Scaling File System Metadata Throughput using IndexFS

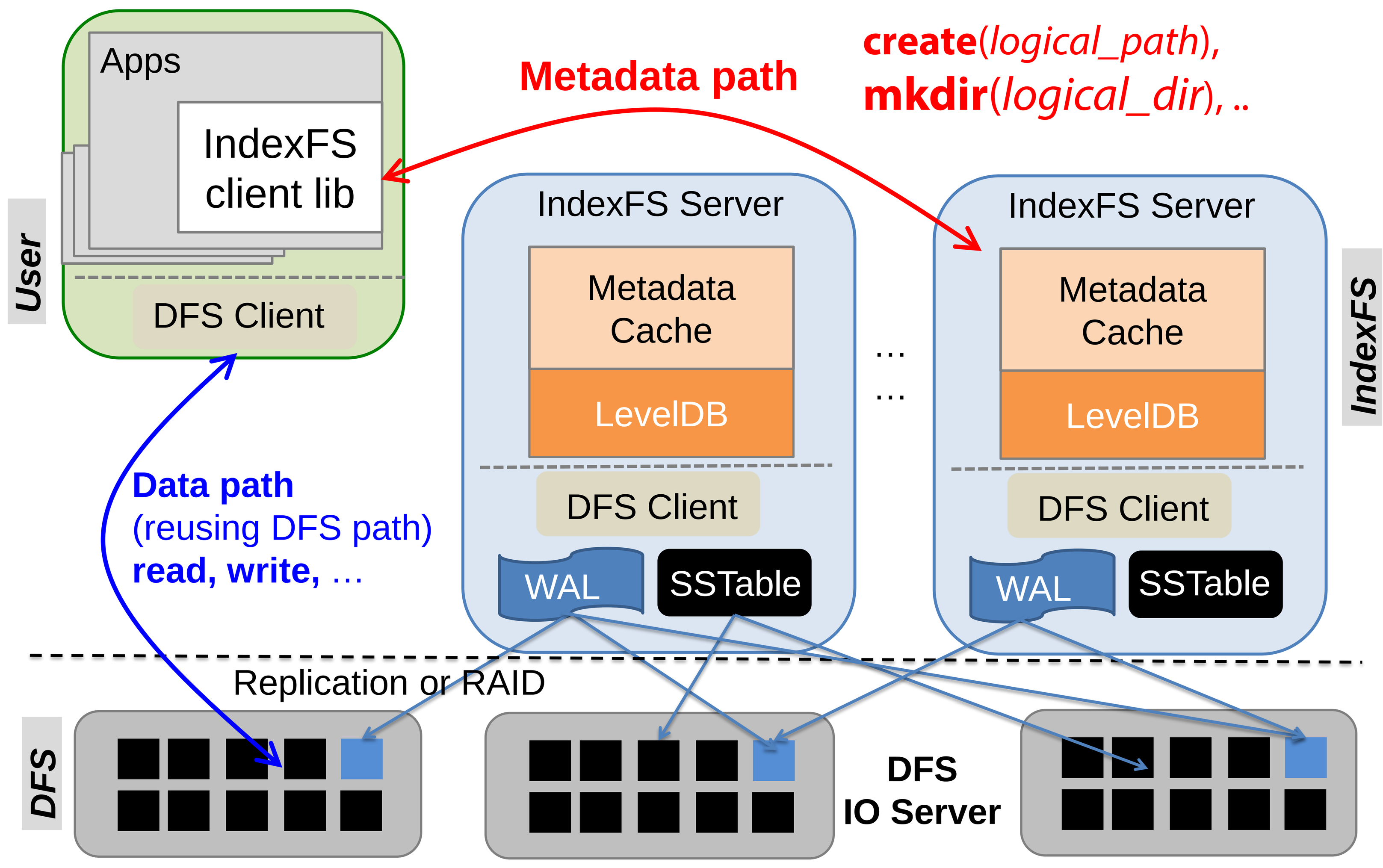
Kai Ren, Qing Zheng, Swapnil Patil, Garth Gibson

Overview

Problem: Existing distributed file systems do not provide a scalable global shared namespace. Accesses to *metadata* and *small files* are a performance bottleneck.

