



异构多芯粒系统缓存一致性 协议设计与性能评估

答辩人：梅家诚

指导教师：尹捷明



目录

CONTENTS



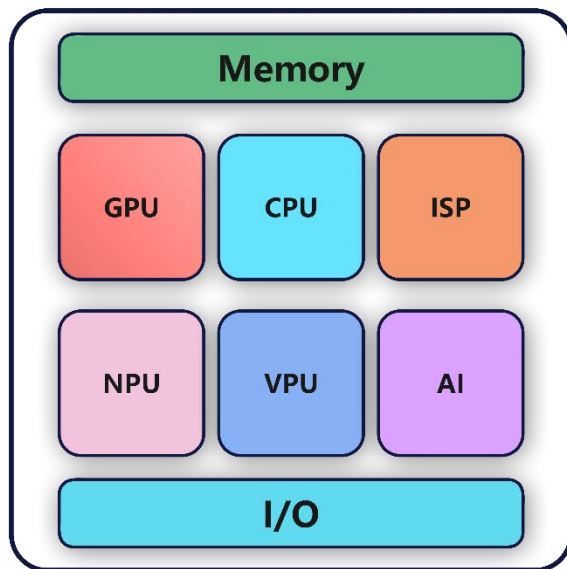
- 
- 1 选题背景与意义
 - 2 研究思路与解决方案
 - 3 实验成果与性能分析
 - 4 课题总结与展望



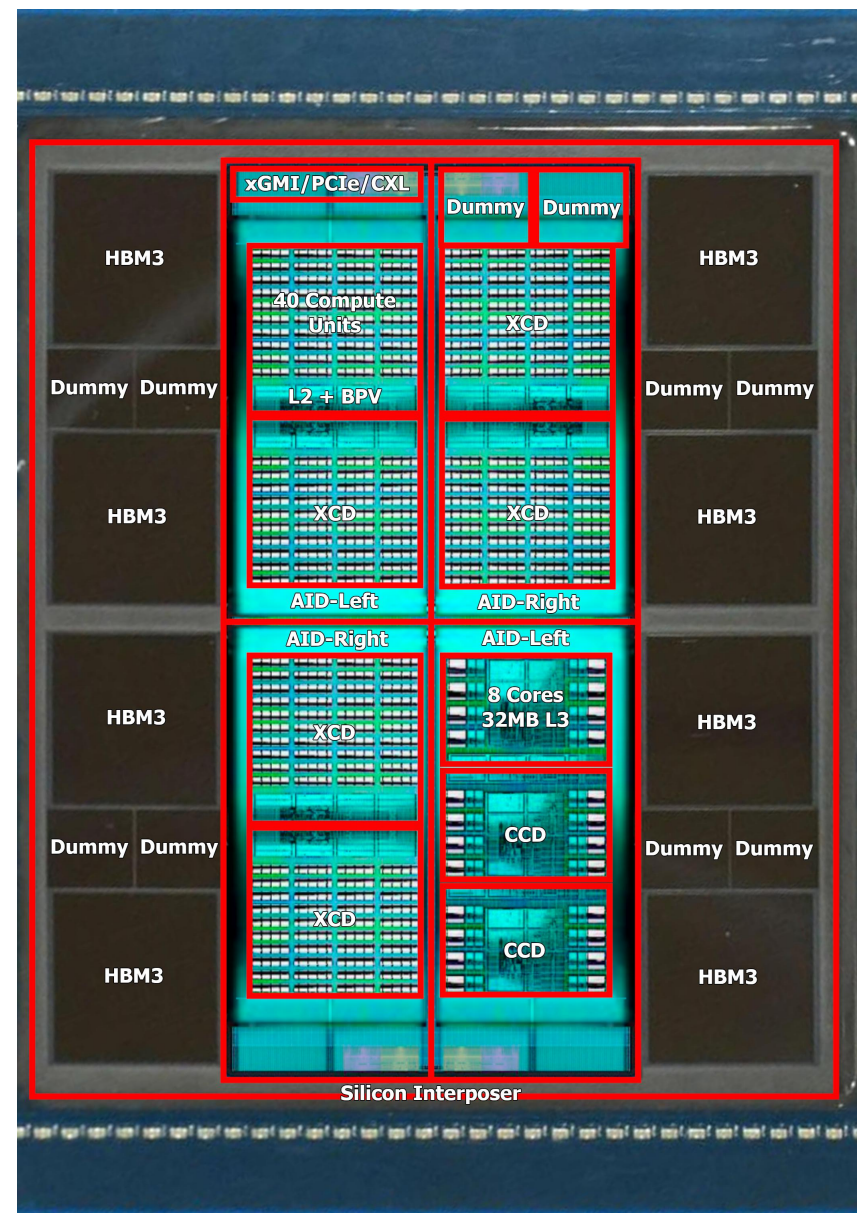
01

选题背景与意义

1.多芯粒技术 (Chiplet)



多芯粒技术，即Chiplet，又称小芯片和微芯片，是一种将复杂芯片拆分成多个小型的，独立且可以复用的模块的设计方法。这些独立的模块可以是处理器核心，如CPU、GPU，也可以是NPU、内存芯片及其他类型的集成电路。这些具有特定功能的模块通过先进的封装技术组合在一起，形成一个完整的系统芯片。



2.缓存一致性问题

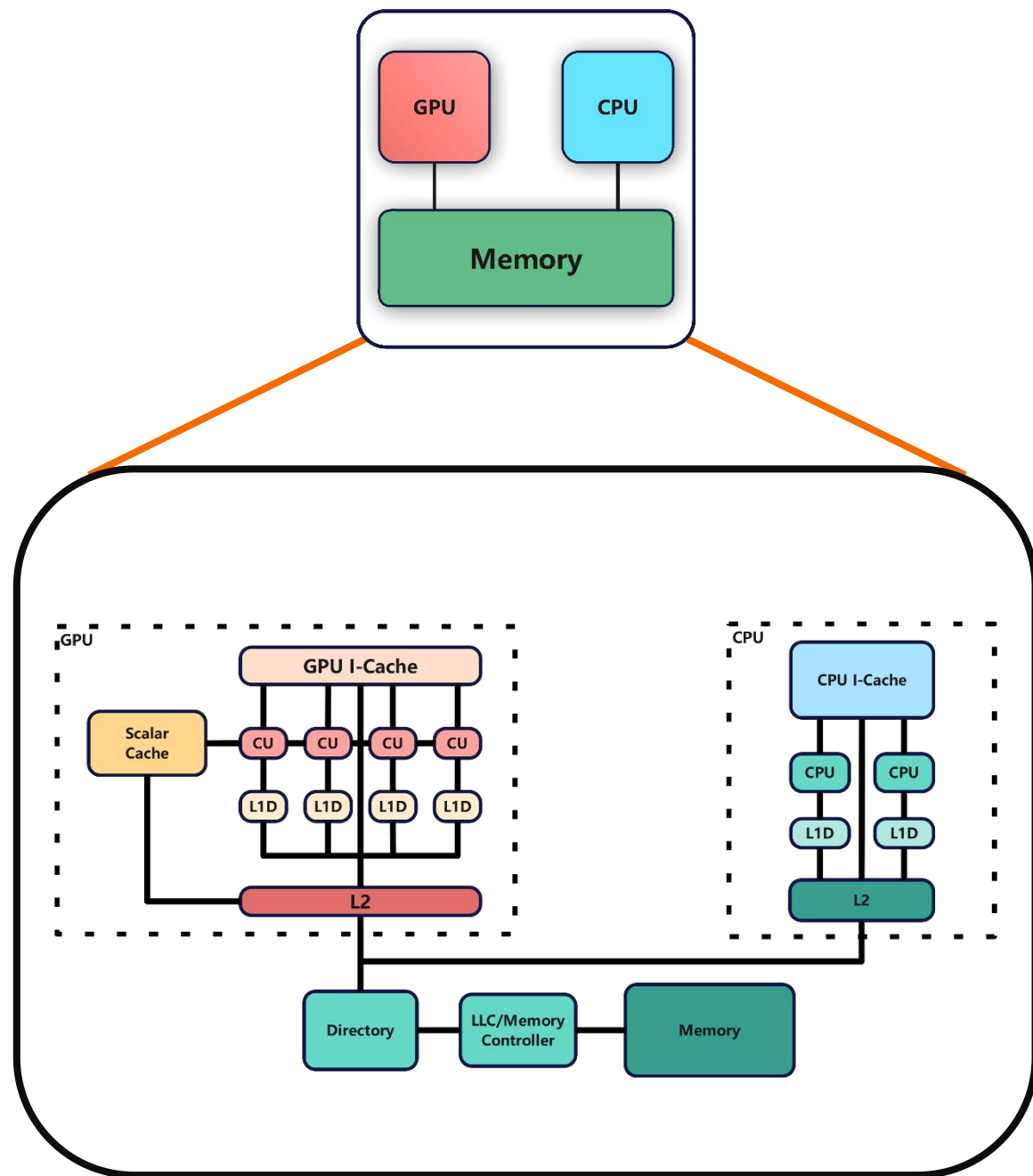
多芯粒系统中的各模块通常拥有多级缓存，而各模块之间又要进行数据的共享。为了确保所有芯粒看到的数据是一致的，即数据的正确共享，需要良好的缓存一致性协议的支持。

