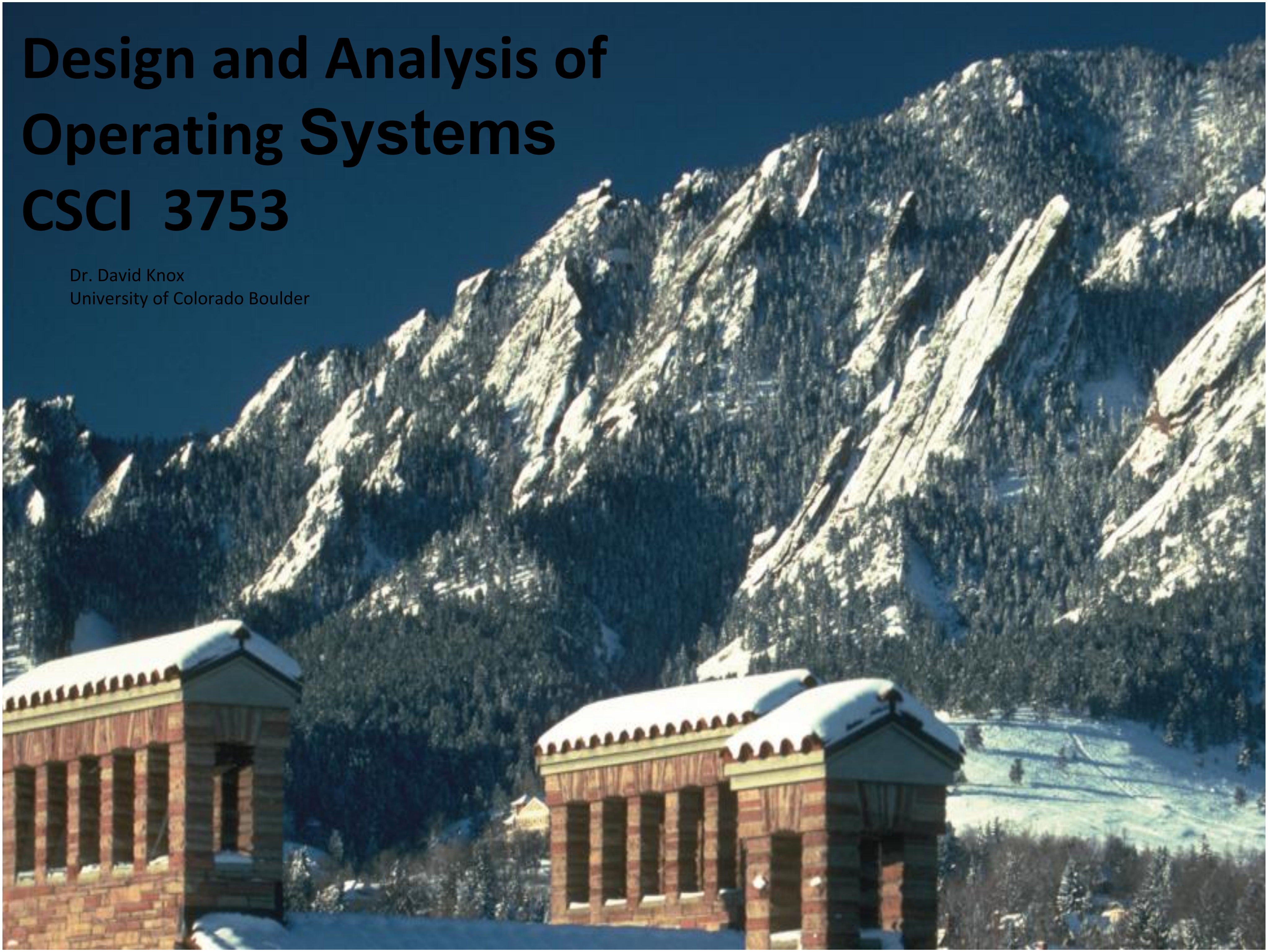
Design and Analysis of Operating Systems CSCI 3753

Dr. David Knox
University of Colorado Boulder



Design and Analysis of Operating Systems CSCI 3753

Redundant Arrays of Inexpensive Disks (RAID)