IndexFS: Middleware layered on existing distributed FSs

- Represent metadata in log-structured merge tree for speed
- Delay object store creates until file is non-trivial in size
- Bypass metadata server for large data access



LSM Tree and Column-Style Storage

Schema: cluster directory entries on disk

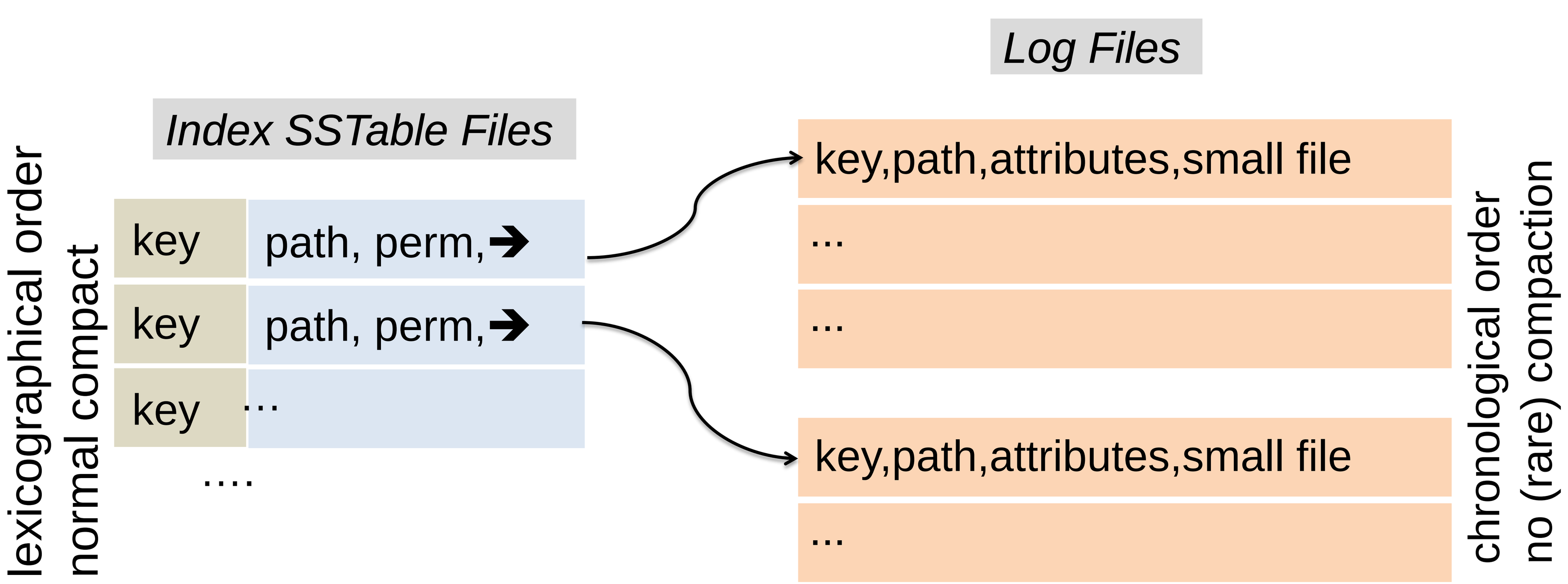
- key: <parent directory ID, partition ID, hash(file name)>
- value: file attributes, small files, pointer

Column-style Storage Format: for fast insertion

- Metadata / files append to non-LevelDB log files
- LevelDB only stores pointers to metadata and small files
- Delay data sorting and space cleaning for most metadata

Bulk-insertion: even faster insertion

- Use client-side write-back cache to build SSTable locally
- No RPC overhead per operation
- Assume clients only insert new tree, no conflict operations between clients



