Zen: a High-Throughput Log-Free OLTP Engine for Non-Volatile Main Memory

陈祺汇
M202173480

Motivation

NVM

- 大容量：最大6TB
- 3DXPoint 比 DRAM 慢 2-3倍
- 写带宽相比于 DRAM 小很多

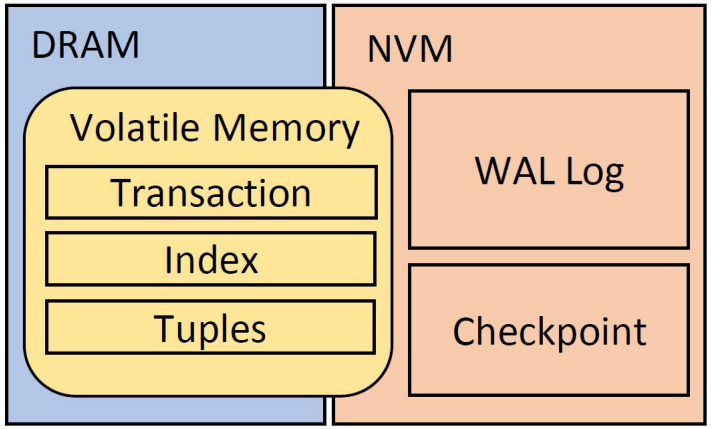
主存 OLTP 引擎

- 超过百万TPS
- 持久化：带来额外 I/O 开销
- 容量：受到DRAM限制

本文的目标：基于NVM的OLTP

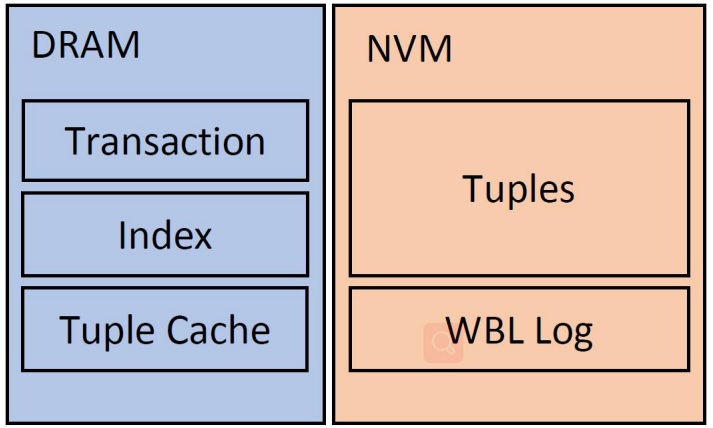
- NVM 作为持久性存储介质
- 比基于DRAM的引擎大得多的容量
- 高吞吐 + 快速恢复

现存的基于NVM的OLTP设计



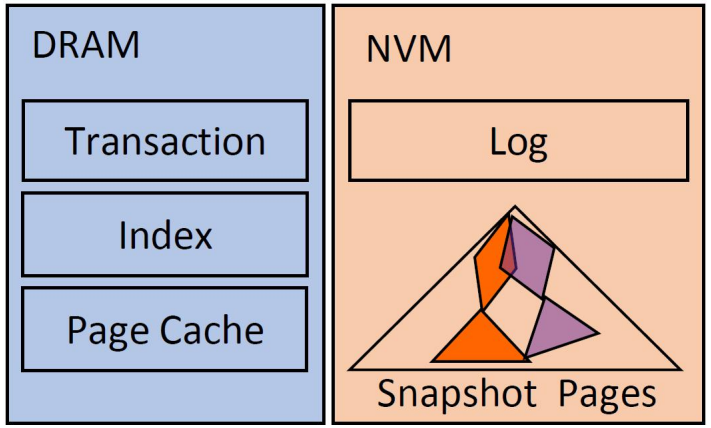
MMDB

- 冗余的NVM写操作
- 数据库增大时，性能下降严重



WBL (VLDB' 16)

