# LB+-Trees: Optimizing Persistent Index Performance on 3DXPoint Memory

**Jihang Liu, Shimin Chen***

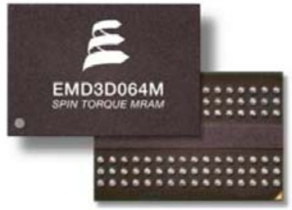
Institute of Computing Technology
Chinese Academy of Sciences

**Lujun Wang**

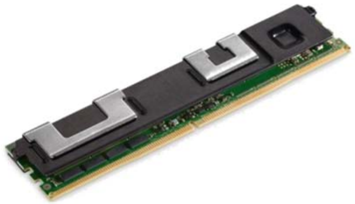
Alibaba Group

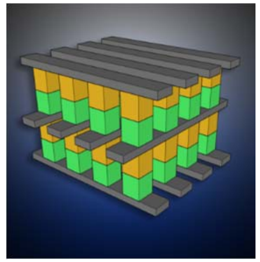
# Non-Volatile Memory

- **Multiple competing technologies**
  - ❑ PCM, STT-RAM, Memristor, 3DXPoint memory

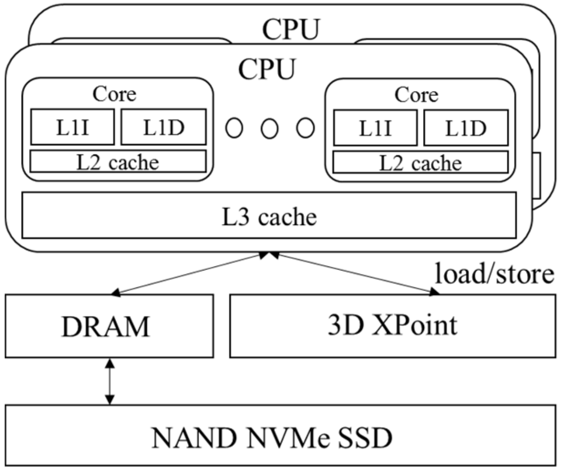



- **3DXPoint (Intel Optane DC Persistent Memory)**
  - ❑ 2015, Intel & Micron announced 3DXPoint
  - ❑ 2017, Optane SSD products based on 3DXPoint
  - ❑ **2019.4, 3DXPoint memory products**





- **Up to 6TB in a dual-socket server**
  - ❑ App Direct Mode
  - ❑ PMDK to map NVM to virtual address space



# Motivation

- **3DXPoint Characteristics**
  - ❑ 3DXPoint 2-3x slower than DRAM
  - ❑ 256B internal data transfer size
  - ❑ Different write content： NO impact on performance
  - ❑ Persist： can be 10x slower than normal writes
    - – CPU cache is volatile
    - – Clwb + sfence to flush data to NVM

