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

北京开源芯片研究院 BEIJING INSTITUTE OF OPEN SOURCE CHIP

香山开源处理器社区 XiangShan Open-Source Processor Community



香山处理器昆明湖 V3 架构访存子系统的设计演进

杨泽辰[1] 燕翼鸣[1]

陈卓[1] 丁昊楠[1] 冯浩原[1] 付华凯[1] 姜龙[2] 李会锦[2] 李昕[2] 李燕琴[4] 刘威丁[2]

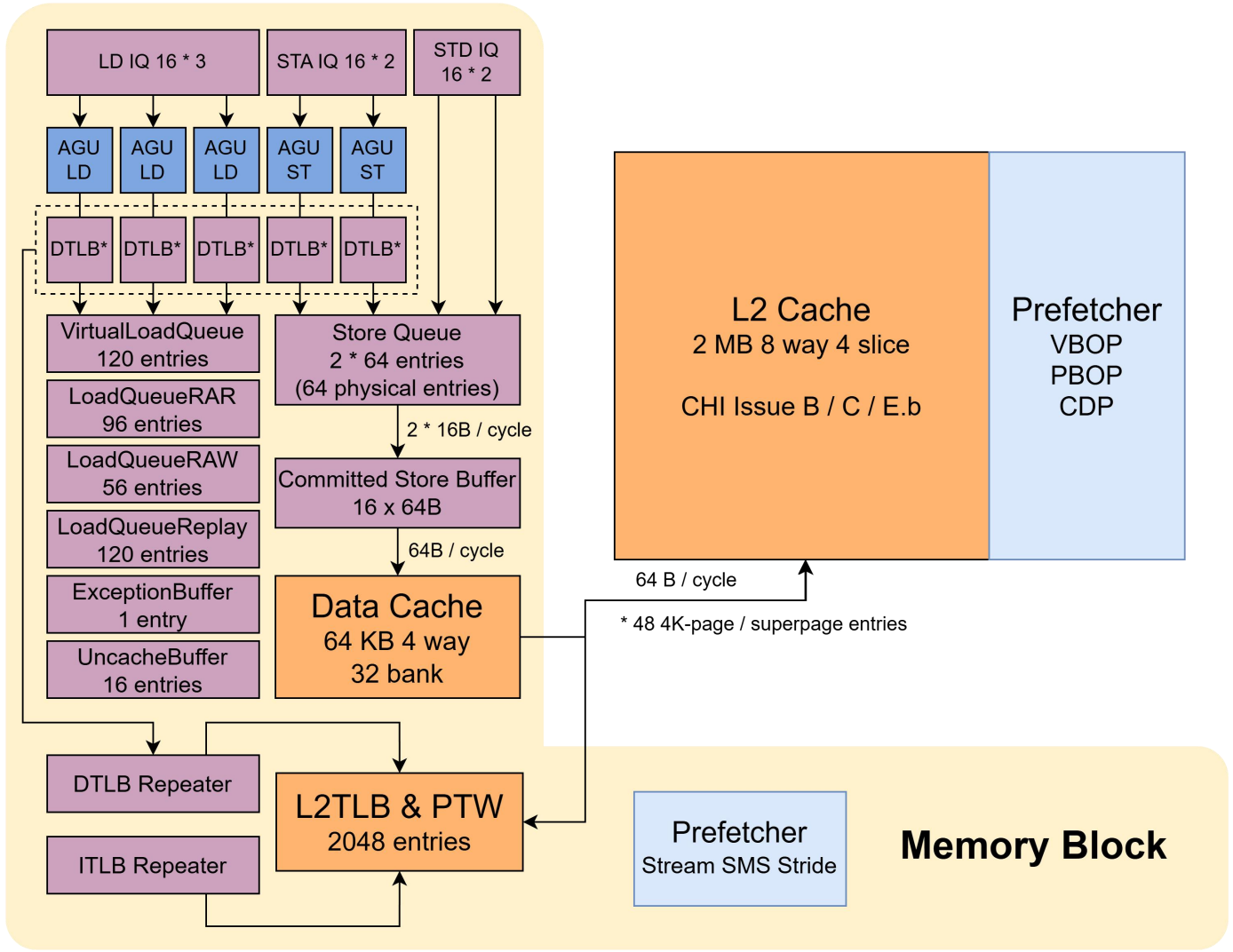
卢非凡[3] 马月骁[1] 彭培智[1] 王植鑫[3] 王郅尊[1] 徐泽凡[1] 于文龙[3] 张林隽[2] 郑忠强[3] 朱昱[2]

[1]中国科学院计算技术研究所 [2]北京开源芯片研究院 [3]上海科技大学 [4]香港科技大学

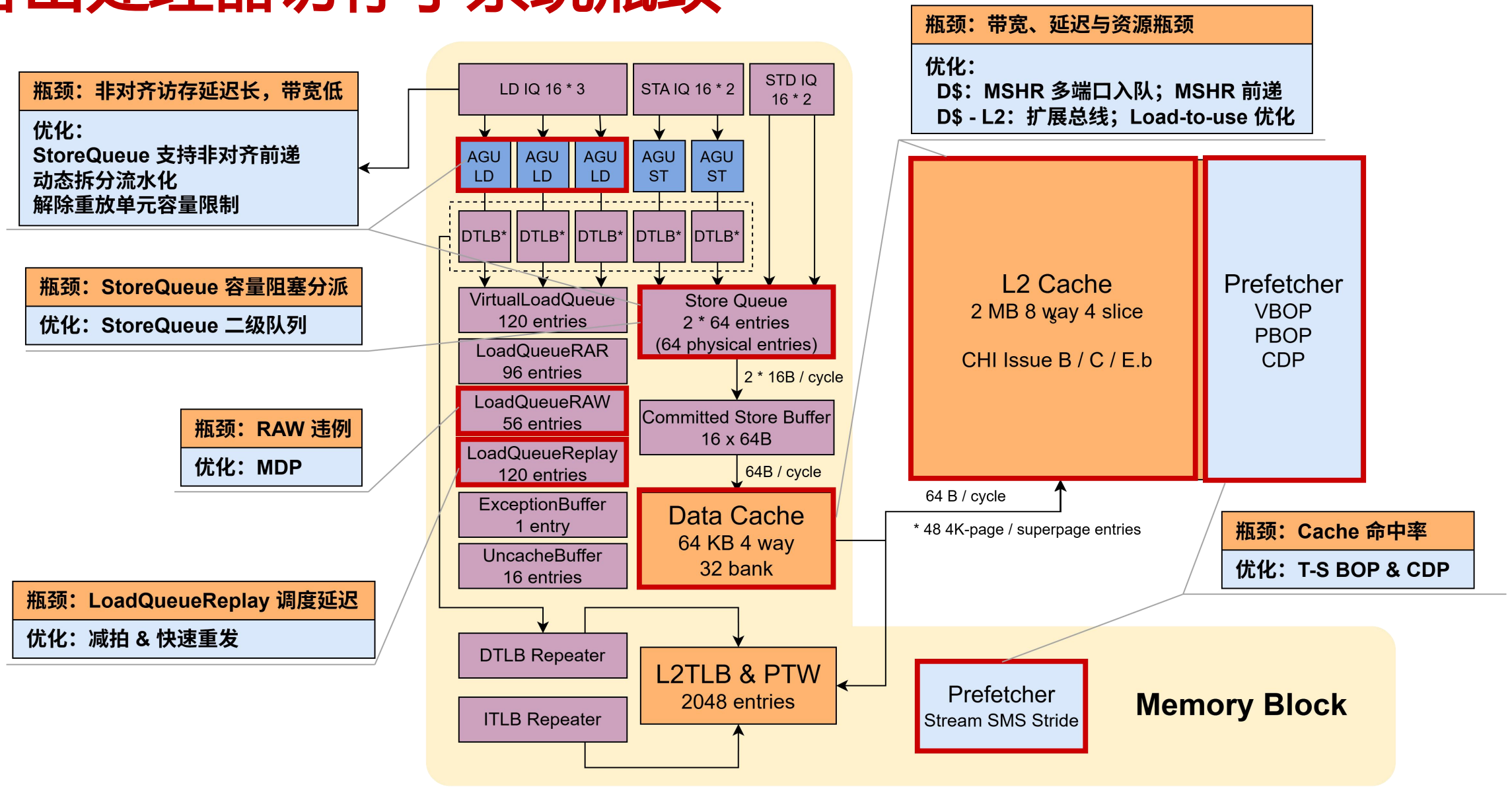
2026 年 10 月 18 日

昆明湖 V3 架构访存子系统

- 相较于昆明湖 V2R2 的变化
  - 更大的配置
  - 带宽与 MSHR 资源优化设计
  - StoreQueue 拆分优化
  - LoadQueueReplay 调优
  - MDP 调优
  - 非对齐访存调优
  - 预取调优
- CHI 工具集



