Universidade Estadual de Campinas
Instituto de Computação
MC504 Sistemas Operacionais

T28 RAID

Referência principal
Ch.38 of *Operating Systems: Three Easy Pieces* by Remzi and Andrea Arpaci-Dusseau (pages.cs.wisc.edu/~remzi/OSTEP/)

Discutido em classe em 05 de novembro de 2018

Arthur João Catto, PhD

2º semestre de 2018

RAID Structure

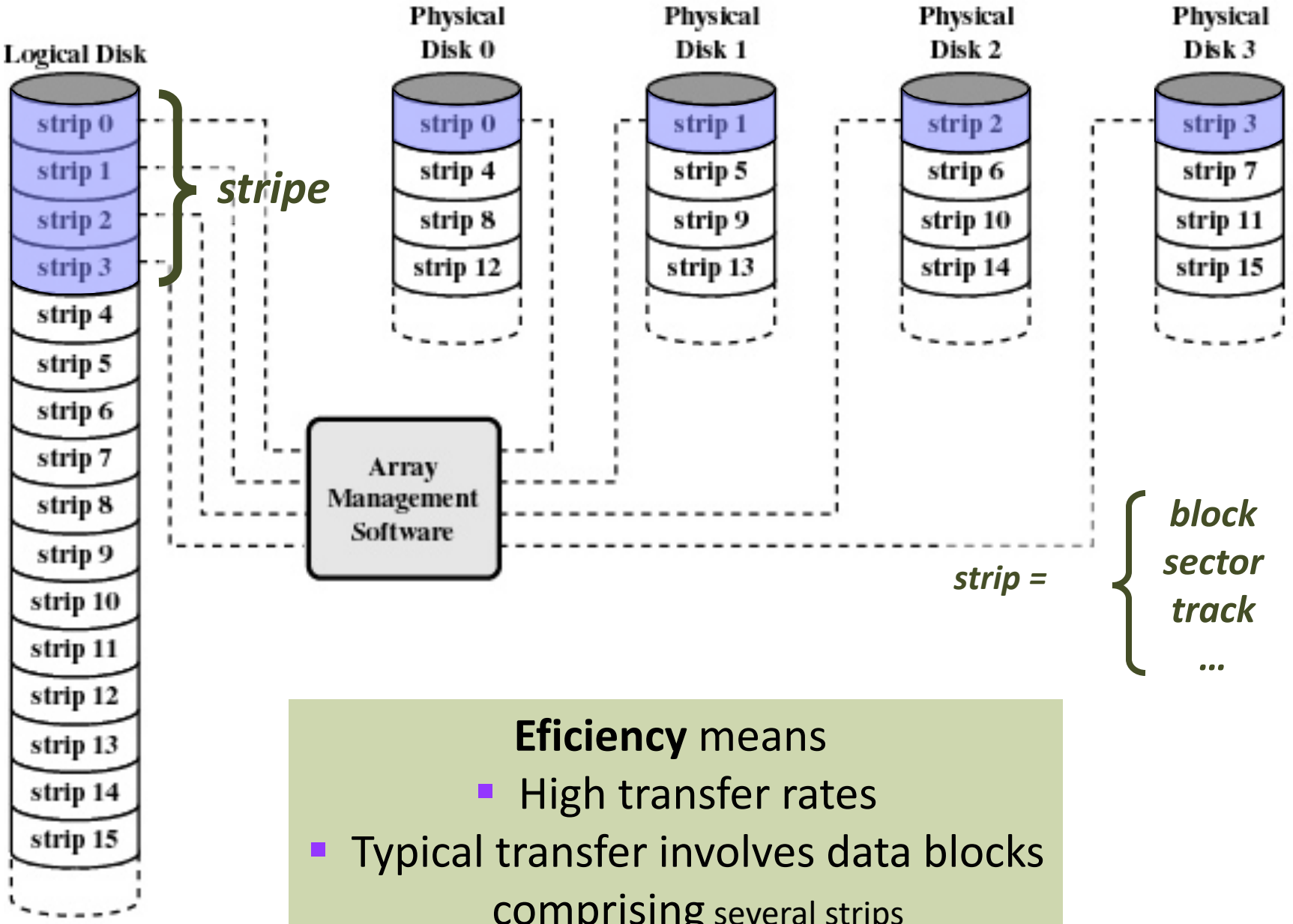
- RAID stands for redundant array of independent (originally *inexpensive*) disks.
- Set of physical disk drives viewed by the operating system as a single logical drive.
- Data are distributed across the physical drives of an array for higher reliability and data-transfer rates.
- Redundant disk capacity is used to store parity information and implement fault tolerance.
- RAID functionality can be implement by the OS or a dedicated controller.
- RAID offers good potential for incremental growth.

Reliability improvement via redundancy

- If the mean time to failure (MTTF) of a single disk is H hours, the MTTF of a disk set comprising N such disks is H/N.
  - This drawback can be compensated via redundancy.
  - Redundancy stands for extra information, not normally needed, that can be used to rebuild lost data in the case of failures.
    - The simplest form of redundancy is mirroring.

Performance improvement via parallelism

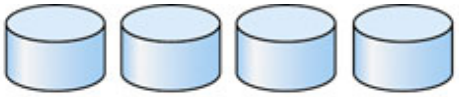
- There are two main goals for parallelism in a disk system
  - Increasing the throughput of multiple small accesses by load balancing.
  - Reducing the response time of large accesses.
- Transfer rates for multiple disks can be improved by striping data across the disks.
  - Bit-level striping
  - Block-level striping



Eficiency means

- High transfer rates
- Typical transfer involves data blocks comprising several strips

Typical Raid Levels



(a) RAID 0: non-redundant striping.

(b) RAID 1: mirrored disks.

(c) RAID 2: memory-style error-correcting codes.

(d) RAID 3: bit-interleaved parity.

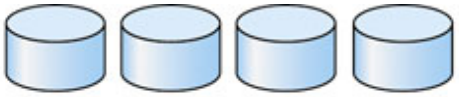
(e) RAID 4: block-interleaved parity.

(f) RAID 5: block-interleaved distributed parity.

(g) RAID 6: P + Q redundancy.

Typical Raid Levels

Typical Raid Levels



(a) RAID 0: non-redundant striping.

(b) RAID 1: mirrored disks.

(c) RAID 2: memory-style error-correcting codes.

(d) RAID 3: bit-interleaved parity.

