ScoutFS Briefing

September 2018

Versity Software

the future of archiving

ARCHIVING IS OUR FOCUS.

We do one thing, and we do it well.

What we do

Rock solid data protection

For

Large data collections

At

Low cost

The long term solution for archival data storage

Versity

ScoutFS

Scale out filesystem (ScoutFS)

Shared-block filesystem

Designed for large namespace

Collapsing the database paradigm

Built for bandwidth

Indexes by **users** for **users**

Handles small files well

ScoutFS

POSIX scale out shared
block file system

Horizontally Scalable on
commodity hardware

No central MDS, no scanning;
metadata processed on all nodes

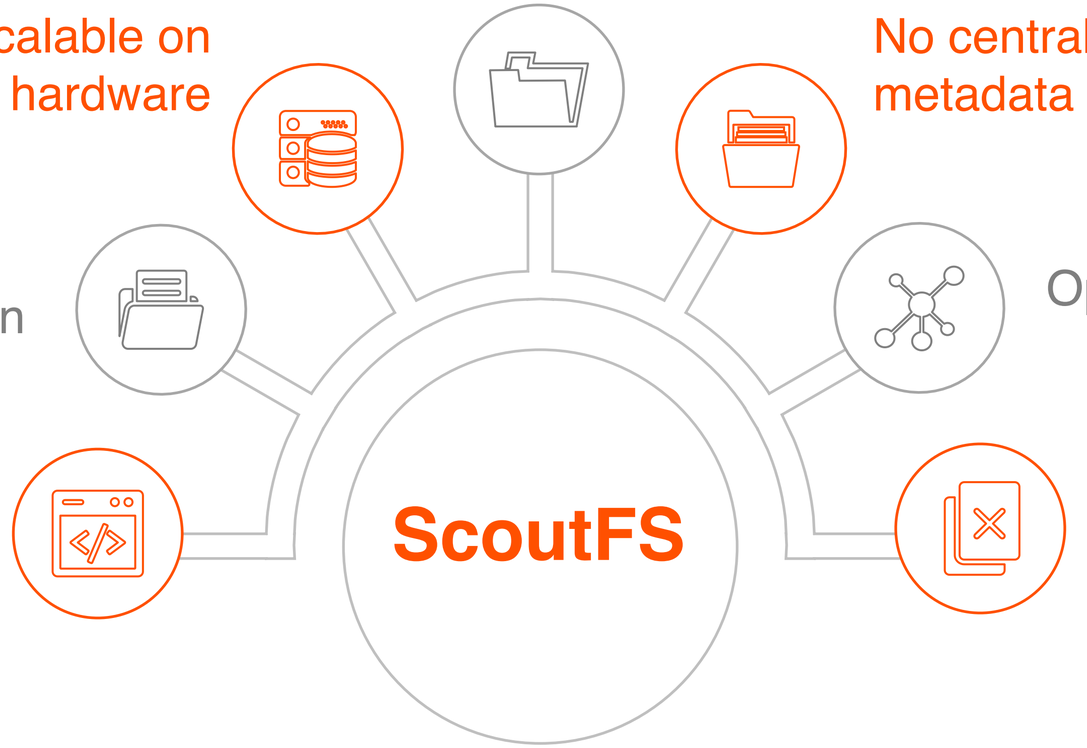
Posix compliant but
minimizes synchronization

