二、分离式内存技术案例分析

讲者：

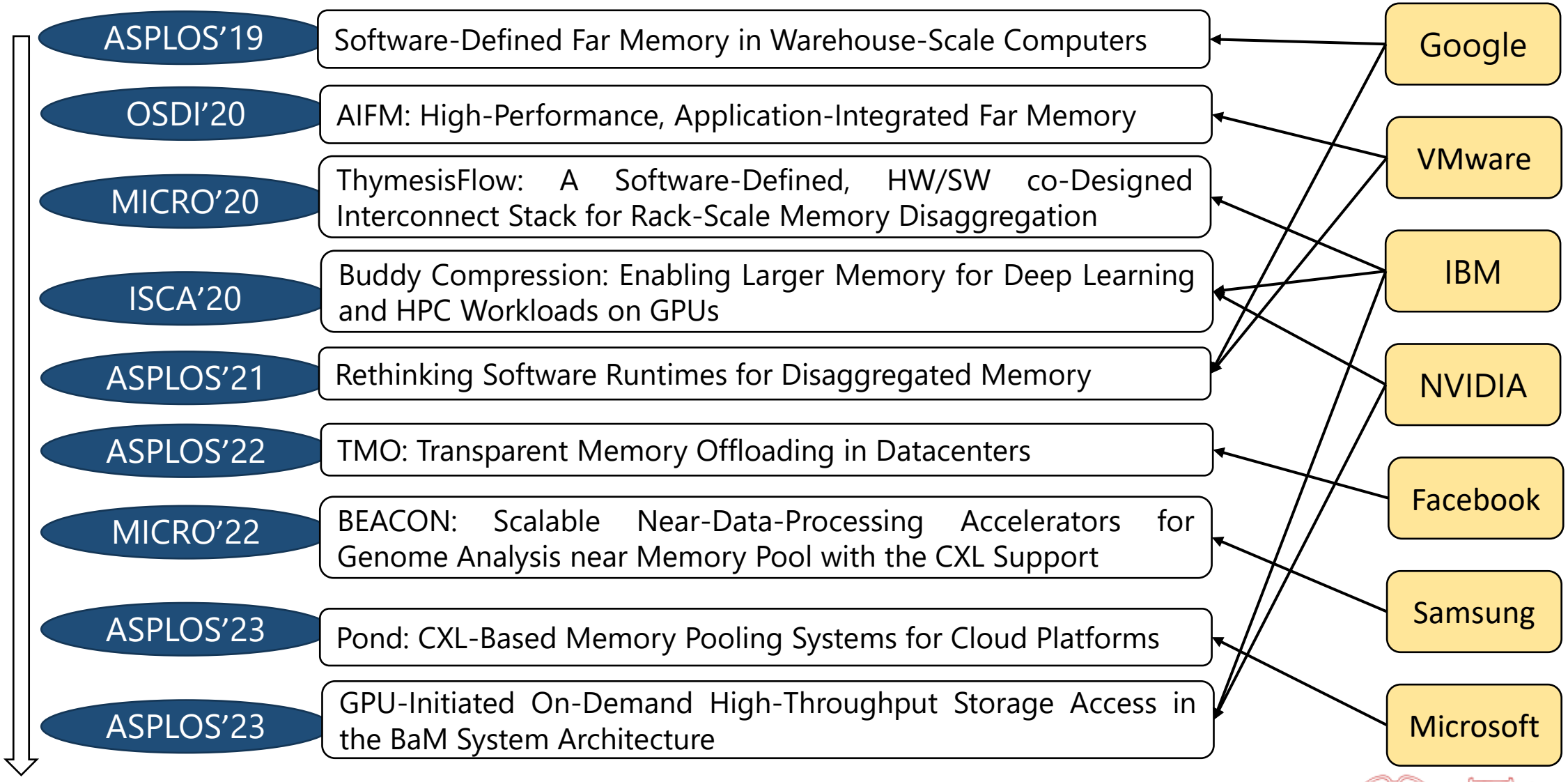
上海交通大学 计算机系 SAIL实验室

2023年2月

上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

目录

1.国外企业技术案例


ASPLOS'19
Software-Defined Far Memory in Warehouse-Scale Computers
OSDI'20
AIFM: High-Performance, Application-Integrated Far Memory
MICRO'20
ThymesisFlow: A Software-Defined, HW/SW co-Designed Interconnect Stack for Rack-Scale Memory Disaggregation
ISCA'20
Buddy Compression: Enabling Larger Memory for Deep Learning and HPC Workloads on GPUs
ASPLOS'21
Rethinking Software Runtimes for Disaggregated Memory
ASPLOS'22
TMO: Transparent Memory Offloading in Datacenters
MICRO'22
BEACON: Scalable Near-Data-Processing Accelerators for Genome Analysis near Memory Pool with the CXL Support
ASPLOS'23
Pond: CXL-Based Memory Pooling Systems for Cloud Platforms
ASPLOS'23
GPU-Initiated On-Demand High-Throughput Storage Access in the BaM System Architecture
Google
VMware
IBM
NVIDIA
Facebook
Samsung
Microsoft

1.国外企业技术案例：Google

ASPLOS'19 | Software-Defined Far Memory in Warehouse-Scale Computers

主要发现：

- 存在大量的冷页碎片（32%）
- 冷页比例变化范围大（1~52%）

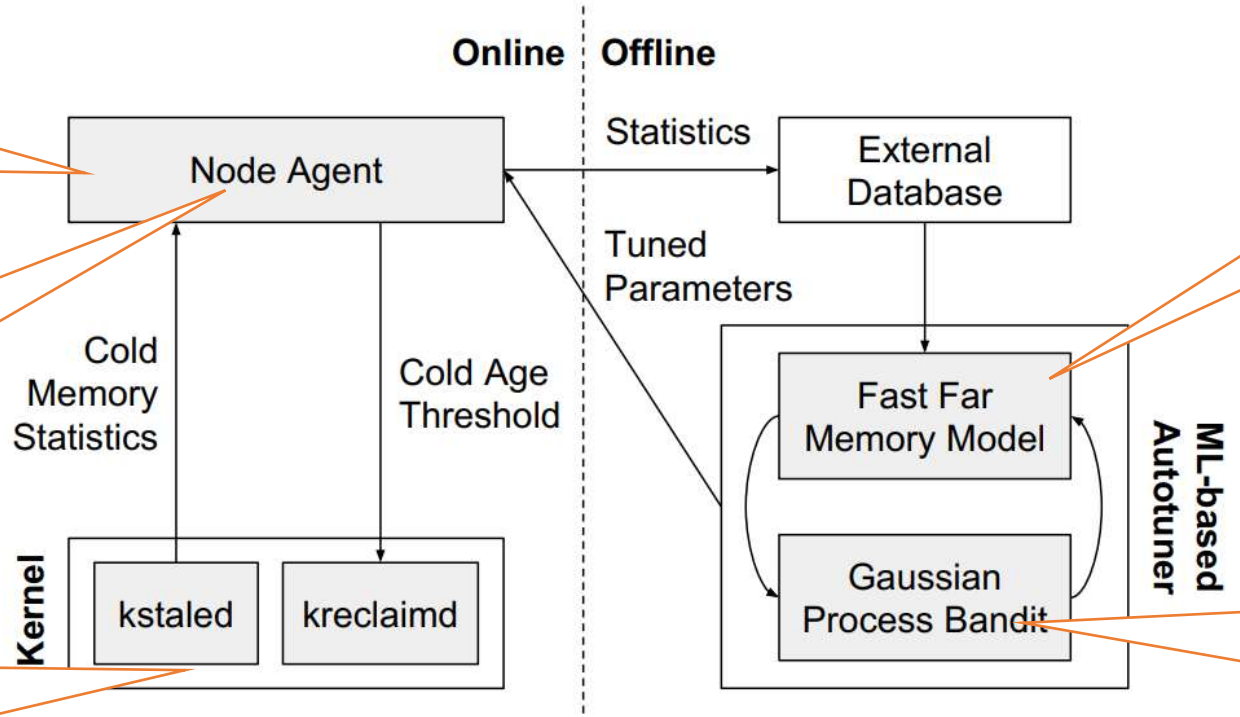
主要提出：

- 使用zswap将冷页压缩
- 按照任务SLO将冷页换出到远内存

周期性地更新冷页阈值，并收集数据供线下分析

Zswap在S秒内启动，设置下一秒的百分数为当前的第K百分位

周期性地扫描页表中的访问位，在DRAM和zswap中移动冷页面



并行化独立控制不同job，根据trace计算S和K

预测算法，寻找重要配置参数，尽量减少试验次数

Figure 4. Overall system design.

1.国外企业技术案例：Google

ASPLOS'19 Software-Defined Far Memory in Warehouse-Scale Computers

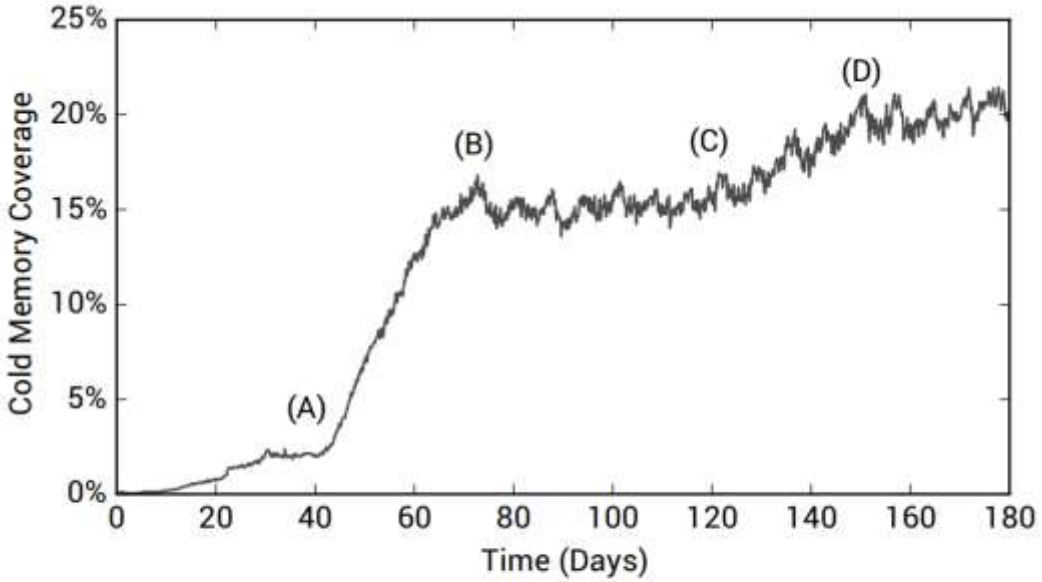


Figure 5. Cold memory coverage over time. zswap with hand-tuned parameters was rolled out during (A) to (B); the autotuner was rolled out during (C) to (D).

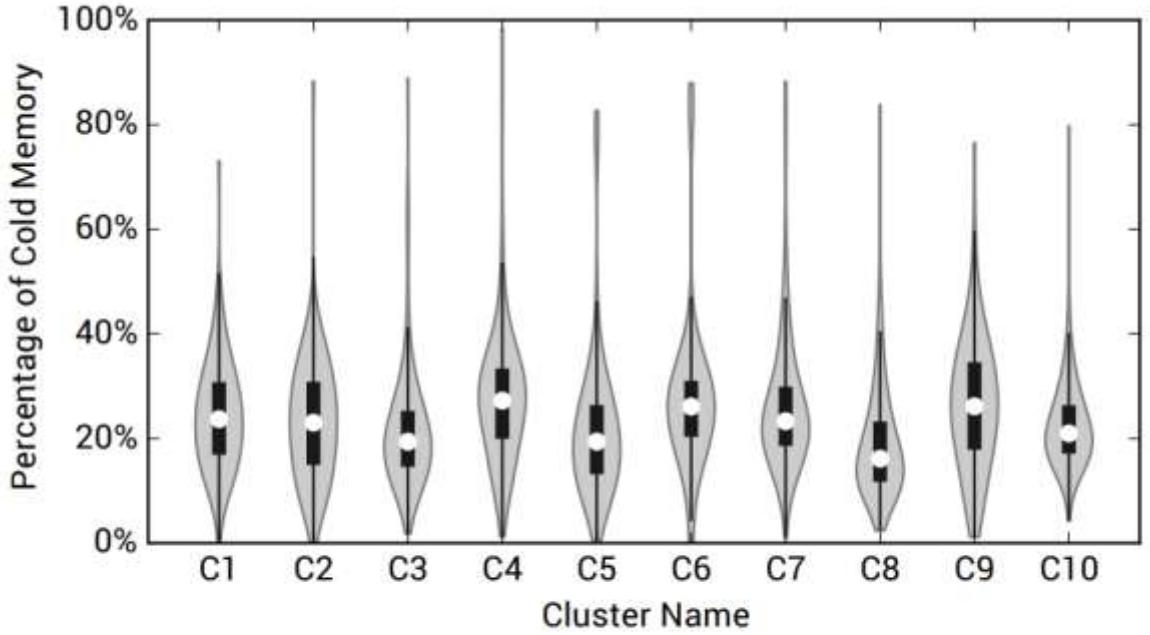


Figure 6. The distribution of the cold memory coverage across the machines in the top 10 largest clusters.

zswap： 15% of cold memory coverage.