Conclusions

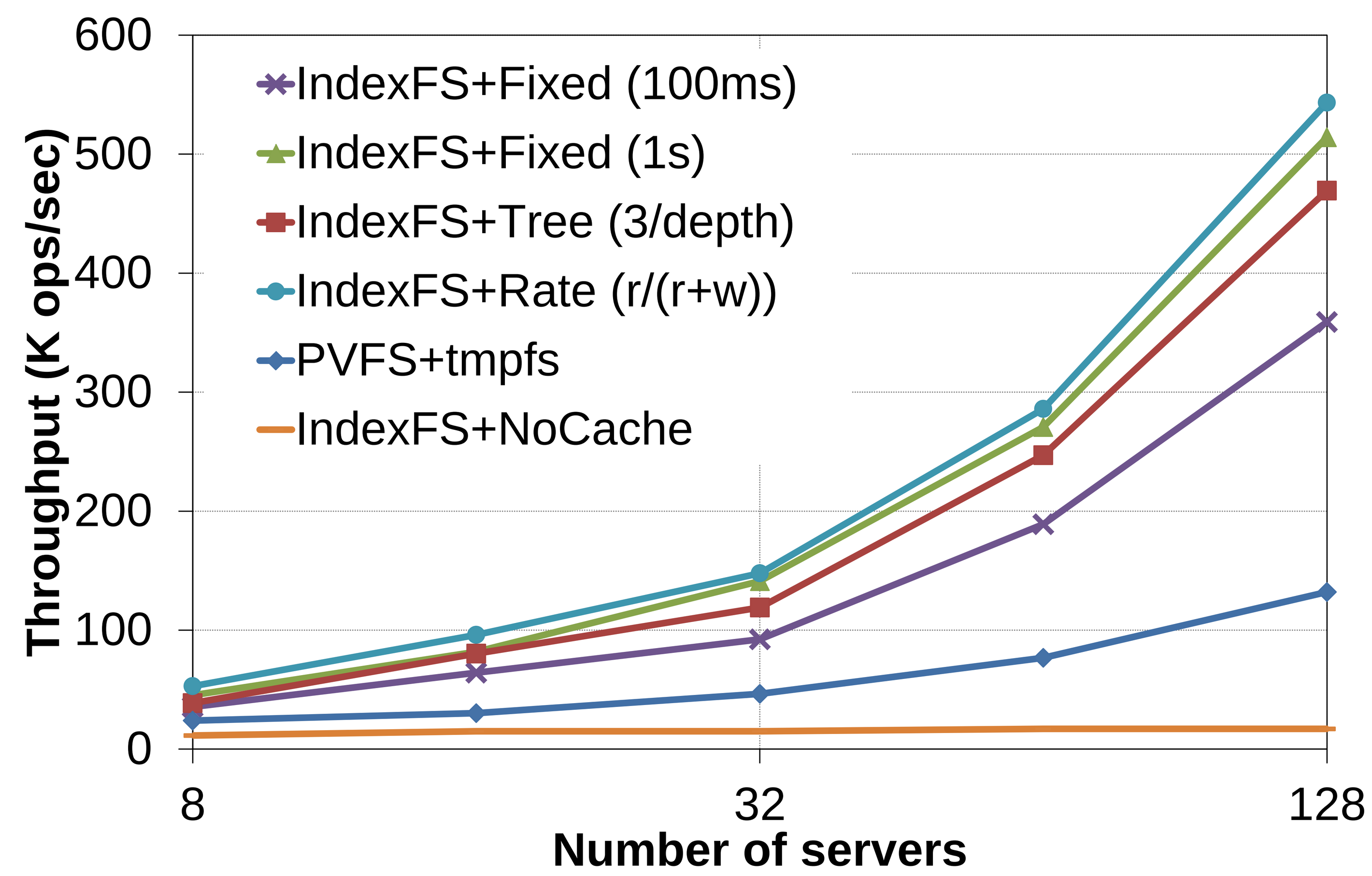
- Sustains high metadata throughput for many servers by using log-structured storage format and storm-free client caching
- Portable (e.g. Lustre, PVFS, HDFS and PanFS so far)

