Facebook’s Tectonic Filesystem: Efficiency from Exascale

Satadru Pan\textsuperscript{1}, Theano Stavrinos\textsuperscript{1,2}, Yunqiao Zhang\textsuperscript{1}, Atul Sikaria\textsuperscript{1}, Pavel Zakharov\textsuperscript{1}, Abhinav Sharma\textsuperscript{1}, Shiva Shankar P\textsuperscript{1}, Mike Shuey\textsuperscript{1}, Richard Wareing\textsuperscript{1}, Monika Gangapuram\textsuperscript{1}, Guanglei Cao\textsuperscript{1}, Christian Preseau\textsuperscript{1}, Pratap Singh\textsuperscript{1}, Kestutis Patiejunas\textsuperscript{1}, JR Tipton\textsuperscript{1}, Ethan Katz-Bassett\textsuperscript{3}, and Wyatt Lloyd\textsuperscript{2}

\textsuperscript{1}Facebook, Inc., \textsuperscript{2}Princeton University, \textsuperscript{3}Columbia University
Exabyte-Scale Storage Use Cases at FB

**Blob storage**
- Photos and videos in Facebook, Messenger attachments
- Exabytes of data
- Several KBs to several MBs in size
- Latency sensitive

**Data warehouse**
- Hive tables for data analytics, machine learning
- Exabytes of data
- Reads are order of multiple MBs, writes are 10s of MBs
- Throughput sensitive
Storage Infrastructure Before Tectonic

**Operational complexity:**
3 different systems tailored to different workloads

**Existing solutions not generic enough**

- **Blob storage**
  - Hot blobs, not storage efficient
  - Warm blobs, no support for uploads

- **Data warehouse**
  - Throughput-efficient, not suitable for small IO
  - Each instance not scalable beyond 10s of PBs

- **Haystack**
  - IOPS: Storage
  - IO-bound: wasted IO
  - Storage-bound: wasted storage

- **f4**
  - IOPS: Storage
  - IO-bound: wasted IO
  - Storage-bound: wasted storage

**Poor resource utilization:** isolated systems could not share resources
Tectonic Overview

Simpler Operations:
Single system to reason about, generic enough to handle all use cases

Better utilization: no stranding of resources

Scalability: support exabyte-scale clusters
Multitenancy: isolate tenants and share resources
Performance: match performance of specialized systems
Scalability: Support Exabyte Scale Clusters

Metadata Store: linearly scalable metadata storage

Block layer

File layer

Name layer

Key-value store

Background Services

Garbage collectors, Rebalancer, ...

Chunk Store: linearly scalable data storage

/ → [dir1, dir2]

dir1 → [file1, file2]

file1 → [block1, block2]

Block1 → [c1, ..., c14]
Scalability: Support Exabyte Scale Clusters

1. FilePath to FileID
2. FileID to list of Blocks
3. Block to list of Chunks
4. Fetch data from chunks

Metadata Store

- Name layer
- File layer
- Block layer

Key-value store

Chunk Store

Client Library

Read FilePath

Background Services

Garbage collectors, Rebalancer, ...

...
Scalability: Support Exabyte Scale Clusters

1. List of suitable nodes
2. Store Chunks to storage nodes
3. Store Block to Chunk map
4. Add Block to File

Metadata Store

Name layer
File layer
Block layer

Key-value store

Chunk Store

Background Services
Garbage collectors, Rebalancer, ...
...
...

Client Library
Performance: Match Specialized Systems

• Specialized storage systems optimize for the specific access pattern and performance requirements

• Tectonic uses *tenant-specific optimizations* to match the performance of specialized systems

• Optimizations are enabled by the Client Library, which runs in application binary

• Client library allows flexible and varying composition of Tectonic operations, which can be configured according to the needs of the tenant
Tenant-specific Optimizations: Appends

Data warehouse

- Append data to in-memory buffer
- Block full? (72MB)
- RS-encode block and store chunks
- Read-after-write consistency only after file close
- Minimize bytes written to store file to improve overall throughput

Blob storage

- Blob sizes are small (100s of KBs)
- Blobs appended to log structured file
- Blobs need to be persisted before acknowledging upload
- Minimize latency for blob uploads, Later optimize storage

Files are large (100s of MBs)
Files are read after the creator closes the file

Read block back

Append blob data to replicas of partial block on storage nodes

Block full? (80MB)

RS(9,6)  
RS(10,4)
Results

• Tectonic clusters are ~10x the size of HDFS clusters, which simplifies production operations

• Blob storage latency in Tectonic comparable to Haystack

• In a multitenant cluster, data warehouse uses surplus IO from blob storage to serve its peaks
Efficiency From Storage Consolidation

Blob storage: surplus IO available
Efficiency From Storage Consolidation

Data warehouse: peaks need excess IO, use surplus from blob storage

Blob storage: surplus IO available
Tectonic Provides Datacenter-Scale Storage

- Replaced previous constellation of specialized storage systems
  - Simpler operations
  - Better resource utilization

- Tectonic's design addresses the key challenges:
  - Scalability: disaggregated linearly scalable components
  - Performance: tenant-specific optimizations via client library
  - ...

13
Thank You