☞ **Our goal: B+-tree on 3DXPoint memory**

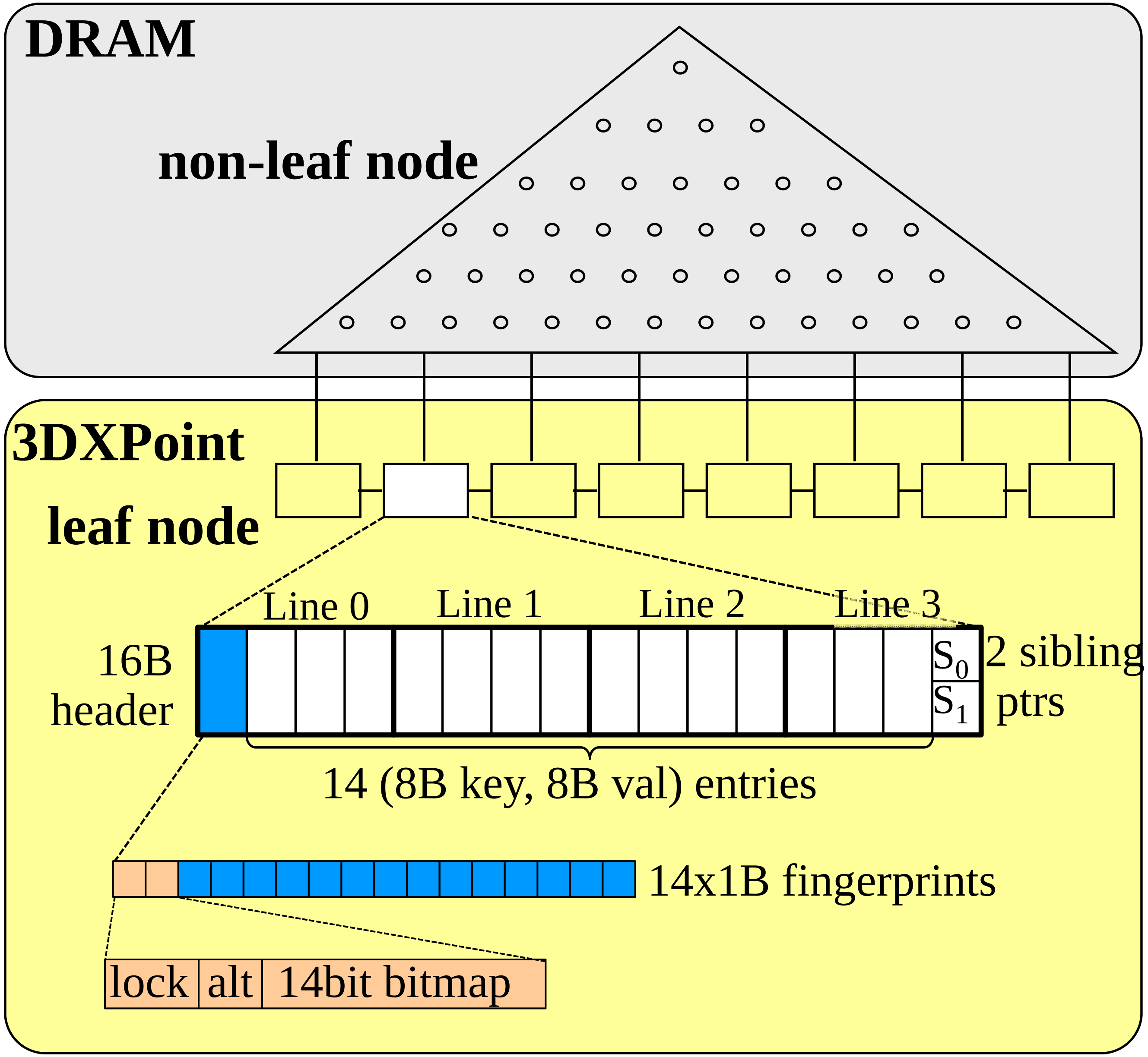
- ❑ Exploit characteristics of real NVM hardware
- ❑ Focus on insertion performance

3DXPoint performance studies:

“*Initial Experience with 3D XPoint Main Memory*”. HardBD & Active workshop, ICDE 2019

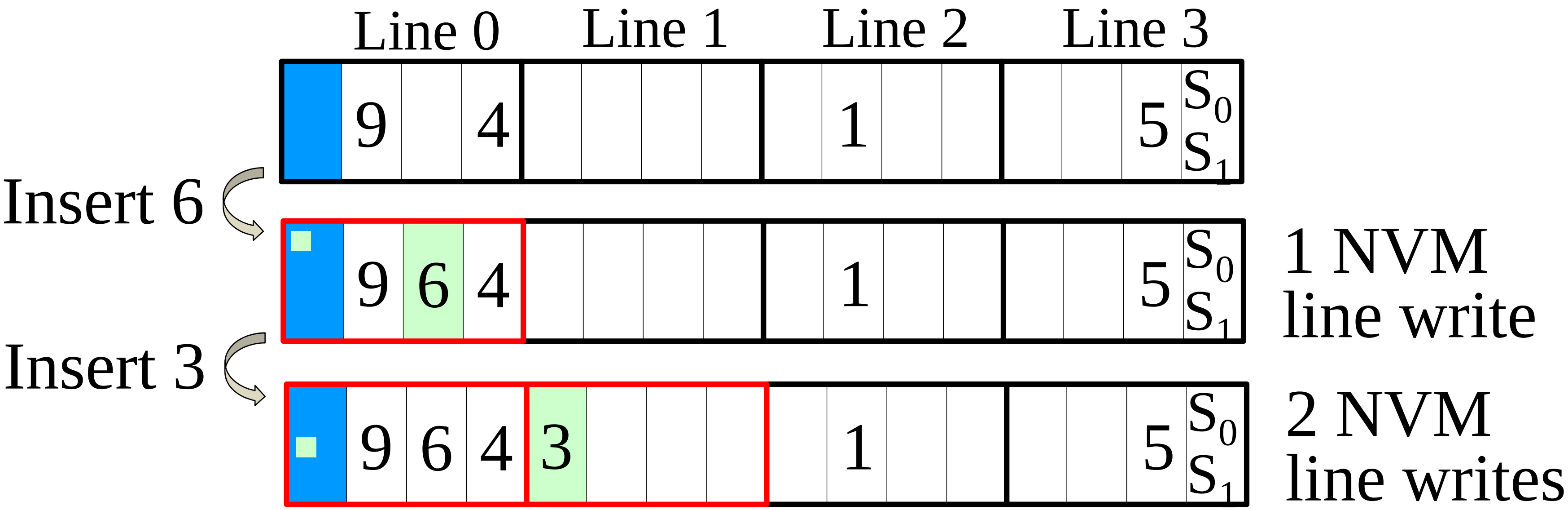
“An Empirical Guide to the Behavior and Use of Scalable Persistent Memory”. FAST 2020

