

# LASER: Light, Accurate Sharing dEtection and Repair

Liang Luo, Akshitha Sriraman, Brooke Fugate, Shiliang Hu, Chris J Newburn,  
Gilles Pokam, Joseph Devietti



# Multicore is Eating the World

- + Performance
- + Energy efficiency
- Performance bugs

# Cache Contention Bugs



## Contention for a single cash line

- Caused significant performance loss (Linux, MySQL, Boost)
- Architecture-specific
- Hard to find and debug

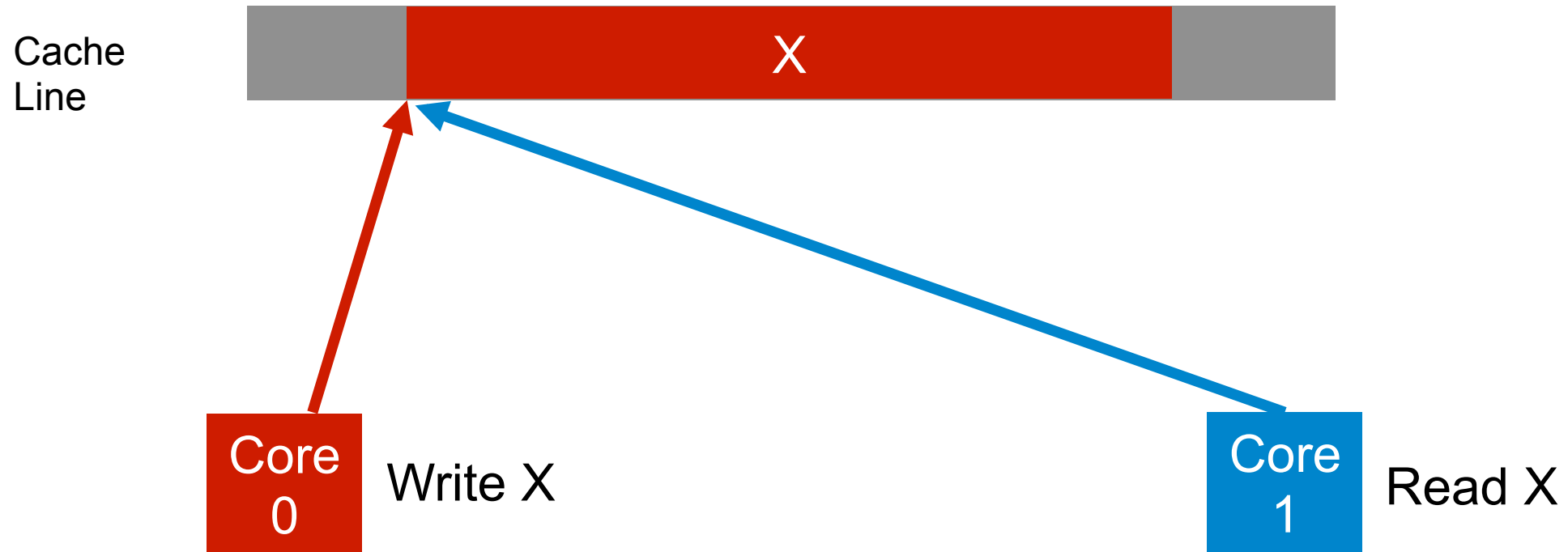
# Background

- Cache coherence keeps private caches in sync
- All protocols share 3 key states: Modified, Shared, Invalid