ML-based autotuner： increased the cold memory coverage to 20%

32% for the upper bound for cold memory ratio

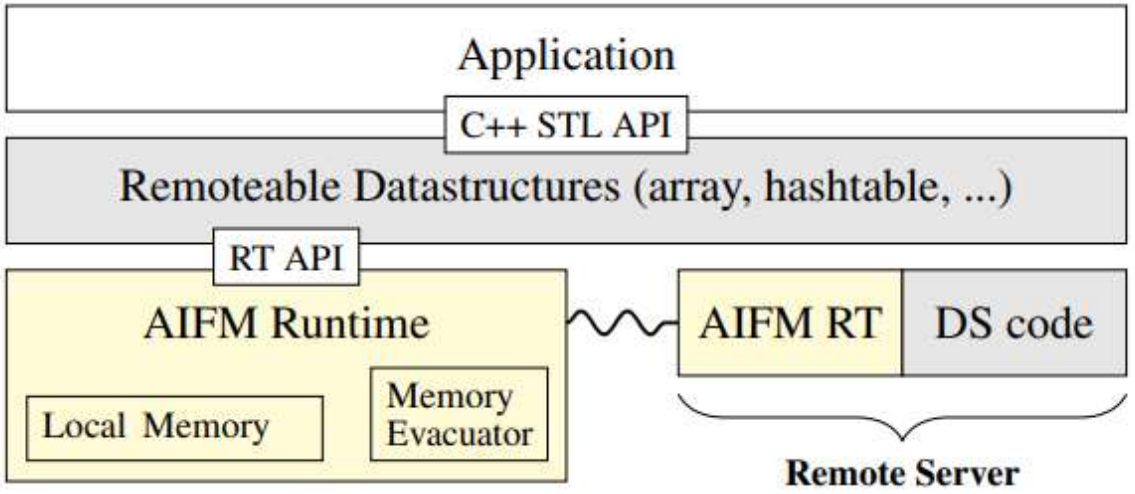
67% cost reduction（3x compression ratio） for compressed pages

1.国外企业技术案例：VMware

OSDI'20 AIFM: High-Performance, Application-Integrated Far Memory

核心思想：

利用应用层的API实现本地内存和远内存的一致访问控制，需要用户根据AIFM提供的接口进行程序的适配




```
std::unordered_map<key_t, int> hashtable;
std::array<data_t> arr;

void print_data(std::vector<key_t>& request_keys) {
  int sum = 0;
  for (auto key : request_keys) {
    sum += hashtable.at(key);
  }
  std::cout << arr.at(sum) << std::endl;
}
```

The same code written using AIFM looks like this:

```
RemHashtable<key_t, int> hashtable;
RemArray<data_t> arr;

void print_data(std::vector<key_t>& request_keys) {
  int sum = 0;
  for (auto key : request_keys) {
    DerefScope s1; // Explained in Section 4.2.2.
    sum += hashtable.at(key, s1);
  }
  DerefScope s2;
  std::cout << arr.at(sum, s2) << std::endl;
}
```

1.国外企业技术案例：IBM

MICRO'20

ThymesisFlow: A Software-Defined, HW/SW co-Designed Interconnect Stack for Rack-Scale Memory Disaggregation

主要贡献：

- 提出了带有软件控制平面的全硬件分离式内存硬件架构原型
- 可以动态创建NUMA节点，并和其Linux内核进行内存的实时卸载和读取
- 在现有商业硬件搭建原型，有着完整的软硬件系统

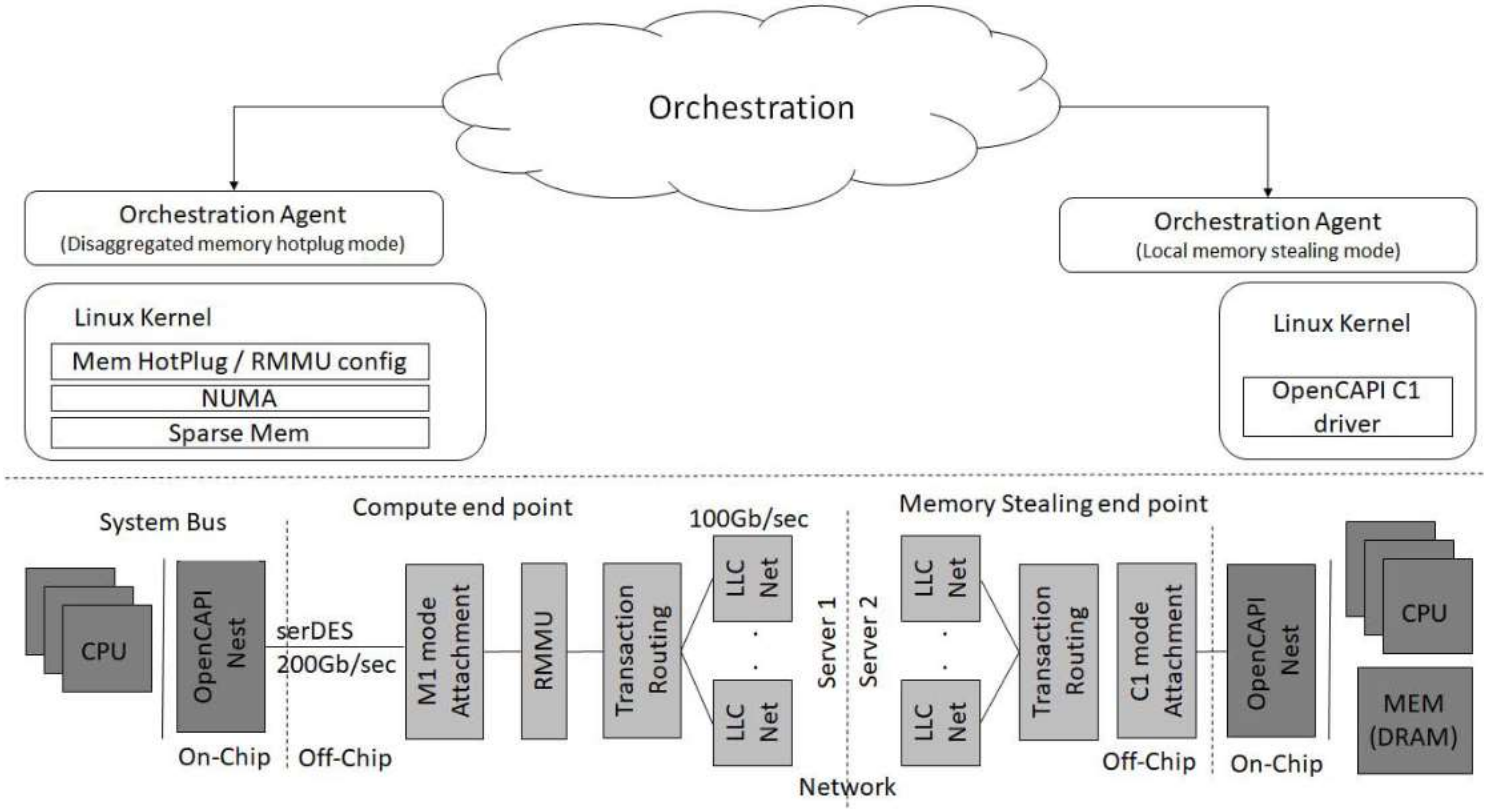


Fig. 2. ThymesisFlow overall architecture.

1.国外企业技术案例：IBM

MICRO'20

ThymesisFlow: A Software-Defined, HW/SW co-Designed Interconnect Stack for Rack-Scale Memory Disaggregation

- 此外，ThymesisFlow中为了实现动态加入内存节点，将地址空间映射到内存节点内核，也为内存池构建一致性的保存问题提供了解决方案

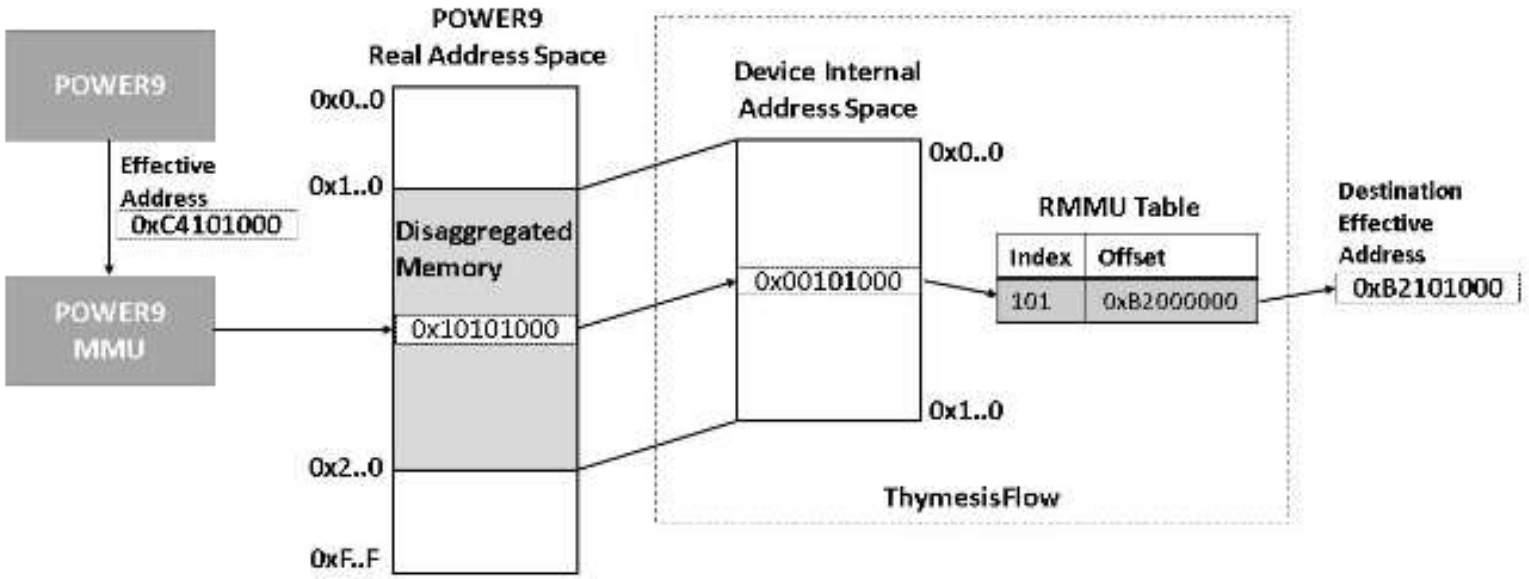


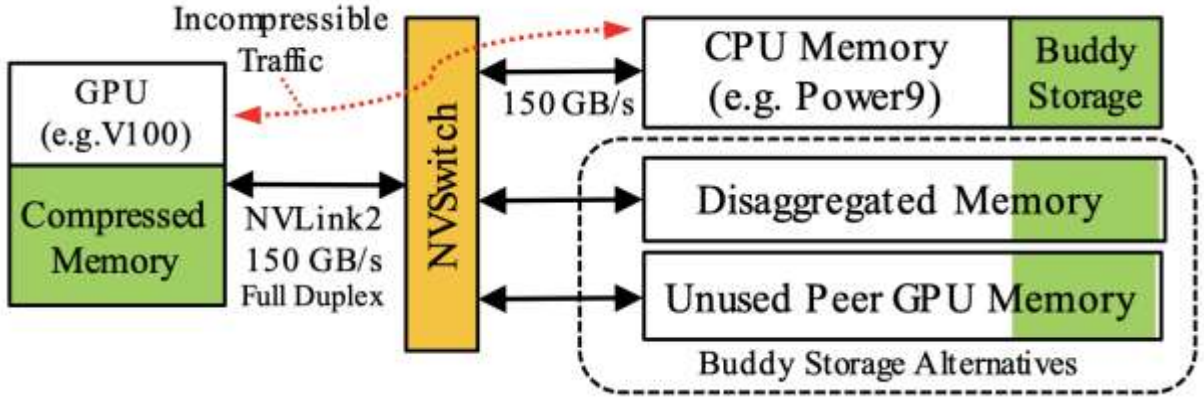
Fig. 3. ThymesisFlow address translation process.

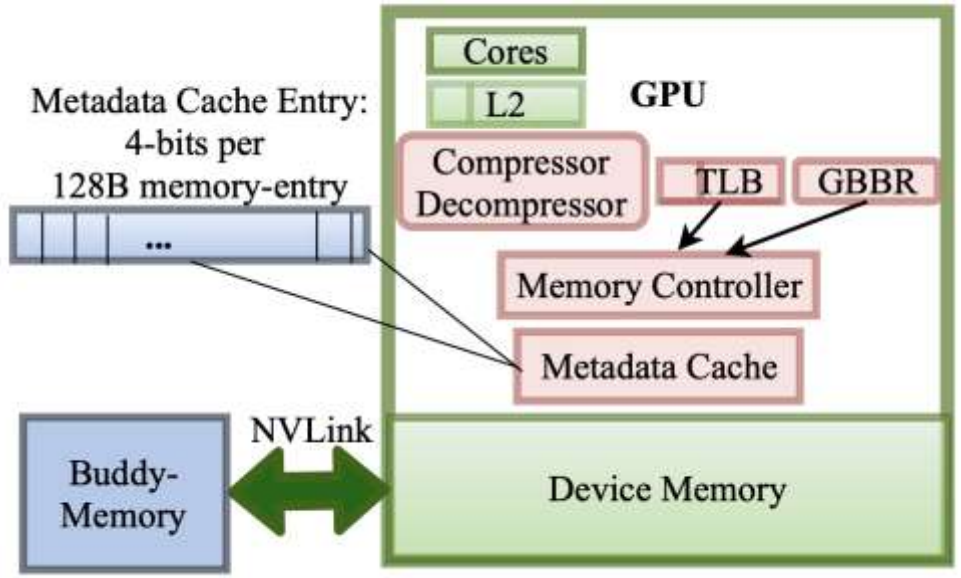
1.国外企业技术案例：NVIDIA

ISCA'20

Buddy Compression: Enabling Larger Memory for Deep Learning and HPC Workloads on GPUs

- 提出Buddy Compression方法，首次实现了使用压缩算法节省GPU内存使用
- 多数可压缩的内存条目完全由GPU内存访问，而少数的溢出则从GPU以外的内存中获取
- 理想情况下，远程存储可以使用系统内存，也可以使用分离式内存（如左图）
- 需对硬件进行相应更改





