RAIDframe & Distributed Storage

url: http://www.bytelabs.org/papers.html

Igor Boehm (igor@bytelabs.org)
&
Jakob Praher (jp@hapra.at)

Institute for Information Processing and Microprocessor Technology
Johannes Kepler University Linz, Austria

Outline

1. RAIDframe
   - Motivation
   - RAID Levels
   - RAIDframe
2. Distributed Storage
   - Introduction
   - Distributed Storage Technologies
   - Distributed Storage over TCP/IP
   - Conclusion

Outline

1. RAIDframe
   - Motivation
   - RAID Levels
   - RAIDframe
2. Distributed Storage
   - Introduction
   - Distributed Storage Technologies
   - Distributed Storage over TCP/IP
   - Conclusion

RAIDframe - Rapid Prototyping Tool for RAID Systems

Purpose of RAID Systems

- Increase I/O performance by increasing parallelism.
- Improve dependability by adding redundant disks.

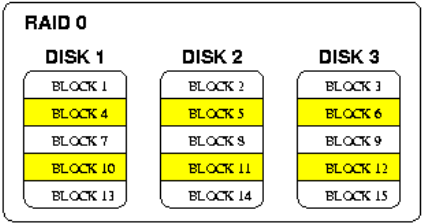
Purpose of RAIDframe

- Decrease complexity of advanced disk array architecture design.
- Seperate architectural policies from executional mechanism.

Outline

1. RAIDframe
   - Motivation
   - RAID Levels
   - RAIDframe
2. Distributed Storage
   - Introduction
   - Distributed Storage Technologies
   - Distributed Storage over TCP/IP
   - Conclusion

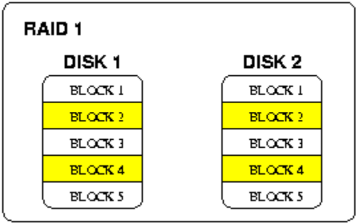
RAID 0 - No Redundancy



RAID 0

- non-redundant
- data striping accross components
- good performance for large disk access since many disks can operate at once

RAID 1 - Mirroring



RAID 1

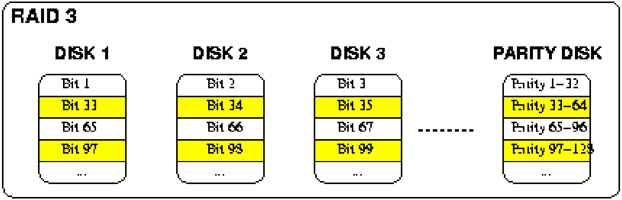
- provides mirroring
- twice as many disks as RAID 0
- reliability → linear multiple of the number of member disks

RAID 3 - Bit Interleaved Parity

RAID 3

- data interleaved **bit-wise** over data-disks
- additional parity disk tolerating single failure
- parity calculation:

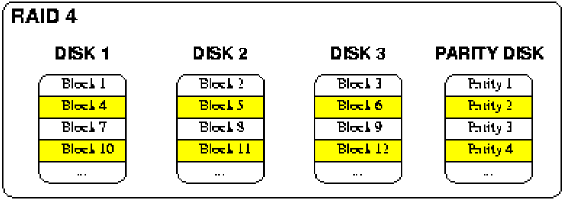
$$(P_{1-32} = B_1 \oplus B_2 \oplus B_3 \oplus \cdots \oplus B_{32}) \rightarrow (B_3 = B_1 \oplus B_2 \oplus P_{1-32} \oplus \cdots \oplus B_{32})$$



RAID 4 - Block-Interleaved Parity

RAID 4

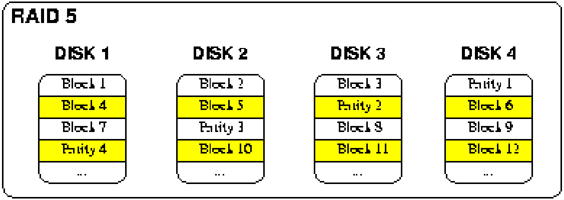
- data interleaved **block-wise** over data-disks
- parity associated with set of data-blocks
- reads smaller than striping unit access only one data-disk



RAID 5 - Block-Interleaved Distributed Parity

RAID 5

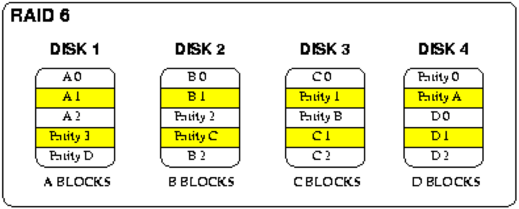
- single parity disk is a bottleneck in RAID 4
- parity associated with each row of data blocks is uniformly distributed over all disks
- thus multiple writes can occur simultaneously



RAID 6 - P+Q Redundancy

RAID 6

- parity based RAID leves 1-5 protect only against single disk failure
- this may not be sufficient for some critical applications
- add second calculation over data blocks with second parity block



Outline

1. RAIDframe
   - Motivation
   - RAID Levels
   - RAIDframe
2. Distributed Storage
   - Introduction
   - Distributed Storage Technologies
   - Distributed Storage over TCP/IP
   - Conclusion

RAIDframe - General Concepts

Increase amount of shared code between architectures by:

- Identifying a set of *primitive RAID operations*:
  - `Rd:` copy data from disk to buffer
  - `Wd:` write data from disk to buffer
  - `XOR:` xor contents of buffers
  - ...
- Build RAID operations based on *primitive operations*.
- Model RAID operations as directed acyclic graphs (DAGs).
- Provide simple *state engine* capable of executing DAGs.
- Provide generic reconstruction architecture.