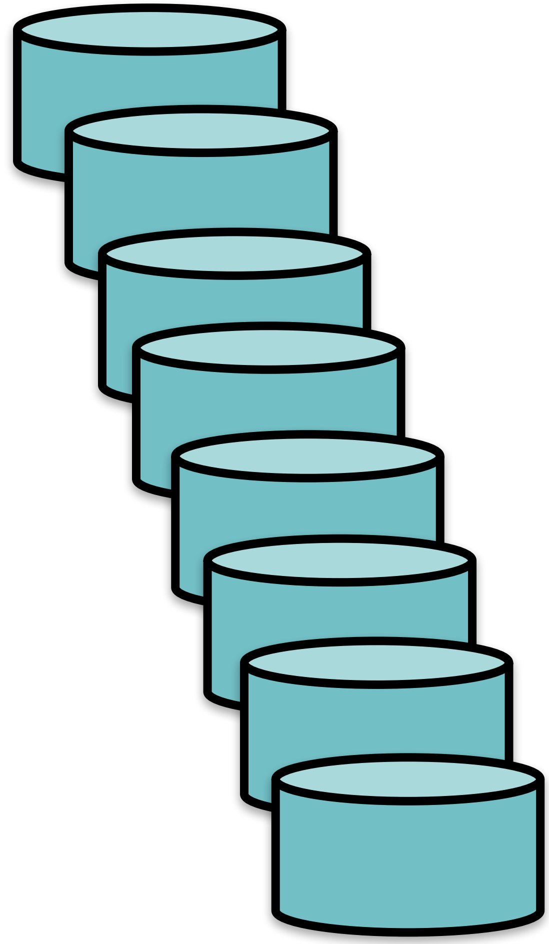
Dr. David Knox

University of Colorado Boulder

Redundant Arrays of Inexpensive Disks (RAID)

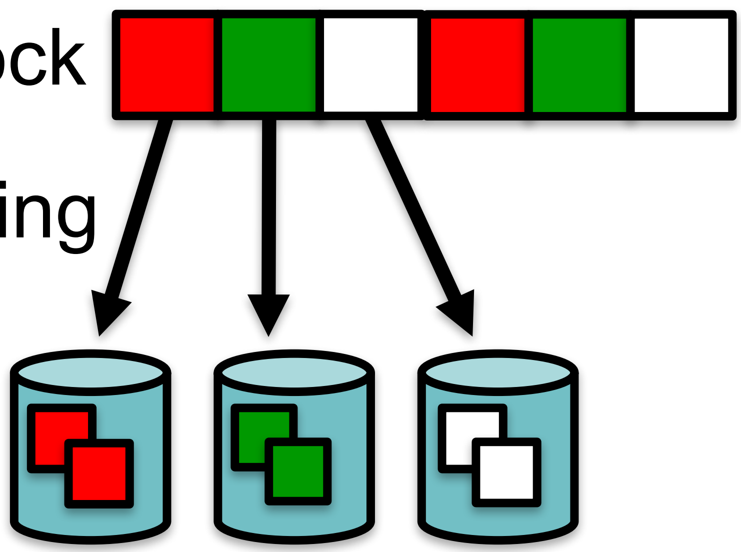
Redundant Arrays of Inexpensive Disks (RAID)

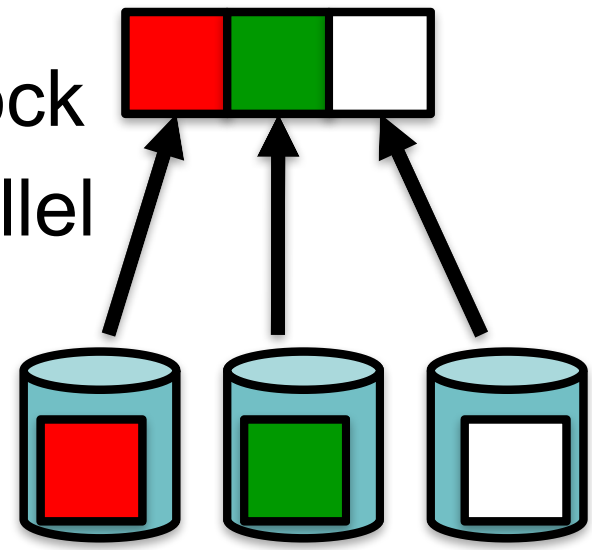
- **Magnetic disks are cheap these days.**
- **Attaching an array of magnetic disks to a computer brings several advantages compared to a single disk:**
  - Faster read/write access to data by having multiple reads/writes in parallel.
    - Data is striped across different disks, e.g. each byte of an 8-byte word is striped onto a different disk
  - Better fault tolerance/reliability
    - if one disk fails, a copy of the data could be stored on another disk – redundancy to the rescue!



