BoDNoC: Providing Bandwidth-on-Demand Interconnection for Multi-Granularity Memory Systems

Shiqi Lian, Ying Wang, Yinhe Han and Xiaowei Li

State Key Laboratory of Computer Architecture
Institute of Computing Technology, Chinese Academy of Science

ASP-DAC'2017, Chiba/Tokyo, Japan
19 January 2017

计算机体系结构国家重点实验室
State Key Laboratory of Computer Architecture, ICT, CAS

中国科学院计算技术研究所
INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES

Outline

- Introduction
- Previous Work
- Bandwidth-on-Demand NoC
- Experimental Results
- Conclusions

Introduction

Multi-Granularity Memory System

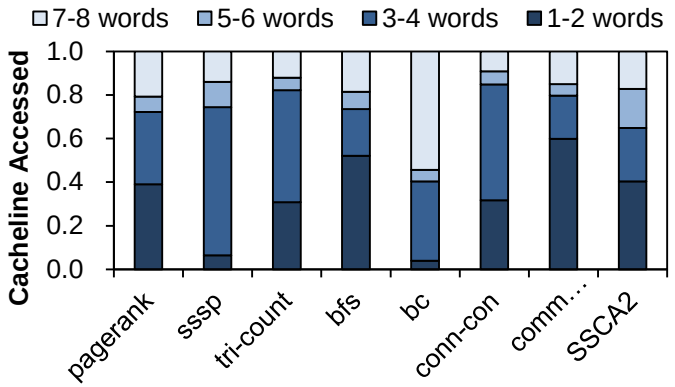
- Various spatial locality in graph analysis applications
  - >70% cache lines just access less than half line
- Multi-Granularity memory systems in different hierarchies
  - **Cache**:

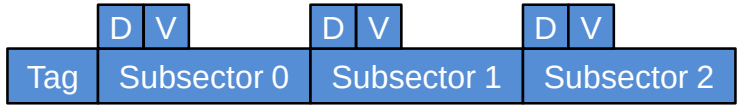
    Sector cache [*IBM System Journal*],

    Amoeba-Cache [*MICRO'45*]
  - **DRAM**:

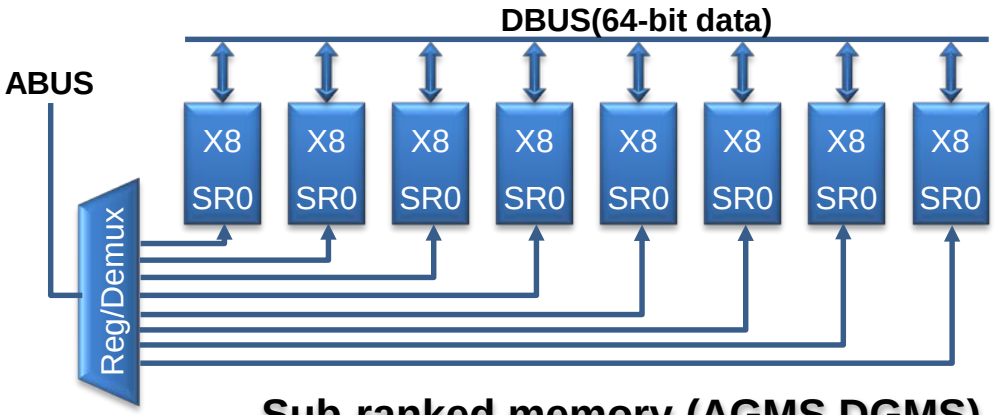
    AGMS [*ISCA'11*],

    DGMS [*ISCA'12*]





