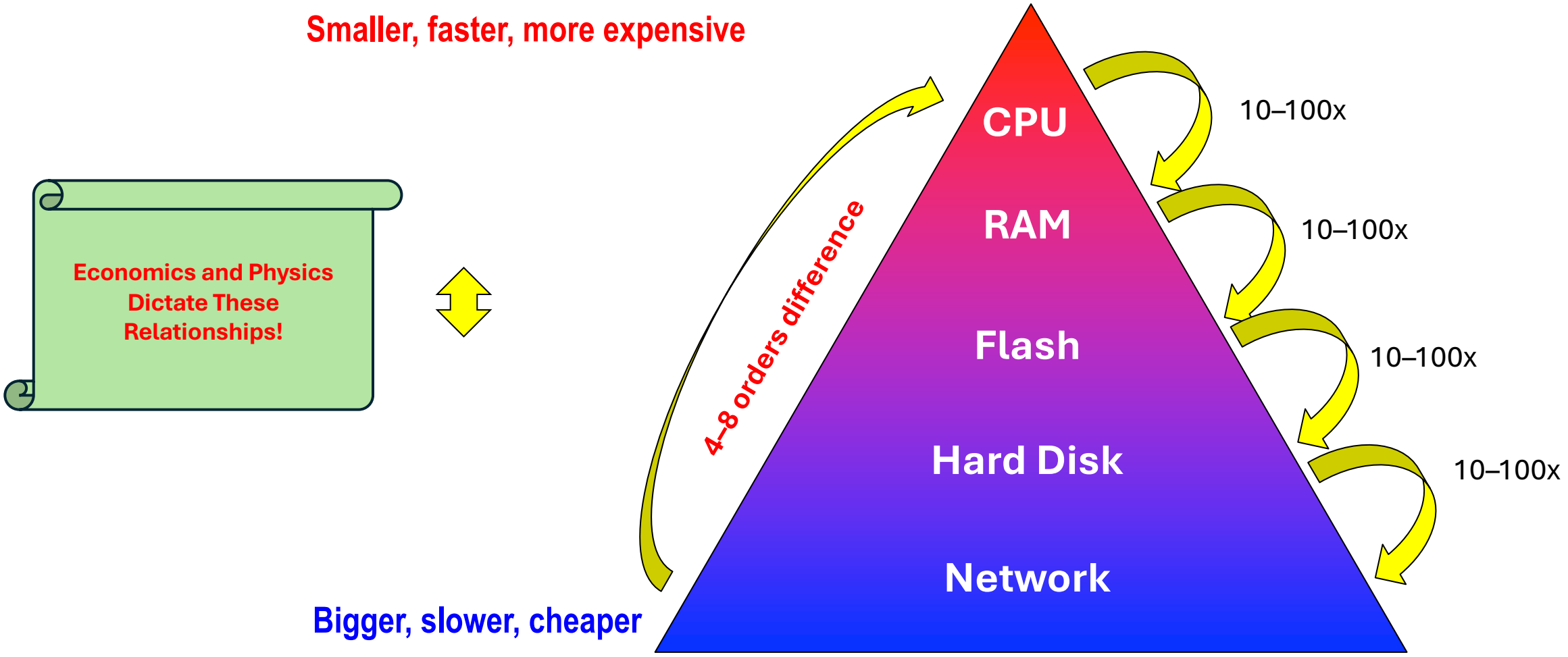
Everything You Always Wanted to Know about Storage Analysis

(But Were Afraid to Ask ;)

Erez Zadok
Stony Brook University

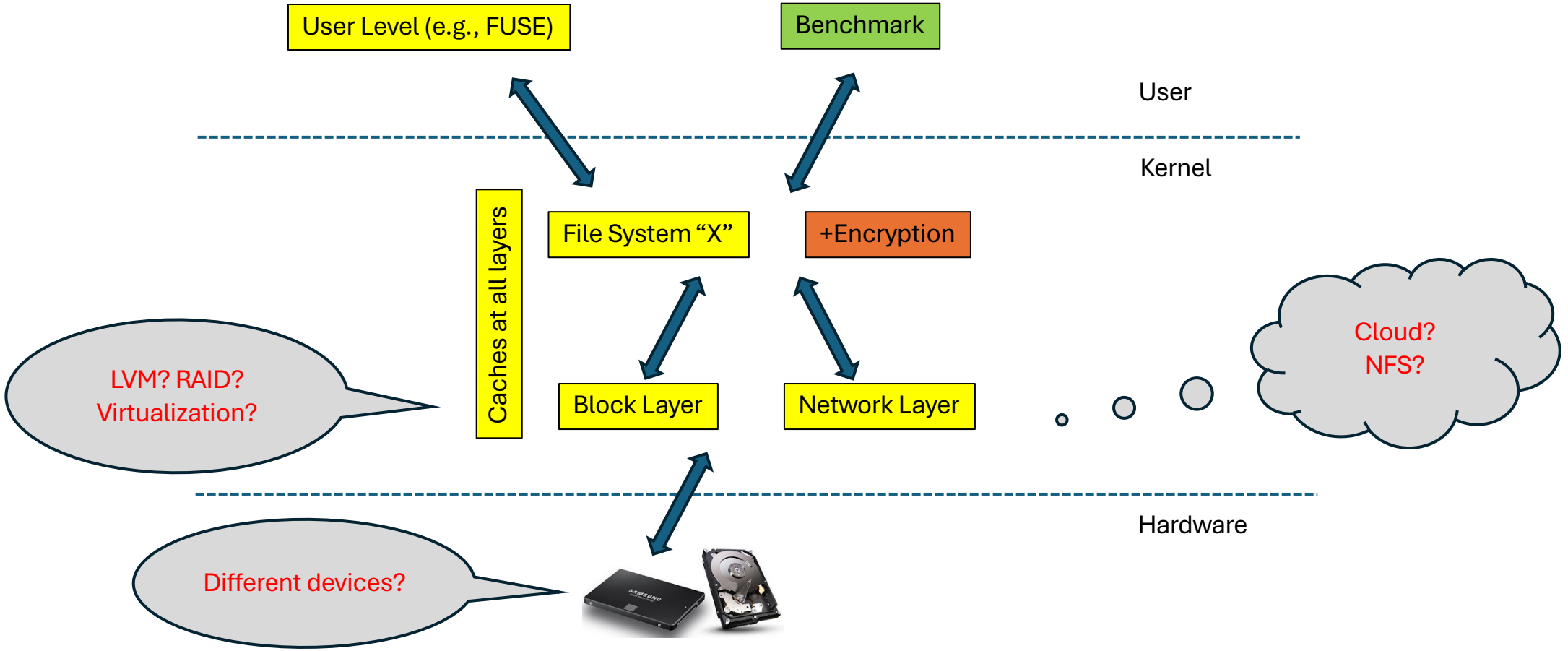
Storage is Complex,
Data is Everywhere,
Life is Good.

The Storage Hierarchy Pyramid



Stony Brook University

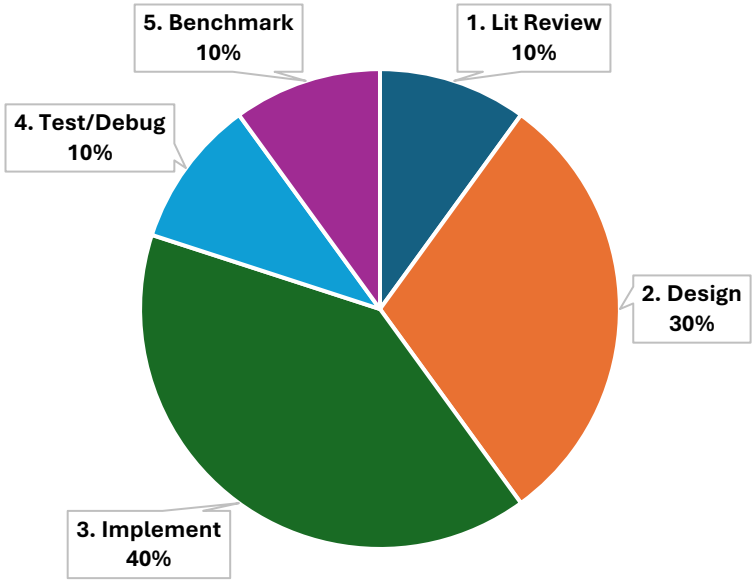
Deep Storage Stack, Data Everywhere



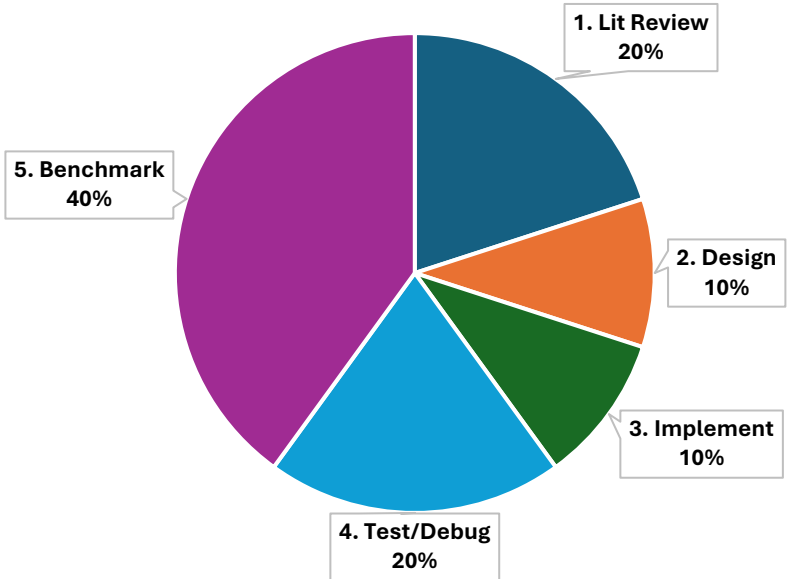
How to isolate the impact of each component on performance?