香山处理器访存子系统瓶颈



访存子系统配置演进

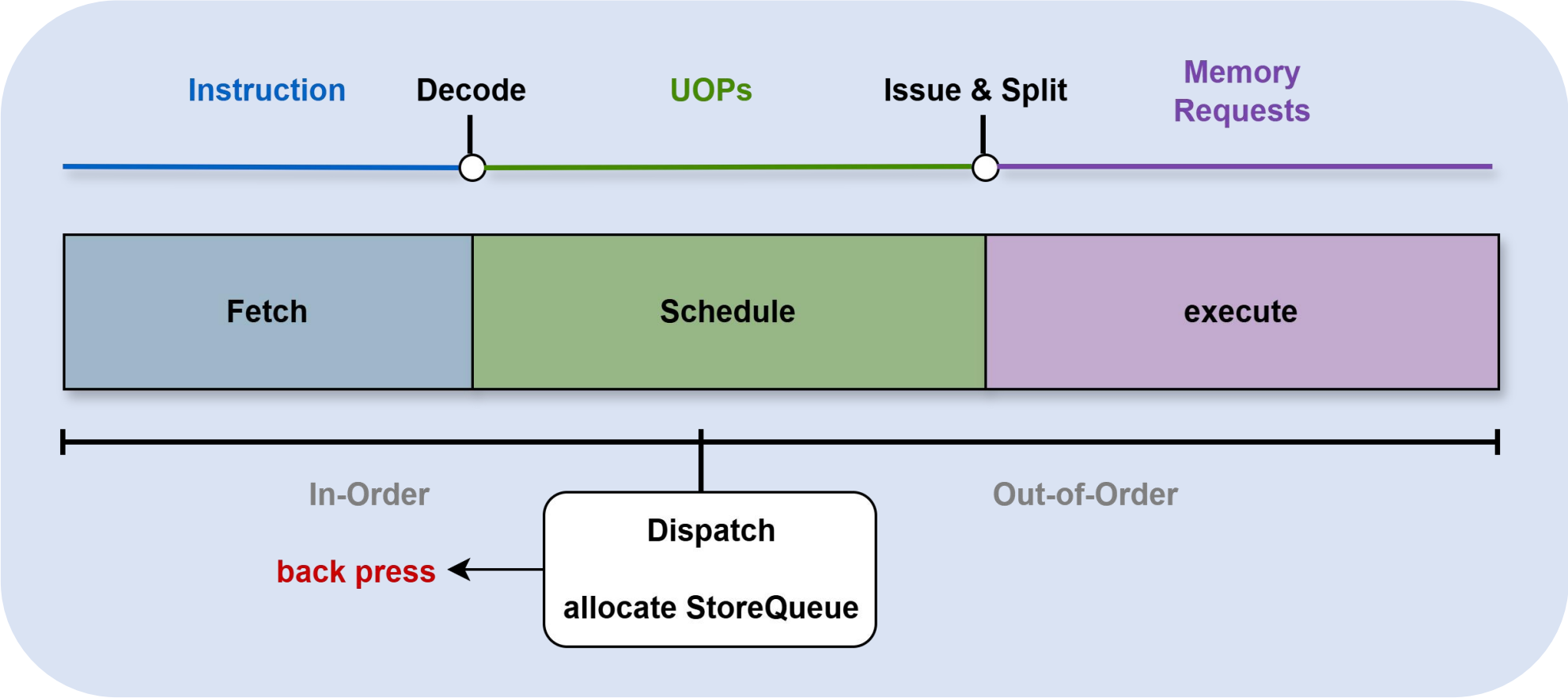
Fetures	雁栖湖	南湖	昆明湖 V2R2	昆明湖 V3
AGUs	2-LD, 2-ST	2-LD, 2-STA	3-LD, 2-STA	3-LD, 2-STA
LQ entries	64	80	72 (splited)	**120 (splited)**
SQ entries	48	64	56	**2 * 64 (64 physical)**
L1 Data Cache	32KB 8-way	128KB 8-way 1-way 8B per bank	64KB 4-way 1-way 8B per bank	64KB 4-way **4-way 2B per bank**
L1 MSHR	1 alloc 1 refill	1 alloc 1 refill	1 alloc 1 refill	**4 alloc 2 refill**
MDP	Wait Table	Store Sets	None	**Optimized Store Sets**
L2 Cache		1MB 8-way 4-bank Support TileLink	1MB 8-way 4-bank Support TileLink, CHI Issue B / E.b	**2MB 8-way 4-bank** **Support CHI Issue B / C / E.b**
Prefetcher	BOP	BOP	Stream & Stride, SMS, BOP	**Optimized Stream & Stride, VBOP & PBOP, CDP**
LLC		6MB	16MB	**32MB**