# LB+-Tree with 256B Nodes



# Insertion Optimization (1)

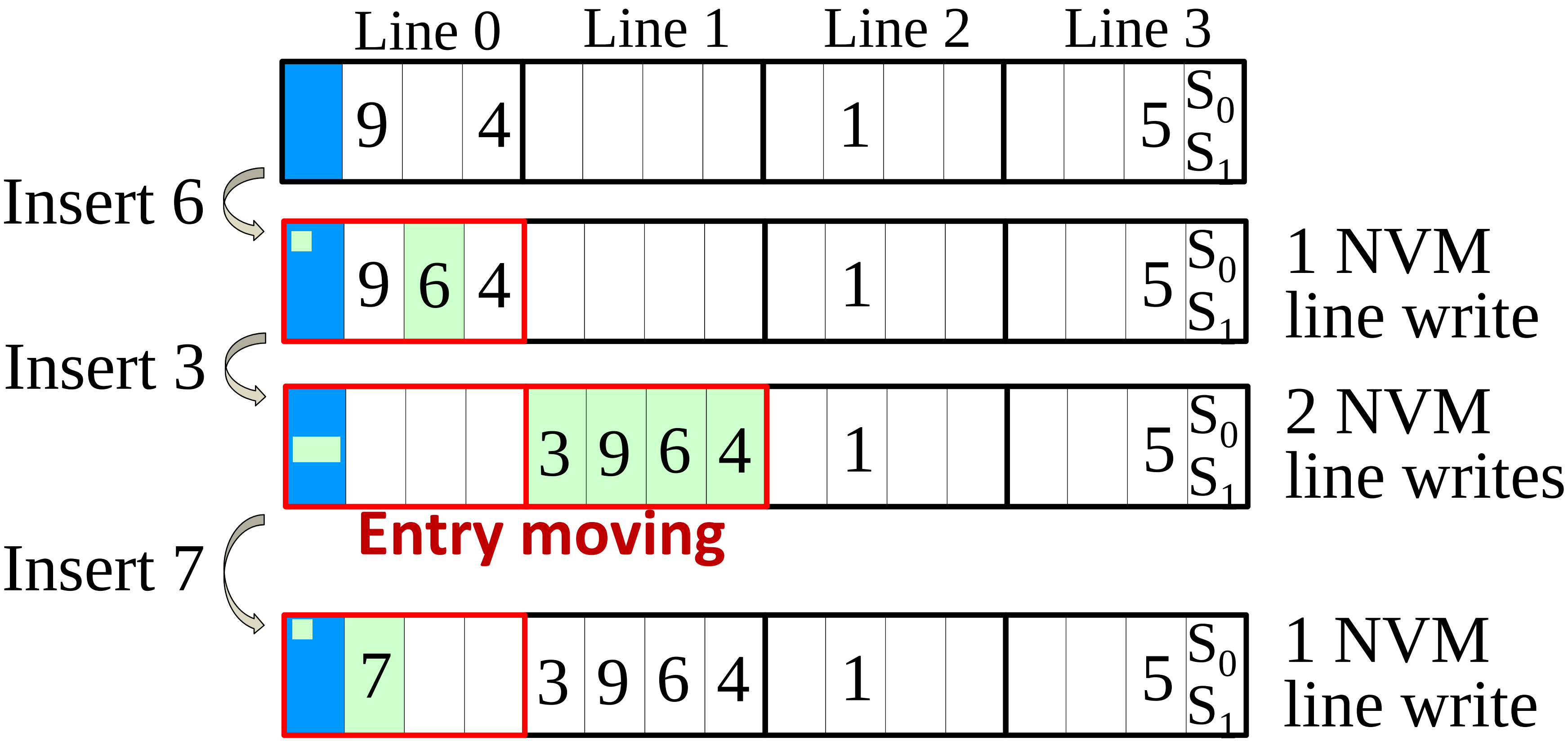
## Entry Moving




