Parallel Filesystems

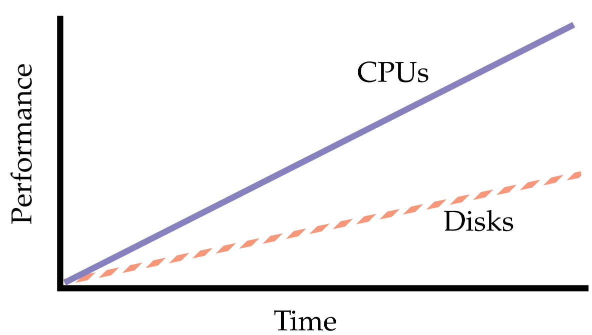
Outline

- Filesystem architecture
- File layout
- Existing filesystems
- Summary

MPPs: "It's the economy, stupid."

Commodity "off-the-shelf" parts have made MPPs and workstation clusters the ~~price/performance~~ winners.

Processor speed is improving faster than disk speed.



The price is right; the performance is...

...only a matter of software.

Classes of Parallel I/O

Application I/O

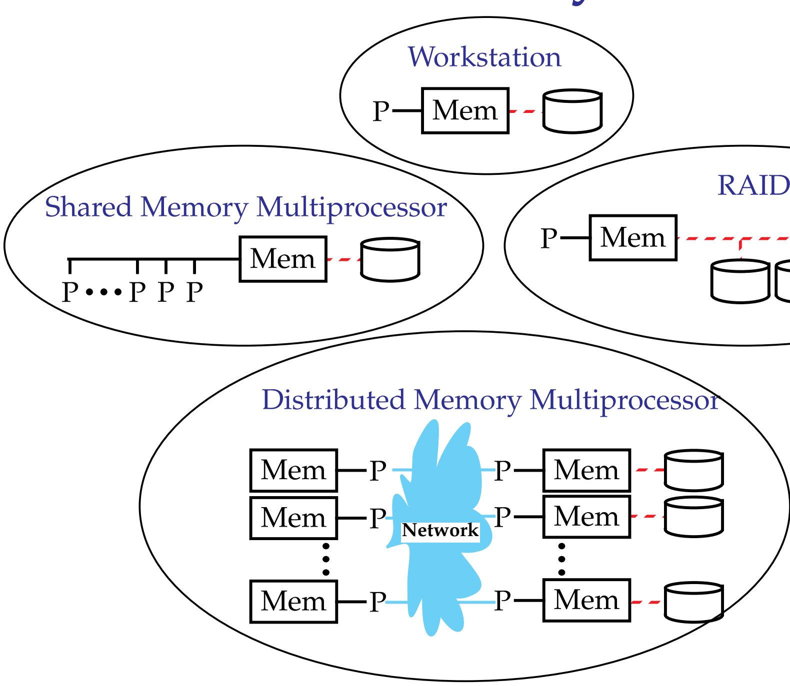
- Program Initialization — read only, once per run
- Data Analysis — read mostly “serial” files
- Visualization/Trace — write only “serial” files
- Out-of-core solvers — read/write “parallel” files
- Checkpointing — write mostly “parallel (?)” files

System I/O

- Program Loading, Paging, Checkpointing
- Data Migration

Supporting a workload...

What Makes Filesystems Hard?



Multiple Independent I/O Channels

Parallel Filesystem Architecture

Interface—how files are accessed

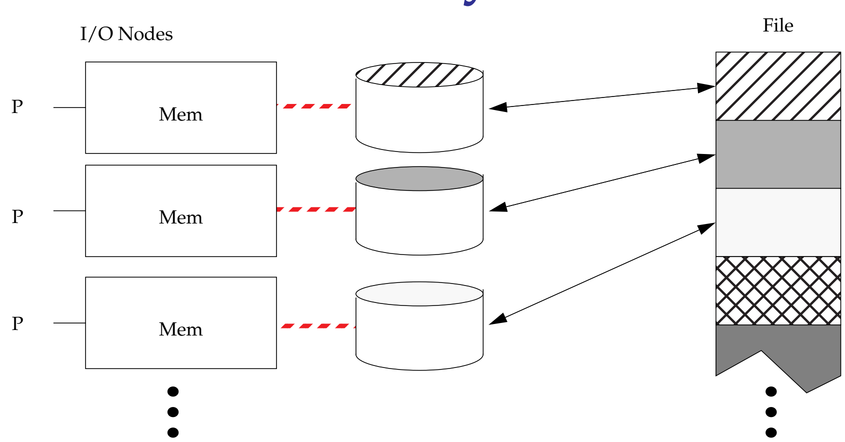
- Library versus operating system
- Semantics of parallel access

File layout—how files are partitioned among I/O nodes

Implementation details

- Reliability (RAID?)
- Interoperability (ability to easily access files)
- Caching (size, location, coherence)
- Data path between applications and files
- I/O node usage

File Layout



Every file is partitioned into file chunks

Chunks are stored among I/O nodes on commodity disks

Performance depends on accessing disks simultaneously

File Layout

Ideally, file layout should

- maximize parallelism (keep all disks active)
  —implies smaller file blocks
- use disks efficiently (maximize bandwidth)
  —implies larger file blocks
- eliminate unnecessary accesses and false sharing

All commercial systems use an HPF CYCLIC(su) file layout

- File block size varies, and may be settable
- Also called “striping”, su = striping unit

Some systems provide alternate layouts (e.g. parallel files)

Intel Paragon PFS

File layout

- HPF CYCLIC(64k) across UFS filesystems
  striping unit is set per filesystem

Interoperability

- No difference between PFS and UFS filesystems
  (except performance)

Caching Fastpath I/O bypasses caching

Data path NORMA IPC over mesh

- Pre-R1.3 NORMA IPC limited performance scaling

I/O Nodes Dedicated to filesystem (mostly)

IBM SP2 PIOFS

File layout

- HPF CYCLIC(32k) across JFS filesystems
  striping unit is set per file
- Supports Vesta file layouts (2D)

Interoperability

- PIOFS filesystems can be mounted anywhere

Caching	AIX JFS standard caching on I/O nodes
Data path	UDP/IP over Switch
I/O Nodes	Dedicated to a single filesystem (mostly)

PIOUS

Moyer & Sunderam, Emory, '94

Highly portable library — PVM based

File layout

- HPF CYCLIC(su) across UNIX filesystems
  striping unit is set per file
- Supports "parallel files"—segmented view

Interoperability

- PIOUS files are only accessible from within PIOUS

Caching	UNIX caching on I/O nodes
Data path	PVM (usually TCP/IP over Ethernet)
I/O Nodes	No separate I/O nodes

Panda

Seamons+, Illinois, '94

Library implementation

File layout

- Uses a 3D file layout for 3D data distributions
- Files are stored on underlying filesystem

Interoperability

- Panda files are only accessible from within Panda
- Complex file metadata must be stored with files

Demonstrated order-of-magnitude performance improvements on the iPSC/860 CFS

Filesystem Summary

Commercial systems

- Use HPF CYCLIC file layouts
- Dedicate I/O nodes
- OS implementation

Non-commercial systems

- Portable library implementations
- Poor interoperability

Under the right conditions (e.g. when benchmarked by designers) all parallel filesystems provide scalable performance.