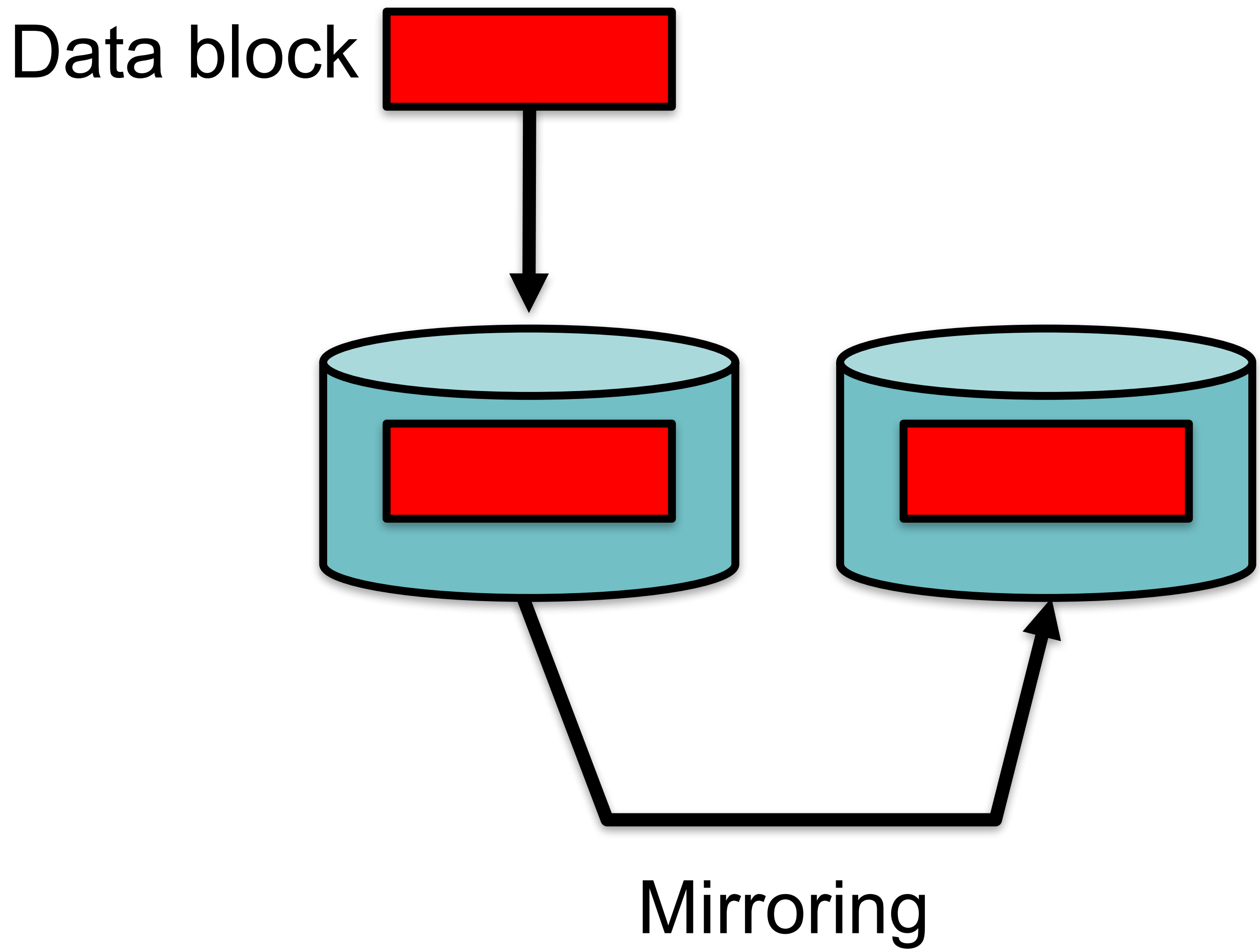
RAID0

- **RAID has different levels with increasing redundancy**
  - RAID0 = data striping with no redundancy





RAID1 = mirror each disk



Get redundancy, but not parallel performance speedup of RAID0.

