GPFS
A shared-disk file system for large computing clusters

Introduction

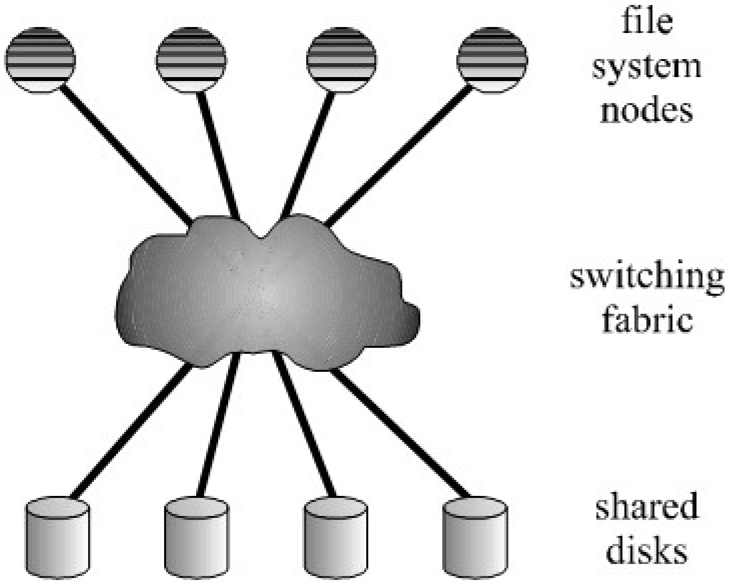
- In today’s world, the computing machines are getting more and more powerful and the problems are also getting bigger.
- So the task is to solve the problem by aggregating a number of powerful machines into clusters rather than solving on a single machine.
- But there are issues related to clusters:
  - Sharing resources
  - Performance and scalability
  - Achieving parallelism and data consistency

GPFS Overview

- GPFS (General Parallel File System) is a parallel file system for cluster computers.
- GPFS is used on six of the ten most supercomputers in the world.
- GPFS supports parallel access to both file data and file metadata.
- GPFS also performs administrative functions in parallel.

GPFS Overview

- GPFS has a shared disk architecture.
- Supports file systems upto 4096 disks , 1TB each



General Large File System Issues

- Data Striping and Allocation
  - To achieve high throughput to a single large file, data must be striped across multiple disks.
  - Throughput Vs Space Utilization
- Large Directory Support
  - Uses Extendible hashing.
- Logging and Recovery
  - Each mode has a separate log for each file system it mounts.
  - As this log can be read by all other nodes, any other node can perform recovery on behalf of this node.

Managing Parallelism and Consistency in a cluster

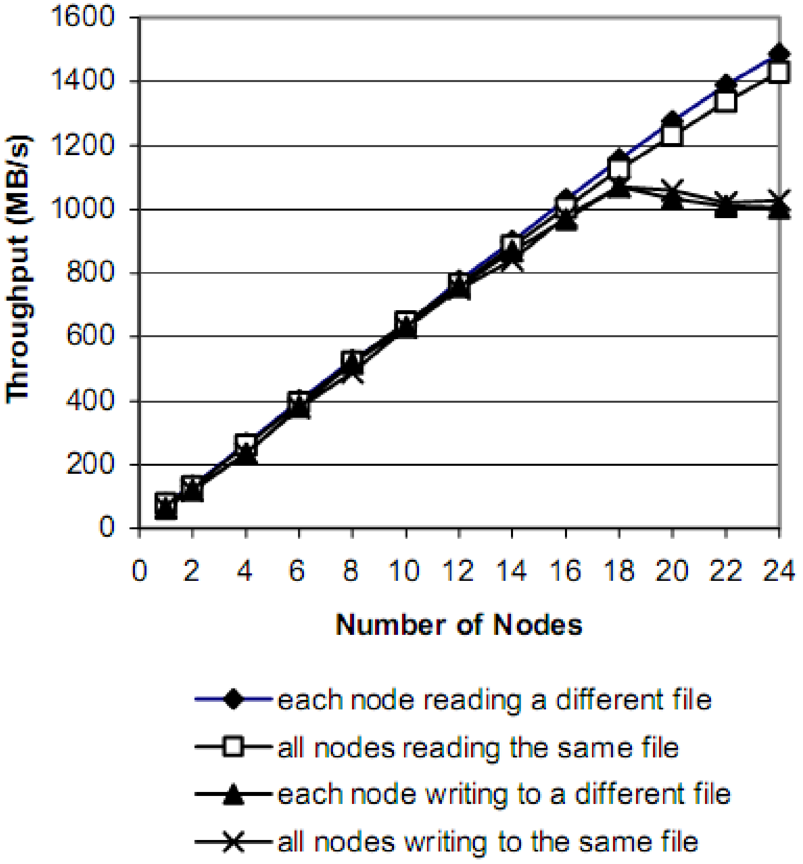
- Distributed Locking Vs Centralized Management
  - Distributed Locking : Consult all other nodes before acquiring a lock.
  - Centralized Management: All conflicting operations are forwarded to a designated node, which performs the requested read or update.
- Lock granularity
  - Too small : High overhead, due to more lock requests
  - Too large: More lock conflicts

