Building Workload Independent Storage with VT-trees

Pradeep Shetty, Richard Spillane, Ravikant Malpani, Binesh Andrews, Justin Seyster, and Erez Zadok

Stony Brook University

http://www.fsl.cs.stonybrook.edu/

STONY BROOK STATE UNIVERSITY OF NEW YORK

File Systems Should…

- Keeps the data persistent
- Provides crash recovery
- Performs efficiently
  - Sequential reads and writes
    - Large files: video files, photos
  - Random reads and writes
    - Small files or tuples: file metadata, mp3 tags
- Provides low overhead application level transactions

STONY BROOK STATE UNIVERSITY OF NEW YORK

Existing File Systems

Data Structure	SeqRead	SeqWrite	RandRead	RandWrite
(Extents + Block tree)				
LFS				
COW-Btree				
LSM-tree				

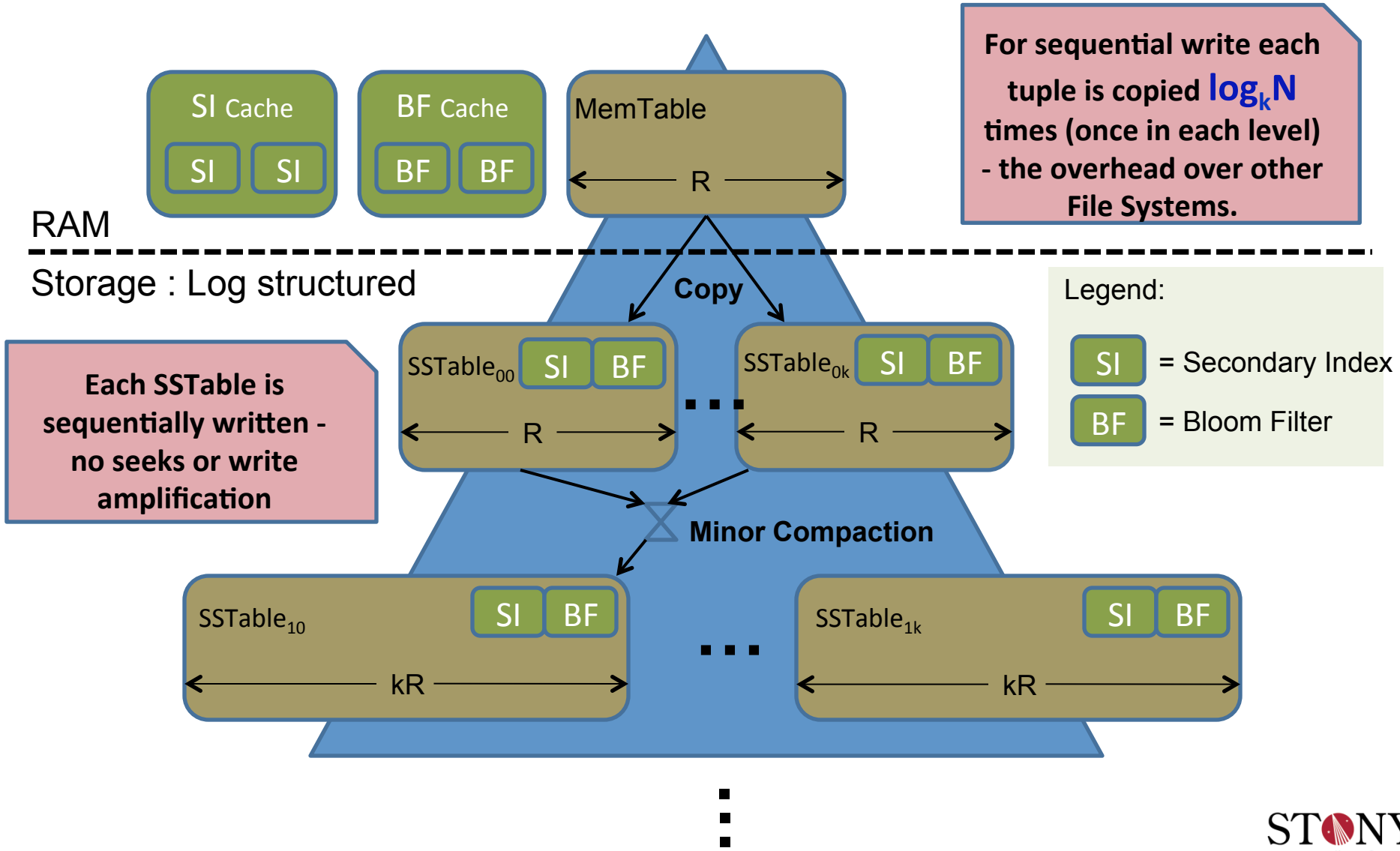
STONY BROOK STATE UNIVERSITY OF NEW YORK

Our Solution

- Developed a single data structure
  - VT-trees - A novel extension to LSM-trees
  - Performs comparable or superior to other file systems on all the four workloads
  - Lends itself to an efficient transactional architecture
  - Flexible:
    - Span continuum from LSM-trees to LFSs

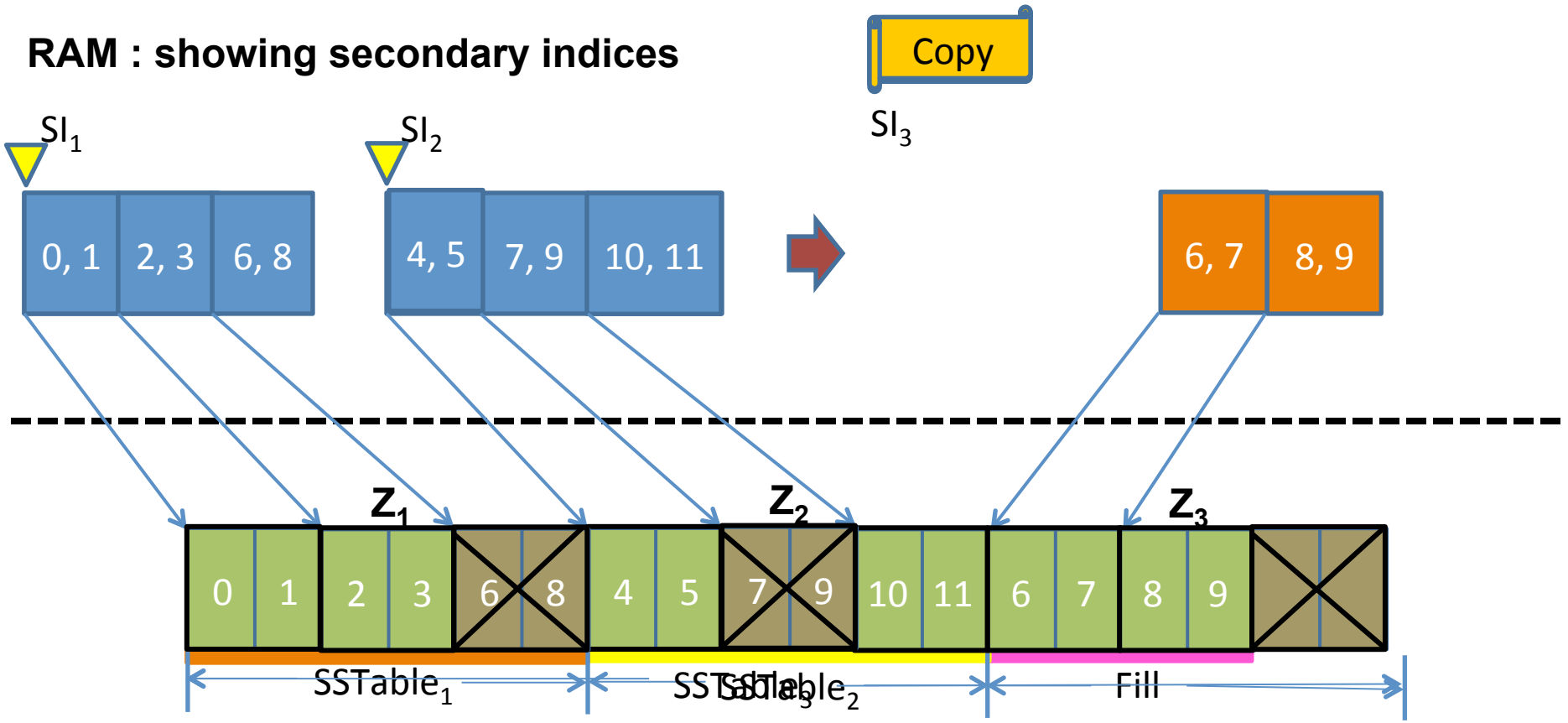
STONY BROOK STATE UNIVERSITY OF NEW YORK

