Distributed File Systems

CS 256/456

Dept. of Computer Science, University of Rochester

Distributed System

- A collection of loosely coupled processors interconnected by a communication network
  - Do not share memory or a clock
  - Remote versus local resource access

Network vs. Distributed Operating Systems

- Network operating system:
  - Users aware of multiple machines
  - Data/computation migration -> user's responsibility
- Distributed operating system:
  - Access to remote resources similar to local
  - Data/computation migration under control of OS

Data Migration

- Whole files (automated ftp): e.g., AFS first version
- On-demand (demand paging): e.g., NFS, SMB, AFS newer versions

Computation Migration

- RPC
- Messages (more concurrency)

Process Migration

- Rationale
  - Load balancing
  - Computation speedup
  - Hardware preference
  - Software preference
  - Data access
- Either automated (transparent) or user specifies how the process should migrate

Distributed File System

- One in which
  - clients,
  - servers (file servers/service software)), and
  - storage devices (potentially multiple and independent)

are dispersed among the machines of a distributed system and requires networked/messaging activity

Naming and Transparency

- Possible schemes
  - Host:local-name
  - Attach remote directories (mount: NFS)
  - Single global name structure (AFS)
- Location transparency vs. location independence?

Caching and Coherence

- Remote-service mechanism
  - Stateful vs. stateless
- Caching and coherence
  - Caching granularity
  - Cache update policy (write through vs. delayed write)
  - Client-initiated vs. server-initiated

Reliability and File Replication

- Reliability and file replication
  - Naming transparency
  - Availability vs. consistency

NFS (Network File System)

- Deals with heterogeneity using RPC/XDR
- Stateless – no open and close of files
- Interface transparent to user via VFS

The Andrew File System (AFS)

- Unified namespace (one /afs for everybody)
- One read-write copy
- Protection using access control lists (ACLs)
- Security using Kerberos 5 authentication

AFS: Namespace

- Managed by dedicated servers called “Vice”
- Local (root file system) and shared namespace with mounting similar to NFS
- Files partitioned into volumes
  - typically files of a single client
  - Possible migration and replication of volumes
- Clients run “Virtue” protocol

AFS: Security and Protection

- Security
  - Connection-based communication based on RPC
  - Encrypted
- Protection
  - Directories have access control lists
    - Allowed users or users not allowed
  - Regular UNIX bits for file protection

AFS: Caching and Coherence

- Caching of entire files
  - Callback mechanism to eliminate cached copies
  - One read-write copy

Distributed File Systems Issues

- Naming and transparency (location transparency versus location independence)
  - Host:local-name
  - Attach remote directories (mount)
  - Single global name structure
- Remote file access
  - Remote-service mechanism
    - Stateful vs. stateless
  - Caching and coherence
    - Cache update policy (write through vs. delayed write)
    - Client-initiated vs. server-initiated
- Reliability and file replication
  - Naming transparency
  - Availability vs. consistency

Distributed File Systems: Issues of Scale

Google File System - Motivation

- Component failures are normal
  - Fault tolerance and automatic recovery are needed
- Huge files are common (Multi GB)
  - Revisited I/O and block size assumptions
- Record appends are prevalent than random writes
  - Appending is the focus of performance optimization and atomicity guarantees
- Co-designing applications and the file-system API for flexibility
  - Relaxed consistency model
  - Atomic append

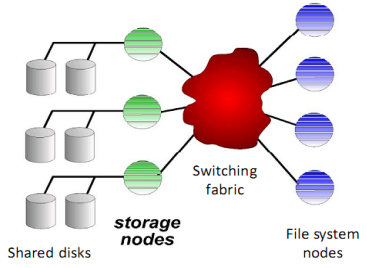
What is GPFS (1/2)

- A *parallel, shared-disk* file system for *cluster* computers
- Developed by *IBM* since 1993
- Available on AIX, linux and MS Windows Server 2003.
- Used on 6 of the 10 most powerful supercomputers in the world
  Extreme scalability!!

What is GPFS (2/2)

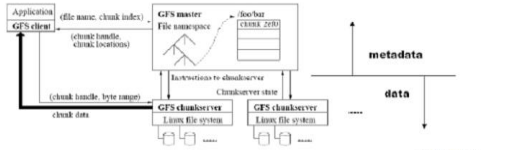
- *Extreme scalability from its shared-disk architecture*

Disk number: up to 4k
Disk size: up to 1TB



GPFS vs. GFS

GFS: Single Master + Multiply trunk server + Multiple client

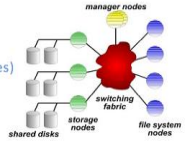


GPFS:

File system nodes

Manager nodes (could be any of the file system nodes)

Storage nodes



	Google File System	GPFS
Application Type	Large, distributed data-intensive	Supercomputing other…
Data access assumption	Large streaming r/w Mainly record appends	None
File Size assumption	Usually huge	None
Consistency	Relaxed	-
Synchronization	Centralized Management	Distributed locking + Centralized Management
Caching	Not needed	Needed
Data Unit	Chunk (64MB)	Block(typically 256KB)
Fault Tolerance	Constant monitoring Fast recovery Replication	Logging & recovery Support of RAID Replication