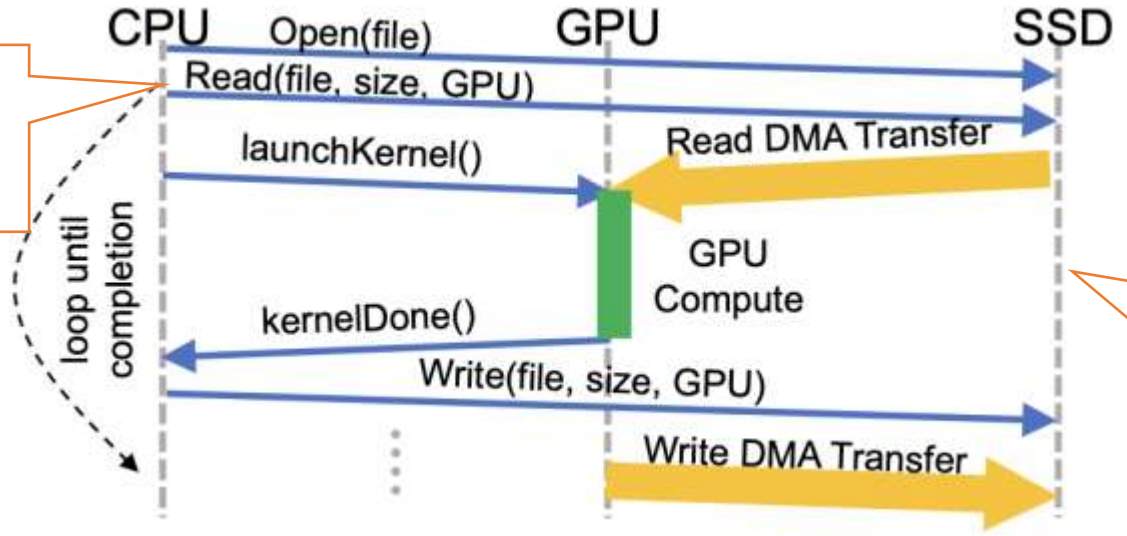
系统整体架构

1.国外企业技术案例：NVIDIA

ASPLOS'23 GPU-Initiated On-Demand High-Throughput Storage Access in the BaM System Architecture

- 提出一种以加速器为中心的系统架构，GPU线程按需访问SSD中的数据（无需CPU参与）
- 实现高吞吐量的细粒度存储访问
- 提供软件定义缓存，用户利用局部性可以更好提升性能

缺点：传统处理器为中心的架构下，需要CPU控制数据分布



缺点：CPU与GPU频繁切换，会引入同步开销

缺点：某些计算任务中，CPU难以预知所需数据，可能预取大量无效数据

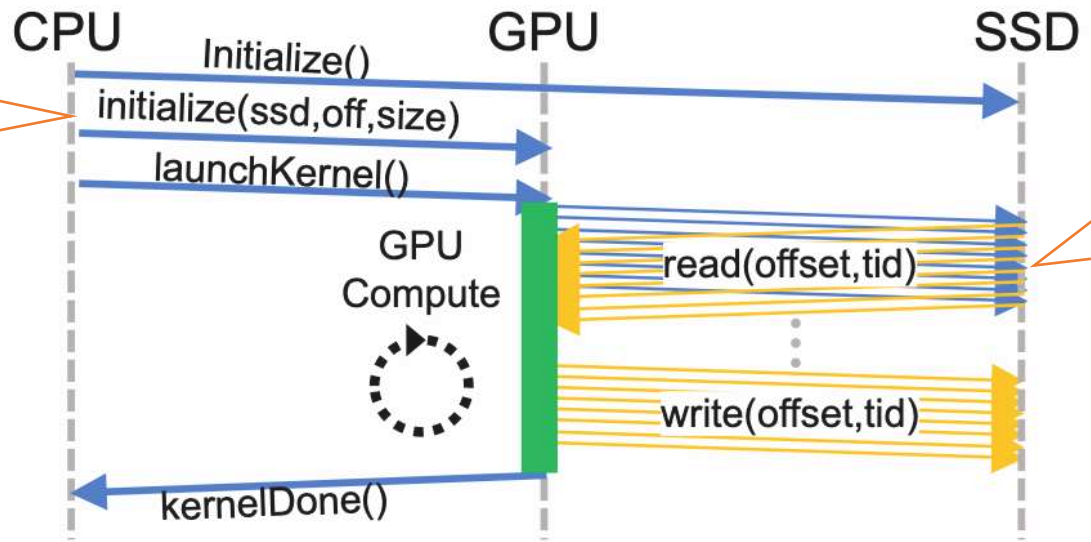
1.国外企业技术案例：NVIDIA

ASPLOS'23

GPU-Initiated On-Demand High-Throughput Storage Access in the BaM System Architecture

- 提出一种以加速器为中心的系统架构，GPU线程按需访问SSD中的数据（无需CPU参与）
- 实现高吞吐量的细粒度存储访问
- 提供软件定义缓存，用户利用局部性可以更好提升性能

优势：CPU只需进行相关初始化工作



优势：仅在计算需要某部分数据时读入，减少无效数据传输

优势：利用GPU的高并发性，自动实现计算与数据传输阶段的重叠

1.国外企业技术案例： Microsoft

ASPLOS'23 | Pond: CXL-Based Memory Pooling Systems for Cloud Platforms

问题背景：

- 公有云的提供商需要平衡性能要求和硬件成本
- 现有的解决方案下，内存成本最多可占据总硬件成本的50%
- 对不同trace进行分析，内存搁浅（即某些资源全部租借但仍存在可用内存）是内存浪费和内存成本升高的重要来源
- 现有的解决方案（如压缩技术）会带来较大延时，无法满足用户要求

解决思路：

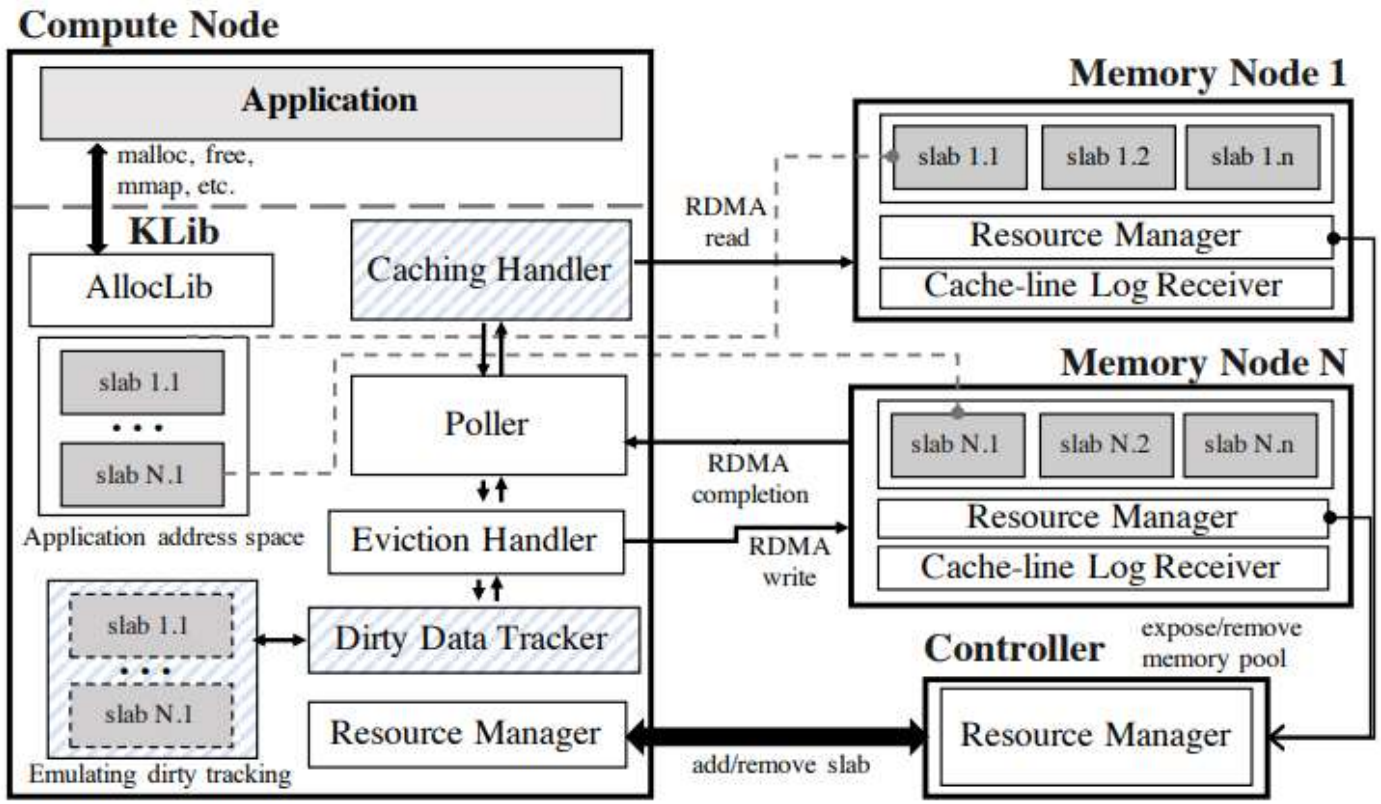
- CXL互连协议可以实现ns级别的内存访问
- 聚合过多socket的内存会增大时延，合理控制聚合socket数量可以达到时延和内存利用率的平衡
- 不同工作负载对于访问池化内存的敏感性不同，通过机器学习模型预测工作负载特性并对非敏感负载分配池化内存
- 引入监控机制，更新预测错误的负载的内存分配

1.国外企业技术案例：Google

ASPLOS'21 | Rethinking Software Runtimes for Disaggregated Memory

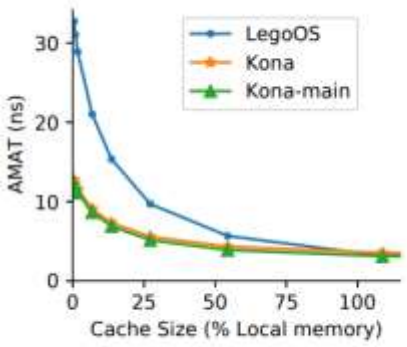
主要贡献：

- 分析了现有分离式内存系统存在较大overhead和脏数据放大的缺点
- 提出了一种新的基于软件的分离式内存方案，利用硬件原语进行远程内存缓存和基于缓存一致性的缓存线脏数据跟踪
- 设计并实现了kona：一个使用新的硬件原语高效运行的软件框架

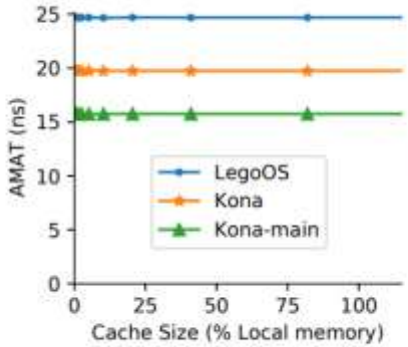


1.国外企业技术案例：Google

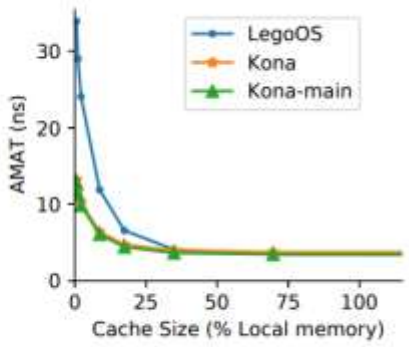
ASPLOS'21 Rethinking Software Runtimes for Disaggregated Memory



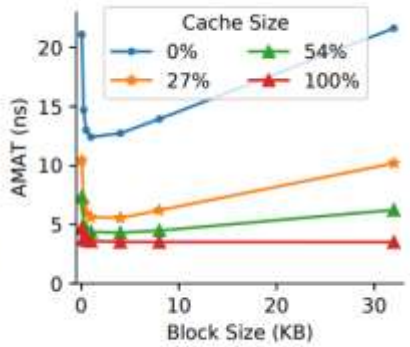
(a) Redis Rand



(b) Linear Regression



(c) Graph Coloring



(d) Redis Rand - Data fetch size

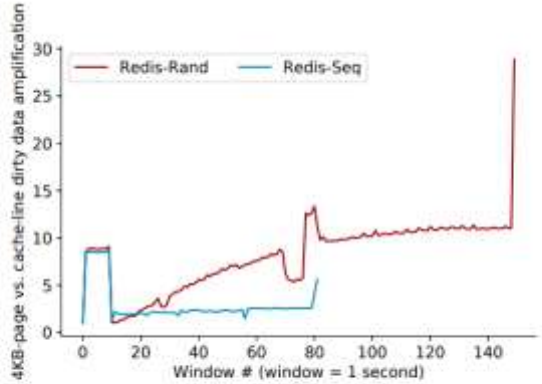


Figure 9: Dirty data amplification reduction.

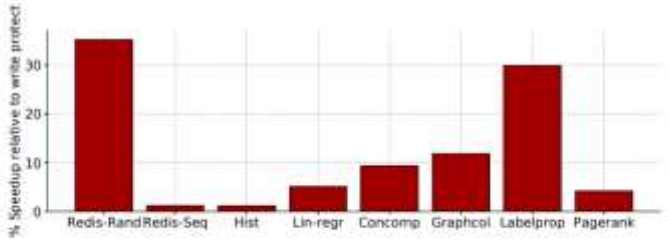


Figure 10: Speedup relative to write-protection.