Traditional LSM Tree



STONY BROOK STATE UNIVERSITY OF NEW YORK

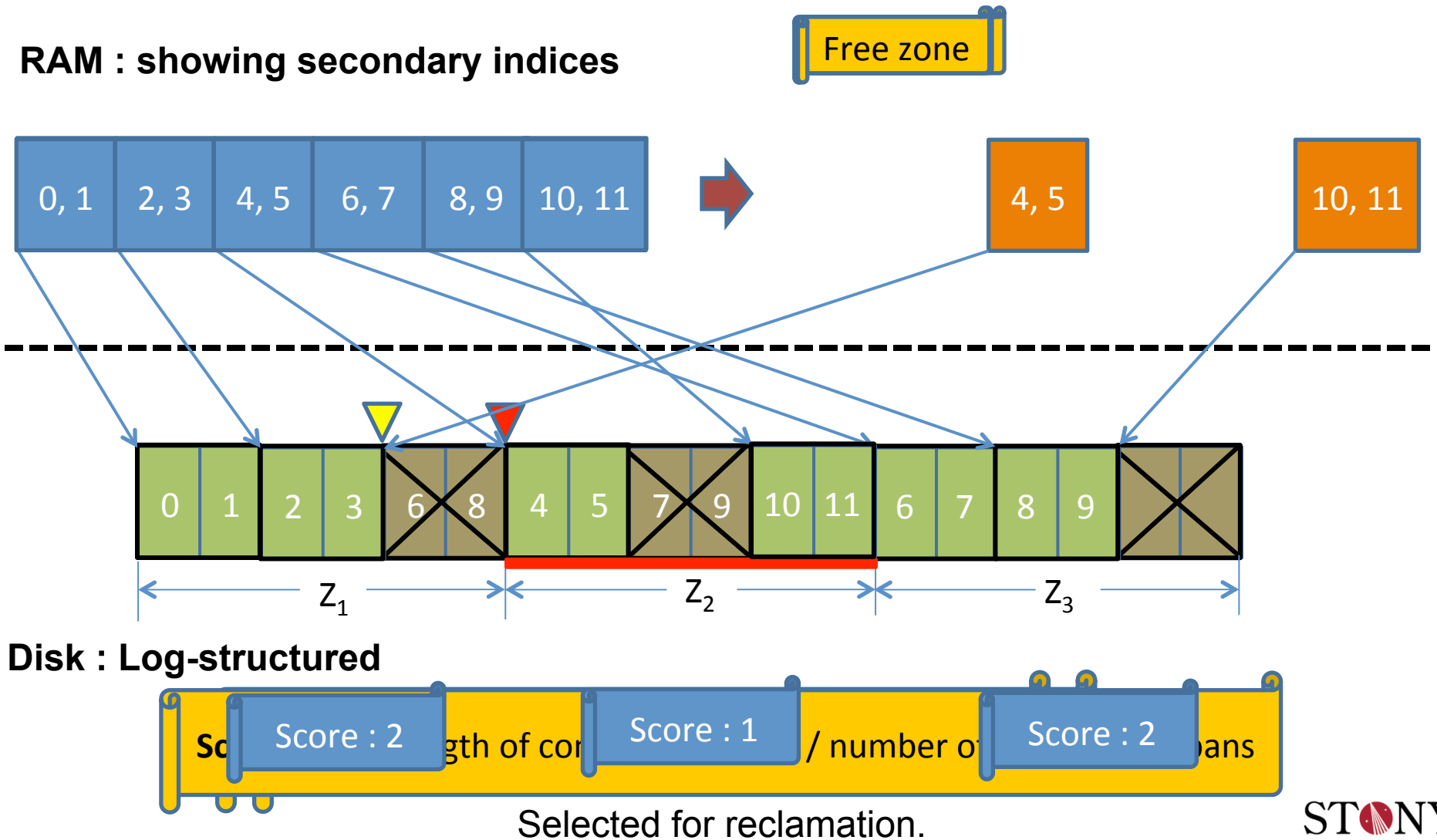
VT-trees and Stitching



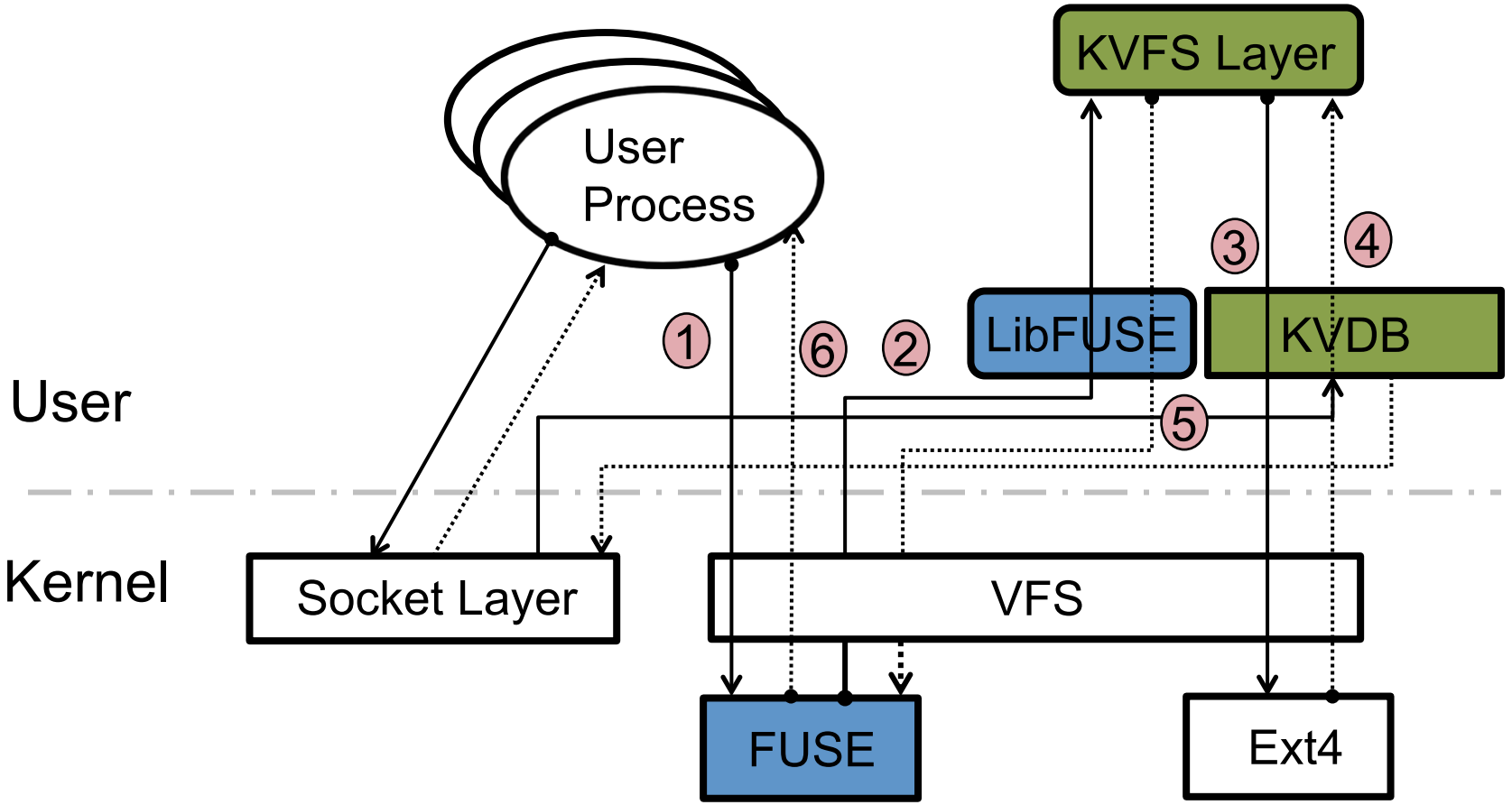
Stitching threshold = N contiguous non-overlapping blocks