Namespace Distribution

- Newly created directory is randomly assigned to a server
- Binary splitting a directory partition using GIGA+ [FAST11]
- Want client caching of directory entries to mitigate hotspots
  - Don't want storms of cache invalidation callbacks
  - Use leases with only expiration deadlines per directory
  - Affect only rmdir, rename and chmod directory
  - Lease duration: fixed duration (100ms / 1s) or depth based (3sec/depth) or rate based (r/(r+w)) sec)

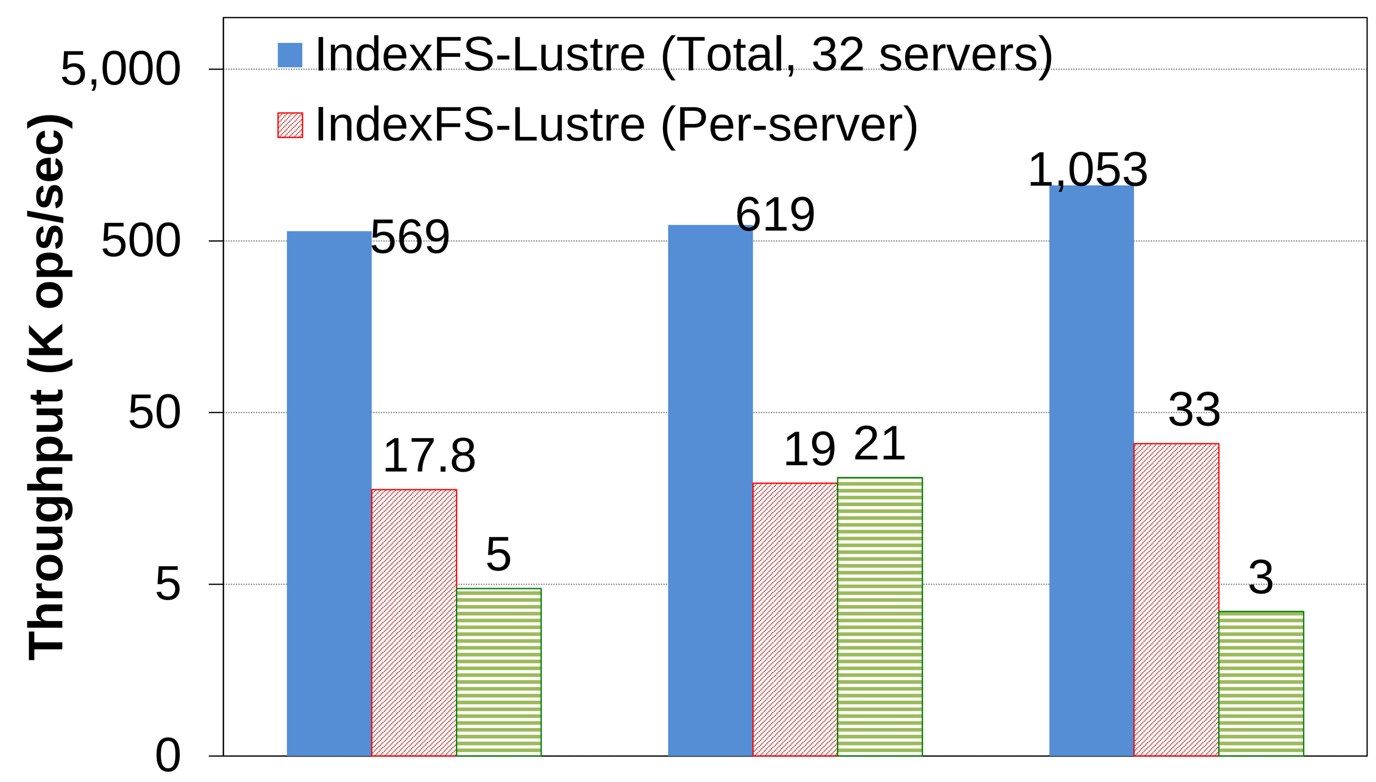
Scalability Test on PVFS: Replay Linked-In one-day HDFS traces