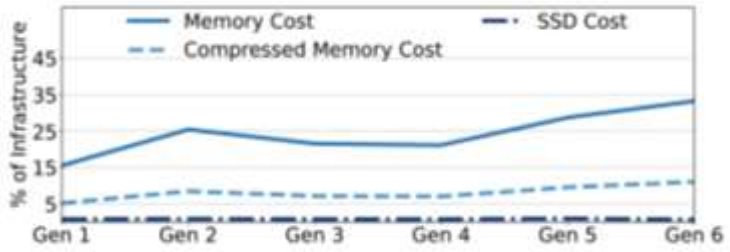
- Kona在远端访问、脏数据卸载上有着更高的效率

1.国外企业技术案例： Facebook

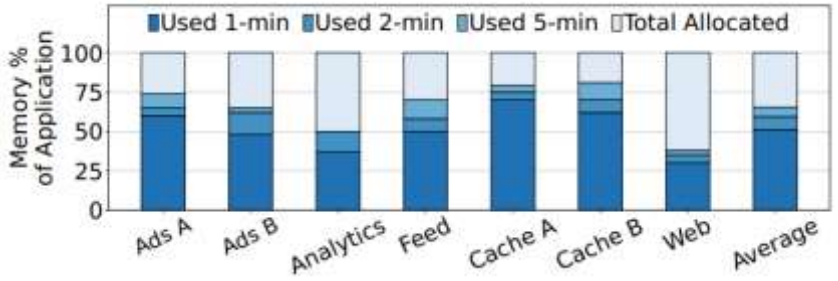
ASPLOS'22 TMO: Transparent Memory Offloading in Datacenters

主要发现：

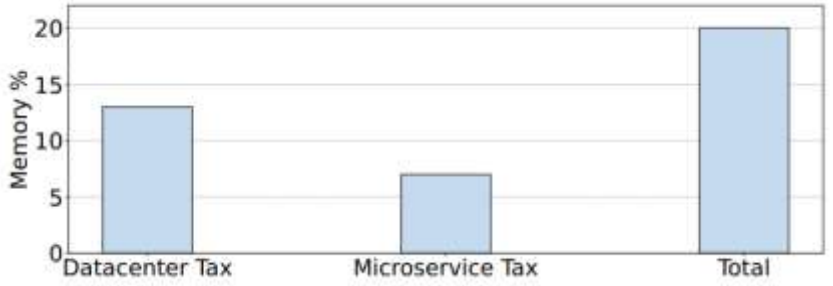
1. 内存成本逐渐升高；



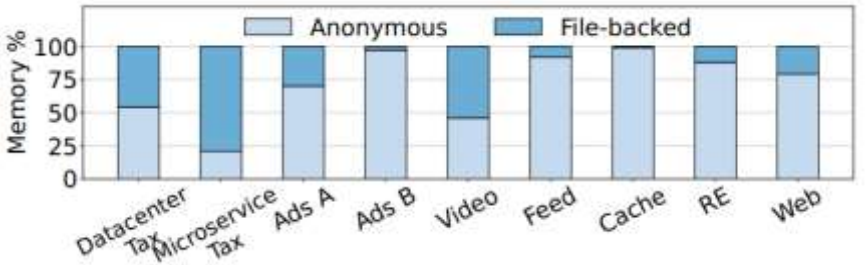
2. 不同应用冷热内存比例差异明显；



3. 内存税占总内存较大比例；



4. 不同应用页面类型差异明显；



主要贡献：

1. 提出PSI指标用以表征应用内存压力；
2. 提出用户态内核态结合的TMO架构；
3. 设计了基于压力的内存卸载算法并上传到Linux kernel

1.国外企业技术案例： Facebook

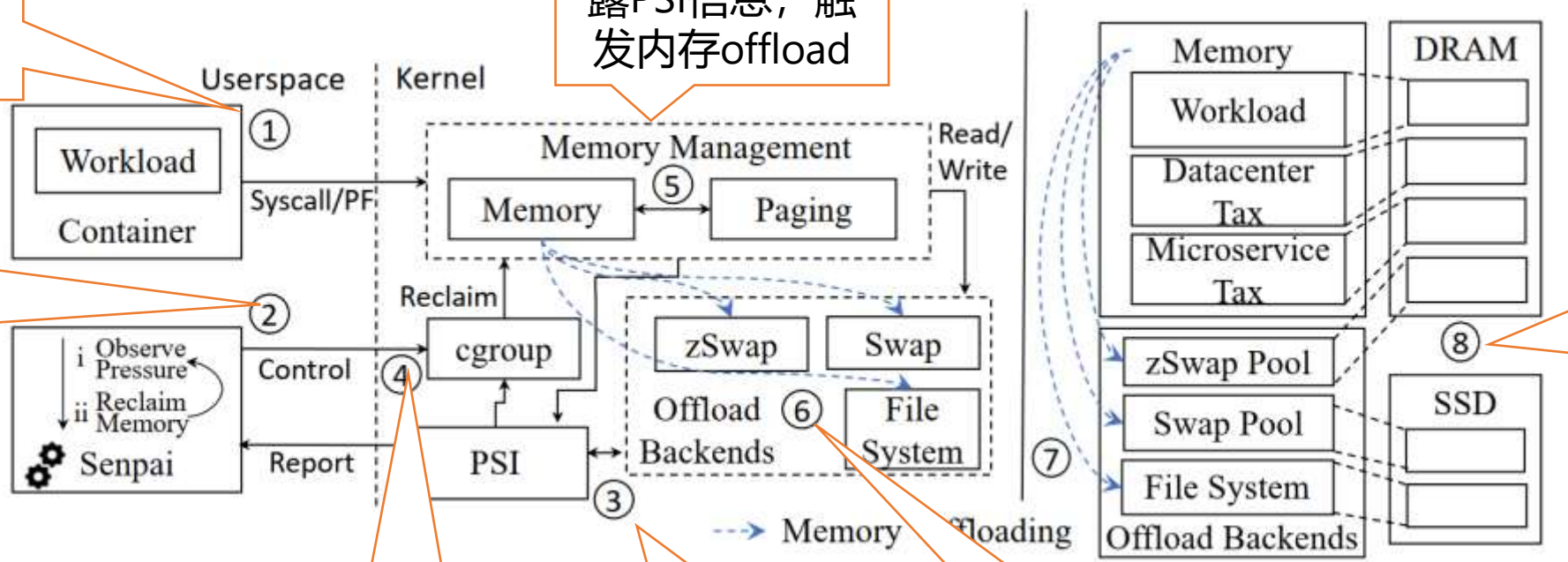
ASPLOS'22

TMO: Transparent Memory Offloading in Datacenters

工作流发生pagefault，通过Syscal处理

内存管理系统披露PSI信息，触发内存offload

监控组件Senpai，监控进程压力信息，控制进程



TMO的内存存储整体概况

通过cgroup调整进程内存，控制内存回收

内核态PSI组件实时计算进程PSI指数

多种offload后端，目前只支持单一后端

1.国外企业技术案例：Samsung

MICRO'22

BEACON: Scalable Near-Data-Processing Accelerators for Genome Analysis near Memory Pool with the CXL Support

主要贡献：

- 基于CXL设计了分离式内存池架构，用于基因分析运算加速
- BEACON的架构使得未经修改的CXL-DIMM 能够有效地按需扩展内存，并消除了通信的性能瓶颈

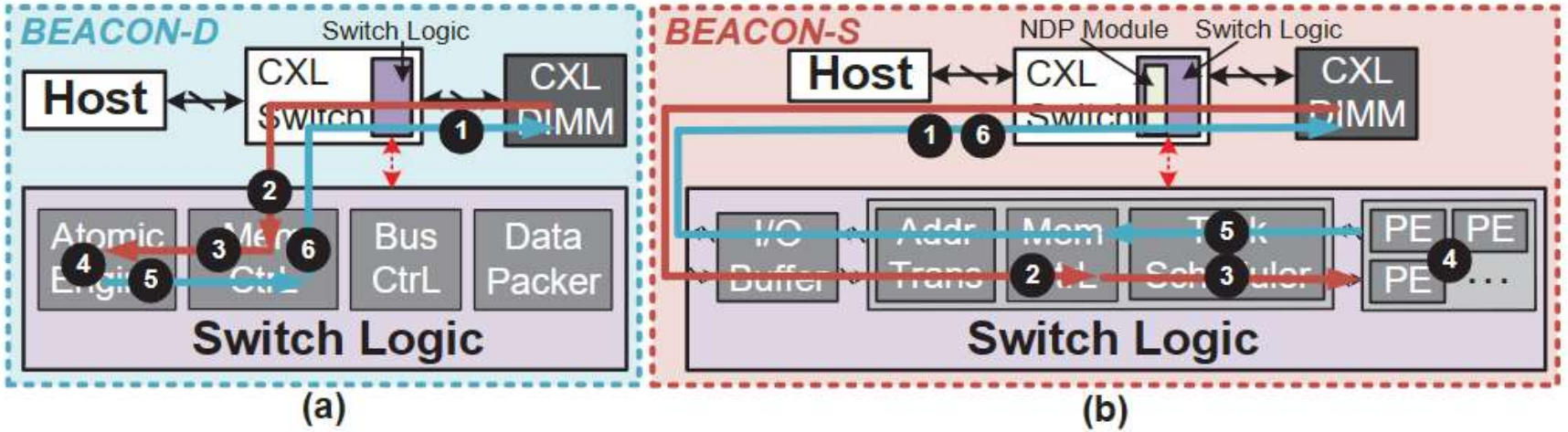


Figure 7. Workflow of performing atomic memory operations. (a) BEACON-D. (b) BEACON-S.

① MC向CXL-DIMM发送内存请求

② 数据带回到MC

③ 数据流向计算单元

④ 数据在单元内完成计算

⑤ 数据流回MC

⑥ MC写回内存池

上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

目录

2.学术界研究分享

上海交通大学 SHANGHAI JIAO TONG UNIVERSITY

硬件架构

大内存节点
- String finger, HPCA'19
- pDPM, ATC'20

可编程网卡
- Cilo,ASPLOS'22

智能交换机
- MIND, SOSP'21

系统设计

非透明远内存
- LITE, SOSP'17
- MemLiner, OSDI'22

透明远内存
- Infiniswap, NSDI'17

混合内存
- Hybrid^2, HPCA'20

应用加速

图计算加速
- Fargraph, IPDPS'22

DL应用加速
- COARSE, HPCA'22

Severless管理
- Jiffy, EuroSys'22

资源管理

资源压力感知
- Canvas, NSDI'23

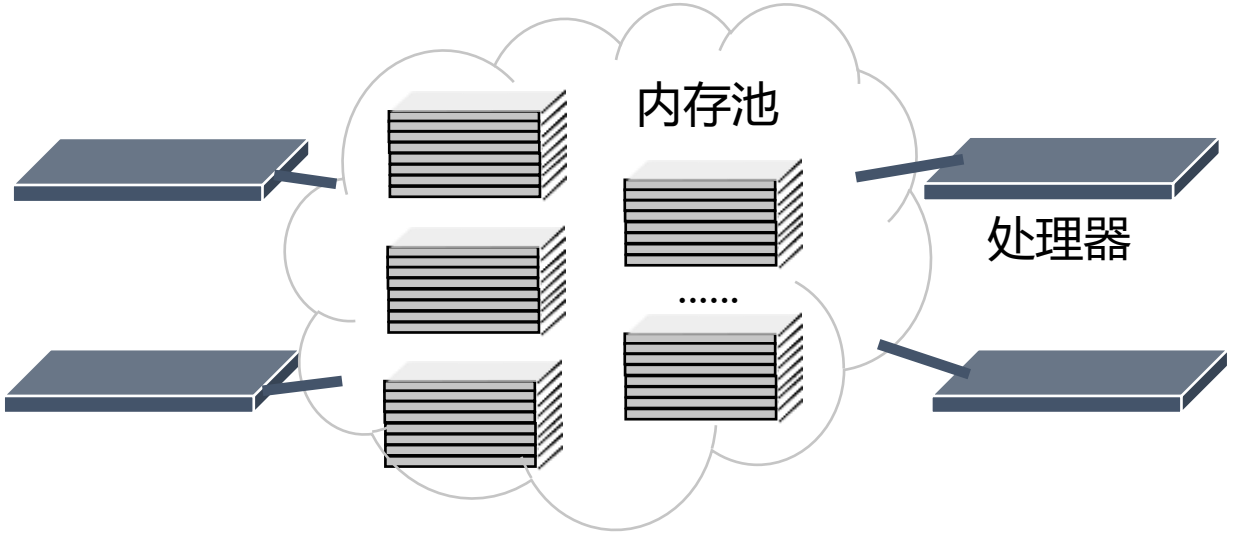
敏感性分析
- HyFarM, ICCD'22

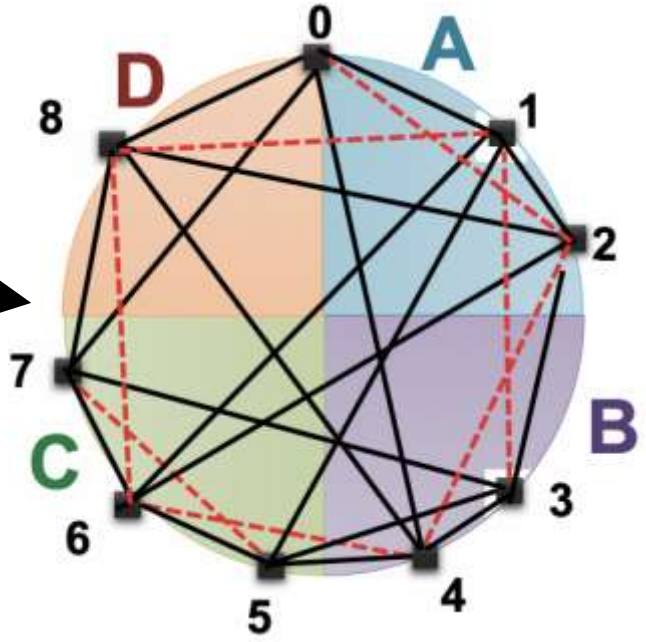
功耗管理
- Zombieland, EuroSys'18

2.学术界研究分享之硬件架构：大内存节点

String Figure:

- 实现了弹性可扩展的内存网络（中心化内存池）
- 提出了随机拓扑算法，以维持任意数量的存储节点的高吞吐量互连
- 提出混合路由协议，保证了路由开销的亚线性增加
- 实验表明，String Figure可以支持至多1296个内存节点的高吞吐互连





