STORAGE

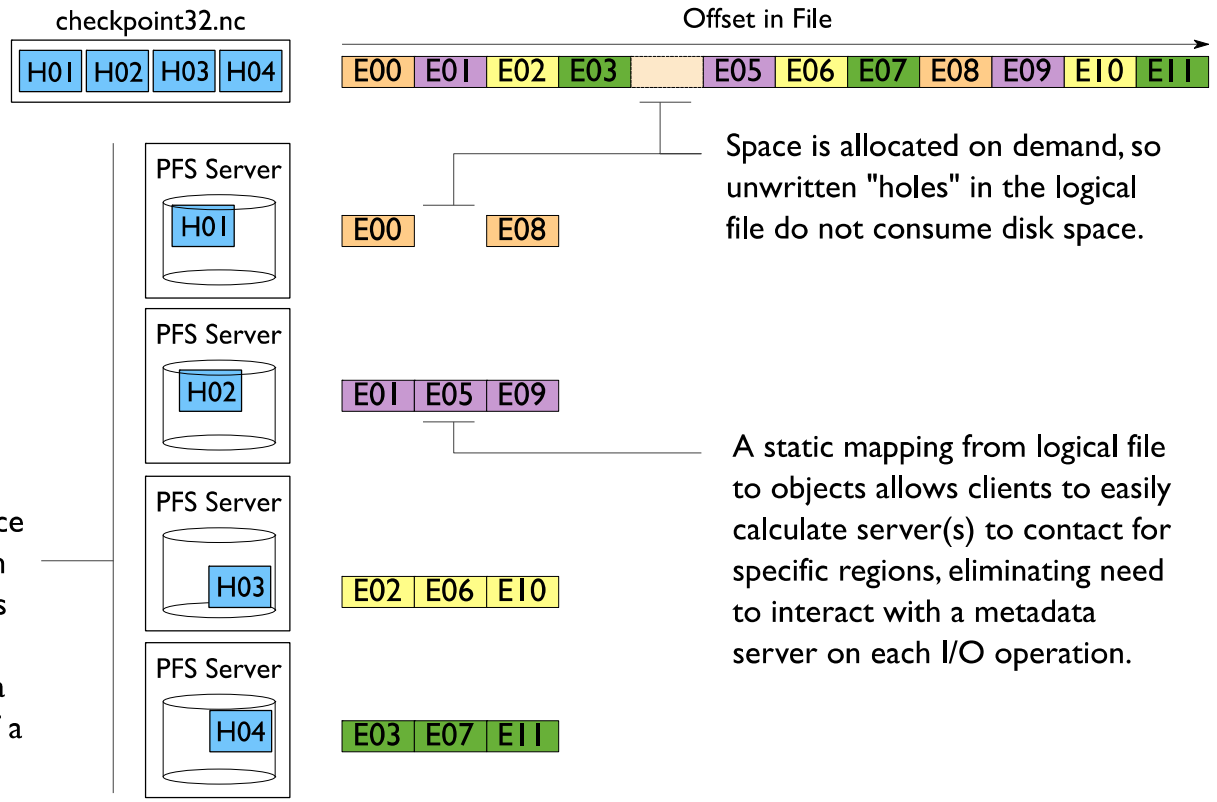
- File systems, blob storage, databases, keyval stores
- Only have a half day, so let's focus on parallel file systems
  - Specifically, Lustre, but you might encounter GPFS or others
- Software abstractions:
  - "performance portability"
  - almost there but not quite

DATA DISTRIBUTION IN PARALLEL FILE SYSTEMS

Logically a file is an extendable sequence of bytes that can be referenced by offset into the sequence.

Metadata associated with the file specifies a mapping of this sequence of bytes into a set of objects on PFS servers.

Extents in the byte sequence are mapped into objects on PFS servers. This mapping is usually determined at file creation time and is often a round-robin distribution of a fixed extent size over the allocated objects.



Space is allocated on demand, so unwritten "holes" in the logical file do not consume disk space.

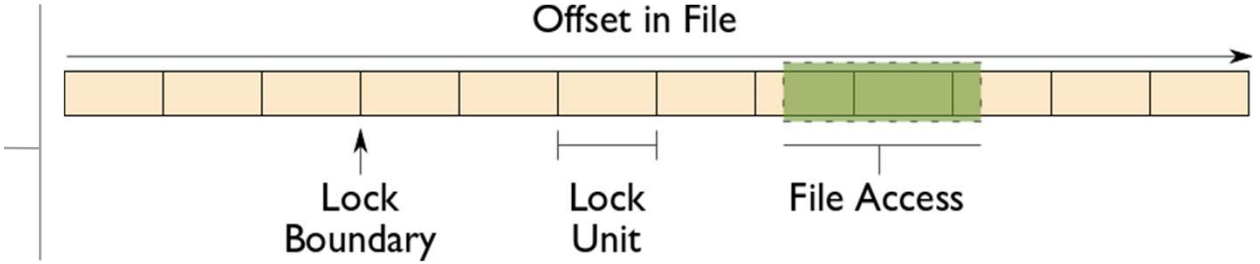
A static mapping from logical file to objects allows clients to easily calculate server(s) to contact for specific regions, eliminating need to interact with a metadata server on each I/O operation.

LOCKING IN PARALLEL FILE SYSTEMS

Most parallel file systems use **locks** to manage concurrent access to files

- Files are divided into lock units aligned with blocks or stripes
- Clients obtain locks on units that they will access before I/O occurs
- Enables caching on clients as well (as long as client has a lock, it knows its cached data is valid)
- Locks are reclaimed from clients when others desire access
- These locks occur behind the scenes: different from locks an application might call explicitly

If an access touches any data in a lock unit, the lock for that region must be obtained before access occurs.



FILESYSTEMS: LUSTRE

https://www.lustre.org/

- Metadata servers (MDT)
  - E.g File creation
- Storage servers (OST)
  - Data lives here
  - So does all the parallelism
- Clients
  - POSIX interface (`open, write, read, close`)