The GPFS Distributed Lock Manager

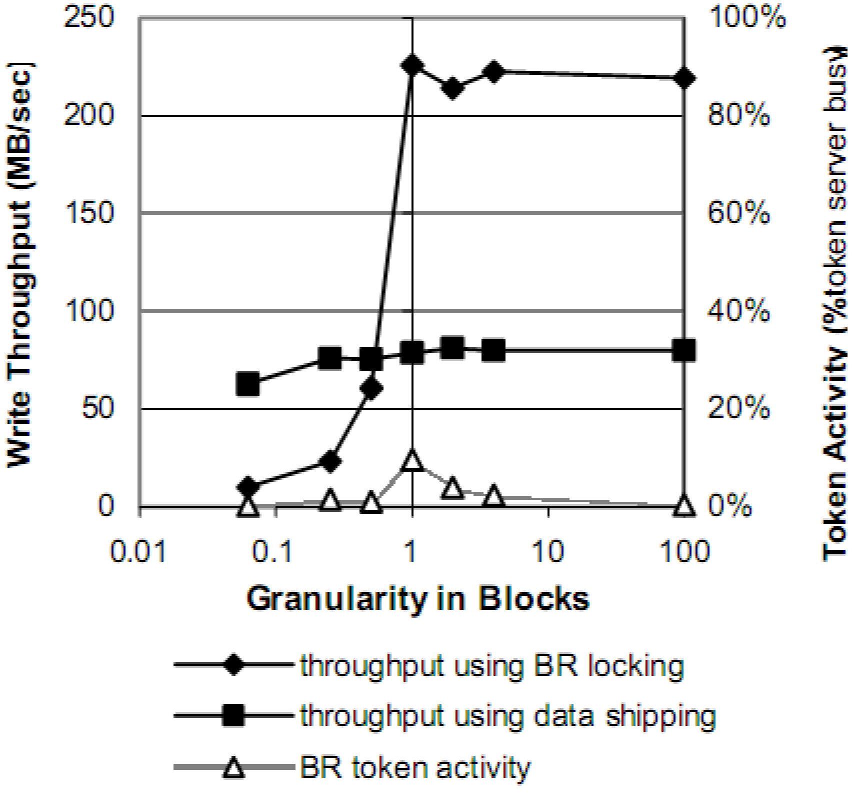
- Centralized global lock manager that runs on one of the nodes in the cluster.
- Local lock manager in each of the node.
- Global lock manager coordinates lock between local lock managers by handling out lock tokens.
- When there is a conflict from other node on the same object, then additional messages are sent to global lock manager.

Parallel Data Access

- Uses Byte-range locking to synchronize reads and writes to file data.
- This allows parallel applications to write concurrently to different parts of the same file.
- GPFS uses byte-range tokens to synchronize data block allocation and therefore rounds byte-range tokens to block boundaries.
- False sharing.