Storage Benchmarking is Time Consuming

Typical Process



Better (More Realistic) Process



Benchmarking reveals:

- Bugs (crash)
- Performance bottlenecks
- Implementation flaws
- Design flaws

Other:

- Hardening
- Regressions
- Code/Data Release

Stony Brook University

So, You Want to be a Storage Professional?

What Tools to Benchmark?

A. Micro-Benchmarks? (e.g., FIO, dd, many more)
   i. Useful to test specific features (e.g., added encryption)
   ii. Can evaluate worst-case behavior, isolate features/components
   iii. But, not representative of "real world" workloads

B. Macro-benchmarks? (e.g., Filebench, RocksDB/LevelDB, many more)
   i. Useful to evaluate more realistic workloads
   ii. But, still synthetically generated

C. Trace Replay? (e.g., SNIA IOTTA trace repository)
   i. Considered most realistic, based on actual system traces
   ii. But: traces can be stale, take long time to replay (e.g., clock-time, AFAP?)

☛ ***Answer: "D" – All of the above, but justify***

How Many Parameters to Vary?

Type	Values	Number of Tests
Benchmark	Micro, Macro, Traces	3
Workloads	Sequential, random, mixed	3
Workload/Cache size	Small, medium/default, large	3
No. of Systems	At least two CPUs	2
No. of Threads	1, 4, 16	3
Custom (hash, cipher, key sizes)	5 (?)	5
No. of runs per experiment	5	5
Total No. of Experiments		**4,050**
Total time to benchmark	**If 1 min per experiment**	**2.8 days**
Total time to benchmark	**If 2 min per experiment**	**5.6 days**
Total time to benchmark	**If 15 min per experiment**	**42 days**

These are conservative values!

The Dark, Often Ignored Art of Presentation

Presenting Data (1)

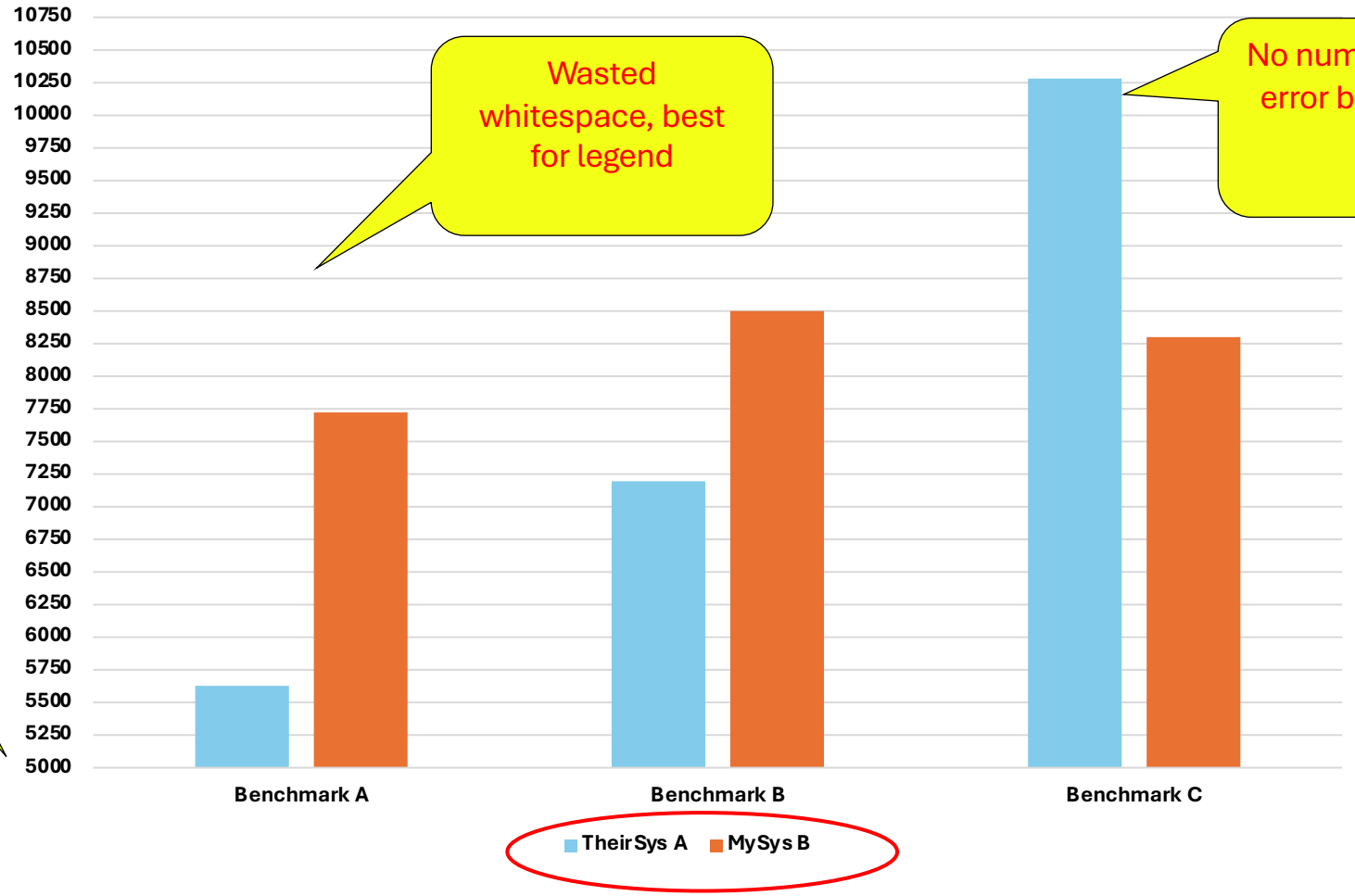
Too many number labels?

Y axis label missing

Y axis not at 0, magnifies differences, misleading?

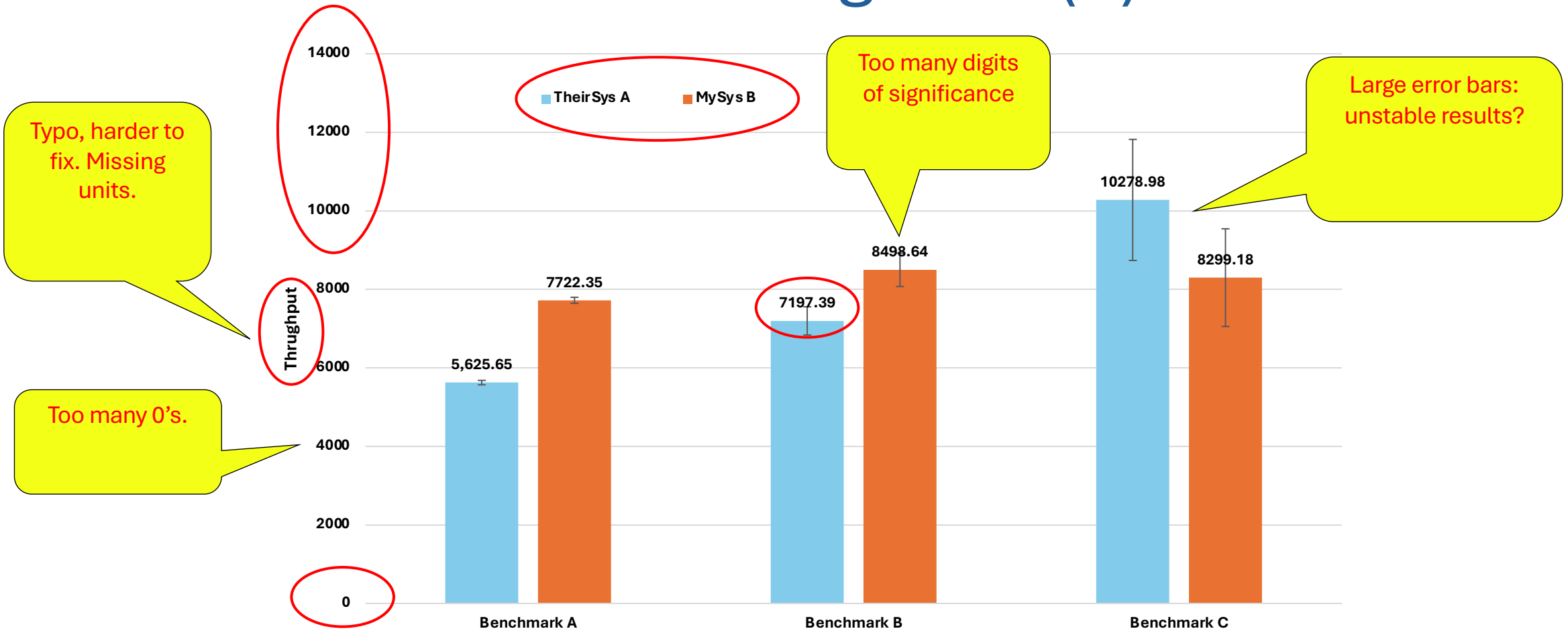
Wasted whitespace, best for legend

No numbers on top, error bars missing



Stony Brook University

Presenting Data (2)

