
- **Diversity of ML/DL Algorithms:** Few algorithms, such as reinforcement learning and neural networks, have been explored for PAT prediction in NoC, while CNN and regression models remain unexplored.

Problem Statement

- **To develop an integrated machine learning framework for predicting power, area, and thermal characteristics of 3D Network-on-Chip (NoC) systems, addressing scalability and adaptability to different routing algorithms, using diverse ML/DL models.**

Objectives

- To design a machine learning framework for predicting power, area, and thermal characteristics in 3D NoC systems.
- To explore and apply diverse ML/DL models for accurate predictions.
- To address scalability to larger NoC systems.
- To ensure adaptability of the framework to different routing algorithms in NoC designs.

Proposed Approach: Dataset Creation

Dataset Generation Process:

- Use the configurations in Table 1 to simulate a variety of NoC system setups.
- Simulate the configurations with the **PAT-Noxim simulator** to generate PAT data (Power, Area, and Thermal information).

NoC Parameter	**Parameter Values**
Topology	3D Mesh Topology
Routing Algorithm	XYZ Routing, OE_3D, Full Adaptive Routing
Network Size	dimX: 2 to 16, dimY: 2 to 16, dimZ: 2
Traffic Pattern	Random
Buffer Size	4, 6, 8, 10
Packet Size	4
Packet Injection Rate (PIR)	0.01 to 0.1
Sample Period	200000

Table: NoC Parameters and Values

Proposed Approach Overview

Objective: To predict power, area, and thermal (PAT) metrics in NoC systems using **AdaBoost** with **Decision Tree Regressors**.

- The approach combines AdaBoost, an ensemble learning method, with Decision Trees as base learners to predict NoC system metrics.
- AdaBoost improves the prediction accuracy by focusing on hard-to-predict instances, refining model performance over multiple iterations.
- Decision Trees are chosen as base learners due to their simplicity and ability to model non-linear relationships effectively.
- Separate models are trained for each PAT metric (Power, Area, Thermal).

AdaBoost with Decision Tree Regressor

- *AdaBoost* (Adaptive Boosting) is an ensemble technique that combines weak learners, such as *Decision Trees*, to improve predictive accuracy.
- It works by assigning more weight to misclassified data points, so subsequent learners focus on these harder examples.
- The final model is a weighted combination of the weak learners, enhancing overall model performance.
- *Decision Tree Regressors* are non-linear models that split data based on feature values to predict a target.
- They are effective for modeling complex relationships and capturing non-linear interactions between NoC parameters.
- In AdaBoost, Decision Trees are used as base learners, improving the model's performance through ensemble learning.

AdaBoost with Decision Tree Regressor Algorithm

Algorithm 1 AdaBoost with Decision Tree Regressor

```
**Initialize:** Set initial sample weights equally, set $T$ as the total number of iterations (Number of Decision Trees).
**for** each iteration $t = 1, 2, ..., T$ **do**
    **Train Model:** Train a Decision Tree Regressor on the weighted data.
    **Calculate Error:** Compute the weighted error rate as the difference between actual and predicted values.
    **for** each sample **do**
        Increase the weight for samples with higher prediction error.
        Decrease the weight for samples with lower prediction error.
    **end for**
    **Calculate Model Weight:** Assign a weight to the trained model based on its error rate.
**end for**
**Final Prediction:** Combine the predictions of all models, weighted by their performance, to obtain the final prediction.
```