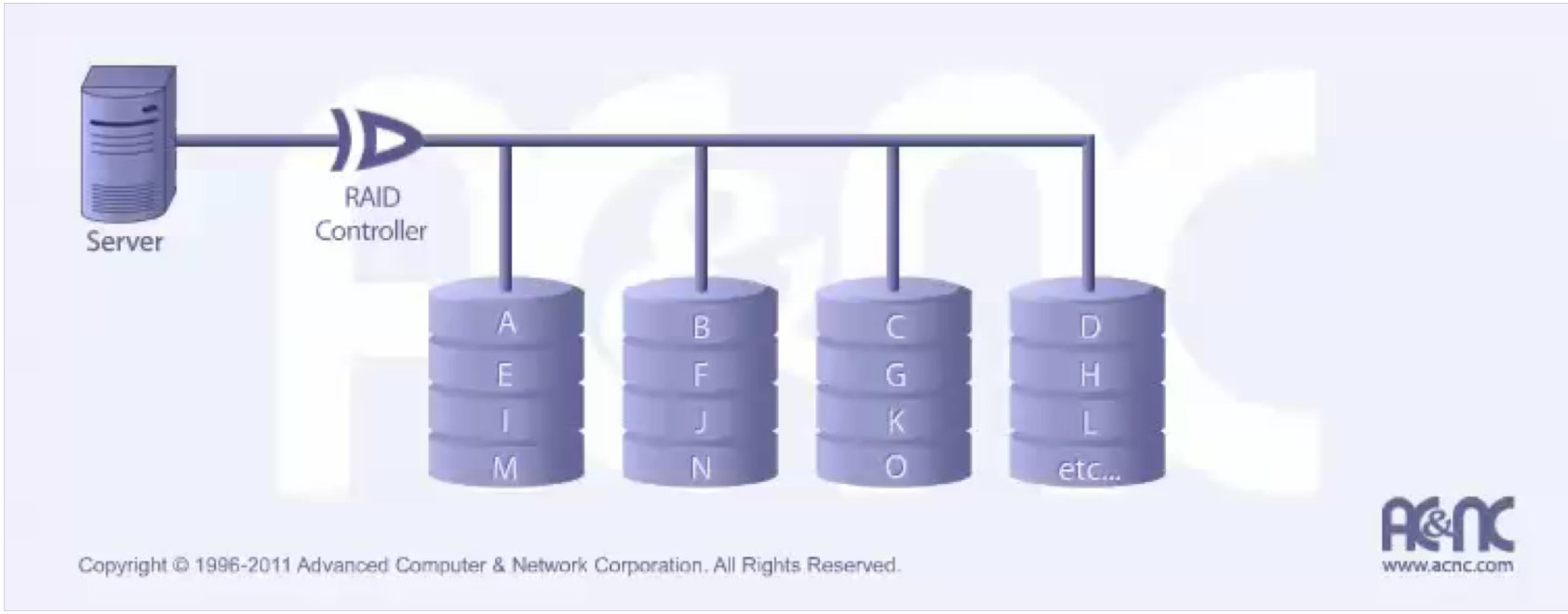
(e) RAID 4: block-interleaved parity.

(f) RAID 5: block-interleaved distributed parity.

(g) RAID 6: P + Q redundancy.

RAID 0 (non-redundant)

- A disk array with striping at the block level but without any redundancy.



RAID 0 (non-redundant)

Disadvantages

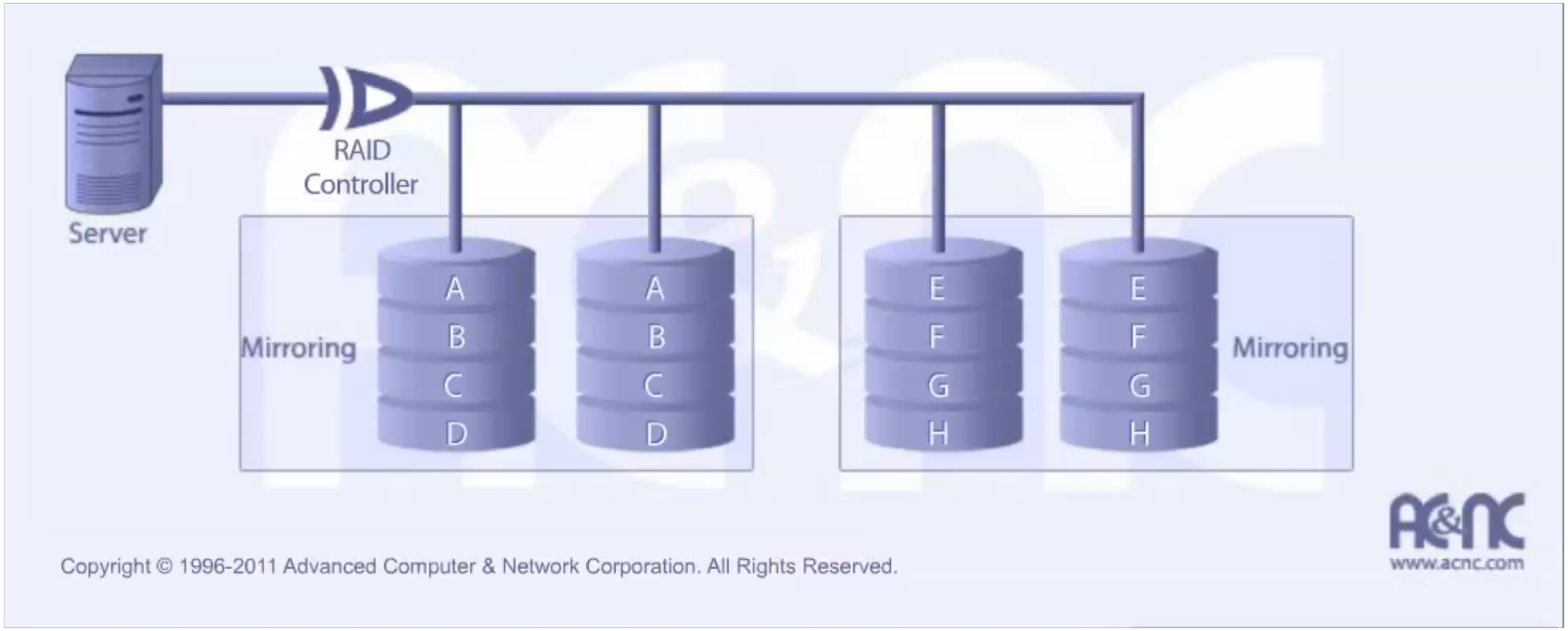
- Improved I/O performance
- Best when data is striped across multiple controllers with only one drive per controller
- No parity calculation overhead
- Very simple design
- Easy to implement

Characteristics & Advantages

- Not a “true” RAID because there is NO redundancy or fault-tolerance
- Failure of just one drive will result in all data in an array being lost
- Should never be used in mission critical environments

RAID 1 (mirrored)

- A disk array with simple disk mirroring.



RAID 1 (mirrored)