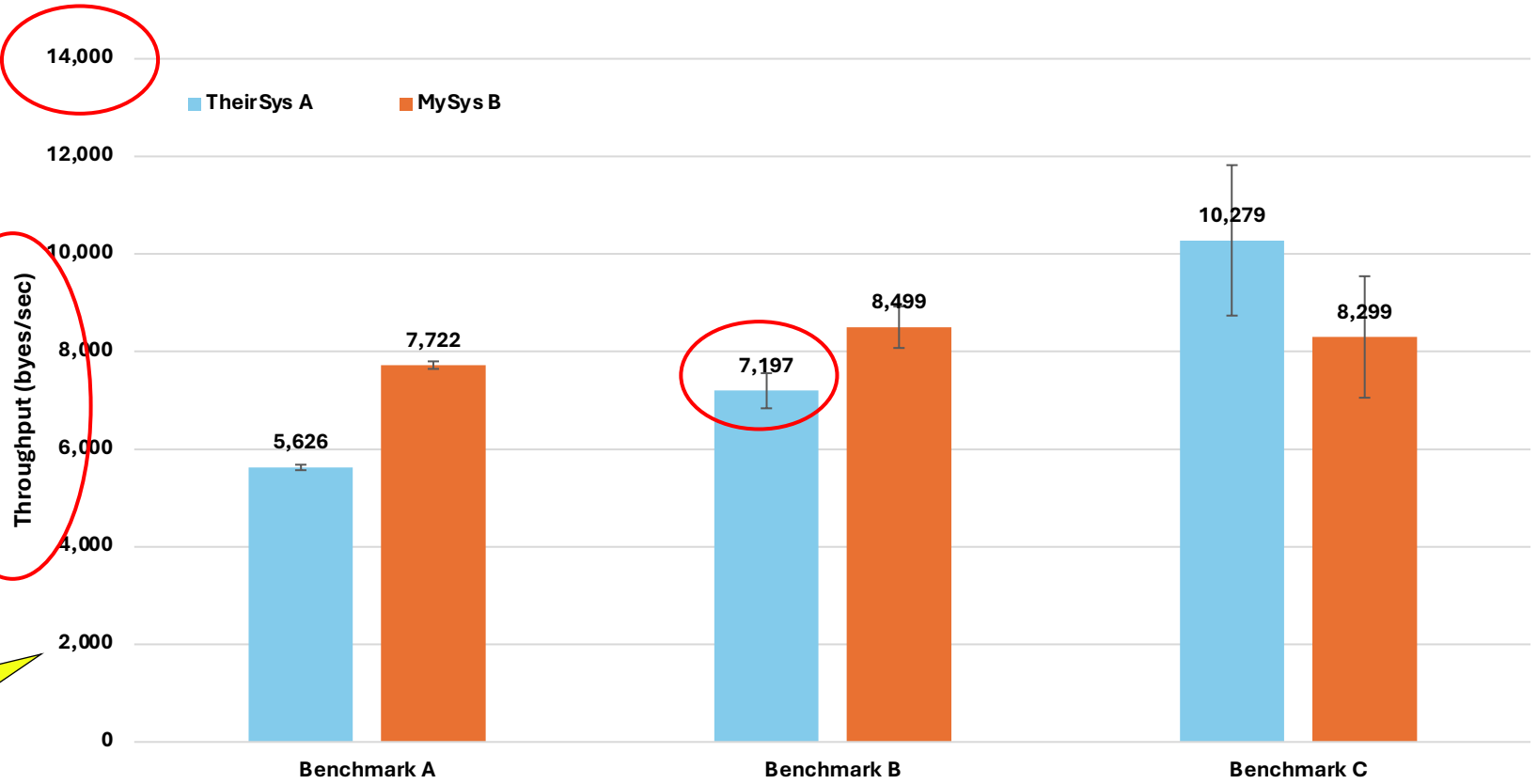


Stony Brook University

Presenting Data (3)

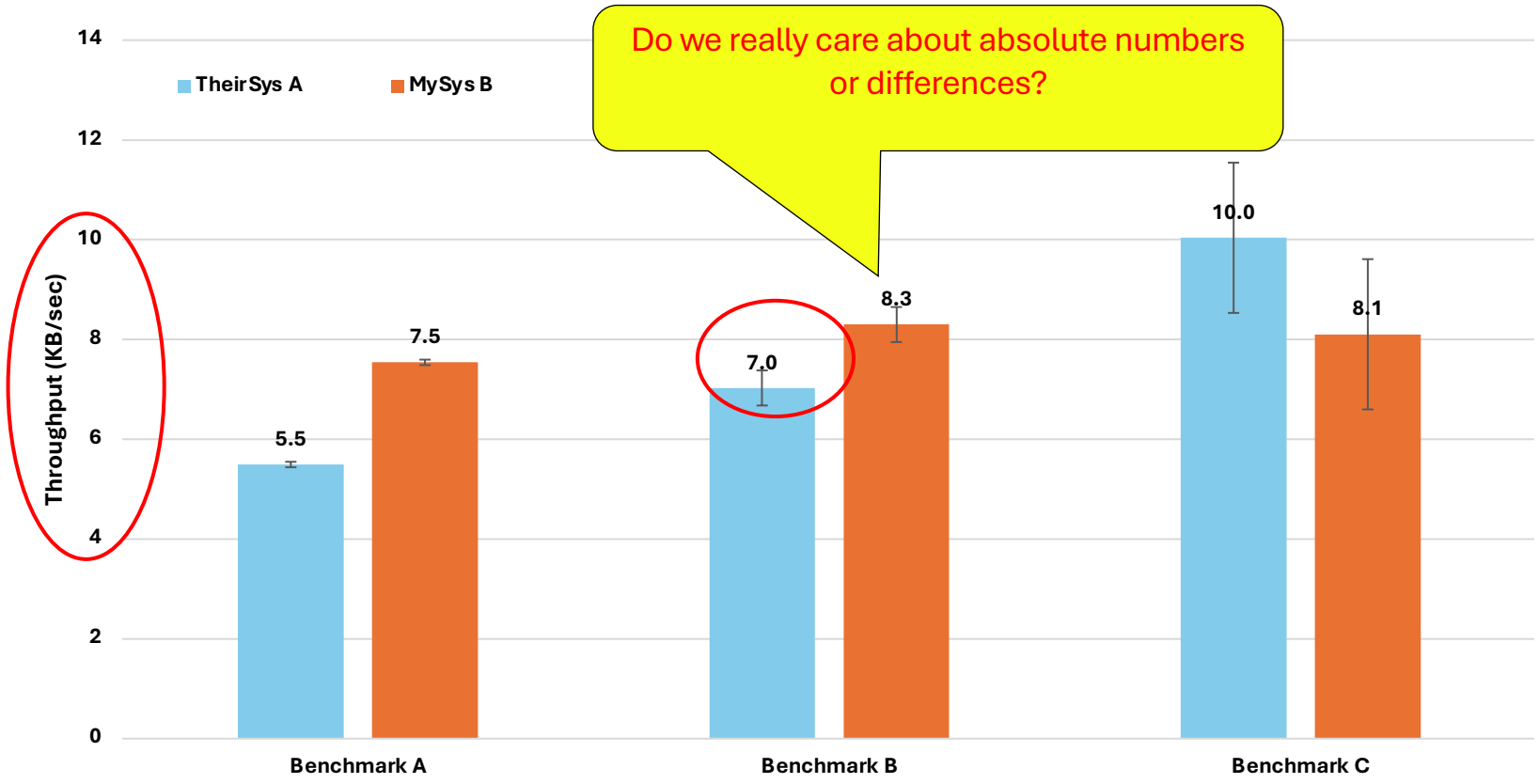
Scale too small (bytes/sec)

Still, too many 0s, hard to read



Stony Brook University

Presenting Data (4)