第一部分：访存流水线

- 虚拟 StoreQueue 与 Store 预分配
- Load Replay 延迟优化
- 内存依赖预测器（MDP）优化
- 非对齐访存动态拆分与前递

虚拟 StoreQueue：背景

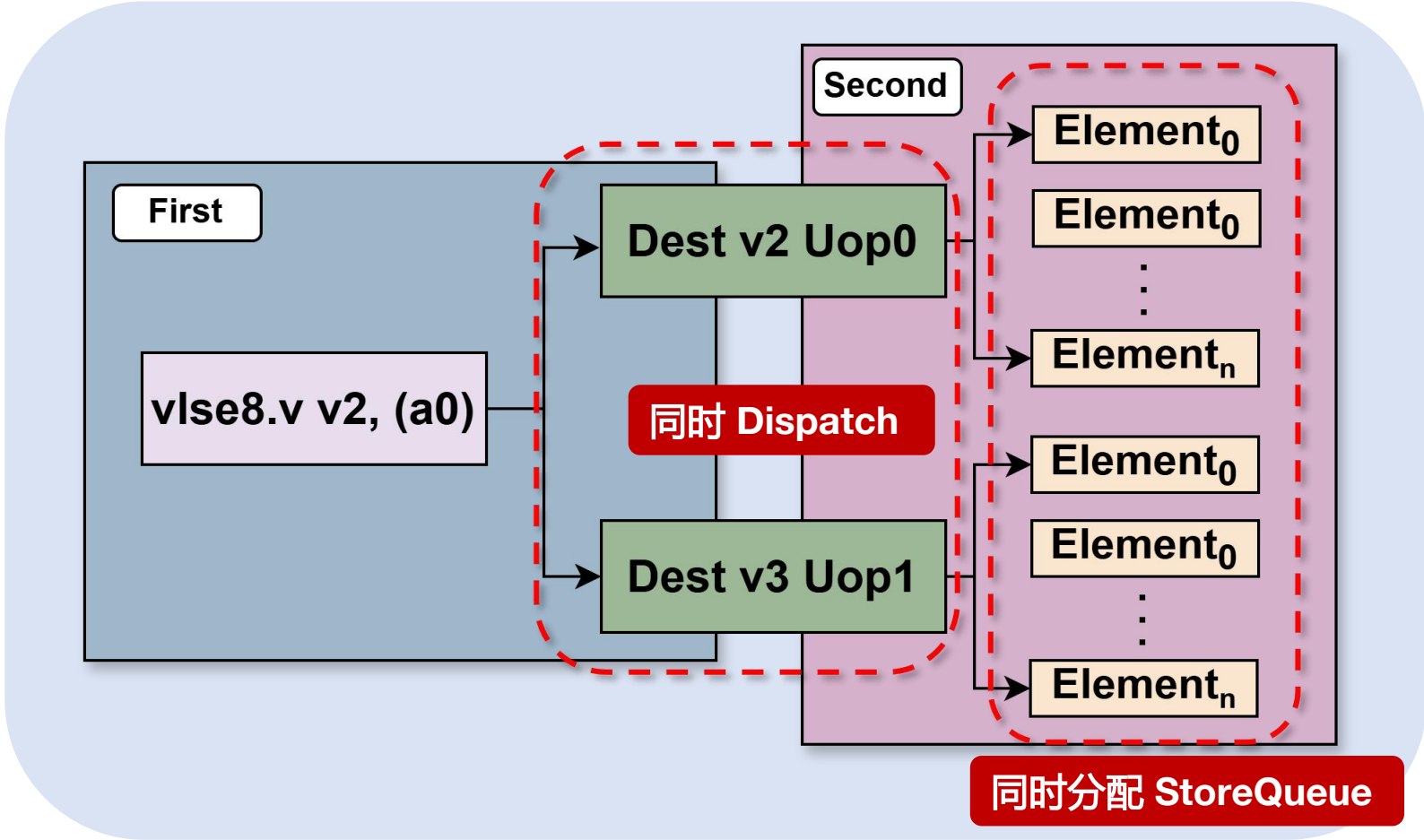
阻塞指令分派：StoreQueue 容量成为乱序调度瓶颈



虚拟 StoreQueue：背景

阻塞指令分派：StoreQueue 容量成为乱序调度瓶颈

向量加剧容量瓶颈：单 UOP 最多拆分 16 项，单个周期需要分配上百个表项

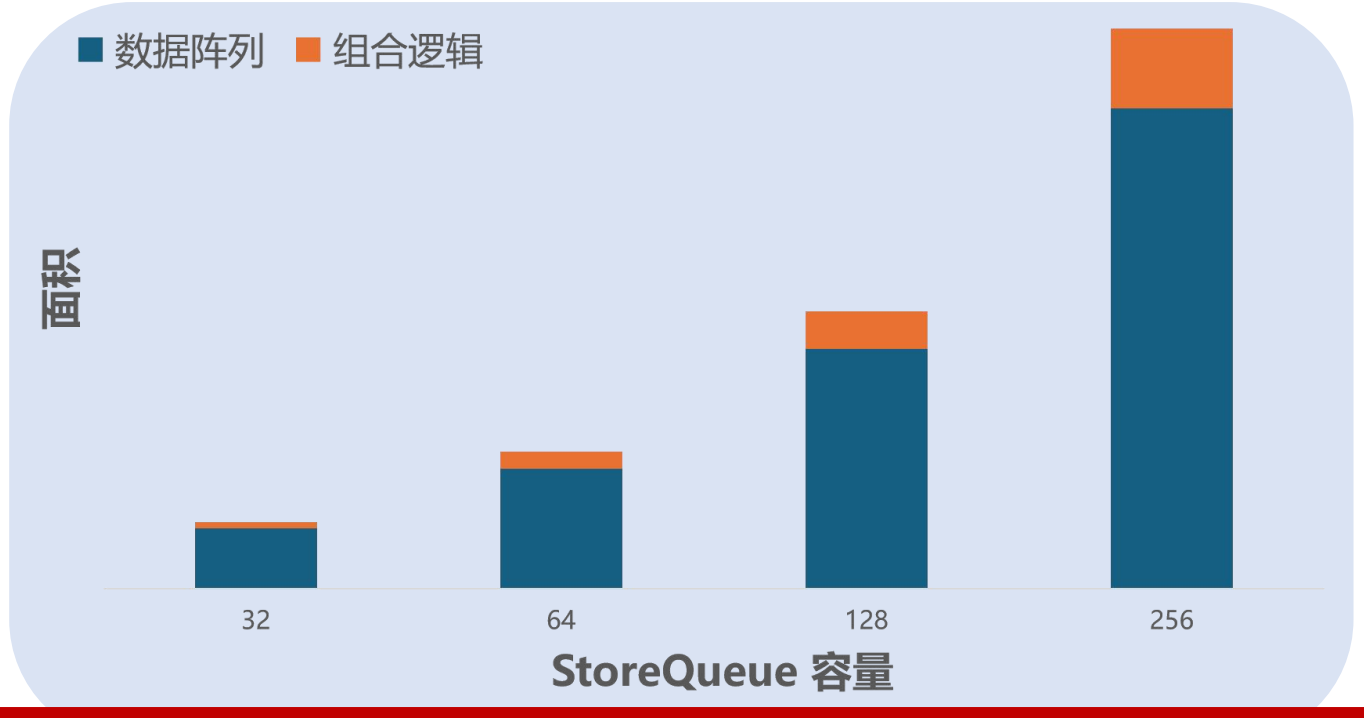


虚拟 StoreQueue：背景

阻塞指令分派：StoreQueue 容量成为乱序调度瓶颈

向量加剧容量瓶颈：单 UOP 最多拆分 16 项，单个周期需要分配上百个表项

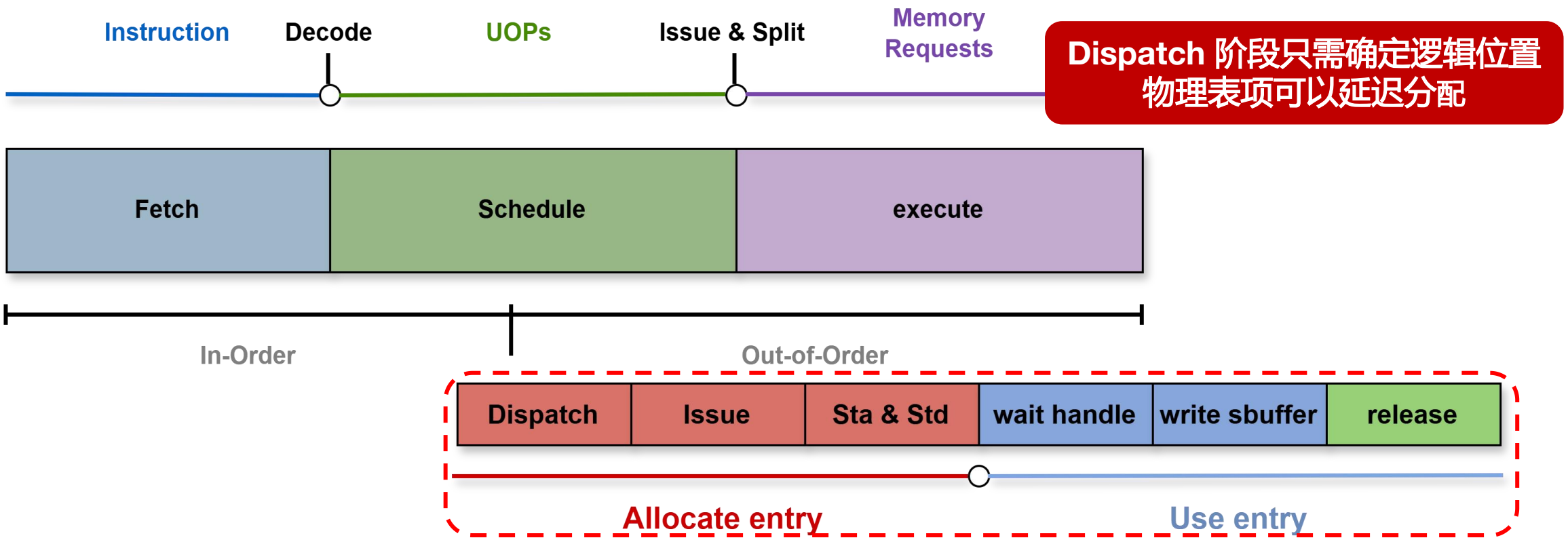
时序面积资源有限：StoreQueue 无法无限扩容



StoreQueue 容量扩展会持续堆高面积成本，且控制逻辑规模也会放大

虚拟 StoreQueue

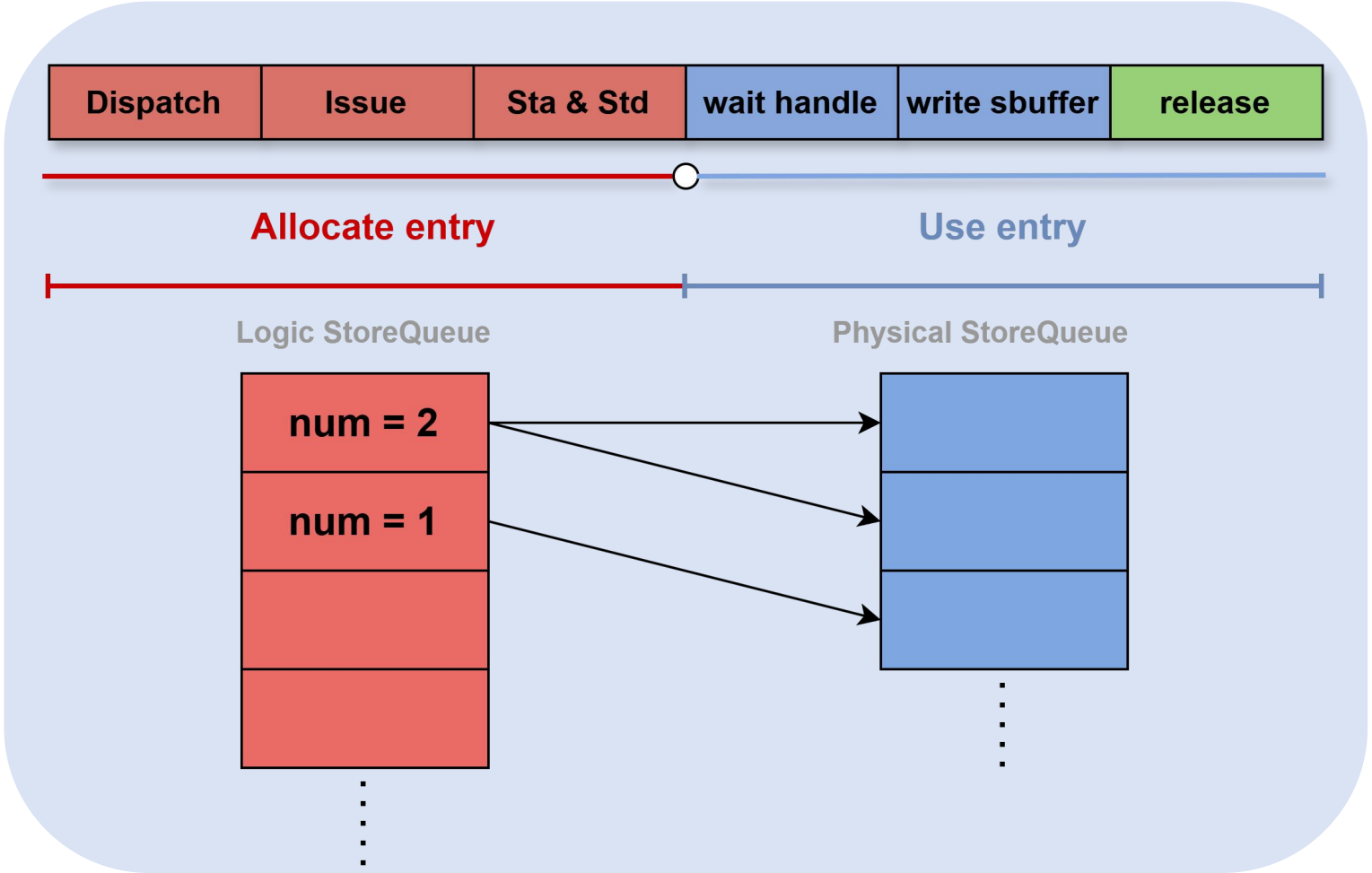
Store 在 Dispatch 阶段无需占用 StoreQueue 物理表项



设计思路：逻辑队列维护指令序，Store 执行阶段才实际分配物理 StoreQueue

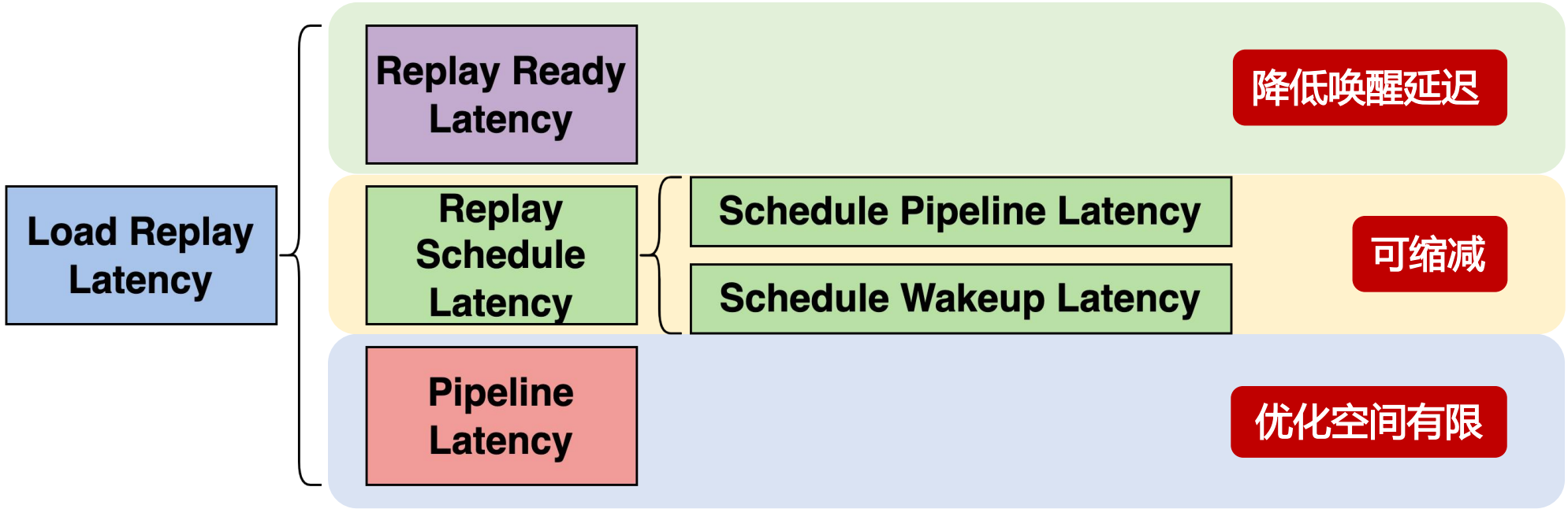
虚拟 StoreQueue 与 Store 预分配

- **逻辑队列** Dispatch 时分配
  - 保存 Uop 级信息
  - 维护 Uop 的生命周期
- **物理队列** Issue 时分配
  - 保存访存操作级信息
  - 维护访存操作的生命周期
  - 提供数据前递
  - 提供访存违例预测校验
  - 非对齐拆分写入 Sbuffer



Load Replay 延迟优化

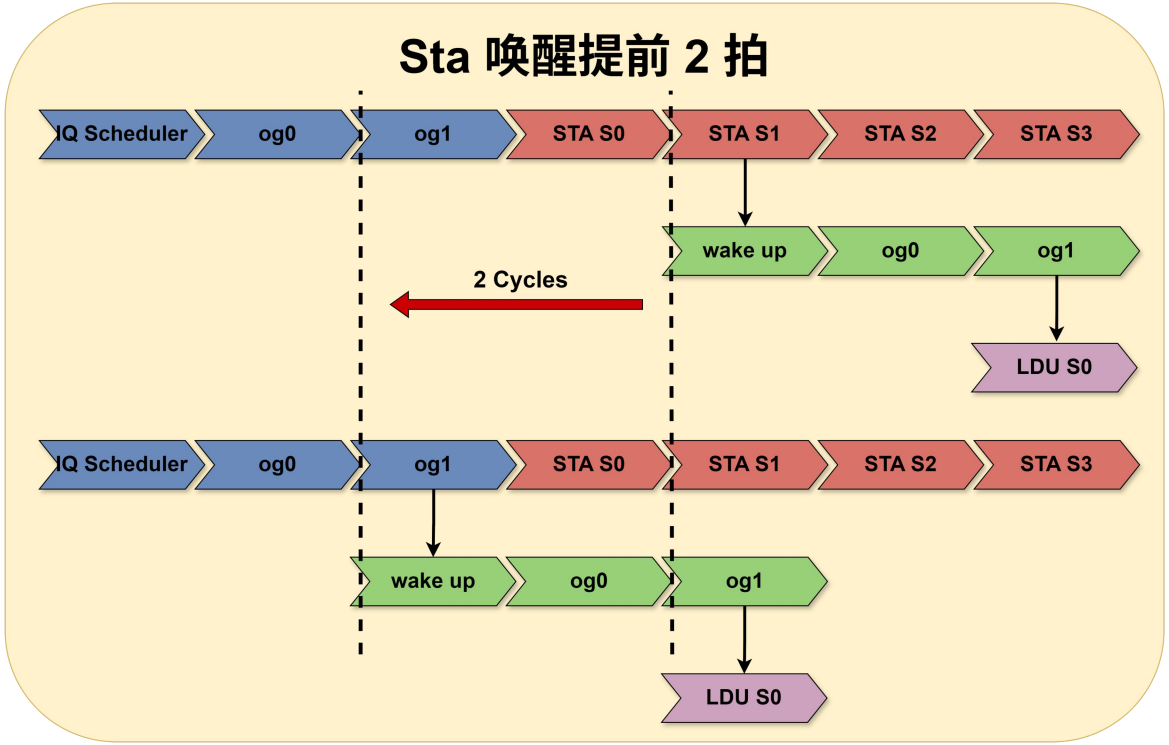
Load Replay 延迟 = **重发就绪延迟 + 重发调度延迟** + 流水线执行延迟