STONY BROOK STATE UNIVERSITY OF NEW YORK

Defragmentation



KVFS Architecture



STONY BROOK STATE UNIVERSITY OF NEW YORK

Schema for File System

- nmap:
  - Name space entries, similar to dentries

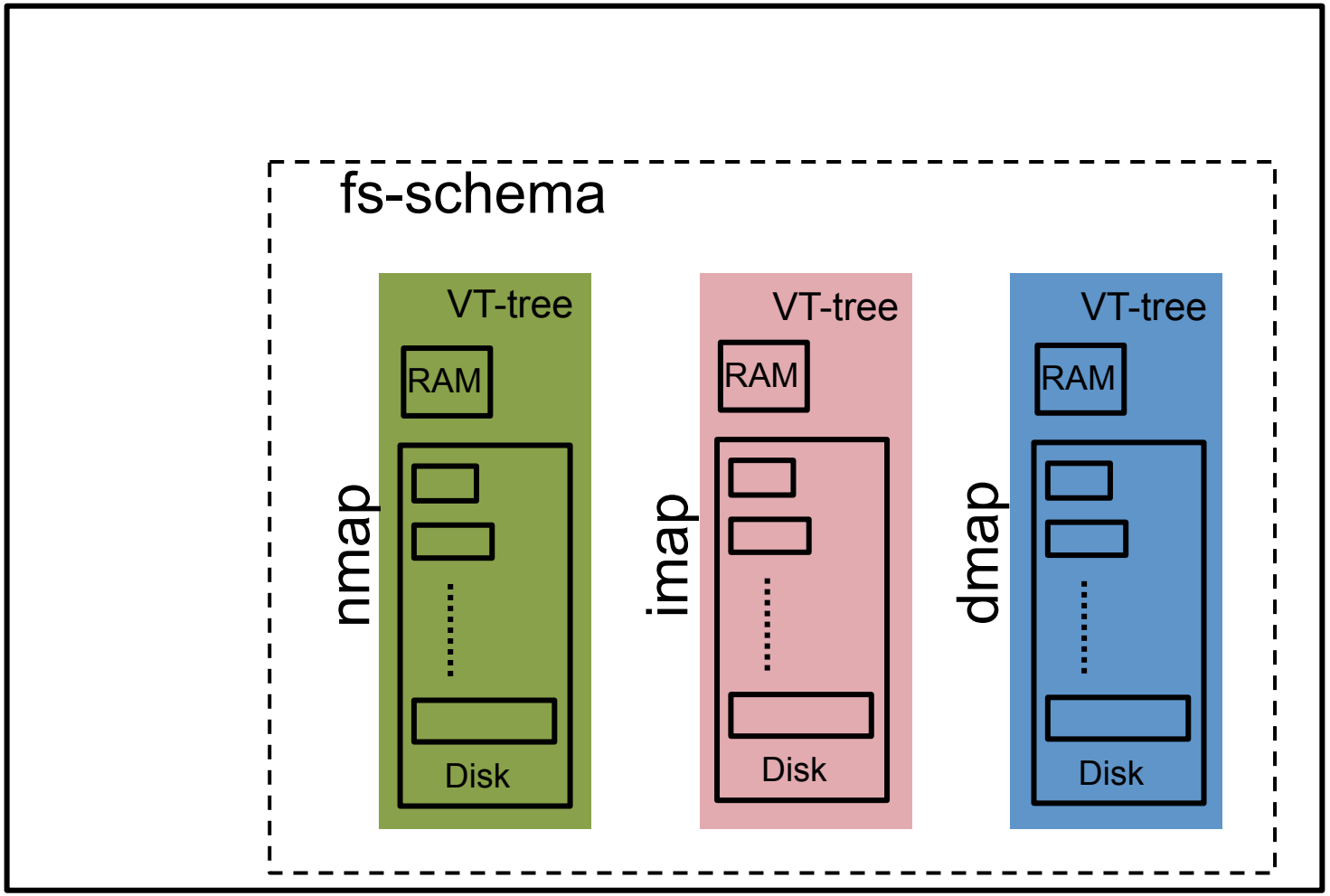
  ⟨{parent-inode#, path-component}, inode#⟩
- imap: (meta-data)
  - Inode attributes

