Oregon State University

COLLEGE OF ENGINEERING | School of Electrical Engineering and Computer Science

Enabling Large Architectural Design Space Exploration Using Machine Learning

Dr. Lizhong Chen

System Technology and Architecture Research (STAR) Lab

Oregon State University

October 5, 2021

S.T.A.R Oregon State

Introduction

Machine learning surpasses humans

- Humans played Go > 2000 years
- AlphaGo surpass humans after weeks

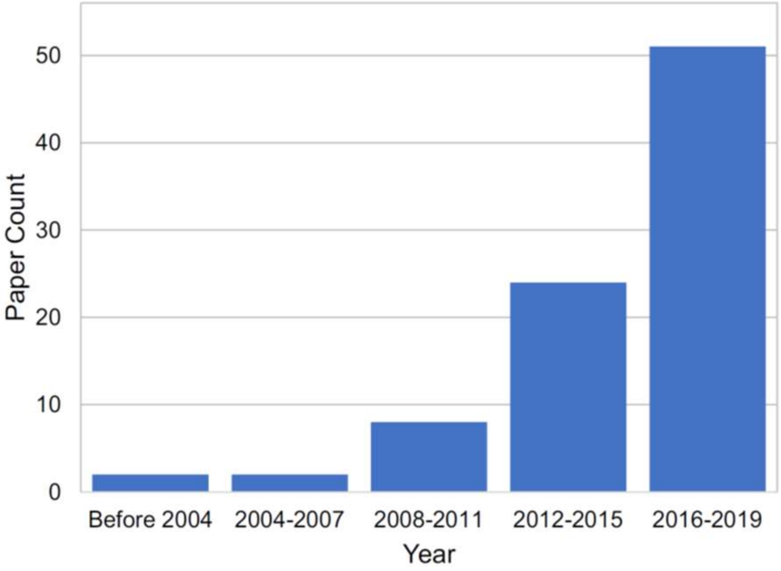
Can machine learning surpass human in arch design?

- Computer architecture < 80 years
- Humans are limited by processing, memory, frequency
- AI for architecture!

This talk: utilize AI for architectural design space exploration

Introduction

Publications on AI applied to architecture



Drew Penney, and Lizhong Chen, "*A Survey of Machine Learning Applied to Computer Architecture Design*", arXiv 1909.12373, September 2019.

Agenda

Introduction

Challenges for Architectural Exploration

Example Solutions

- Routerless network-on-chip design in CPUs
- Memory controller placement in GPUs
- Resource Allocation in Edge Servers

Conclusion & Acknowledgment

AI for Architectural Design Space Exploration

Challenges for architectural design exploration

- How to guide search in the vast design space that even exceeds the game of Go
- How to deal with the lack of training data
- How to evaluate design points rapidly without incurring cumbersome full system simulations each time

=> Need careful selection of AI/ML approaches!

Agenda

Introduction

Challenges for Architectural Exploration

Example Solutions

- Routerless network-on-chip design in CPUs
- Memory controller placement in GPUs
- Resource Allocation in Edge Servers

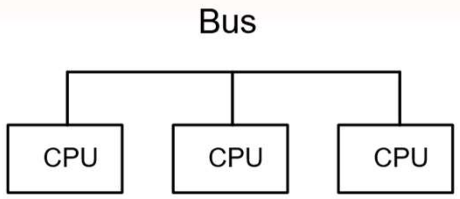
Conclusion & Acknowledgment

S.T.A.R
Oregon State

Background: Network-on-Chip (NoC)

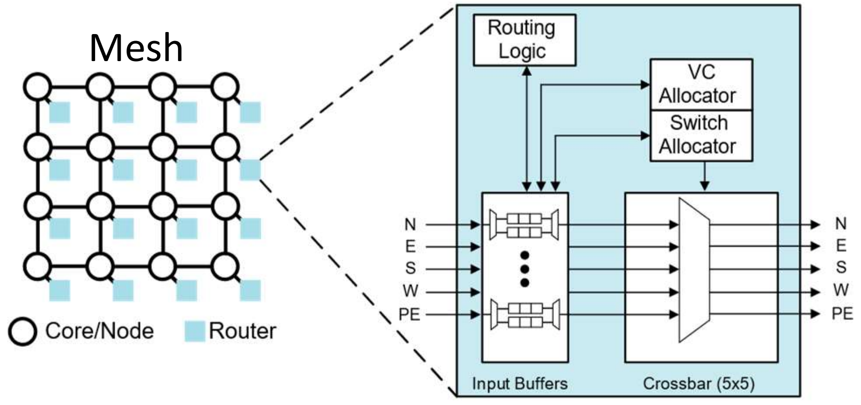
Bus architecture

- Simple & low cost
- Limited bandwidth & scaling



Router-based NoC

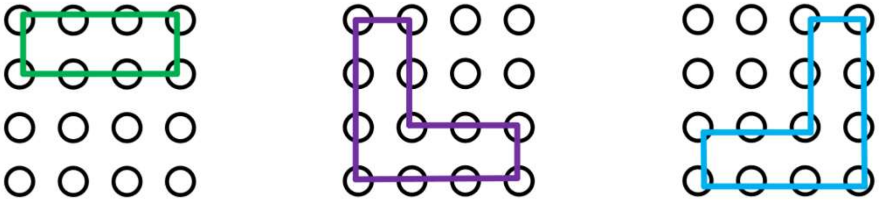
- High bandwidth & scaling
- Router > 80% of NoC power
- Router pipeline latency



How can we remove router overhead?