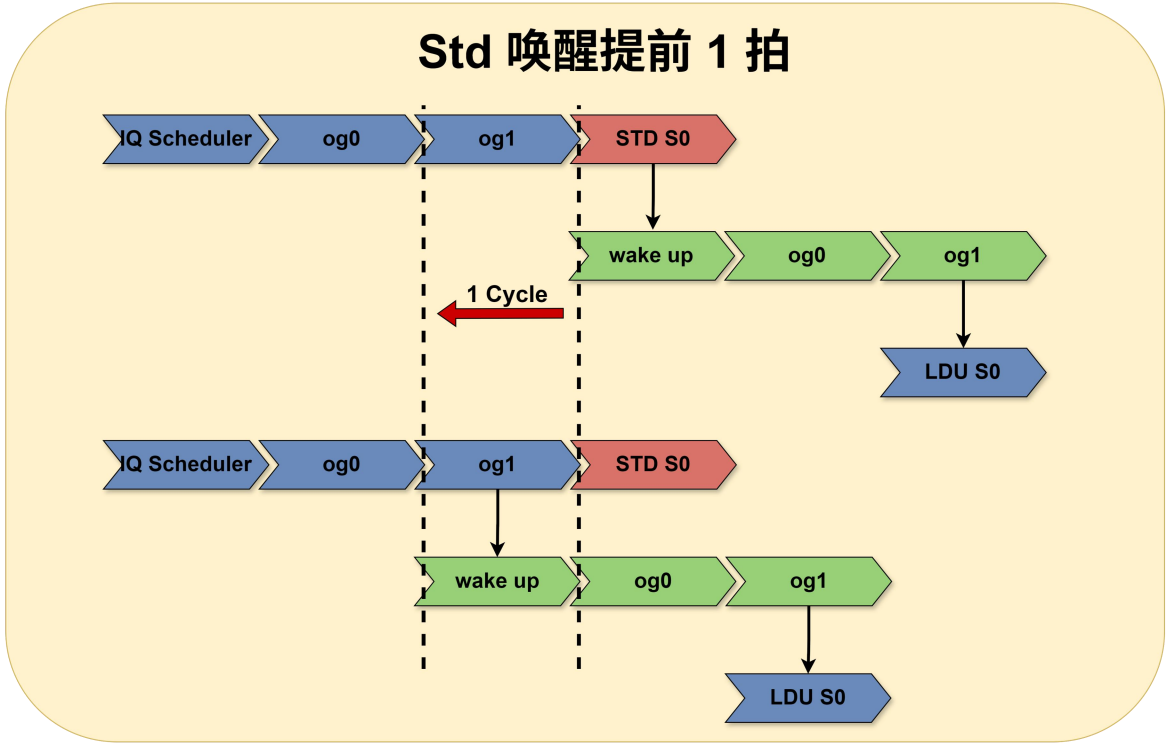


优化思路：通过提前唤醒掩盖调度延迟

Load Replay 唤醒延迟优化

提前唤醒 1：Sta/Std 发射时唤醒 Replay Load，降低 Store-Load 依赖等待延迟



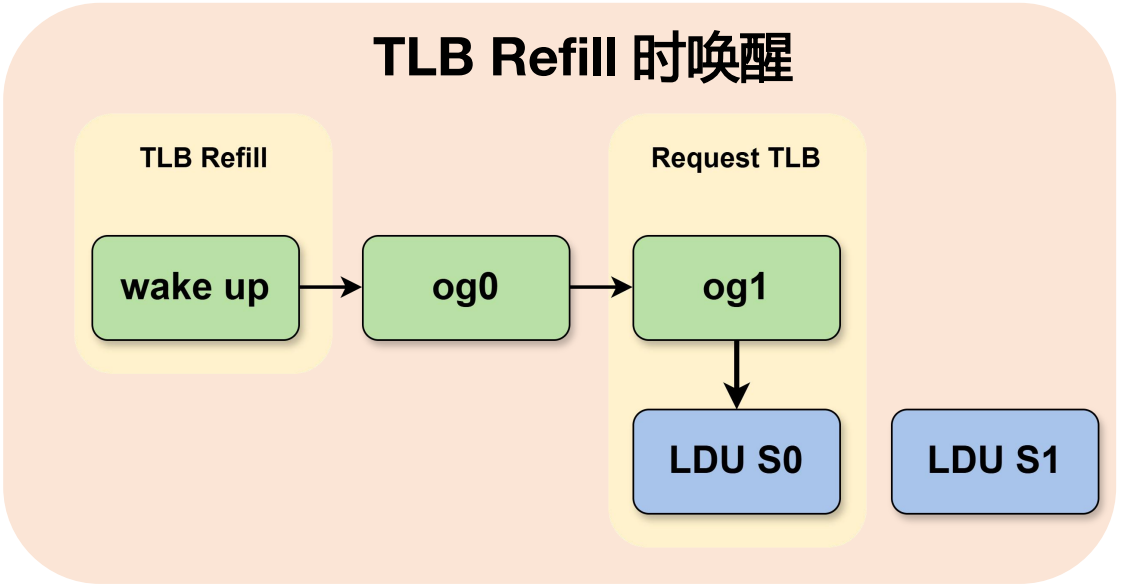


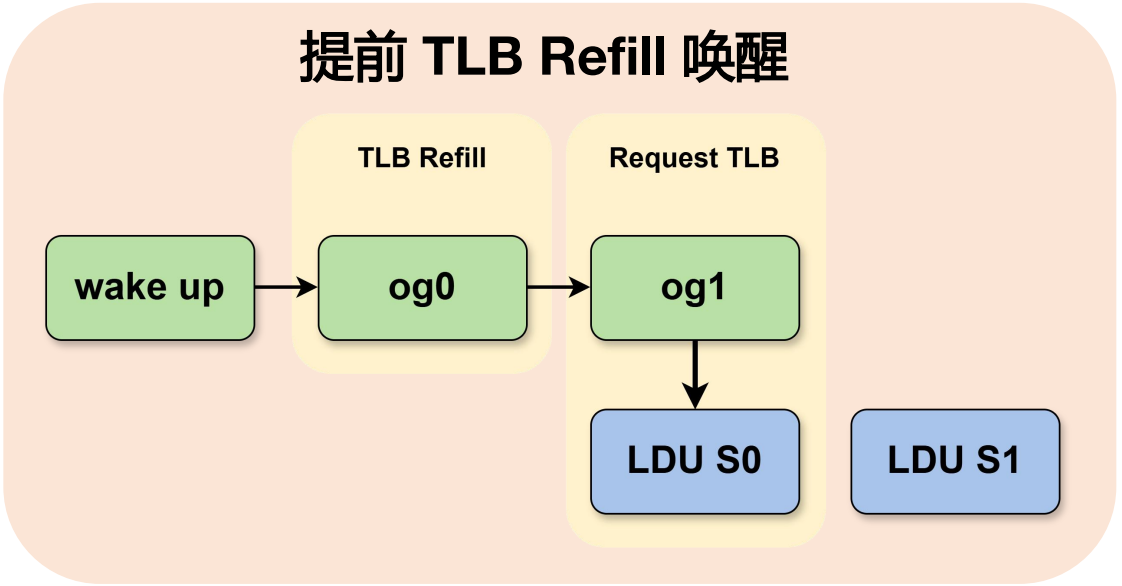
性能收益：SPECint2006 +0.41%，SPECfp2006 +0.36%

Load Replay 调度延迟优化

提前唤醒 1：Sta/Std 发射时唤醒 Replay Load，降低 Store-Load 依赖等待延迟

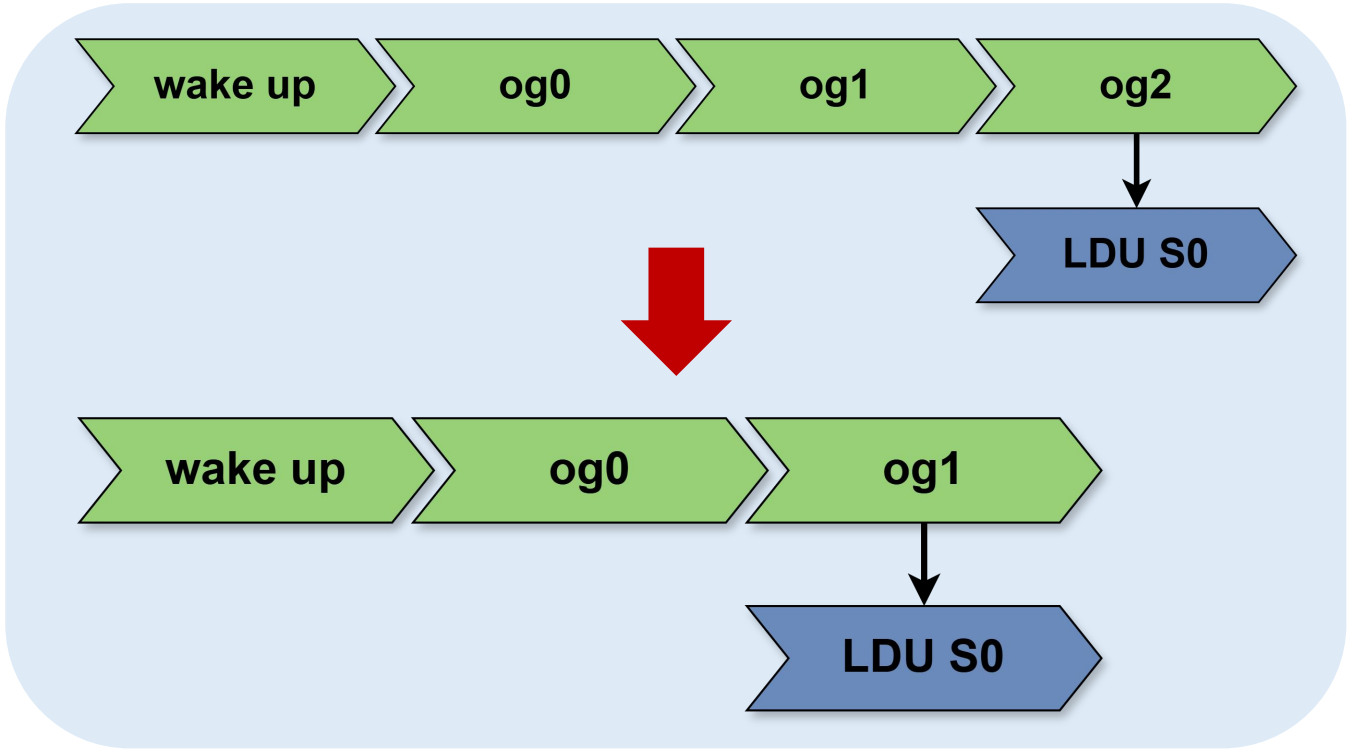
提前唤醒 2：TLB 回填提前唤醒 Replay Load，降低 TLB miss 等待延迟