Sector cache

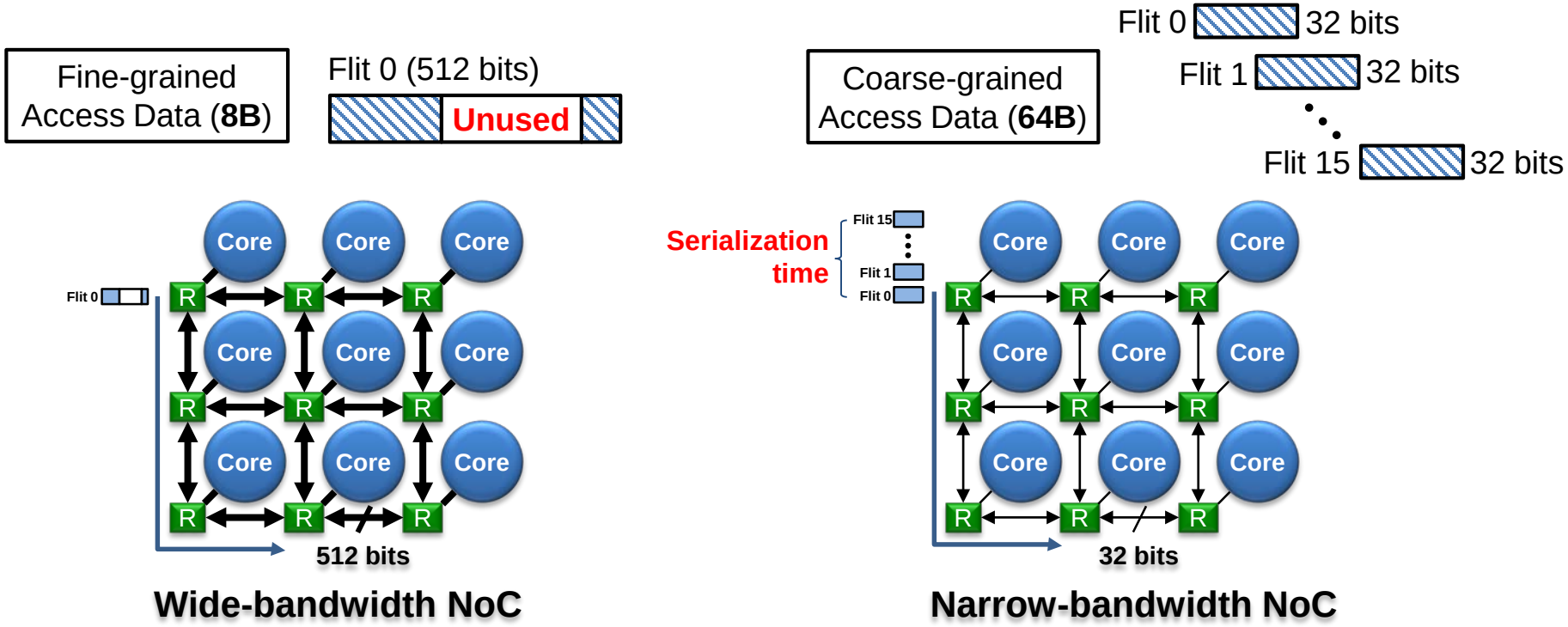


Sub-ranked memory (AGMS,DGMS)

Introduction

On-size-bandwidth NoC

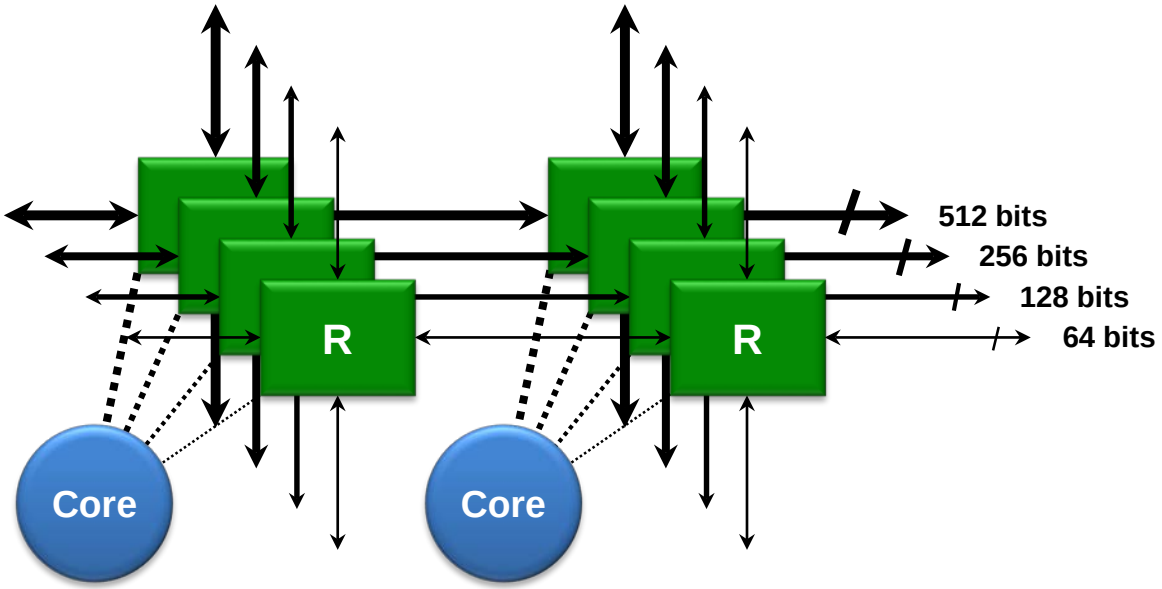
- Wide bandwidth
  - Fine-grained access data cannot fill a flit
- Narrow bandwidth
  - Lead long serialization time for coarse-grained access data



Previous Method

Heterogeneous Multi-NoC designs

- Composed of multiple heterogeneous-bandwidth subnets (64 bits, 128 bits, 256 bits, 512 bits)
- Each subnet is applied to particular granularity access

