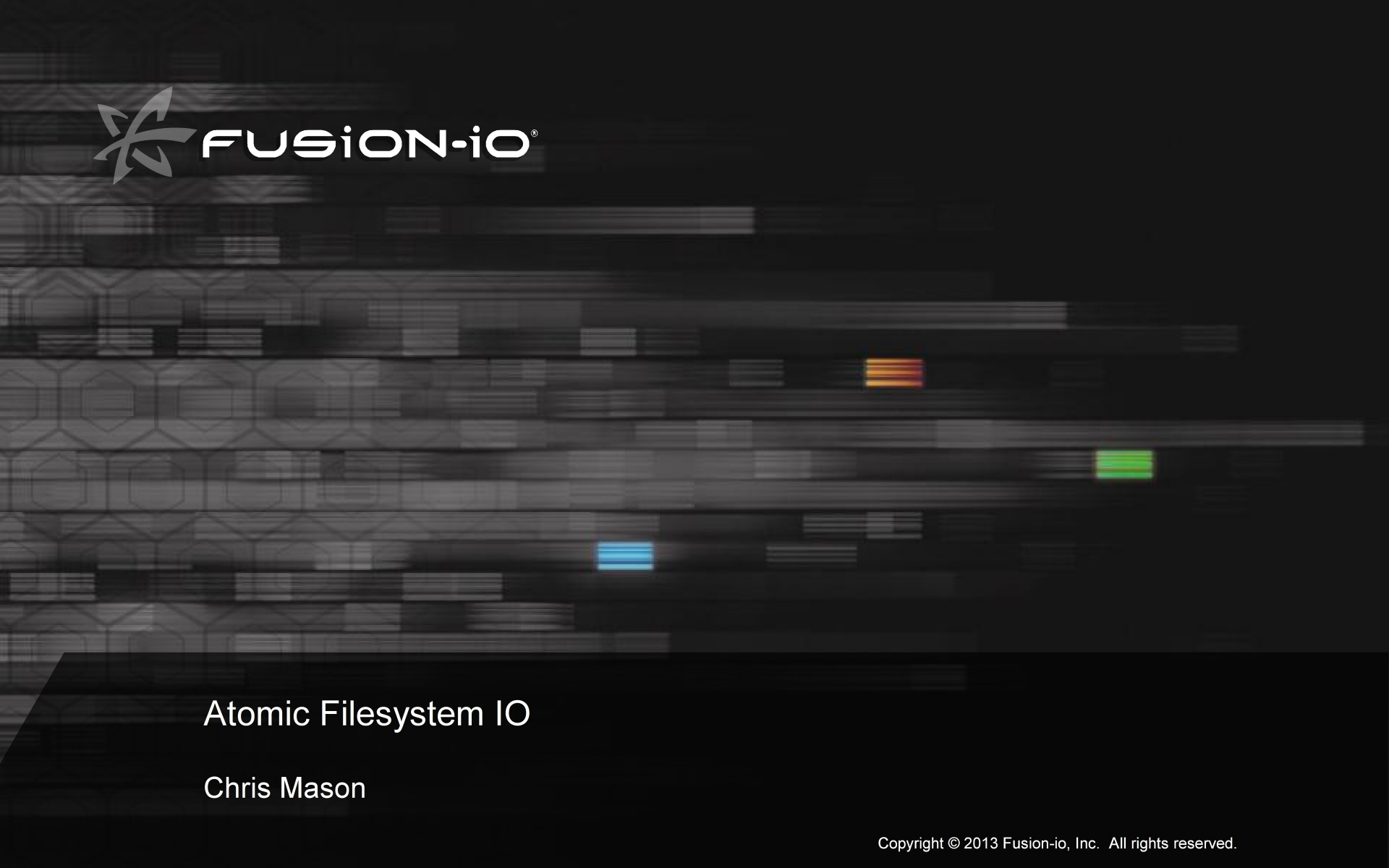


Atomic Filesystem IO

Chris Mason

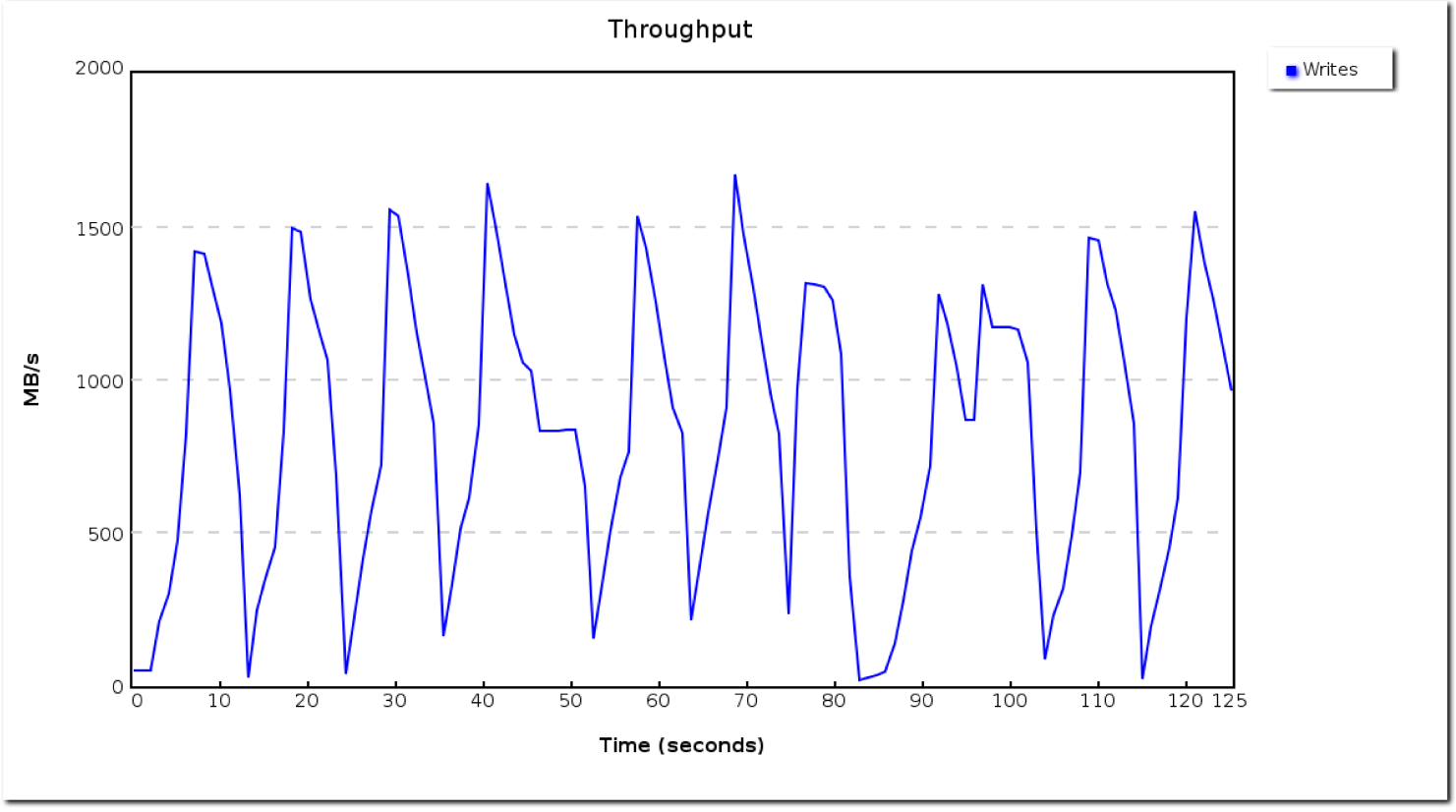
Tuning Filesystems on Flash

FUSION-IO

Most of the time,
The software is the bottleneck

Example: Large Buffered Writes

FUSION-IO



Storage provides features the filesystems and applications are not using

Typical Filesystem Commit

FUSION-IO

- Write file data
  - Wait
- Log critical metadata
  - Wait
- Write commit block or super block
  - Wait
- These waits are critical to filesystem integrity

Waiting…

- Commit latency is critical to many database workloads
- Fewer waits would mean less latency
- Fewer waits would increase average queue depth and average IO size
  - Less flash garbage collection

Atomic IO

FUSION-IO

- Waiting gives us IO ordering guarantees
- Flash and high end storage arrays can explicitly provide the same thing
- T10 recently approved an initial atomic spec
  - So far, it only deals with contiguous IO
  - Vectored IO is coming as well