9个四端口路由的内存节点互连示意图

[1] String Figure: A Scalable and Elastic Memory Network Architecture, HPCA'19

2.学术界研究分享之硬件架构：大内存节点

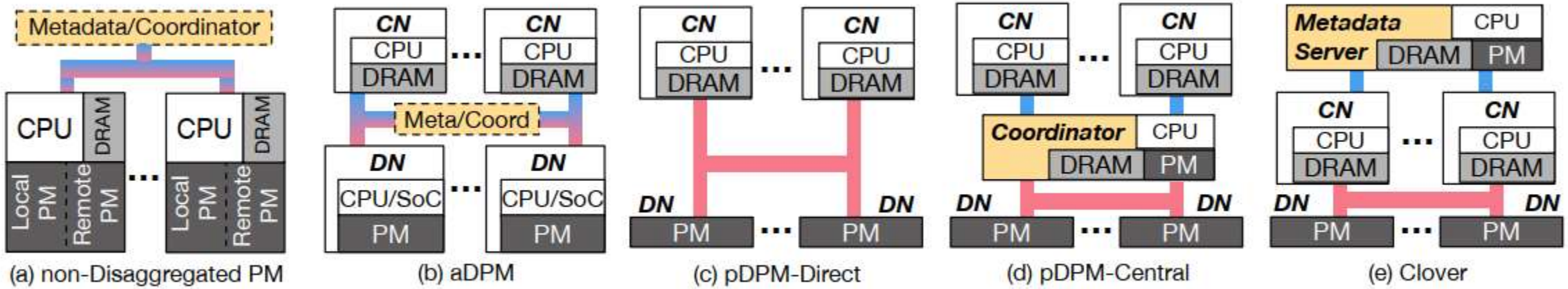


Figure 1: **PM Organization Comparison.** *Blue bars indicate two-way communication and pink ones indicate one-way communication. Bars with both blue and pink mean support for both. Dashed boxes mean some but not all existing solutions adopt centralized metadata server (or a coordinator).*

- 从计算服务器远程管理持久式内存构成的分离式内存节点
- 允许所有计算节点直接访问和管理存储节点，并使用中央协调器来协调计算节点和存储节点之间的通信。同时分离了数据平面和元数据/控制平面的位置、通信机制和管理策略。
- pDPM显著降低了货币和能源成本，并避免了存储服务器上的可伸缩性瓶颈

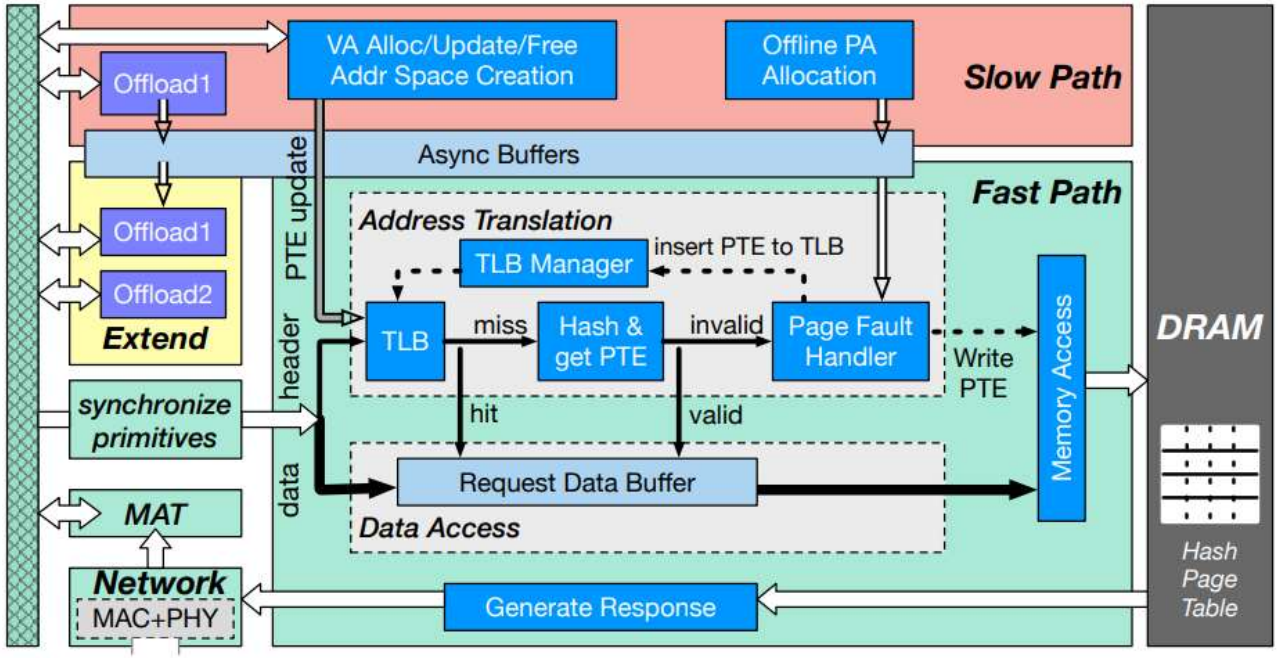
[2] Disaggregating Persistent Memory and Controlling Them Remotely: An Exploration of Passive Disaggregated Key-Value Stores, ATC'2020

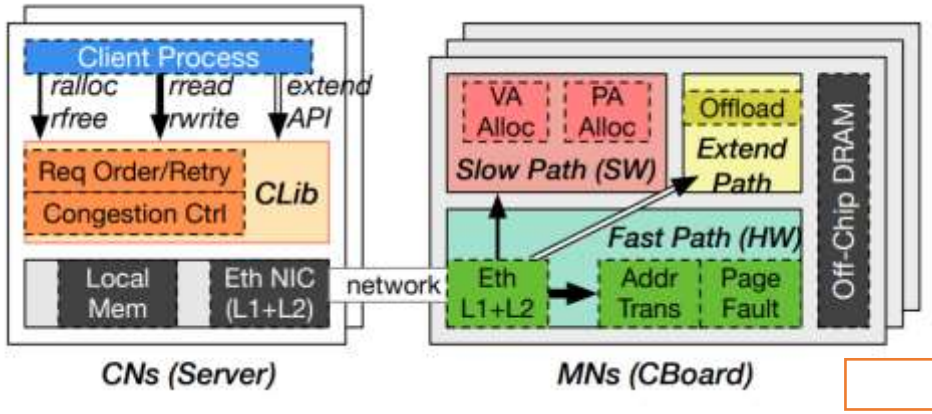
2.学术界研究分享之硬件架构：可编程网卡

Clio：

核心思想：提出一种软硬件协同的分离式内存方案，在内存节点和单纯的内存节点之间取得均衡表现

实现方式：构建了操作系统功能结构，硬件架构，网络系统，计算节点和内存节点

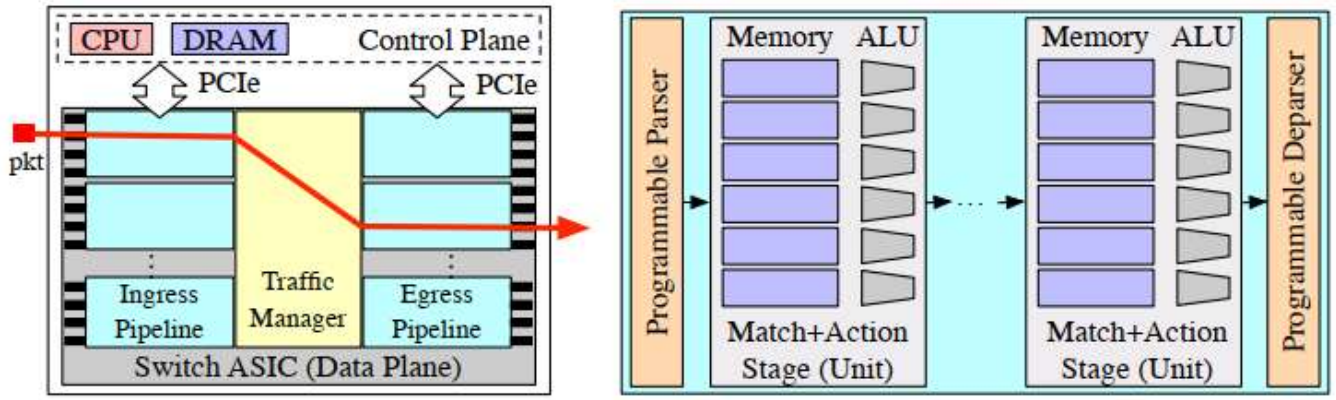




其中NM单元由Cboard实现其功能，Cboard由FPGA实现原型的快速路径，ARM处理元数据和控制的慢路径，FPGA承载内存卸载的计算扩展路径组成

[3] Clio: a hardware-software co-designed disaggregated memory system, ASPLOS'2022

2.学术界研究分享之硬件架构：智能交换机



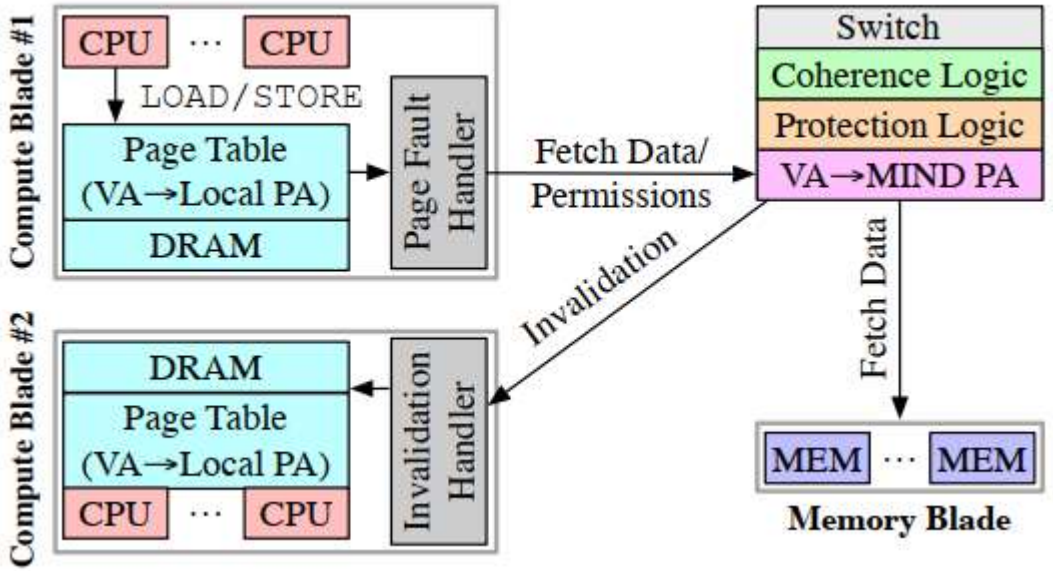


Fig. 1. Enabling technologies for MIND. (left) Programmable switch architecture and (right) Switch ingress/egress pipeline.

- MIND采用所有进程共享的全局虚拟地址空间，使用物理内存分配机制，可以跨内存刀片负载均衡
- 将内存权限的存储与地址转换项分离，实现了细粒度和灵活的保护，同时减少了ASIC切换的开销。
- 利用交换机ASIC中的多播等网络中心硬件原语来有效地实现其一致性协议

[4] MIND: In-Network Memory Management for Disaggregated Data Centers, SOSP'2021

2.学术界研究分享

上海交通大学 SHANGHAI JIAO TONG UNIVERSITY

硬件架构

大内存节点
- String finger, HPCA'19
- pDPM, ATC'20

可编程网卡
- Cilo,ASPLOS'22

智能交换机
- MIND, SOSP'21

系统设计

非透明远内存
- LITE, SOSP'17
- MemLiner, OSDI'22

透明远内存
- Infiniswap, NSDI'17

混合内存
- Hybrid^2, HPCA'20

应用加速

图计算加速
- Fargraph, IPDPS'22

DL应用加速
- COARSE, HPCA'22

Severless管理
- Jiffy, EuroSys'22

资源管理

资源压力感知
- Canvas, NSDI'23

敏感性分析
- HyFarM, ICCD'22

功耗管理
- Zombieland, EuroSys'18

2.学术界研究分享之系统设计：非透明远内存

- 指出在数据中心环境中使用本机RDMA的三个主要问题：
  - RDMA不易使用，它的抽象和数据中心应用程序的需求之间的不匹配。
  - SRAM有限的情况下，RDMA性能基本上很难在MRs, MRs的总大小，qp的总数进行扩展
  - RDMA不提供任何机制来安全地共享资源，如qp、cq、内存缓冲区等

- LITE提出了自己的一套API并解决了，在提升易用性的同时也提升了性能。

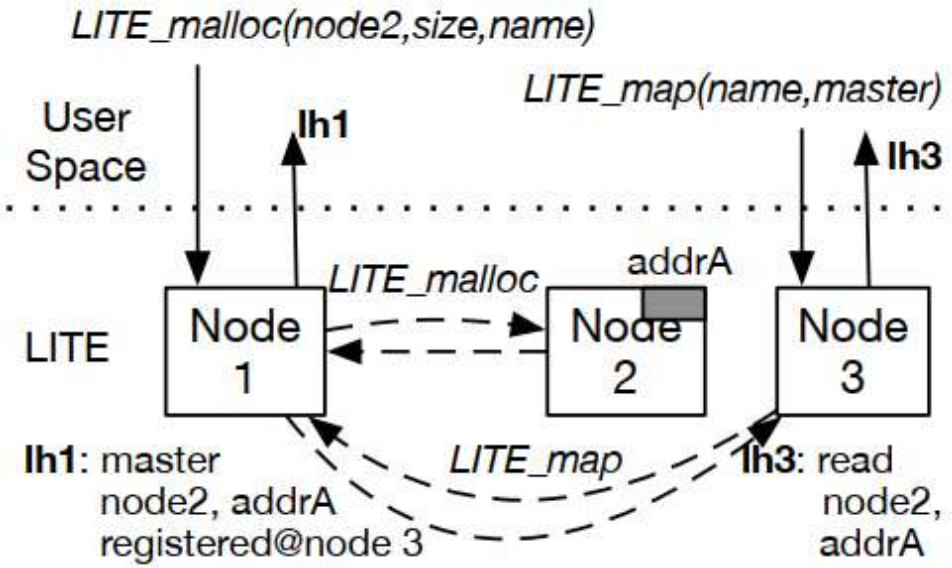


Figure 3: LITE *lh* Example. *Node1 allocates an LMR from node2 and is the master of it. Node3 maps the LMR with read permission by contacting Node1.*