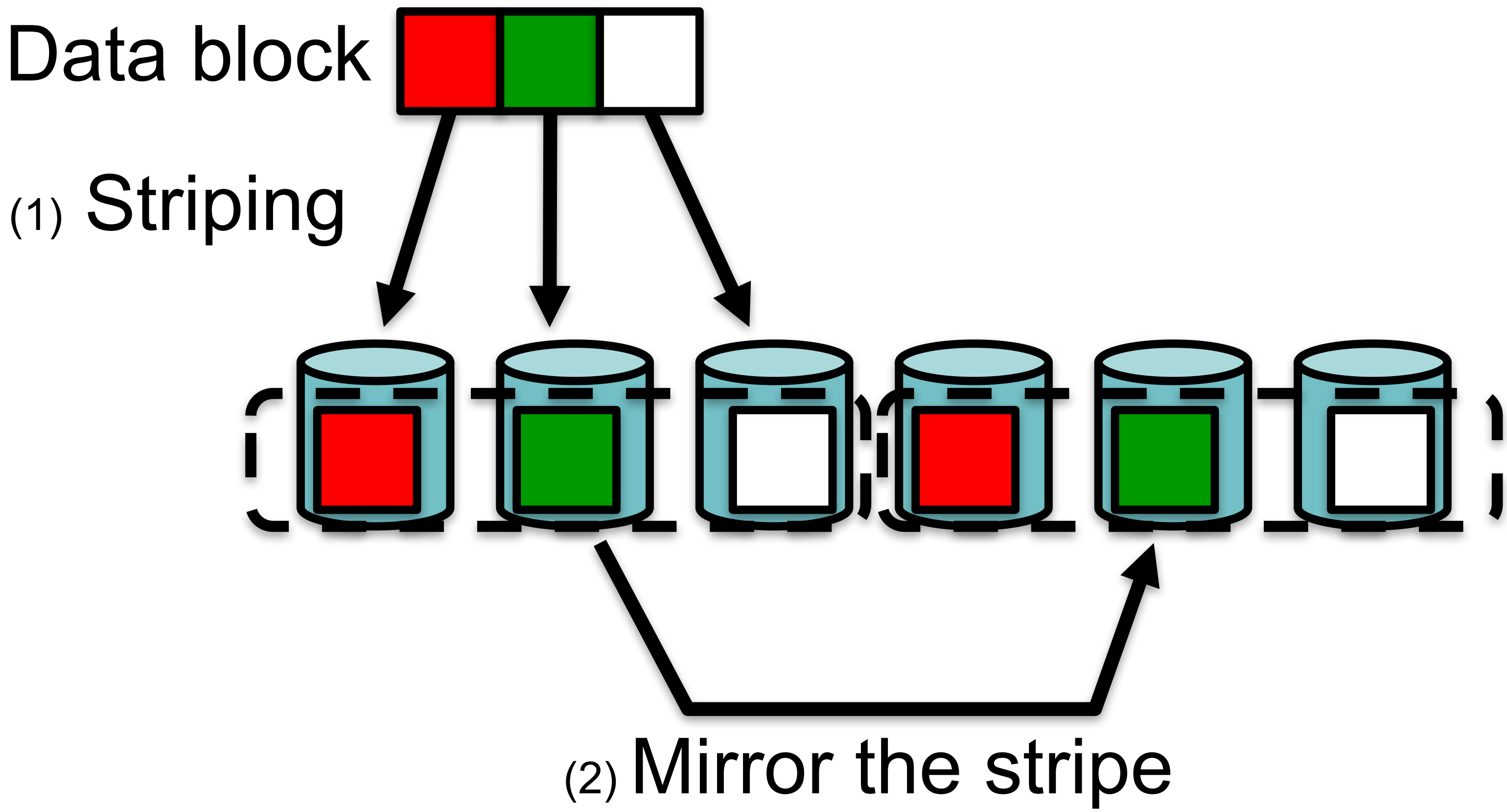
Combine RAID0 with RAID1 (next slide)

Can get limited read speedup by submitting the read to all mirrors, and taking the 1st data result