Routerless NoCs

- Connect cores using loops (bundles of wires)
- Can overlap loops to build interconnect



- Packets do not switch loops => no routers needed!
- Every two cores connected by at least one loop

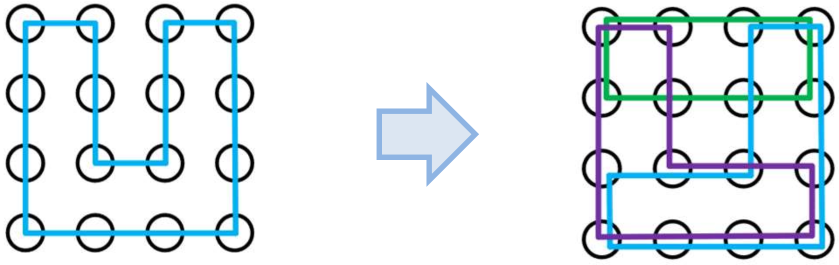
How to find a good set of loops?

F. Alazemi, A. Azizimazreah, B. Bose, and L. Chen, "*Routerless Networks-on-Chip*", in the IEEE International Symposium on High-Performance Computer Architecture (HPCA), 2018.

Vast Design Space of Routerless NoCs

Routerless NoC Design

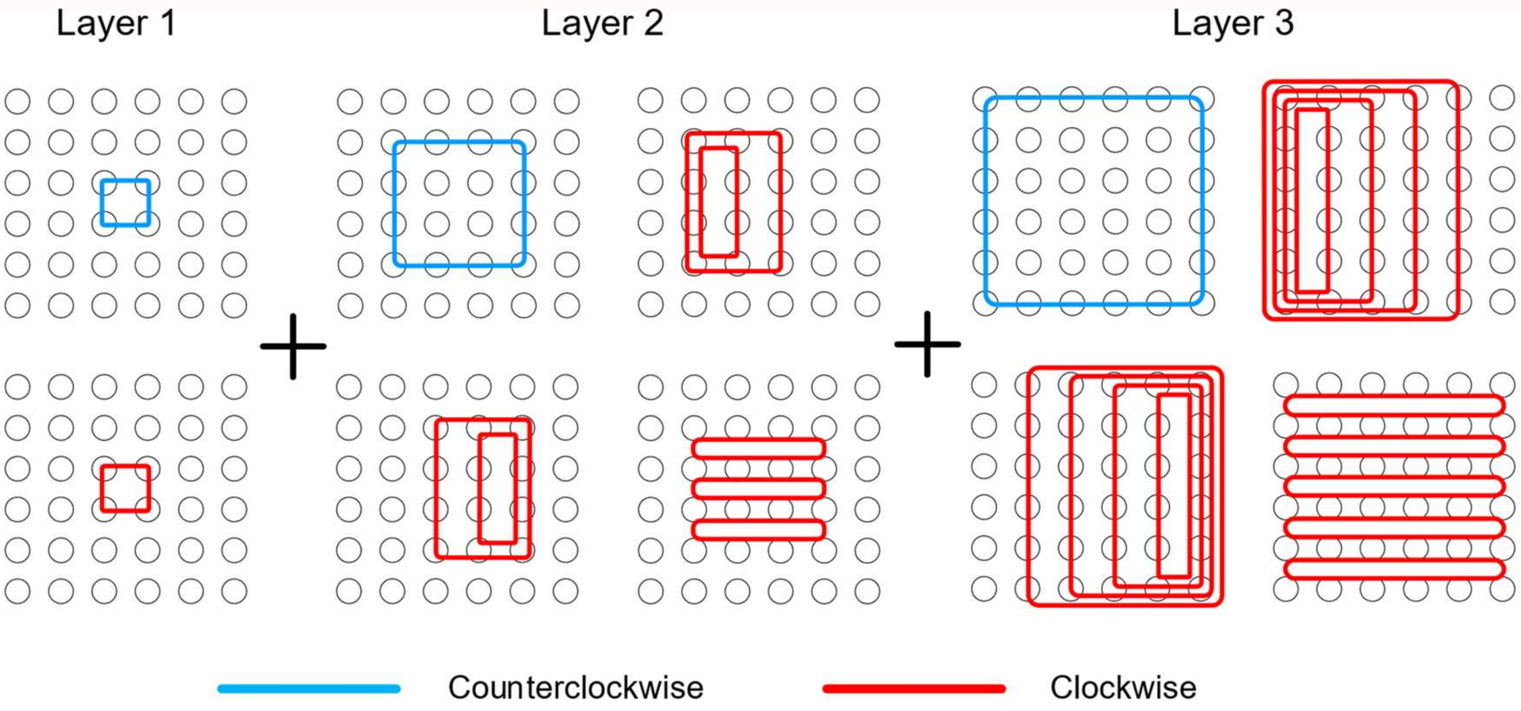
- Single long loop (Hamiltonian cycle)
  - High hop count, not scalable
- Multiple loops
  - Optimize NoC performance through efficient loop placement
  - Immense design space
  - Consider wiring constraints



N	# of Loops
2	2
4	426
6	> 2 Million
8	> 1 Billion

Larger than Go!