- 频繁的tuple元数据修改
- Tuple级别的NVM空间管理



FOEDUS (SIGMOD' 15)

- 读/写放大 (Page)
- 以 I/O 接口访问NVM

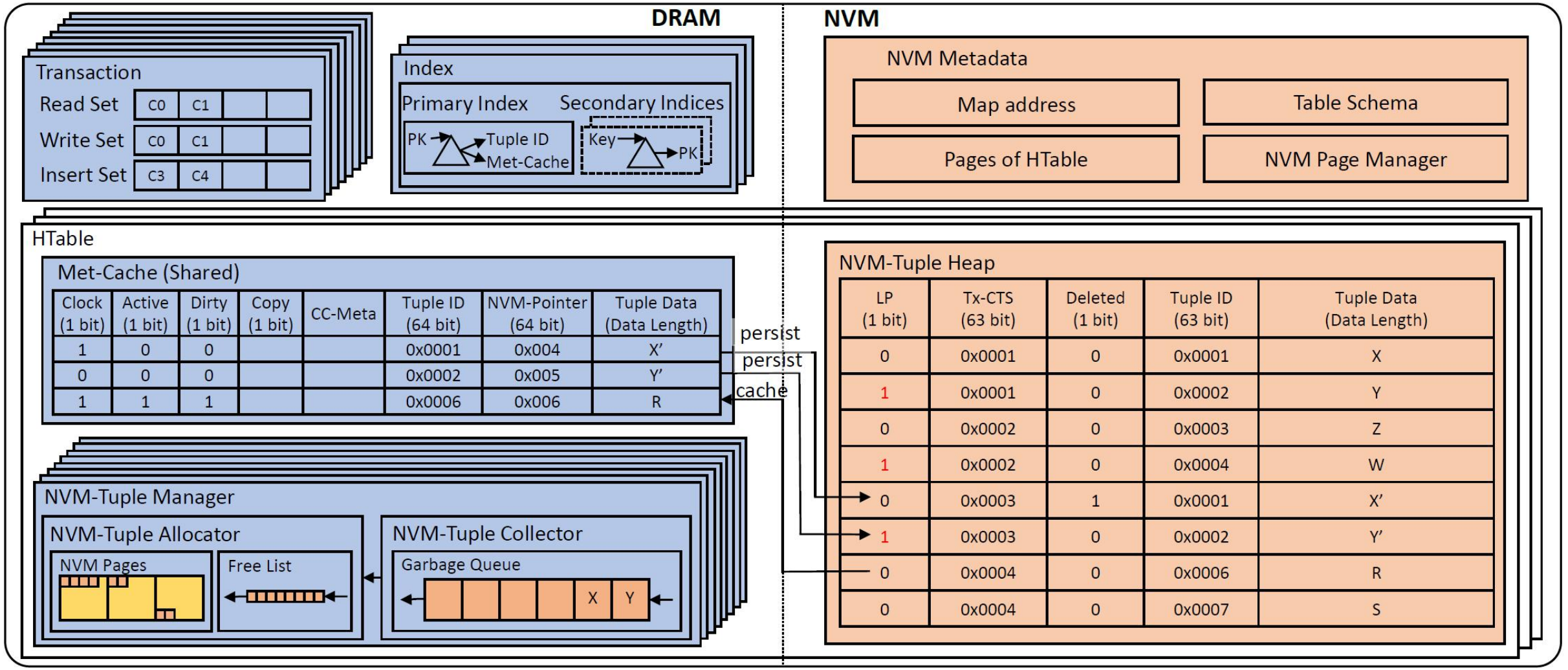
设计的三个挑战

Tuple 元数据修改

NVM 写冗余

NVM 空间管理

本文的设计：Zen Architecture



Metadata Enhanced Tuple Cache

Met-Cache (Shared)

Clock (1 bit)	Active (1 bit)	Dirty (1 bit)	Copy (1 bit)	CC-Meta	Tuple ID (64 bit)	NVM-Pointer (64 bit)	Tuple Data (Data Length)
1	0	0			0x0001	0x004	X’
0	0	0			0x0002	0x005	Y’
1	1	1			0x0006	0x006	R

