
ORACLE®

ORACLE®

The Influence of Malloc Placement on TSX Hardware Transactional Memory

Dave Dice, Tim Harris, Alex Kogan, Yossi Lev, Victor Luchangco
» Oracle Labs
» **Transact 2016**

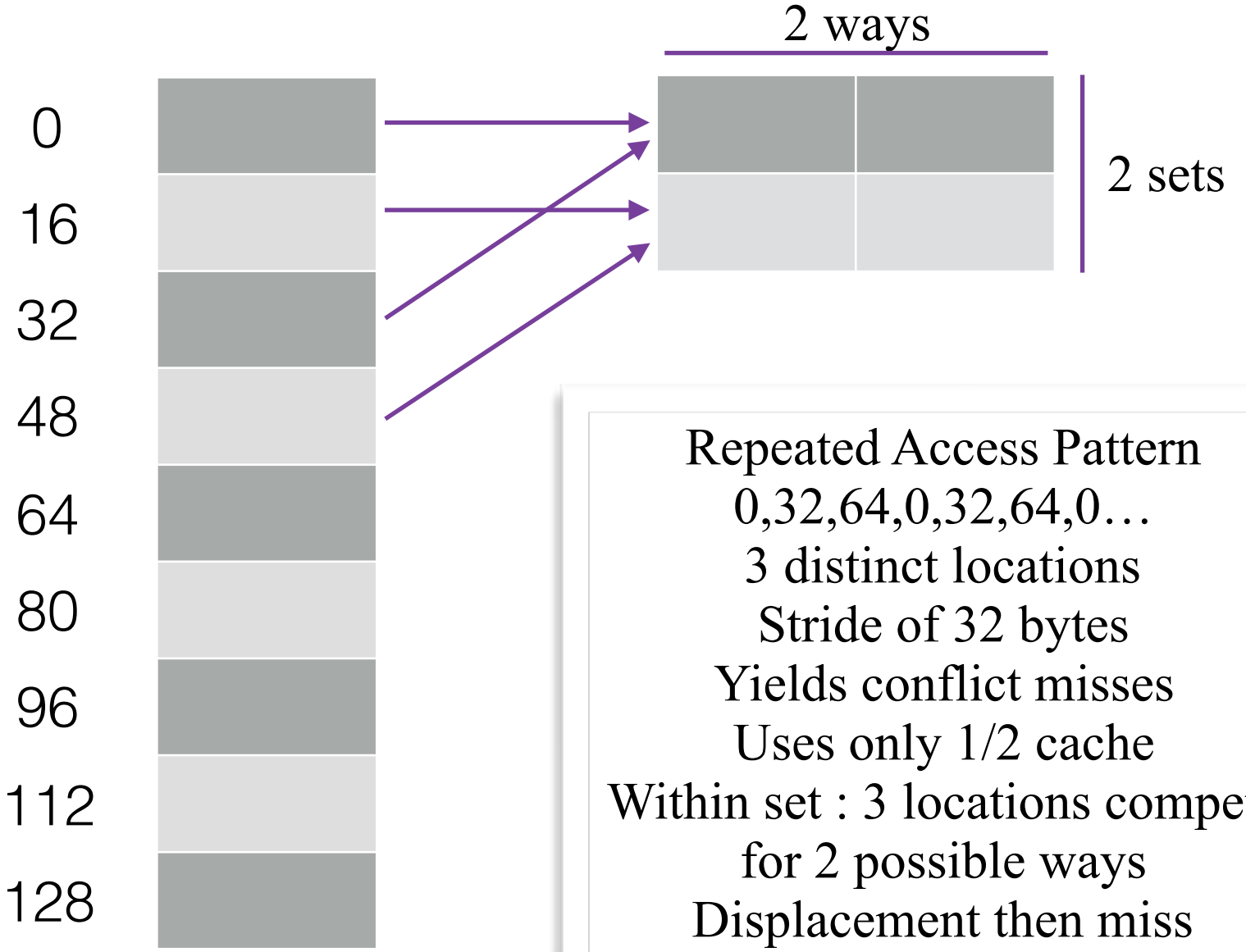
Summary

- Virtual address malloc() placement decisions …
- Thread-local conflict misses in L1 cache … (Associativity misses)
- Persistent TSX-RTM aborts … misses can become aborts
- TLE fall-back to central lock — serialization
  TLE = Transactional Lock Elision