  ⟨inode#, inode attributes⟩
- dmap: (data)
  - Files' data blocks

  ⟨{inode#, offset}, data-block⟩

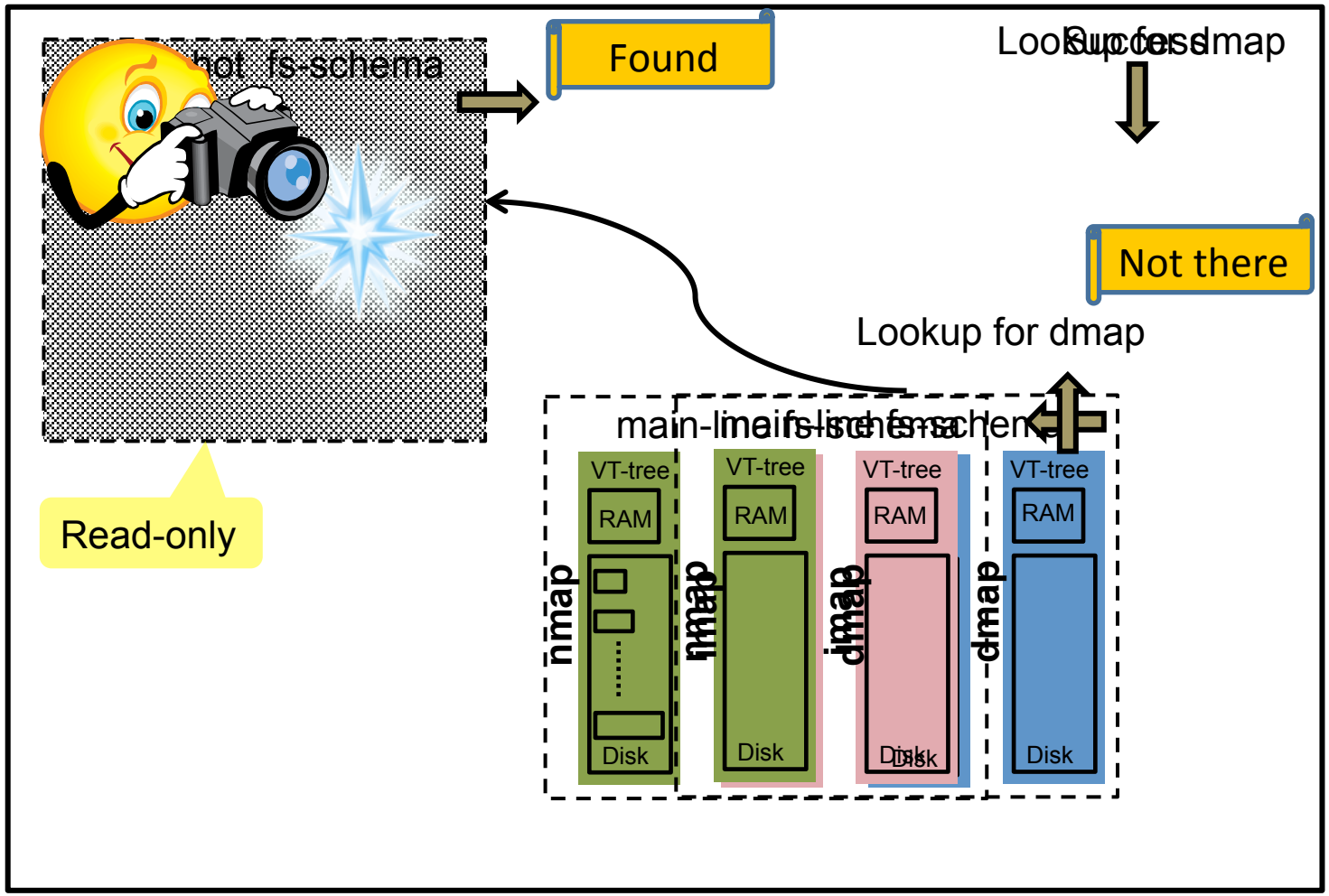
STONY BROOK STATE UNIVERSITY OF NEW YORK

KVDB Architecture



STONY BROOK STATE UNIVERSITY OF NEW YORK

Snapshots



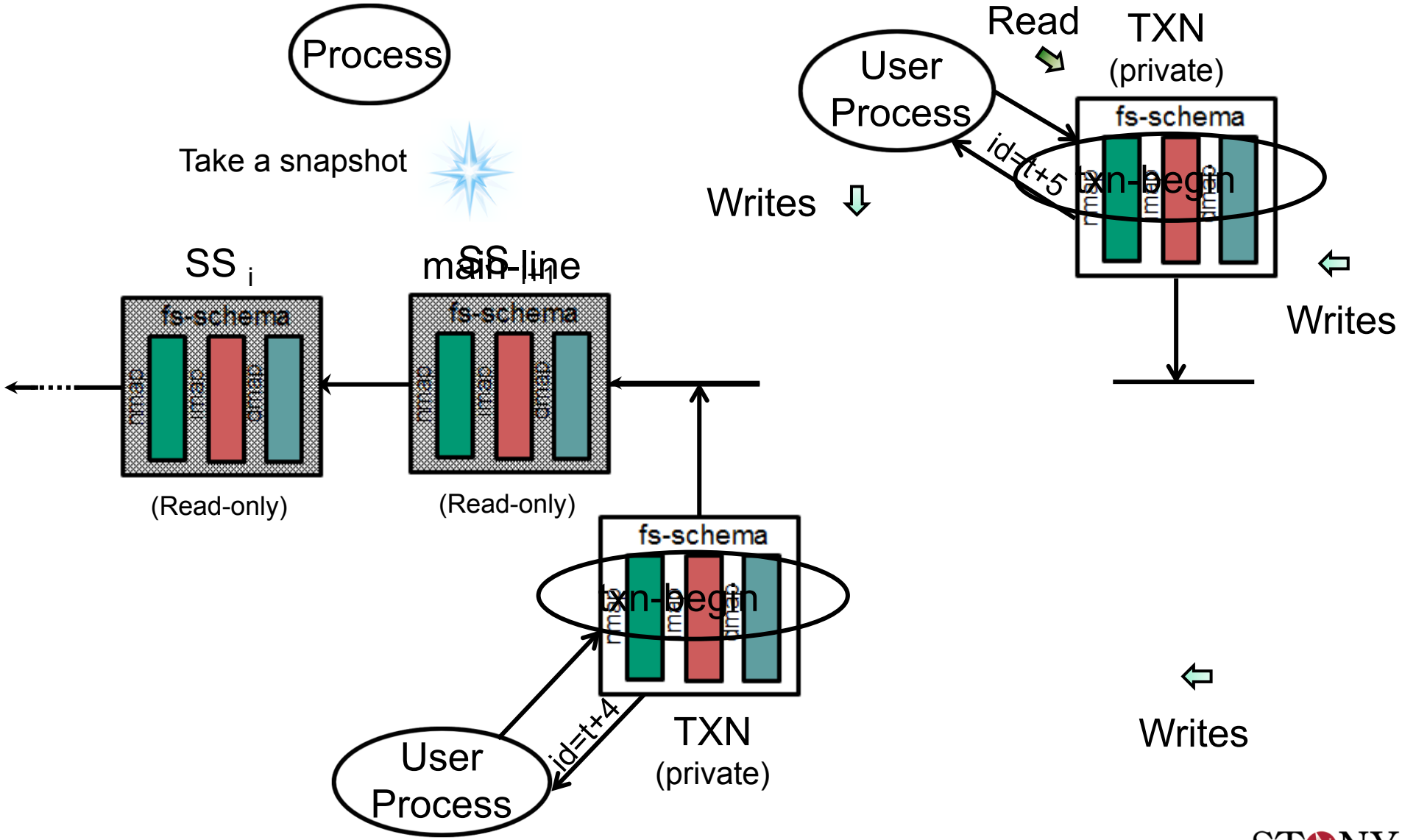
STONY BROOK
STATE UNIVERSITY OF NEW YORK

Transactions

- VT-tree is inherently transactional
- Snapshot based, larger-than-RAM
  - Each transaction gets its own private snapshot backed by VT-trees
- Avoids double writes, only redo images
- txn-begin, txn-commit, txn-abort
- ACID with two modes of isolation
  - Snapshot isolation
  - Serializability
    - Strongest