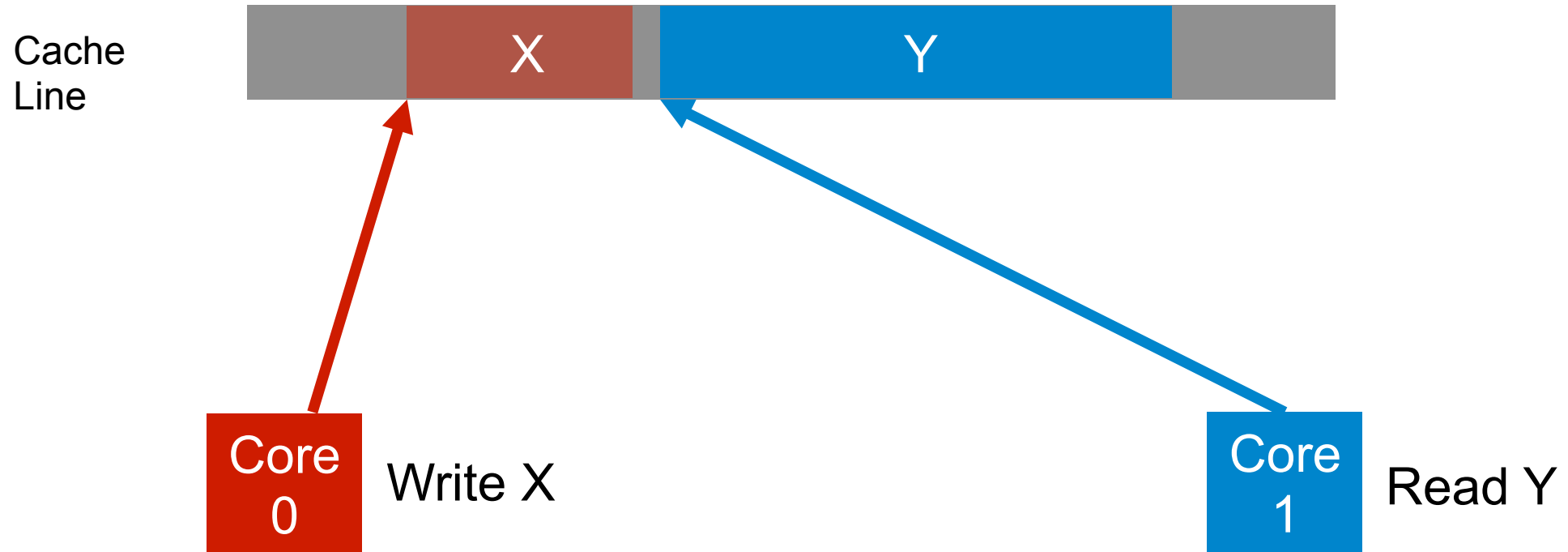
# Two Types of Contention

Same Bytes == True Sharing



# Two Types of Contention

Different Bytes == False Sharing



# Related Work

- **Sheriff**  
[Liu and Berger, OOPSLA 2011]
- **Plastic**  
[Nanavati et al., EuroSys 2013]
- **Cheetah**  
[Liu and Liu, CGO 2016]
- **vTune Amplifier XE**  
[Intel]

