ScoutFS: POSIX Archiving at Extreme Scale

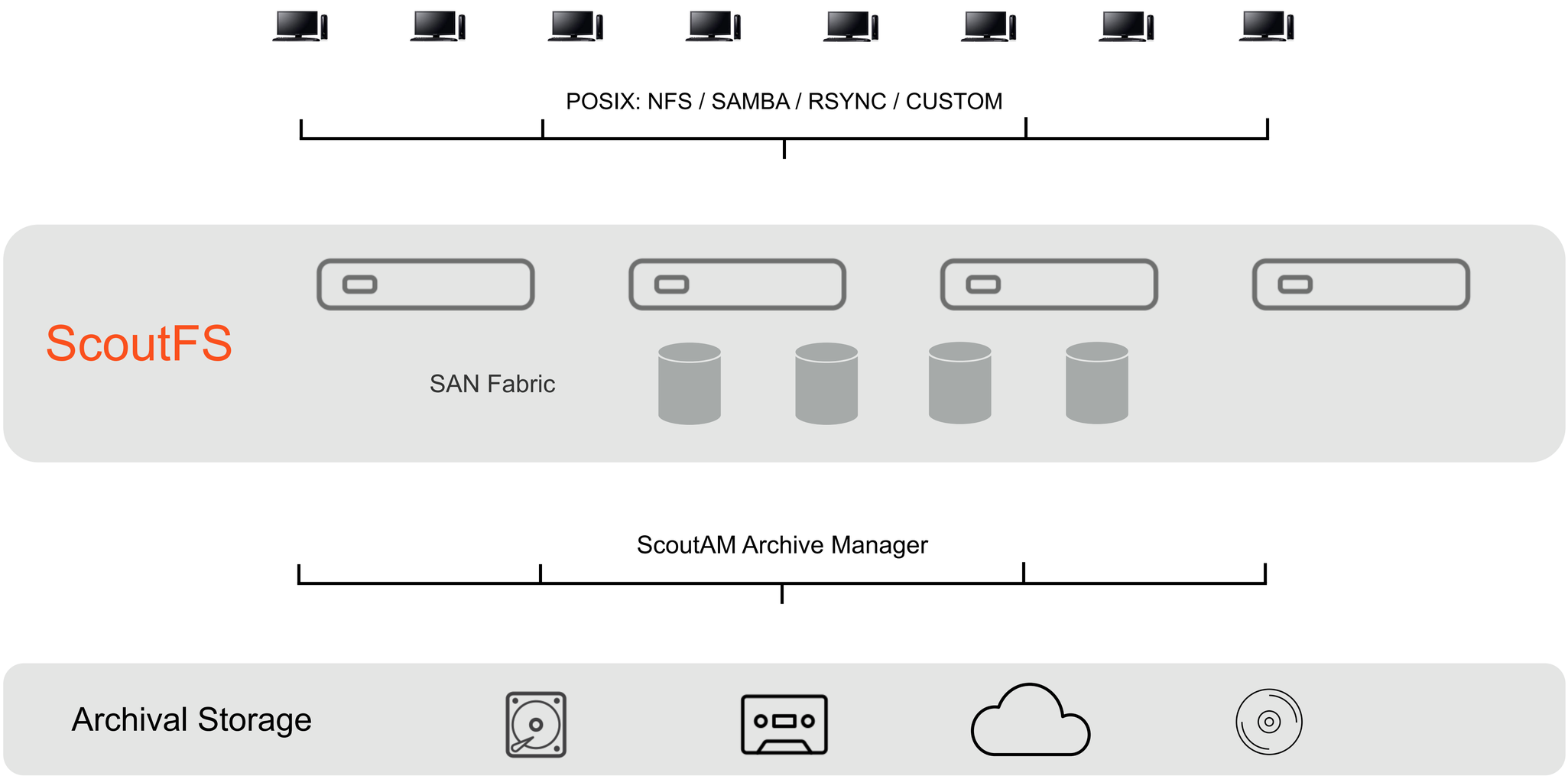
Zach Brown, Versity

MSST 2019

ARCHIVING IS OUR FOCUS.

We do one thing, and we do it well.

POSIX Archiving with ScoutFS



Archival Filesystem Differences

- Built for archive transfer rate, not for total data capacity
- Almost all metadata at rest, file data stored on archive tiers
- File count no longer constrained by data capacity
- Support both user-facing file transfer and internal tier management load
- Files have metadata which describes locations in archives
- Archive tier management software needs to search through files
- Strong desire for open source implementation

Archival Filesystem Challenges

- Exhaustive file searches are in the critical path
  - Are there new or modified files that need to be archived?
  - Which files were on that archive media that just caught fire?
  - Which large archived files were least recently used and can be released?
  - (.. and users would love efficient searching of their files!)

