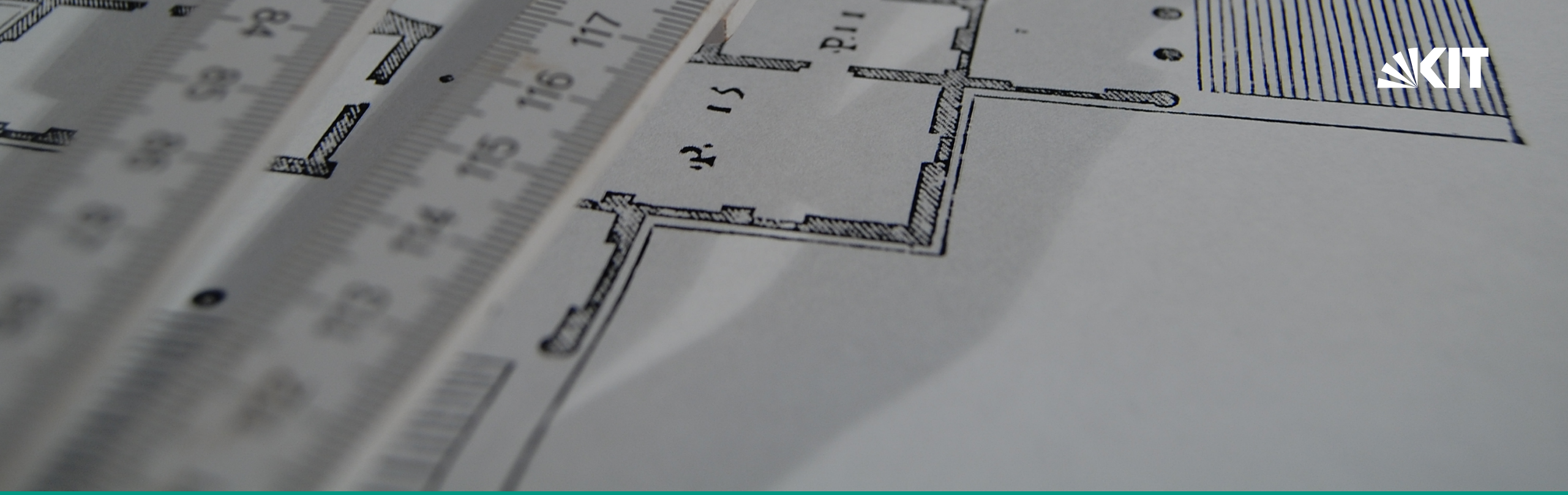


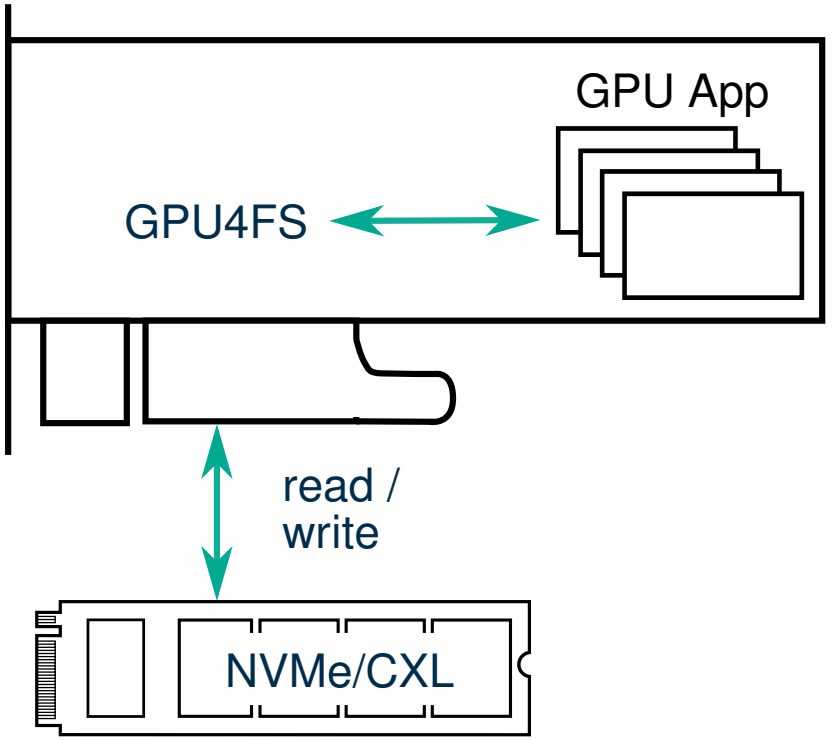
Are Your GPU Atomics Secretly Contending?