[5] LITE Kernel RDMA Support for Datacenter Applications, SOSP'2017

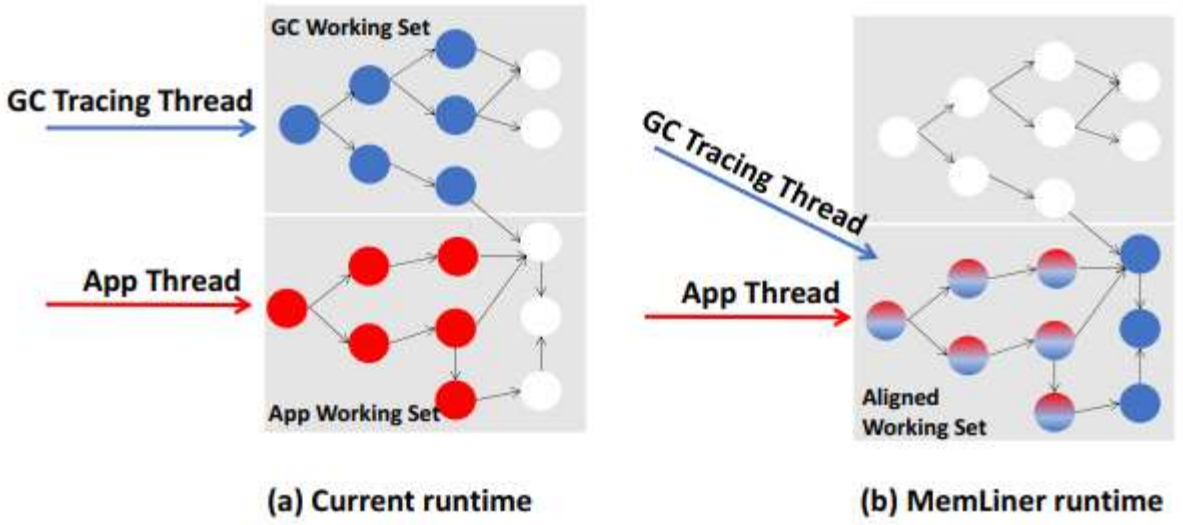
2.学术界研究分享之系统设计：非透明远内存

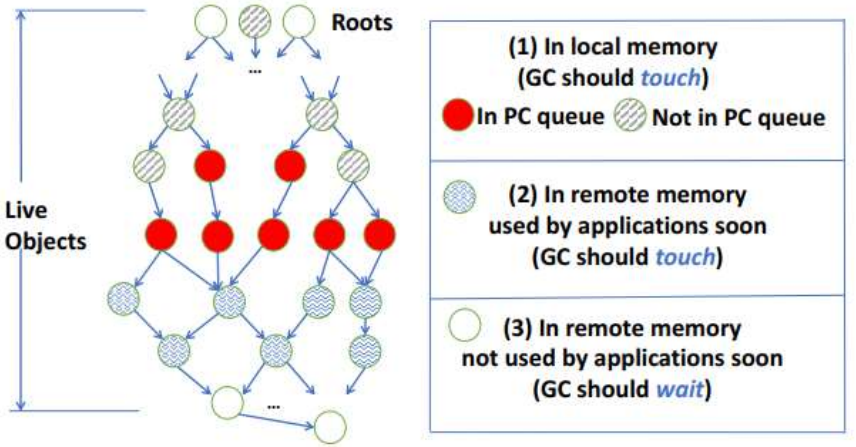
核心思想：

- 设计了基于Java虚拟机的远内存访问运行时优化
- 在Java中垃圾回收GC机制中实现，保留了一定的透明性

实现方式：

- 将来自应用程序和GC的内存访问进行“排列”和分类，使它们遵循相似的内存访问路径，从而减少本地工作及并提升预取性能





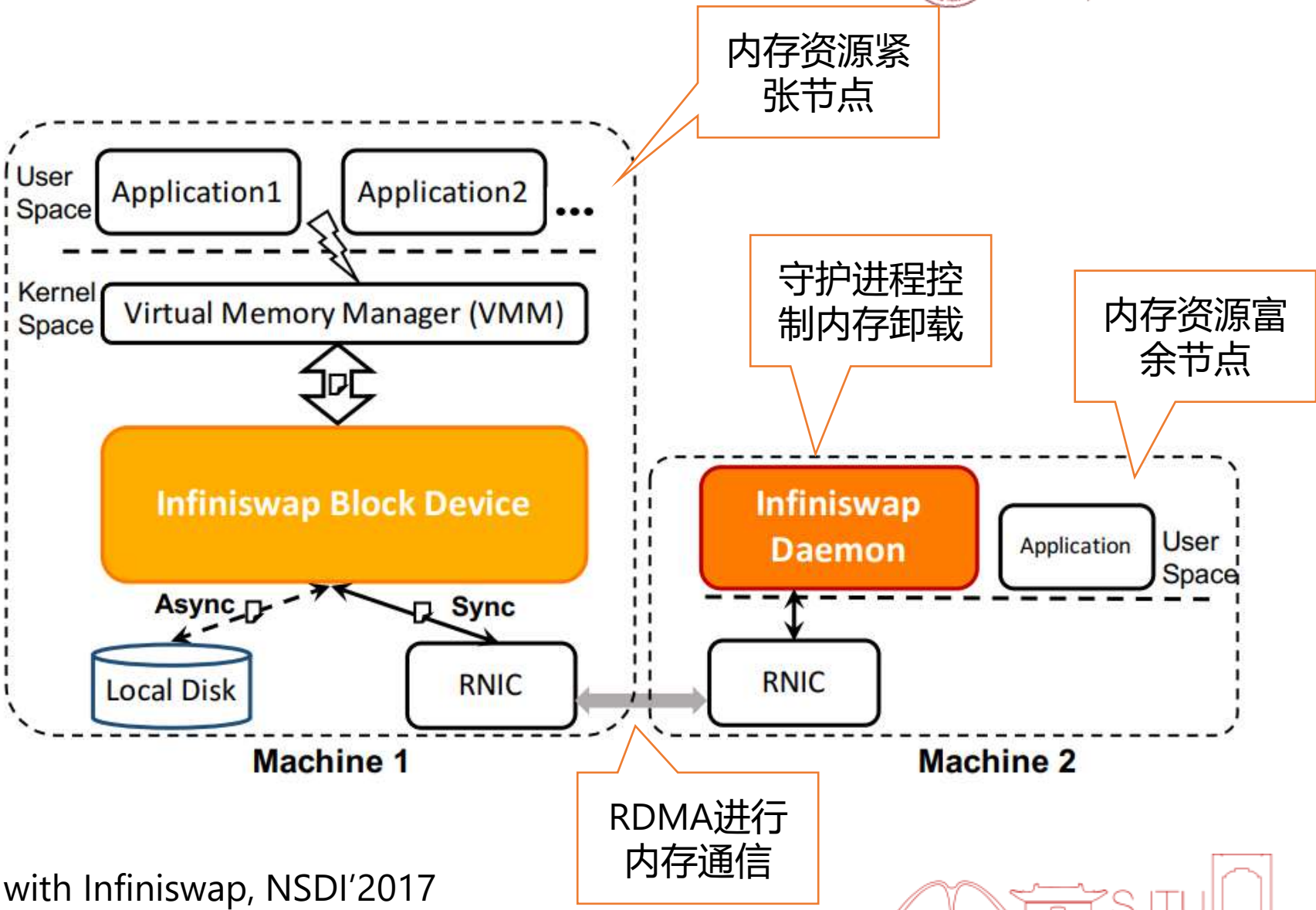
[6] MemLiner: Lining up Tracing and Application for a Far-Memory-Friendly Runtime, OSDI’2022

2.学术界研究分享之系统设计：透明远内存

Inifiniswap：

核心思想：将服务器中的空闲内存资源提供给集群中的其它服务器

实现方式：block device实现地址空间的管理、去中心化的slab分布管理、处理I/O请求、处理内存回收、处理远程中断，INFINISWAP daemon进程，管理内存资源。



[7] Efficient Memory Disaggregation with Infiniswap, NSDI'2017

2.学术界研究分享之系统设计：混合内存

Hybrid2：

核心思想：组织不同具有不同特性的内存，在性能和容量上取得平衡性能

实现方式：使用速度快，容量小，成本高的NM和速度较慢，容量大，成本低的FM组成内存结构，NM的一部分用于缓存，剩余的和FM构成同一地址空间，不损失性能的情况下拥有较大容量

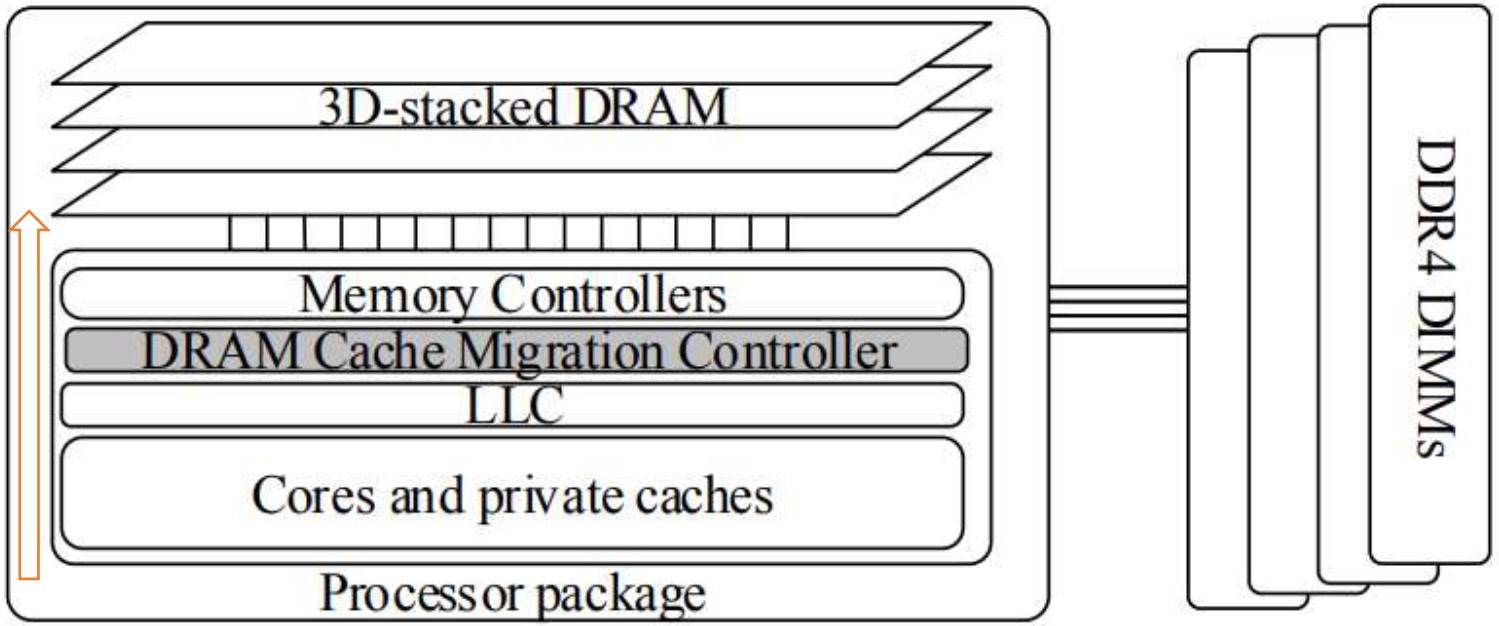


Figure 3: System Overview.