Local malloc() placement decisions turn into global scaling concern

Cache Geometry

- Conflict misses
  - associativity
  - collisions in hash function : address to index
  - underutilize cache : hot vs cold indices
  - parking lot analogy
  - a given location is restricted to just one set
- Trivial example
  - 4 lines total capacity; 16 bytes per line
  - 2-way set associative
  - Address = [Tag ; Index (2 bits); Offset (4 bits)]



Repeated Access Pattern
0,32,64,0,32,64,0…
3 distinct locations
Stride of 32 bytes
Yields conflict misses
Uses only 1/2 cache
Within set : 3 locations compete for 2 possible ways
Displacement then miss

Haswell i7-4770 L1 Geometry

- 64-byte lines
- 32KB capacity = 512 lines
- 8-way set associative
- 64 possible indices
- “Cache Page Size” is 4KB
  Virtual addresses mod 4K map to same line
- Address = [Tag; index:6 ; offset:6]
- Classic page-based coloring not possible
- Page MMU translation doesn’t overlap index

Backstory

- TL2 STM
  - Lock Records collided with data in L1
    Data maps to same L1 index as lock record
  - Fixes
    - Hash function
    - Color offset of lock record array
- Index-aware malloc() allocator
- Application to HTM

malloc() allocator - desiderata

- Scalable
- Minimize sharing :
  - metadata and data
  - false sharing and true sharing
- NUMA-aware
- Fragmentation; overheads; wastage
- Distribute blocks — beware address regularity
  - over cache indices : index-aware
  - coherence planes
  - Cache banks and DRAM channels

malloc() allocator - desiderata

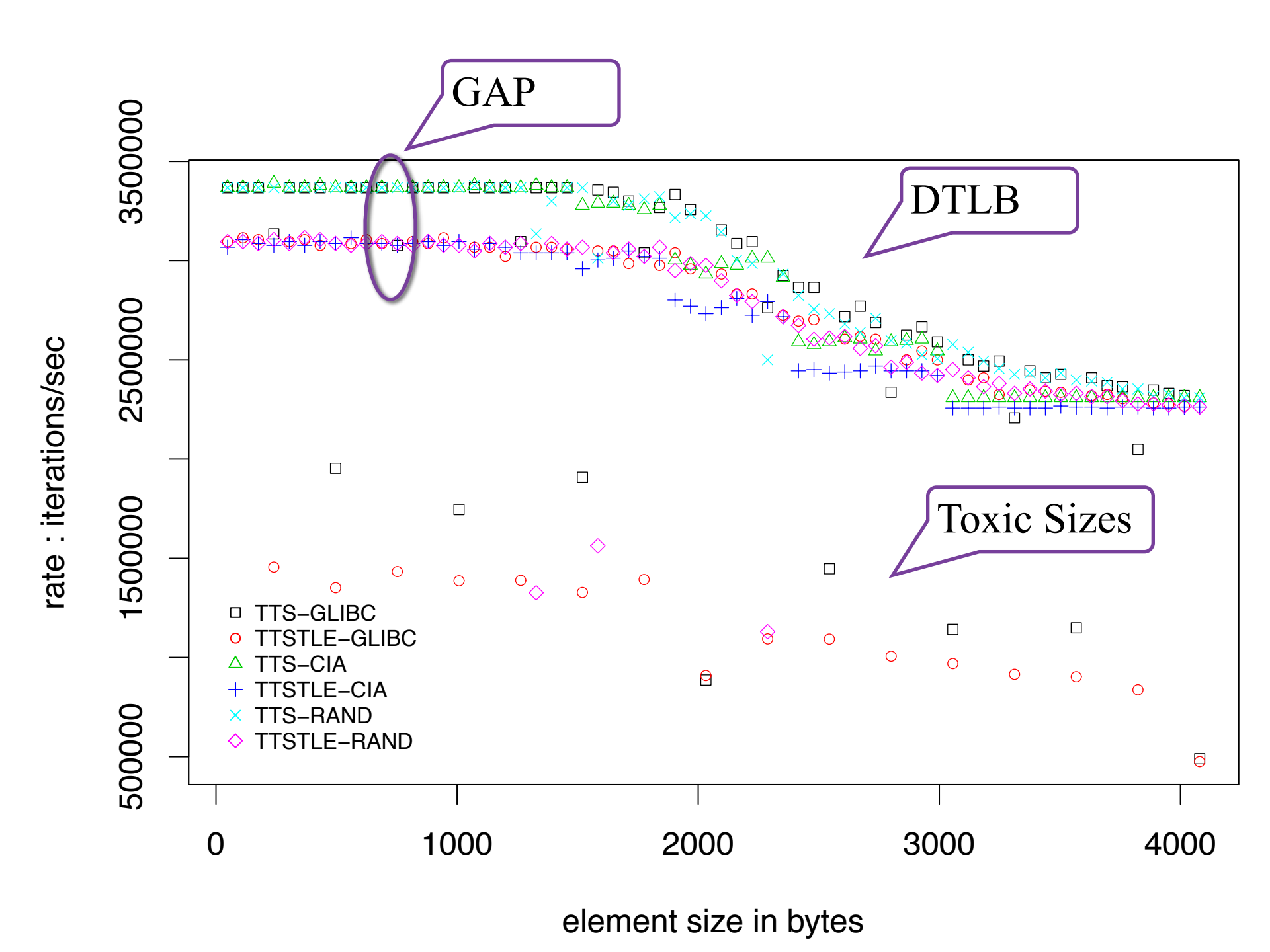
- Footprint and density :
  - TLB span and large pages
  - caches lines
  - number of pages
- minimize the # of lines underlying a given block
  - “working set” of a block
  - prefer aligned to cache boundary
- Memory retention and OS system call traffic
- KEY : Attributes of scalable and HTM-friendly allocator align well. No trade-off