NVM-Tuple Heap

LP (1 bit)	Tx-CTS (63 bit)	Deleted (1 bit)	Tuple ID (63 bit)	Tuple Data (Data Length)
0	0x0001	0	0x0001	X
1	0x0001	0	0x0002	Y
0	0x0002	0	0x0003	Z
1	0x0002	0	0x0004	W
0	0x0003	1	0x0001	X’
1	0x0003	0	0x0002	Y’
0	0x0004	0	0x0006	R
0	0x0004	0	0x0007	S

- Miss时，从NVM读取tuple
- 为每个tuple添加元数据
- Read可能会改变元数据

挑战1：Tuple元数据修改

Log-Free Persistent Transactions

- 完全消除了log、checkpoint和snapshot

Met-Cache (Shared)

Clock (1 bit)	Active (1 bit)	Dirty (1 bit)	Copy (1 bit)	CC-Meta	Tuple ID (64 bit)	NVM-Pointer (64 bit)	Tuple Data (Data Length)
1	0	0			0x0001	0x004	X’
0	0	0			0x0002	0x005	Y’
1	1	1			0x0006	0x006	R

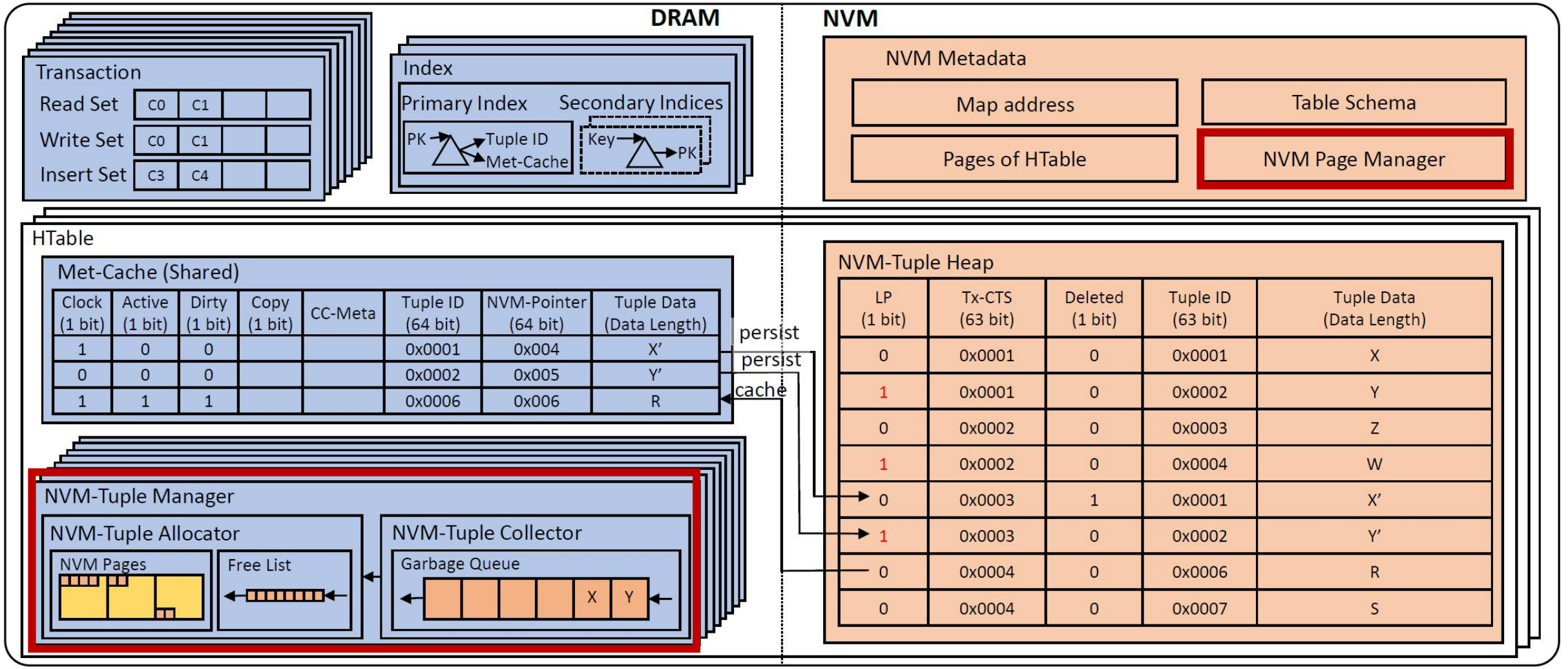
NVM-Tuple Heap