[8] Hybrid2: Combining Caching and Migration in Hybrid Memory Systems, HPCA'2020

2.学术界研究分享

上海交通大学 SHANGHAI JIAO TONG UNIVERSITY

硬件架构

大内存节点
- String finger, HPCA'19
- pDPM, ATC'20

可编程网卡
- Cilo,ASPLOS'22

智能交换机
- MIND, SOSP'21

系统设计

非透明远内存
- LITE, SOSP'17
- MemLiner, OSDI'22

透明远内存
- Infiniswap, NSDI'17

混合内存
- Hybrid^2, HPCA'20

应用加速

图计算加速
- Fargraph, IPDPS'22

DL应用加速
- COARSE, HPCA'22

Severless管理
- Jiffy, EuroSys'22

资源管理

资源压力感知
- Canvas, NSDI'23

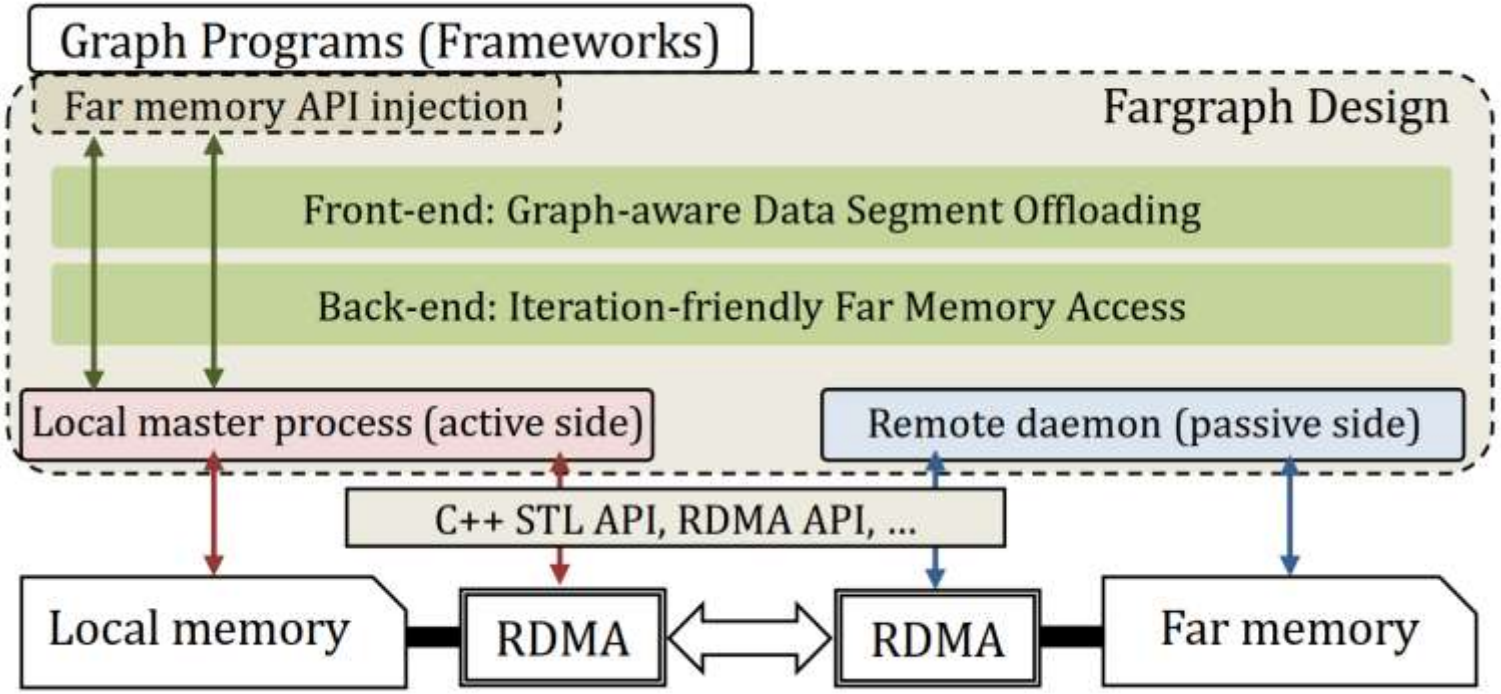
敏感性分析
- HyFarM, ICCD'22

功耗管理
- Zombieland, EuroSys'18

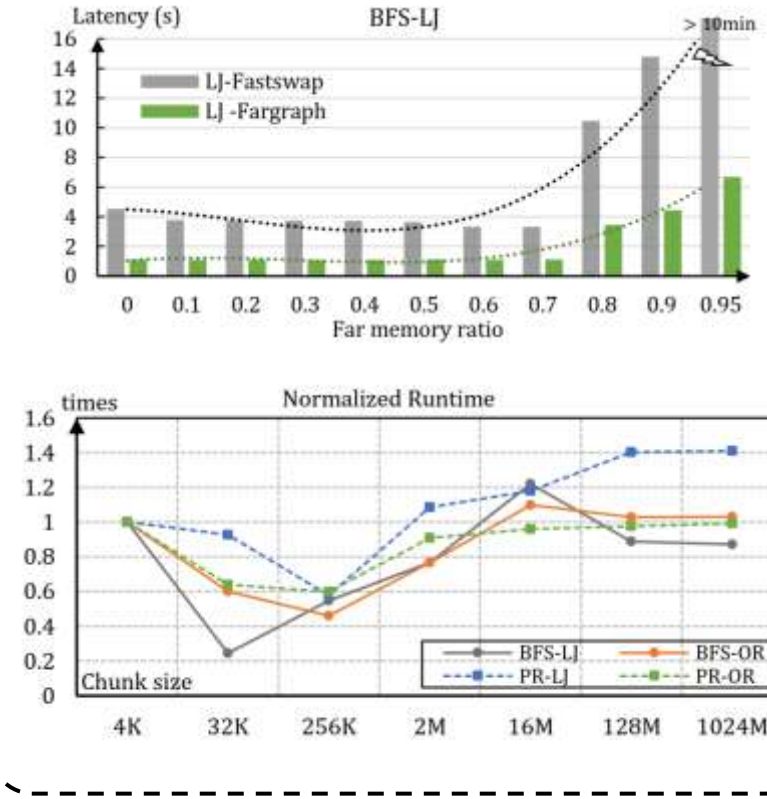
2.学术界研究分享之应用加速：图计算加速

主要贡献：

- 第一个针对图计算任务的远内存系统
- 提出图感知的memory offloading策略
- 利用计算与传输的overlap提高远内存访问性能



取得的性能提升



[9] Excavating the Potential of Graph Workload on RDMA-based Far Memory Architecture, IPDPS'2022

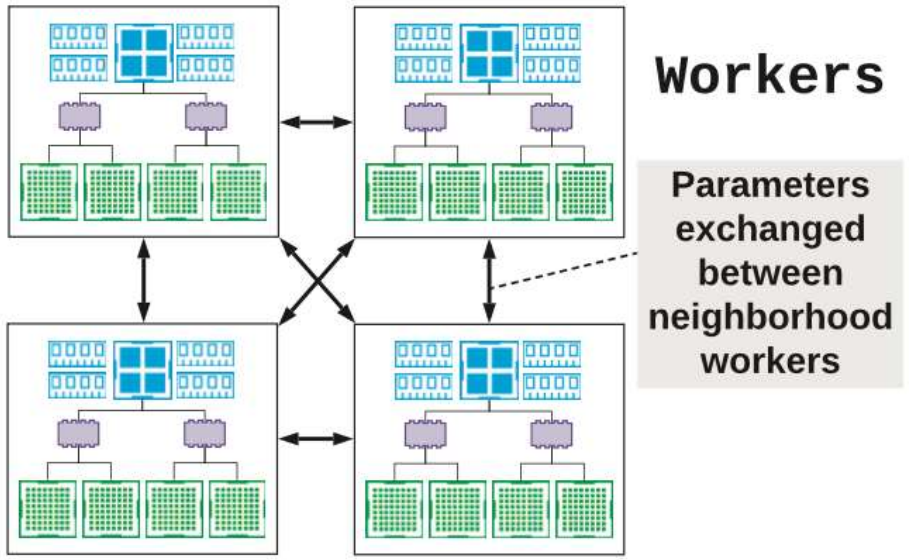
2.学术界研究分享之应用加速：DL应用加速

问题背景：

- 现有分布式DL训练受到 (1)单机并行度和片上内存容量 (2)跨节点通信开销 的限制
- 跨设备通信协议（如PCIe协议）带宽比设备内存带宽低一个量级，过度增加并行度会增加通信瓶颈

本文贡献：

- 使用缓存一致互连CCI协议，通过分离式内存系统加速DL训练
- 提出分散式参数通信方案，将参数同步和本地化参数存储分离
- 提出张量路由和分区方案，以充分利用串行总线带宽



分布式DL训练通信示意图

[10] Enabling Efficient Large-Scale Deep Learning Training with Cache Coherent Disaggregated Memory Systems, HPCA'2022

2.学术界研究分享之应用加速：Serverless管理

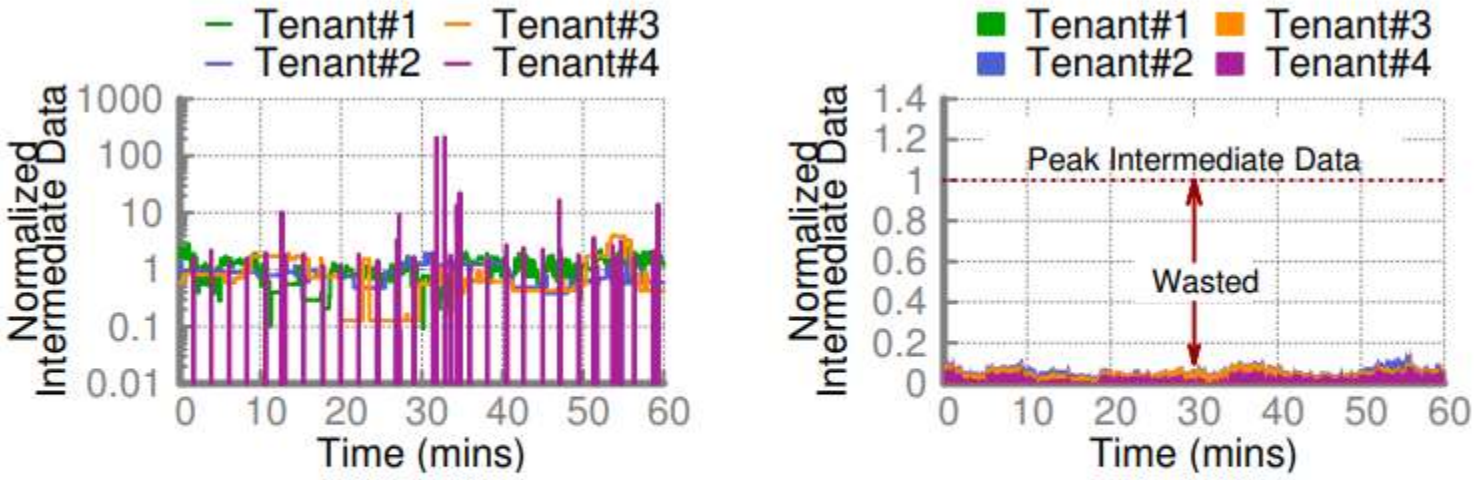
创新点：

- 小内存块的粒度上执行资源分配
- 弹性资源分配
- 对中间数据进行显式生命周期管理

效果：

在并发运行的作业之间复用内存容量，减少了读取和写入较慢的持久存储的开销，任务执行时间提高了1.6–2.5倍。

无服务计算任务有大量数据卸载的潜力



(a) Intermediate data (normalized by mean usage)

(b) Cumulative intermediate data (normalized by peak usage)

[11] Jiffy: Elastic Far-Memory for Stateful Serverless Analytics, EuroSys'2022

2.学术界研究分享

上海交通大学 SHANGHAI JIAO TONG UNIVERSITY

硬件架构

大内存节点
- String finger, HPCA'19
- pDPM, ATC'20

可编程网卡
- Cilo,ASPLOS'22

智能交换机
- MIND, SOSP'21

系统设计

非透明远内存
- LITE, SOSP'17
- MemLiner, OSDI'22

透明远内存
- Infiniswap, NSDI'17

混合内存
- Hybrid^2, HPCA'20

应用加速

图计算加速
- Fargraph, IPDPS'22

DL应用加速
- COARSE, HPCA'22

Severless管理
- Jiffy, EuroSys'22

资源管理

资源压力感知
- Canvas, NSDI'23

敏感性分析
- HyFarM, ICCD'22

功耗管理
- Zombieland, EuroSys'18

2.学术界研究分享之资源管理：资源压力感知

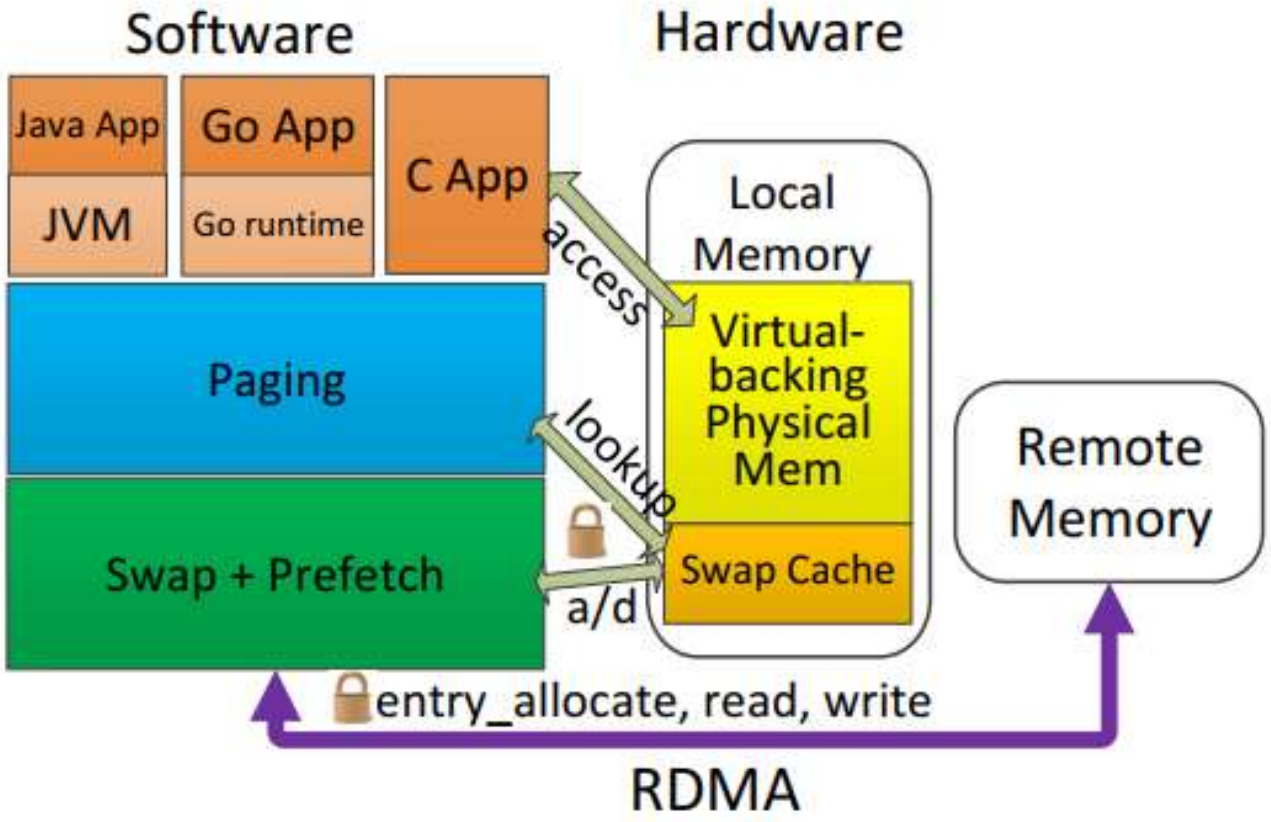
研究内容

创新点：

- 研究远内存在swap区的争抢，隔离了远程内存应用程序的交换路径。
- 允许每个应用程序拥有其专用的交换分区、交换缓存，根据资源限制设计预取器和分配RDMA带宽