Open Source

Data integrity +

ScoutFS

Highly available



ScoutFS

Versity

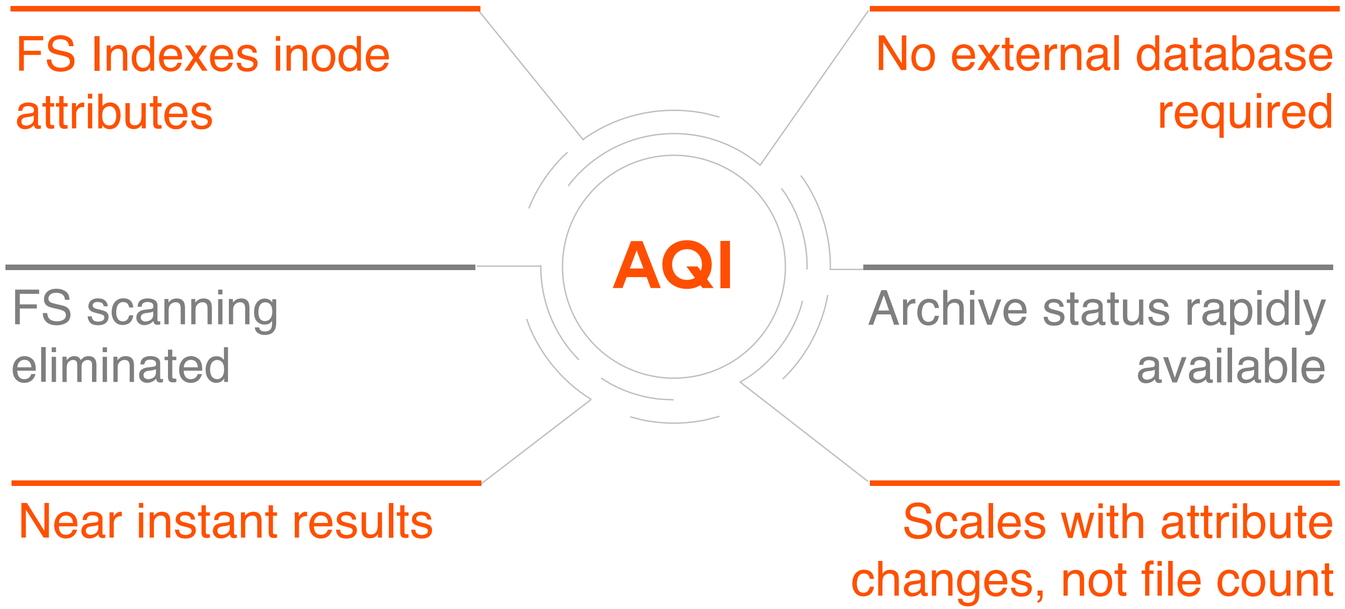
- Single purpose built for **archiving fast** at scale
  - Clustered across multiple hosts to allow for **horizontally scalable** system throughput on commodity cluster hardware
  - Work is rarely shared
    - Built for **minimum synchronization** between hosts
  - Performance is comparable to a group of standalone filesystems
  - Metadata and data are stored separately
    - Results in **efficient 'ls -l'** type operations