LP (1 bit)	Tx-CTS (63 bit)	Deleted (1 bit)	Tuple ID (63 bit)	Tuple Data (Data Length)
0	0x0001	0	0x0001	X
1	0x0001	0	0x0002	Y
0	0x0002	0	0x0003	Z
1	0x0002	0	0x0004	W
0	0x0003	1	0x0001	X’
1	0x0003	0	0x0002	Y’
0	0x0004	0	0x0006	R
0	0x0004	0	0x0007	S

- 每个事务最后持久化的tuple会有一个LP=1的标志
- 断电时通过扫码NVM来确定请求完成情况
- 断电时通过扫码NVM来确定请求完成情况

挑战2：NVM冗余写

Lightweight NVM Space Management

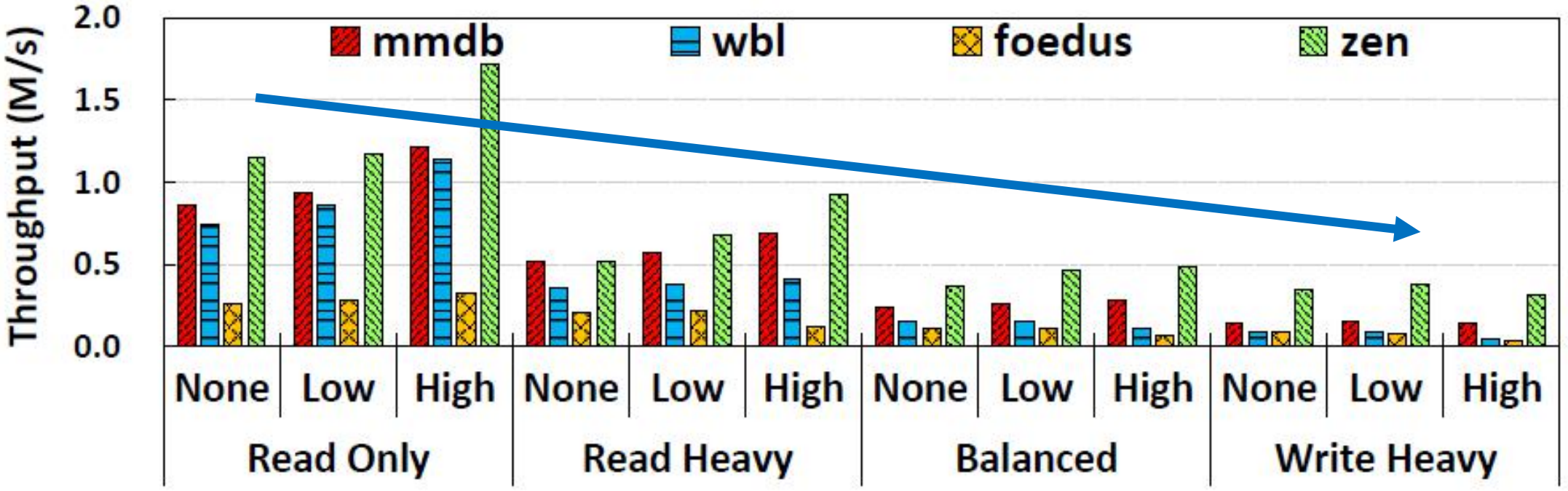


挑战3：NVM空间管理的开销

实验配置

- Zen的实现
  - Cicada (SIGMOD’17)
  - DBx1000 (VLDB’14)：9种并发控制方法
- 机器配置
  - 2 Intel Xeon Gold 5218 CPUs (16核 / 32线程 per CPU)
  - 12 * 32GB DRAM, 12x128GB Intel Optane DC PM
  - Ubuntu 18.04.3 LTS
- Benchmark
  - YCSB：256GB data base
  - TPCC-NP：初始大小=205GB