Experiment

- Intel i7-4770 4x2 running Ubuntu
- Single-threaded microbenchmark
- malloc() 128 “nodes”, vary size on X-axis
- Form into circularly linked list
- Acquire lock; traverse 128 elements; release
  w = w->Next ; w->Value = 0
  Access only 2 fields; remainder untouched
- Report traversal rate on Y-axis

Experiment

- Locks implement via LD_PRELOAD interposition
- 2 locks :
  - TTS
  - TTS-TLE (simplistic TLE)
    TLE Retry usually useless for conflict misses
- 3 allocators :
  - default glibc
  - default + randomized size
  - “CIA” = cache index aware allocator


GAP
DTLB
Toxic Sizes
rate : iterations/sec
3500000
2500000
1500000
500000
0
1000
2000
3000
4000
element size in bytes
TTS−GLIBC
TTSTLE−GLIBC
TTS−CIA
TTSTLE−CIA
TTS−RAND
TTSTLE−RAND

The HTM Gap

- TLE HTM vs simple lock
- HTM “constant” overhead
  - XBEGIN; XEND
  - speculation barrier
  - fence semantics
- Transactional store to line already modified
  - Write-set tracked in L1
  - Line in “M” modified state in L1
  - start txn
  - store to line
  - must transfer contents from L1 to L2!
  - conjecture!

Lessons

- Some sizes are pathological : stride
  Depends on the allocator
- Address regularity may be harmful
- Histogram addresses if unsure
- Intentional or allocator-induced
- Beware excessive alignment : false sharing avoidance
- Binary buddy allocators are toxic
- Randomization or index-aware allocators
  Consider Supermalloc
- Vaccinate against rare but painful condition

