Parallelization and Characterization of SIFT on Multi-Core Systems

Hao Feng, Eric Li, Yurong Chen, Yimin Zhang

Presenter: Victor Lee

Application Research Lab, Intel China Research Center

Contributions in This Paper

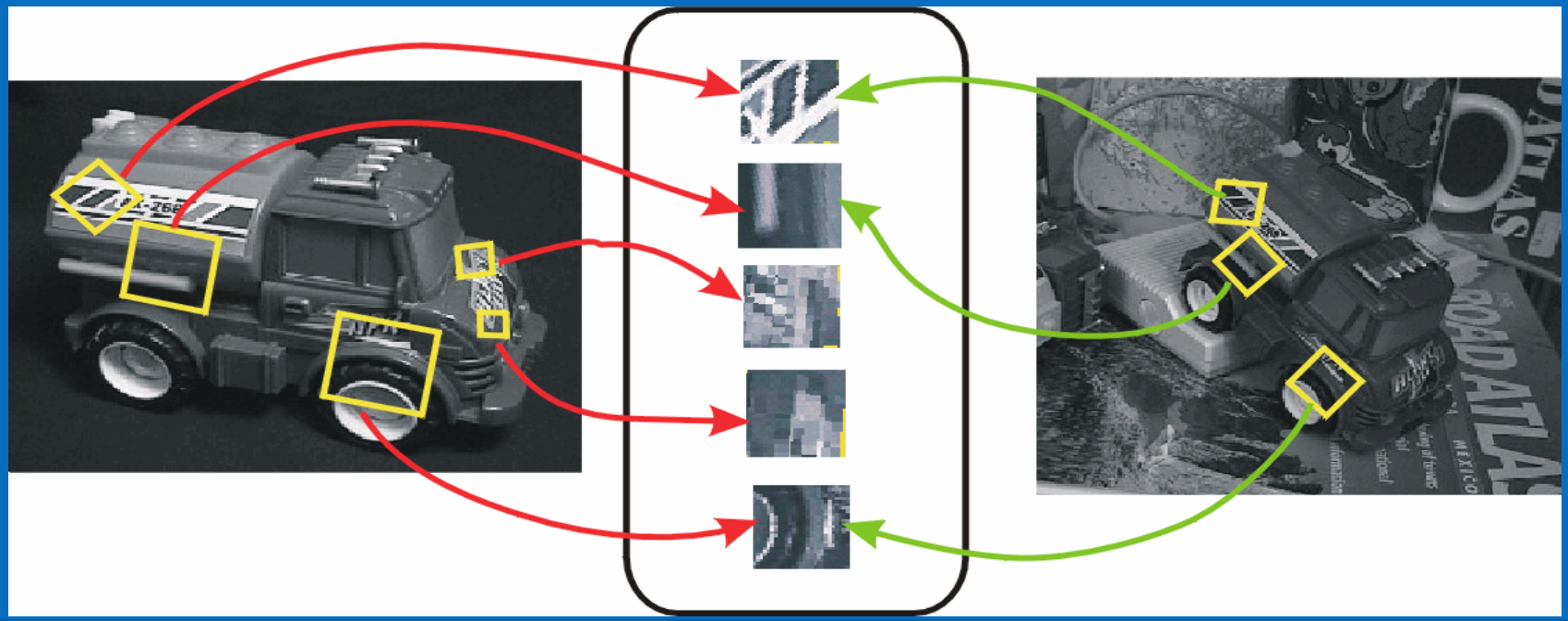
- We parallelized and optimized SIFT on the multi-core SMP/CMP system.
- Improve performance from nearly 1 FPS (current serial algorithm) to 30 FPS for HDTV images on 64-core CMP
- Our findings:
  - 9~11x speedup on 16-core SMP and 38~52x speedup on 64-core CMP
  - Load imbalance is the primary limiting factor for scalability
  - Coherent traffic on multi-socket SMP is a cause of performance loss

Outline

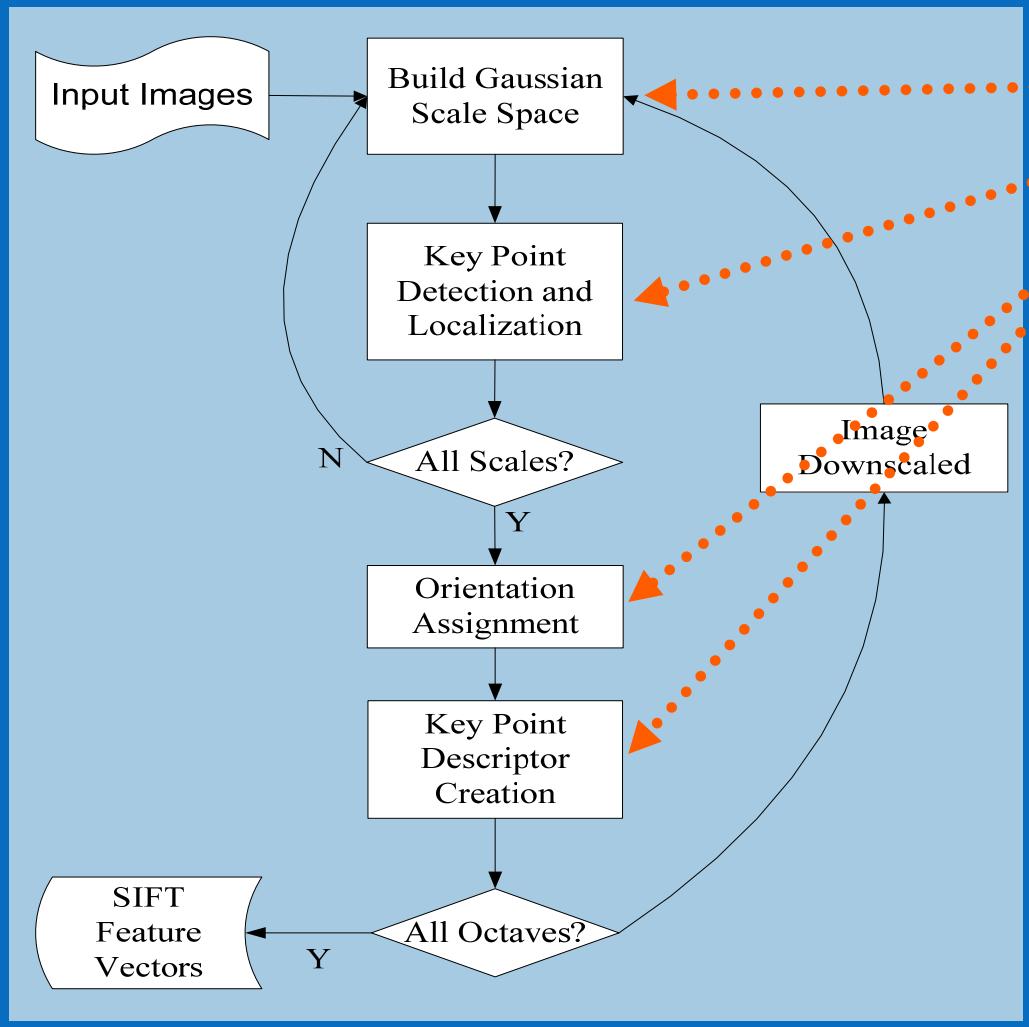
- SIFT algorithm introduction
- SIFT parallelization and optimization
- Performance analysis on the 16-core SMP
- Performance analysis on the 64-core CMP
- Conclusion