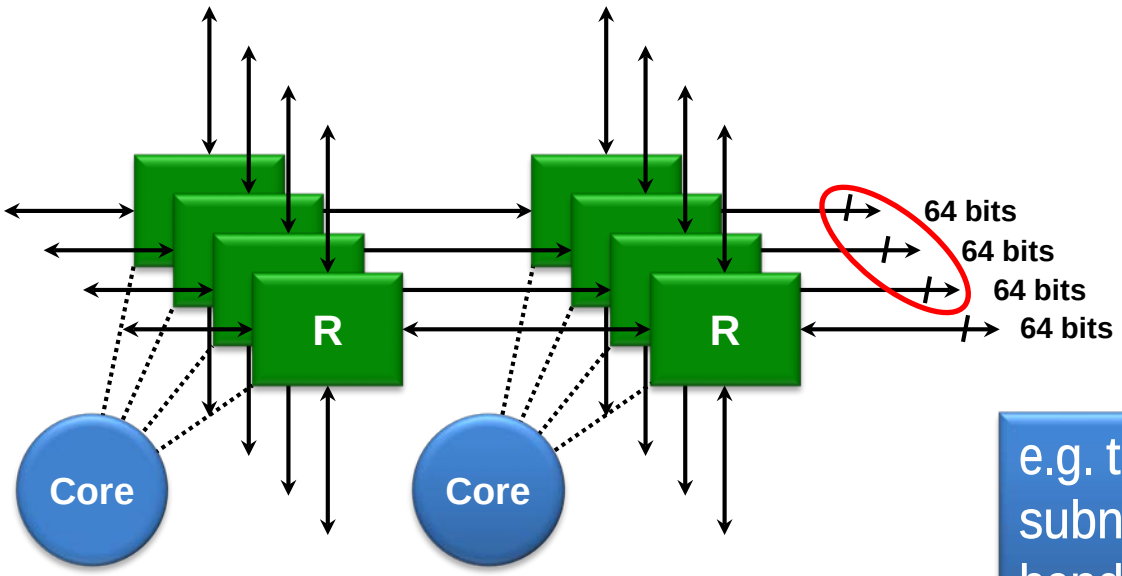


A. K. Mishra, O. Mutlu and C. R. Das, “A heterogeneous multiple network-on-chip design: An application-aware approach,” In *DAC*, 2013.

Our Method: Bandwidth-on-Demand NoC

Main Proposal

- Composed of multiple homogeneous-bandwidth subnets
- Merge multiple subnets to provide special bandwidth
  - one coarse-grained data block can be transmitted through multiple subnets **simultaneously**

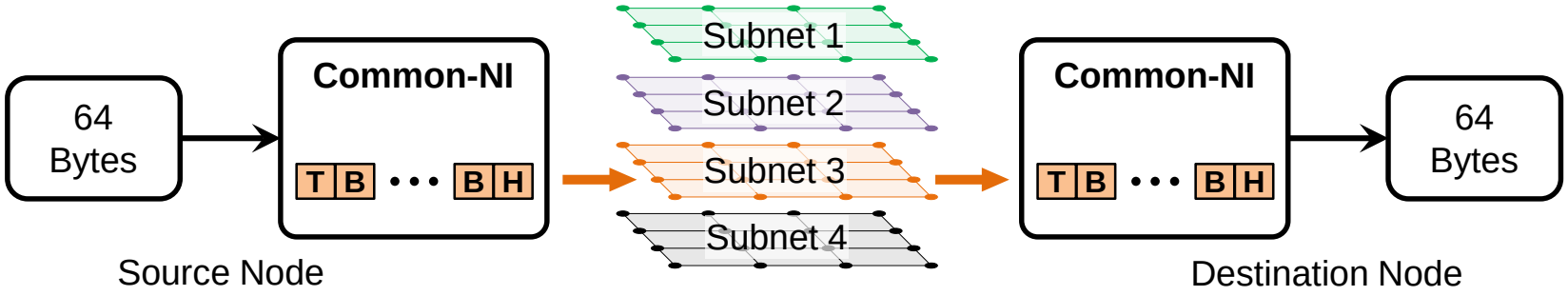


e.g. three narrow bandwidth (64b) subnets can be merged as a wider bandwidth (192b) network