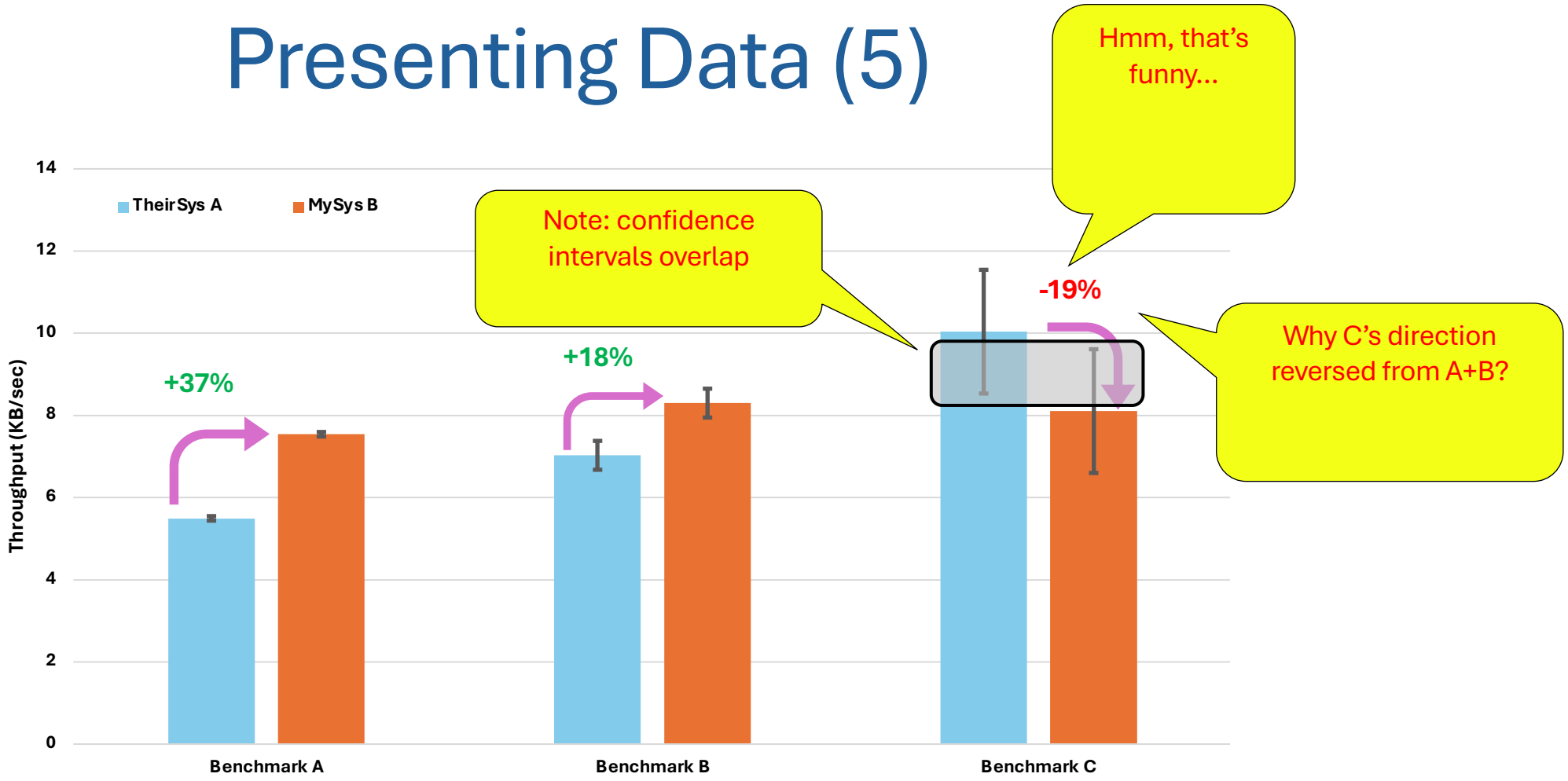


Stony Brook University

The most exciting phrase to hear in science, the one that heralds new discoveries, is not "Eureka!" but "That's funny..."

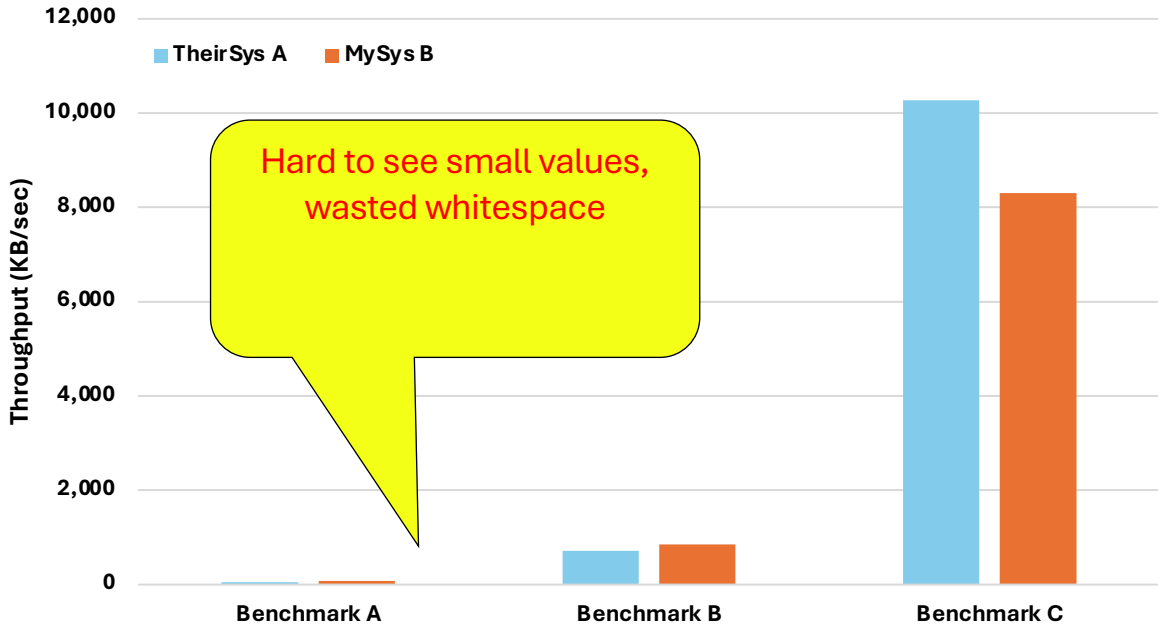
-Isaac Asimov

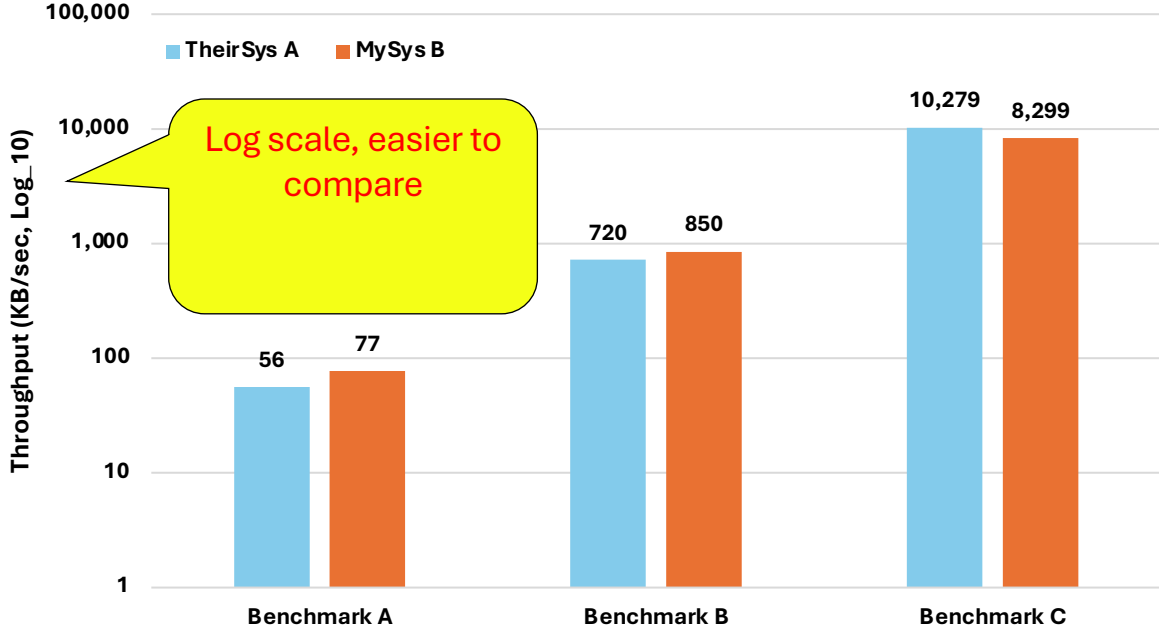
Presenting Data (5)