Peter Maucher, Nick Djerfi, Lennard Kittner, Lukas Werling, Frank Bellosa | October 13, 2025

Guideline-Driven Development

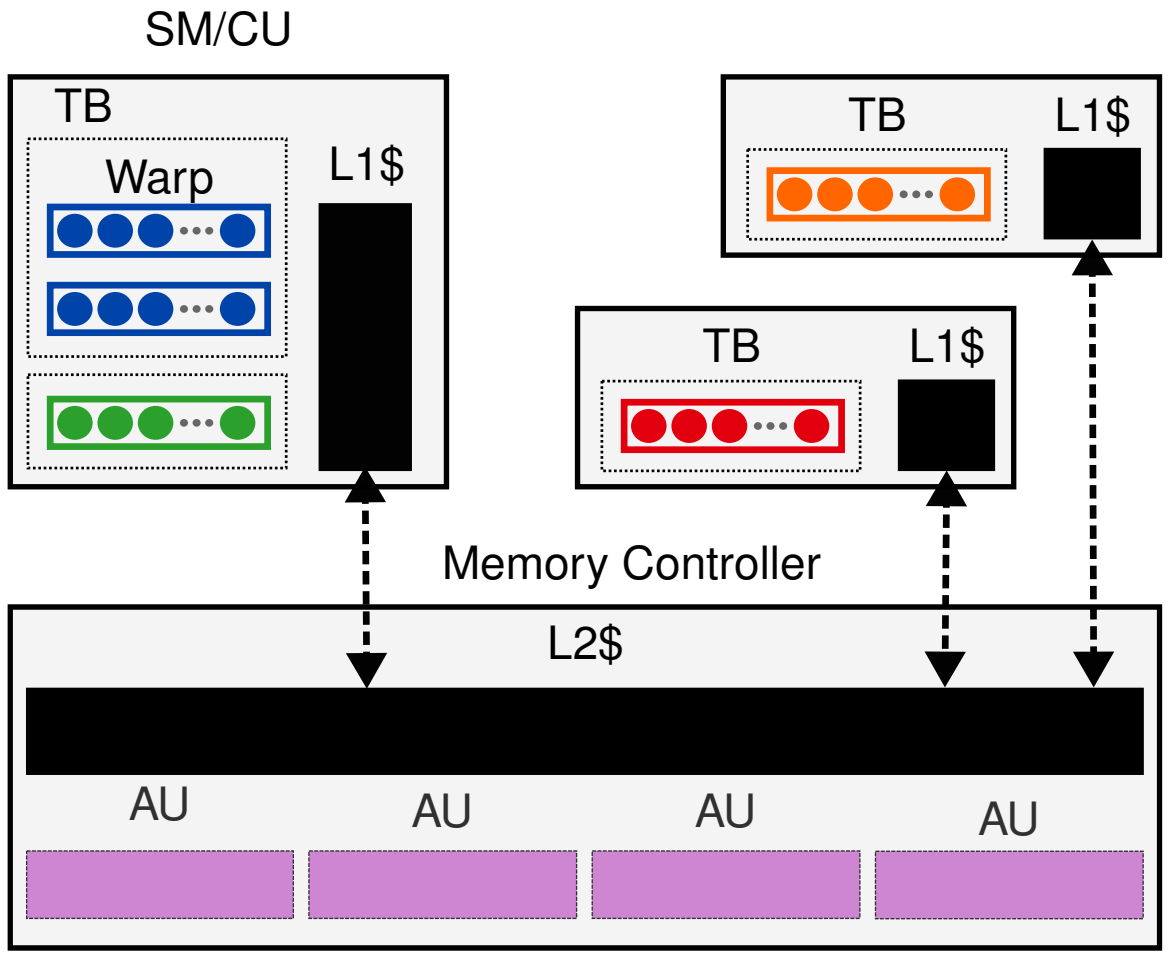
- GPUs: general purpose parallel applications
- GPU hardware: optimized for independent threads
- Inter-thread communication
  - No high-level API support, no reliable performance description
  - ⇒ Programmers: rely on imprecise *guidelines* with low-level operations
- This work: investigate guidelines
  - Failed for our GPU-side file system