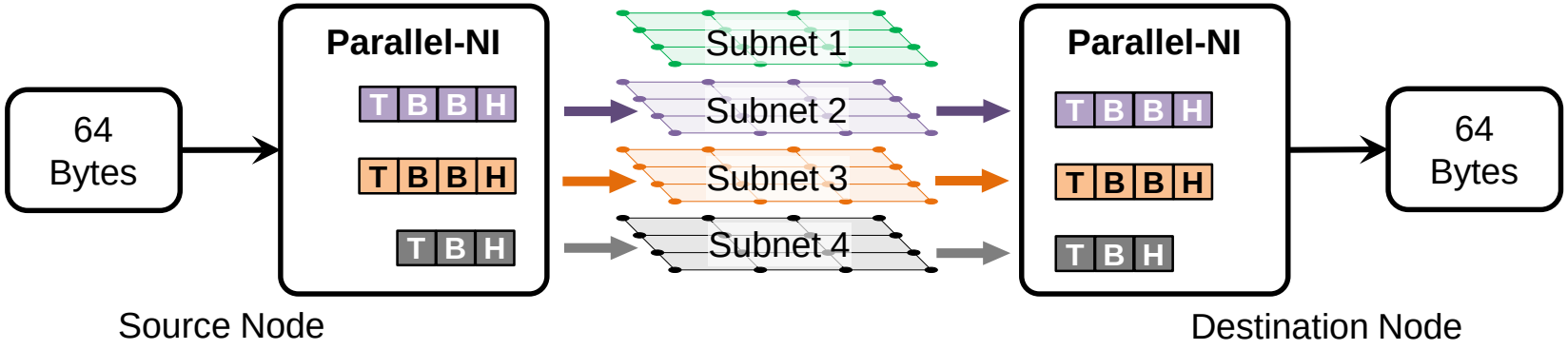
BoDNoC Design(1): Parallel-NI

Parallel-NI

- Access the subnets via multiple narrow interfaces **in parallel**
- Each interface equipped with individual buffer space
- **Hold and reassemble** the sub-packets of one data block at the destination node



Step1: Divide data into sub-packets | Step2: Select subnets | Step3: Hold & reassemble the sub-packets



BoDNoC Design(2): Subnets Allocation Policy

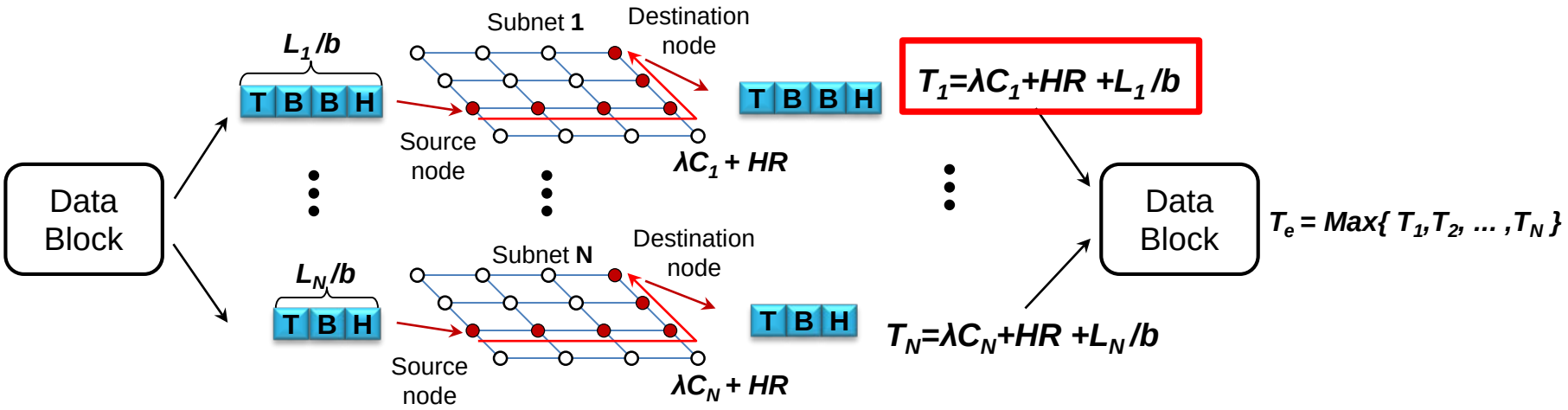
- **Three major design issues for Parallel-NI:**

1. ***how many*** *sub-packets for access data?*
2. ***what's the size*** *of each sub-packet?*
3. ***which*** *subnets should be selected to transmit data?*

BoDNoC Design(2): Subnets Allocation Policy

Question Description

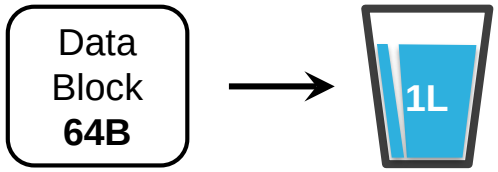
- **Objective:** to **minimize** the transmission latency of the data block T_e
- T_e is determined by the slowest sub-packet
  - T_e = Max { T_1, T_2, …, T_N }
- The latency of a sub-packet T_i is determined by three parts:
  - the size of sub-packet L_i
  - the congestion of subnet C_i
  - the distance between the source and the destination HR