良好的缓存一致性协议，可以：

(1) 避免数据不一致性引起的错误和异常行为。

(2) 促进并行计算和任务协作，使得多个芯粒可以通过数据共享进行信息交换和任务分配

(3) 提高系统的运行效率





02

研究思路与解决方案

2.1研究思路

现有的对缓存一致性设计的研究，主要是针对基于某种协议实现的系统进行相应的改进，其改进主要包括三个方面：



1. 架构设计的改进



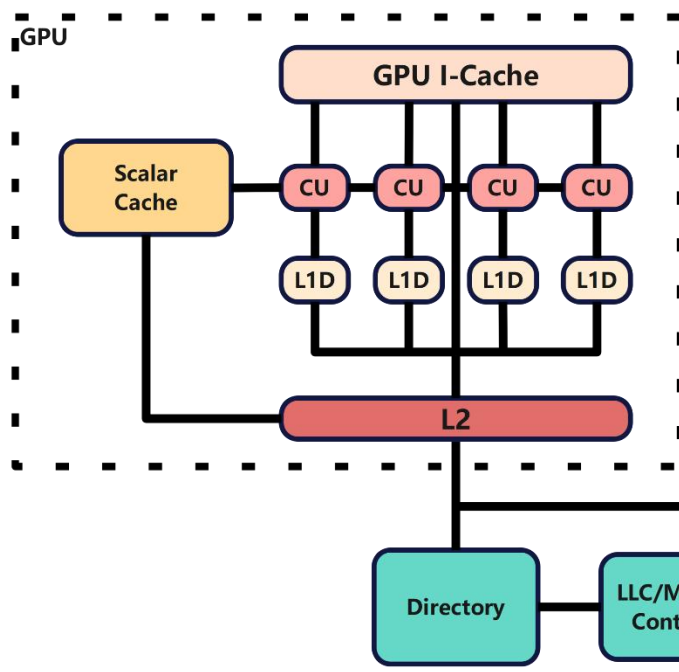
2. 片上网络的改进



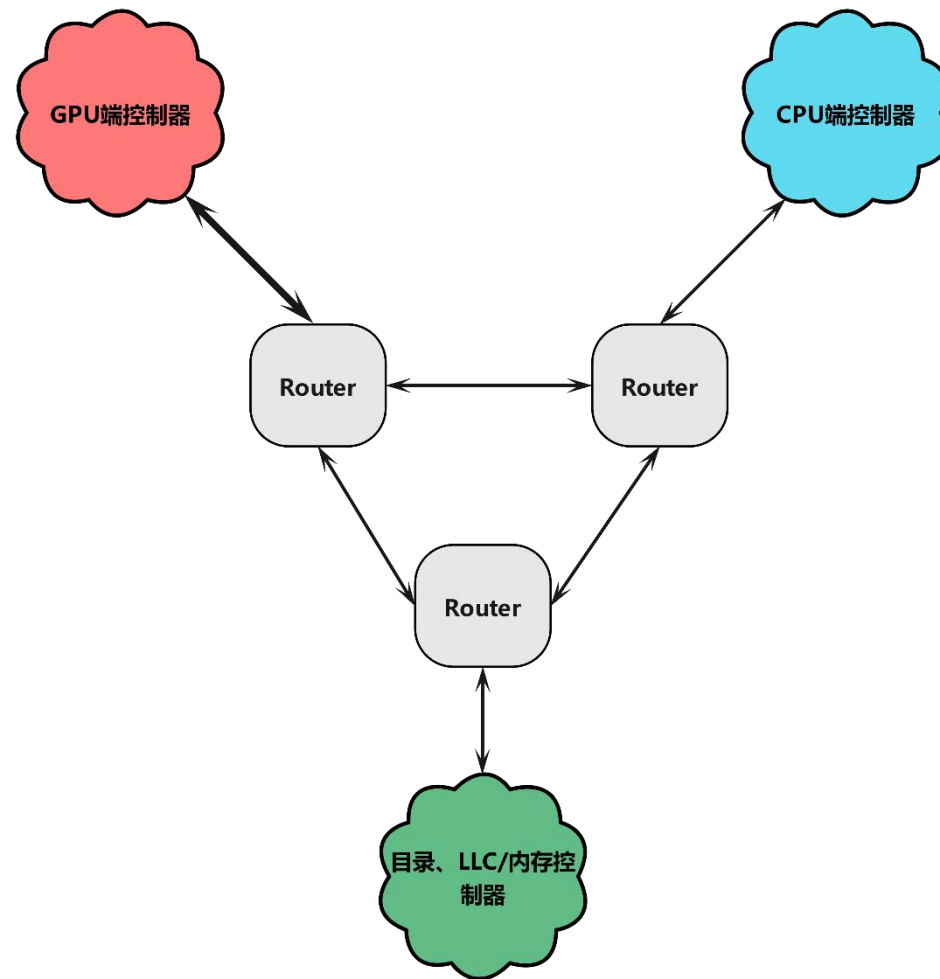
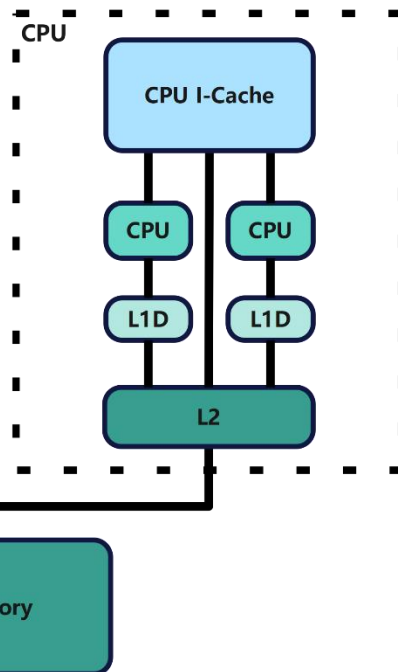
3. 底层协议的改进