State-of-the-art Routerless NoC Design

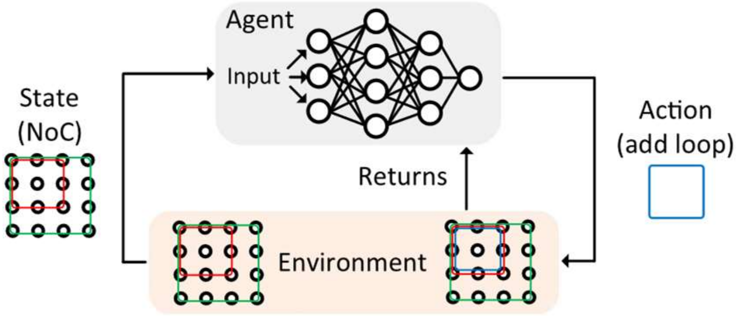


- L. Chen, F. Alazemi, B. Bose, US Patent #10657216B2

Proposed AI-based Framework

To avoid the need of training data

=> Deep reinforcement learning



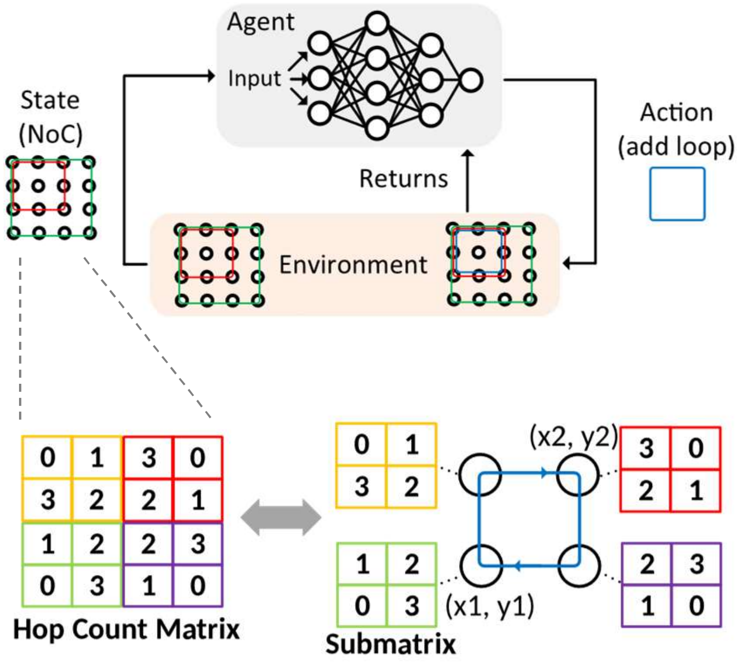
Proposed AI-based Framework

To avoid the need of training data

=> Deep reinforcement learning

To circumvent full system simulation

=> Use NoC metric (hop count) to approximate system performance



Hop Count Matrix

0	1	3	0
3	2	2	1
1	2	2	3
0	3	1	0

Submatrix

0	1
3	2

3	0
2	1

1	2
0	3

2	3
1	0

Proposed AI-based Framework

To avoid the need of training data

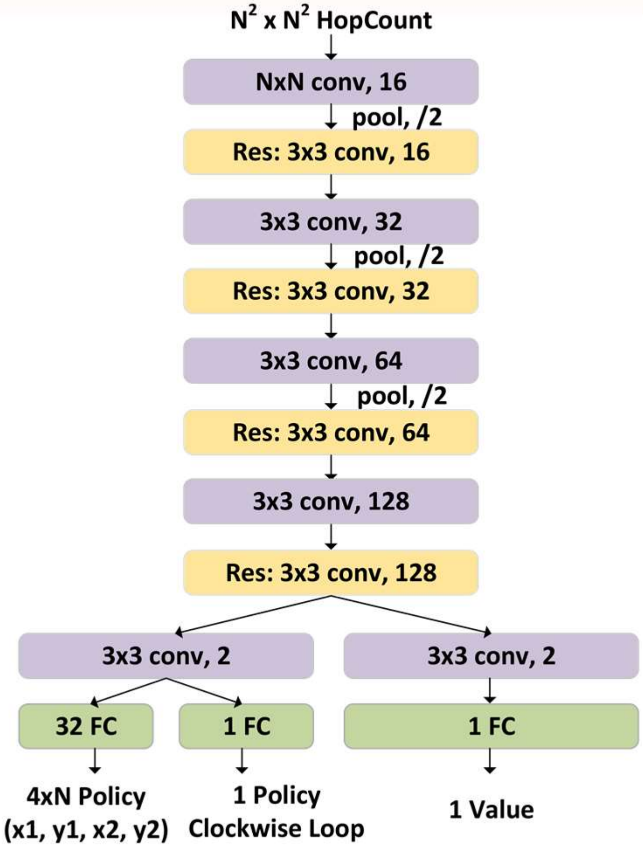
=> Deep reinforcement learning

To circumvent full system simulation

=> Use NoC metric (hop count) to approximate system performance

To handle immerse design space

=> Residual neural network with convolutional layers



Proposed AI-based Framework

To avoid the need of training data

=> Deep reinforcement learning

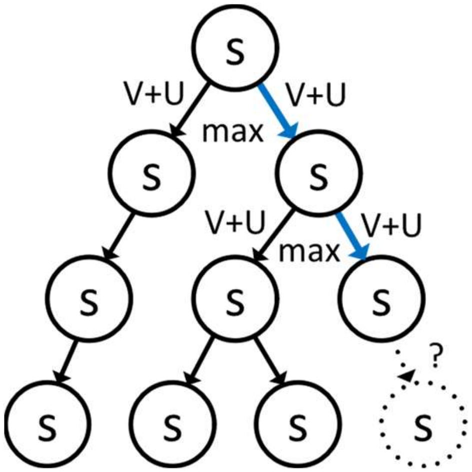
To circumvent full system simulation

=> Use NoC metric (hop count) to approximate system performance