SIFT (Scale-Invariant Features)

- Image content is transformed into local feature coordinates that are invariant to translation, rotation, scale, and other imaging parameters
- Popular technique in computer vision, e.g., Panorama Generation, Microsoft Photosynth, and Content Based Image Retrieval



SIFT Flowchart and Key Modules



- BGSS
- KDL
- OA/KD
- MO (Matrix Operation)

The four modules take up to 99.8% of the runtime, and BGSS and OAKD take up to 80%

Parallelization Strategy

Granularity	Coarse	Fine
Parallelization	Between frames	Tiling within frame
Programmability	Easy	Difficult
Memory Requirements	High	Same as serial
Scalability	Poor	Good

intel

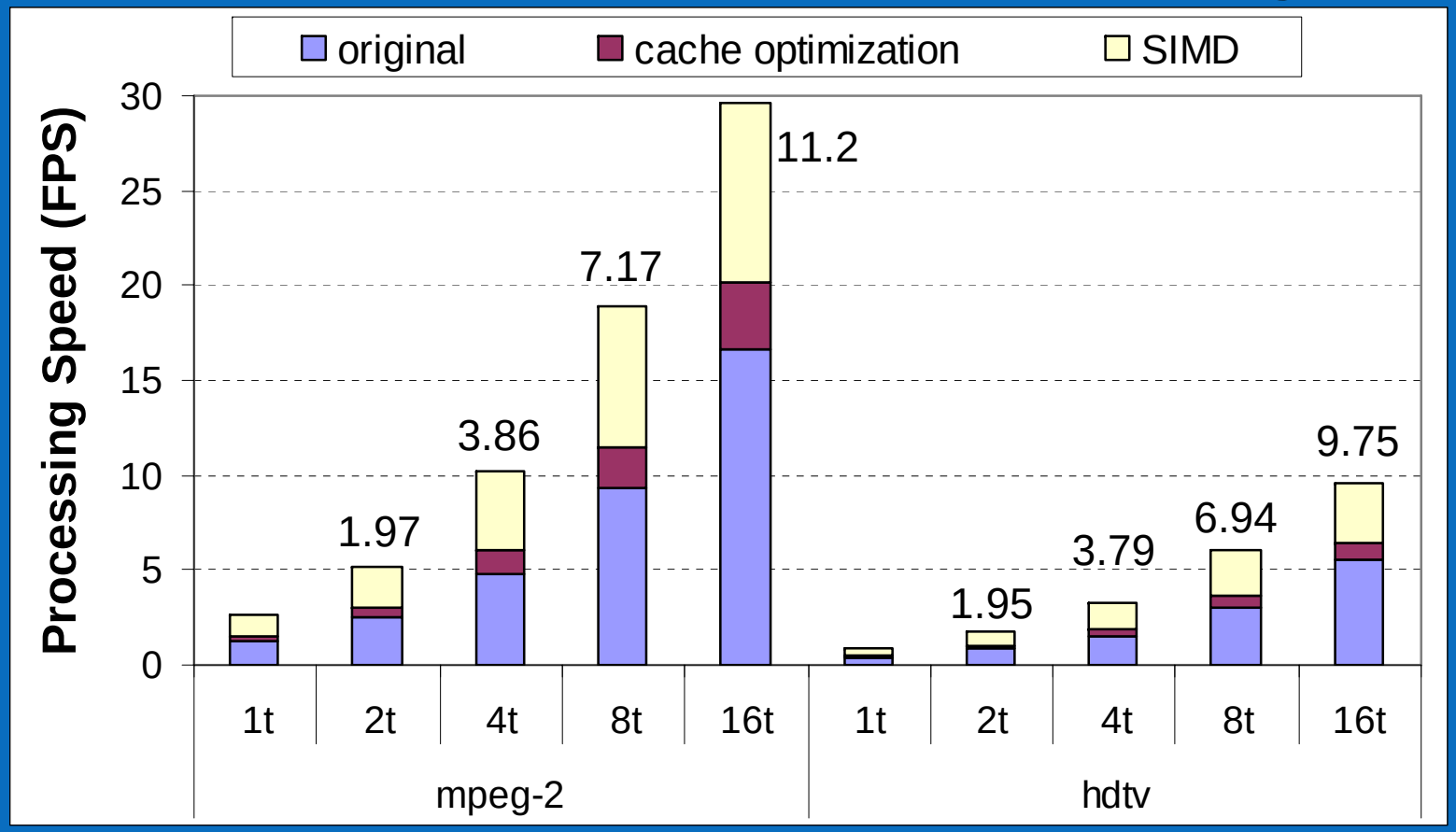
Optimizations

- SIMD (1.4x ~ 1.7x speedup)
  - Applied only to BGSS and MO
  - Irregular control flow and histogram operation limits speedup for KDL and OAKD
- Cache/memory optimizations (1.2x ~ 1.25x speedup)
  - Loop fission and loop interchanging
  - Eliminate false sharing
  - Remove redundant memory copy operations in BGSS
- Thread affinity (1.02x ~ 1.1x speedup)
  - Schedules threads on the cores which share the same L2 cache