Disadvantages

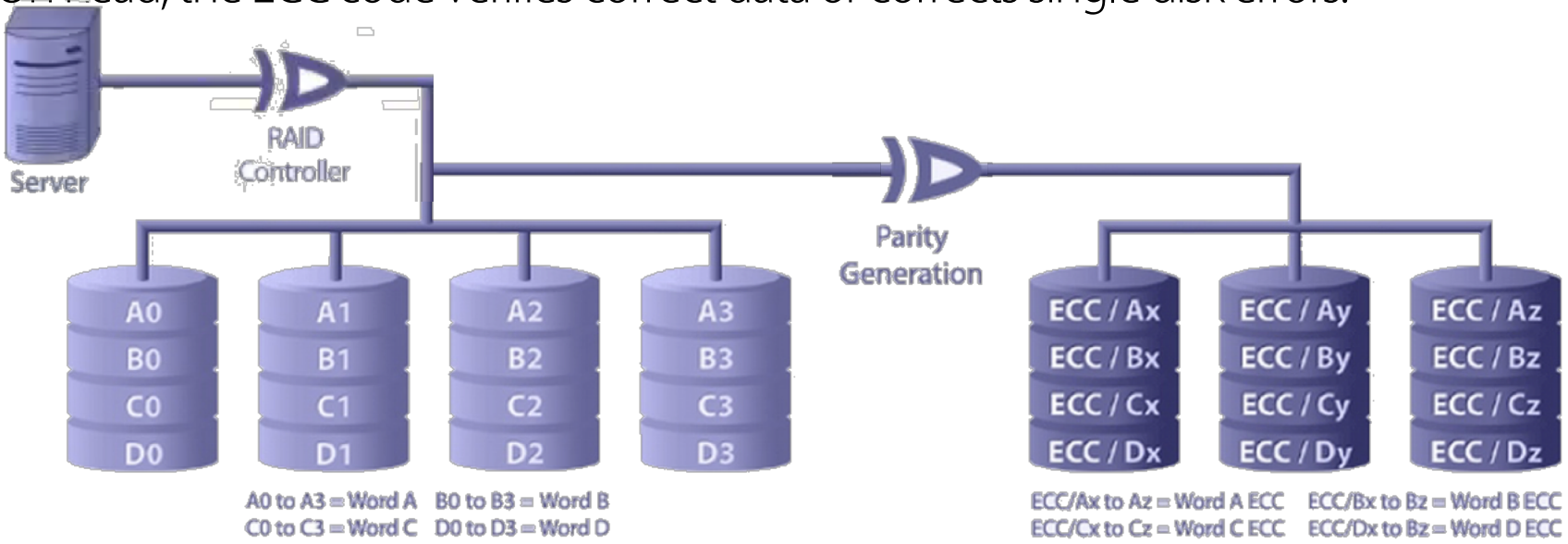
- One Write or two Reads possible per mirrored pair.
- Twice Read and same Write transaction rate of single disk.
- 100% redundancy.
- Same transfer rate per block of a single disk.
- Can sustain multiple simultaneous drive failures.
- Simplest RAID

Characteristics & Advantages

- Highest disk overhead of all RAIDs – inefficient
- RAID function typically done by system software
- Hardware implementation is strongly recommended
- May not support hot swap of failed disk when implemented in “software”

RAID 2 (redundancy via Hamming code)

- Each bit of data word is written to a data disk drive (4 in this example).
- Each data word has its Hamming Code ECC word recorded on the ECC disks (3 in this example).
- On Read, the ECC code verifies correct data or corrects single disk errors.



RAID 2 (redundancy via Hamming code)

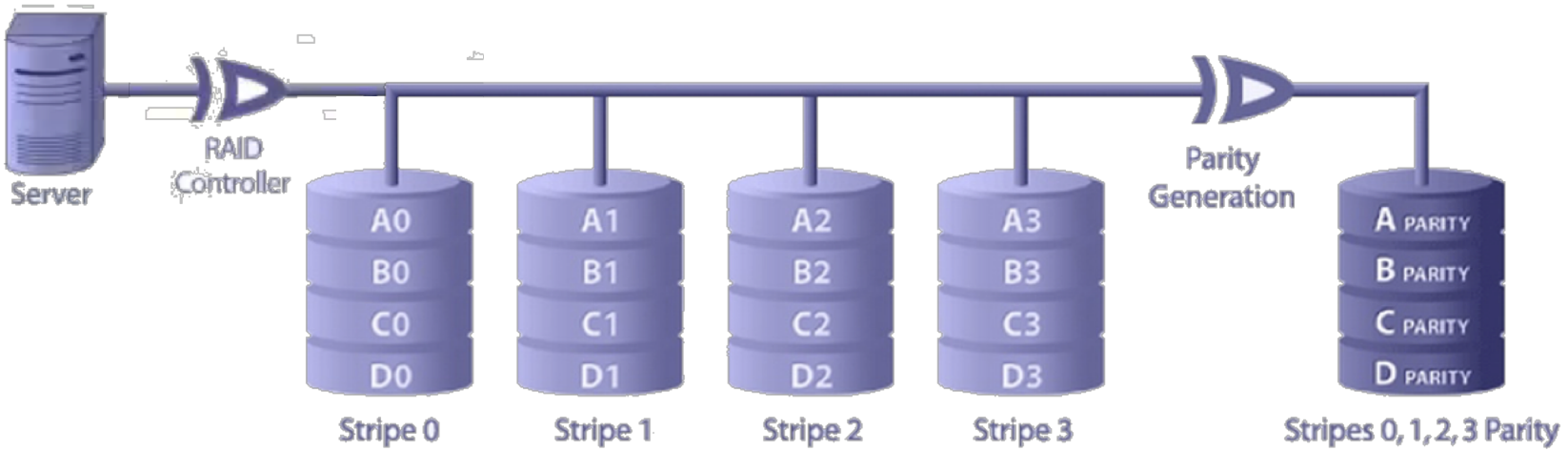
Disadvantages

- "On the fly" data error correction
- Extremely high data transfer rates possible
- The larger the word size, the better the ratio of data disks to ECC disks
- Relatively simple controller design compared to RAID levels 3,4 & 5

Characteristics & Advantages

- Very high ratio of ECC disks to data disks with smaller word sizes – inefficient
- Entry level cost very high thus requiring very high transfer rate need to justify
- Transaction rate is equal to that of a single disk at best
- No commercial implementations exist / not commercially viable

RAID 3 (bit-interleaved parity)



- $P(A) = A_0 \oplus A_1 \oplus A_2 \oplus A_3$, which enables recalculating any A_i
  - For example: $A_0 = P(A) \oplus A_1 \oplus A_2 \oplus A_3$

RAID 3 (bit-interleaved parity)

Disadvantages

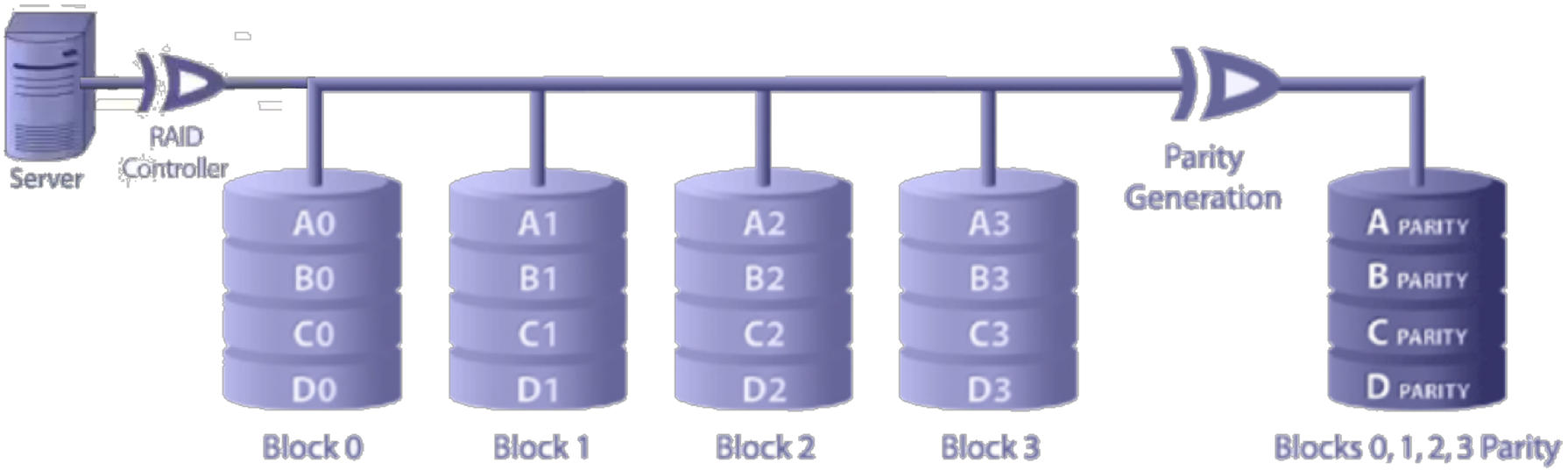
- Very high read data transfer rate
- Very high write data transfer rate
- Disk failure has an insignificant impact on throughput
- Low ratio of ECC (Parity) disks to data disks means high efficiency