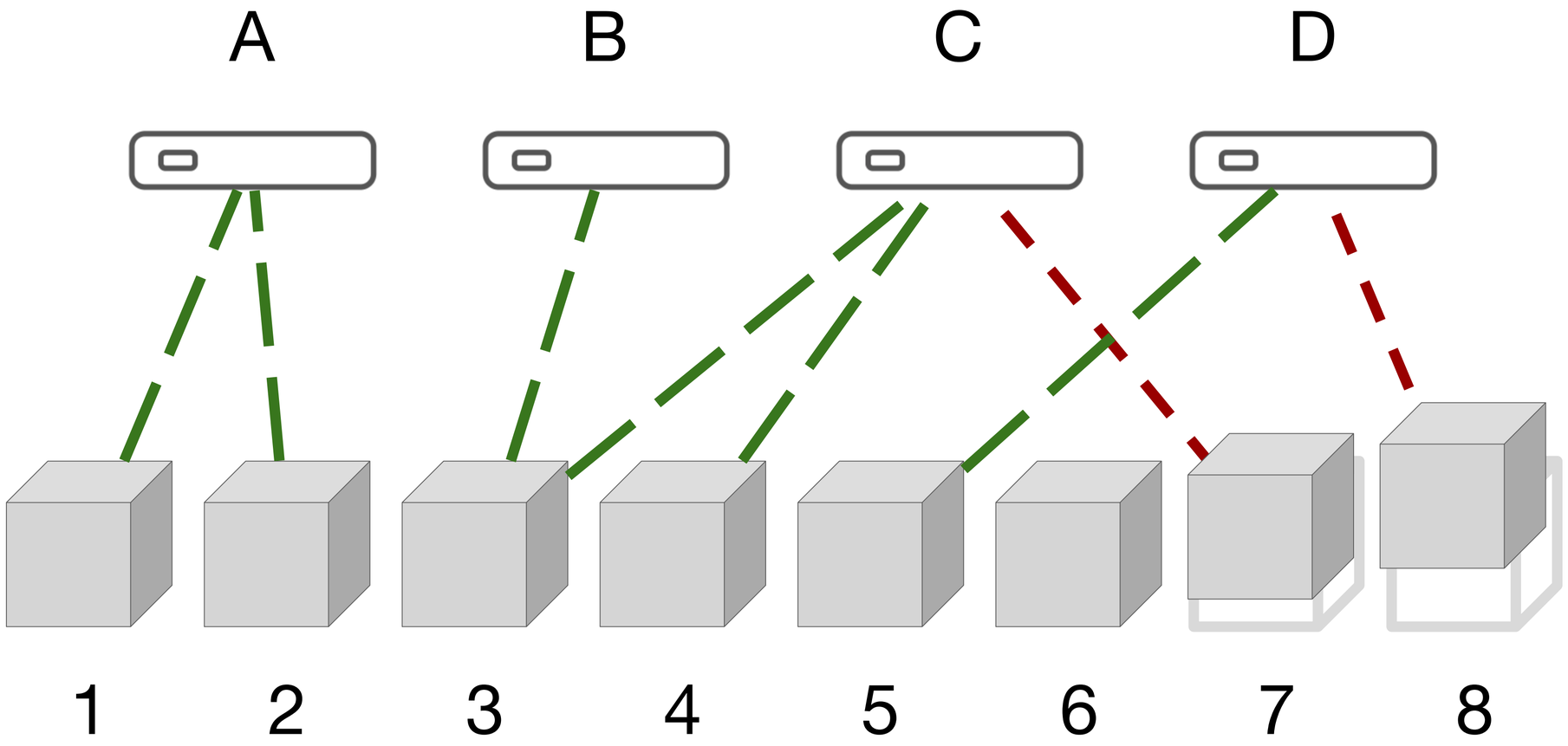
- Must saturate streaming archive tier throughput
  - Efficient large file IO
  - Small files need high metadata rates to produce sufficient archive data

