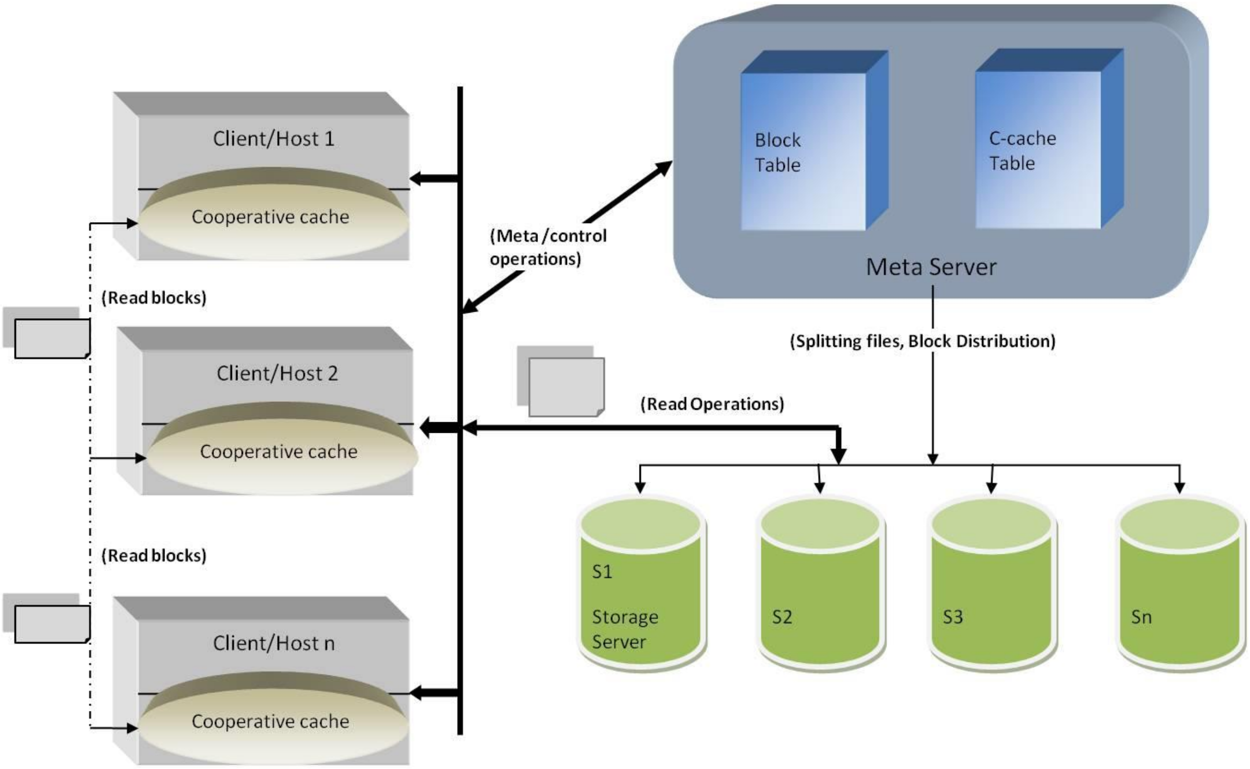
Distributed High Performance File System with Client Cooperative Cache

DHPFS Architecture



The DHPFS with client cooperative cache consists of meta-server which acts as master/control server for the entire architecture, this meta-server is responsible for dividing large files into small data blocks and distributing across different storage servers, it also maintains all block locations of storage servers and client cooperative caching. Storage servers are responsible for data management, which stores actual data blocks that can accessed or read by different clients. Each client as its own cooperative cache that can be accessed by other clients, this cooperative cache stores previously accessed data blocks in memory, thus multiple client request can be served immediately when they are trying to access same data blocks.

Initially client sends a block request for meta-server, meta-server will have information including data blocks location of storage server on block table (B-table) and client cooperative location on cooperative cache (C-cache) table. Meta-server provides data block location/handle either from block or C-cache table, based on which clients try to access data blocks from storage server disks or from client cooperative cache. This would eliminate bottleneck on storage servers when large number of clients tries to read same data blocks and would partially decrease workload.

Meta Server Communication

§ Meta server acts as master/control server in overall DHPFS architecture, it divides huge files into small blocks which are 3 – 25 MB in size.

§ Data blocks are distributed across different storage servers in round robin fashion.§ It consists of two tables including:

o Block table (B-table) maintains all data blocks locations which are present in storage server,o Cooperative cache (C-cache) table stores all clients with C-cache location.

§ Based on client request, meta-server sends block location/handle accessing either from B-table or C-cache table.§ Meta server receives update from different clients when blocks are evicted from client cooperative cache using replacement algorithm and providing more space to frequently accessed blocks in cooperative cache.

Storage Server Communication

§ Storage servers are responsible for data block management, multiple data blocks are stored across different storage servers. § Based on meta-data information provided by meta-server, each client sends block handle or contacts respective storage server.
§ If the data block is present in storage server, then client start reading blocks from storage servers or disks.

Client Communication and Cooperative cache

§ Client stores previously accessed data blocks in memory with cooperative cache, each client try to read data blocks that are cached by other clients. § Client cooperative cache location will be stored in meta-server; C-cache table will be updated when blocks are excluded from cooperative cache list. § Client contacts meta-server for data block location, based on which each client try to read from storage server disk, if the blocks are not previously accessed.
§ If the participating clients trying to access blocks that are already present in client cooperative cache, then different clients can read data blocks directly from remote client memory or cooperative cache instead of reaching storage servers.
§ Client cooperative cache can eliminate bottlenecks when multiple clients trying to read same data block and significantly reduce workload on storage servers.
§ Client cooperative cache can achieve better performance since they are providing access to data blocks from remote machine memory.