**Take this opportunity to make empty slots in Line 0**

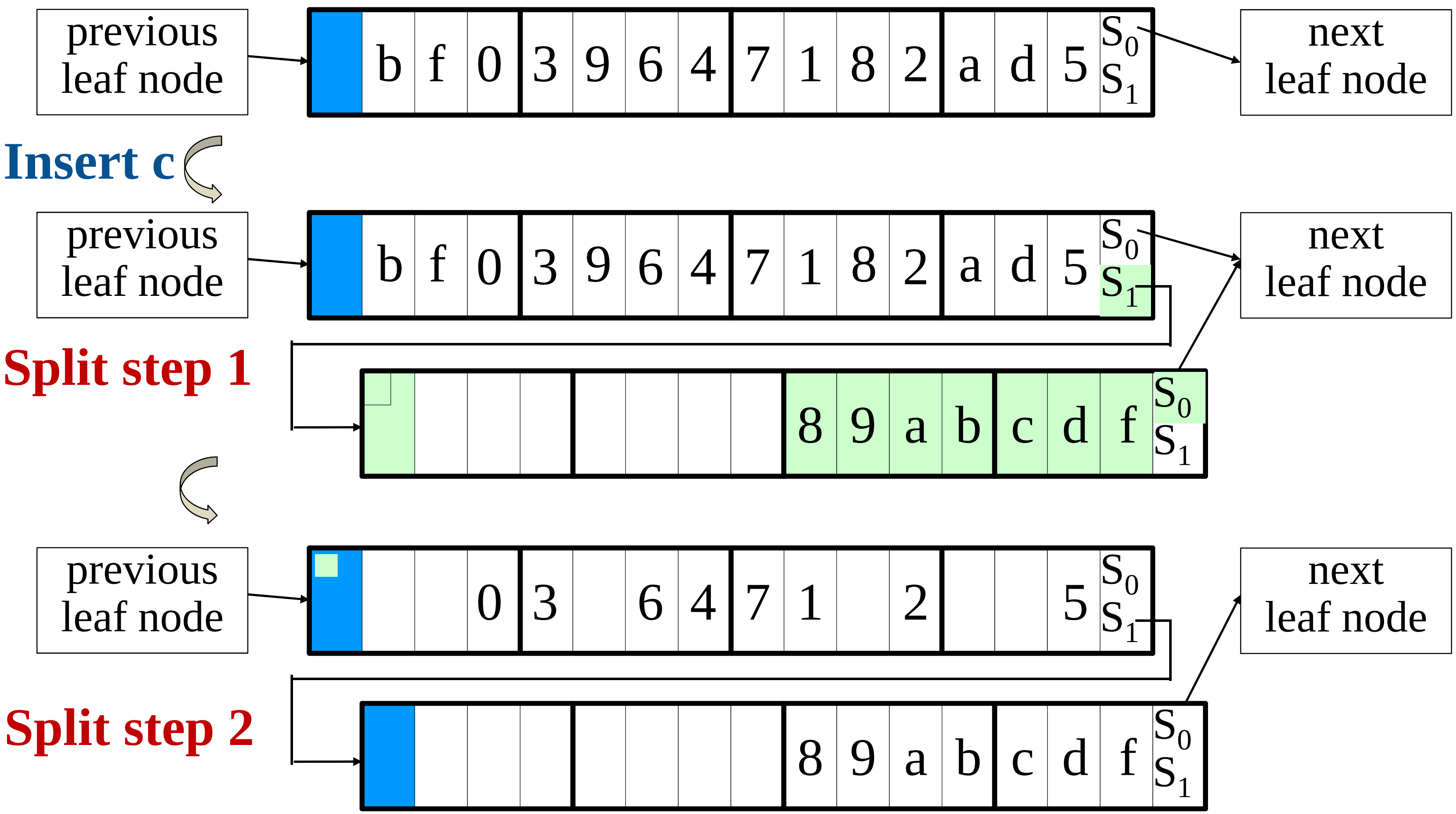
# Insertion Optimization (1)

