

# RAID: An Introduction

Derek Konigsberg

AITP @ UCF

# What is RAID?

- Redundant Array of Inexpensive Disks
- Originally described in a paper from UC Berkeley in 1988
- RAID is a concept that allows multiple physical disks to be combined into a single logical disk, often adding redundancy in the process

Note: Today RAID is often defined as a “Redundant Array of Independent Disks,” because the drives used in higher-end RAID solutions are hardly inexpensive

# Why do we need it?

- Disks can be too small
- Disks can be too slow
- Disks can be too unreliable

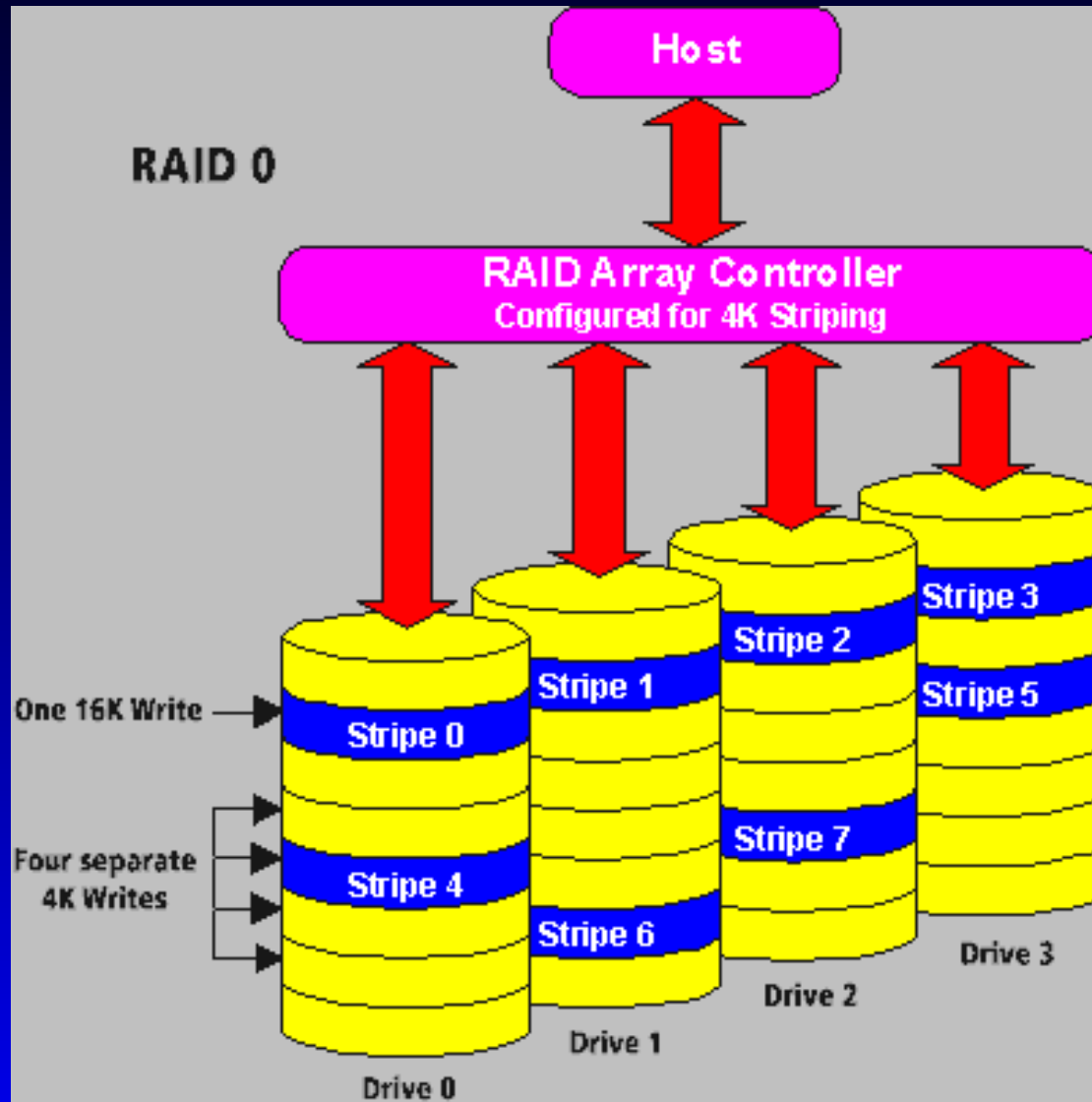
# How does it work?