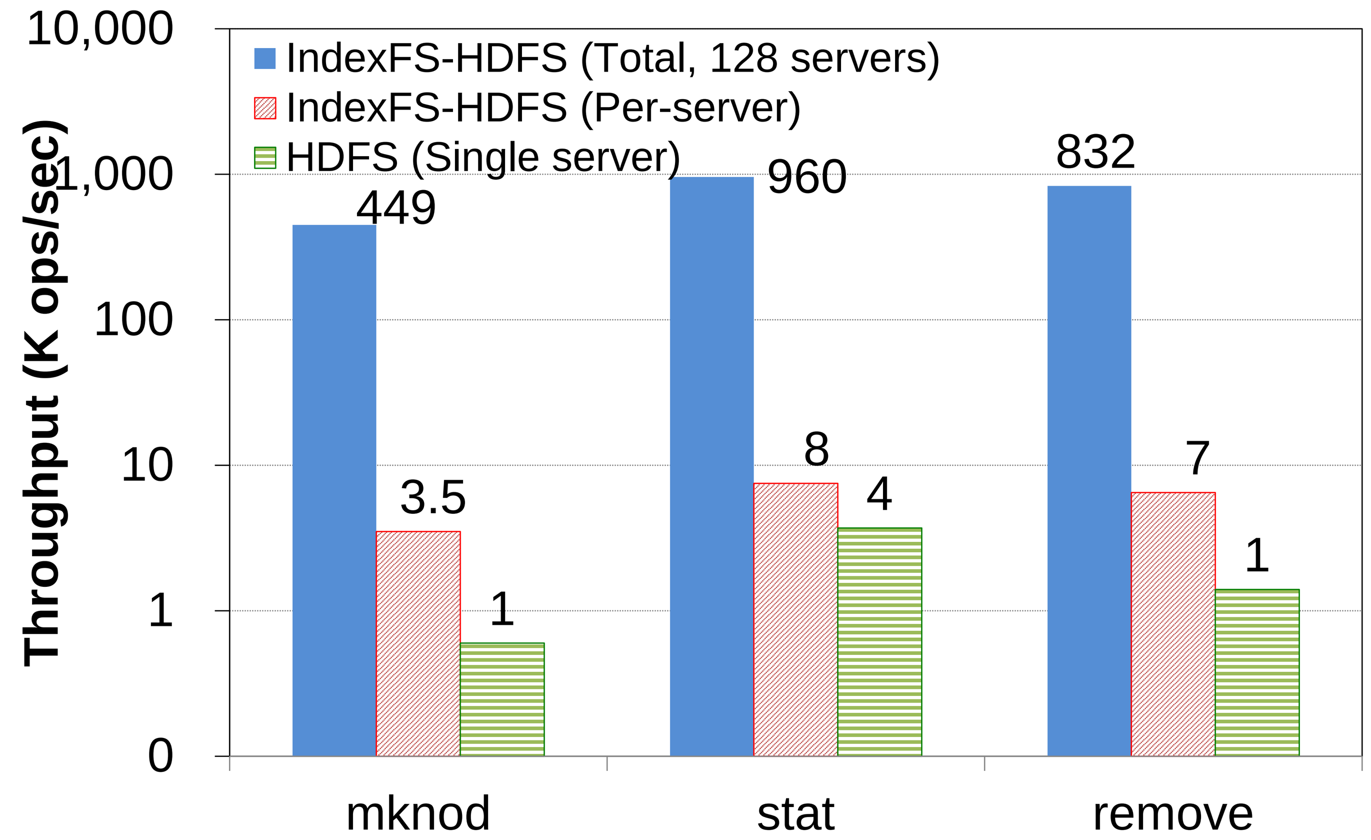
- Scale the number of server/client machines from 4 to 128
- Each machine has dual core, 8GB memory and 1GE NIC
- PVFS uses tmpfs as disk to counter use of BDB transactions



Mdtest on Lustre & HDFS: Three-phases HPC benchmark

- Create / Stat / Delete 32 million files in a shared directory
- Lustre on a 32-node LANL cluster, HDFS on a 128-node cluster





Intel Science & Technology Center for Cloud Computing

Carnegie Mellon University · Georgia Tech · intel · Princeton University · UC Berkeley · University of Washington