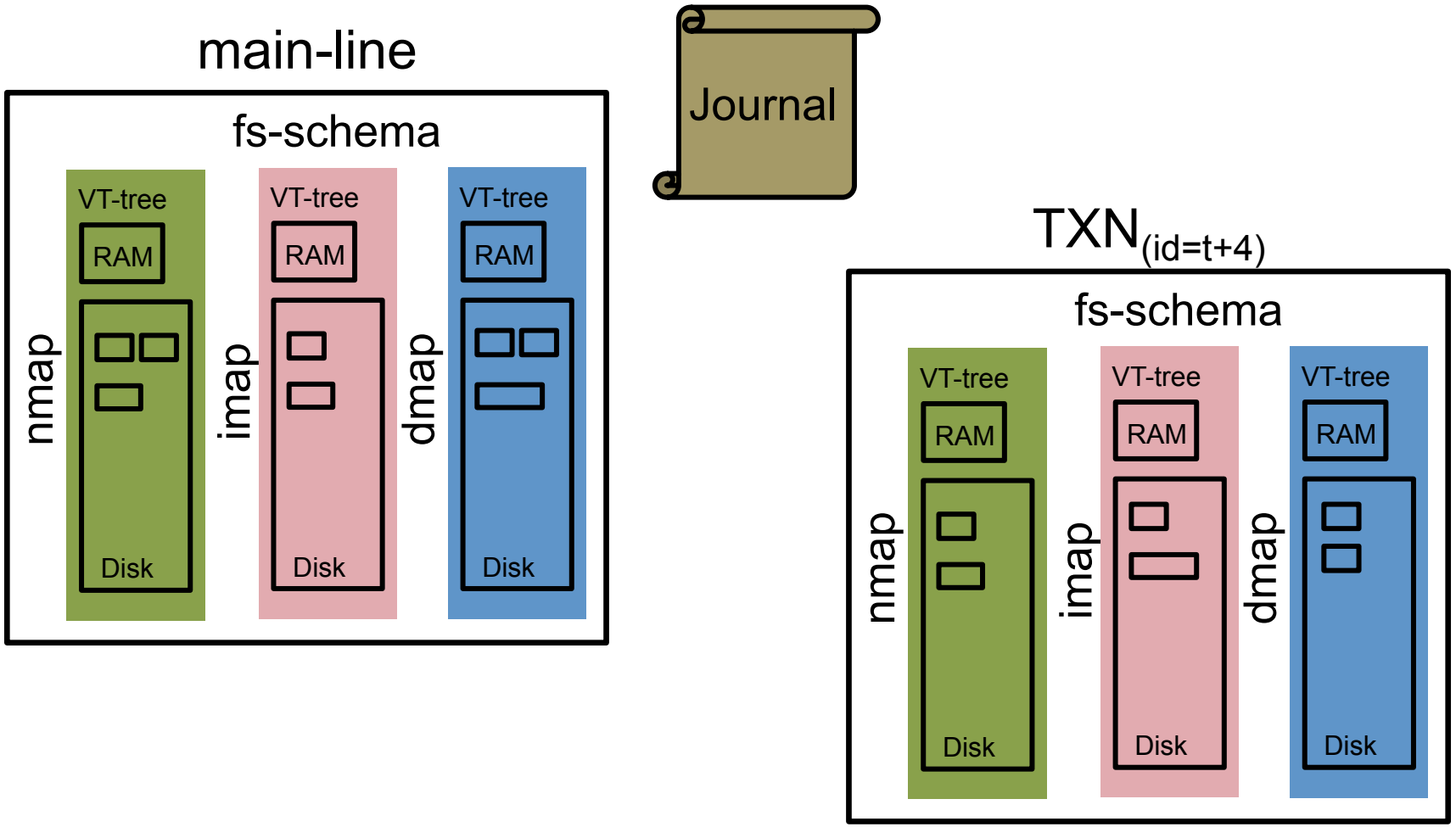
STONY BROOK STATE UNIVERSITY OF NEW YORK

Serializability



STONY BROOK STATE UNIVERSITY OF NEW YORK

Transaction Commit



Experimental Setup

- Two Intel Xeon X5650
  - 4 cores, 2.66GHz, 12MB L2
- Booted with 3GB RAM
- 160GB SATAII Intel X25-M SSD, 250GB 7.2KRPM SATA disk
- Measured throughput of the SSD
  - Random read - 15,000 ops/s
  - Random write - 3,500 ops/s
  - Sequential read - 245MB/s
  - Sequential write - 107MB/s

STONY BROOK STATE UNIVERSITY OF NEW YORK

Stitching - Experimental Setup

- 8MB Memtable
- SSTables start off at 8MB large
- Working sets at least 3.3x RAM
- No transactions, just the data structures

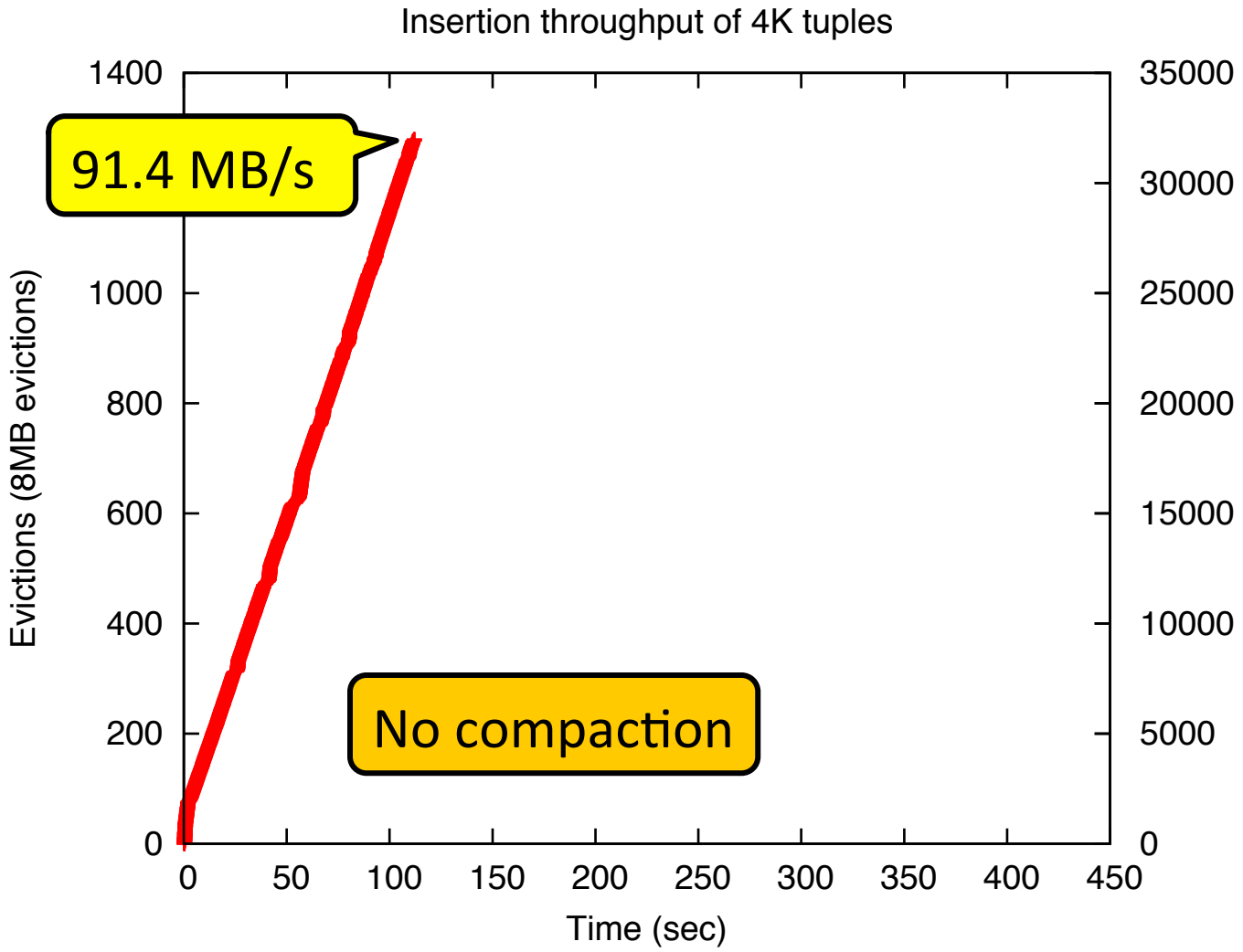
STONY BROOK STATE UNIVERSITY OF NEW YORK

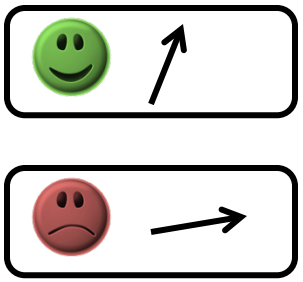
Stitching - Random Append

- We randomly append to 2,040 “files”
- Each append is 4KB
- Configurations
  - no-compact just appends 8MB memtables
  - f=0 always stitches
  - f=3, f=15 stitch progressively more
  - f=16383 is equivalent to full compaction
- We expect to trade off scans for insertions