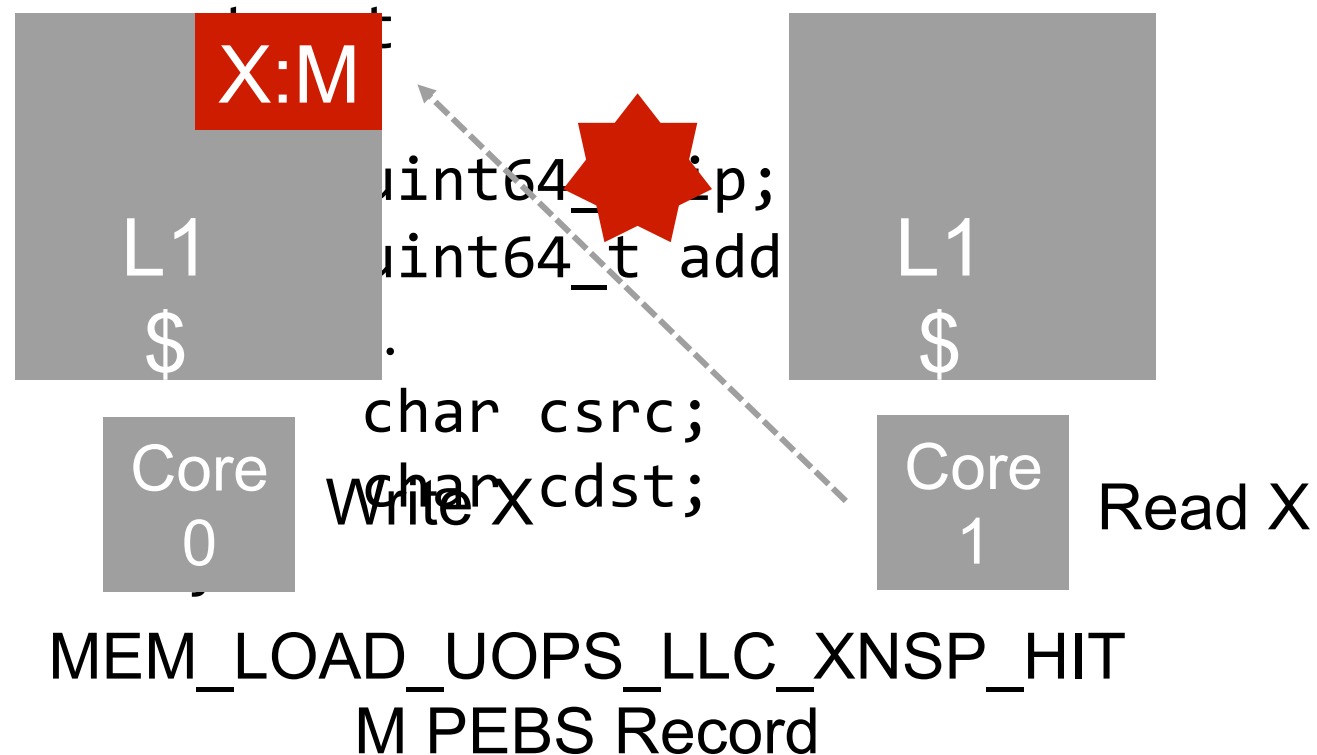
Detection & Repair for  
False Sharing

Offline Analysis  
Predicts speedup from FS

Detection for generic  
events

# HitM Events

- A fundamental part of both types of contention
  - A cache hit in a remote core's cache in M state.