Modelling RAID Operations with DAGs

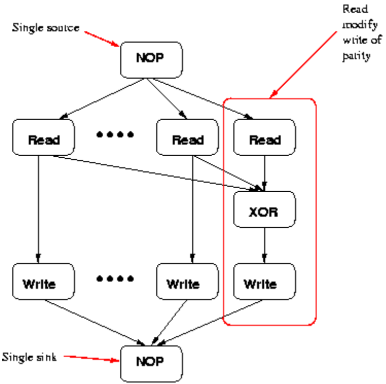
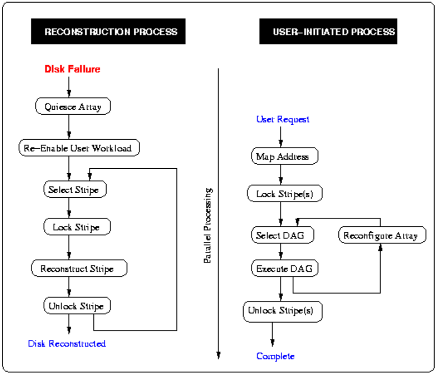


Figure: Small RAID4/5 write op.

Directed Acyclic Graphs

- By modelling RAID operations with DAGs, a strict partial order of those operations is guaranteed.
- Thus for all *a*, *b* and *c* in the set of primitive RAID operations, we have that:
  - $\neg(aRa)$ *(irreflexivity)*
  - $aRb \Rightarrow \neg(bRa)$ *(asymmetry)*
  - $aRb \wedge bRc \Rightarrow aRc$ *(transitivity)*

Reconstruction Architecture



Reconstruction Algorithm

- If a disk fails, RAIDframe uses a *diskoriented* instead of a *stripe oriented* algorithm.
- The *disk oriented* algorithm performs much better at utilizing disk bandwidth not absorbed by user requests.
- C reconstruction processes, where C is the amount of disks in the array, are spawned.
- $C - 1$ processes are associated with the surviving disks.
- The remaining process is associated with the replacement disk.
- ...

Reconstruction Architecture continued...

Algorithm for surviving disks

- **repeat**
  - find lowest numbered unit on **this** disk necessary for reconstruction
  - read unit into buffer
  - submit unit's data to centralized buffer manager for further processing
- **until** *(all necessary units have been read)*

Algorithm for replacement disks

- **repeat**
  - request buffer of reconstructed data from centralized buffer manager
  - write buffer to replacement disk
- **until** *(failed disk has been reconstructed)*

Internal Architecture and Extensibility

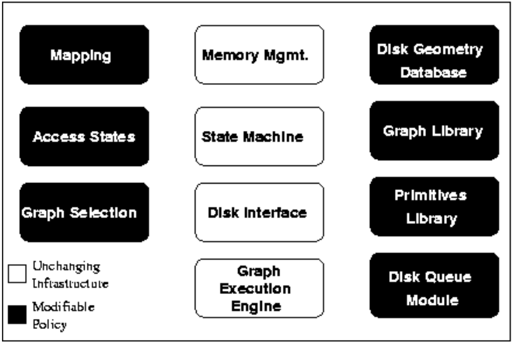


Figure: Internal Architecture - provides extensibility through separation of architectural policy from execution mechanism.

Outline

1. RAIDframe
   - Motivation
   - RAID Levels
   - RAIDframe
2. Distributed Storage
   - Introduction
   - Distributed Storage Technologies
   - Distributed Storage over TCP/IP
   - Conclusion

Bird’s Eye View of Distributed Storage

- Distributed Storage refers to Storage over Computer Networks.
- The main objective ist to reduce total cost of data management.
- Moving from per server storage to per network storage management.
- Distributed Storage is made possible by faster and faster network technologies.
- We will refer to local storage as Direct Attached Storage (DAS).

Outline

1. RAIDframe
   - Motivation
   - RAID Levels
   - RAIDframe
2. Distributed Storage
   - Introduction
   - Distributed Storage Technologies
   - Distributed Storage over TCP/IP
   - Conclusion

Direct Attached Storage (DAS)

- DAS refers to local or non networked storage.
- A typical server environment uses Small Computer System Interface (SCSI) nowadays also Serial Advanced Technology Attachement (S-ATA).
- A SCSI setup consists of:
  - Host Bus Adapter (HBA)
  - SCSI controller on every SCSI storage device
- HBA sits on one of the computer's local bus
- Address of *unit* is the path `<host,bus,target,LUN>`
- There is only one path for each *unit*
- The path identifies the *unit*

Distributed Storage

Storage Area Networks (SAN)

- bluntly: SAN is DAS with a longer wire.
- DAS and SAN use a raw **block based** access to data.
- Most SANs are operated in a point-to-point fashion.
- The filesystem is on the client side.