⇒ Open source microbenchmark suite



GPU Memory Subsystem

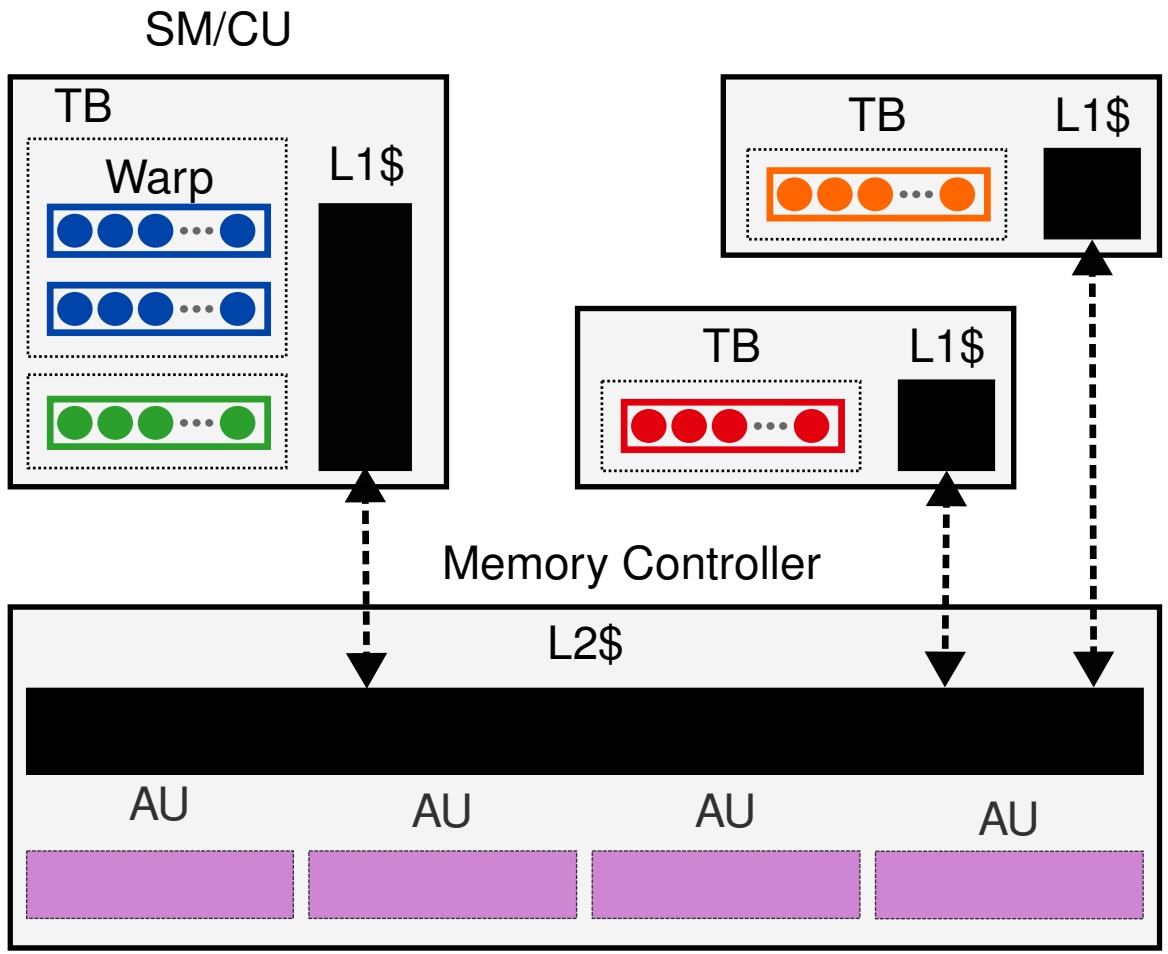
- 32 threads: run in lockstep in a *warp*
- Multiple warps: scheduled together in a *Thread Block (TB)* on
  - NVIDIA: *Streaming Multiprocessor (SM)*
  - AMD: *Compute Unit (CU)*
- Caches: (generally) private L1, shared L2
- Atomic operations: multiple scopes (per TB, GPU wide, system wide through bus)
- *Atomic Units (AU)*: execute operations, located at L2 cache