Outline

- SIFT algorithm introduction
- SIFT parallelization and optimization
- Performance analysis on the 16-core SMP
- Performance analysis on the 64-core CMP
- Conclusion

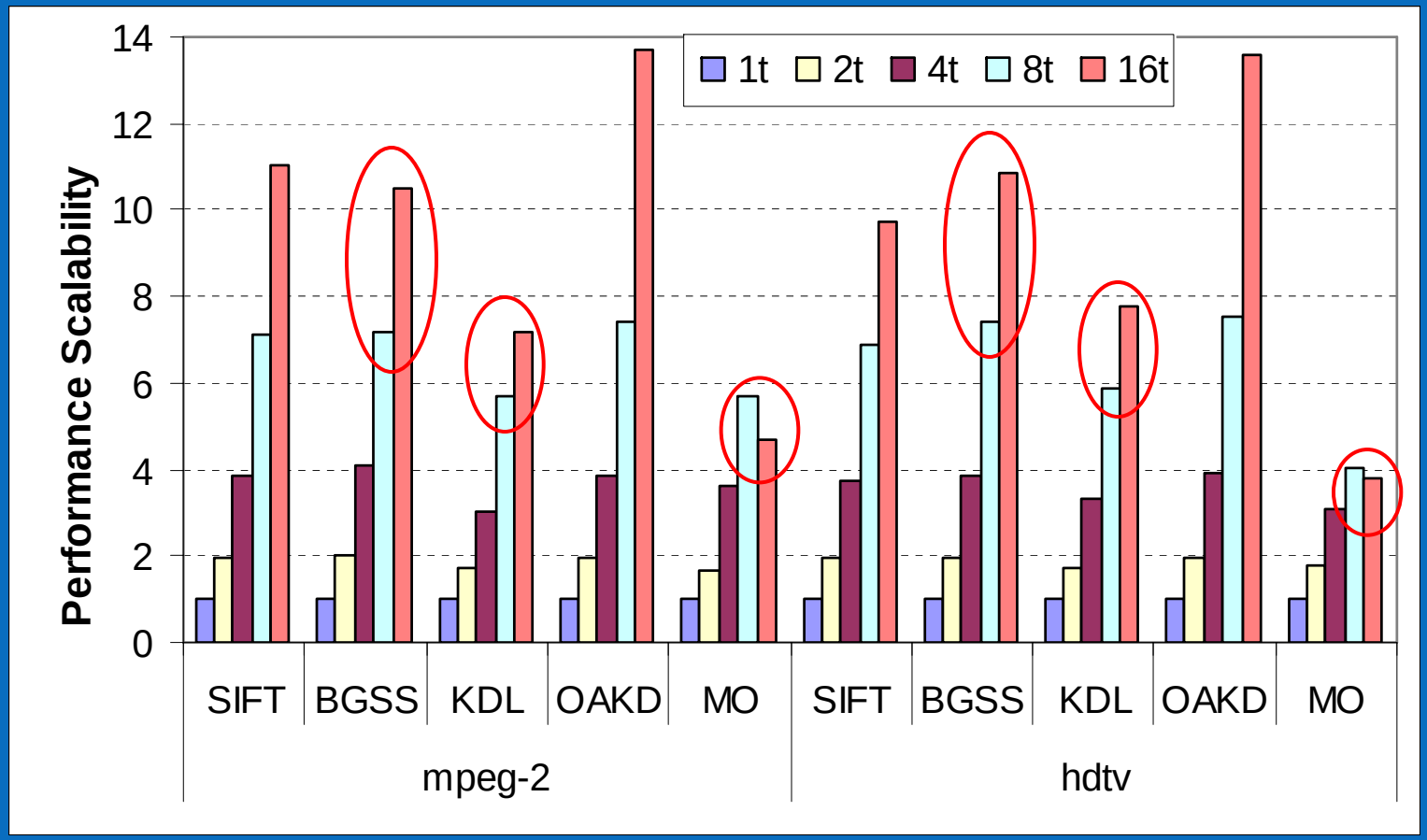
SMP Performance and Scalability



- 1.7x from SIMD speedup, 1.2x from memory optimization and 1.1x from thread affinity