Twice the delay on synchronous writes, but OS can mask this by delaying writes

RAID0+1

- RAID0+1

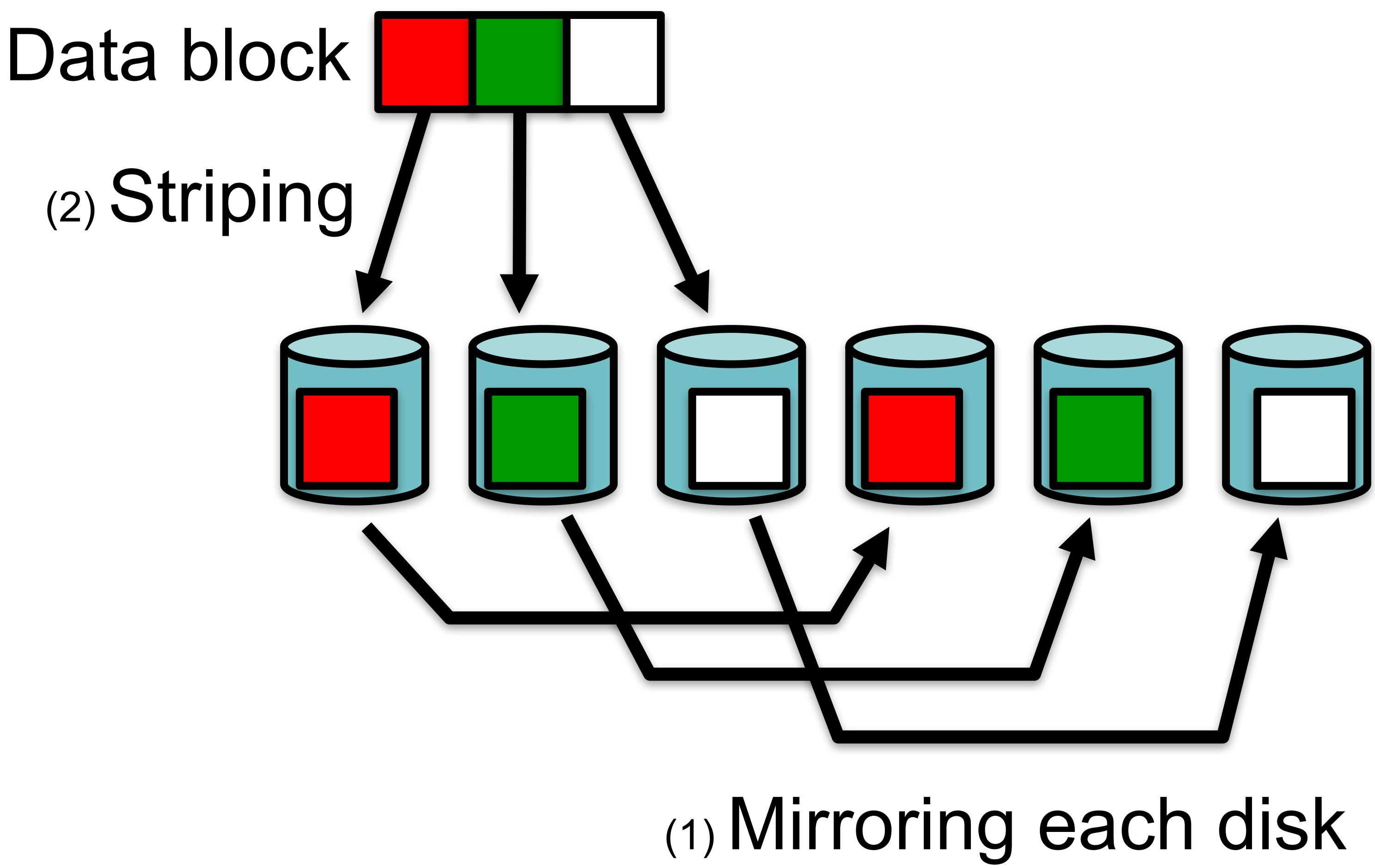


RAID 0+1 = RAID 0 data striping for performance + RAID 1 mirroring of stripes for redundancy

Note: if any one disk fails in the primary stripe, and any one disk fails in the second stripe, then the entire data block is lost

RAID1+0

- RAID1+0, aka “RAID 10”



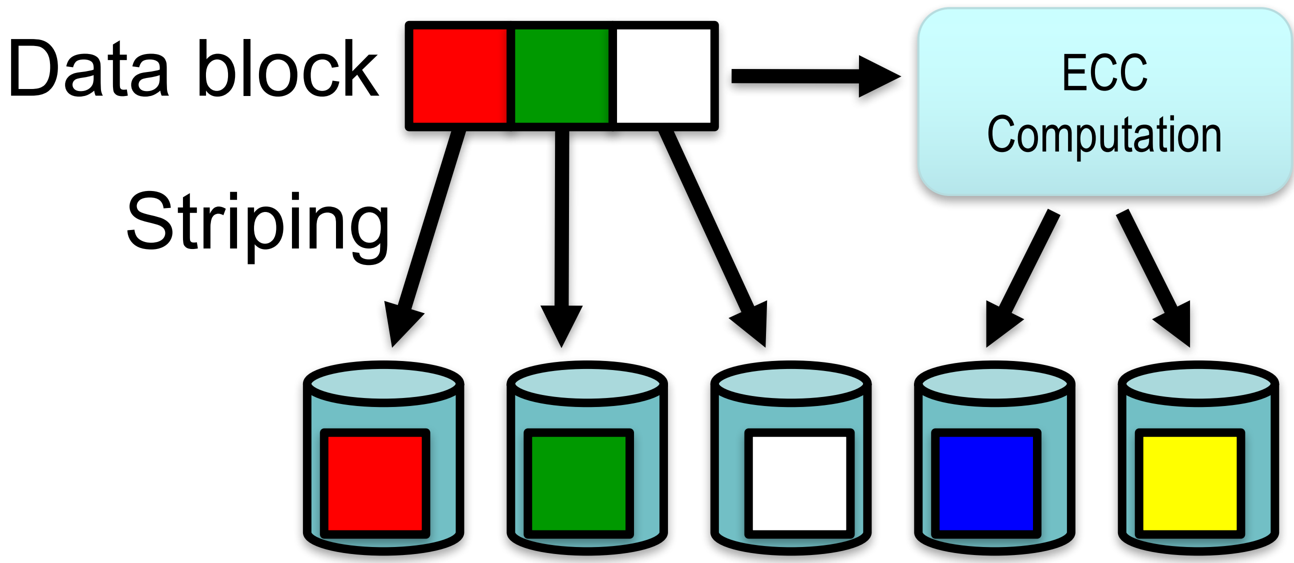
RAID 1+0 = mirror each disk then stripe.