实验结果：

- 可以将最先进的固结技术的能源效率提高86%，而额外的复杂性最小。

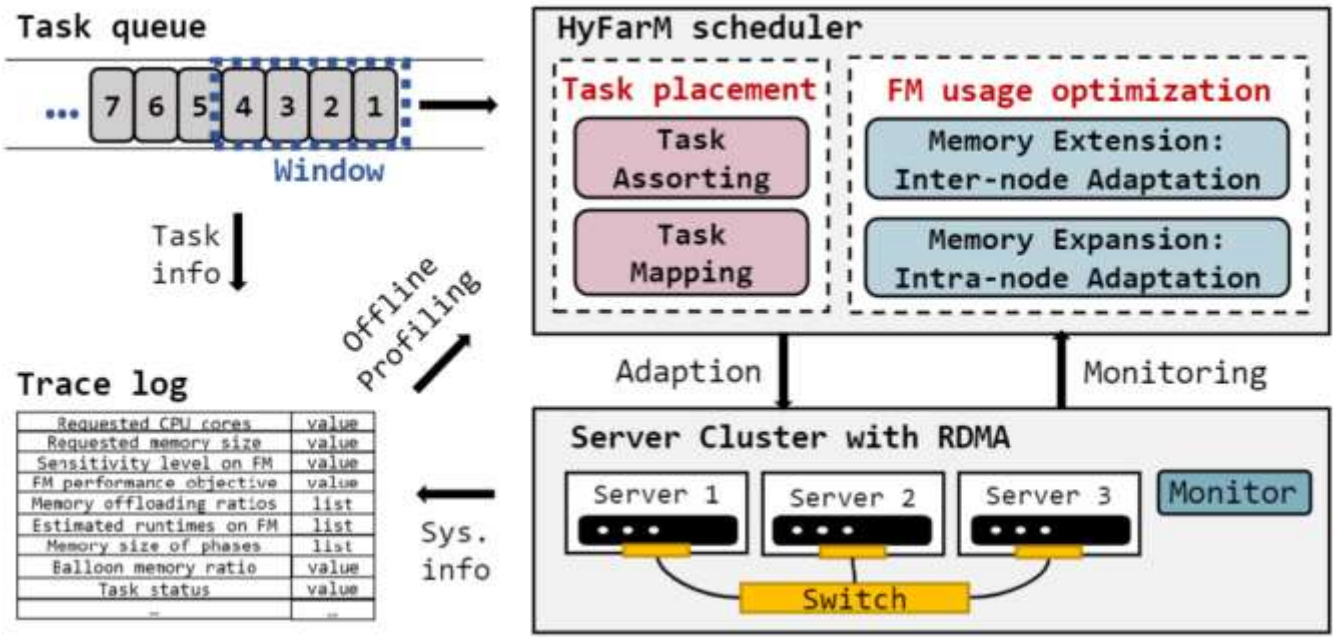


[12] Canvas: Isolated and Adaptive Swapping for Multi-Applications on Remote Memory, NSDI'23

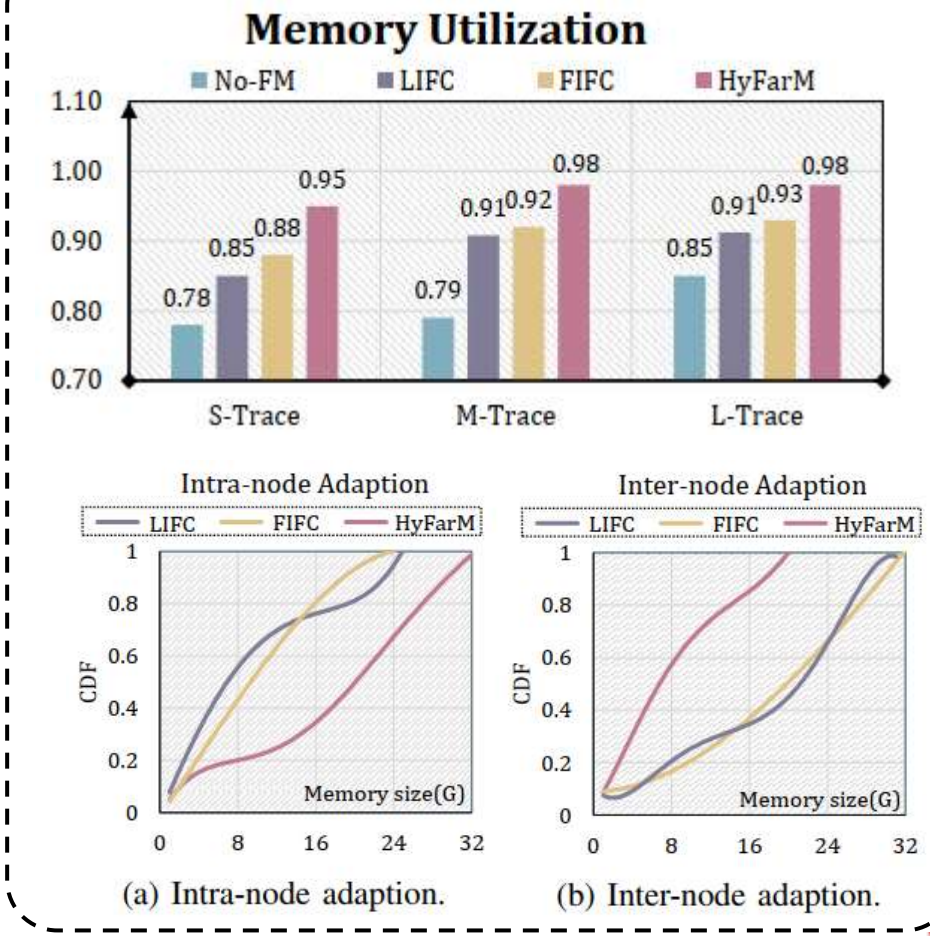
2.学术界研究分享之资源管理：敏感型分析

主要贡献：

- 第一个基于混合远内存架构的任务调度框架
- 设计了不同远内存敏感性的应用混部部署策略
- 动态控制并调整横纵远内存的使用



取得的效率提升



[13] HyFarM: Task Orchestration on Hybrid Far Memory for High Performance Per Bit, ICCD'2022

2.学术界研究分享之资源管理：功耗管理

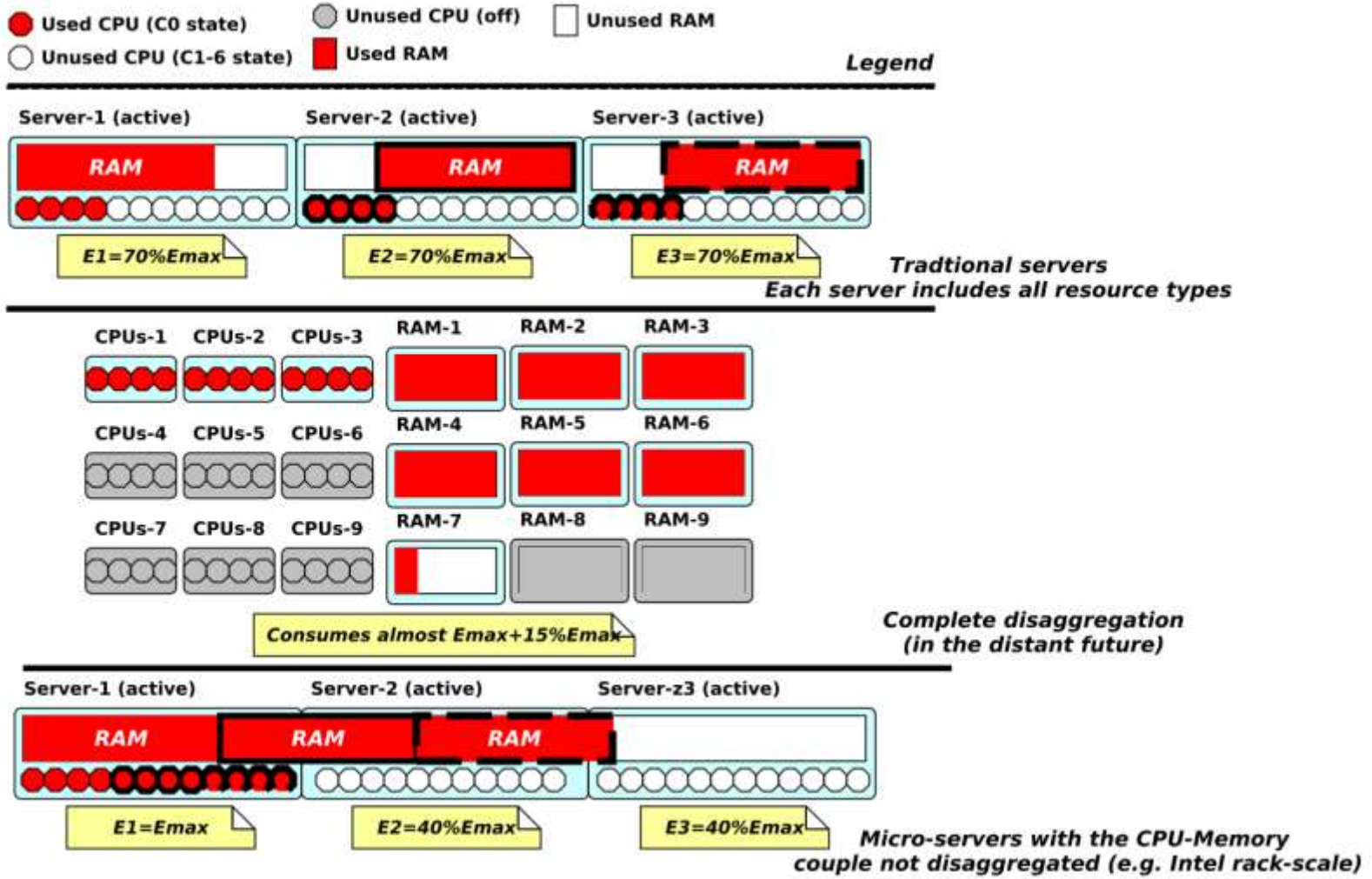
研究内容

创新点：

- 设计僵尸节点，允许暂停服务器从而节省能源)，同时使其内存远程访问
- 允许整个机架资源的透明利用(避免资源浪费)。

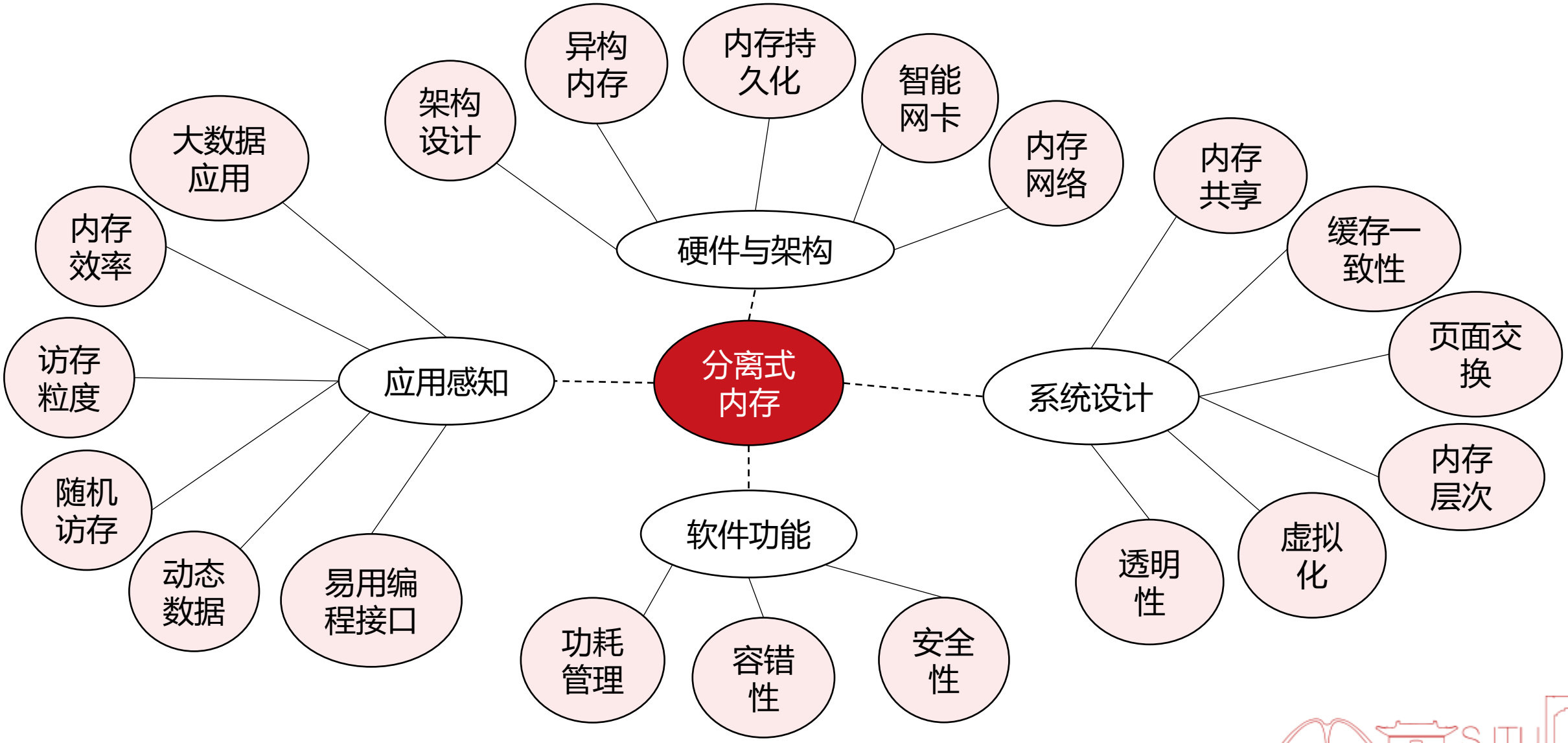
实验结果：

- 可以将最先进的固结技术的能源效率提高86%，而额外的复杂性最小。



[14] Welcome to zombieland: Practical and Energy-efficient memory disaggregation in a datacenter, Eurosys'2018

总结：分离式内存相关研究



上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

Thank You