Stony Brook University

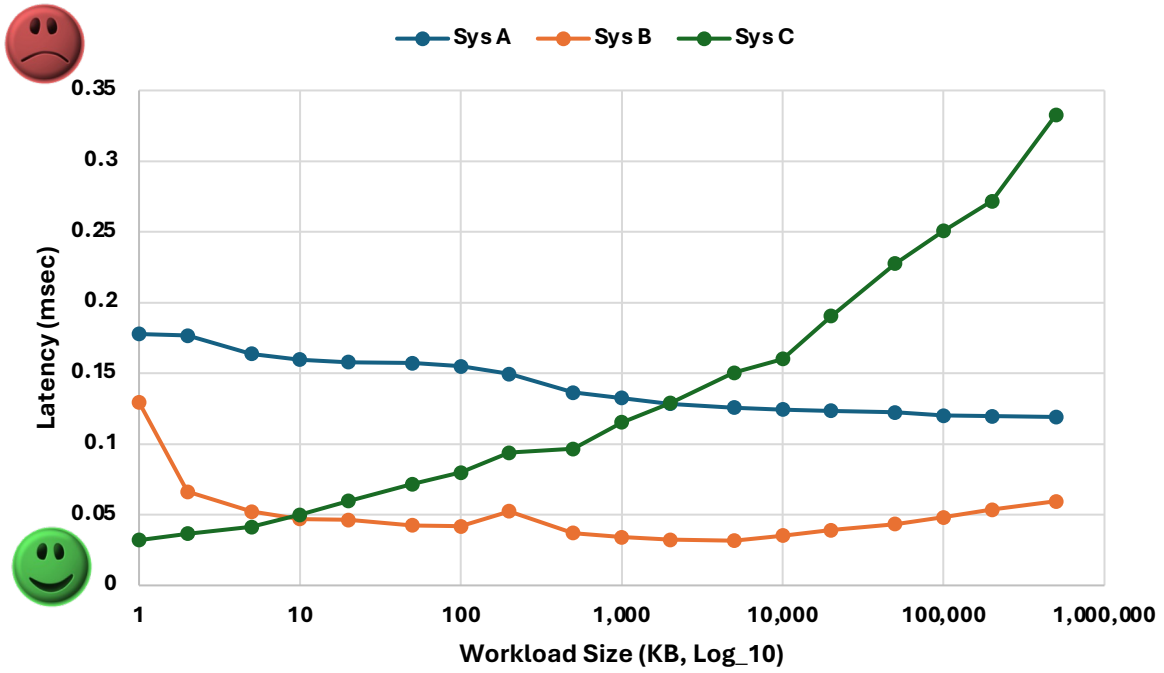
Presentation Scales





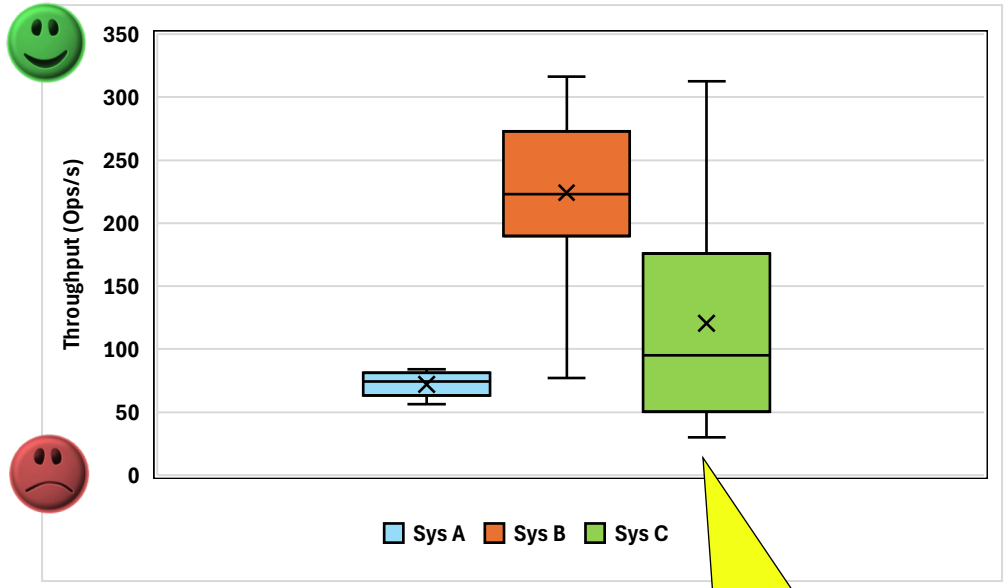
Stony Brook University

Directionality, Trends, Cross-Overs

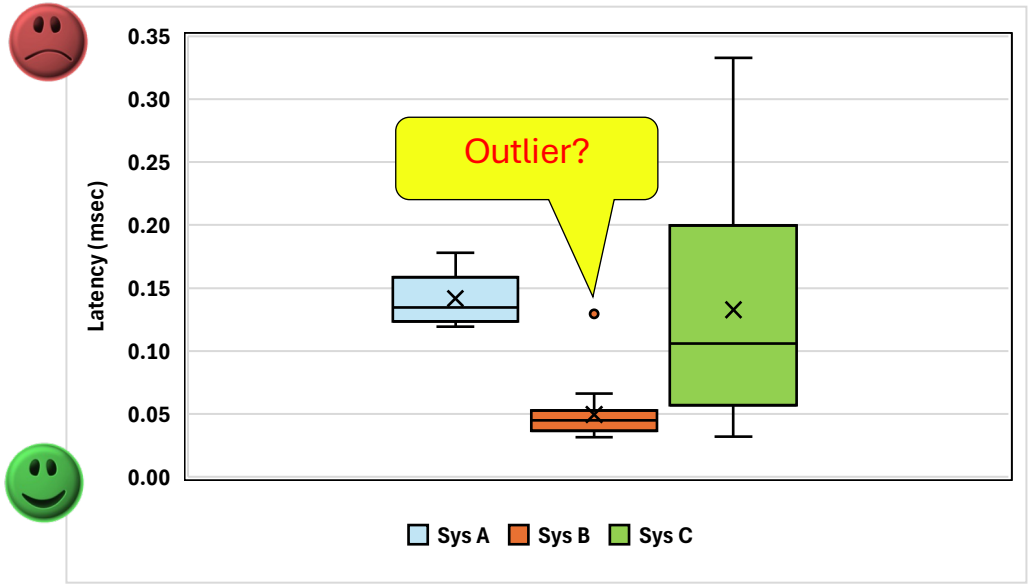


Stony Brook University

Who Needs Bar Plots?

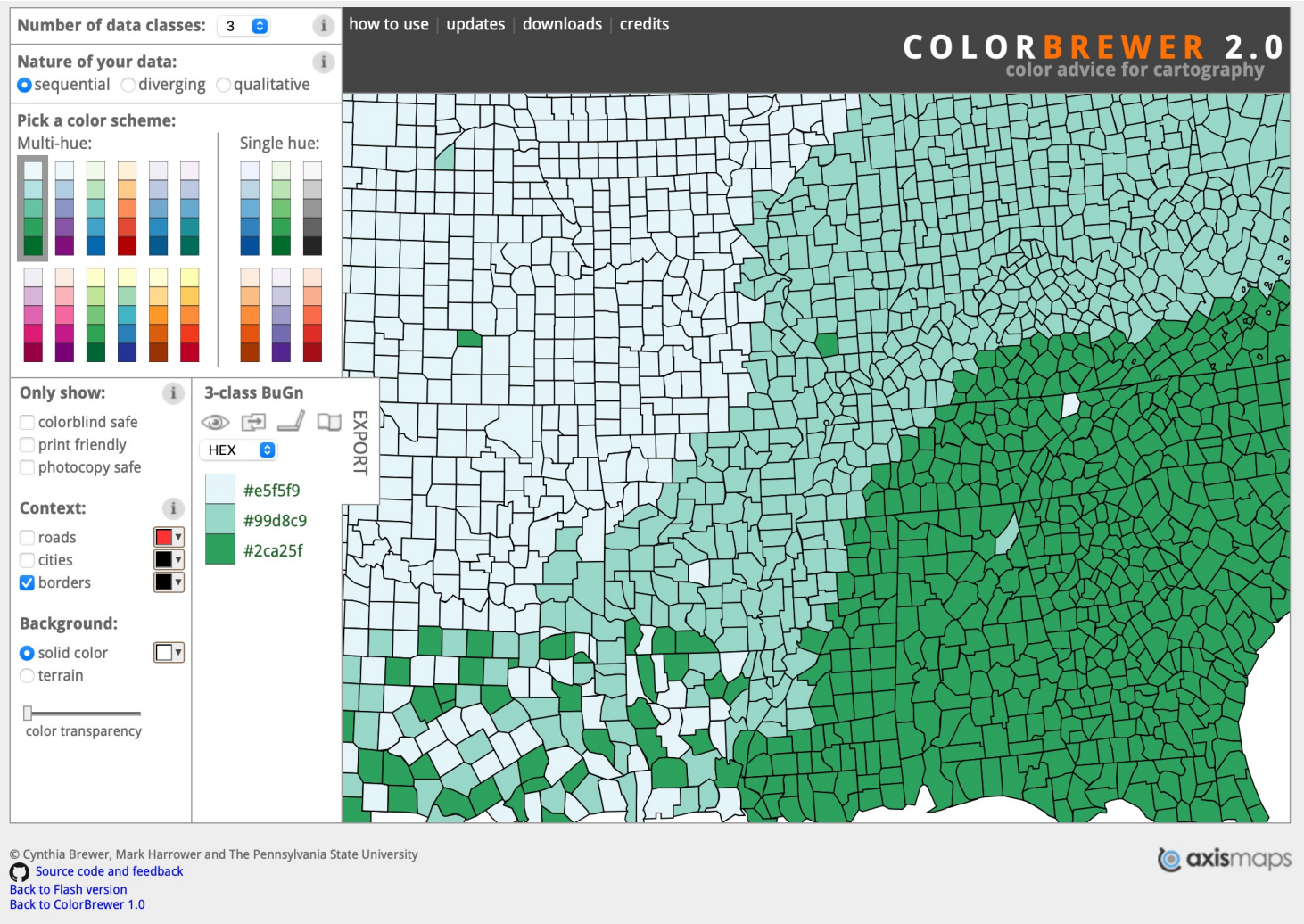


Boxplots: four quartiles, mean (X), median (lines)