Characteristics & Advantages

- Transaction rate equal to that of a single disk drive at best (if spindles are synchronized)
- Controller design is fairly complex
- Very difficult and resource intensive to do as a “software” RAID

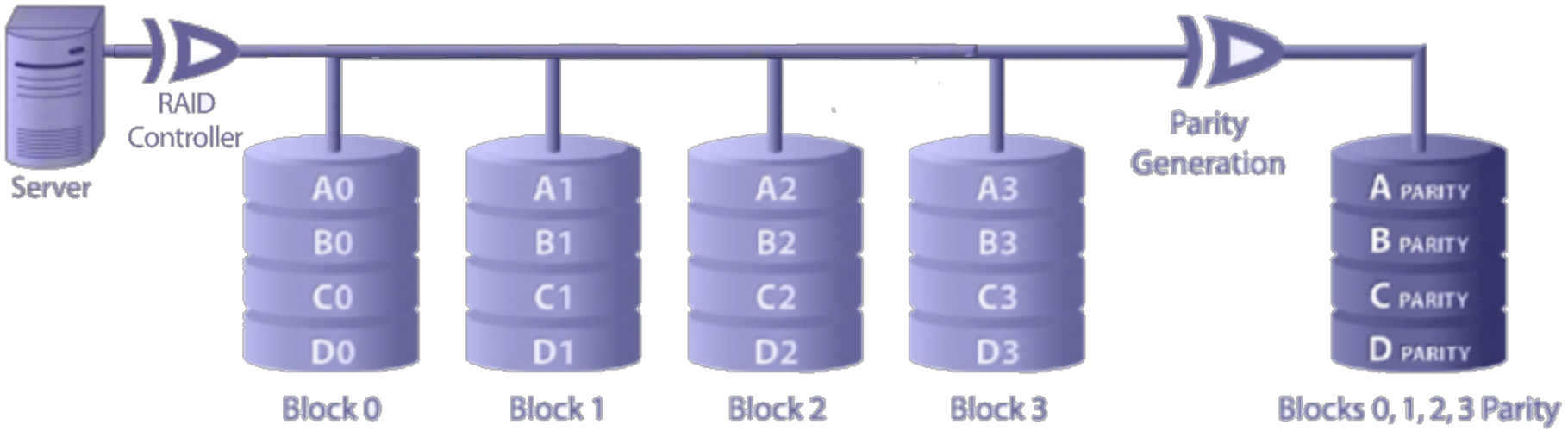
RAID 4 (block-interleaved parity)

- Each entire block is written onto a data disk.
- Parity for same rank blocks is generated on Writes, recorded on the parity disk and checked on Reads.
- RAID Level 4 requires a minimum of 3 drives to implement



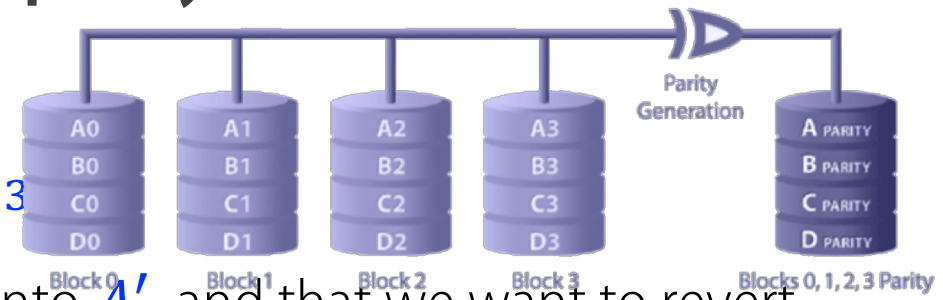
RAID 4 (block-interleaved parity)

- Each entire block is written onto a data disk.
- Parity for same rank blocks is generated on Writes, recorded on the parity disk and checked on Reads.
- RAID Level 4 requires a minimum of 3 drives to implement



RAID 4 (block-interleaved parity)

- Block parity is calculated as
  - $P(A) = A_0 \oplus A_1 \oplus A_2 \oplus A_3$



- Now, assume that A_0 is changed into A_0' and that we want to revert that change.

- We can calculate
  - $P'(A) = A_0' \oplus A_1 \oplus A_2 \oplus \oplus A_3$ ∴

    $P'(A) = A_0' \oplus A_1 \oplus A_2 \oplus A_3 \oplus A_0 \oplus A_0$ ∴

    $P'(A) = A_0' \oplus P(A) \oplus A_0$

  and then recalculate
  - $A_0 = A_0' \oplus P(A) \oplus P'(A)$

RAID 4 (block-interleaved parity)

Disadvantages

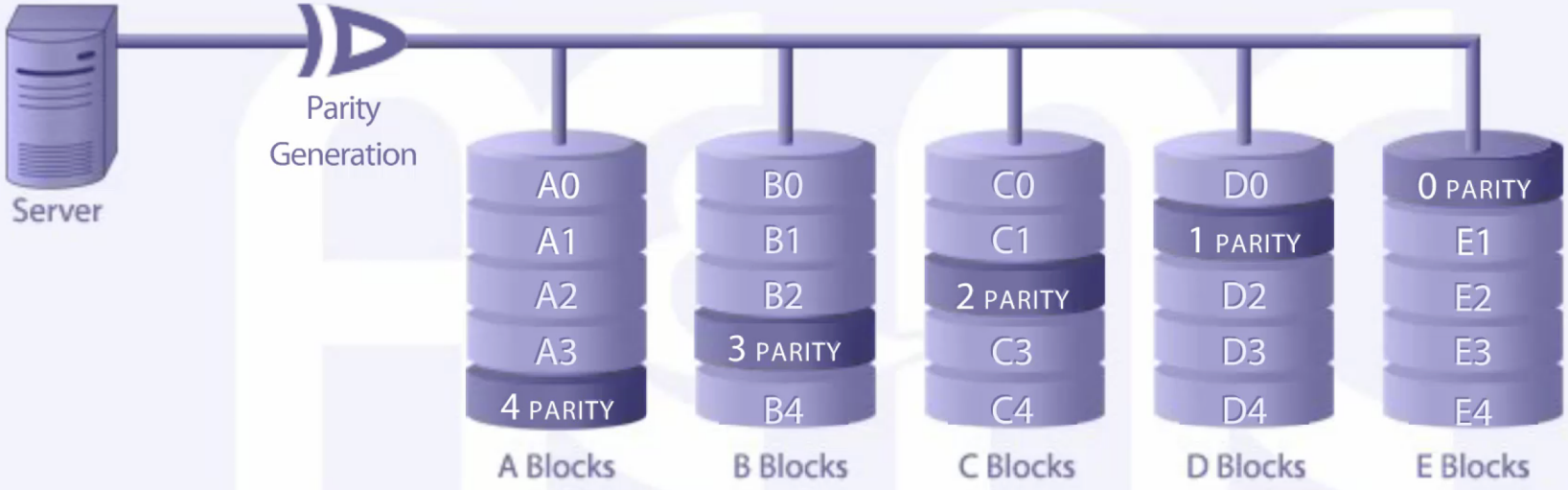
- Very high read data transaction rate
- Low ratio of ECC (Parity) disks to data disks means high efficiency
- High aggregate read transfer rate