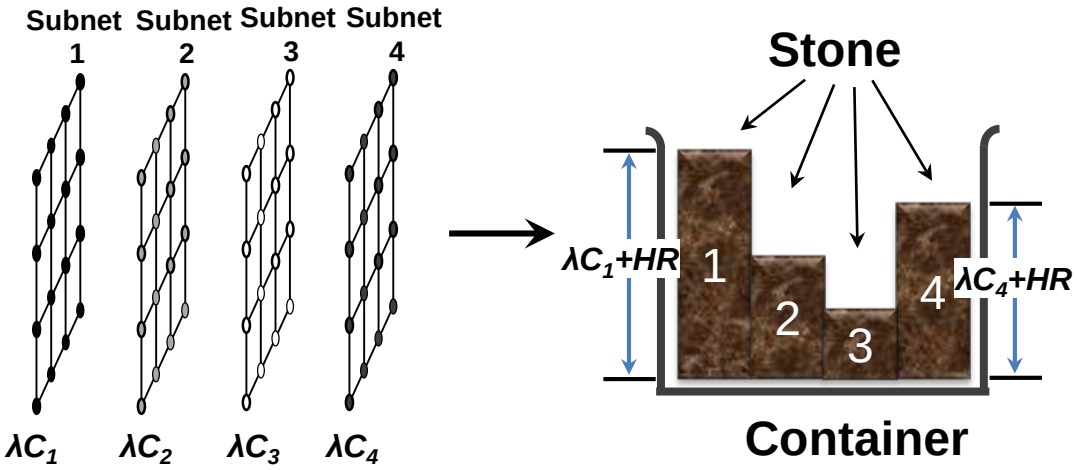


BoDNoC Design(2): Subnets Allocation Policy

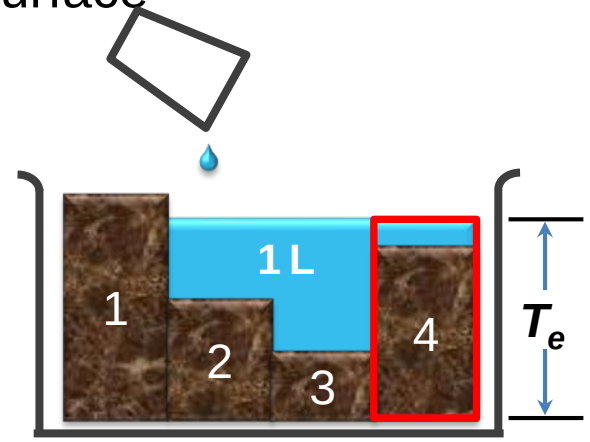
Water-Filling Algorithm

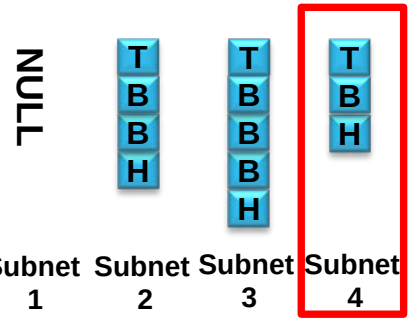
- Above question can be regarded as “Water Filling”
  - The access data block → Water in the cup
  - Transmit the data block → Pour water into the container
  - Latency of data block → Height of water surface





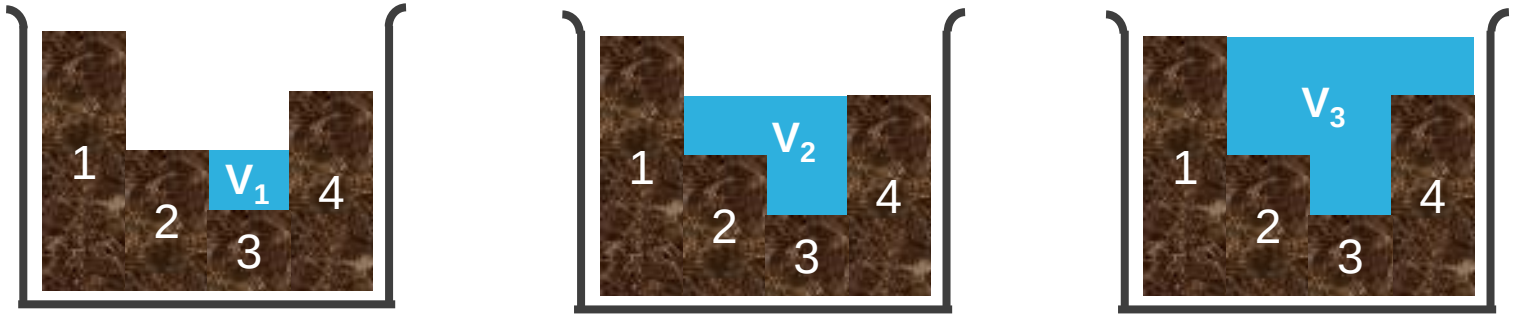
Congestion status of different subnets





BoDNoC Design(2): Subnets Allocation Policy

Water-Filling Algorithm



	$V \leq V_1$	$V_1 \leq V \leq V_2$	$V_2 \leq V \leq V_3$	$V_3 \leq V$
Number of subnets	1	2	3	4
Selected subnets	Subnet 3	Subnet 2,3	Subnet 2,3,4	All
Size of each sub-packet	V	$\frac{V}{2}-\frac{V_1}{2}, \frac{V}{2}+\frac{V_1}{2}$	$\frac{V}{3}-\frac{V_2}{6}-\frac{V_1}{2}$, …	…