Any two disks may fail, and the data can still be retrieved, unless the two disks mirror the same data.

Thus, RAID 1+0 is more robust than RAID 0+1.

RAID2

- RAID2 = put Error Correction Code (ECC) bits on other disks

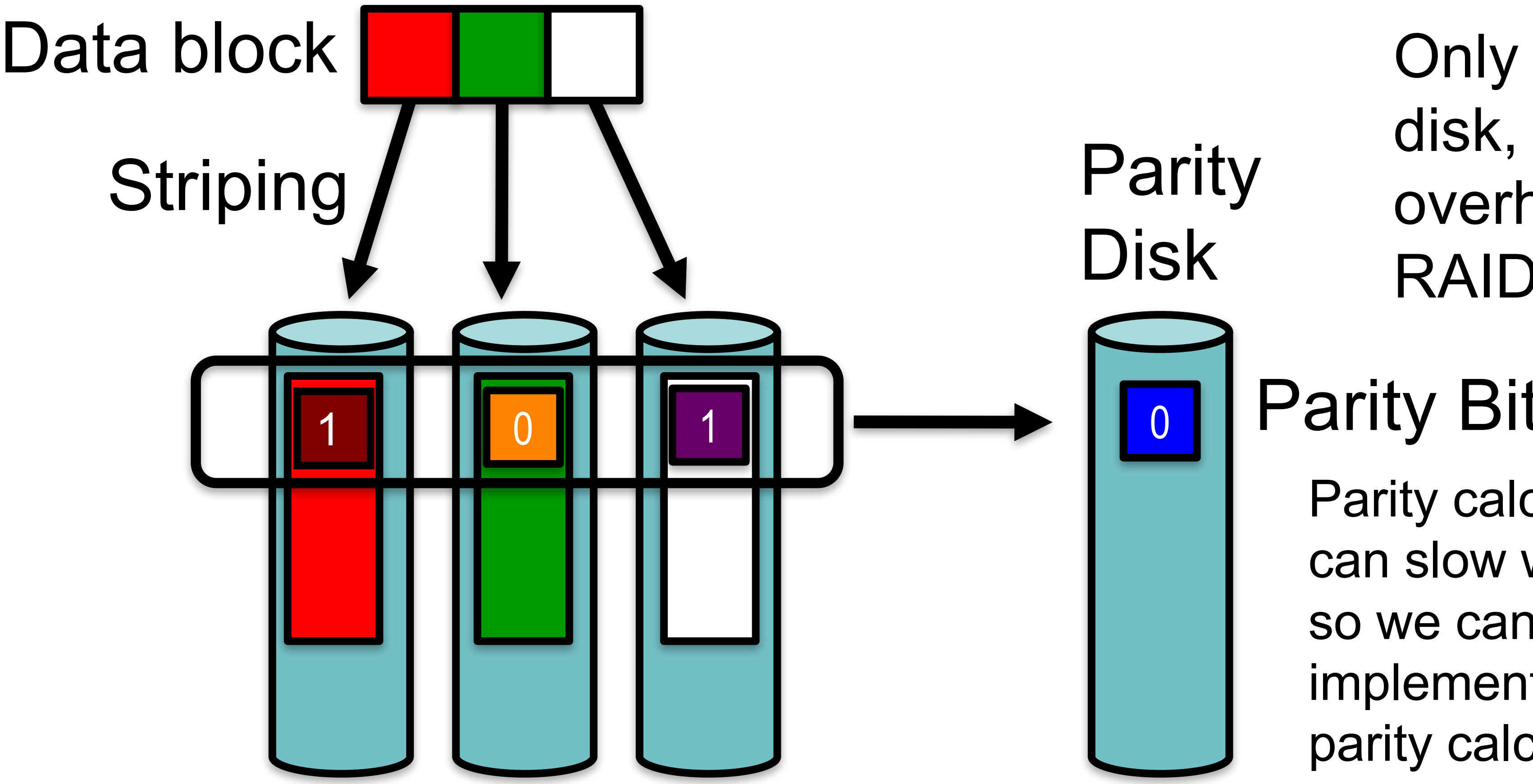


Error correction codes are more compact than just copying the data, and provide strong statistical guarantees against error

e.g. a crashed disk's lost data can be corrected with the redundant data stored in the ECC disks

RAID3

- RAID3 = bit-interleaved parity: for each bit at the same location on different disks, compute a parity bit, that is stored on a parity disk