STONY BROOK STATE UNIVERSITY OF NEW YORK

Random Append

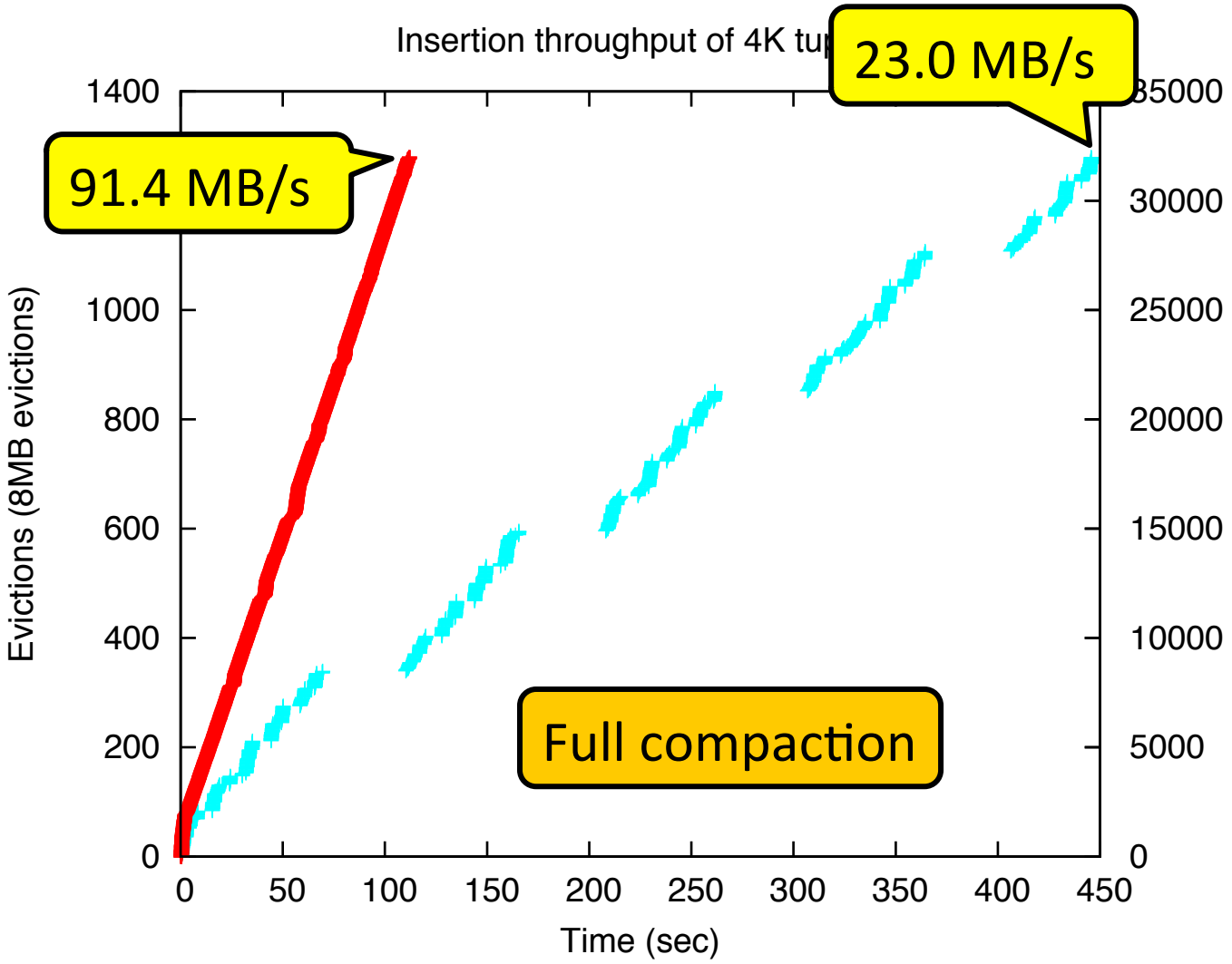


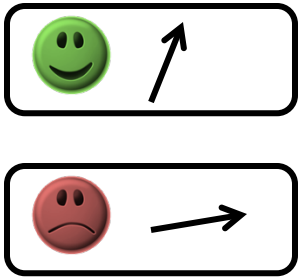


+ No compaction
+ f = 0
+ f = 3
+ f = 15
+ f = 16,383

STONY BROOK STATE UNIVERSITY OF NEW YORK

Random Append

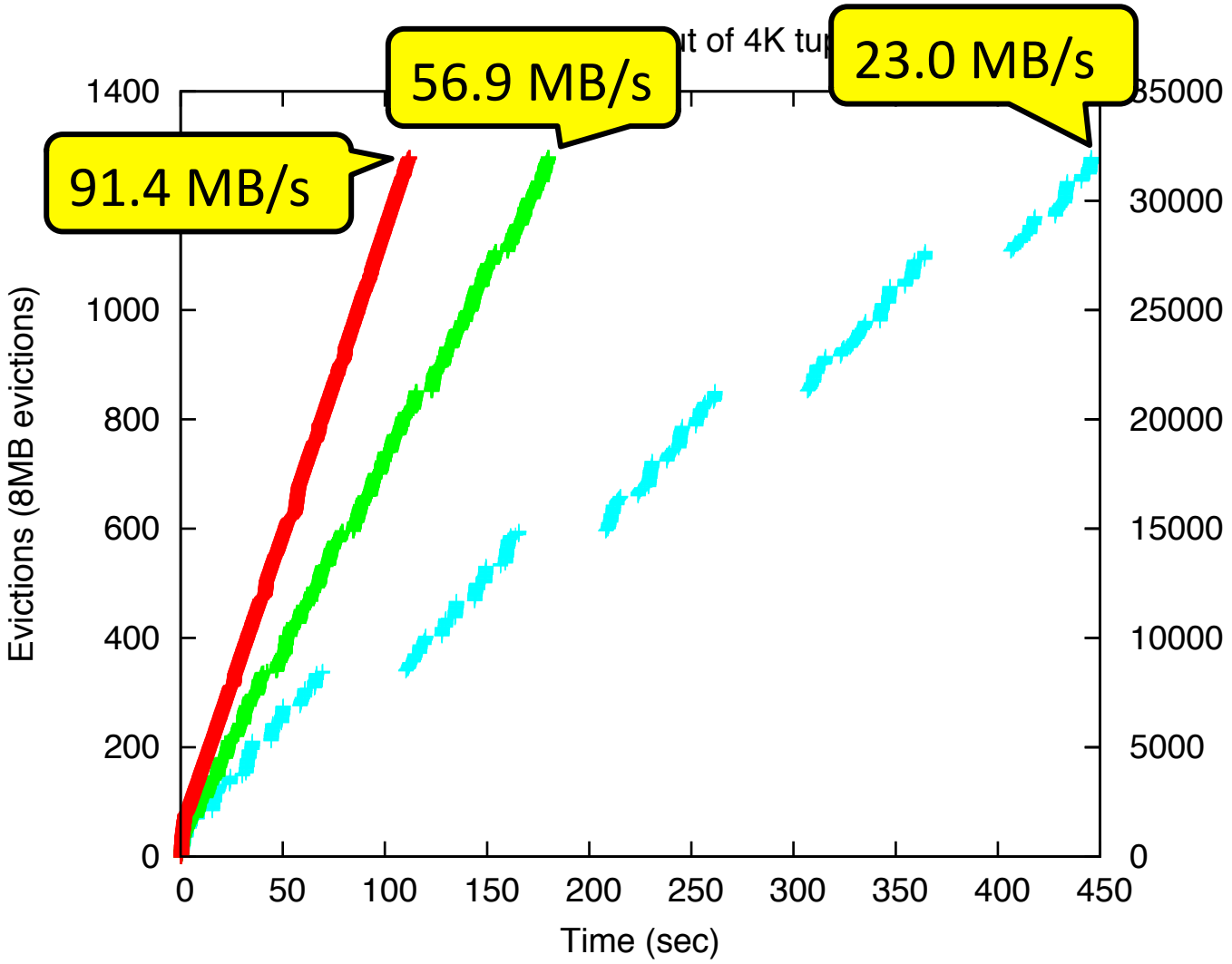


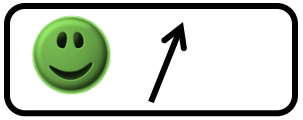


+ No compaction
+ f = 0
+ f = 3