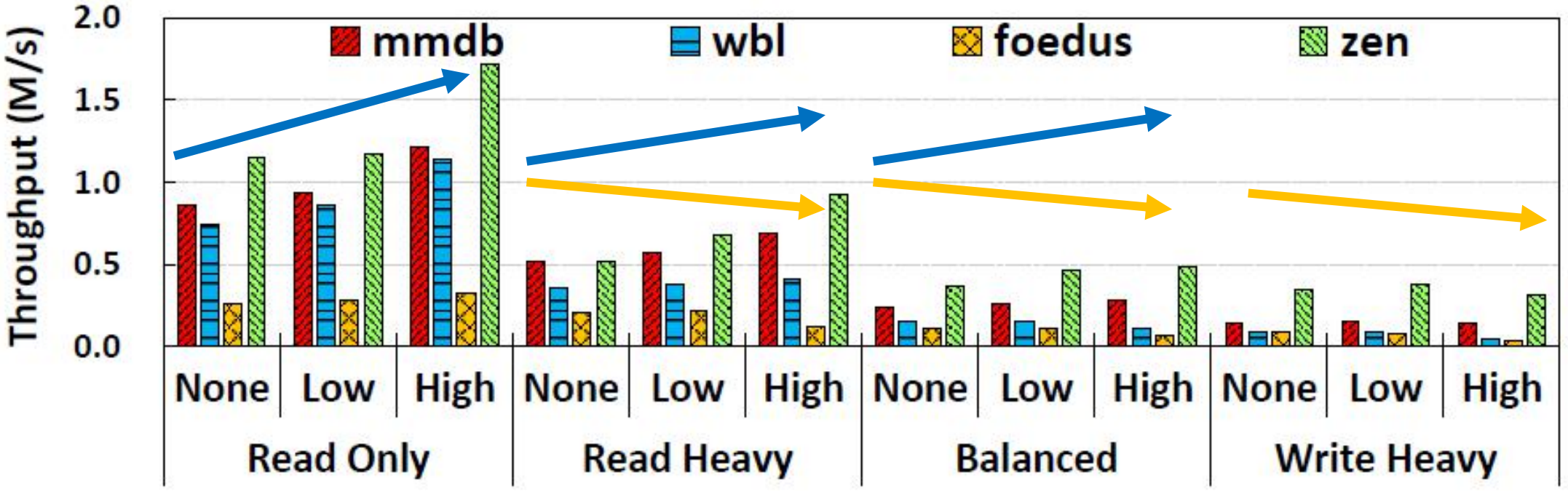
High Throughput (YCSB)