Guideline: Reduce Warp-Level Contention

Avoid accessing a single atomic variable with multiple threads in the same warp:

- Inner-warp communication possible and efficient
- Guideline: only one access per warp to reduce contention
- Problem: additional code and inner-warp synchronization required
- Finding: lightly contended atomics need no inner-warp synchronization

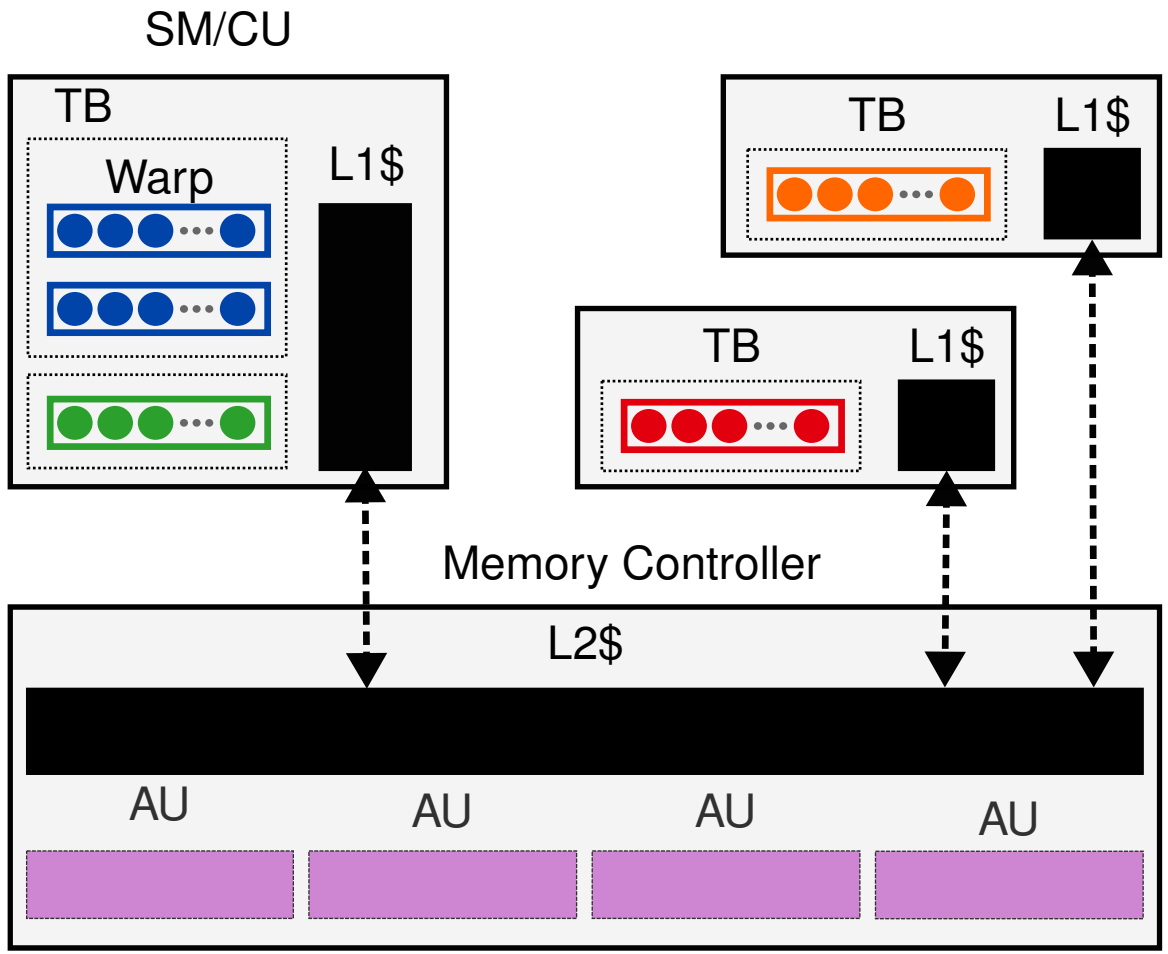


Guideline: Reduce Cross-Contention

To avoid contention, place your atomic variables in different cache lines:

- Atomic units: located at global caches
- Assumption: one atomic unit per cache line
  - ⇒ Atomic variables on distinct cache lines do not cross-contend
- Finding: atomic variables on distinct cache lines can cross-contend in multiple cases

⇒ Space atomic variables 256 B (and more on AMD) apart

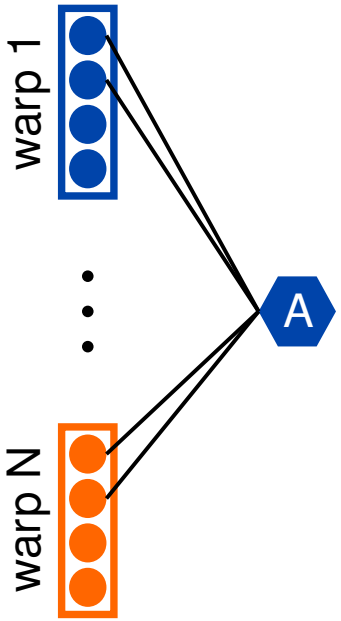


Benchmark Suite

- Multi-vendor: AMD, NVIDIA
- Multi-generation: GPUs from 2020 to 2023
- Low-level API and measurements
- Explore performance implications
- Inform future high-level primitives
- Open source microbenchmark suite designed for further tests
- *No* microarchitecture reverse engineering

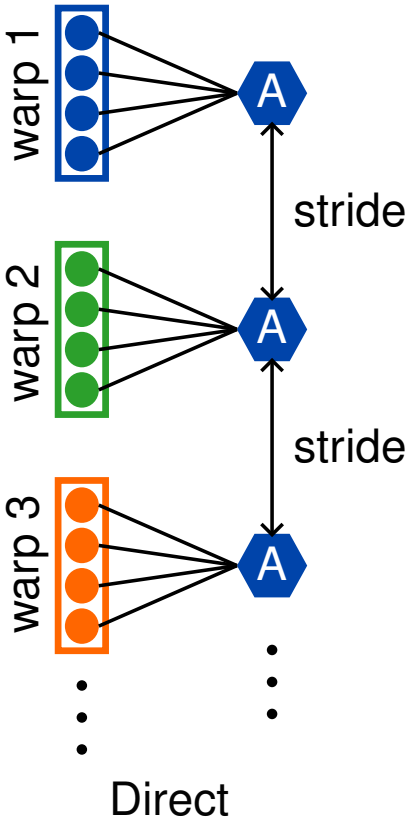
Threads per Warp

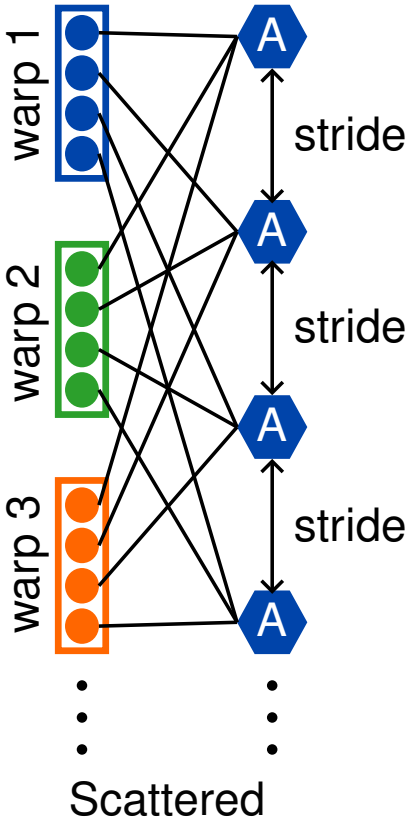
- Explore effects of warp optimizations
- Access single atomic variable (max possible contention)
- Scale in two dimensions:
  - Number of warps
  - Number of threads per warp (1-32)
- Expectation: contention scales linearly with number of threads



Cross-Contention

- Explore interactions between accesses to independent atomic variables
- Scale in two dimensions:
  - Number of variables
  - Memory stride
- Expectation: faster accesses if atomic variables on different cache lines
- Two sub-tests for different access patterns:
  - *Direct*: one warp accesses same atomic
  - *Scattered*: adjacent threads access adjacent atomics