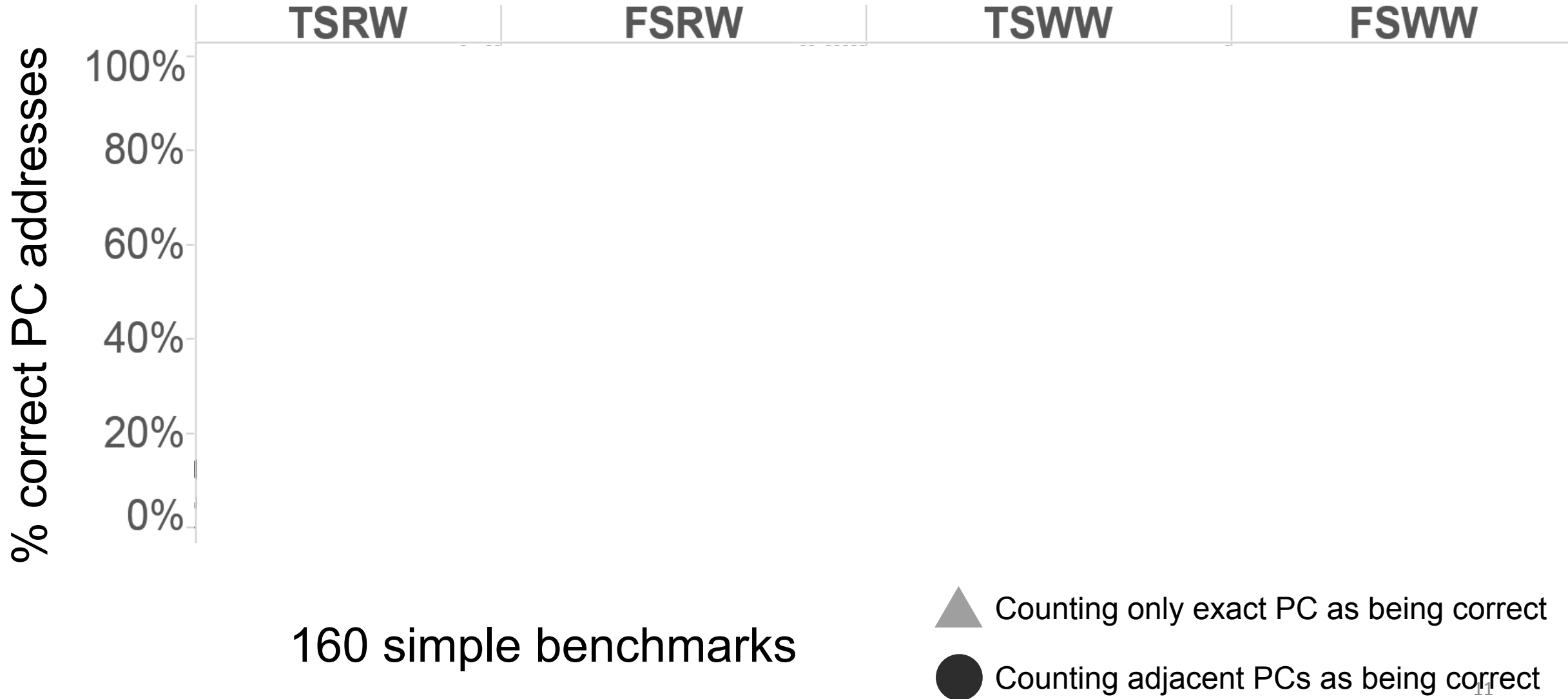
# HITM Record Accuracy

- 160 simple programs with read/write and write/write true/false/no sharing
- Intel Core i7-4770K 3.4GHz Haswell 4-core processor

# HITM Record Accuracy(1/2)



# HITM Record Accuracy(2/2)



# LASER

- **L**ight: leverage Haswell h/w, no s/w or OS changes
- **A**ccurate: low false positives and false negatives
- **S**haring: detects both true and false sharing
- **dE**tection: works as a profiling tool
- **R**epair: automatically repairs false sharing at runtime