Characteristics & Advantages

- Quite complex controller design
- Worst write transaction rate and write aggregate transfer rate
- Difficult and inefficient data rebuild in the event of disk failure
- Block read transfer rate equal to that of a single disk

RAID 5 (block-interleaved distributed parity)

- Each entire data block is written on a data disk.
- Parity for blocks in the same rank is generated on Writes, recorded in a distributed location and checked on Reads.
- RAID Level 5 requires a minimum of 3 drives to implement



AC&NC
www.acnc.com

RAID 5 (block-interleaved distributed parity)

Disadvantages

- Highest Read data transaction rate
- Medium Write data transaction rate
- Low ratio of ECC (Parity) disks to data disks means high efficiency
- Good aggregate transfer rate

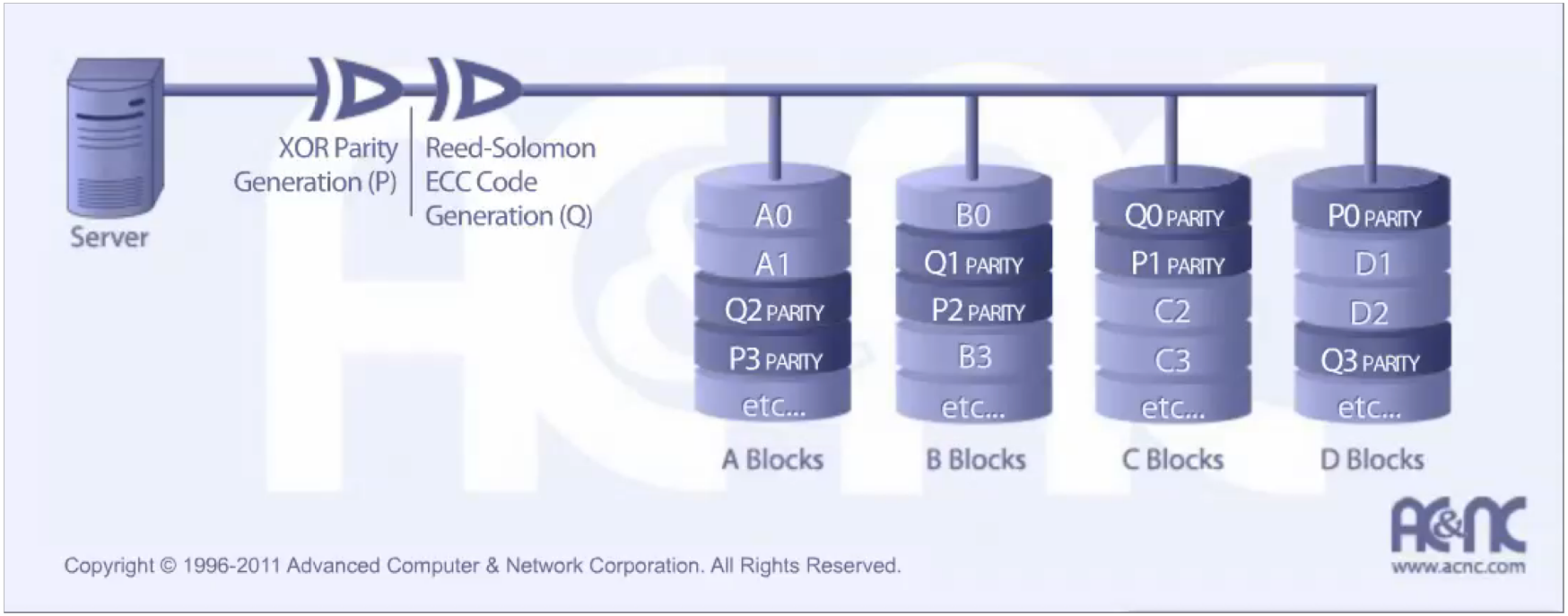
Characteristics & Advantages

- Disk failure has a medium impact on throughput
- Most complex controller design
- Difficult to rebuild in the event of a disk failure (as compared to RAID level 1)
- Individual block data transfer rate same as single disk

RAID 6 (dual redundancy)

- RAID 6 is essentially an extension of RAID level 5 which allows for additional fault tolerance by using a second independent distributed parity scheme (dual parity)
- Data is striped on a block level across a set of drives, just like in RAID 5, and a second set of parity is calculated by a different algorithm and written across all the drives
- RAID Level 6 requires a minimum of 4 drives to implement

RAID 6 (dual redundancy)



RAID 6 (dual redundancy)

Characteristics & Advantages

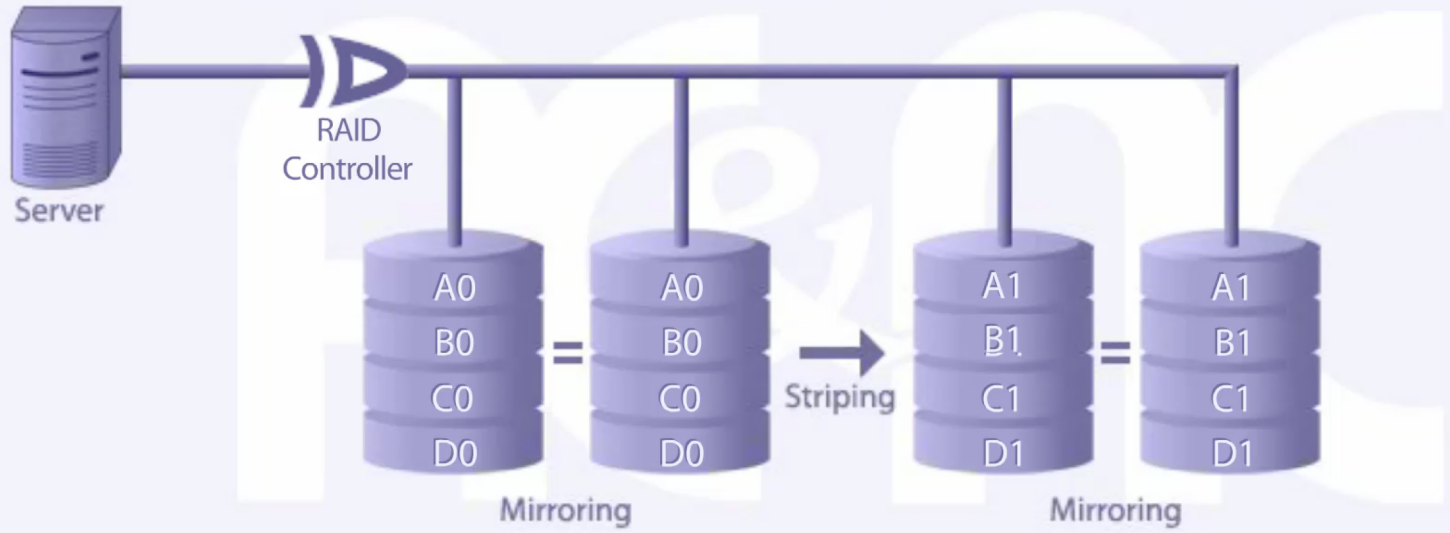
- RAID 6 provides for an extremely high data fault tolerance and can sustain multiple simultaneous drive failures
- RAID 6 protects against multiple bad block failures while non-degraded (all disks ok)
- RAID 6 protects against a single bad block failure while operating in degraded mode (one faulty disk)
- Perfect solution for mission critical applications