2.2 基准系统

VI 协议



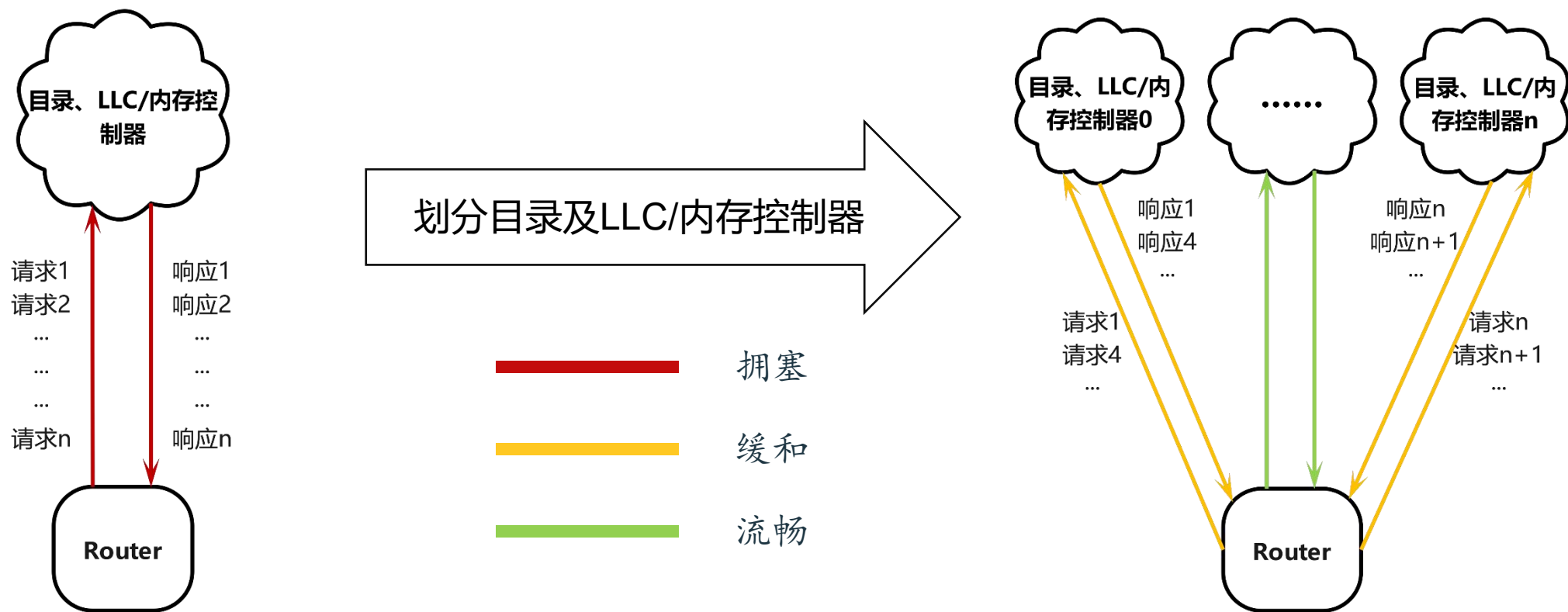
MOESI 协议



2.3 架构设计的改进——目录划分

问题：单个目录（Diretory）及LLC/内存控制器无法并行化处理同种请求。

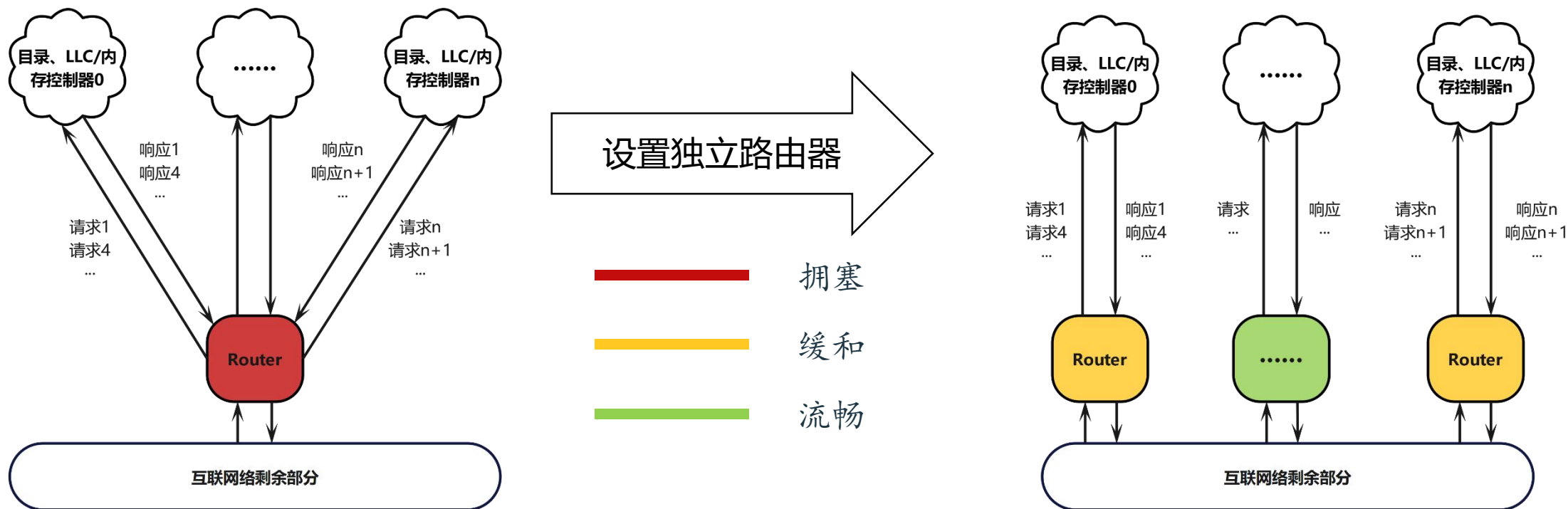
解决方案：划分目录及LLC/内存控制器，各子目录并行处理请求。



2.4 片上网络的改进——独立路由器

问题：单路由器资源有限，大量请求与响应消息容易造成拥塞。

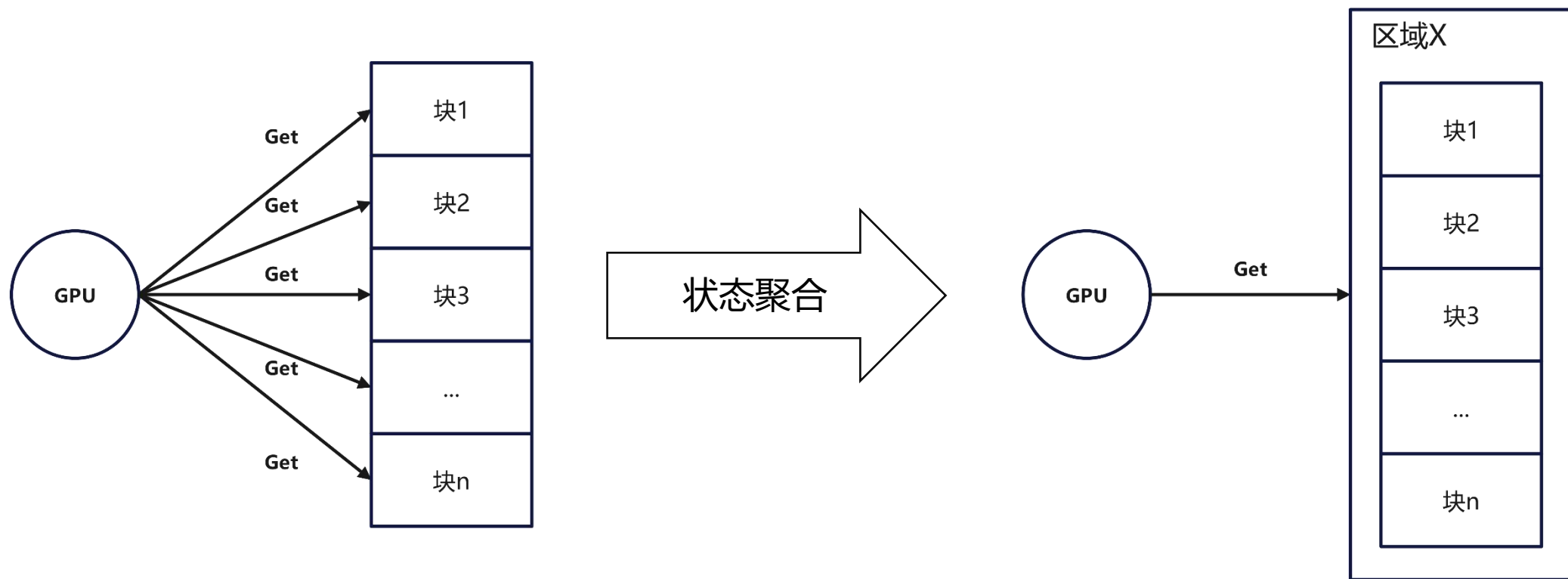
解决方案：为每一个子目录都设置一个独立的路由器。



2.5 (a) 底层协议的改进——状态聚合

问题： GPU对大量连续地址数据的访问会产生许多重复的缓存一致性消息。