+ f = 15
+ f = 16,383

STONY BROOK STATE UNIVERSITY OF NEW YORK

Random Append



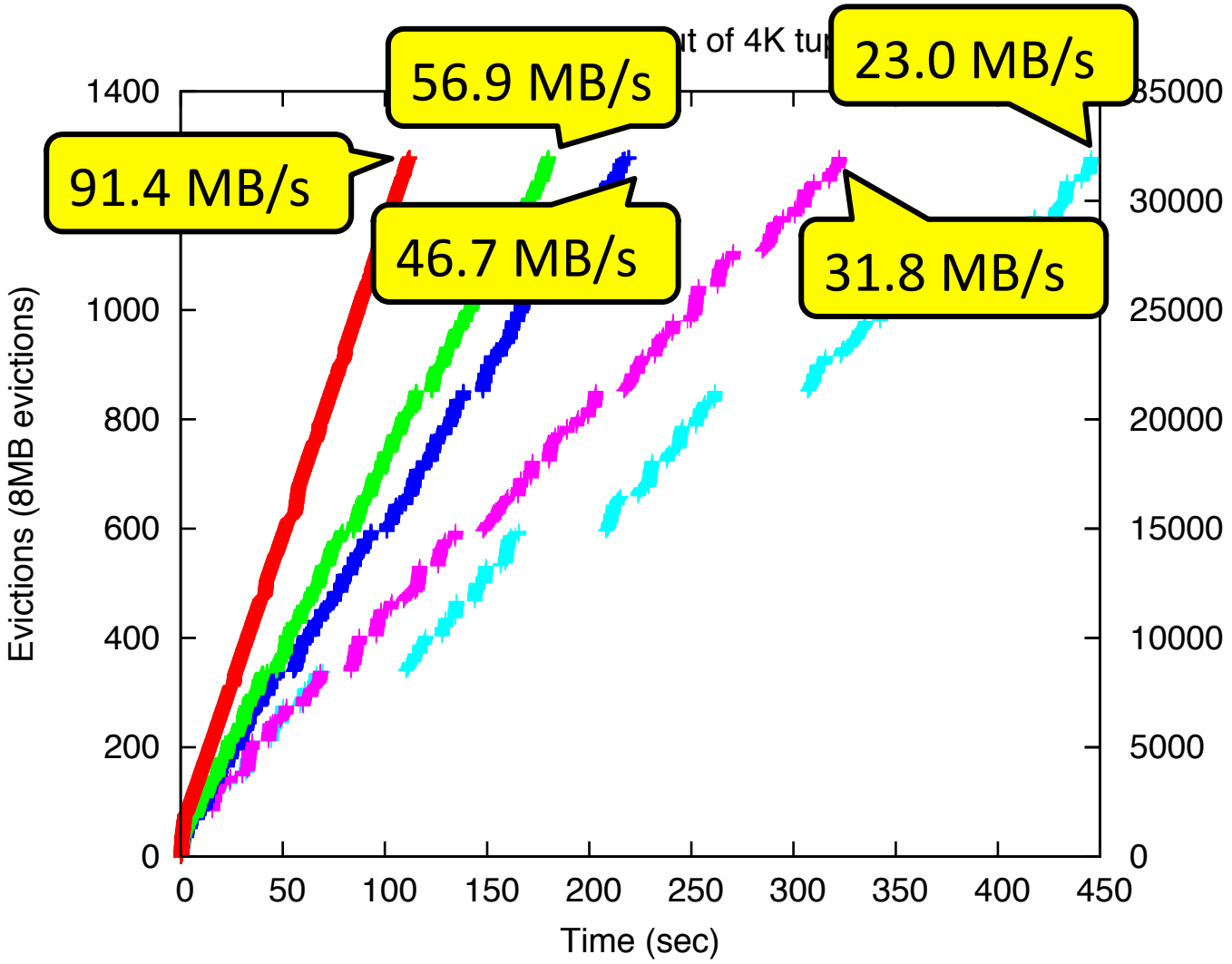


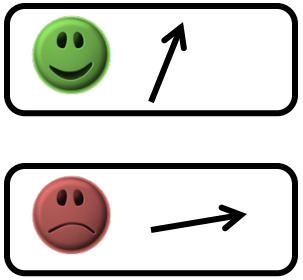


+ No compaction
+ f = 0
+ f = 3
+ f = 15
+ f = 16,383

STONY BROOK STATE UNIVERSITY OF NEW YORK

Random Append





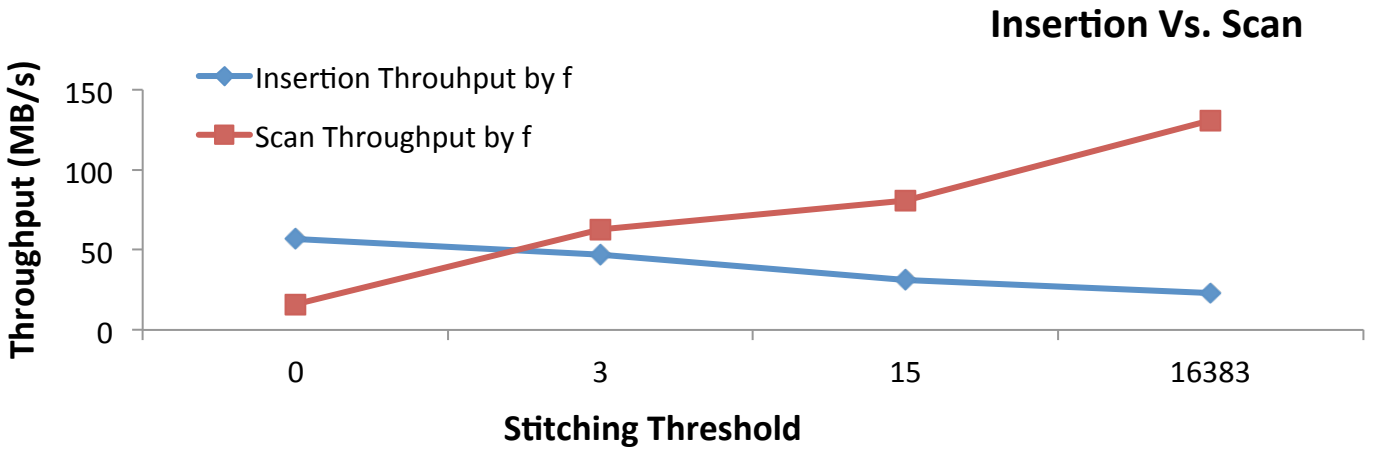
+ No compaction
+ f = 0
+ f = 3
+ f = 15
+ f = 16,383