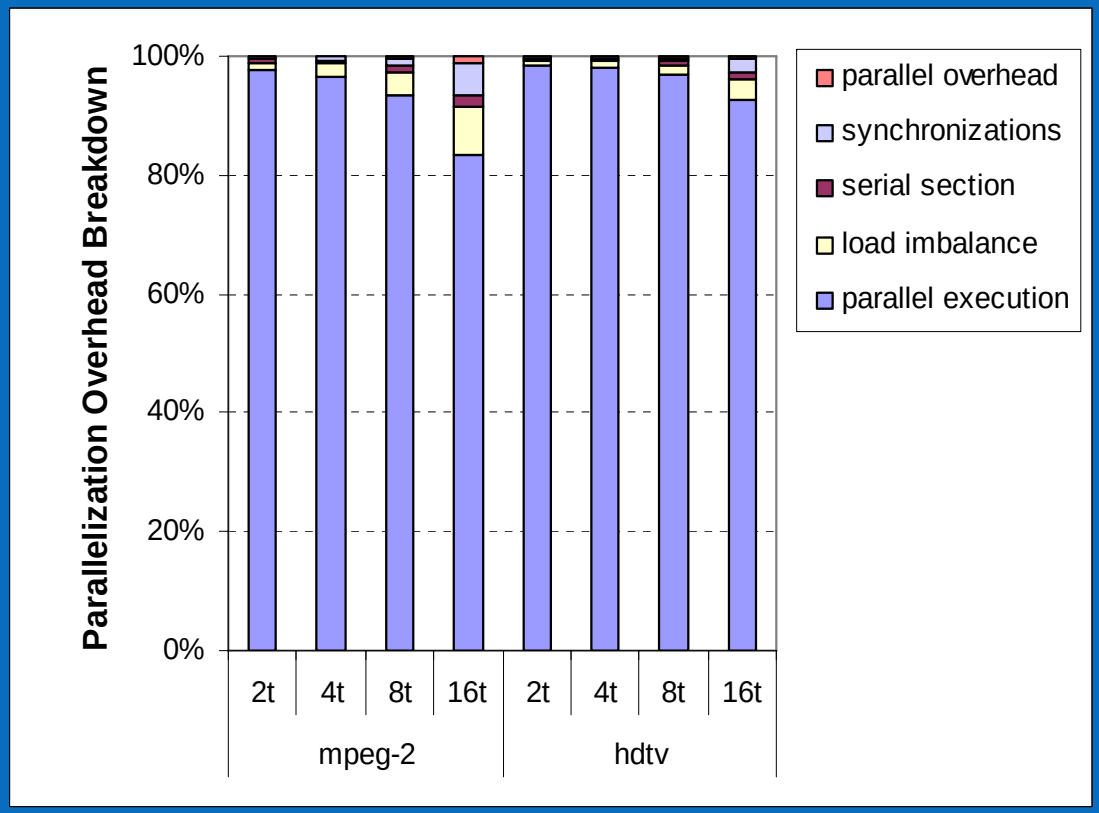
~10x Speedup for Mpeg2 and HDTV

Execution Time Breakdown



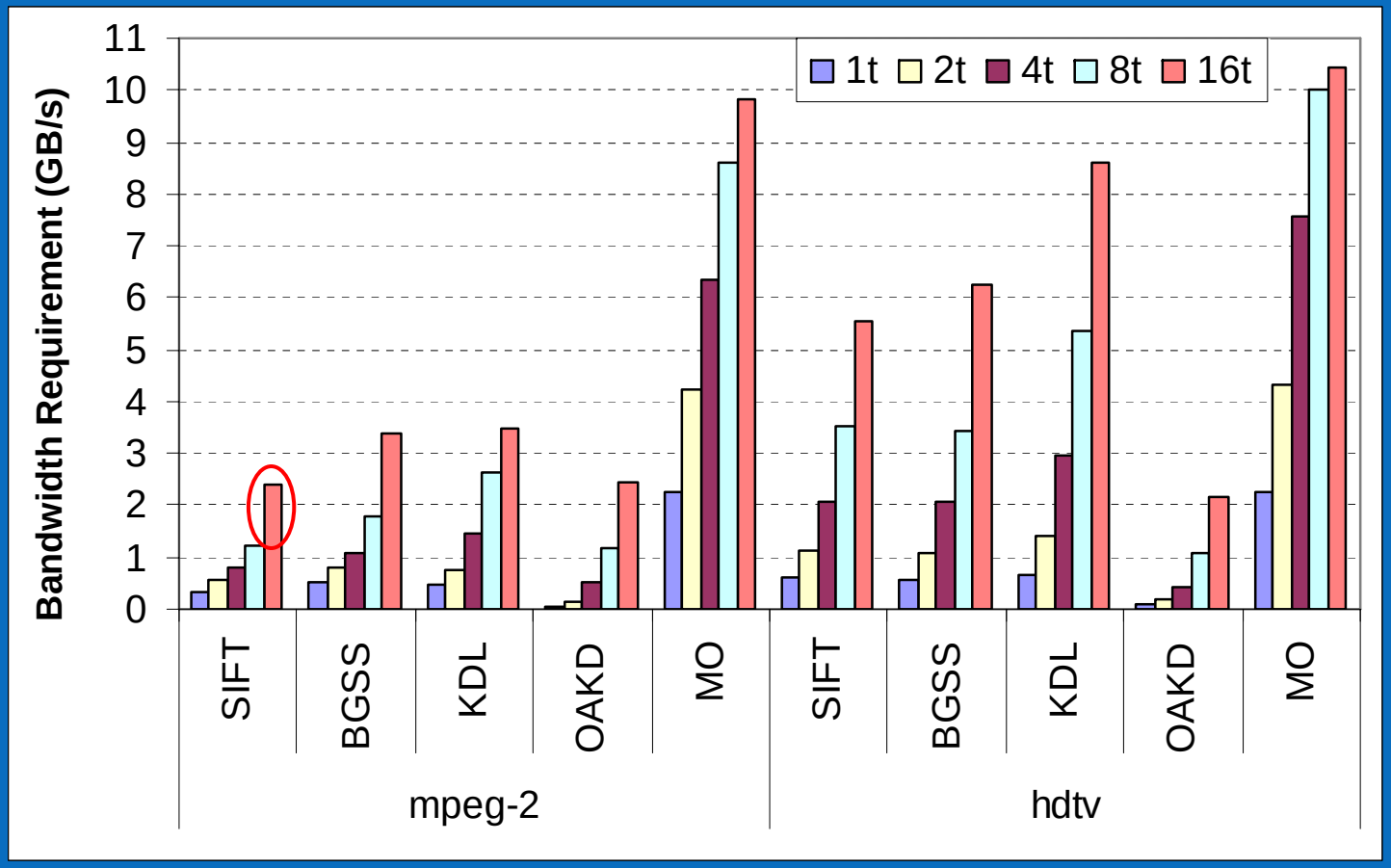
- Other than OAKD, BGSS / KDL / MO all have limited scalability on 16-core

Parallelization Overhead



- load imbalance: different # of key points detected in KDL
- synchronization: between row and column convolution in BGSS
  merging detected key points in KDL