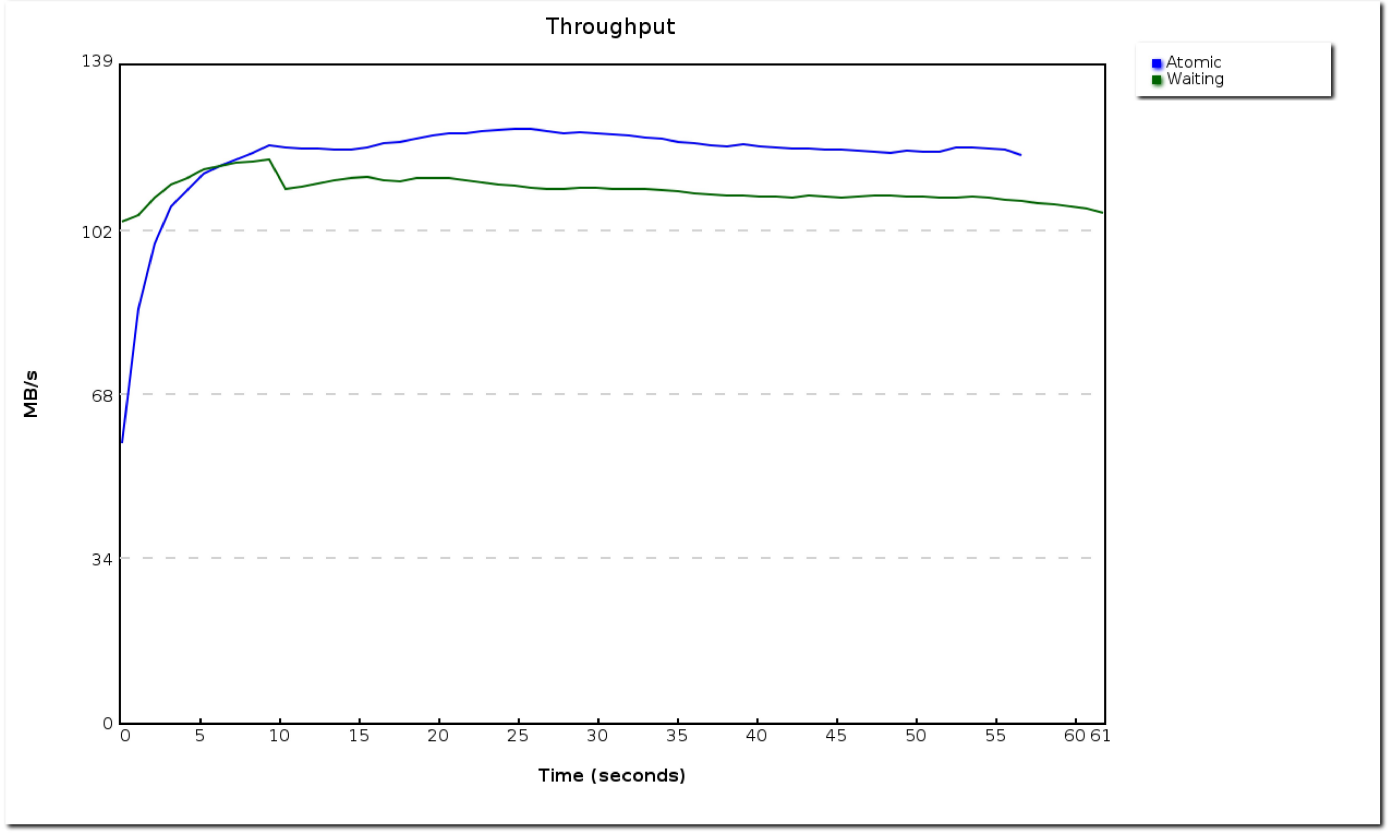
Btrfs Fsync Log

FUSION-IO

- Similar to a logical log
- Each commit waits twice
  - Combined file data and tree log metadata wait
  - Super block wait
- Cutting one wait saves 10-15%
  - Highly loaded systems see bigger benefits due to scheduling latencies

Atomic Performance Benefits

FUSION-IO



Atomic Applications

FUSION-IO

- MariaDB and Percona are both atomic enabled today with the Fusion-io SDK
- Limited to DirectFS and the block device
- Replaces the double-buffering mode
- 43% more transactions per second
- ½ the wear

O_DIRECT | O_ATOMIC

FUSION-IO

- Bringing generic atomic support to all the filesystems
- Vectored atomics required for non-contiguous filesystem blocks
- Most O_DIRECT users can benefit
  - No more fractured blocks
- O_ATOMIC adds new restrictions on maximum IO size

O_DIRECT | O_ATOMIC with AIO

FUSION-IO

- Asynchronous IO API can send multiple operations to multiple files in a single system call
- Can be extended to group these operations into an atomic unit
- Can simplify database transaction engines while improving performance
- May (finally) expand the small list of O_DIRECT and AIO users

Plumbing the Atomics

FUSION-IO

- Device driver advertises a maximum number of atomic segments
- Plugging architecture extended to hold IO until an explicit unplug is called
- AIO and DIO modified to hold plugs for atomic writes

Atomic Error Handling

FUSION-IO

- Filesystems must remain consistent if an atomic group becomes too large to finish
- Filesystems must not deadlock waiting for plugged IO
- New test suites must be created to verify hardware atomics are working properly

Atomic Patch Status

FUSION-IO

- Vectored atomic support in Fusion-io driver
  - NVME patches in progress (Jens Axboe)
  - SCSI patches will probably wait for specification
- Vectored atomic support during Btrfs transaction commits
  - Other filesystems patches pending
- Initial atomic plugging completed
  - Many cleanups left

Next Steps

FUSION-IO

- Finalize API changes and submit for review
  - AIO and DIO changes will require many iterations
- Extend FIO to benchmark and test atomic IO
- Enable atomics in more applications

Atomic Limitations

FUSION-IO

- Relatively small maximum IO size
- AIO and DIO code complexity
  - Every filesystem warps the DIO code in different ways
- Potential performance variations between hardware vendors
  - Fusion-io atomics are almost zero cost
- Does not work across multiple devices
  - Software RAID and LVM need modifications to provide atomics to applications

Atomic Alternatives

FUSION-IO

- Explicit IO ordering
  - Many transaction engines only need the commit block to be written last
  - Much easier to provide large ordered writes than large atomic writes
  - Past ordering attempts caused many problems
- Explicit IO checksumming
  - Checksum a group of blocks
  - After a crash, reject groups that do not self validate
  - Well suited to append only logs

THANK YOU

Chris Mason chris.mason@fusionio.com