VSM Indexed Metadata

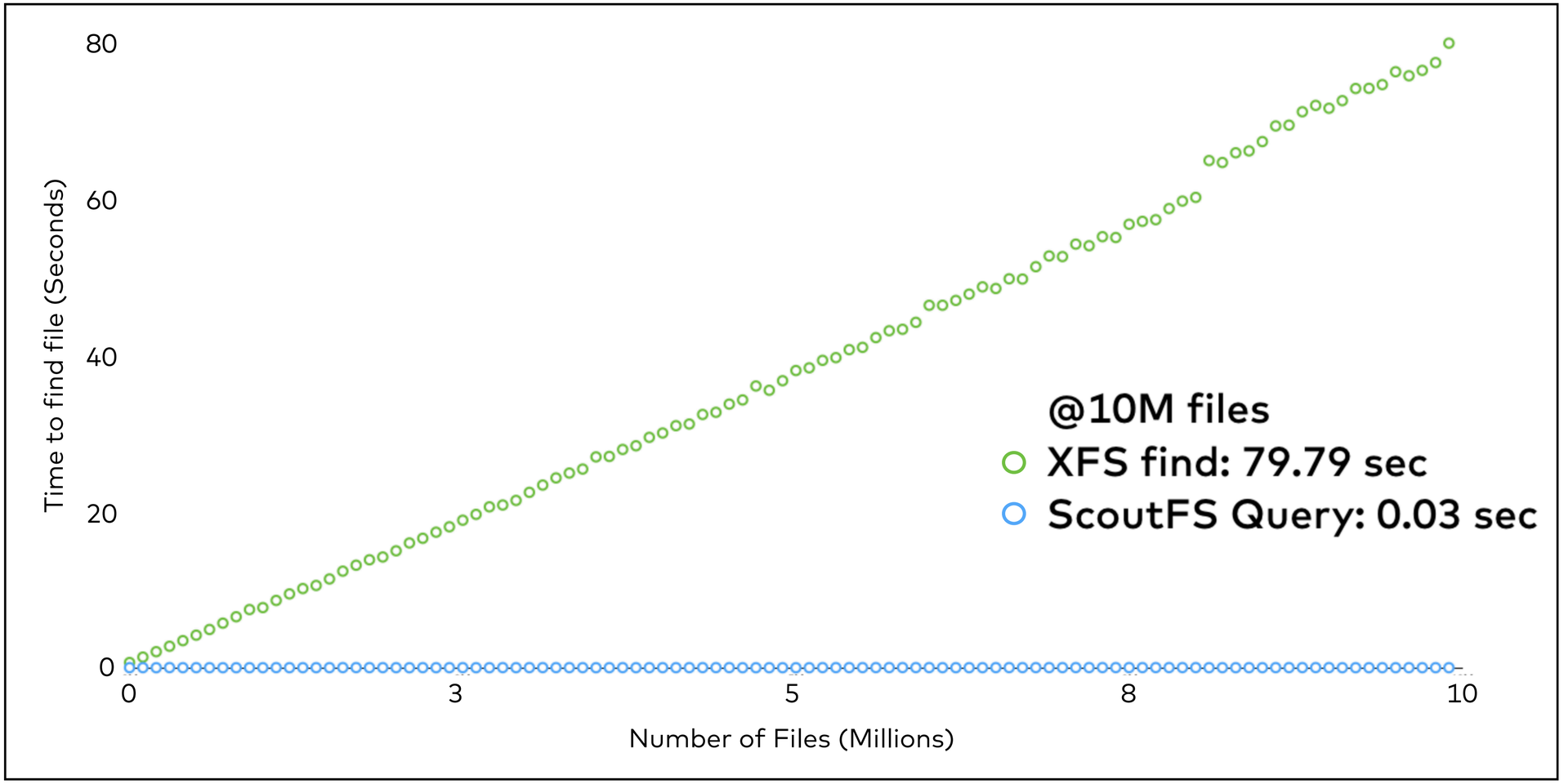
- Metadata Sequence Number (when inode attributes change)
- Data Sequence Number (when file contents change)
- Specific Extended Attributes

ScoutFS Scalability

The Accelerated Query Interface (AQI)



AQI Performance



Eliminates the file scanning bottleneck

ScoutFS: Metadata Handling

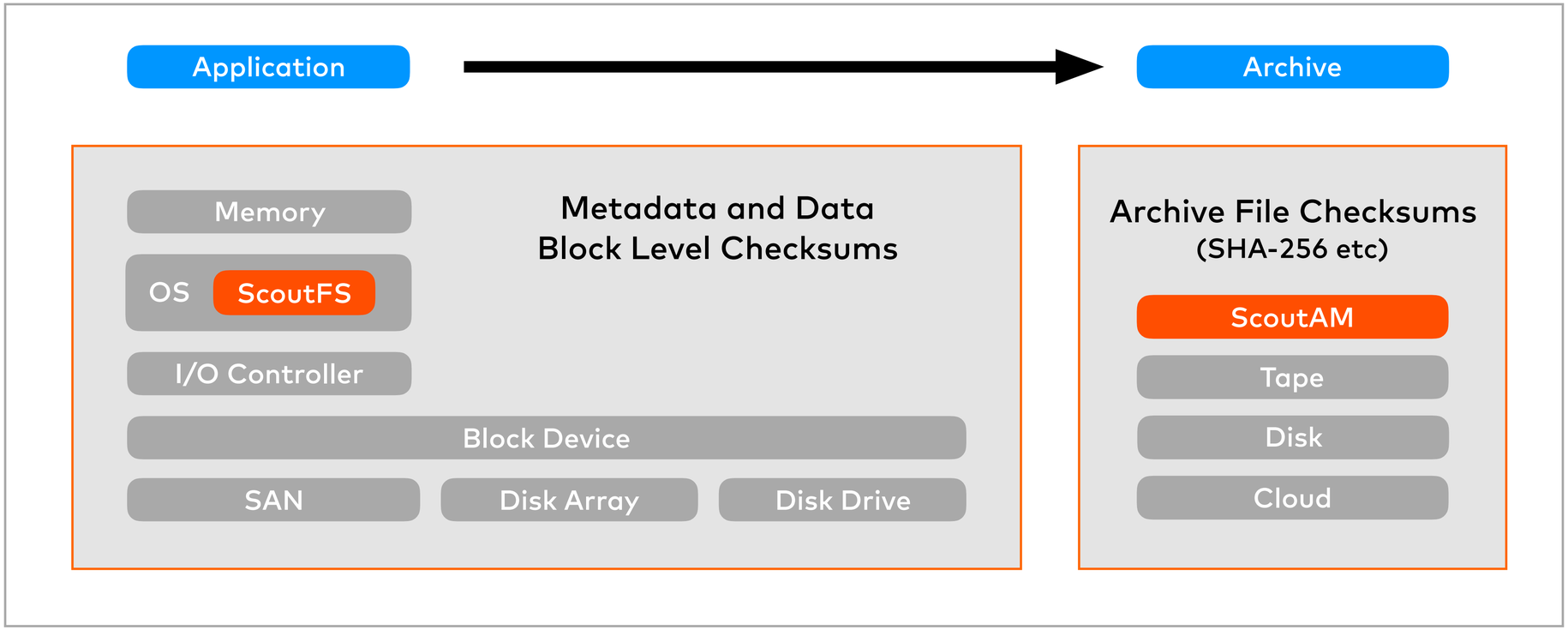
- Modified Log-Structured Merge (LSM) Tree
  - Not limited by IOPS
- Index stored in the metadata
  - Maintained atomically
- A single IO transaction with multiple operations is used to update the metadata structure
- Inodes organized so the archiver can implement policy quickly
  - structures that aren't typical in other filesystems