Memory Bandwidth



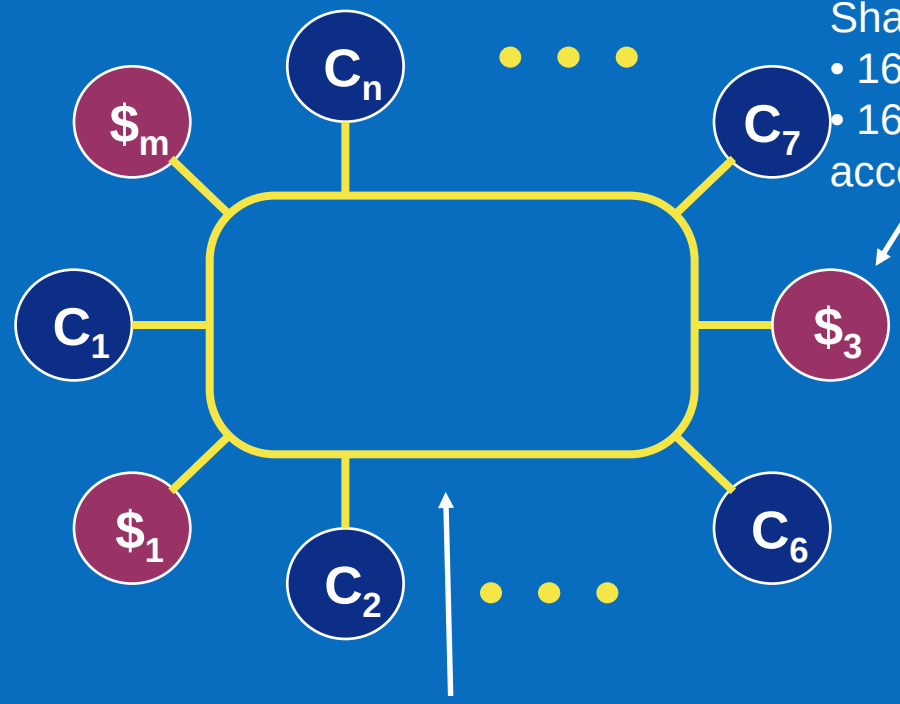
- Instantaneous bandwidth limits MO scalability

Outline

- SIFT algorithm introduction
- SIFT parallelization and optimization
- Performance analysis on the 16-core SMP
- Performance analysis on the 64-core CMP
- Conclusion