# LASER System Overview

User Level App



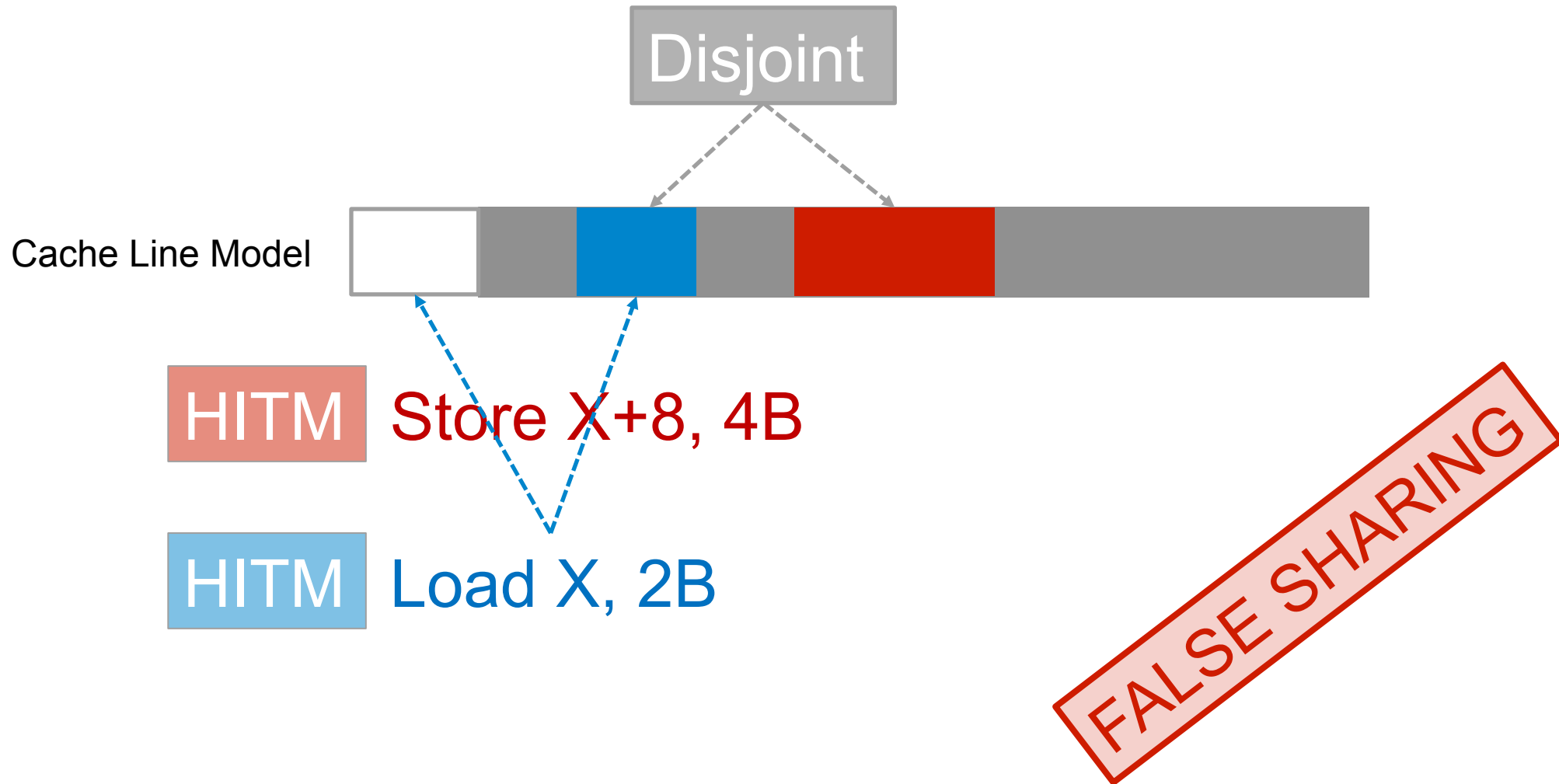
Operating System



Hardware



# Detection Algorithm



# Detection Algorithm



Aggregate  
Byte Source  
Code Line



Foo.c:23

# Evaluating LASER Detection

- Intel Core i7-4770K 3.4GHz Haswell 4-core processor
- 33 workloads from Phoenix 1.0, Parsec 3.0 and Splash2X