Network Attached Storage (NAS)

- ... storage is just a service **attached** to the network.
- NAS is baseed on a **file based** access to data.
- NAS are much like dedicated file servers.
- The filesystem is on the server (storage) side.

NAS and SAN

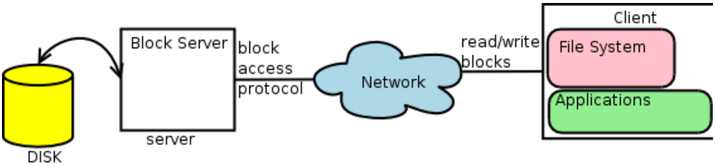


Figure: Block-Access (like in SAN)

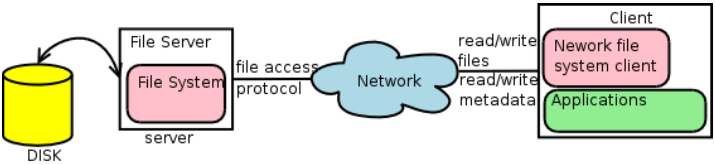


Figure: File-Access (like in NAS)

Outline

1 RAIDframe
- Motivation
- RAID Levels
- RAIDframe

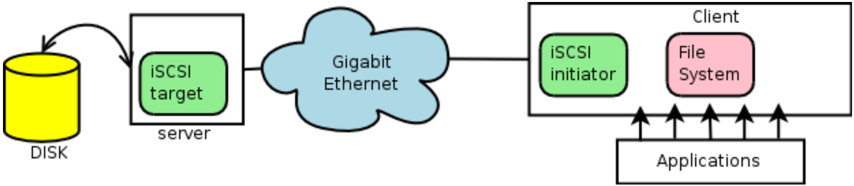
2 Distributed Storage
- Introduction
- Distributed Storage Technologies
- **Distributed Storage over TCP/IP**
- Conclusion

Motivation

- Ethernet is getting faster and faster
- Ethernet switching is very mature and provides point-to-point connections (like FiberChannel switching)
- Mass network hardware is cheap!
- Mass software is very mature! (often tested)
- TCP/IP over (10) Gigabit Ethernet fast enough to be used as a storage network
- We look at two technologies: iSCSI (SAN) and NFS (NAS).

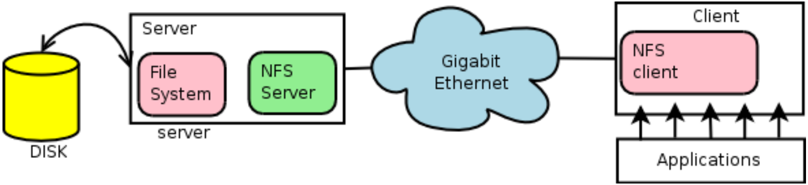
iSCSI

- iSCSI is short for Internet SCSI
- iSCSI uses a subset of the SCSI command set over TCP
- an iSCSI client is called the initiator
- an iSCSI storage is called the target
- iSCSI devices provide block-access
- the client has to provide the filesystem



NFS - Network File System

- NFS is a popular Network filesystem (esp. in Unix environments)
- NFS is defined on top of Remote Procedure Call(RPC) architecture.
- NFS exists in 3 major flavors: Version 2, 3, and 4
- Up to NFSv4, NFS was a **stateless** protocol.
- NFSv3 consisted of 4 protocols: mount, file system, locking, and status monitoring



NFSv4

Major change in NFS design

- Improved access and good performance on the Internet
- Strong security with security negotiation built into the protocol
- Enhanced cross-platform interoperability
- Extensibility of the protocol

Concrete changes

- Elimination of helper protocols (only one protocol)
- Introduction of COMPOUND calls to reduce rountrip time
- Statefulness - introduction of `OPEN` and `CLOSE`
- Only one TCP port required (Firewalls)
- Adds support for GSS-API (Kerberos, PKCS)
- Delegation for single user data (Home directories) - Metadata caching

Comparison - iSCSI vs NFS

- iSCSI provides block-access
- NFS provides file-access
- For data intensive workloads almost equal
- iSCSI supports aggressive meta-data caching (file system resides local)
- For meta-data intensive workloads iSCSI outperforms NFS by factor of two
- NFS could be enhanced to support aggressive meta-data caching

Outline

1 RAIDframe
- Motivation
- RAID Levels
- RAIDframe

2 Distributed Storage
- Introduction
- Distributed Storage Technologies
- Distributed Storage over TCP/IP
- Conclusion

Conclusions

- In the 80ties nobody believed in the scalability of Ethernet and IP
- Gigabit Ethernet and switching technology, TCP Offload Engines, etc. have shown that existing network technology is ready for storage networks
- NAS and SAN is not an either or
- NAS can be used as proxy to SANs
- In future Storage over the Internet? (Today: MBit Internet Access)
- Storage industry is working towards that: iSCSI, NFSv4

Thanks for your attention!