Stony Brook University

Choosing Colors



Optimize for

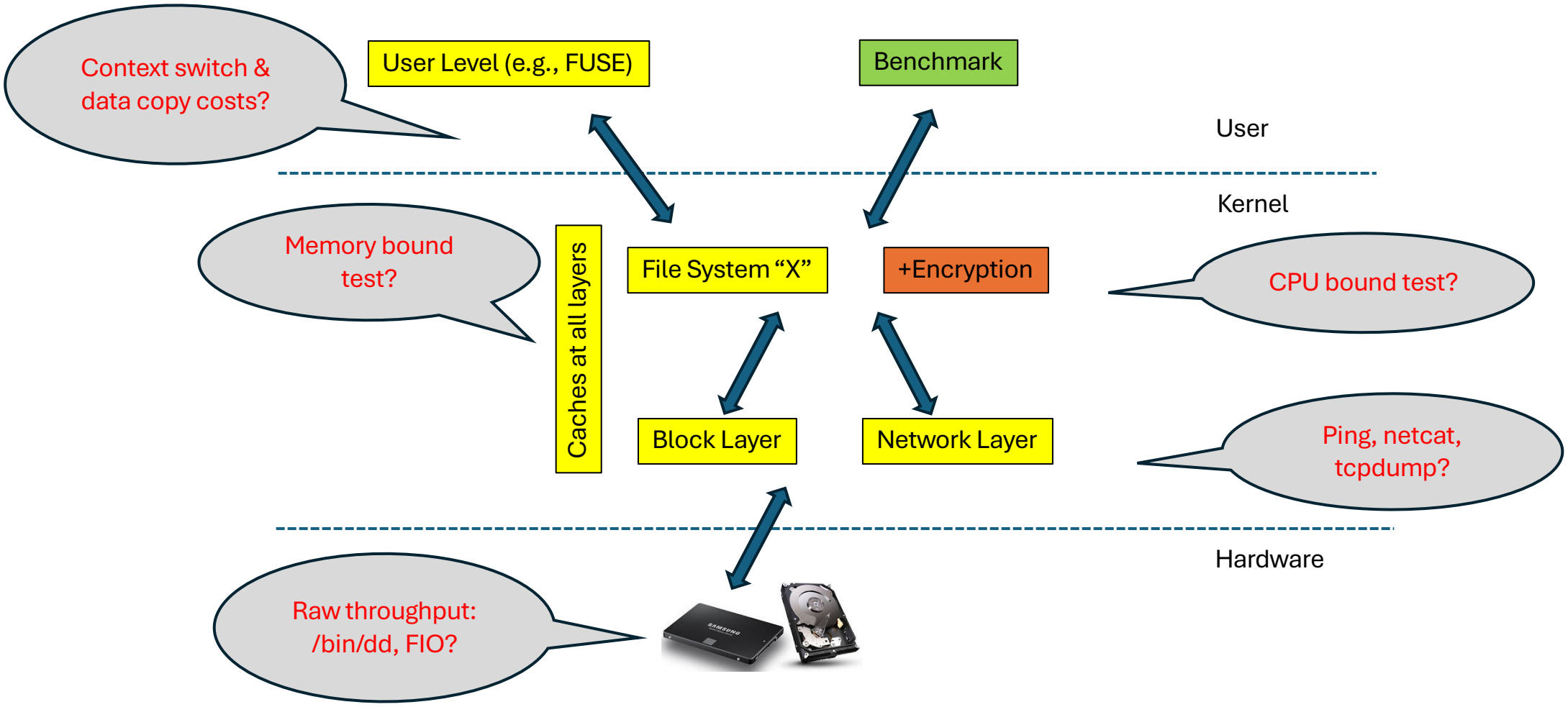
- No. of distinct colors
- Color, B&W, Grayscale
- Print vs. display
- Color blindness
- And more

- https://colorbrewer2.org
- https://projects.susielu.com/viz-palette

Stony Brook University

Analyze This!

Isolating Component Impact



Finer-Grained Impact Isolation

- Suppose I want to find out bottlenecks inside my code?
- How to isolate the impact of specific functions deep inside your code:
  - EBPF, tracepoints, blktrace, tcpdump, etc.
    - Lots of useful info
    - Concerns: overheads, interference, and lost events
  - Do we really need to capture all events?
    - Just calculate averages – not enough information
  - How to capture more information with minimal overhead?

Collecting Histograms Efficiently

Wrap each function with code such as:

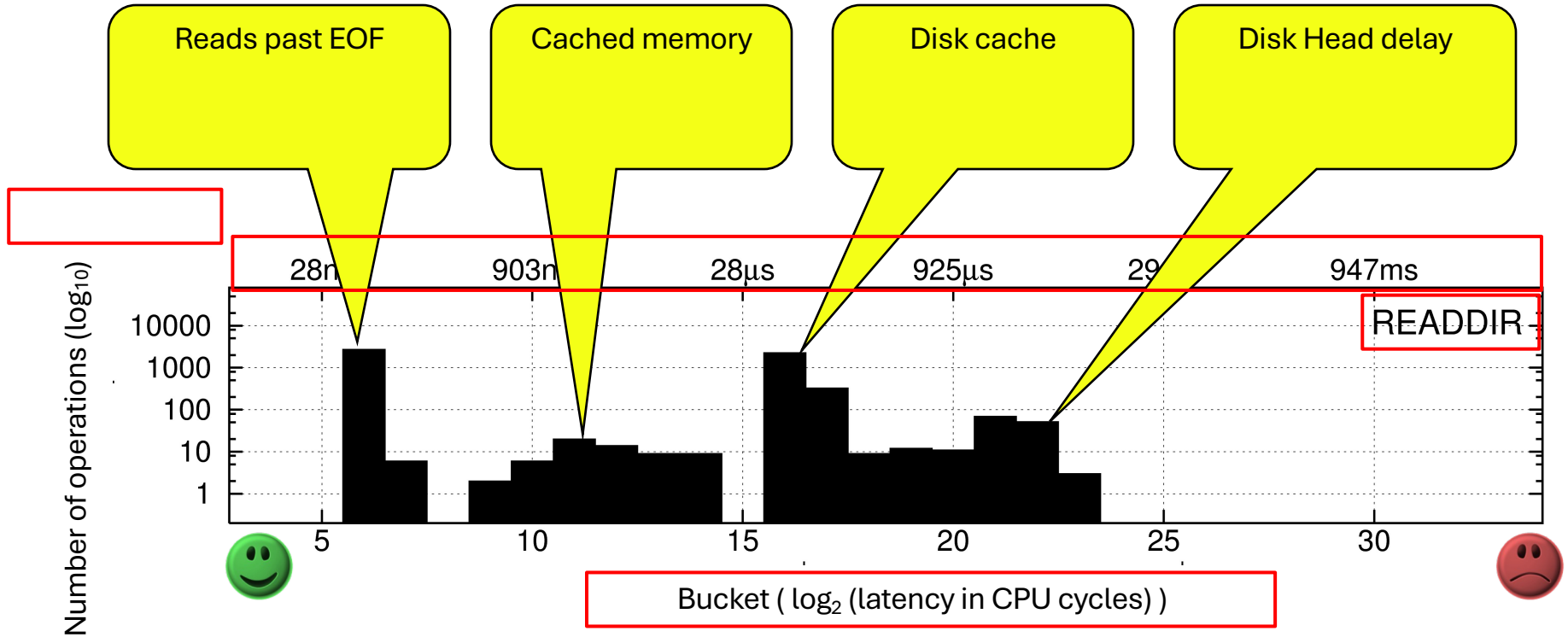
1. Start = TSC() // sample the Time Stamp Counter
2. Run function f()
3. End = TSC()
4. Diff = End – Start
5. Bucket=0; while (Diff > 0) {Diff>>=1; Bucket++;} // calc bucket no.
6. Histogram[Bucket]++; // record counts per bucket
7. // Offline: sample Histogram[] vector periodically or at end of experiment

Creates Log_2 buckets

Negligible memory use, CPU overheads (c. 2006) ~4%, mainly TSC()

[OSProf, OSDI 2006]