## Entry Moving



# Insertion Optimization (2)

## Logless Node Split



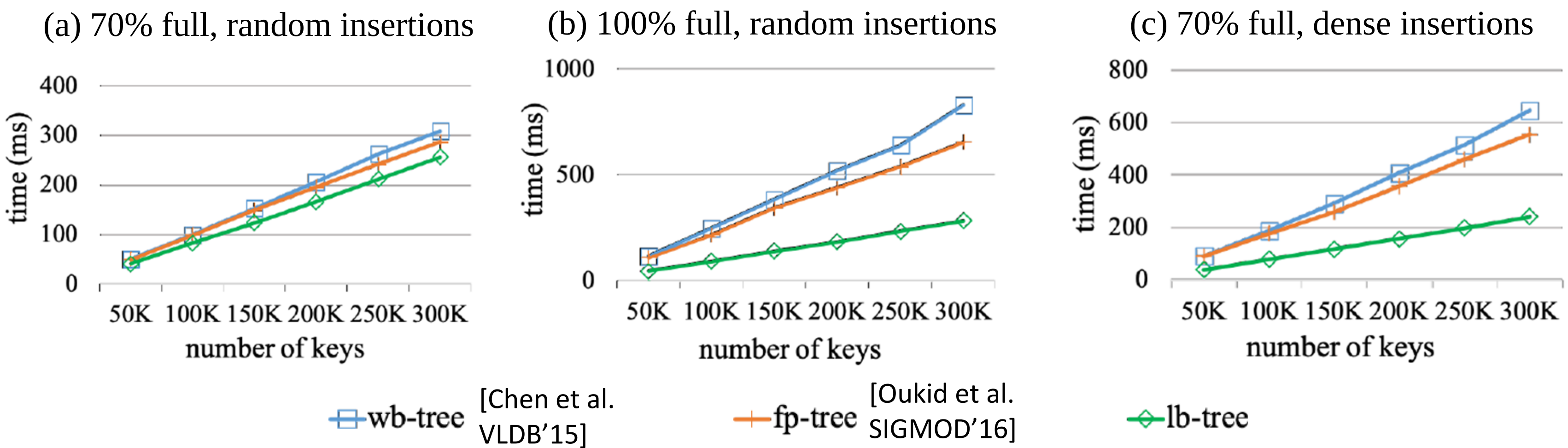
# Experiments

- **Bulkloading**
  - 70% or 100% full
  - 2 billion (8B key, 8B ptr) entries
  - Over 1/8 NVM capacity
- **Test**
  - Random insertions
  - Dense insertions