YCSB负载，Cicada，NVM和DRAM容量4:1，16线程，no NUMA effect

More Writes ➔ Lower throuthput

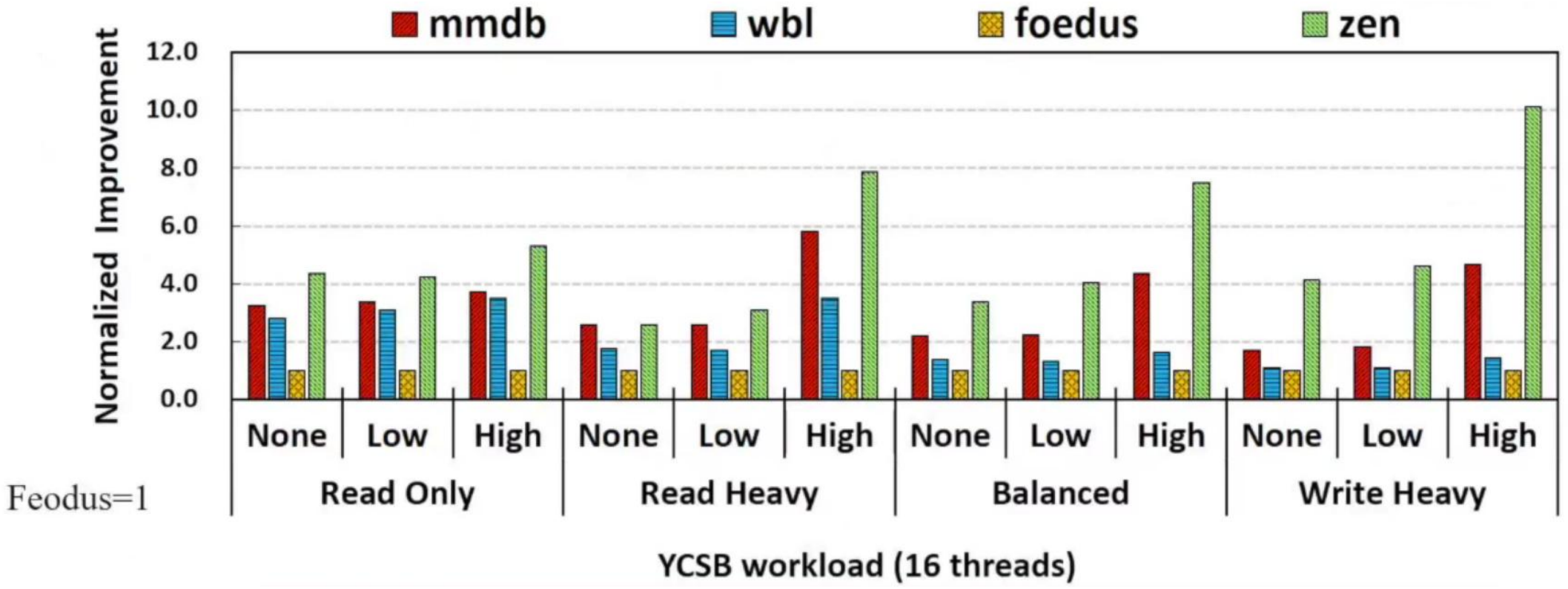
High Throughput (YCSB)



YCSB负载，Cicada，NVM和DRAM容量4:1，16线程，no NUMA effect

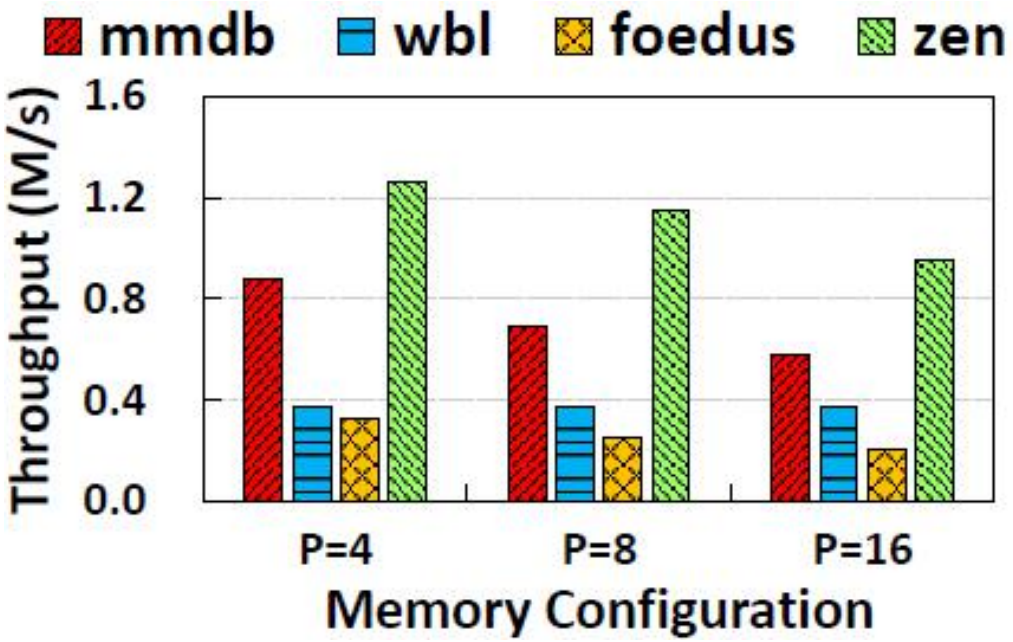
Higher skews ➔ More read hits + more write conflicts