Disadvantages

- More complex controller design
- Controller overhead to compute parity addresses is extremely high
- Write performance can be brought on par with RAID Level 5 by using a custom ASIC for computing Reed-Solomon parity
- Requires N+2 drives to implement because of dual parity scheme

RAID 10 High Reliability and Performance

- RAID 10 is implemented as a striped array of RAID 1 arrays
- RAID Level 10 requires a minimum of 4 drives



RAID 10 (dual redundancy)

Disadvantages

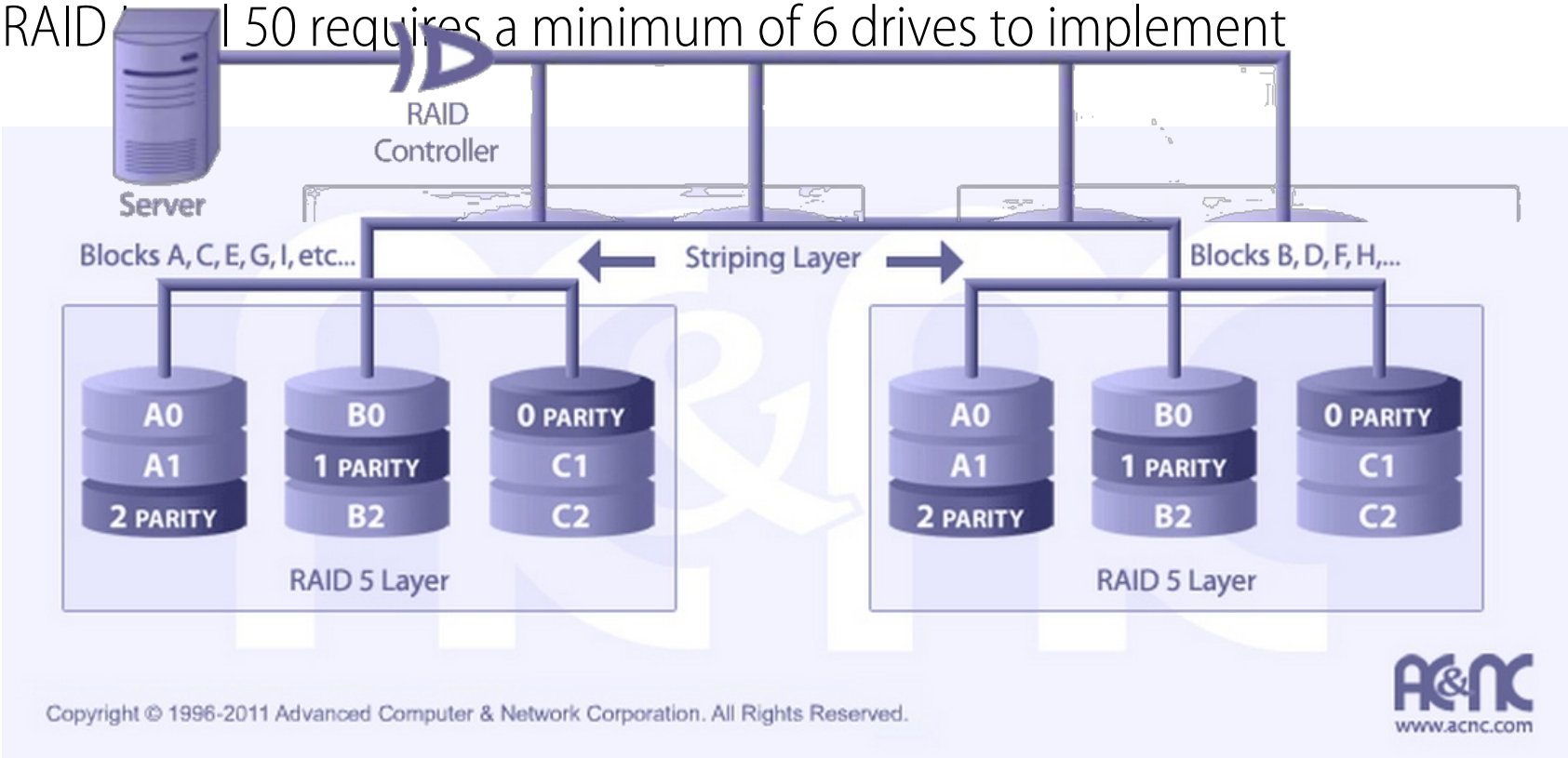
- RAID 10 has the same fault tolerance as RAID level 1
- RAID 10 has the same overhead for fault-tolerance as mirroring alone
- High I/O rates are achieved by striping RAID 1 segments
- Under certain circumstances. RAID 10 array can sustain multiple simultaneous drive failures
- Excellent solution for sites that would have otherwise gone with RAID 1 but need some additional performance boost

Characteristics & Advantages

- Very expensive / High overhead
- All drives must move in parallel to proper track lowering sustained performance
- Very limited scalability at a very high inherent cost

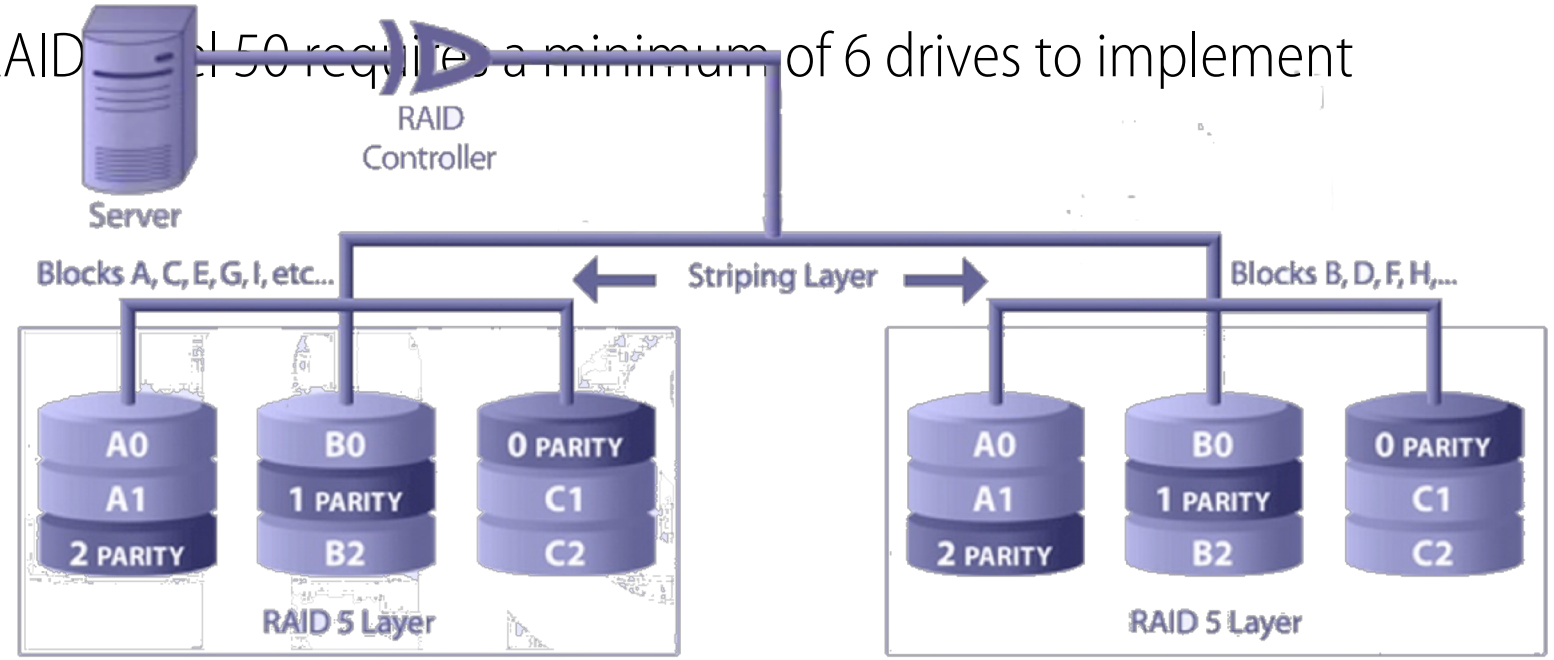
RAID 50 High I/O and Data Transfer Rates