STONY BROOK STATE UNIVERSITY OF NEW YORK

Insertion/Scan Trade-Off

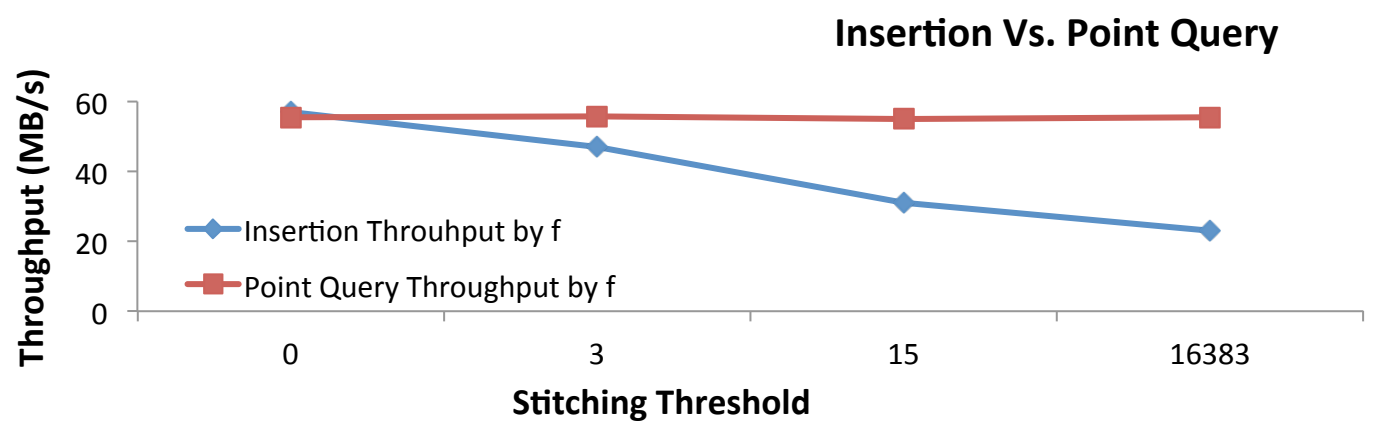






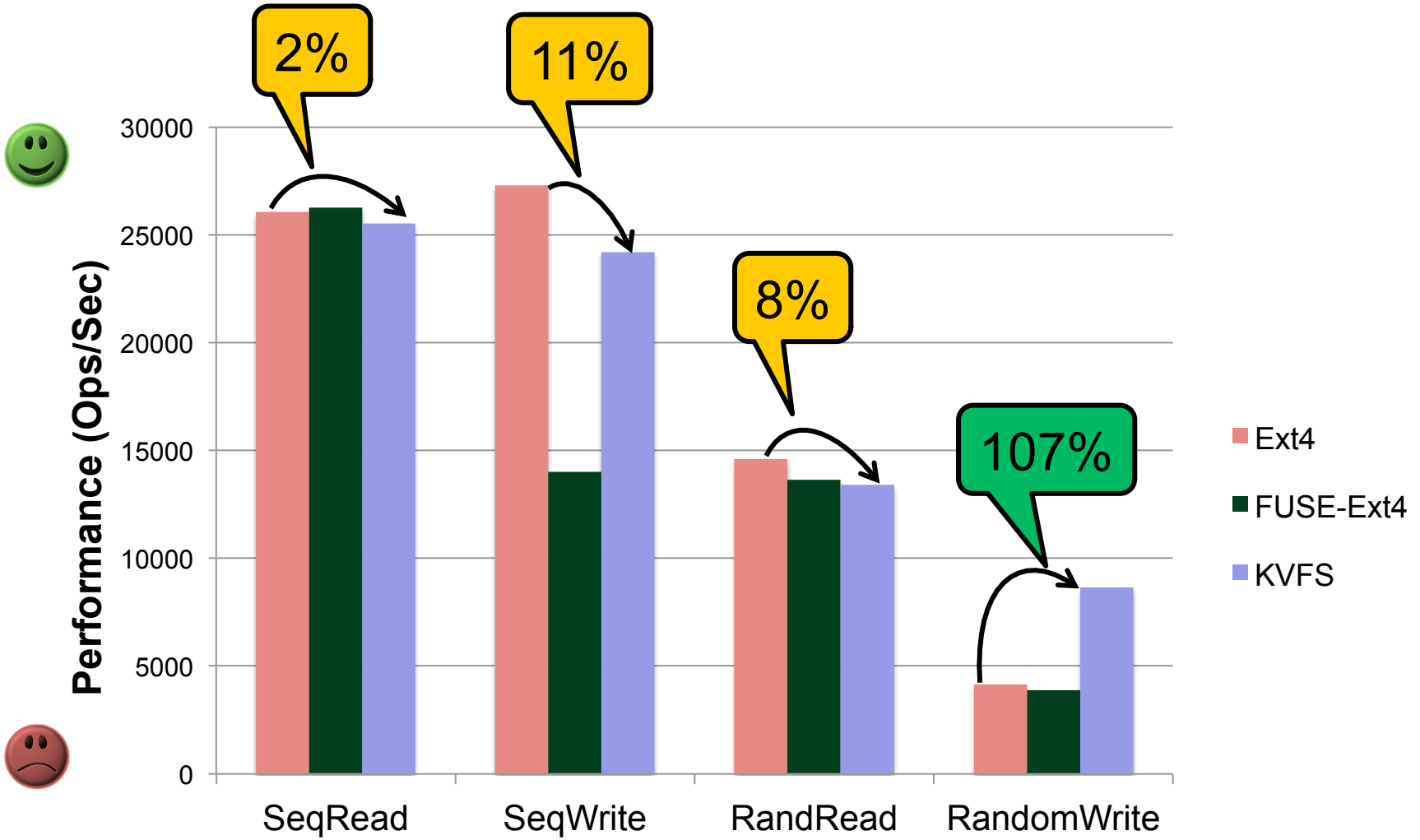






STONY BROOK STATE UNIVERSITY OF NEW YORK

Micro Benchmark on KVFS



File size: 30GB, I/O size: 4KB

STONY BROOK STATE UNIVERSITY OF NEW YORK

Other Experiments

- KVDB vs. LevelDB
  - KVDB 23% faster for small and 6x for large tuples
- File system macro benchmark
  - Fileserver (22% faster), Webserver (7% slow)
- Transactional Performance
  - 10% overhead (strong isolation), 4% faster (Non-overlapping parallel txns)
- Evaluating defragmentation

STONY BROOK STATE UNIVERSITY OF NEW YORK

Conclusions

- VT-trees performs well for all the four workloads
- KVFS performs comparable or superior to Ext4 for most of the workloads
  - Suitable for desktops and servers
- Transactions in KVFS are highly concurrent and has minimal overhead
- Defragmentation works as a complimentary to Stitching

STONY BROOK STATE UNIVERSITY OF NEW YORK

Q&A

pradeepjshetty@gmail.com

Pradeep J. Shetty

Stony Brook University

http://www.fsl.cs.sunysb.edu/

STONY BROOK
STATE UNIVERSITY OF NEW YORK

Related Work

- Write-optimized databases
  - GTSSL, Anvil, BDBJE
- Log-structured file systems
  - LFS, ZFS, WAFL
- Transactional file systems
  - Amino, Valor, TxOS, TxF, QuickSilver
- Indexing Systems
  - TokuFS, Spyglass

STONY BROOK STATE UNIVERSITY OF NEW YORK

Future Work

- Online defragmenting, try other techniques as well
- Modify and evaluate applications like iTunes.
- Support in-line deduplication in KVFS

STONY BROOK STATE UNIVERSITY OF NEW YORK