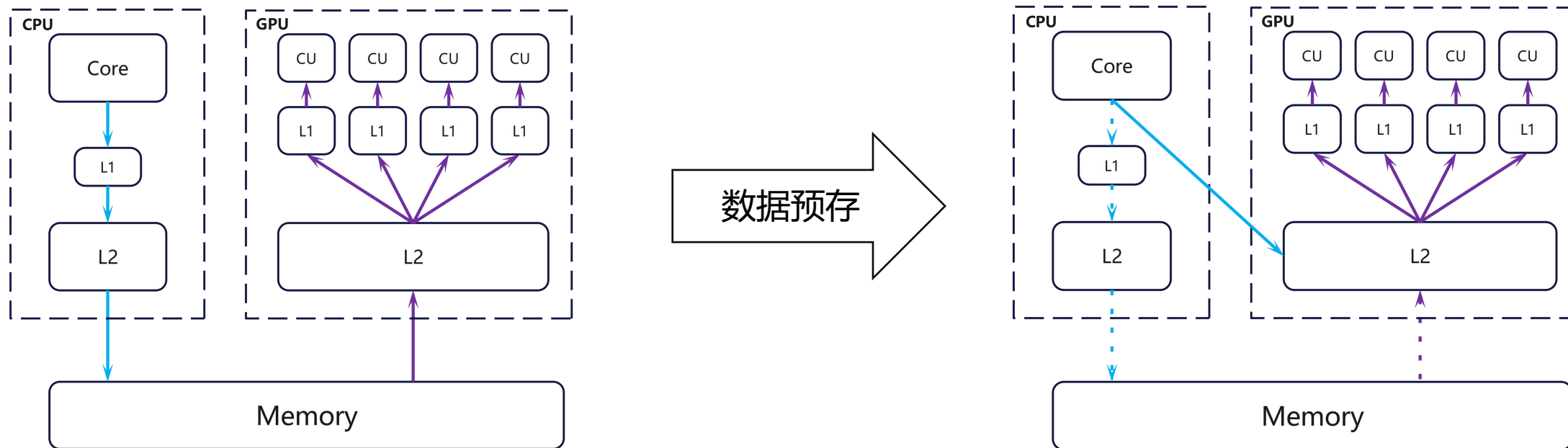
解决方案： 将若干连续块的状态聚合为一个态。



2.5 (b) 底层协议的改进——数据预存

问题： GPU需要从共享内存取CPU产生的数据，处理速度慢

改进办法： CPU将GPU所需数据尽可能直接存储到GPU的L2缓存（需要对协议进行改动，因为两个模块所用一致性不同）

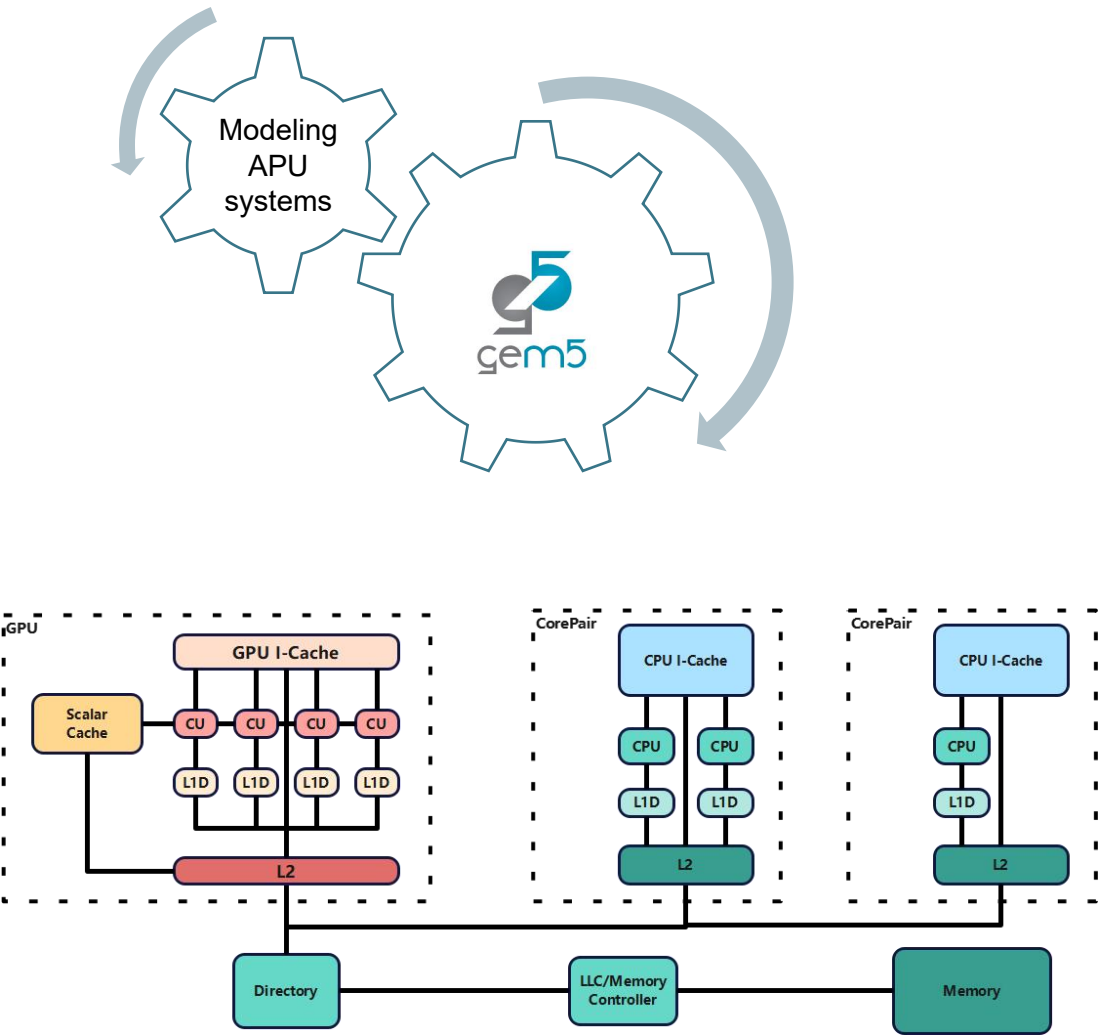




03

实验结果与性能分析

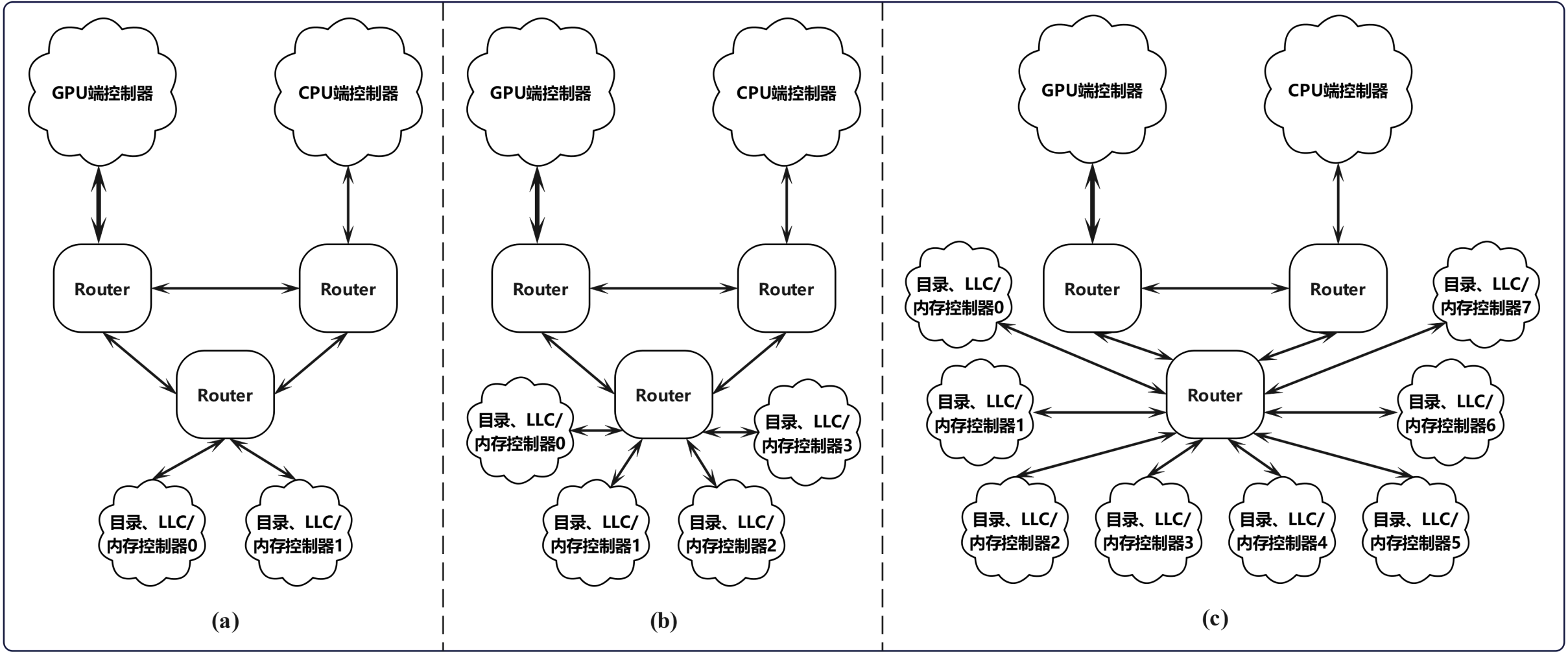
3.1 实验平台与仿真参数



系统参数表

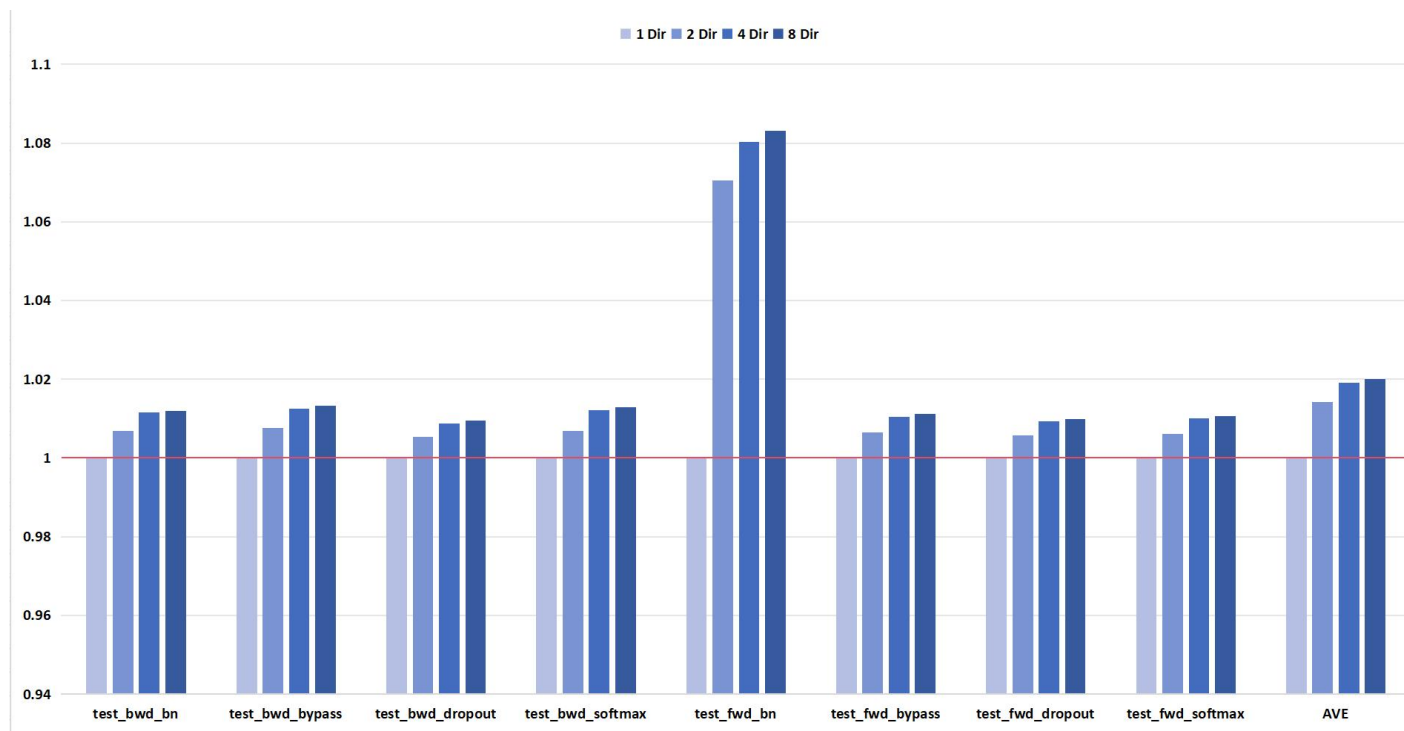
CPU config	3 CPU core,1 thread per core
CPU L1I Cache	32KB-2way,64B Cache line
CPU L1D Cache	64KB-2way,64B Cache line
CPU L2 Cache	2MB-8way,64B Cache line
LLC(L3)Cache	16MB-16way,64B Cache line
GPU config	4CU,4 simd-per-cu,10 wavefront-per-simd,64 per wavefront,4CU per SQC
GPU L1I Cache	32KB-8way,64B Cache line
GPU L1DCache	256KB-16way,64B Cache line
GPU L2 Cache	16KB-16way,64B Cache line
Dir	1/2/4/8
Memory	8GB