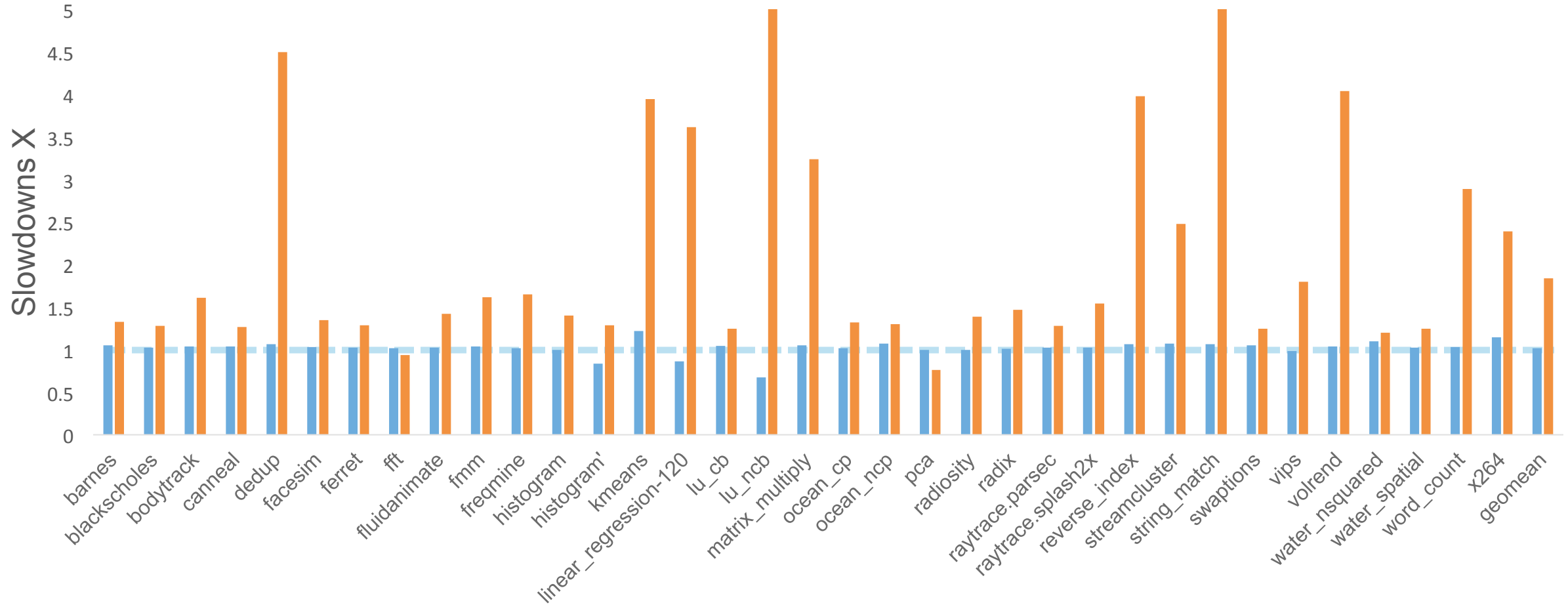
# LASER Detection Accuracy

Created a database of manually-validated cache contention bugs

- 9 bugs total across 33 workloads
- 4 new bugs discovered by **LASER**

	<b>False Negative</b>	<b>False Positive</b>	<b>Runnable Benchmarks</b>
<b>LASER</b>	0	24	33/33
VTUNE	1	64	33/33
SHERIFF	3	4	12/33

# Profiling Performance Comparison

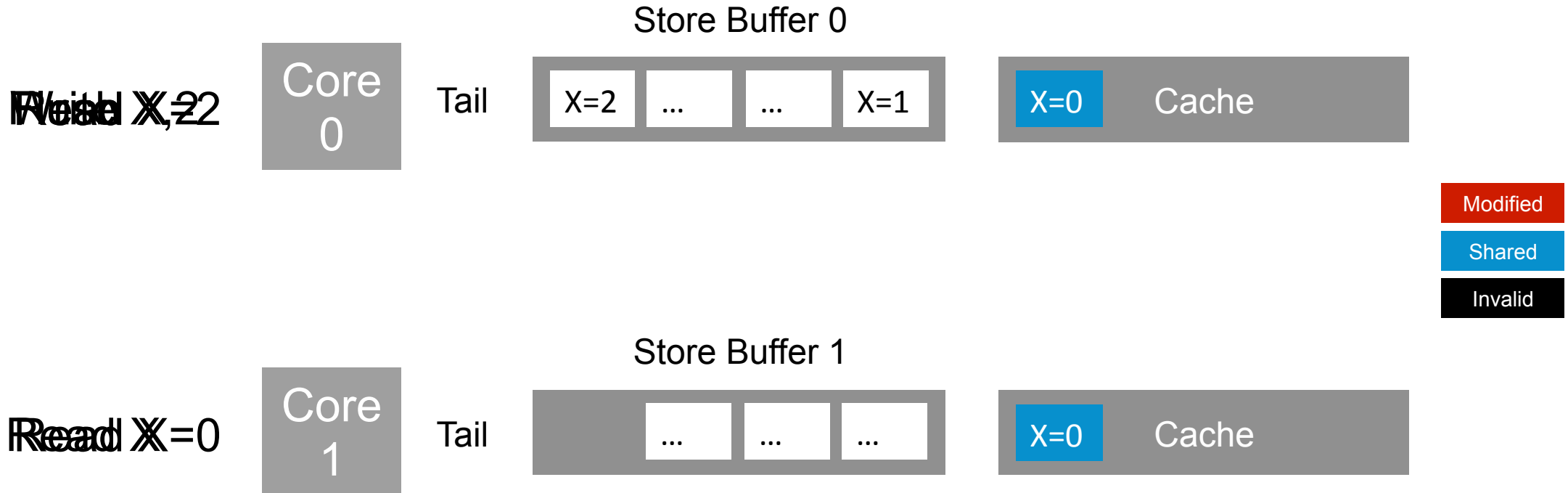


**LASER (1.01x) vs VTune Amplifier XE 2015 (1.8x)**

# Repairing FS with SSB

- Needs Online False Sharing elimination!
  - Legacy programs – no access to source code.
  - Programs that need to be always running.
- Challenge is needing to rewrite the program without breaking it as it is running.
- Solve the problem of online FS repair with a Software Store Buffer (SSB). **LASER** repair tool is launched to attempt fix of FS.
  - Implemented with Intel Pin

# Repairing FS with SSB



Flush at synchronization points for TSO compliance and basic block end for performance.