ScoutFS Design Highlights

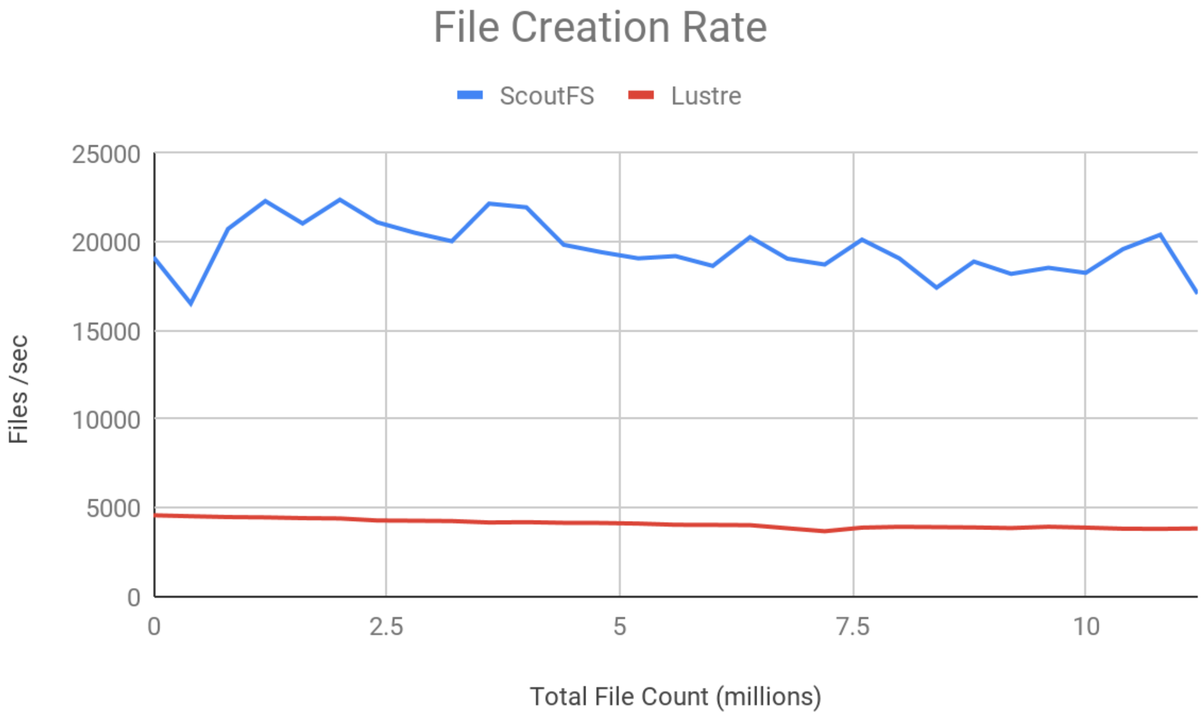
Versity

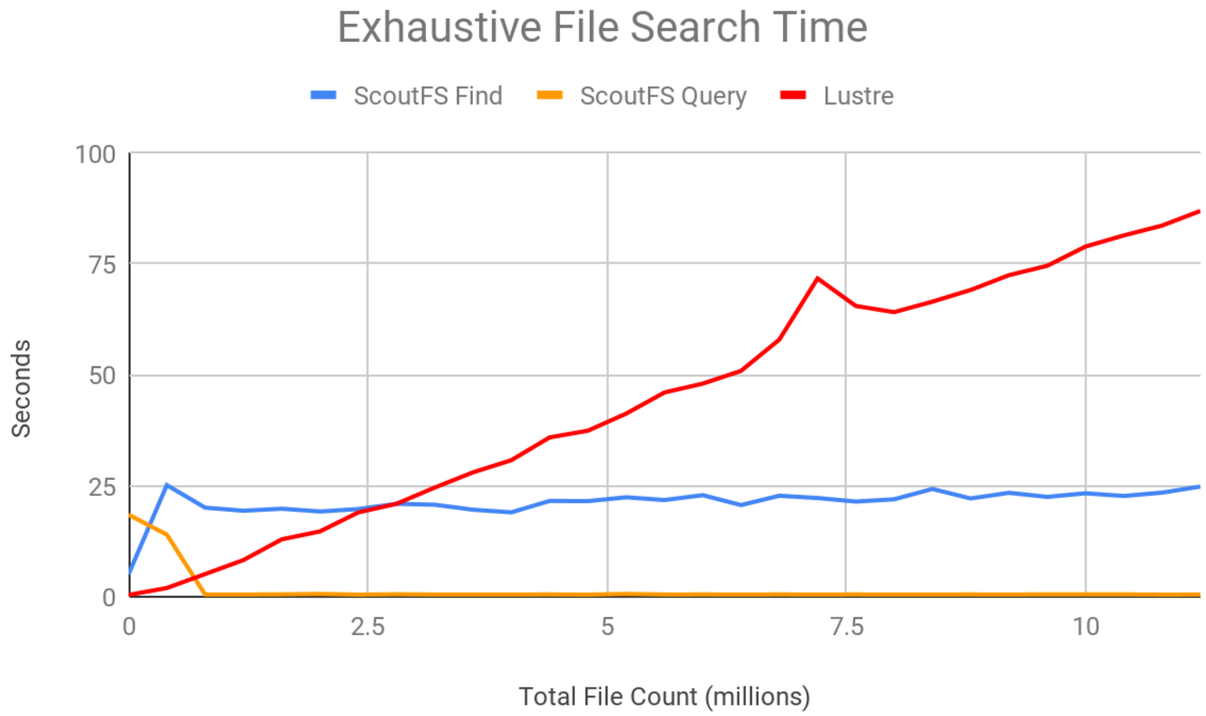
- Start with an in-kernel coherent key/value item store:
  - “Logical” locking protects item consistency and governs caching
  - Log-based “physical” storage allows concurrent item reads and writes
  - Item writes grouped into atomic log fragment writes
  - Fundamental unit of metadata IO is large log fragments
- Build a robust POSIX filesystem out of these items:
  - Full POSIX semantics, data extents, atomic metadata transactions
- Maintain persistent file index items along with FS metadata items:
  - Sort files by metadata: size, mtime, xattrs, etc
  - Index items modified in the same transaction as primary metadata items
  - Concurrent write lock mode avoids global serialization

Concurrent Metadata Reads and Writes



4 Nodes All Search While Creating





Versity

Thank You

zab@versity.com

@versitysoftware