- It is important to note that the term “RAID” describes a series of concepts, that can be implemented in software and/or hardware, and even combined with each other, to form a storage solution
- Commonly, a number of drives are combined to form a “RAID Array,” which is presented to the computer as a single large disk
- RAID is officially defined in a series of five “levels,” each approaching the problems in their own ways.
- Levels 0, 1, 3, and 5 are widely used, while levels 2 and 4 have never seriously been adopted by anyone

# RAID 0

- Striped drives without redundancy, using any number of drives
- This level is not defined in the original Berkeley paper
- The benefit of RAID 0 is in terms of raw performance at the cost of reliability, as any drive failure will result in a loss of the entire array
- Common uses of this level are for transient data, such as video editing

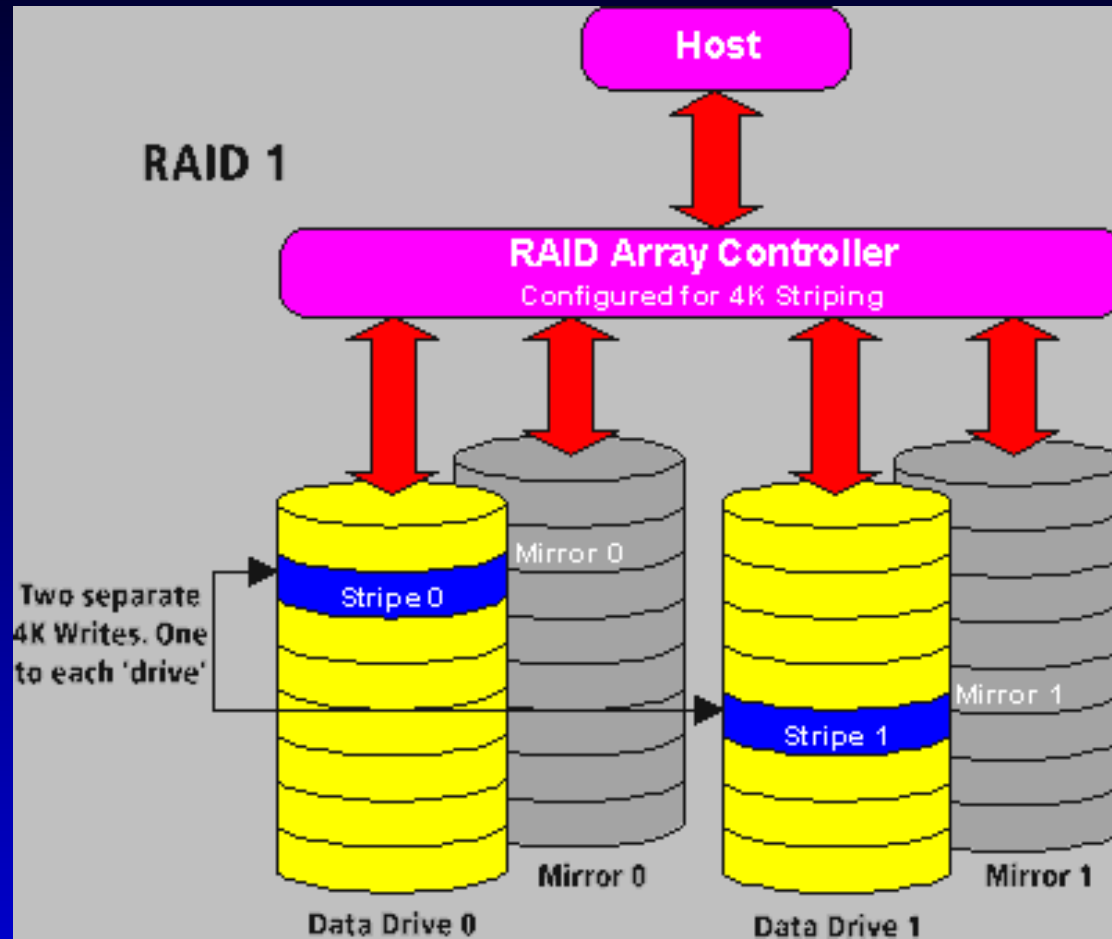
# RAID 0



# RAID 1

- Drive mirroring, using 2 drives ( $N*2$  redundancy)
- Write speed is a bit slower than a single drive, but read speed can be almost doubled
- This is a simple configuration, commonly used for things like system disks where you need redundancy but not a lot of capacity
- One drive failure can be tolerated, with no loss in performance