3.2 架构设计的改进——目录划分



3.2 (a) 目录划分实验结果的加速比分析

- (1) 三种划分均对系统带来一定的性能提升，并且数量更多的划分带来的性能提升也更多。
- (2) 2划分带来的平均性能提升为1.4%，最高可达7.1%；4划分带来的平均性能提升1.9%，最高可达8.0%；8划分带来的平均性能提升为2.0%，最高可达8.3%。
- (3) 2划分相较于不划分带来1.4%左右的性能提升，4划分相较于2划分，8划分相较于4划分几乎没有提升。
- (4) 划分会带来额外的硬件开销，控制器的划分并不是越多越好

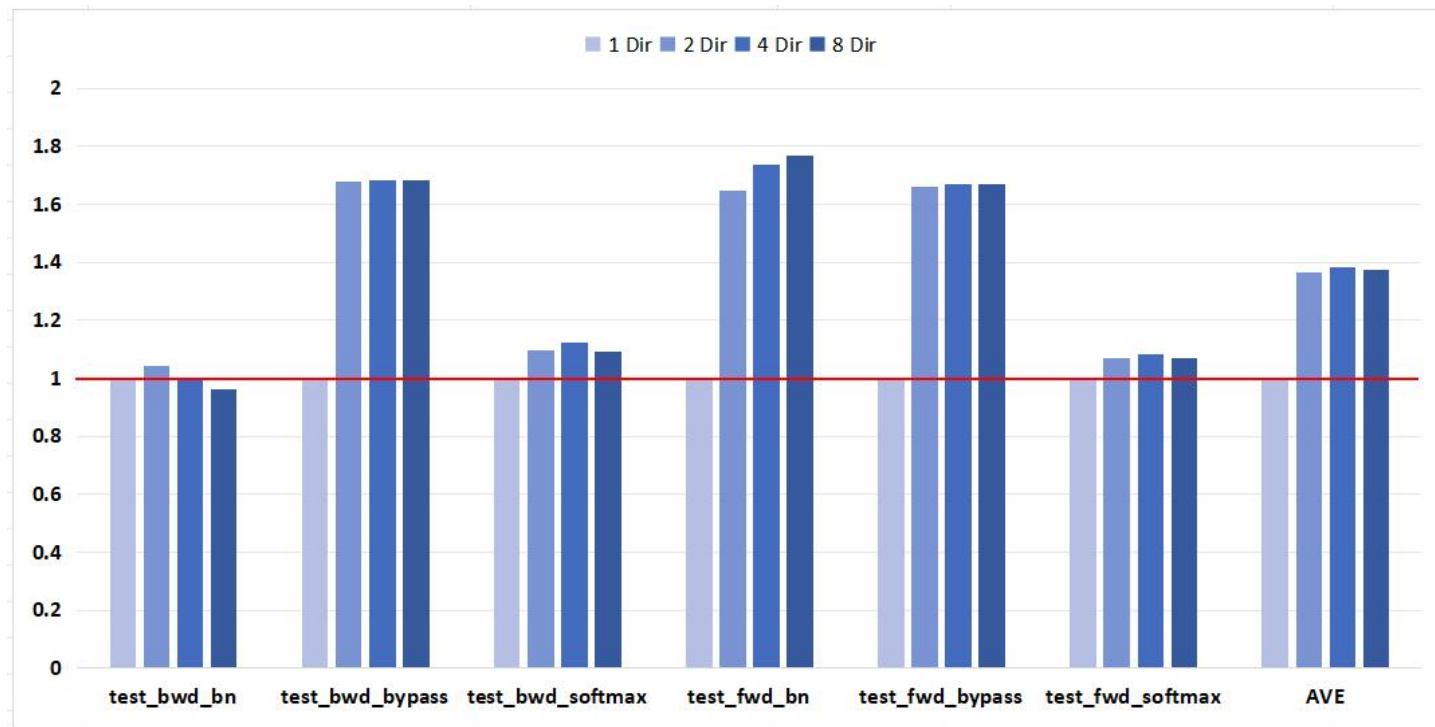


三种划分相较于基准系统的加速比

3.2 (b) 目录划分实验结果的VPC分析

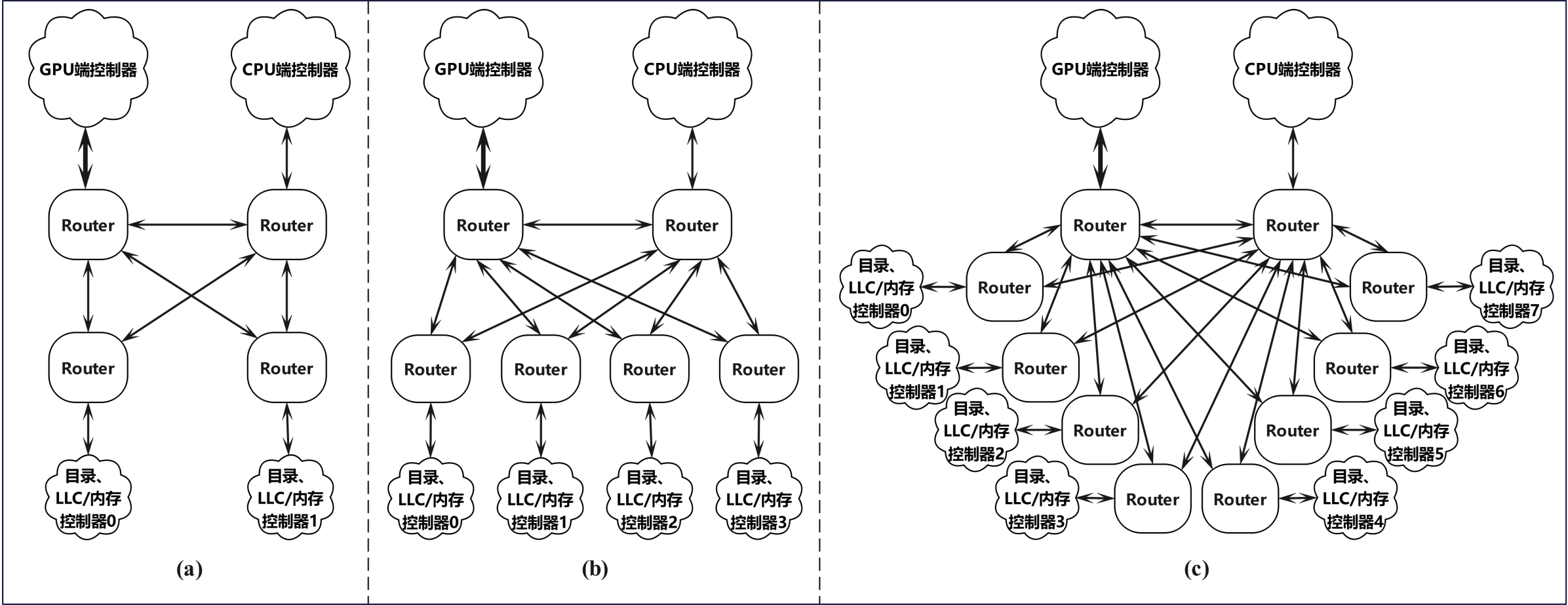
(1) 2划分的平均VPC是基准系统的1.36倍，最高可达1.68倍；4划分、8划分的平均VPC分别是基准系统的1.38，1.37倍，最高分别可达1.73，1.76倍