To handle immerse design space

=> Residual neural network with convolutional layers

=> Monte Carlo search tree to organize and re-use prior results



Proposed AI-based Framework

To avoid the need of training data

=> Deep reinforcement learning

To circumvent full system simulation

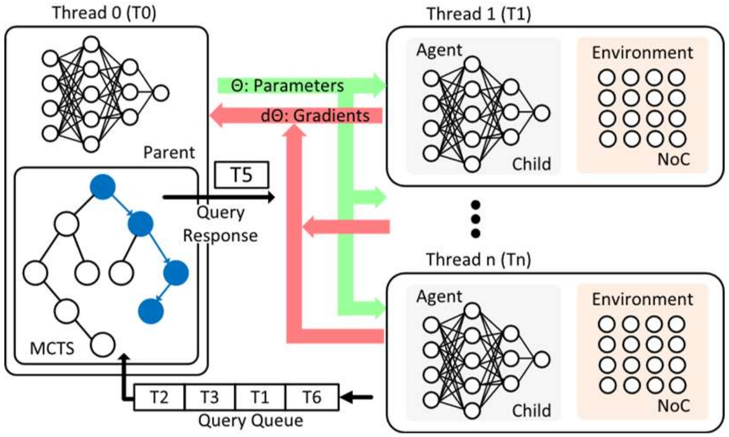
=> Use NoC metric (hop count) to approximate system performance

To handle immerse design space

=> Residual neural network with convolutional layers

=> Monte Carlo search tree to organize and re-use prior results

=> Multi-threaded execution



Results/Effectiveness

Speed of the framework

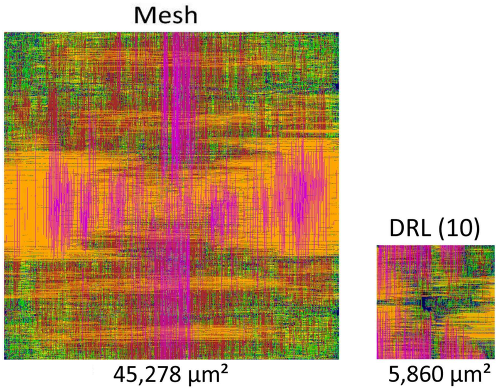
- Traditional methods (SA/GA) are extremely slow beyond 8x8
- Proposed (**DRL**): generate high-quality 20x20 routerless in hours

Performance improvement (vs. mesh)

- 4.12X increase in NoC throughput
- 60% reduction in NoC latency

Area comparison

- 7.7x reduction in area



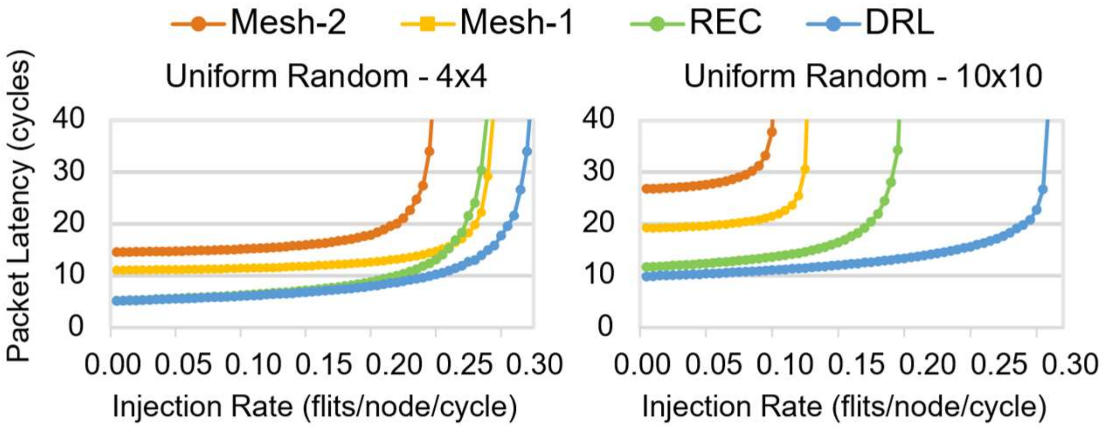
T.R. Lin, D. Penney, M. Pedram, and L. Chen, "*A Deep Reinforcement Learning Framework for Architectural Exploration: A Routerless NoC Case Study*", in HPCA 2020 (Best Paper Runner-up Award).