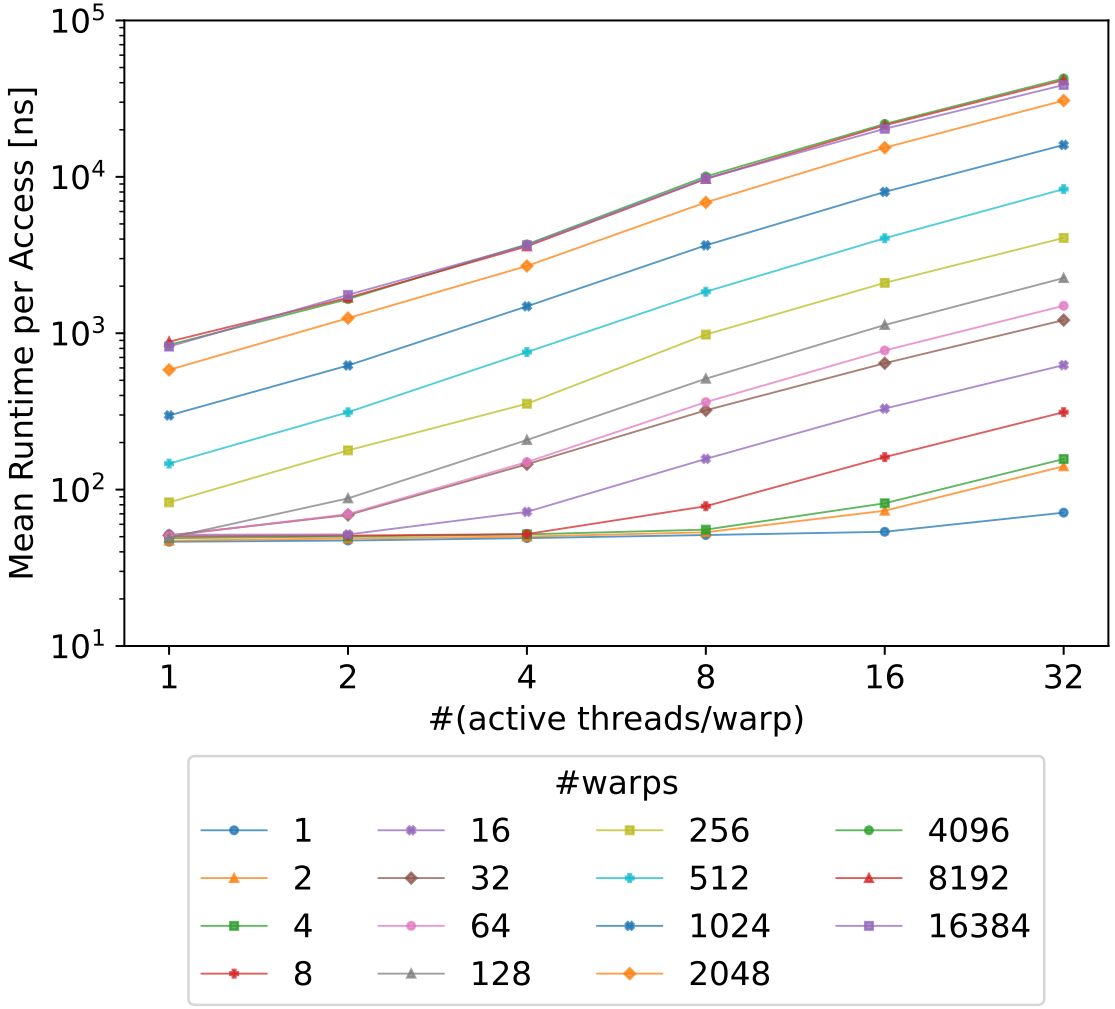


Test Systems

- Two vendors: AMD (A) and NVIDIA (N)
- Two generations each
- Consumer-grade GPUs for ease of access
- Future work: enterprise-class GPUs, results probably transferable