- RAID 50 should have been called “RAID 03” because it was implemented as
  a striped (RAID level 0) array whose segments were RAID 3 arrays (during mid-90s)
- RAID level 50 requires a minimum of 6 drives to implement



RAID 50 High I/O and Data Transfer Rates

- RAID 50 should have been called “RAID 03” because it was implemented as
  a striped (RAID level 0) array whose segments were RAID 3 arrays (during mid-90s)
- RAID level 50 requires a minimum of 6 drives to implement



RAID 50 (high I/O and data transfer rates)

Disadvantages

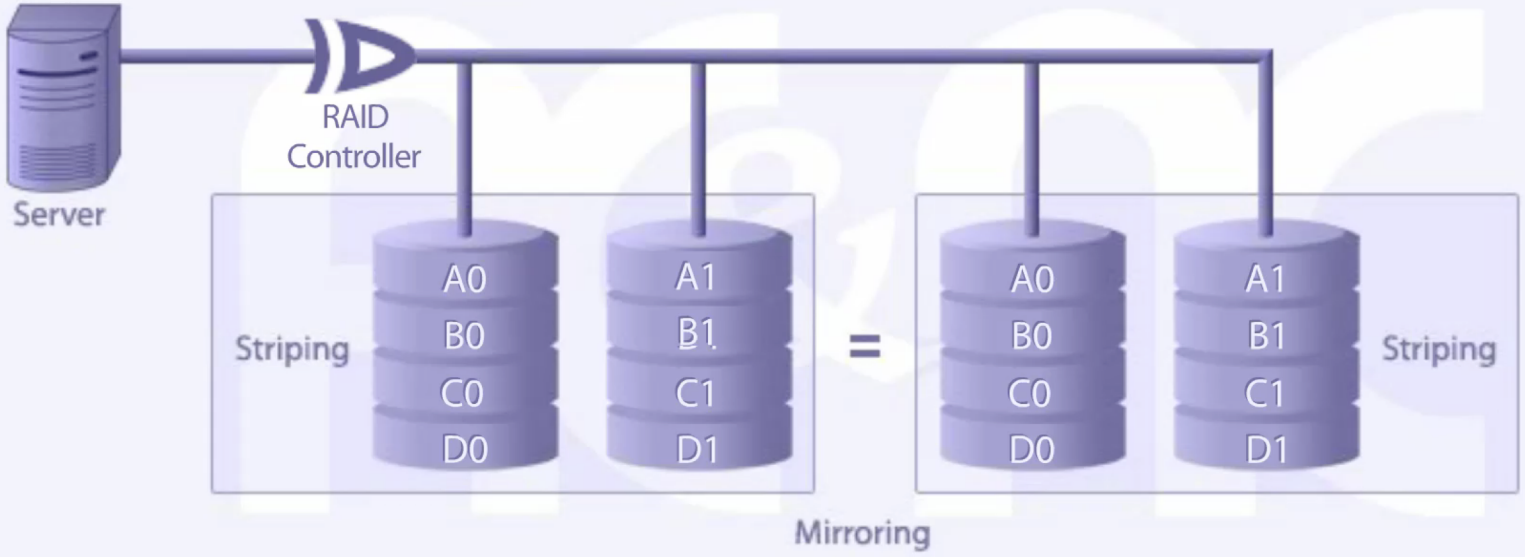
- RAID 50 is more fault tolerant than RAID 5 but has twice the parity overhead
- High data transfer rates are achieved thanks to its RAID 5 array segments
- High I/O rates for small requests are achieved thanks to its RAID 0 striping
- Maybe a good solution for sites who would have otherwise gone with RAID 5 but need some additional performance boost

Characteristics & Advantages

- Very expensive to implement
- All disk spindles must be synchronized, which limits the choice of drives
- Failure of two drives in one of the RAID 5 segments renders the whole array unusable

RAID 0+1 (high data transfer performance)

- RAID 0+1 is implemented as a mirrored array whose segments are RAID 0 arrays
- RAID Level 0+1 requires a minimum of 4 drives to implement



RAID 0+1 (high data transfer performance)

Disadvantages

- RAID 0+1 has the same fault tolerance as RAID level 5
- RAID 0+1 has the same overhead for fault-tolerance as mirroring alone
- High I/O rates are achieved thanks to multiple stripe segments
- Excellent solution for sites that need high performance but are not concerned with achieving maximum reliability

Characteristics & Advantages

- RAID 0+1 is NOT to be confused with RAID 10.
- A single drive failure will cause the whole array to become, in essence, a RAID Level 0 array
- Very expensive / High overhead
- All drives must move in parallel to proper track lowering sustained performance
- Very limited scalability at a very high inherent cost

Category	RAID Level	Description	Disks required	Data availability	Large I/O data transfer capacity	Small I/O request rate
Striping	0	Nonredundant	N	Lower than single disk	Very high	Very high for both read and write
Mirroring	1	Mirrored	N	Higher than RAID 2, 3, 4, or 5; lower than RAID 6	Higher than single disk for read; similar to single disk for write	Up to twice that of a single disk for read; similar to single disk for write
Parallel access	2	Redundant via Hamming code	$N + m$	Much higher than single disk; comparable to RAID 3, 4, or 5	Highest of all listed alternatives	Approximately twice that of a single disk
	3	Bit-interleaved parity	$N + 1$	Much higher than single disk; similar to RAID 2, 4, or 5	Highest of all listed alternatives	Approximately twice that of a single disk
Independent access	4	Block-interleaved parity	$N + 1$	Much higher than single disk; comparable to RAID 2, 3, or 5	Similar to RAID 0 for read; rather lower than single disk for write	Similar to RAID 0 for read; significantly lower than single disk for write
	5	Block-interleaved distributed parity	$N + 1$	Much higher than single disk; comparable to RAID 2, 3, or 4	Similar to RAID 0 for read; lower than single disk for write	Similar to RAID 0 for read; generally lower than single disk for write
	6	Block-interleaved dual distributed parity	$N + 2$	Highest of all listed alternatives	Similar to RAID 0 for read; lower than RAID 5 for write	Similar to RAID 0 for read; significantly lower than RAID 5 for write