CMP Simulation Methodology

- 64-core CMP platform

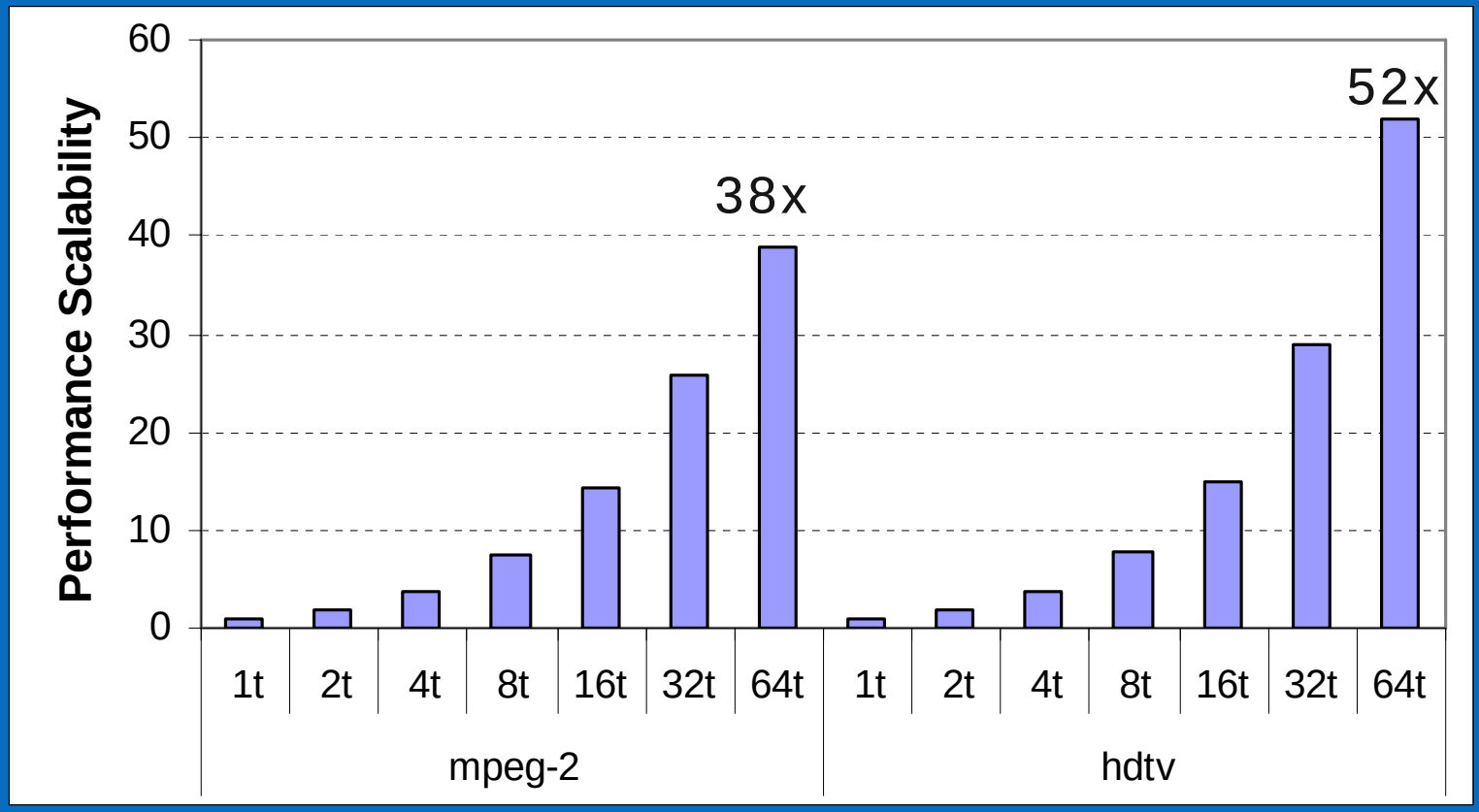


64 cores
• 2-wide, in-order
• 32KB L1, 4 ways, 3 cycle hits

Shared L2
• 16 partitions
• 16MB, 16 ways, 12 cycle access

Bi-directional ring
• 40 ring stops
• Full cache line moves 1 stop/cycle

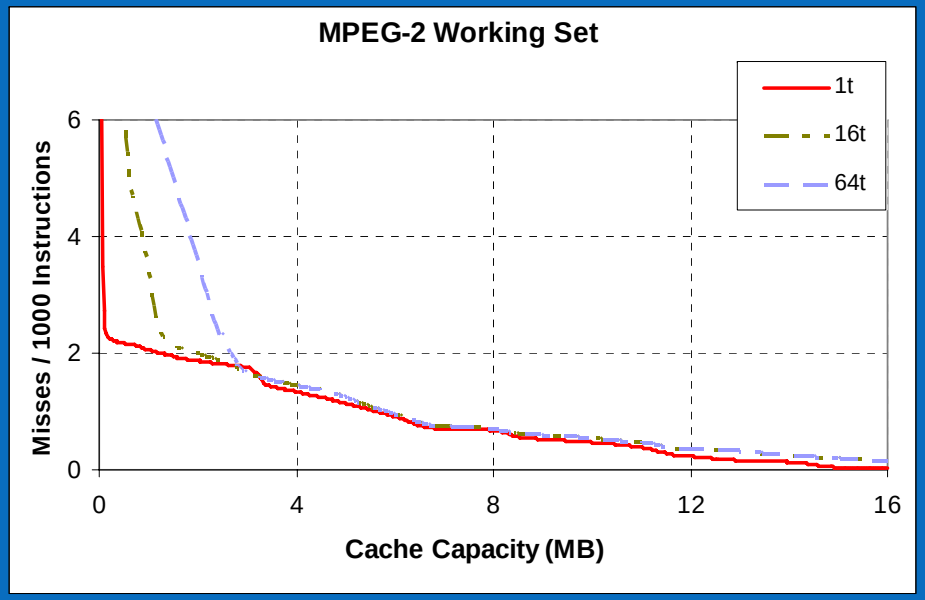
CMP Scalability

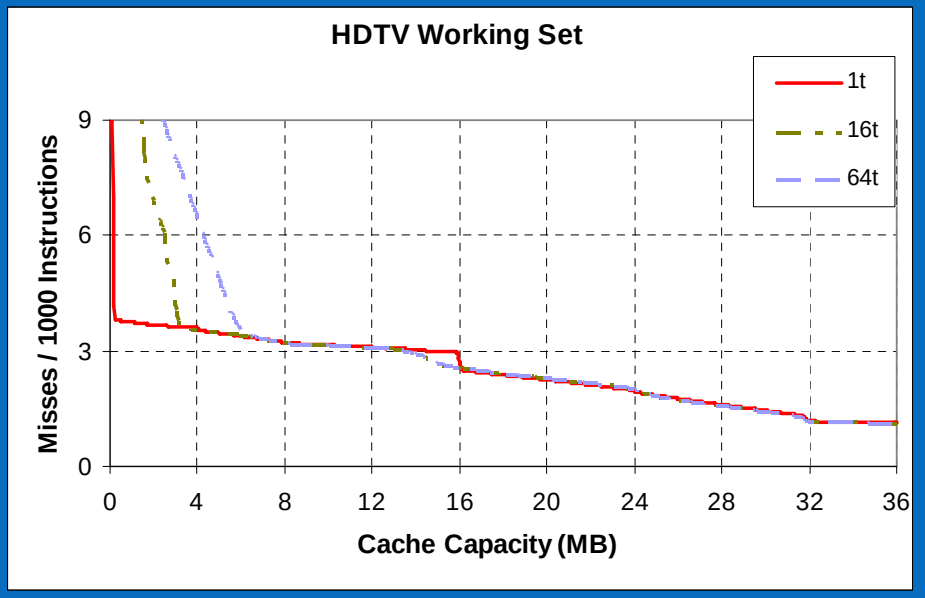


- Continue scaling to 64-core CMP
- Load imbalance and parallel overhead limit the scalability

Real-time performance for HDTV (33FPS with 64T)

Working Set

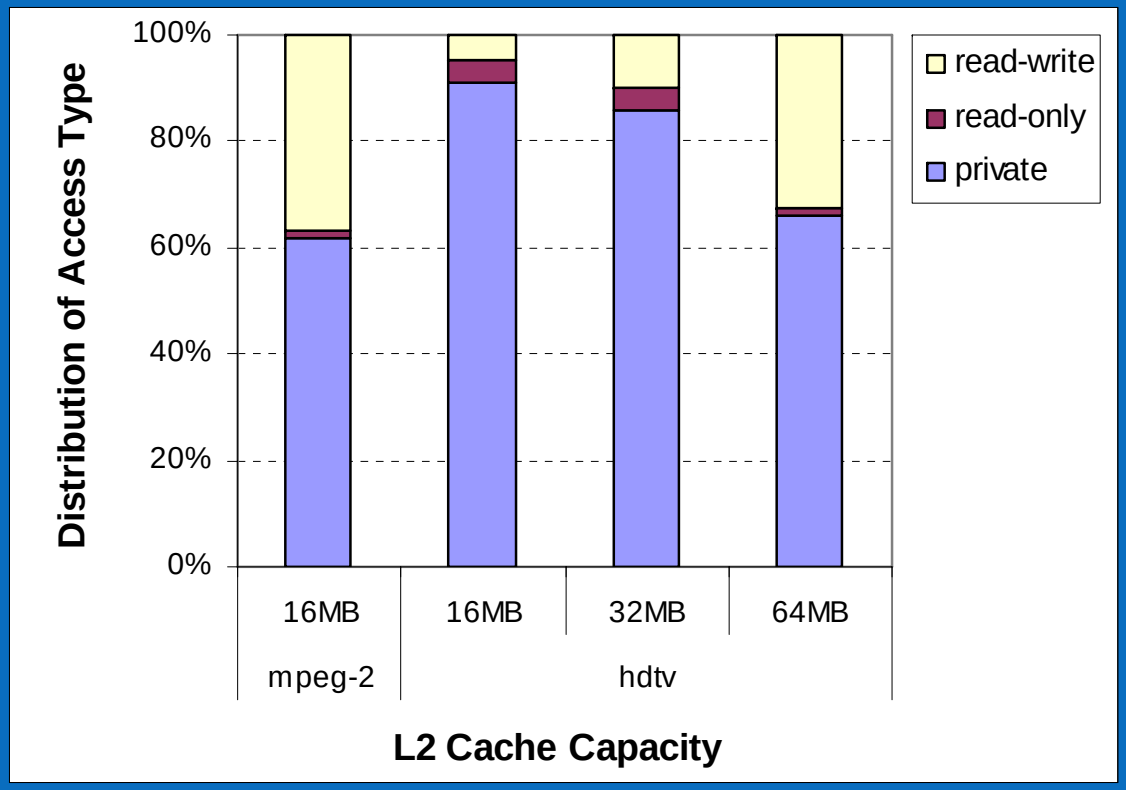




- 1st Level Working Set: 32kB per thread
- 2nd Level Working Set: 12MB for Mpeg2, 32MB for HDTV