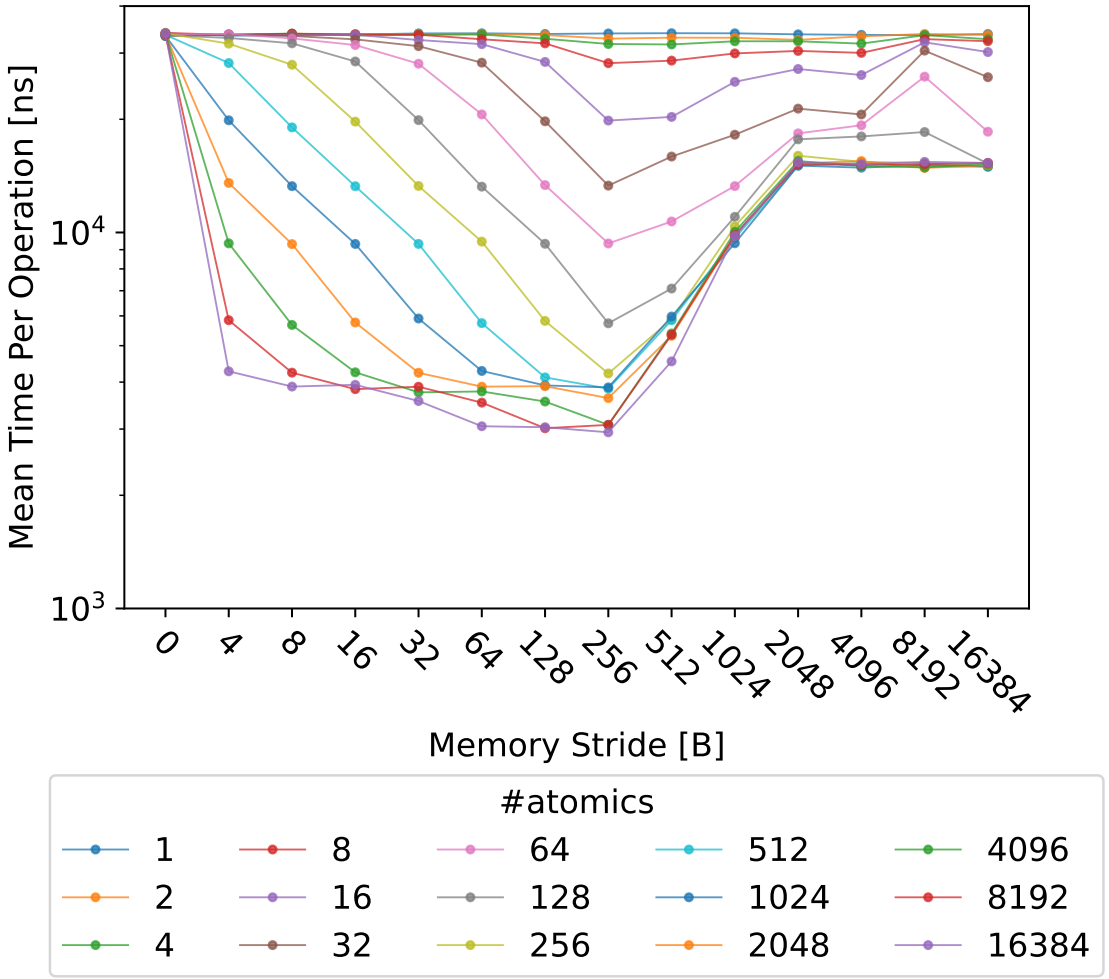
ID	GPU	# SM/CU	max clock
A1	AMD RX 6950 XT	80	2310 MHz
A2	AMD RX 7900 XTX	96	2500 MHz
N1	NVIDIA RTX A4500	56	1650 MHz
N2	NVIDIA RTX 4070	46	2475 MHz

Threads per Warp



- GPU A2: AMD RX 7900 XTX (all other GPUs similar)
- Runtime *independent* of distribution to warp
- Accesses fast for ≤ 32 total threads
- > 32 total threads: doubling of threads doubles runtime
- Ceiling ≥ 4096 warps: full GPU occupancy, so no further contention