(2) 2划分系统相较于基准系统VPC提升明显，而4划分相较于2划分，8划分相较于4划分的VPC近乎无变化。



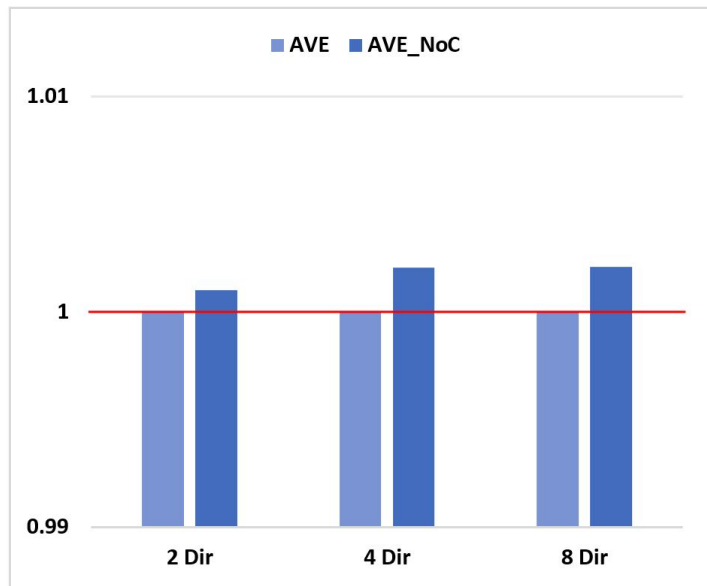
三种划分相较于基准系统的VPC

3.3 片上网络的改进——独立路由器

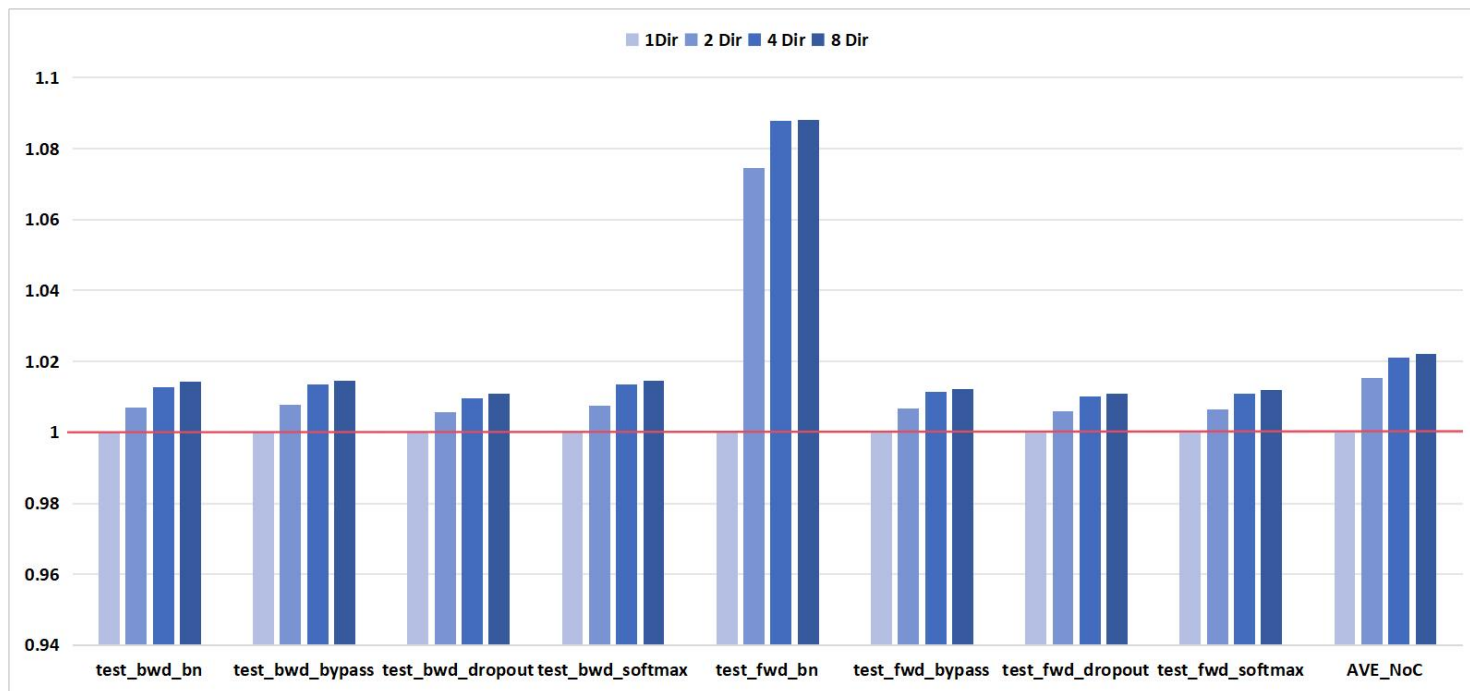


3.3 (a) 网络更改实验结果的加速比分析

- (1) 2划分拓扑带来的平均性能提升为**1.5%**，4划分、8划分拓扑带来的平均性能提升均在**2.1%**左右。
- (2) 2划分带来的性能提升明显，4划分相较于2划分，8划分相较于4划分提升有限
- (3) 设置路由器带来的加速比提升并不明显（相较于不设置**仅1%不到**）。