Load Replay 调度延迟优化

提前唤醒 1：Sta/Std 发射时唤醒 Replay Load，降低 Store-Load 依赖等待延迟
提前唤醒 2：TLB 回填提前唤醒 Replay Load，降低 TLB miss 等待延迟
调度延迟优化：Replay Load 调度流水延迟从 2 cycle 优化至 1 cycle



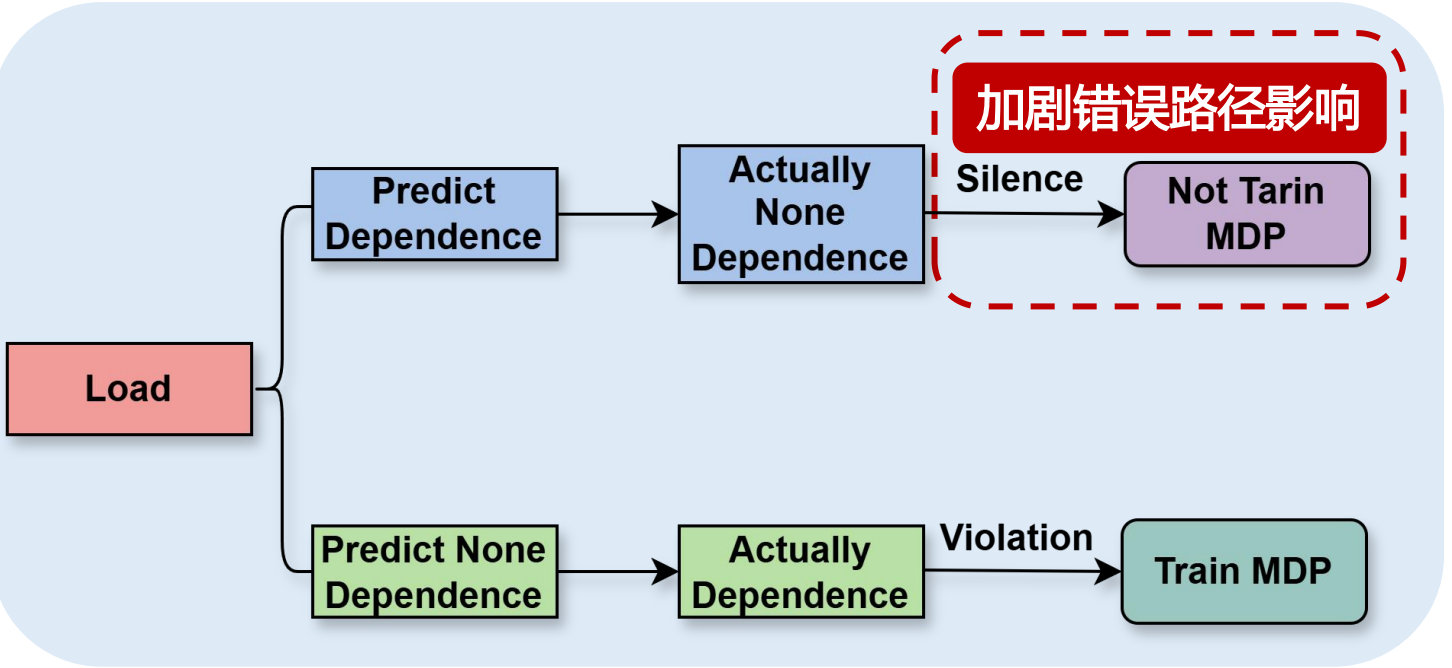
TLB 回填唤醒 + 调度延迟优化性能收益：SPECint2006 +1.05%，SPECfp2006 +0.64%

MDP 优化

观察：推测执行引入错误路径污染；部分错误预测缺乏反馈训练

优化：

- 提供更准确的训练信息
- Strict 预测表项提前释放
- 路径感知的 Strict 预测生成

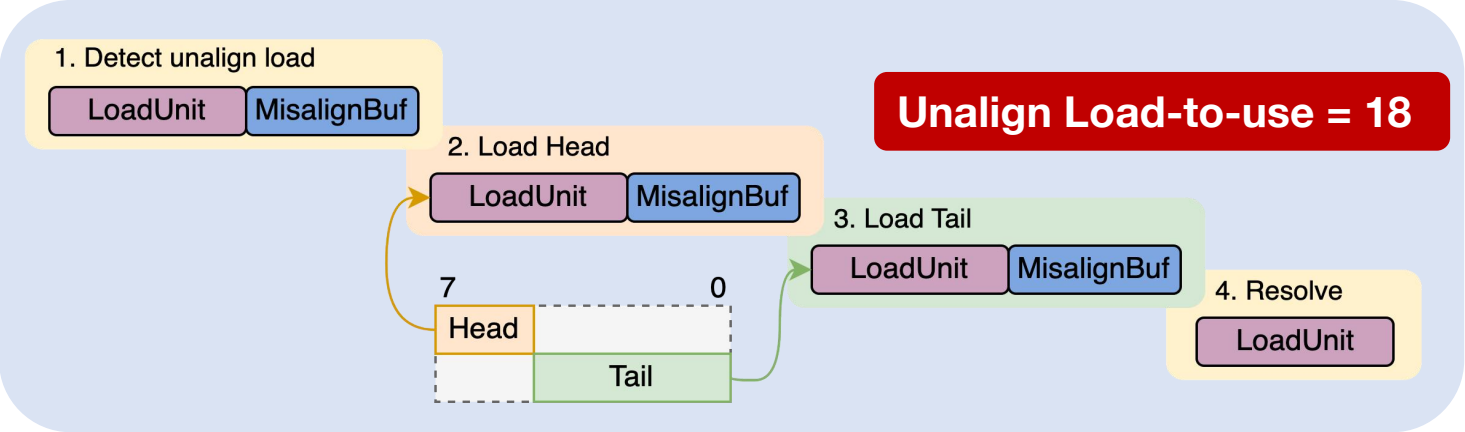


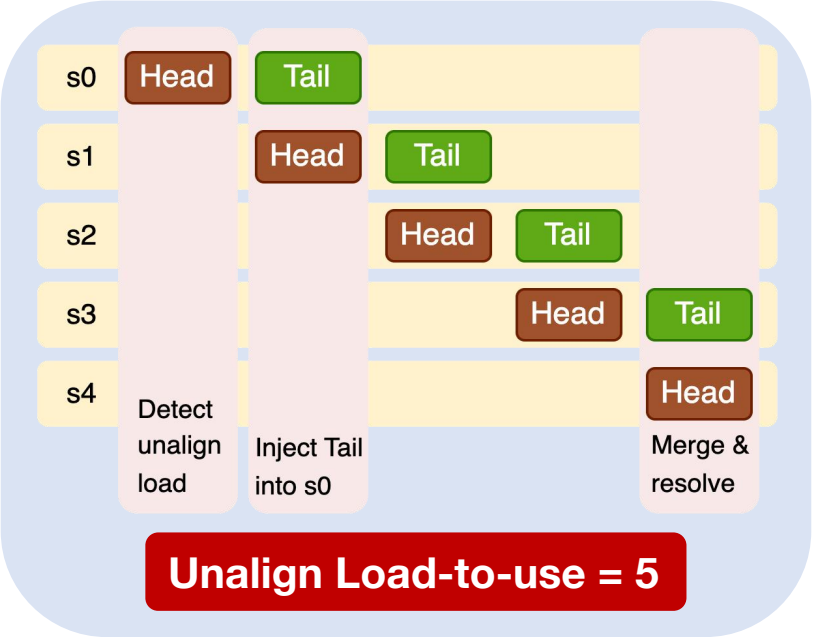
性能收益：**SPECint2006 +0.17%，SPECfp2006 +0.02%**

[1] G. Z. Chrysos and J. S. Emer, "Memory dependence prediction using store sets," *Proceedings. 25th Annual International Symposium on Computer Architecture (Cat. No.98CB36235)*, Barcelona, Spain, 1998, pp. 142-153, doi: 10.1109/ISCA.1998.694770.

非对齐访存优化

- **延迟**：Load-to-use 从 **18** cycles 优化至 **5** cycles
  - **基于流水线的动态拆分机制**：LoadUnit 入口检测 + 动态拆分为双流背靠背执行
  - **非对齐 STLF 的支持**：StoreQueue 添加 UnalignQueue 提供非对齐访存前递
- **带宽**：解除重放单元 (MisalignBuffer) 容量限制，显著提升并行度