High Throughput (YCSB)

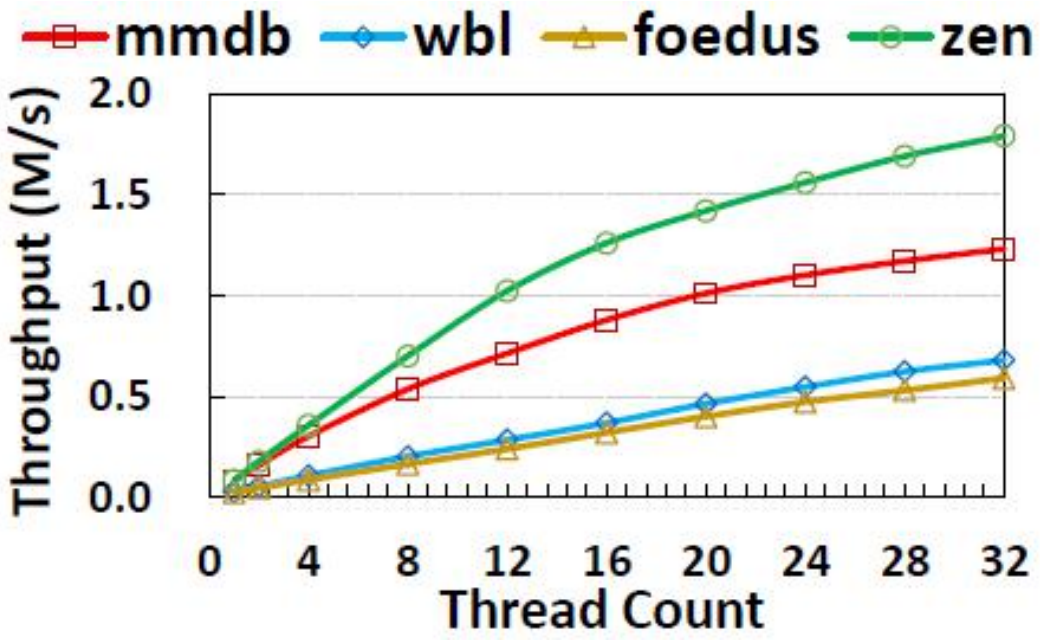


总的来说，在不同的工作负载下，Zen比现有的基于NVM的OLTP引擎，性能提升了**1.1倍到10.1倍**

TPCC-NP



改变NVM与DRAM容量比



改变线程数

Zen的性能优于现存的基于NVL的OLTP设计，
并且扩展性很好（32线程性能也很好）

Fast Recovery

- 在运行4M和16M事务后停止
- 使用16个线程执行recovery

Zen实现了快速恢复

总结

- NVM在OLTP上存在巨大潜力
  - 大容量；持久性；可字节寻址
- Zen：提出三个设计来解决三个问题
  - 元数据增强的tuple cache ➔ 元数据修改导致的NVM写
  - Log-free的持久性事务 ➔ NVM的冗余写
  - 双级别的空间分配策略 ➔ 轻量级NVM空间管理
- 优点
  - 高吞吐量
  - 快速恢复
  - 灵活的并发控制支持