# Challenges with Optimizing SSB

- The conventional hardware store buffer is not good enough for speedup.
- Needs optimizations for better performance
  - May cause subtle memory consistency issues.
  - For good performance, requires coalescing store buffer. But coalescing violates TSO. E.g. Sheriff does not provide TSO compliance.

# LASER's TSO-Compliant, Coalescing SSB

- Instrument regions from Laser's input by walking through the CFG.
- Coalescing, TSO compliant SSB made possible with Intel TSX.

Read X:

```
Result = SSB[X]
If (Result == null)
    Result = *X;
Return Result;
```

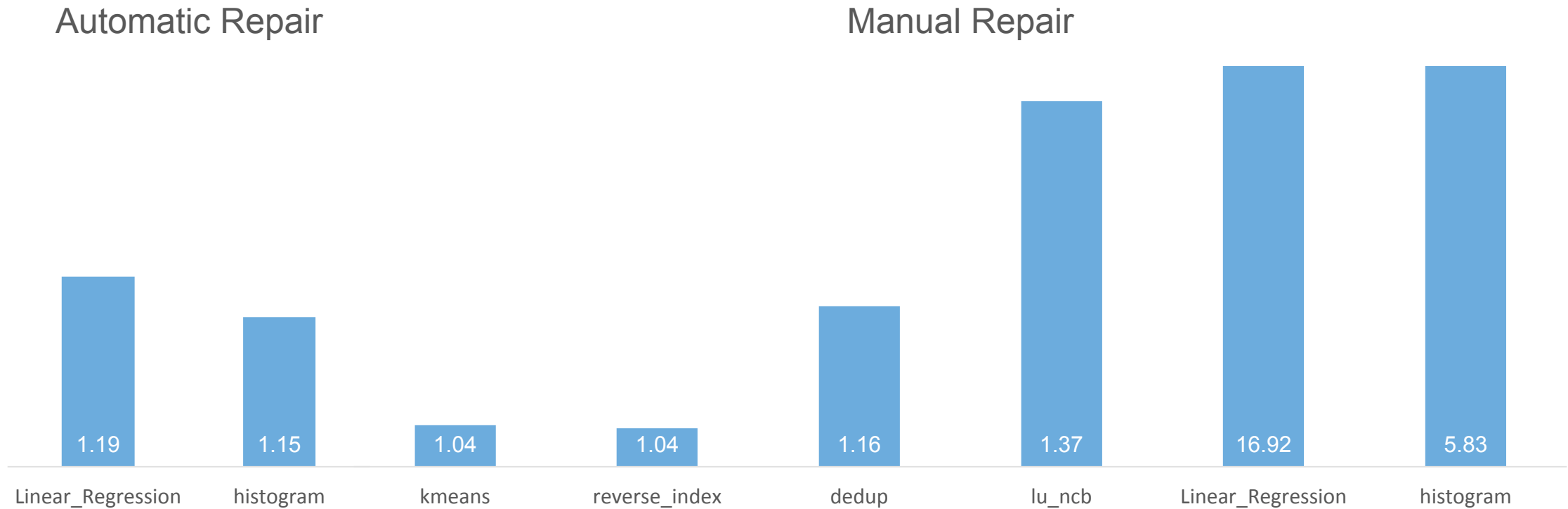
Write X, Val:

```
If (SSB[X] == null &&
    SSB.full())
    Flush();
SSB[X] = Val;
```

Flush:

```
TSX_Begin_Transaction
Foreach pair in SSB
    *pair.memory = pair.value;
TSX_End_Transaction
Redo_TSX_Transaction_If_Fa
ils
```

# LASER FS Repair Performance



Automatic speedups of up to 19%  
**LASER** profiling informs manual fixes of up to 17x

# Conclusions

- Cache contention bugs undermine the promise of multicore
- **LASER** uses Intel's Haswell platform for fast, precise contention detection and automatic false sharing repair
- Many opportunities to leverage Haswell's sharing detection capabilities

## Researchers Who Read This Paper Also Read

REMIX: Online Detection and Repair of Cache Contention for the JVM

[PLDI 2016]

Add to read list