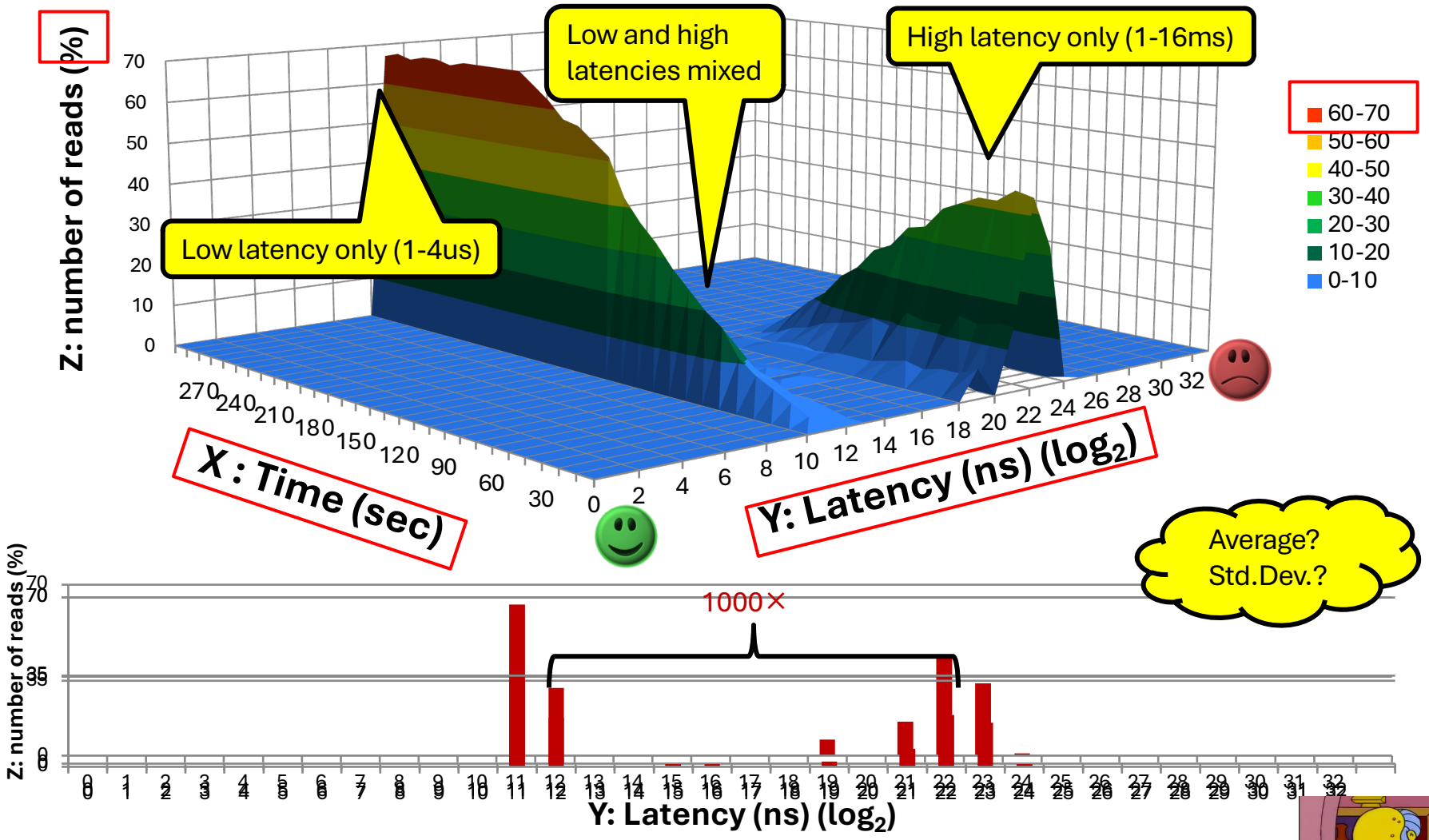
Multi-Modal Behavior



Linux 2.6.11, Ext3 on HDD, **`grep -r`** on source tree

Temporal Modality

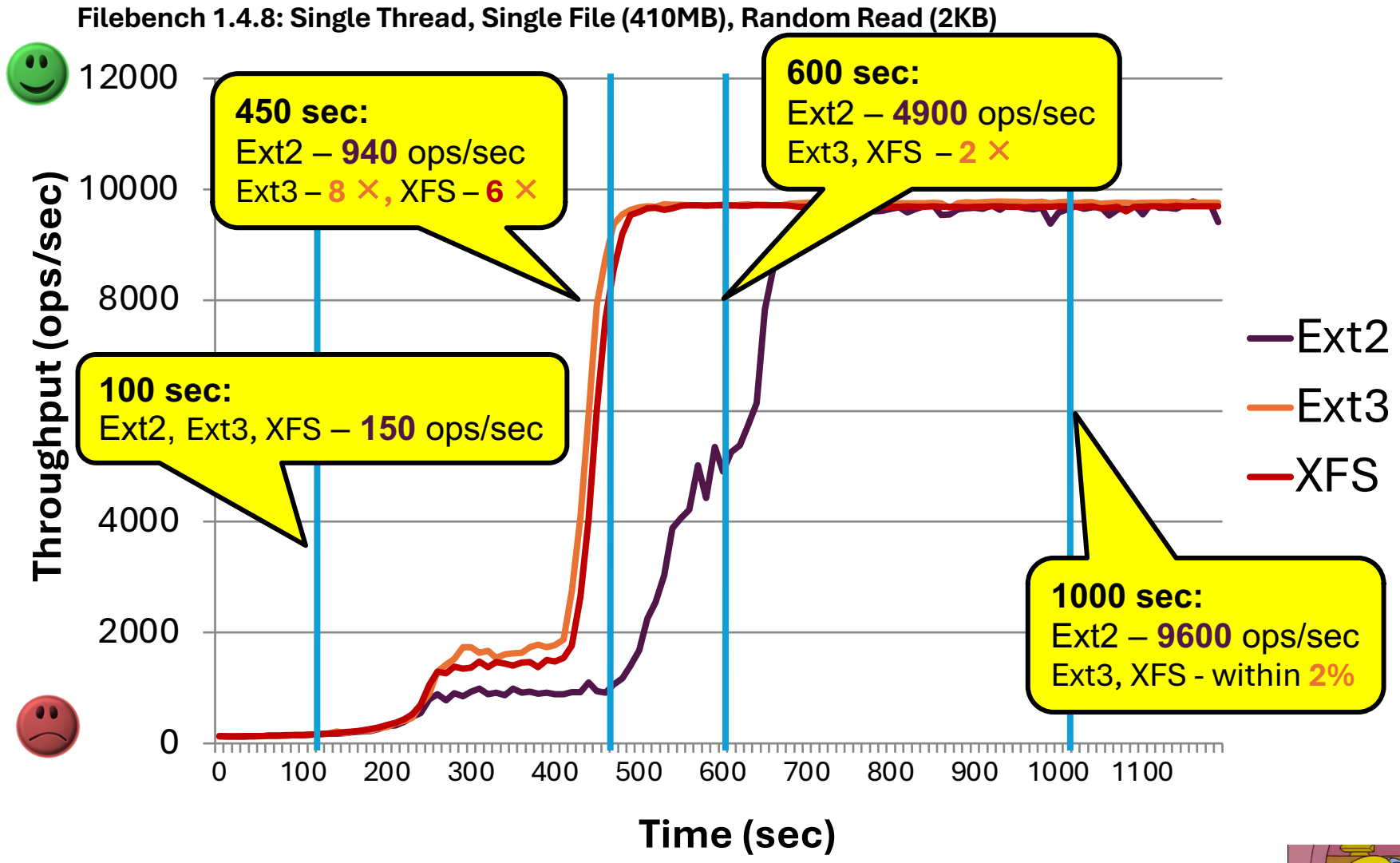
Filebench 1.4.8 (modif.): Single Thread, Single File (256MB), Random Read (2KB), Ext2



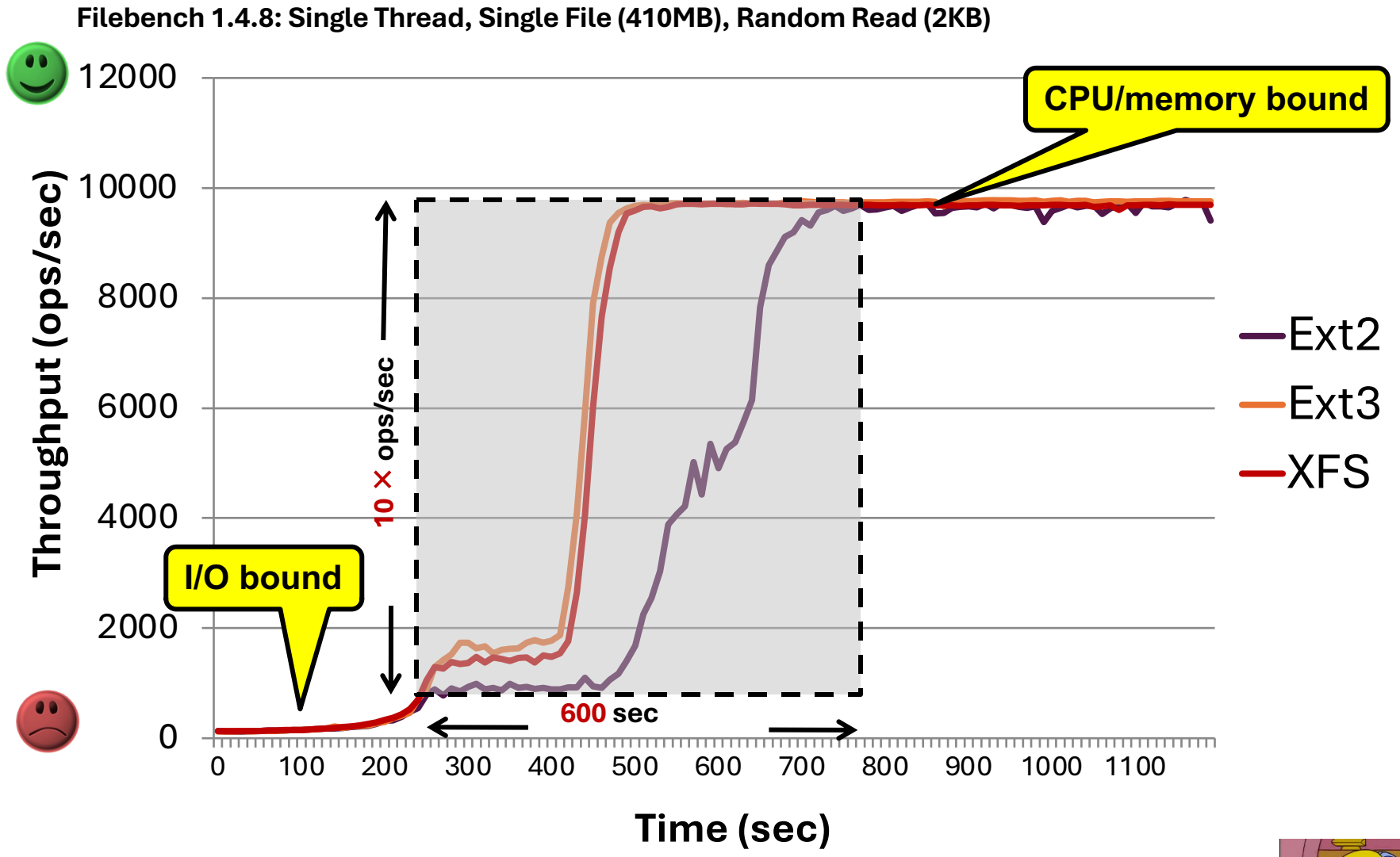
[HotOS 2011]

Stony Brook University

Using Different File Systems



Using Different File Systems



The Silent Treatment

Shhh... look around the room. How many people with their heads down at their laptops, do you think, are now revising their talk slides and even rerunning experiments? ☺

The Long Goodbye

Data is Everywhere!