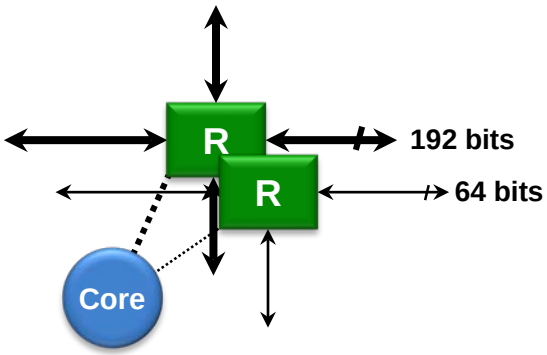
Experimental Setup

Platform Setup

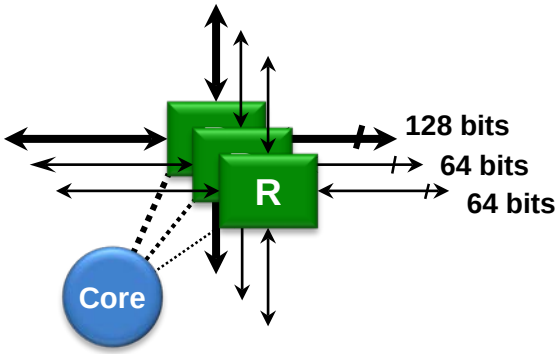
- Multi-Granularity Memory Systems: DGMS + Sector Cache
- Graph analysis workloads: CRONO and SSCA2

Baselines Setup

- Two Single-NoC designs (Single-64 and Single-256)
- Two Heterogeneous-NoC (Hetero-NoC-64-192 and Hetero-NoC-64x2-128)
- Two Homogeneous-NoC (Homo-NoC-64x4 and Homo-NoC-128x2)
  - With common NI, one data block just can be injected one subnet