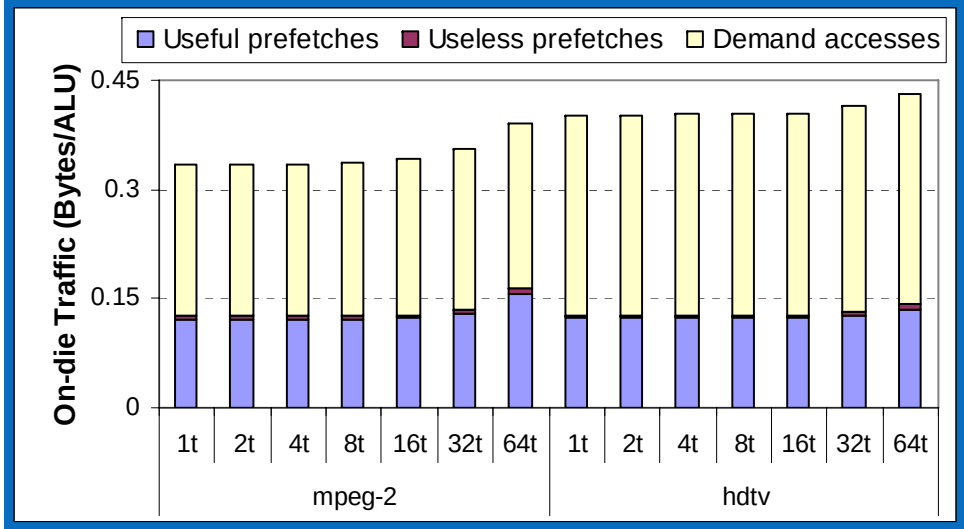
CMP's shared LLC works better for SIFT

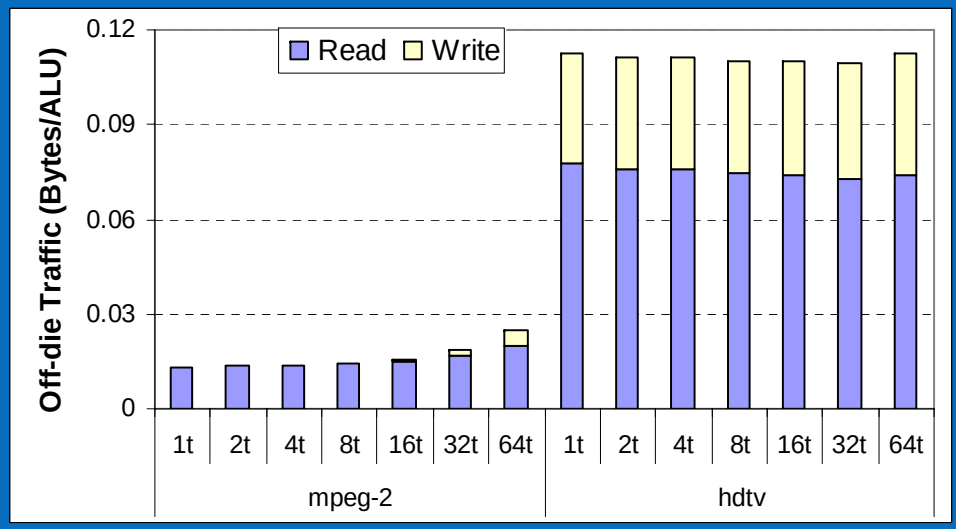
Thread Sharing



- SIFT contains ~ 40% read-write share accesses
- Eliminate off-chip coherent traffic improves performance