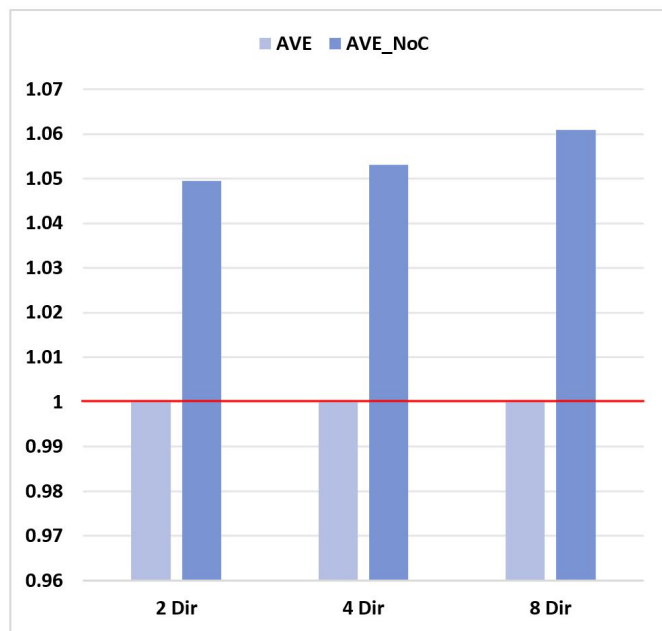
三种划分更改网络前后的平均加速比比较
(以更改前为基准)



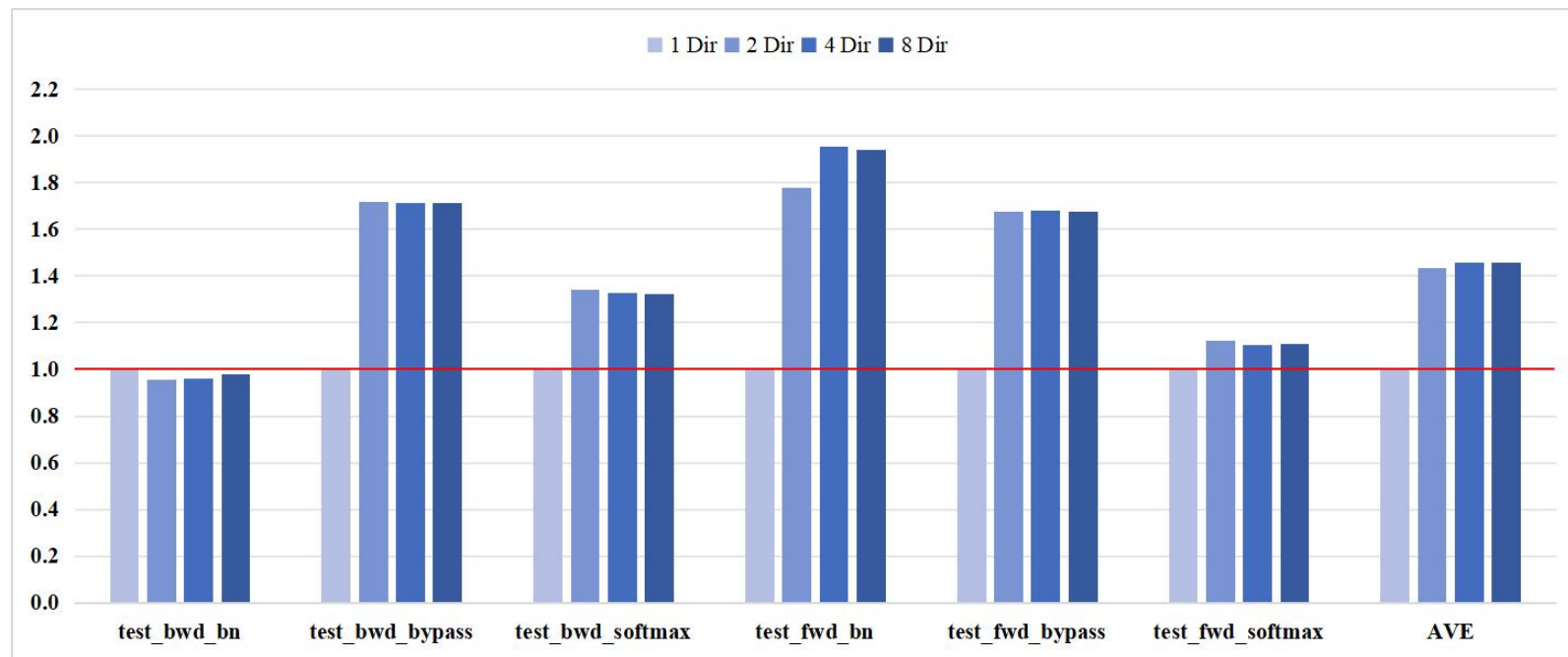
三种划分在更改网络拓扑结构后相较于基准系统的加速比

3.3 (b) 网络更改实验结果的VPC分析

- (1) 2划分的平均VPC是基准系统的1.44倍，最高可达1.78倍；4划分、8划分的平均VPC均是基准系统的1.45倍，最高分别可达1.95，1.94倍
- (2) 2划分相较于不划分VPC提升最明显，各划分之间VPC近乎持平。
- (3) 设置路由器相较于不设置而言，平均VPC提高超5%。



三种划分更改网络前后的平均VPC比较
(以更改前为基准)



三种划分在更改网络拓扑结构后相较于基准系统的VPC



04

总结与展望

4.1 课题总结

本文对异构多芯粒缓存一致性的设计空间进行了全面的探索，分析了设计中可能会遇到的缓存一致性协议瓶颈、架构设计瓶颈以及片上网络拓扑结构瓶颈。

(1) 针对对协议上的瓶颈我们提出了两种解决办法：将数个块的状态聚合为一个区域状态以及将CPU产生的GPU所需的数据直接交付给GPU的缓存。

(2) 对于架构上的瓶颈，我们提出了划分目录及LLC/内存控制器的解决方案，以提高目录的并行处理能力。

(3) 然后我们结合划分方案以及系统的片上网络拓扑结构，提出了进一步的改进方案：为每一个划分后的目录及LLC/内存控制器设置独立的路由器来解决可能出现的片上网络瓶颈。

我们通过gem5模拟器对后两种方案进行了实验设计和验证。并从加速比和VPC两个角度对实验结果进行了分析、比较以及综合。实验结果表明，对目录及LLC/内存控制器进行2划分并为每个划分后的目录及LLC/内存控制器设立独立的路由器是最佳的解决方案，该方案可使系统整体性能提高1.5%，平均VPC提高1.44倍，并且相较于4划分和8划分所需的硬件支持更少。

4.2 课题展望

(1) 由于时间原因，本文仅对协议瓶颈改进办法做出了描述而并未进行实际验证。这两种方案虽然易于理解，但在具体实现中遇到了不少的困难，需要耗费大量的时间。未来的工作可以集中实现并验证两种解决方案带来的具体效果。

(2) 针对设置独立路由器这一方法带来的VPC提高，加速比近乎不变现象，本文没有进行深入的研究。未来的研究可以对这方面进行深入。

(3) 本文对片上网络的探索比较浅薄。片上网络的设计实际上相当复杂，异构系统中的片上网络还会受到缓存一致性协议具体实现的影响。本文仅仅是从最简单的拓扑结构角度进行了研究，并没有做更深入的讨论。未来的工作可以在片上网络上进行深入探索，比如设计良好的路由算法、流控算法或研究路由器的微架构设计等。

(4) 本文仅对部分设计空间进行了探索，而实际上的设计空间是非常广泛的。本文所提出的三个方向的改进办法，实际上可以进行任意的组合排列，从而得出不同的系统设计。未来的工作可以对这些不同的系统设计进行深入的研究。



恳请各位老师指正!

汇报人：B20030428梅家诚 汇报时间：2024.6.6