# RAID 1





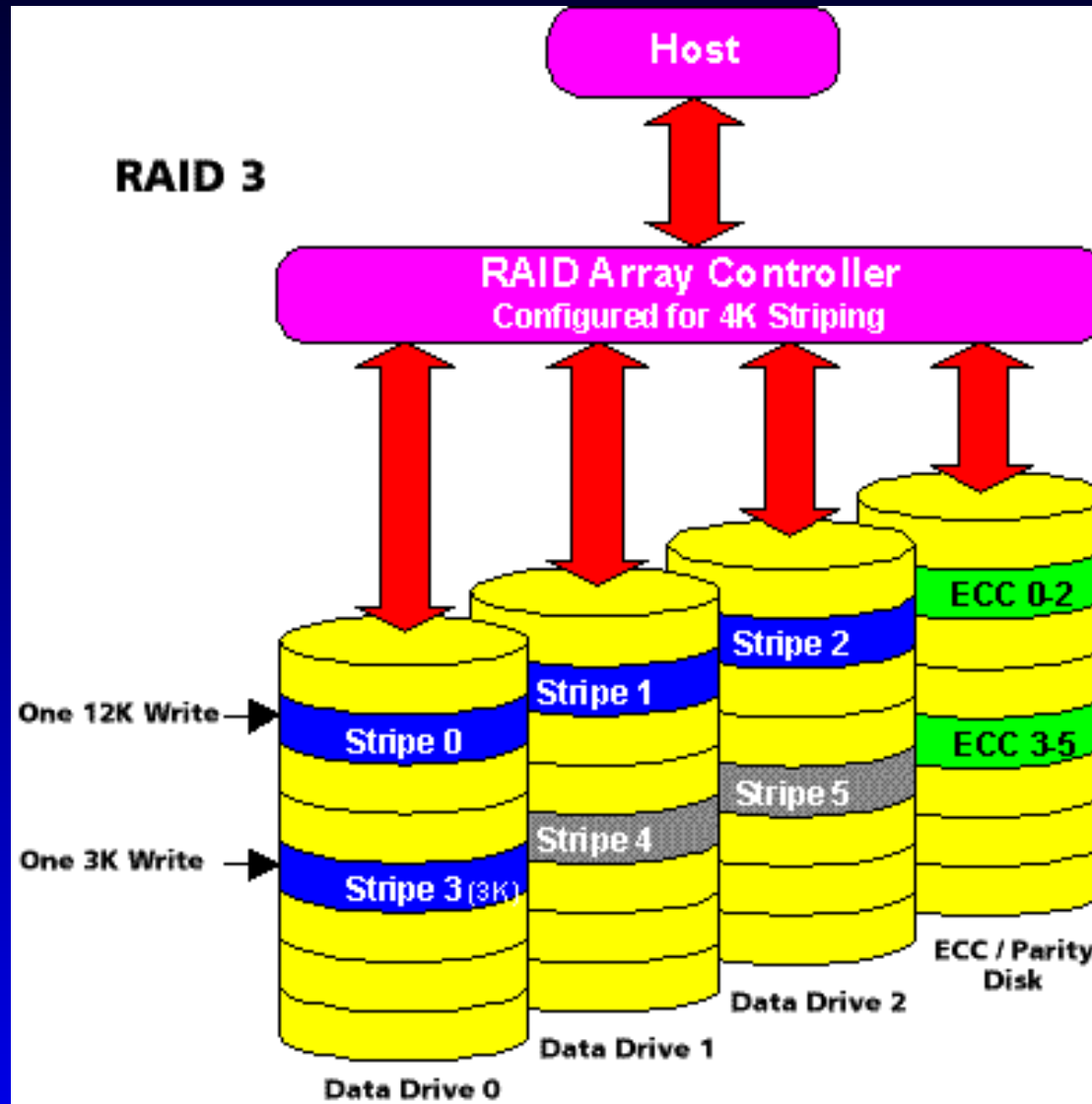
# RAID 1+0

- Striping and mirroring, using an even number of drives ( $N*2$  redundancy)
- This is a very popular combination of two RAID levels, which allows the performance and redundancy benefits of RAID 1 to scale across four or more drives
- Up to half the drives can fail, depending on which drives they are, with no loss in performance
- Another popular combo is RAID 0+1, which can only tolerate up to two failures

# RAID 3

- Striping with dedicated parity drive, using 3 or more drives (N+1 redundancy)
- This level accesses all the drives in parallel, and gives good read performance for sequential access
- The need to calculate parity data can impacts performance
- One drive failure can be tolerated, with a significant performance impact if it is any drive other than the parity drive