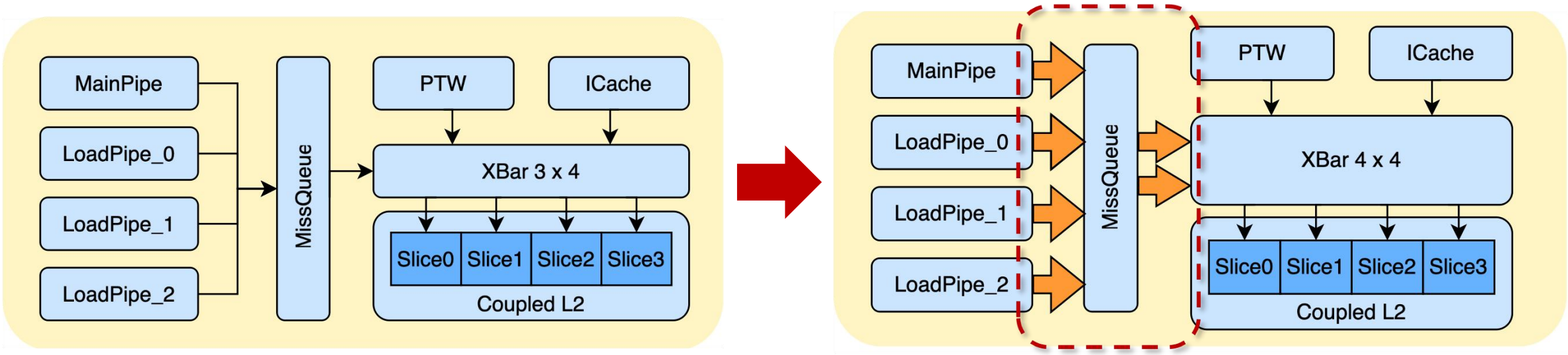


第二部分：缓存子系统

- L1-L2 带宽优化
- DCache MSHR 前递优化
- L2-Load-to-use 优化

L1-L2 带宽优化

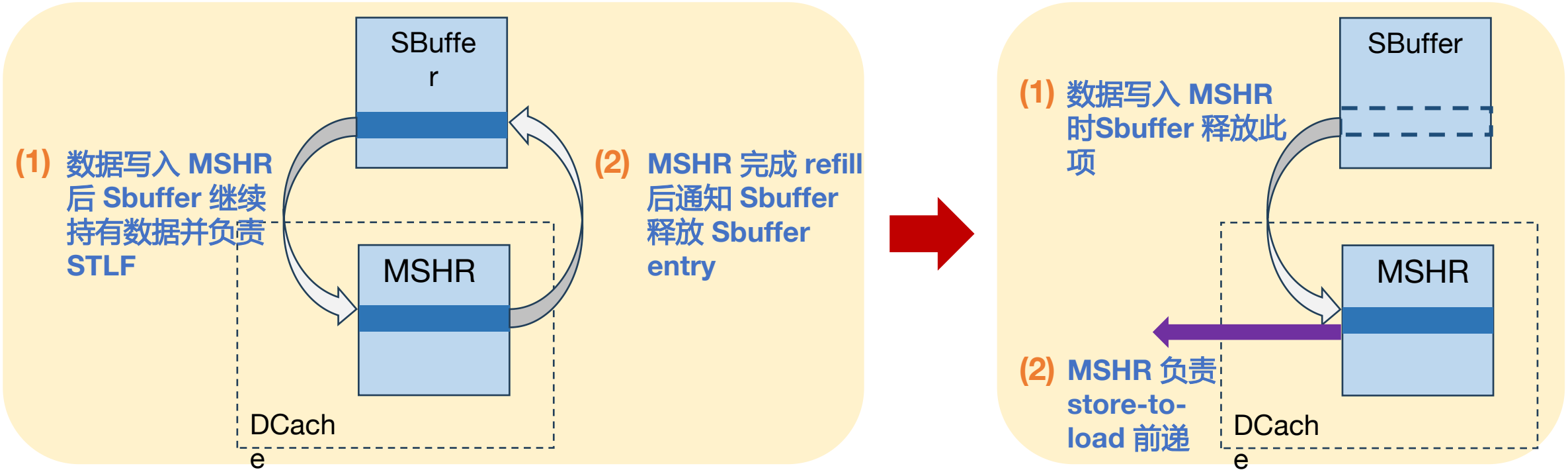
- **MSHR 入队带宽**：解决每周期只处理一个 miss 请求的带宽瓶颈
  - 支持一拍分配多项 MSHR
  - 支持多个 miss 请求间进行同地址压缩
- **MSHR 回填带宽**：缓解回填仲裁失败导致的阻塞
  - 基于地址低位交织的双通道 TileLink 带宽扩展方案



性能收益：SPECint2006 +0.6%，SPECfp2006 +1.44%

DCache MSHR 前递优化

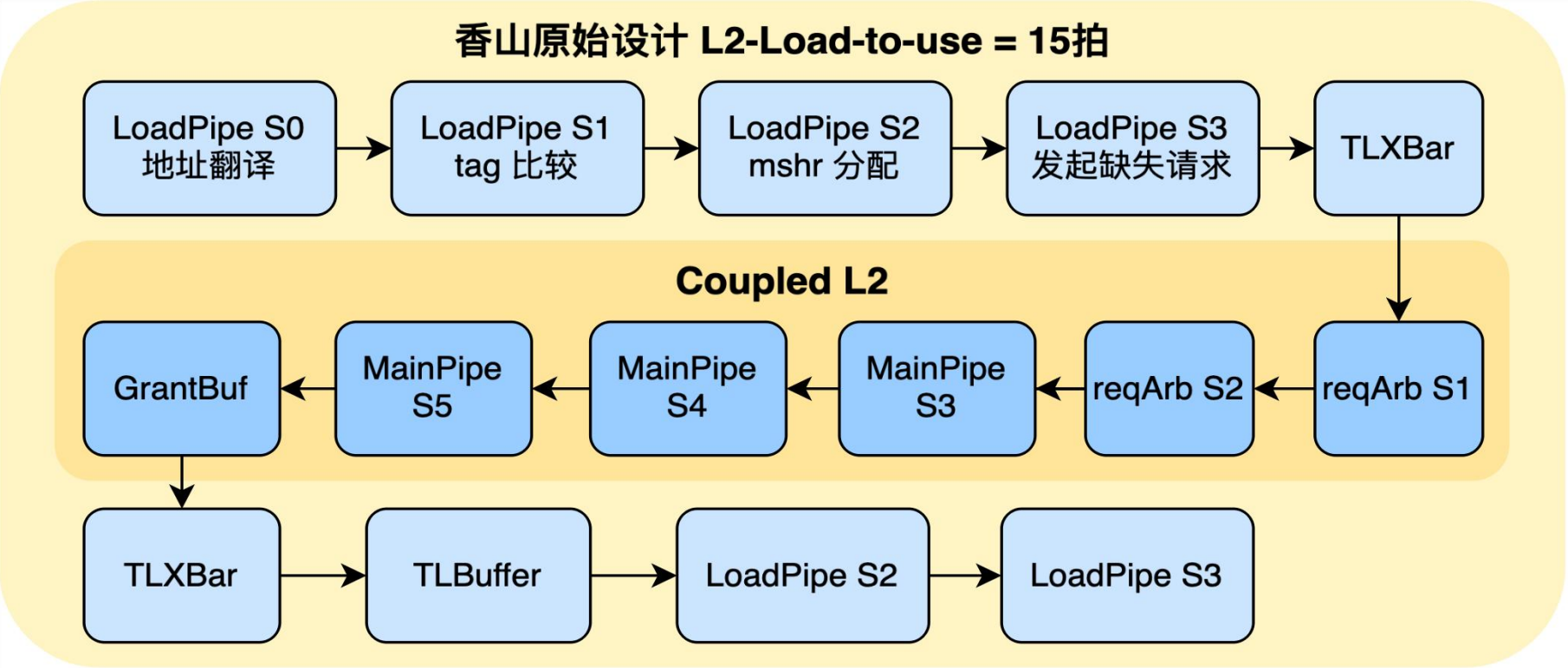
- **背景**：Sbuffer 替换数据在 Inflight 状态下继续持有数据并负责前递数据
- **优化**：DCache MSHR 接管数据前递，加速 Sbuffer 释放



性能收益：SPEC CPU2006 +0.26%

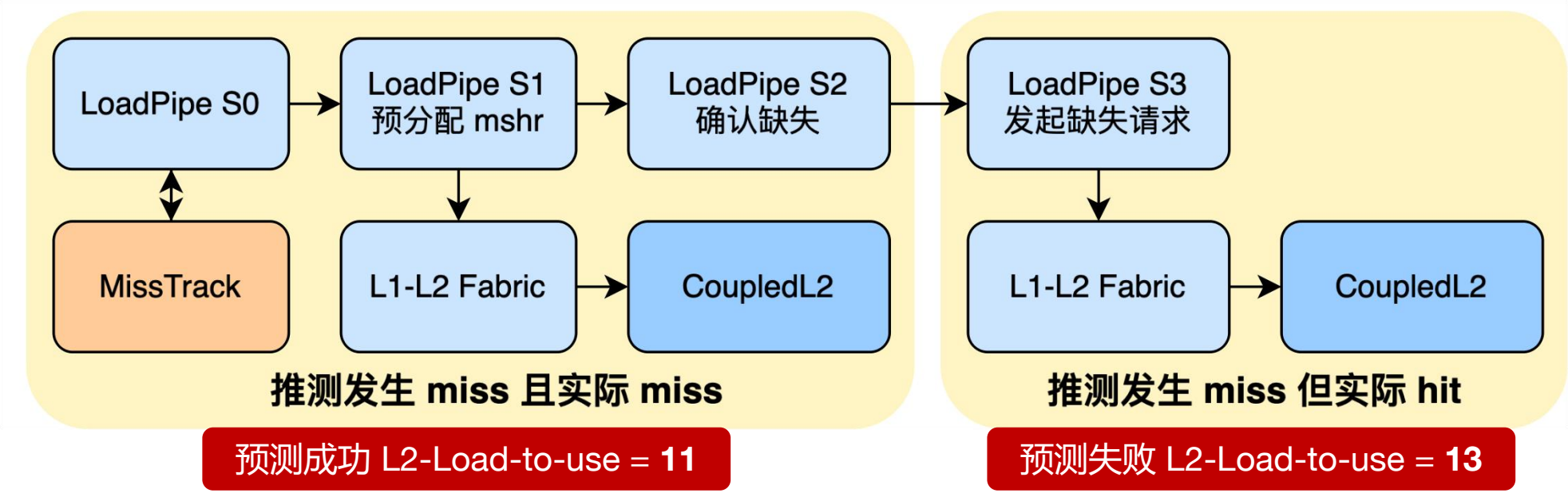
L2-Load-to-use 优化：背景

- L2-Load-to-use = 15（Load 指令在 L1 miss L2 hit 的最短延迟）
- L1-L2 请求通路采用保守的串行执行



L2-Load-to-use 优化

- **基于 MissTrack 的 L2 推测访问**
  - 预测 load 指令在 L1 是否命中提前推测发送回填请求
- **基于时序划分优化的推测路径重构**
  - 采用更平衡的时序划分消除 2 拍冗余级数