On/Off-die Traffic





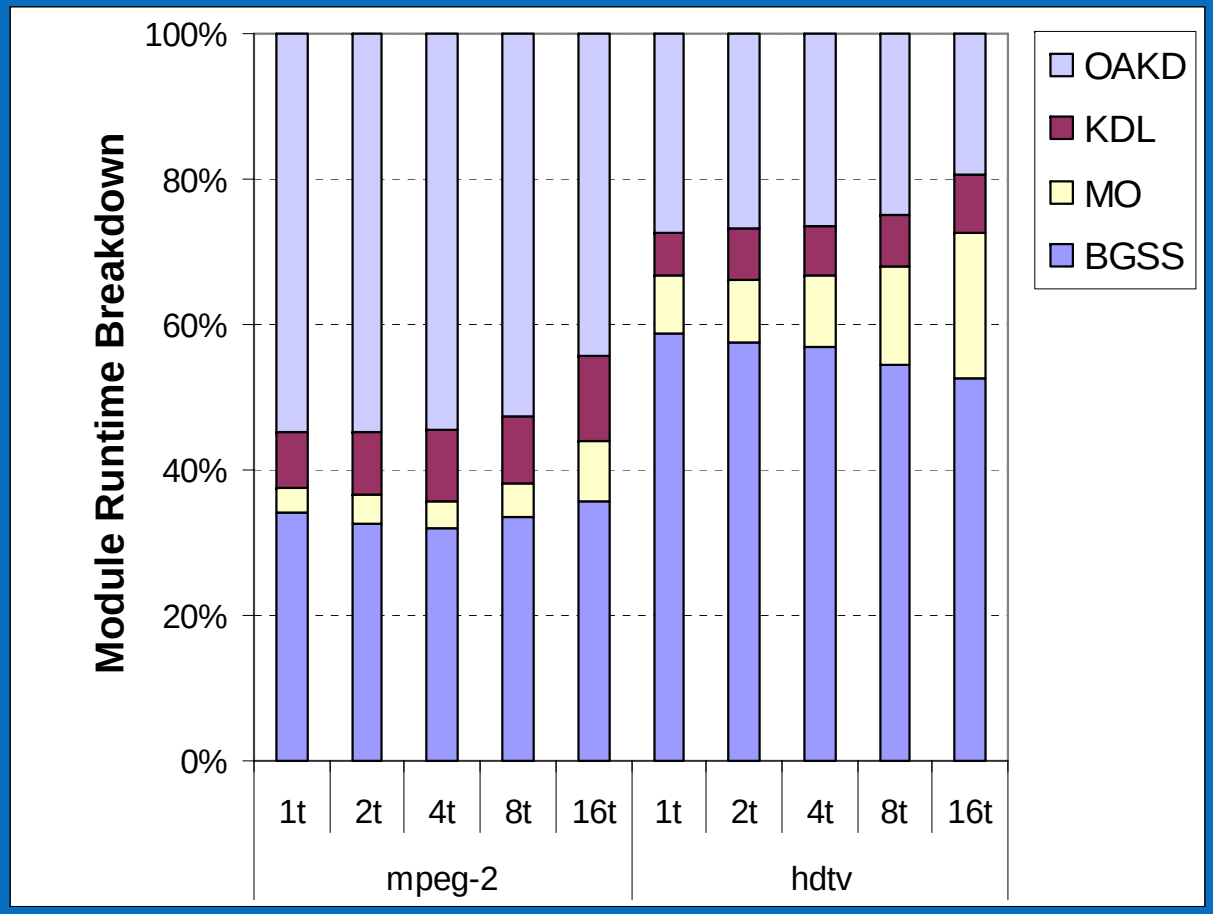
- Mild bandwidth requirements:
  - 54 GB/s on-die
  - 14 GB/s off-die

Summary of the Paper

- Parallelized and optimized SIFT
  - Achieved real-time (>30FPS) for mpeg2 and hdtv on64-core CMP
- Characterized performance on SMP and CMP systems
  - 9~11x speedup on 16-core SMP and 38~52x speedup on 64-core CMP
  - Load imbalance is the primary factor for scalability
  - CMP's shared LLC holds more of the working set and reduces external bandwidth requirement
  - Coherent traffic on multi-socket SMP is a cause of performance loss

Thanks

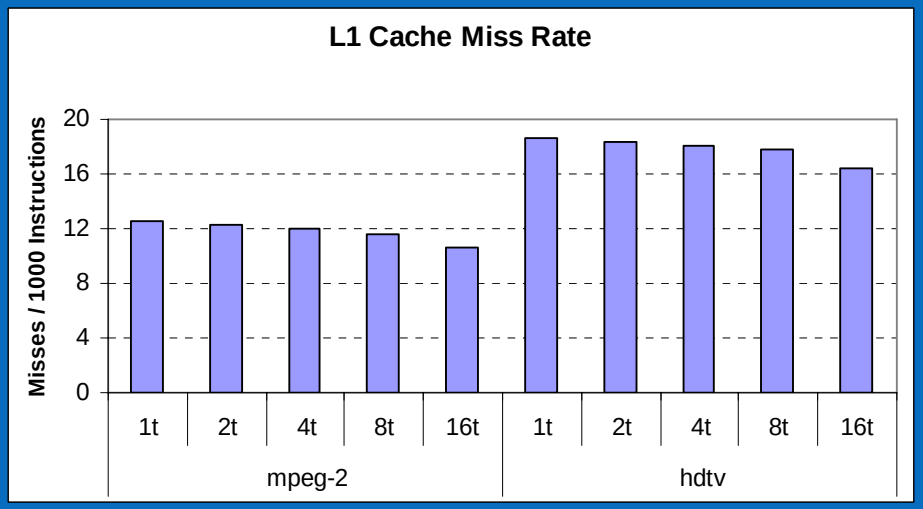
Modules Runtime Breakdown

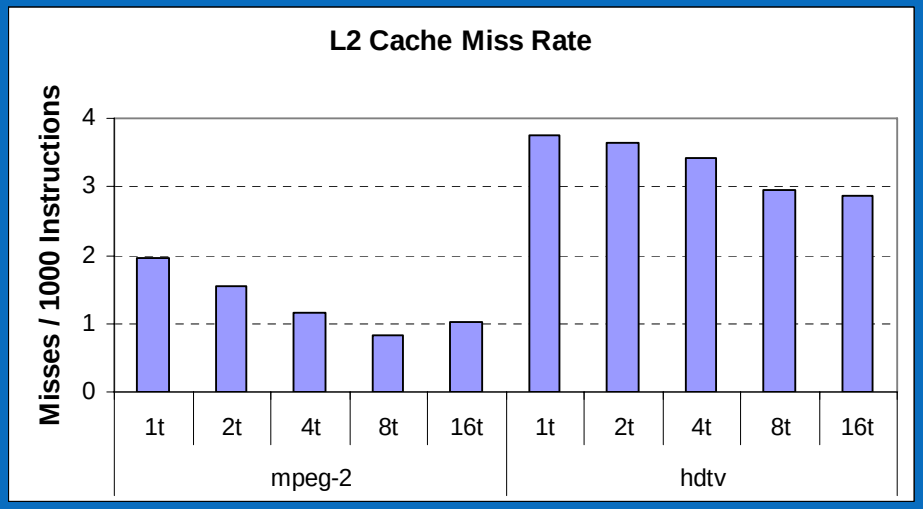


Parallelization

- Coarse-grained
  - process different frames in parallel
  - easy to implement
  - does not scale well because of its large aggregate working set
- Fine-grained
  - exploit row-wise or pixel-wise data parallelisms
  - straightforward and not hard to implement for SIFT
  - scale well and capable of on-line processing

Cache Misses on 16-core SMP





- It is because of the coherence miss that the L2 miss increases for MPEG-2 while scales from 8-thread to 16-thread
- Since HDTV has much larger working set, thus less shared data are contained in L2 cache, coherence miss does not have serious impact. The L2 cache miss keeps falling