Scalability

Near-ideal throughput scaling with NoC size

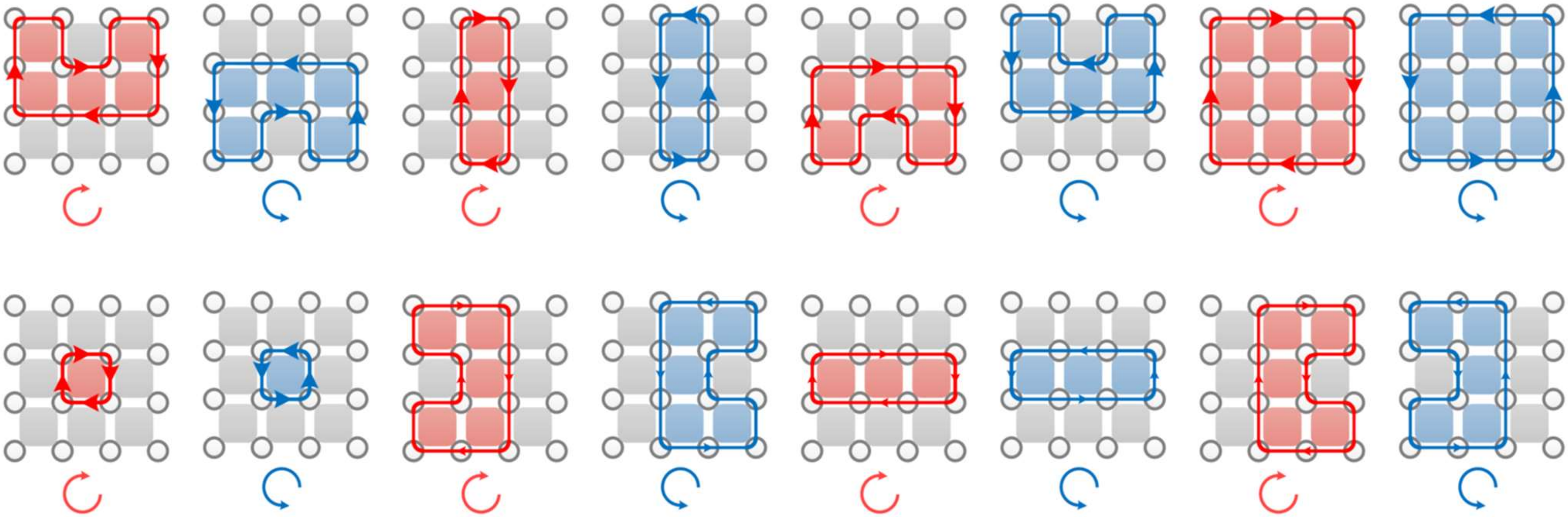
From 4x4 to 10x10:

- REC throughput decreases by 32%
- DRL throughput decreases by 4%



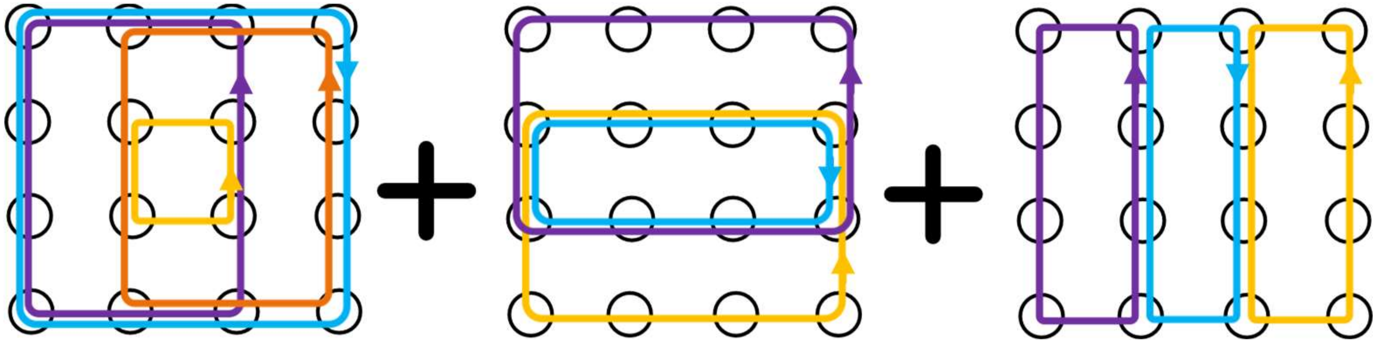
Design Comparison: GA vs. DRL (4x4)

Genetic Algorithm: 16 loops, highly irregular



Design Comparison: GA vs. DRL (4x4)

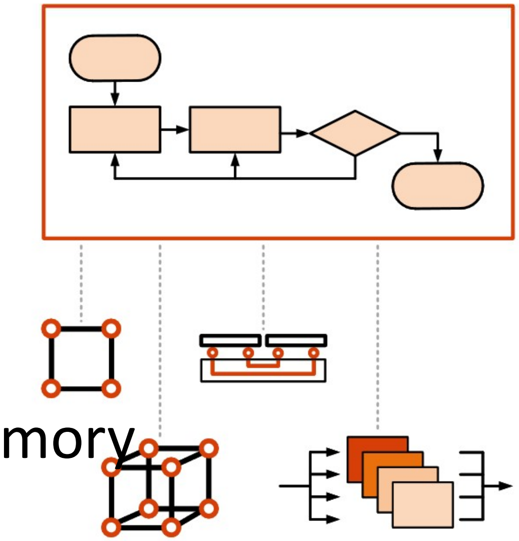
DRL: 10 loops, regular, symmetric



Framework – Broad Applicability

Deep RL framework allows generic implementation & search

- Routerless NoC design
- 3-D NoC design
  - Explore novel configurations
- Interposers/chiplets
  - Improve wiring efficiency & throughput
- Accelerators
  - Explore connectivity between PEs and memory