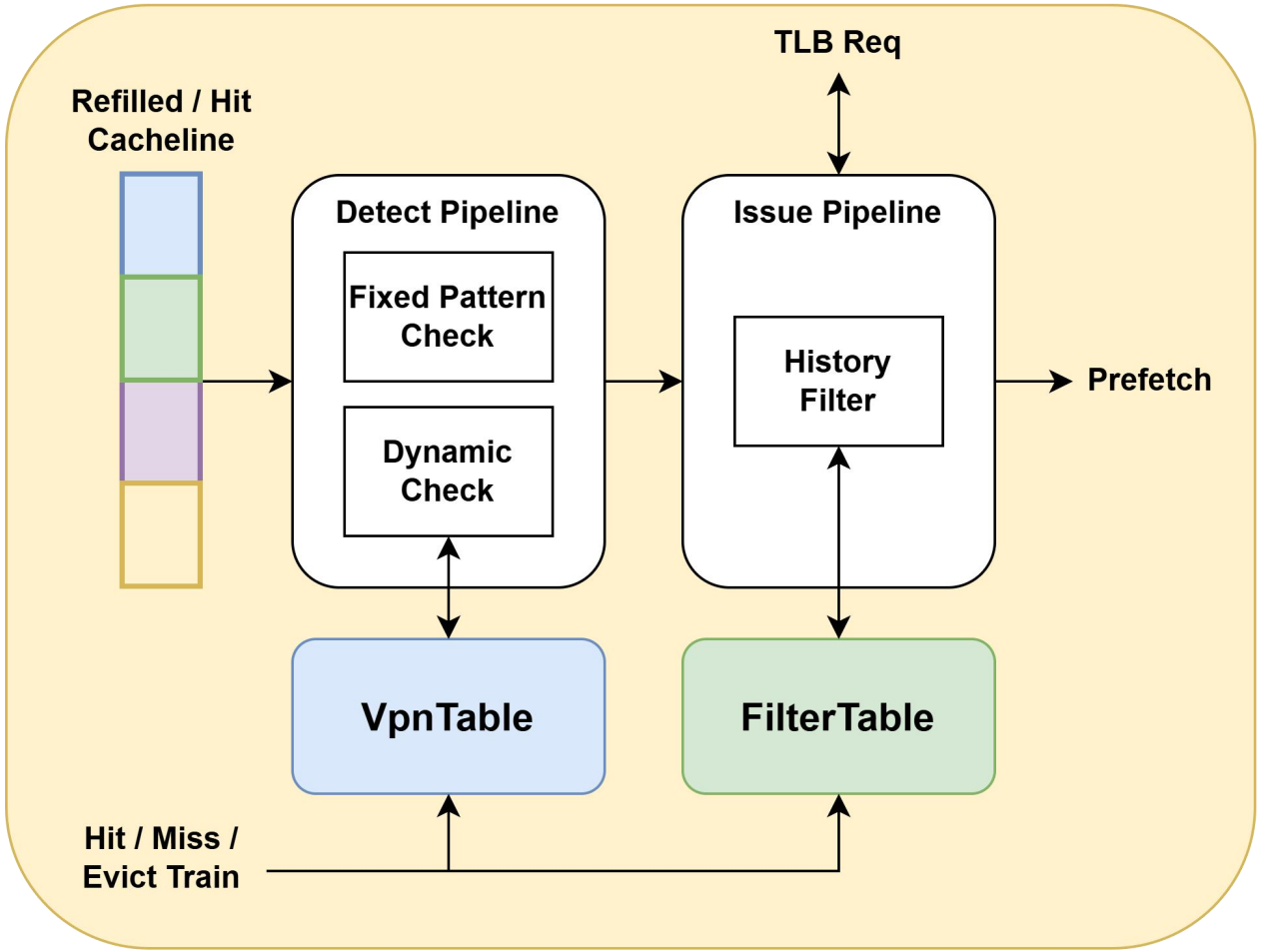
性能收益：SPECint2006 +1.28%,SPECfp2006 +0.67%

第三部分：预取

- Content-Directed Prefetcher（CDP）
- Teacher-Student Best Offset Prefetcher（T-S BOP）

CDP 预取

- 指针密集的负载访存压力大
- L2 CDP[1] 预取：
  - 检测回填/命中的缓存行
  - 对潜在地址提前发出预取请求



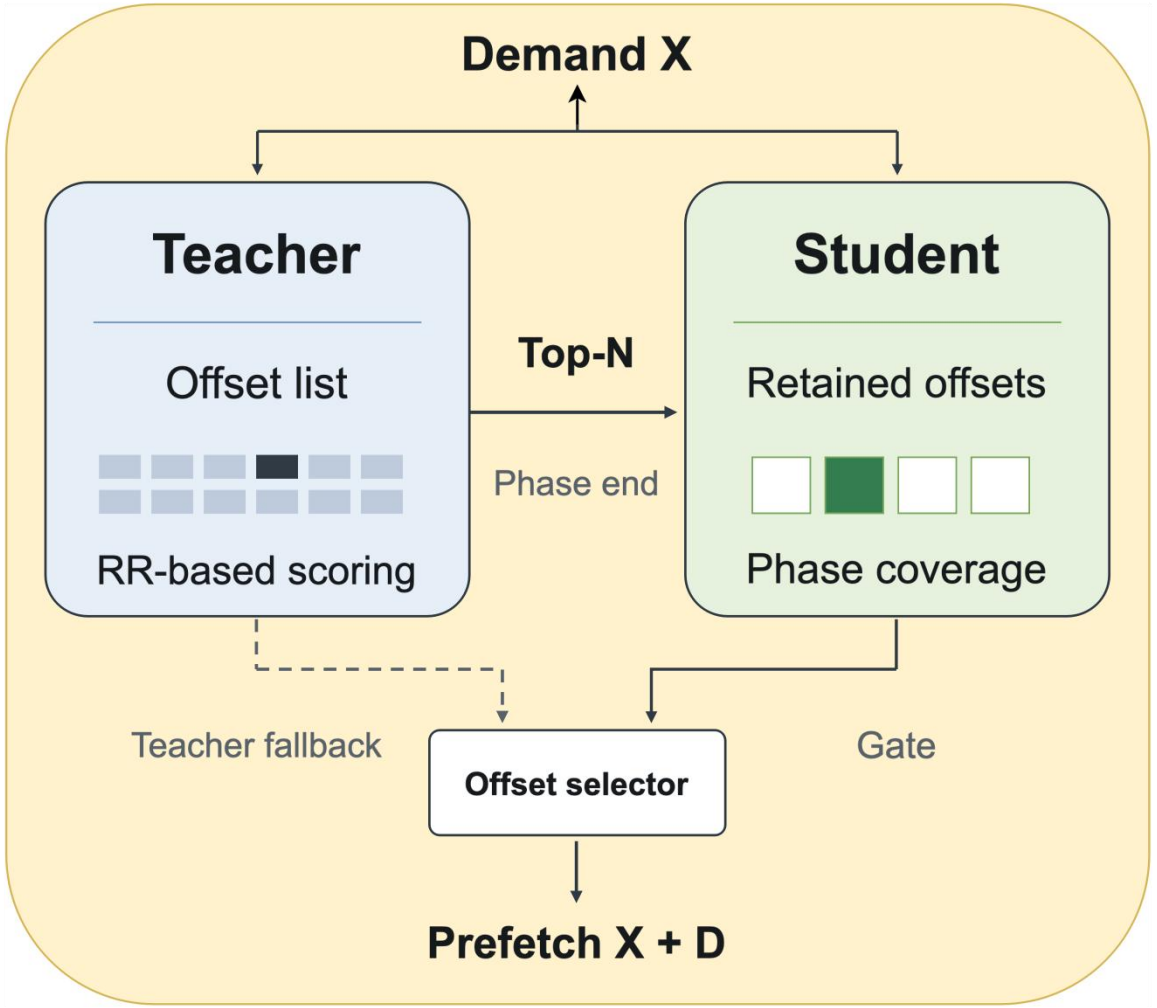
有效提升指针密集型负载性能

SPECint2006 +1.24%, SPECfp2006 –0.13%

[1] Robert Cooksey, Stephan Jourdan, and Dirk Grunwald. 2002. A stateless, content-directed data prefetching mechanism. SIGOPS Oper. Syst. Rev. 36, 5 (December 2002), 279–290. https://doi.org/10.1145/635508.605427

Teacher-Student BOP

- 经典 BOP:
  - X–d 命中 RR 表 → 候选 d 得分 +1
  - RR 得分最优 ≠ 覆盖率最优
- Teacher-Student 协同学习:
  - Teacher: RR 得分筛选候选偏移
  - Student: 少量候选项按覆盖率择优



进一步提升 BOP 预取性能

SPECint2006 +0.39%, SPECfp2006 +0.62%

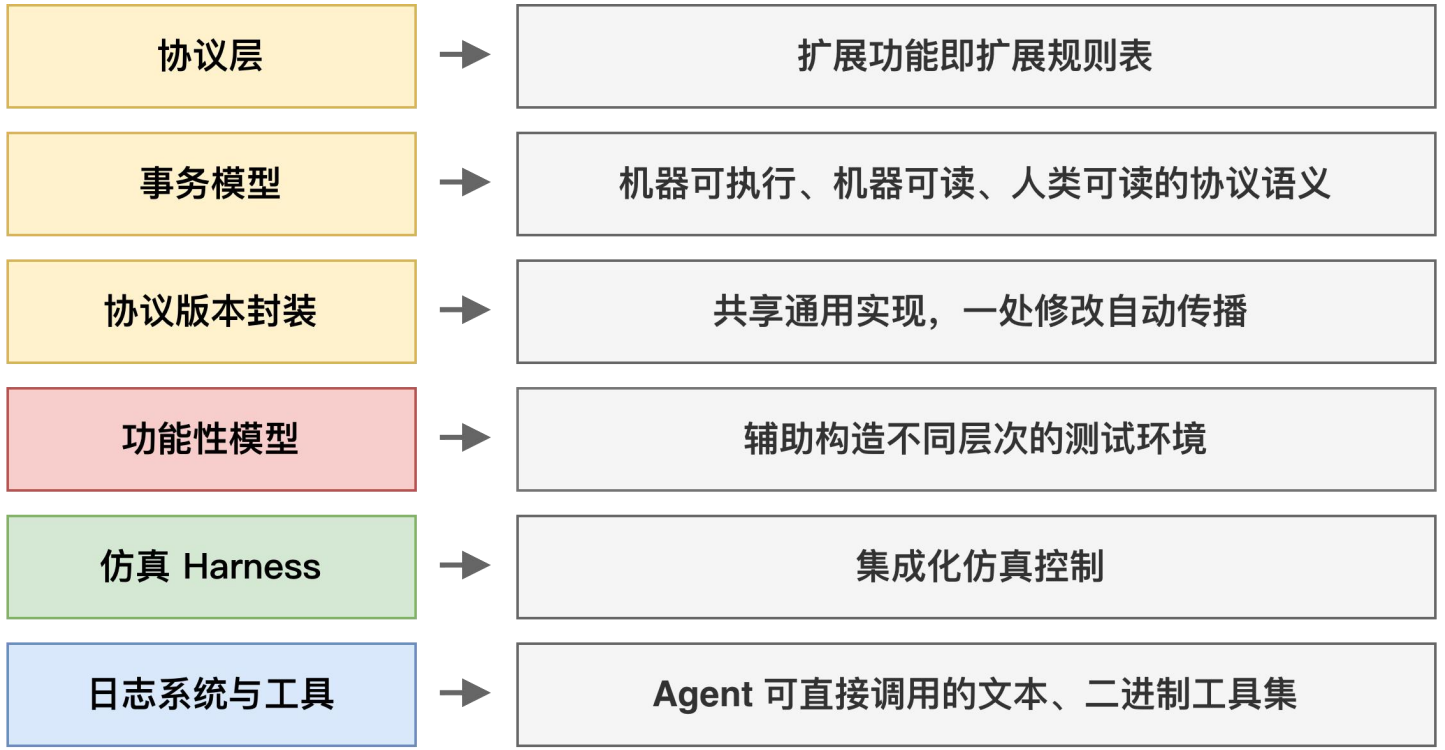
第四部分：CHI 工具集

- **CHIron: 开源 AMBA CHI 基础设施**
  - **世界首个**开源 AMBA CHI 工具集
  - 使用 C++ 描述 AMBA CHI 标准，利用编译器特性实现类 Machine Readable Specification
  - **已成为基础设施的基础设施（Infra of Infra）**
    - 同时 适合人读 和 适合机器读，**可以被 LLM 高效地吸收、扩展**
    - 朴素的编程语言（例如 C++）在有序编排后就已经**足够形式化**
      - 使用 C++ 对 AMBA CHI 的事务进行建模，其语义远比直接读取 协议手册 高效、清晰、歧义更少
      - 少有 C++ 编程能力差的 LLM，大部分 Dirty Work 已经可以完全交给 LLM
      - **CHIron C++ 本身就可运行，自然可以 Offload LLM Thinking， 最大程度减少幻觉**
      - **CHIron C++ 本身就已经携带了足够的形式化信息，可以指导 LLM 执行大规模的静态检查**