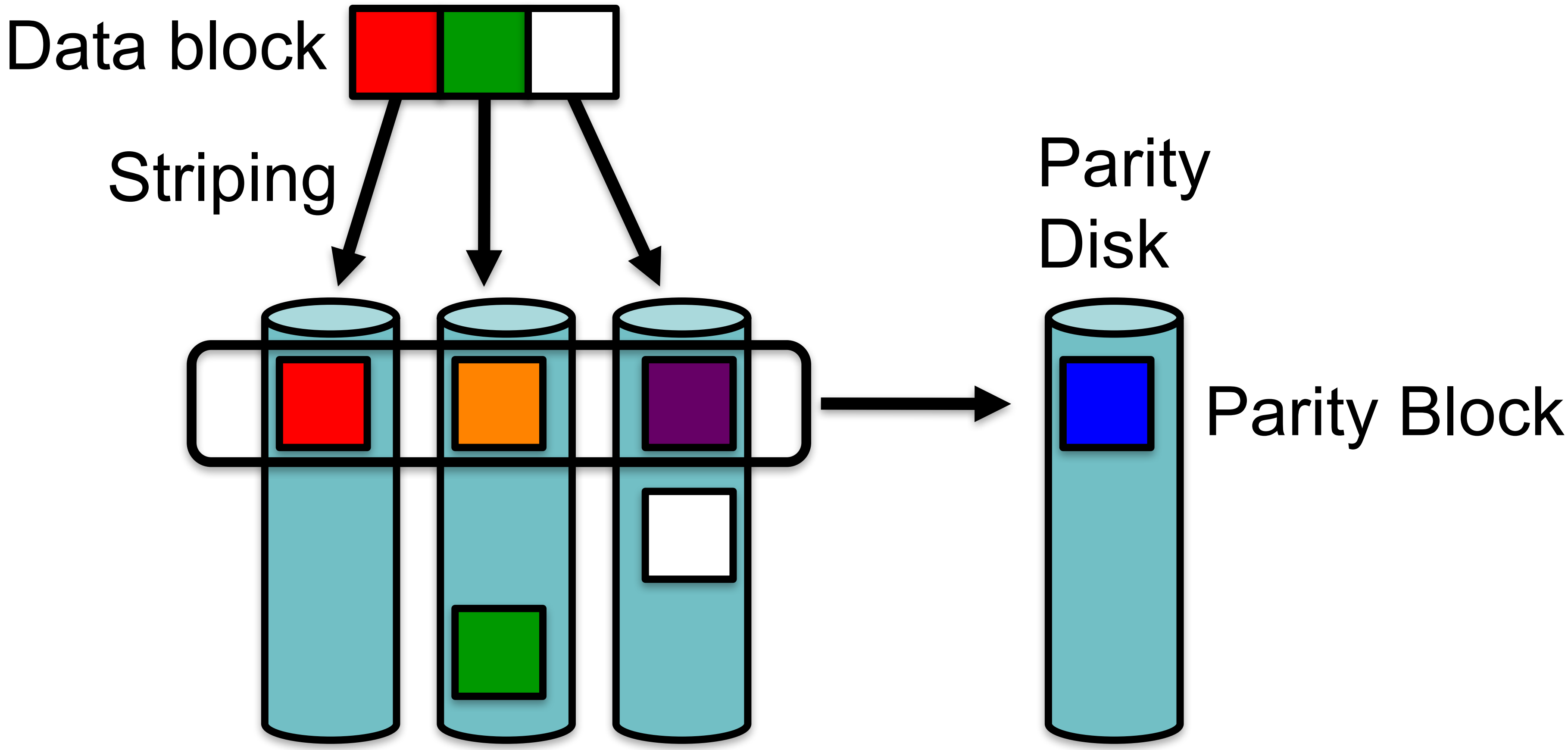


Only 1 extra disk, so less overhead than RAID2

Parity calculation can slow writes, so we can implement parity calculation in hardware