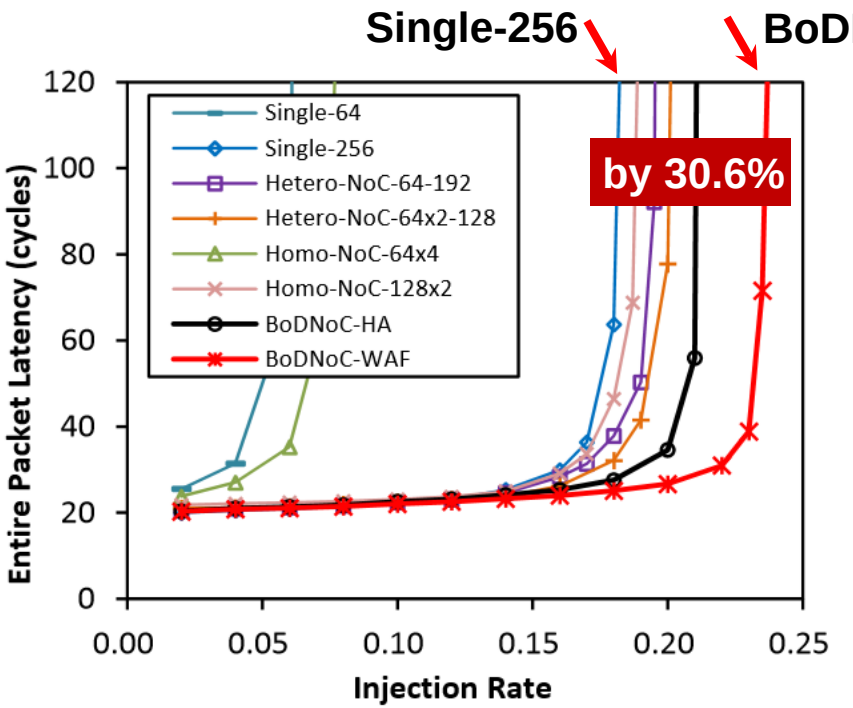
Hetero-NoC-64-192



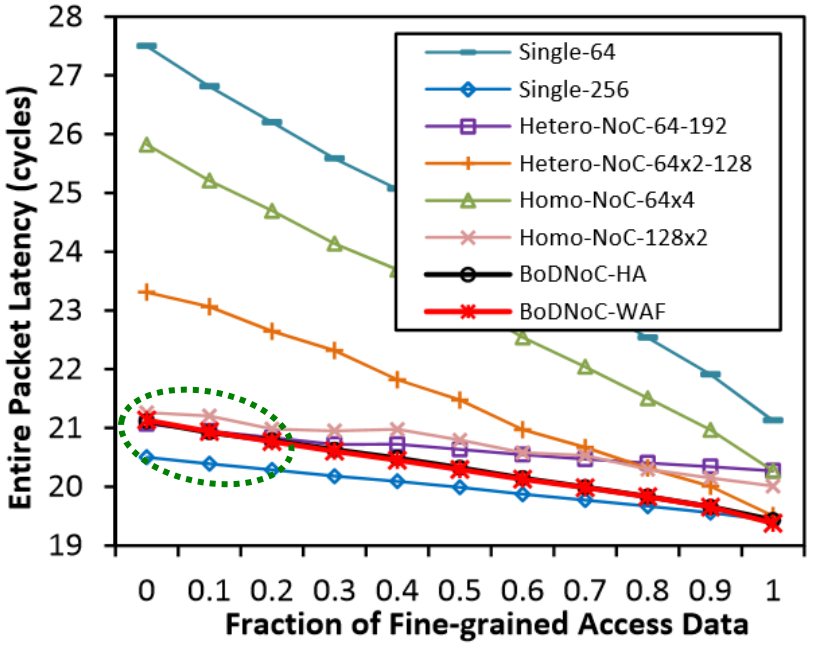
Hetero-NoC-64x2-128

Results: Synthetic Traffics (Uniform Random)

- Latency under various injection rates
  - Improve the throughput **by 30.6%** than Single-256
- Latency under different fraction of fine-grained access data
  - Introduces slight longer serialization latency than Single-256
  - **Overdo schedule** in non-congestion scene

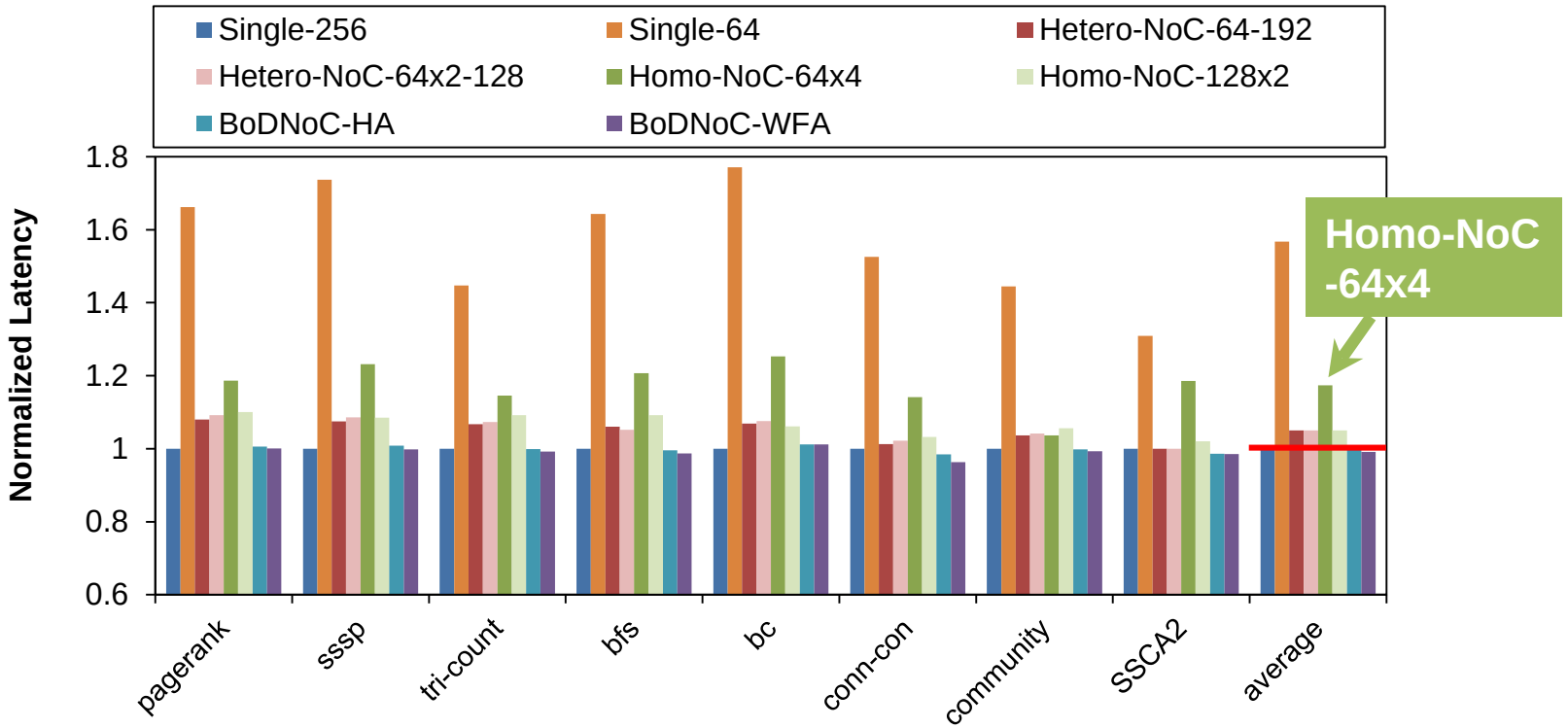


(50% fine-grained access data)