1600
1400
1200
1000
800
600
400
200
0
Throughput (MB/s)
0
2
4
6
8
10
12
14
16
18
20
22
24
Number of Nodes
each node reading a different file
all nodes reading the same file
each node writing to a different file
all nodes writing to the same file



250
200
150
100
50
0
Write Throughput (MB/sec)
100%
80%
60%
40%
20%
0%
Token Activity (%token server busy
0.01
0.1
1
10
100
Granularity in Blocks
throughput using BR locking
throughput using data shipping
BR token activity

Synchronizing access to file metadata

- GPFS uses inodes and indirect blocks to store file attributes and data block addresses.
- So multiple nodes writing to the same file will result in concurrent updates to the inode and indirect blocks of the file.
- Uses *shared-write lock* on the inode to allow concurrent writers on multiple nodes.
- Metanode
  - Elected dynamically
  - Responsible for updating inodes

Allocation Maps

- The allocation map records the allocation status of all disk blocks in the file system.
- 32 bits per disk block.
- The map is divided into a large, fixed number n of separately lockable regions.
- Allocation manager node maintains the free space statistics.

Token Manager Scaling

- The token manager keeps track of all lock tokens granted to all nodes in the cluster.
- Acquiring, relinquishing, upgrading, or downgrading a token requires a message to the token manager.
- Because of the lock conflicts that cause token revocation, there is a high load on token manager.
- GPFS uses a token protocol that significantly reduces the cost of token management.

Fault Tolerance

- Node failures
  - When a node fails, GPFS must
    - Restore metadata being update by failed node.
    - Release any tokens held by failed node.
    - Appoint replacements for special roles (metanode, allocation manager) played by failed node.
  - As GPFS stores logs on shared disks, any surviving node can perform log recovery on behalf of failed node.

Fault Tolerance

- Communication failures
  - Network partition
  - GPFS fences nodes that are no longer members of the group from accessing the shared disks
  - It invokes primitives available in the disk subsystem to stop accepting I/O requests from the other nodes.

Fault Tolerance

- Disk failures
  - GPFS supports replication.
  - When enabled, GPFS allocates space for two copies of each data or metadata block on two different disks and writes them to both locations.
  - Replication can be enabled separately for data and metadata.
  - If a part of a disk becomes unreadable (bad blocks), metadata replication in the file system ensures that only a few data blocks will be affected, rather than rendering a whole set of files inaccessible.

Scalable Online System Utilities

- GPFS allows adding, deleting, or replacing disks in an existing file system.
- When replication is enabled and a group of disks that were down becomes available again, GPFS must perform a metadata scan to find files with missing updates that need to be applied to these disks.
- GPFS appoints one of the nodes as a file system manager for each file system, which is responsible for coordinating these activities.

Experiences

- The granules of work handed out must be sufficiently small and of approximately equal size.
- Restricting management functions to a designated set of administrative nodes.
- Use of multiple threads to prefetch inodes for other files in the same directory.

Related Work

- Storage area network
  - Centralized metadata server
- SGI’s XFS file system
  - Not a clustered file system
- Frangipani
  - It implements whole-file locking only and therefore does not allow concurrent writes to the same file from multiple nodes.
- Global File System
  - External distributed lock manager

	GFS	GPFS
Scalability	Y	Y
Parallelism	N	Y
Cross-platform	N	N
Failure recovery	N	Y
Byte-range locking	Y	Y

GPFS V3.4

- Significant advances have been made recently in the area of data management where file systems, often in cooperation with other data management software (e.g. IBM's Tivoli Storage Manager or the collaborative High Performance Storage System (HPSS))
- GPFS now supports IBM® Blue Gene® and IBM® eServer Cluster systems
- GPFS is the file system for the ASC Purple Supercomputer
- IBM Research - Almaden is working with IBM's product divisions to extend GPFS to support a new 2011-2012 generation of supercomputers featuring up to 16,000 nodes and 500,000 processor cores

Summary and Conclusions

- Uses distributed locking and recovery techniques.
- Uses byte-range locking.
- Uses replication for fault tolerance.
- Able to scale up to the largest supercomputers in the world.