Hamming Code

Hamming code for detecting and correcting a single error

- Consider m data bits $d_m \ldots d_3 d_2 d_1$
- Adding 1 parity bit we create a code which can detect one error using $m + 1$ bits

Hamming code for detecting and correcting a single error

- In order to detect and correct one error the code must also indicate the position of such error
  - Assume that for that purpose k extra bits are required, so that the code now has
    $n = m + k$ bits
  - The k extra bits must be able to
    - spot an error in any of the n bit positions and

$$\therefore 2^k \geq n + 1 \therefore 2^{n-m} \geq n + 1 \therefore \frac{2^n}{n+1} \geq 2^m$$

Hamming code for detecting and correcting a single error

- Let
  - $b_n b_{n-1} \ldots b_2 b_1$ be the bits in the full code (m of which are the data bits)
  - $p_k \ldots p_2 p_1 (\in b_n b_{n-1} \ldots b_2 b_1)$ be the associated parity bits
- We want $p_k \ldots p_2 p_1$ (read as a binary number) to give the position of the error or 0 if the data is correct.
- Let's try to find where to place the parity bits.

Hamming code for detecting and correcting a single error

- We want $p_k \ldots p_2 p_1$ (read as a binary number) to give the position of the error or 0 if the data is correct.

Error in	p_k	…	p_3	p_2	p_1
b_1	0		0	0	1
b_2	0		0	1	0
b_3	0		0	1	1
b_4	0		1	0	0
b_5	0		1	0	1
b_6	0		1	1	0
b_7	0		1	1	1
…					

Hamming code for detecting and correcting a single error

- Now analyze the contribution of each bit b_i to the code bits p_j.
  - Every b_i contributes to several p_j with the exception of $b_1, b_2, b_4, \ldots$ which contribute to exactly only one.
- So, if we use positions $1, 2, 4, \ldots$ for the parity bits, they will be independent of each other.

Error in	p_k	…	p_3	p_2	p_1
b_1	0		0	0	1
b_2	0		0	1	0
b_3	0		0	1	1
b_4	0		1	0	0
b_5	0		1	0	1
b_6	0		1	1	0
b_7	0		1	1	1
…					

Hamming code for detecting and correcting a single error

- So, we will have the full code
b_n	b_{n-1}	…	b_7	b_6	b_5	b_4	b_3	b_2	b_1
		…	d_4	d_3	d_2	p_3	d_1	p_2	p_1

    corresponding to

- Reexamining the contribution table, we see that the parity bits can be generated from the data bits as
  - $p_1 = b_3 \oplus b_5 \oplus b_7 \oplus \cdots$
  - $p_2 = b_3 \oplus b_6 \oplus b_7 \oplus \cdots$
  - $p_3 = b_5 \oplus b_6 \oplus b_7 \oplus \cdots$
  - …

Error in	p_k	…	p_3	p_2	p_1
b_1	0		0	0	1
b_2	0		0	1	0
b_3	0		0	1	1
b_4	0		1	0	0
b_5	0		1	0	1
b_6	0		1	1	0
b_7	0		1	1	1
…					

Hamming code for detecting and correcting a single error

- Given a full code $b_n b_{n-1} \dots b_2 b_1$, verification is done calculating
  - $x_1 = b_1 \oplus b_3 \oplus b_5 \oplus b_7 \oplus \cdots$
  - $x_2 = b_2 \oplus b_3 \oplus b_6 \oplus b_7 \oplus \cdots$
  - $x_3 = b_4 \oplus b_5 \oplus b_6 \oplus b_7 \oplus \cdots$
  - $\dots$
- Since, for instance, $b_1 = b_3 \oplus b_5 \oplus b_7 \oplus \cdots$
  - If all bits are correct, x_1 must be 0.
  - If any single bit is wrong, x_1 will be 1.
- The same applies to $x_2, x_3, \dots$
- If we look at the contribution table again, it is not difficult to conclude that the value $\dots x_3\ x_2\ x_1$ indicates the correctness of the code or the position of the error.

Hamming code for detecting and correcting a single error

- **Example 1:** Given a data code, generate the corresponding full code
  - Assume $m = 4$ and $d_4\ d_3\ d_2\ d_1 = 1\ 1\ 0\ 0$
  - We must find the smallest $n > m$ such that $\frac{2^n}{n+1} \geq 2^m$, which is 7.
  - The full code $b_7\ b_6\ b_5\ b_4\ b_3\ b_2\ b_1$ will correspond to $1_4 1_3 0_2\ p_3\ 0_1\ p_2\ p_1$
  - The parity bits are calculated as
    - $p_1 = b_3 \oplus b_5 \oplus b_7 = 0 \oplus 0 \oplus 1 = 1$
    - $p_2 = b_3 \oplus b_6 \oplus b_7 = 0 \oplus 1 \oplus 1 = 0$
    - $p_3 = b_5 \oplus b_6 \oplus b_7 = 0 \oplus 1 \oplus 1 = 0$
  - Thus, the full code will be $1\ 1\ 0\ 0\ 0\ 0\ 1$

Hamming code for detecting and correcting a single error

- **Example 2:** In example 1, find an error in a data bit
  - The full code was $\mathbf{1\ 1\ 0\ 0\ 0\ 0\ 1}$
  - Suppose that we receive
    $\mathbf{1\ 0\ 0\ 0\ 0\ 0\ 1}$, with a wrong value in bit b_6 (which corresponds to data bit d_3)
  - Code checking is done by calculating
    - $x_1 = b_1 \oplus b_3 \oplus b_5 \oplus b_7 = 1 \oplus 0 \oplus 0 \oplus 1 = 0$
    - $x_2 = b_2 \oplus b_3 \oplus b_6 \oplus b_7 = 0 \oplus 0 \oplus 0 \oplus 1 = 1$
    - $x_3 = b_4 \oplus b_5 \oplus b_6 \oplus b_7 = 0 \oplus 0 \oplus 0 \oplus 1 = 1$
  - There is an error (since $x_3\ x_2\ x_1$ is not 0) and the wrong bit is given by
    $x_3\ x_2\ x_1 = 1_3\ 1_2\ 0_1 = 6$.

Hamming code for detecting and correcting a single error

- **Example 3:** In example 1, find an error in a parity bit
  - The full code was $\mathbf{1\ 1\ 0\ 0\ 0\ 0\ 1}$
  - Suppose that we receive
    $\mathbf{1\ 1\ 0\ 0\ 0}\ 1\ \mathbf{1}$, with a wrong value in bit b_2 (which corresponds to parity bit p_2)
  - Code checking is done by calculating
    - $x_1 = b_1 \oplus b_3 \oplus b_5 \oplus b_7 = 1 \oplus 0 \oplus 0 \oplus 1 = 0$
    - $x_2 = b_2 \oplus b_3 \oplus b_6 \oplus b_7 = 1 \oplus 0 \oplus 1 \oplus 1 = 1$
    - $x_3 = b_4 \oplus b_5 \oplus b_6 \oplus b_7 = 0 \oplus 0 \oplus 1 \oplus 1 = 0$
  - There is an error (since $x_3\ x_2\ x_1$ is not 0) and the wrong bit is given by
    $x_3\ x_2\ x_1 = 0_3\ 1_2\ 0_1 = 2$.