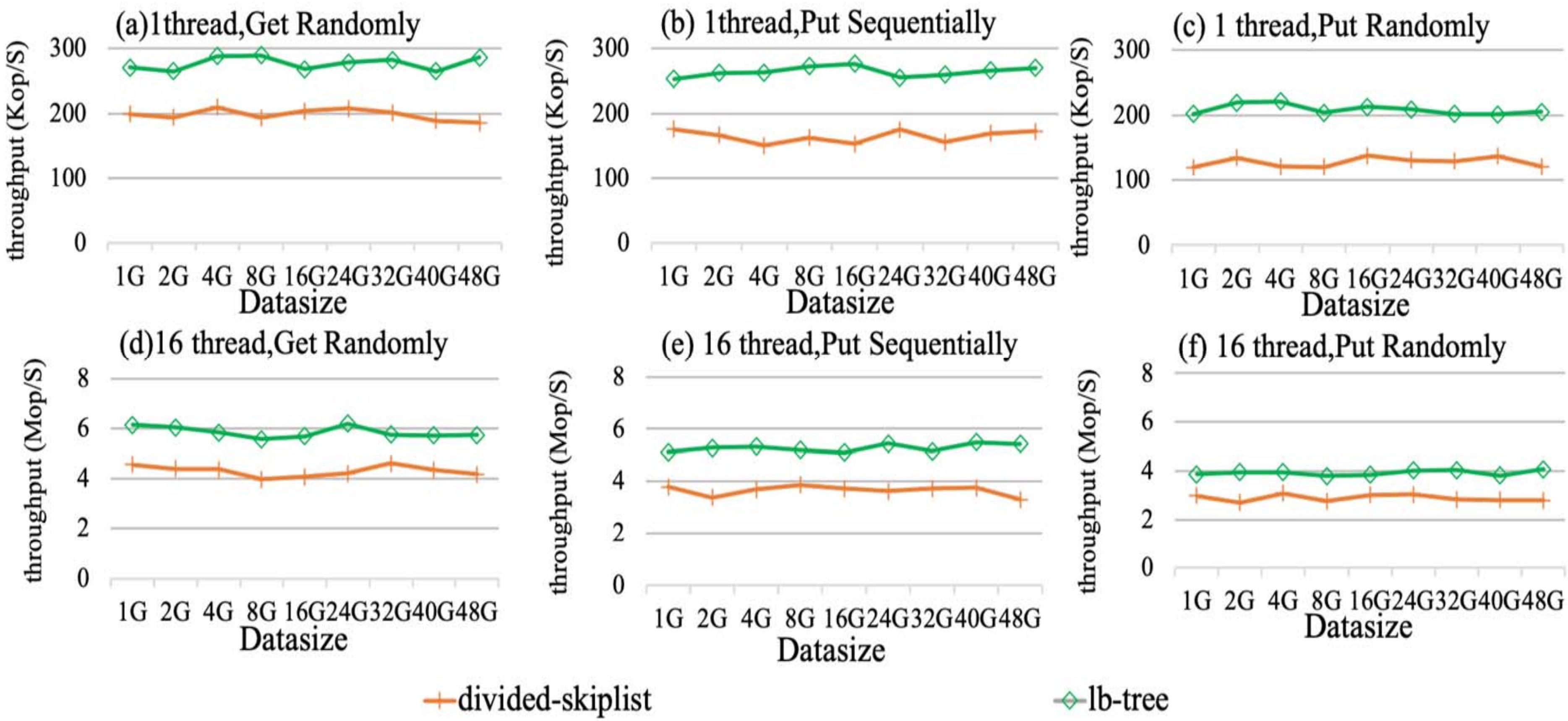
Machine Configuration

| | |
|---|---|
| CPU | Intel Cascade Lake-SP, Dual-socket, 28 cores at 2.5 GHz (Turbo Boost at 3.8GHZ) |
| L1 Cache | 32 KB iCache & 32 KB dCache (per-core) |
| L2 Cache | 1 MB (per-core) |
| L3 Cache | 39 MB (shared) |
| Total DRAM | 394 GB |
| NVMM Spec | Intel Optane DC 2666 MHz QS (000006A) |
| Total NVMM | 512 GB [2 (socket) x 2 (channel) x 128 GB] |
| Linux Kernel | 4.9.135 |
| CPUFreq Governor | Performance |
| Hyper-Threading | Disabled |
| NVDIMM | Firmware 01.01.00.5253, App direct mode |
| Power Budget | Avg. 15W, Peak 20W |

- **1.12-2.92x improvements over existing NVM optimized trees**



# Alibaba X-Engine Performance




- **LB+-Tree significantly better than skiplist**
  - ❑ 1.25—1.83x improvements

# More Details in the Paper

- LB+-Tree with multi-256B nodes
- Search, insert, delete algorithms
- Theoretical proof for entry moving benefit
- Extensive performance results

# Conclusion

- **NVM is here!**
- **NVM has different characteristics from DRAM**
  - ❑ Much larger capacity (up to 6TB for a dual-socket server)
  - ❑ 2-3x slower than DRAM
  - ❑ Large persist cost
- **LB+-Tree: a promising solution**
  - ❑ Similar read performance
  - ❑ Much better write performance

https://github.com/schencoding/lbtree

# Thank you!