Agenda

Introduction

Challenges for Architectural Exploration

Example Solutions

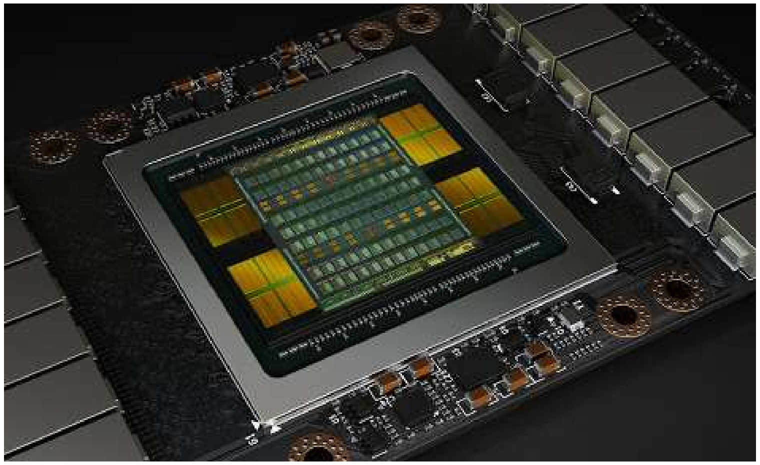
- Routerless network-on-chip design in CPUs
- Memory controller placement in GPUs
- Resource Allocation in Edge Servers

Conclusion & Acknowledgment

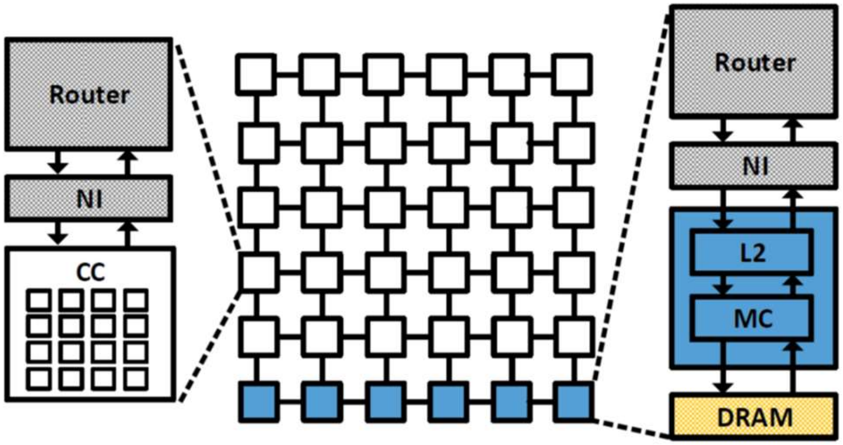
S.T.A.R Oregon State

Memory Controller Placement in GPUs

- GPU workloads are data-intensive
- All memory traffic is through **memory controllers (MCs)**
- Locations of MCs are critical to performance



Nvidia V100



Finding Best MC Placement

Large but still manageable design space

- Can use genetic algorithm (GA) to search

Evaluating design points

- Need more accurate metric than simple distance

T.R. Lin, Y. Li, M. Pedram, and L. Chen, "*Design Space Exploration of Memory Controller Placement in Throughput Processors with Deep Learning*", in the IEEE Computer Architecture Letters (CAL), 2019.

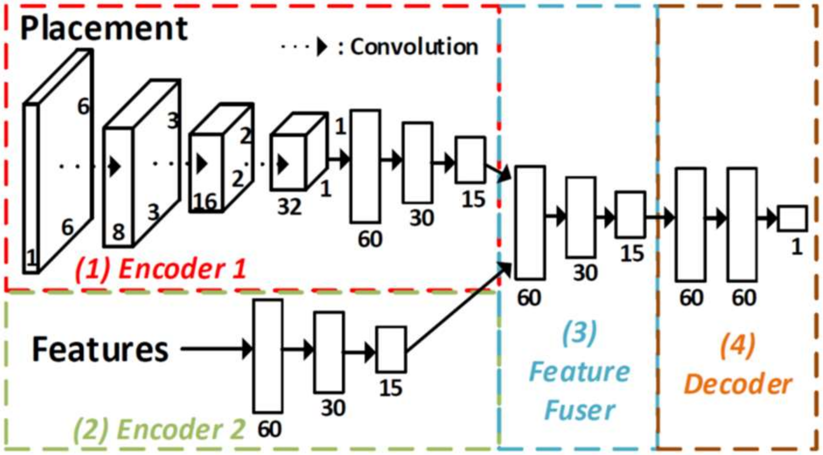
Finding Best MC Placement

Large but still manageable design space

- Can use genetic algorithm (GA) to search

Evaluating design points

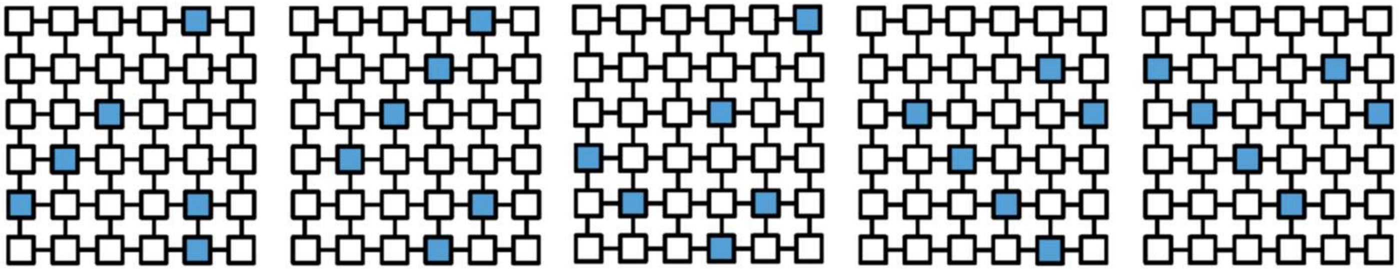
- Need more accurate metric than simple distance
- Use DNN to predict IPC



T.R. Lin, Y. Li, M. Pedram, and L. Chen, "*Design Space Exploration of Memory Controller Placement in Throughput Processors with Deep Learning*", in the IEEE Computer Architecture Letters (CAL), 2019.