Storage “research” is *intra*-disciplinary:

- Formal methods, runtime verification
  - Finding bugs and performance anomalies, efficient tracing
- Cryptography and security
  - Long term archival security
- Natural Language Processing
  - Analyzing storage RFCs, finding ambiguities, translating to models/code
- Visual Analytics
  - Analyzing massive multi-dimensional data

Quest for More Data

- Data is critical
- Lack of current, large data sets/traces
- SNIA IOTTA has many
- FSL Dedup data set
  - Collected over six years
  - 5TB compressed
  - Mentioned or cited 250+ times in papers
  - Subsets downloaded 36,000+ times

Use Zenodo to create citable DOI for data released

Zadok's Law (2026)

"Computer systems,

when scaled,

become a storage problem."

Shameless Plug for Journals

- Two of top-10 most downloaded ACM TOS papers were mine:
  - 9-year study of file system benchmarking [2008]
  - FUSE performance and optimization [2019]
- No upper page limit
- Good for expanded conference versions
- Excellent for survey papers
  - E.g., ACM TOS “Past, Present, and Future of Storage Systems”
    - Special issue papers on DNA, Silica/Glass, Holographic, SSD, SMR, NVM, Lustre, Tape

Remember to look for and cite journal versions

Storage Research is Hard, and…

"Kernel hacking is hard!

Really hard.

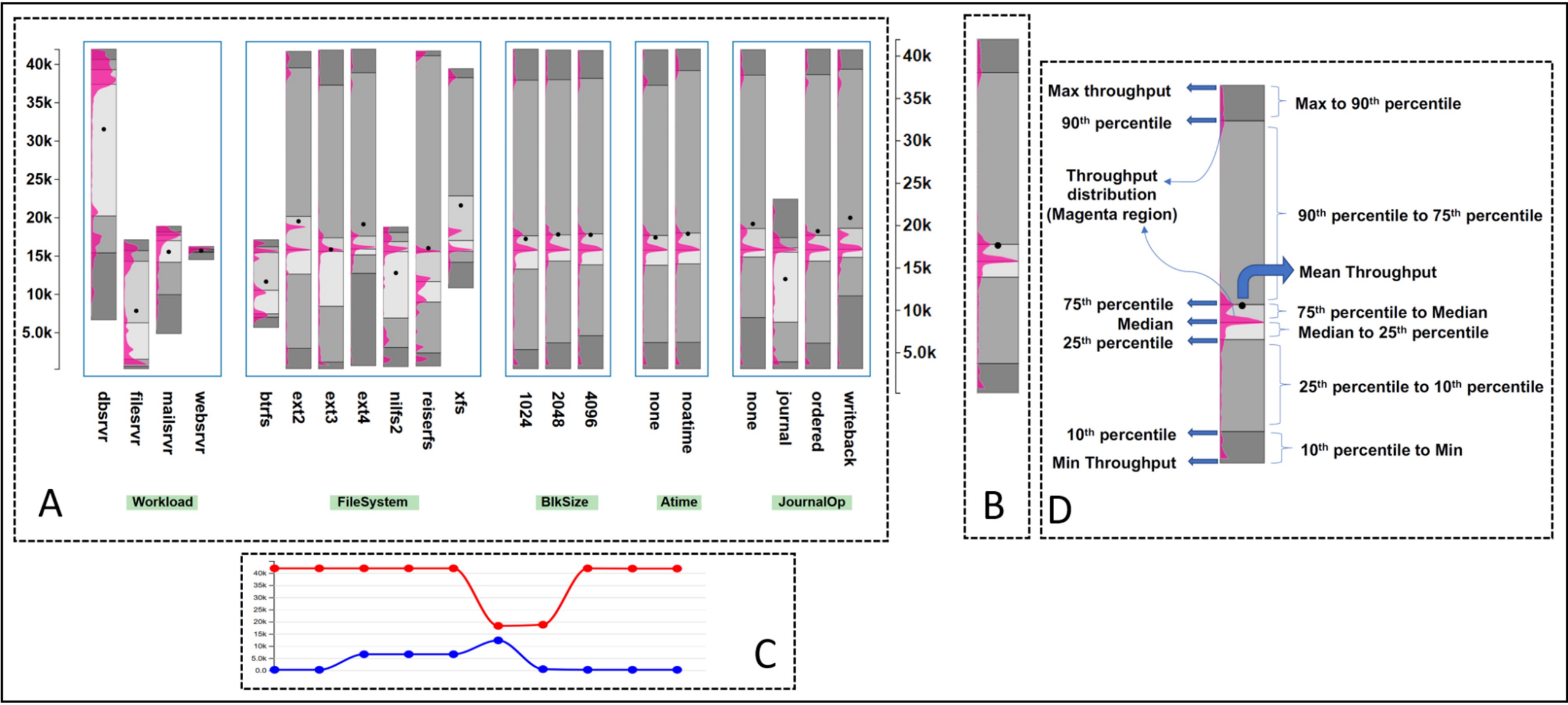
But once it works, man.

It's better than sex.

Everything You Always Wanted to Know about Storage Analysis

(But Were Afraid to Ask ;)

Erez Zadok
Stony Brook University



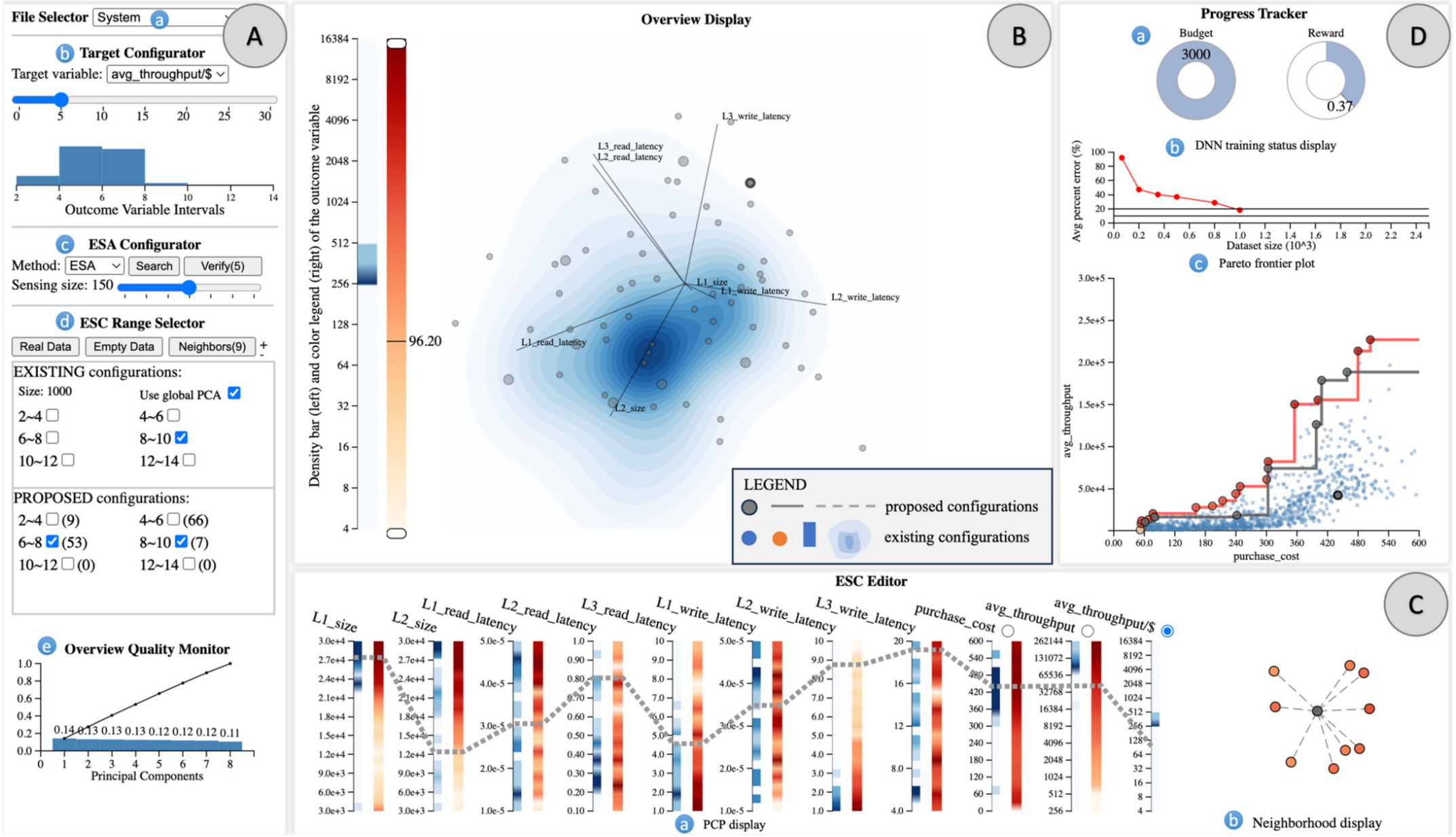
ICE: Interactive Configuration Explorer for High Dimensional Categorical Parameter Spaces (VAST '19)

VAST 2019, HotStorage 2019

Stony Brook University

PC-Expo: A Metrics-Based Interactive Axes Reordering Method for Parallel Coordinate Displays (TVCG '22)

Stony Brook University



Into the Void: Mapping the Unseen Gaps in High Dimensional Data (TVCG '25)

Stony Brook University