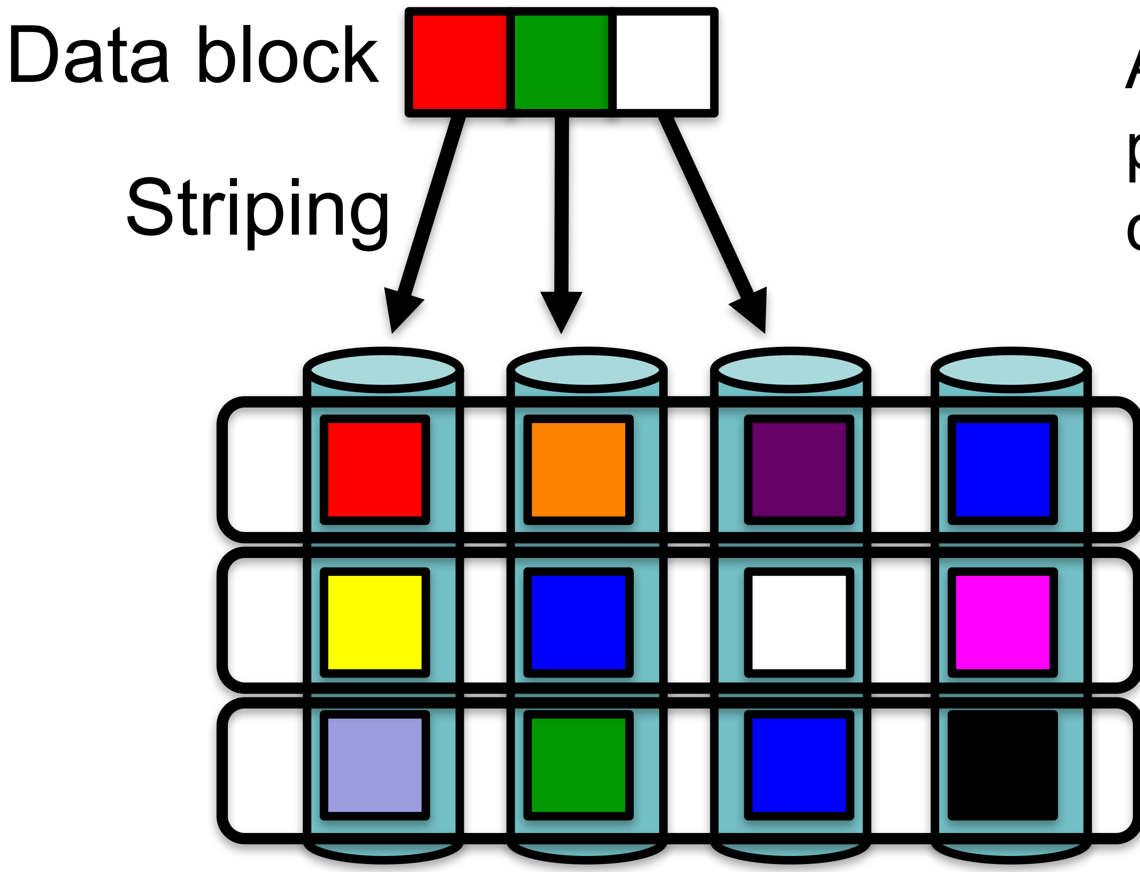
RAID4

- RAID4 = block-interleaved parity: for each block at the same location on different disks, compute a parity block, that is stored on a parity disk

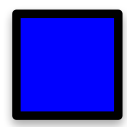


RAID5

- RAID5 = block-interleaved *distributed* parity: spread the parity blocks to different disks, so every disk has both data and parity blocks



Avoids overuse of RAID4's parity disk, and is the most common parity RAID system

 = Parity Blocks

RAID6 replaces parity blocks with stronger Reed-Solomon ECC for multiple disk failures

RAID

- **Implementations of RAID**
  1. Plug in disks to your computer bus and implement RAID management in software in OS kernel
     - In Linux, use the mdadm package to manage RAID arrays
  2. Plug in disks to your computer's I/O bus, and an intelligent hardware RAID controller recognizes them and integrates them into a RAID system automatically
  3. Plug in disks to a RAID array, which is a stand-alone hardware unit with a RAID controller that looks like one physical device, e.g. SCSI, to your computer bus
     - File system doesn't have to know about RAID to use and benefit from this RAID disk array

RAID Implementation

Three methods:

Software Based,

Firmware Based,

Hardware Based

1. **Software Based: Plug in disks to your computer bus and implement RAID management in software in OS kernel**
   - The RAID software layer sits above the device drivers
   - Cost effective and easy to implement, but not as efficient
   - In Linux, use the mdadm package to manage RAID arrays
   - Cannot boot from RAID

RAID Implementation

2. **Firmware/Driver Based: Plug in disks to your computer's I/O bus, and an intelligent hardware RAID controller recognizes them and integrates them into a RAID system automatically**

   - Standard disk controller chips with special firmware and drivers
   - RAID instructions are stored in the firmware of the device
   - Can boot from RAID: During startup/boot, the RAID is essentially kick-started by the firmware

RAID Implementation

3. **Hardware Based: Plug in disks to a RAID array, which is a stand-alone hardware unit with a RAID controller that looks like one physical device, e.g. SCSI, to your computer bus**

   - File system doesn't have to know about RAID to use and benefit from this RAID disk array
   - Expensive but highly efficient

Design and Analysis of Operating Systems CSCI 3753

Dr. David Knox
University of Colorado Boulder