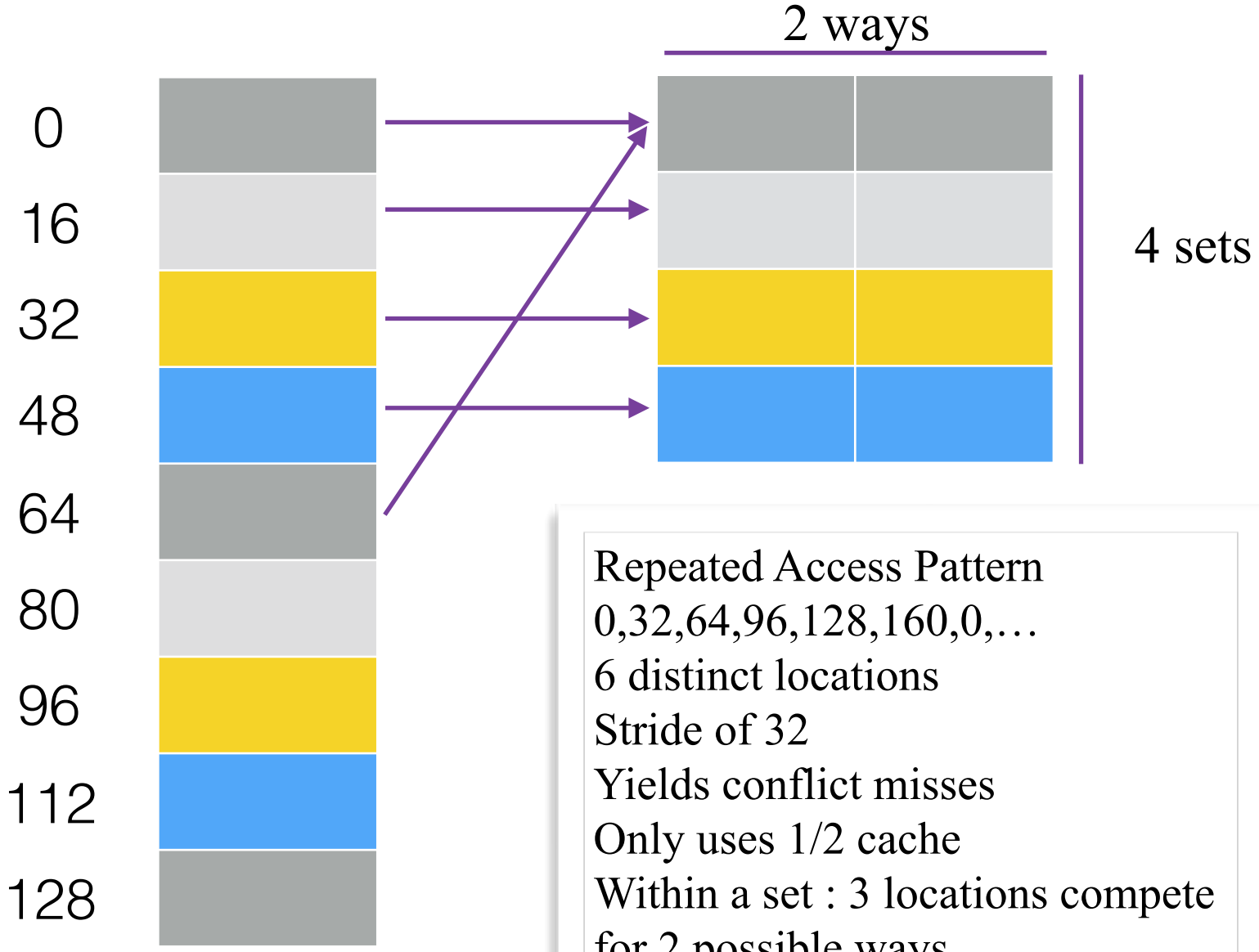
Lessons

- Delay writes until end of txn
  - reads and writes don't commute
  - read-set (L1-L2-L3) vs write-set tracking (L1)
  - ST ; LD 32K vs LD 32K; ST
- Placement …
  Conflict misses …
  Transactional aborts …
  TLE Serialization — fall back to lock

Thank you!

- https://blogs.oracle.com/dave/


A CANAL FOR THE NEW MILLENNIUM
EVER REFINE
PANAMA
TRINIDAD II
OHIO
180
230
285
330
380



Repeated Access Pattern
0,32,64,96,128,160,0,…
6 distinct locations
Stride of 32
Yields conflict misses
Only uses 1/2 cache
Within a set : 3 locations compete for 2 possible ways
Displacement then miss

Backup

- parking lot analogy
- Birthday paradox
- squash redundant writes