https://github.com/RISMicroDevices/CHIron

开源 CHI 基础设施

- 软件工程友好的框架设计
  - 高可维护性、多层次解耦，切分不同需求的出发点

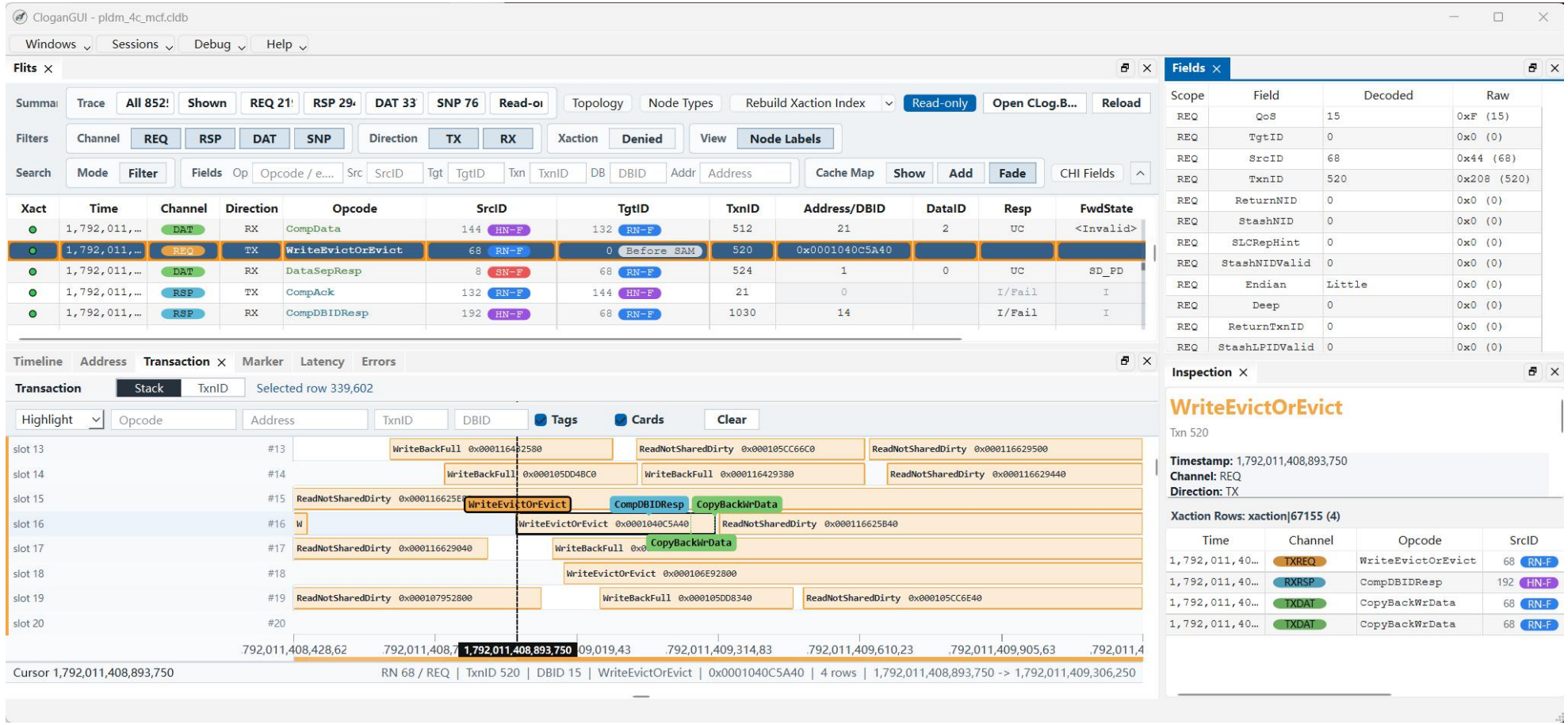


***LLM 的输出从不被直接轻信 —— 每一个层次都有各自的编译流程、测试环境，各层次之间亦可以交叉验证**

开源 CHI 基础设施

- **CloganGUI**: 基于 CHIron 开发的图形化 AMBA CHI 总线专家

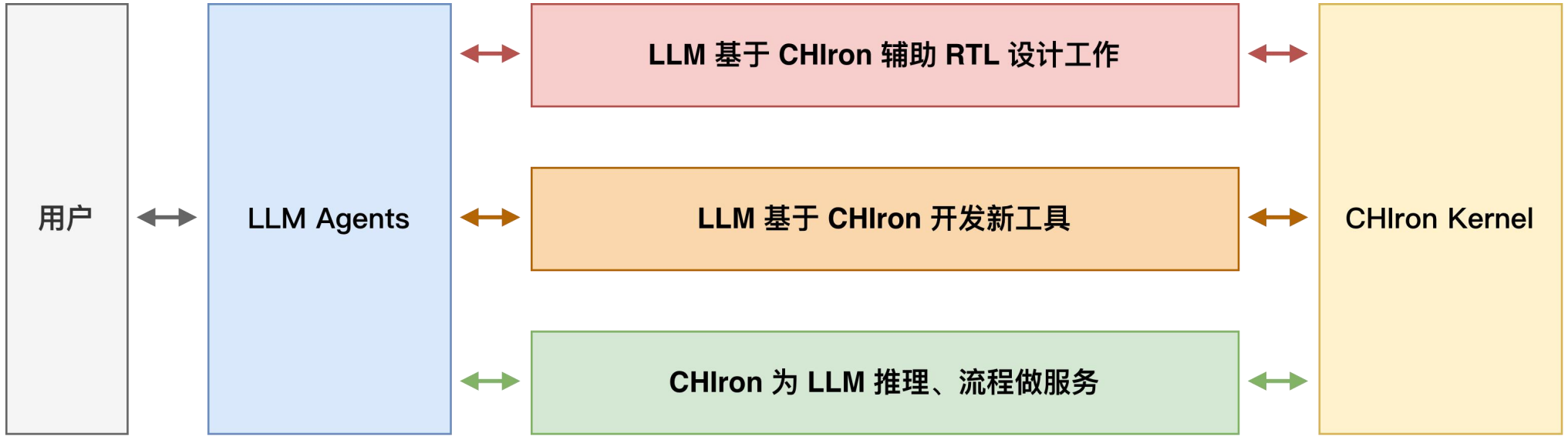
- CHI Trace 分析
- 总线正确性检查
- Flit 的事务匹配
- 地址流分析
- 延迟分析
- ……



https://github.com/RISMicroDevices/CHIron

开源 CHI 基础设施

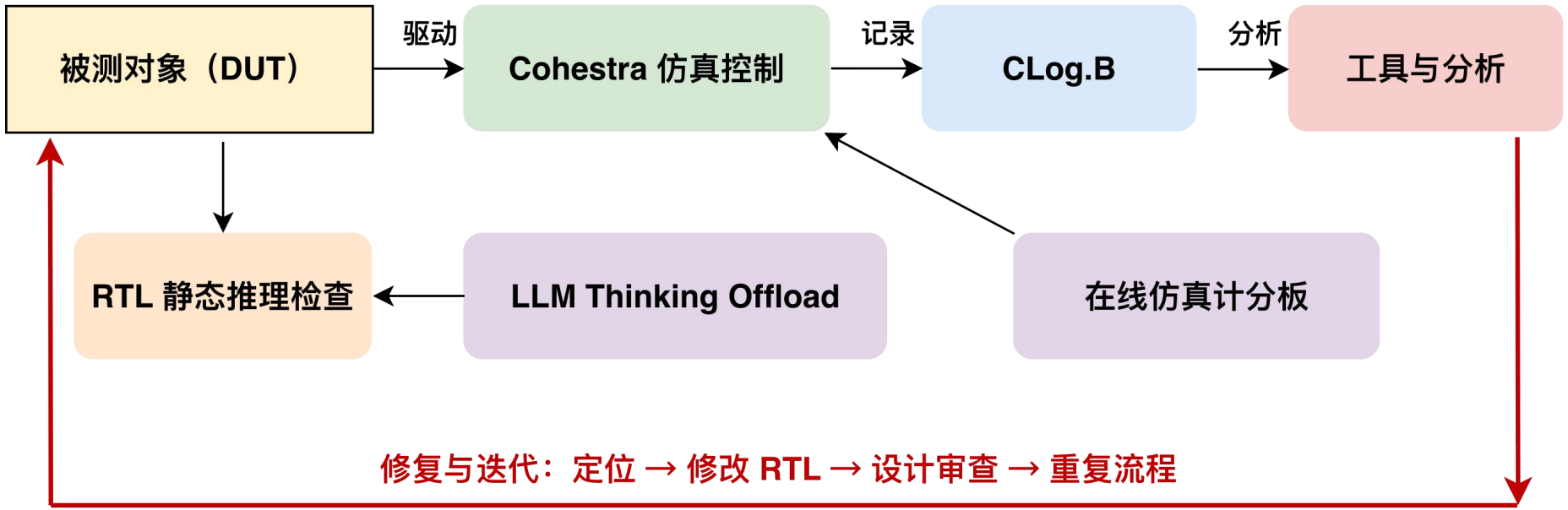
- CHIron × LLM 的三个维度
  - **LLM 基于 CHIron 辅助 RTL 设计工作**：协议扩展、嵌入 RTL 工作流、覆盖率收集
  - **LLM 基于 CHIron 开发新工具**：提供横向扩展工作流能力、自迭代能力
  - **CHIron 为 LLM 推理、流程做服务**：通过阅读代码、调用代码获取知识而非纯粹推理上下文



https://github.com/RISMicroDevices/CHIron

开源 CHI 基础设施

- **驱动下一代缓存的开发与测试**
  - 几乎所有功能以 **API** 形式提供 —— **可嵌入任何验证流程、可嵌入任何 Agent 工作流**
  - **在线**：DPI 嵌入仿真计分板，逐 Flit 检查事务与一致性状态
  - **离线**：CLog.B 数据流 —— 仿真 / FPGA / 硅后同一接口分析
  - 支持香山 Kunminghu V3（AMBA CHI、L1-L2 私有总线）缓存 IP 的开发与验证



敬请批评指正！