Effectiveness

- Train DNN using only 0.53% of all design points (on 6x6)
- GA+DNN is 282X faster than traditional method
- 36.4% higher IPC than edge placement
- Provide design insights: sparse, diagonal, parallel



Top 5 best MC placements found by GA+DNN

Agenda

Introduction

Challenges for Architectural Exploration

Example Solutions

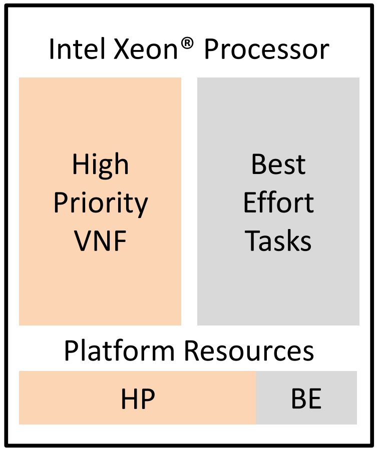
- Routerless network-on-chip design in CPUs
- Memory controller placement in GPUs
- Resource Allocation in Edge Servers

Conclusion & Acknowledgment

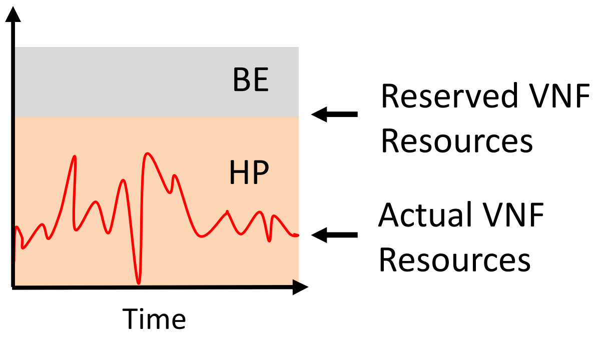
S.T.A.R Oregon State

Resource Allocation in Edge Servers

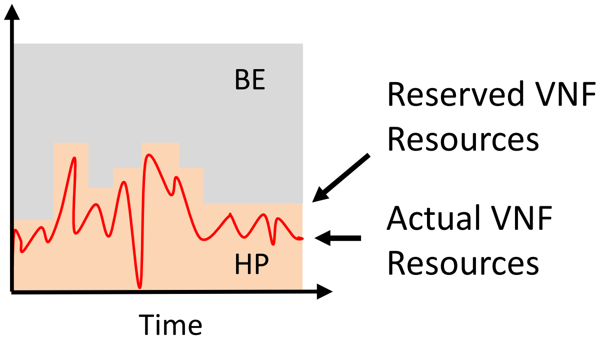
- Concurrent high priority (HP) & best effort (BE) workloads
- Improve resource utilization without increasing QoS violations
- Need to explore allocation space
- Changing workloads and system => online ML approaches