ScoutFS: How Coherency Works

- Manifest Service
  - Maintains consistency of the LSM structure
  - Very little communication needed due to size of LSM blocks
- Clustered locking service maintains POSIX consistency
- Locking done in ranges
  - As opposed to individual items like other filesystems
  - Locks are not ‘per directory’

POSIX that scales

Data Integrity



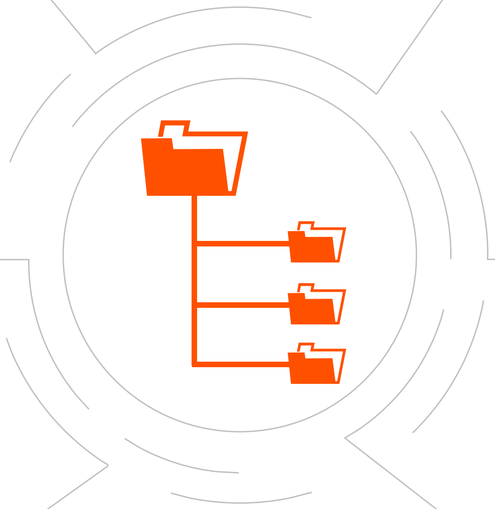
- Application all the way out to media is protected
- Storage devices not blindly trusted
- Three elements are always checked; identity, location, time

Scale out archiving application

ScoutAM

Application is aware of node resources - spreads work

Parallel reads and writes to archive storage media



Supports partial file archive and partial file stage

Parallel within one file or multiple files across multiple hosts

Supports dump/restore from VSM 1.x + other HSM's

Designed to scale as object storage systems scale

ScoutFS Takeaways

- Very efficient at what it does (scale out filesystem for archiving)
- Scales POSIX by changing the way things have traditionally been done (minimizing sync points, big messages)
- Cluster nodes can come and go due to shared block design
- Indexes are designed so the archiver can implement policy quickly

Architectural Principles

- No bottleneck centralized MetaDataServer (MDS is a role)
- No DMAPI
  - Resynchronizing and scanning cause scaling and reliability limitations
- Highly available, add or remove nodes as needed (not failover)
- Fast queries
- Bandwidth over latency
- Enables commodity hardware
  - That can change however often you want
  - Enables cost effective combinations
- No scanning
- Minimized lock interactions
- Simple implementation
  - Modularized, no ‘spaghetti’ code
  - Robust test infrastructure

Thank You

info@versity.com

@versitysoftware