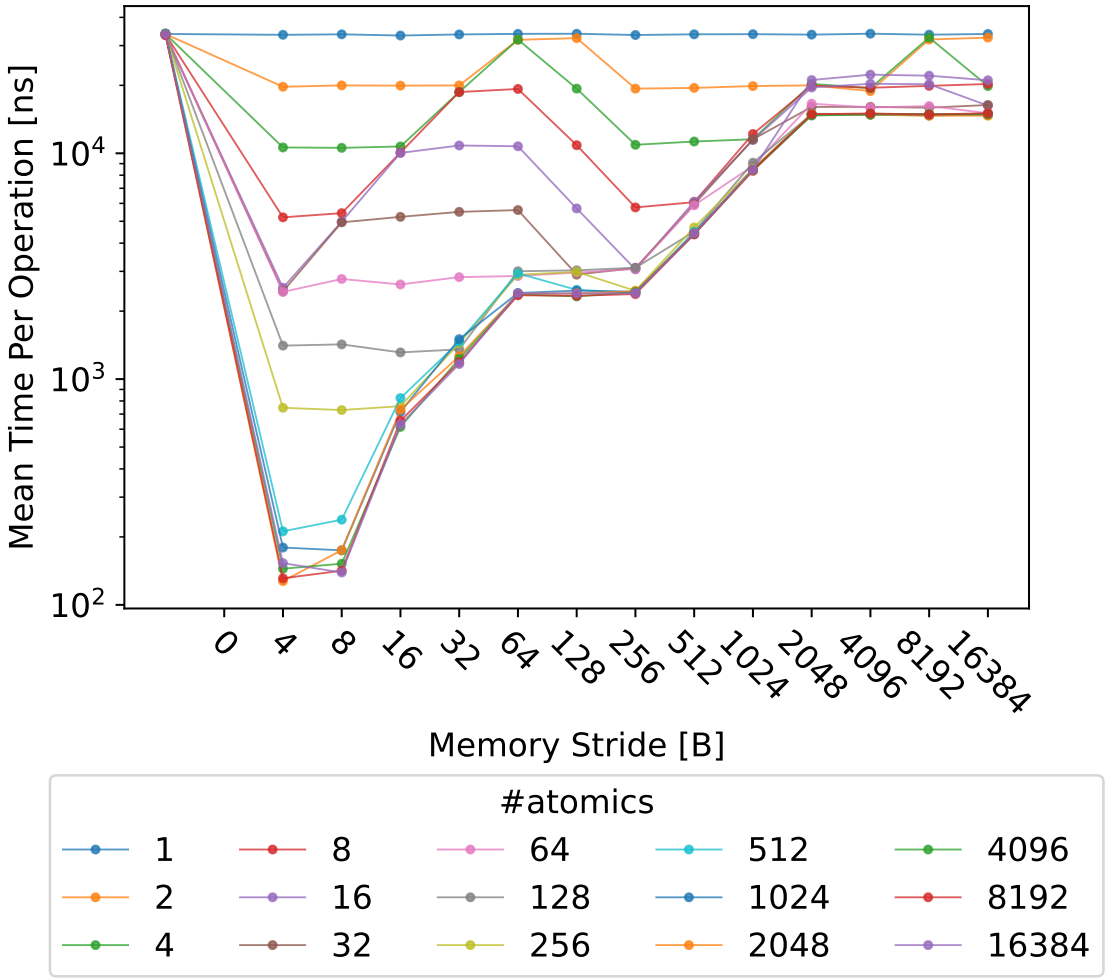
Cross-Contention



- GPU A2: AMD RX 7900 XTX (A1 similar)
- Focus: two atomic variables: min stride where access time decreases ⇔ no cross-contention:
  - Cache line size for all four GPUs: 128 B
  - Access time only decreases with 256 B (also on N1, N2)
- Access time ⇔ cross-contention reduces with number of variables
- On AMD: variables 4096 B apart *show* cross-contention

⇒ Avoid cross-contention: atomic variables 256 B apart (and slightly more on AMD)

Scattered Cross-Contention



- GPU A2: AMD RX 7900 XTX (A1 similar), N1, N2 similar except 4096 B-effects
- Large strides: similar to non-scattered
- Small strides: much faster ⇒ some hardware-acceleration in operation

⇒ High-level primitives: more information basically free (!)

Conclusion

- Guidelines correct in many cases
- For few threads/lightly contended atomics:
  no warp-level optimizations required
- To avoid cross-contention:
  - Space atomic variables 256 B / two cache lines apart
  - On AMD: add extra padding to avoid 4096B effect
- Design high-level primitives with the hardware in mind
- Benchmark your atomic applications
- Verify your guidelines $\Rightarrow$ use our microbenchmarks

`https://github.com/KIT-OSGroup/GPUAtomicContention`