



# The Lightweight File Systems Project

The Lightweight File Systems (LWFS) Project is a joint effort between Sandia National Laboratories and the University of New Mexico to address I/O scalability issues for next-generation computer systems.

## I/O scalability is a well-recognized problem

“Major improvements in scalability throughout I/O and storage are required ... significant investment is required to foster timely improvement.”

*[HECRTF, May 2004]*

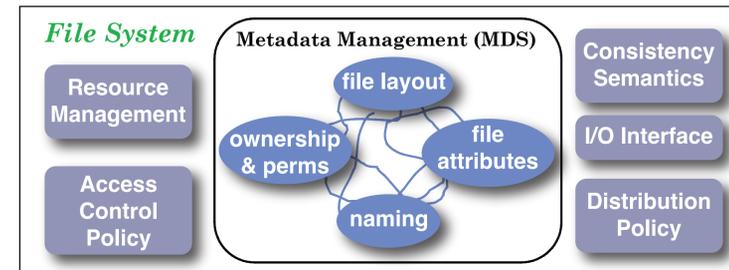
“... the performance of these [I/O] components must be improved, particularly in the area of throughput and scalability.”

*[Office of Science Data Management Challenge, May 2004]*

“Research is needed to explore appropriate methods for decomposing metadata services to enable scalability.”

*[Roadmap for Revitalization of High-End Computing, June 2003]*

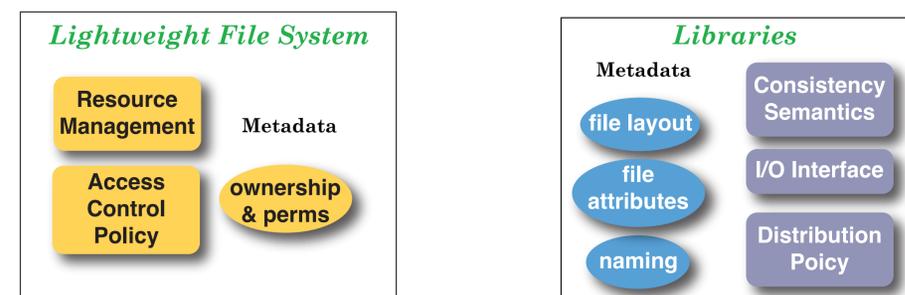
Unneeded functionality hinders application scalability



Luster, PVFS2, Panasas, and most others use this model

## Lightweight file systems provide only what is necessary

LWFS provides security and access to storage

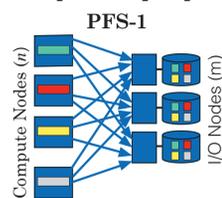


Libraries provide app-specific functionality

## Application-directed checkpoints: a motivating example

### Three different approaches

#### One striped file per process

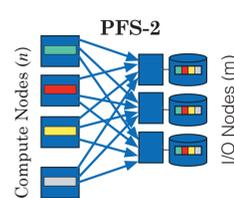


For large systems, we expect the MDS to be a significant bottleneck for file creation.

Create required  $O(nm)$  comms (all synchronized because they go through the MDS).

The most optimistic scaling of Lustre results show creation to take on the order of tens of minutes for next-generation systems.

#### One shared file



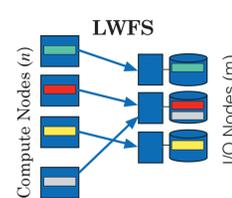
Resolves the file-creation issues from case 1.

Introduces overheads for the write.

Standard distribution schemes create lots of small requests.

Unnecessary, but required consistency semantics effectively synchronize access to I/O nodes.

#### One object per process



Compute nodes create objects in parallel (no need for a MDS involvement).

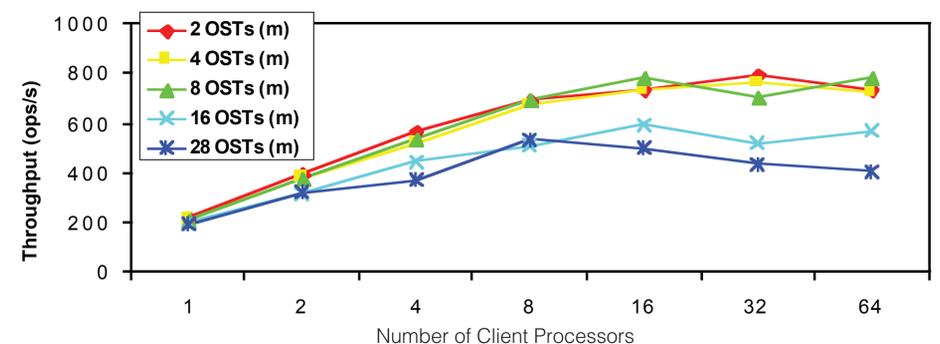
No need to synchronize access to the device (avoids locks all to gather).

Fewer communications

Minimal access with the naming service at end of operation to tie objects together.

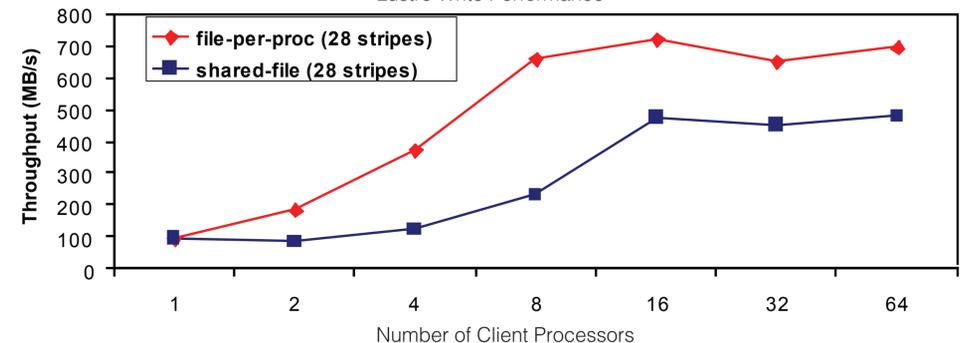
Matches the logical view of what the application wants to do.

Lustre Create Performance



File creation does not scale because all requests go through the MDS; however, some operations (e.g., checkpoint) require minimal use of a centralized naming service.

Lustre Write Performance



The poor write performance of the shared-file version is the result of imposed, but unnecessary, consistency semantics.

LWFS Results Soon!

Visualize the Difference



The University of New Mexico



NNSA National Nuclear Security Administration