Static Resource Allocation

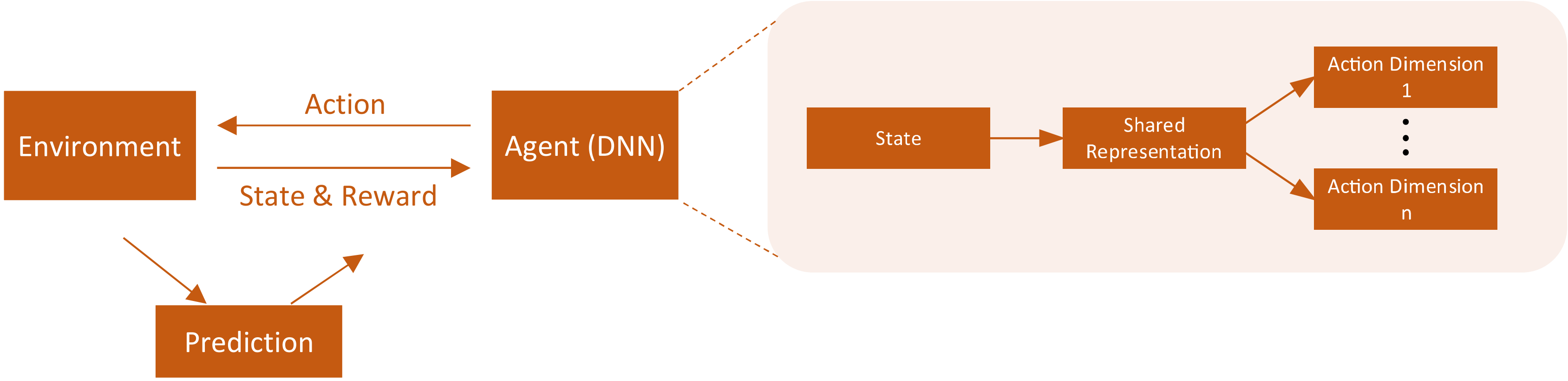


Ideal Resource Allocation



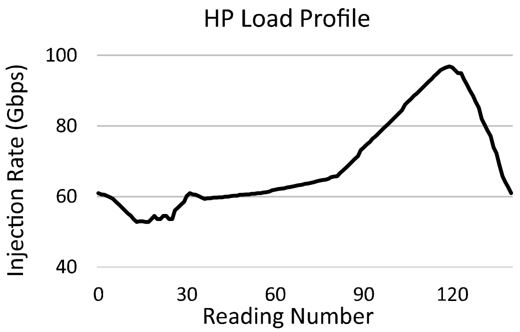
Dynamic Resource Allocation

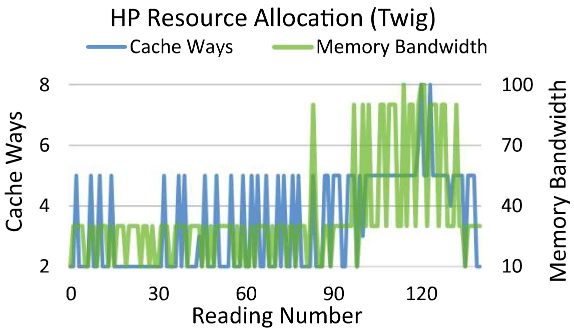
- Iterative adjustment via deep reinforcement learning (DRL)
- Use action-branching architecture to reduce action space
- Proactive: state/reward is predicted, instead of direct measurement
- Execution overhead: 0.15%

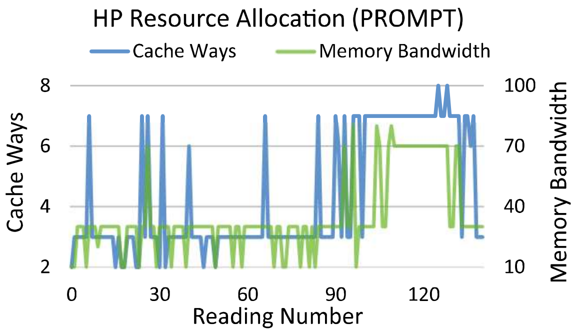


Improvement

- 2.8x reduction in QoS violations for HP
- 2.2x reduction in severity of each QoS violation, on average
- 30% improvement in BE performance
- Significantly reduced oscillation compared with state-of-the-art


HP Load Profile


HP Resource Allocation (Twig)


HP Resource Allocation (PROMPT)

D. Penney, B. Li, J. Sydir, C. Tai, E. Walsh, T. Long, S. Lee, L. Chen, "*PROMPT: Learning Dynamic Resource Allocation Policies for Edge-Network Applications*", arXiv, 2021.

Agenda

Introduction

Challenges for Architectural Exploration

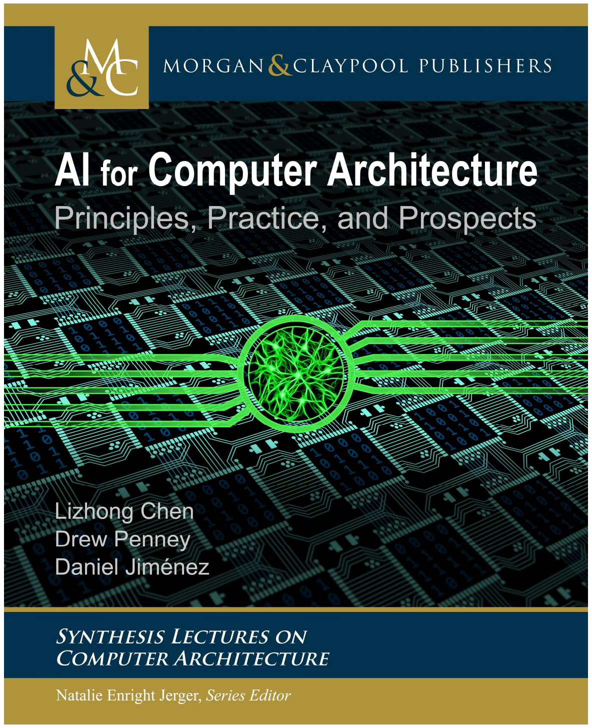
Example Solutions

- Routerless network-on-chip design in CPUs
- Memory controller placement in GPUs
- Resource Allocation in Edge Servers

Conclusion & Acknowledgment

“AI for Computer Architecture”

- **“AI for Computer Architecture: Principles, Practice, and Prospects”**
  - w/ Drew Penny and Daniel Jimenez
- Recently published in “*Synthesis Lectures on Computer Architecture*” series



MORGAN & CLAYPOOL PUBLISHERS

AI for Computer Architecture

Principles, Practice, and Prospects

Lizhong Chen
Drew Penney
Daniel Jiménez

SYNTHESIS LECTURES ON COMPUTER ARCHITECTURE

Natalie Enright Jerger, *Series Editor*

Main Contributors / Acknowledgment

Lizhong Chen

Fawaz Alazemi

Drew Penney

Yongbin Gu

Ting-Ru Lin

Aashish Adhikari

Ryan Gambord

Arash Azizi

Yunfan Li

Conclusion

AI/ML offers a potentially transformative approach for architecture design

Examples

- Deep reinforcement learning for Routerless NoCs
- DNN for MC placement in GPUs
- Resource allocation in edge servers

Call for research on innovative use of AI/ML in architecture!

S.T.A.R Oregon State

Thank You!