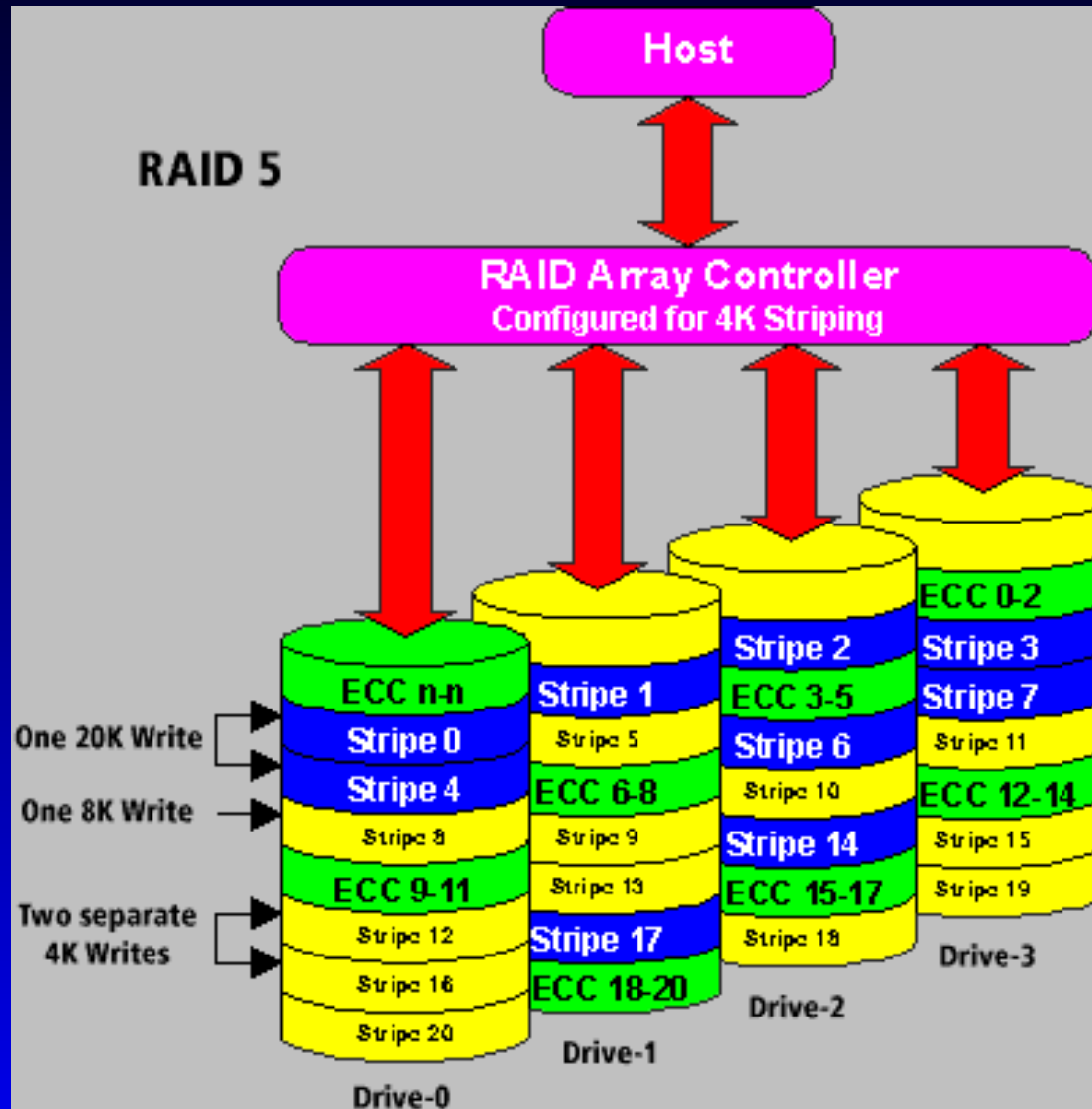
# RAID 3



# RAID 5

- Striping with distributed parity, using 3 or more drives (N+1 redundancy)
- This level allows random access to all the drives, and thus gives good read performance for general purpose applications
- Parity calculation and updates do present a severe impact on write performance, especially with small writes, though a sufficiently large write cache can overcome this hurdle
- One drive failure can be tolerated, with a significant performance impact

# RAID 5



# RAID 2 & 4

- These levels are not used by anyone, so they are described here purely for reference
- RAID 2 - Striping using “Hamming code” parity, with several parity drives
  - Requires a large number of disks (i.e. 10+4 or 32+7)
  - Provides protection primarily against the sort of errors now handled internally to hard drives
- RAID 4 - Striping with dedicated parity drive and very large stripe size
  - Similar to RAID 3 in concept

# Questions?

Slides produced with Prosper and L<sup>A</sup>T<sub>E</sub>X.

<http://prosper.sourceforge.net/>