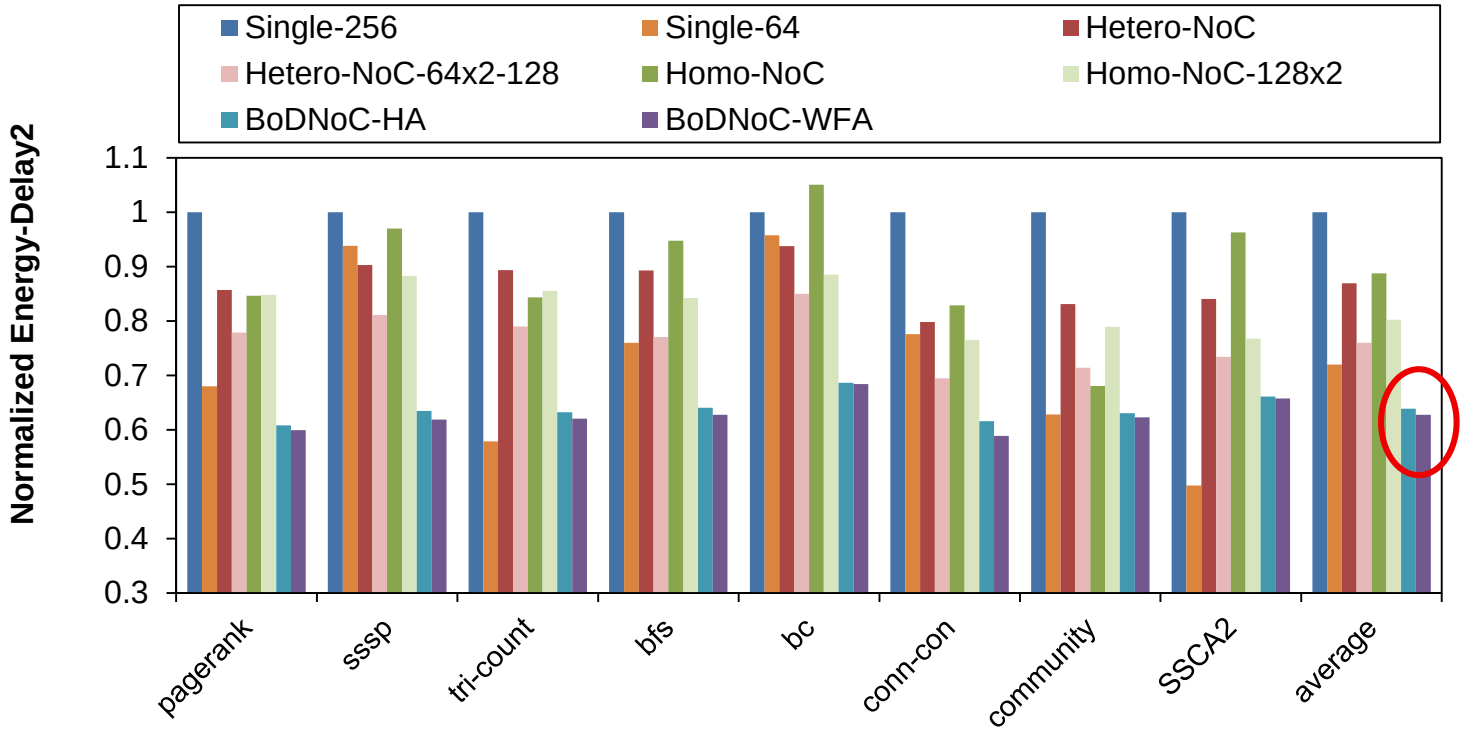
(Injection rate is 1%)

Results: Real Traffics - Performance



- For all applications, BoDNoC gets results similar to Single-256
- Without Parallel-NI, Homo-NoC-64x4 is **~18% higher than** BoD-NoC

Results: Real Traffics - Energy-efficient



- Except SSCA2 and tri-count, BoDNoC-WFA gets the lowest energy-delay^2
- Single-64 gets the best energy-delay^2 performance for SSCA2
  - the very low power dissipation
  - is not practical due to **the enormous performance impact**

Summary

- **Bandwidth-on-Demand NoC (BoDNoC)**
  - BoDNoC provides bandwidth **“as much as demand”** for the various granularities access data
  - **To take full advantage of bandwidth**, BoDNoC adopts Water-Filling algorithm to split and transmit the access data according to the congestion of network
  - BoDNoC gets the **best throughput and energy-efficiency** compared with existing designs

BoDNoC will be a promising design for multi-granularity memory system

Thank you

Q&A

Shiqi Lian, Ying Wang, Yinhe Han and Xiaowei Li

State Key Laboratory of Computer Architecture
Institute of Computing Technology, Chinese Academy of Science

ASP-DAC'2017, Chiba/Tokyo, Japan
19 January 2017