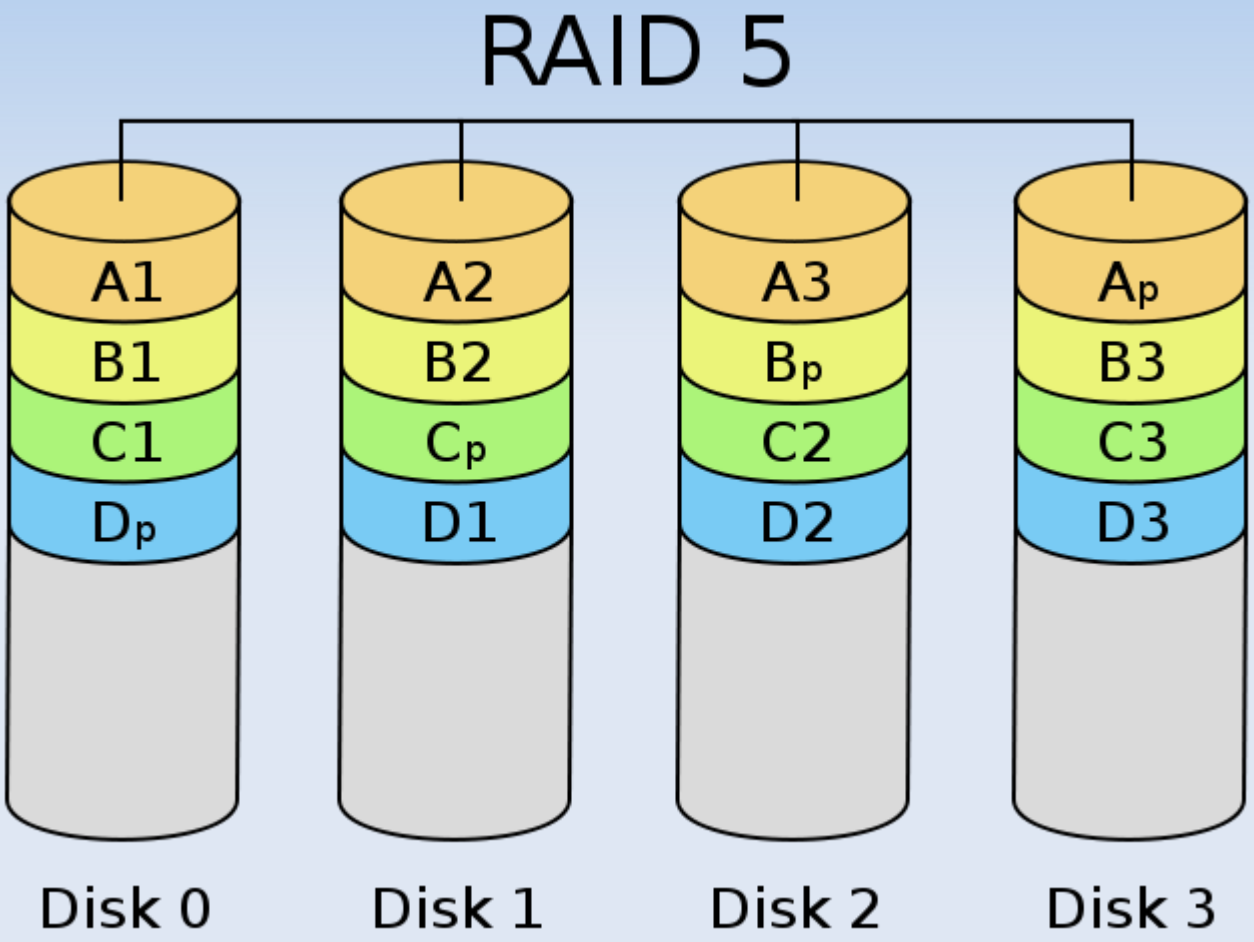
Striping in a RAID level 5 Disk array

Peter M. Chen and Edward K. Lee

Presenter: Luis Useche

About RAID


RAID 5
A1
B1
C1
Dp
A2
B2
Cp
D1
A3
Bp
C2
D2
Ap
B3
C3
D3
Disk 0
Disk 1
Disk 2
Disk 3

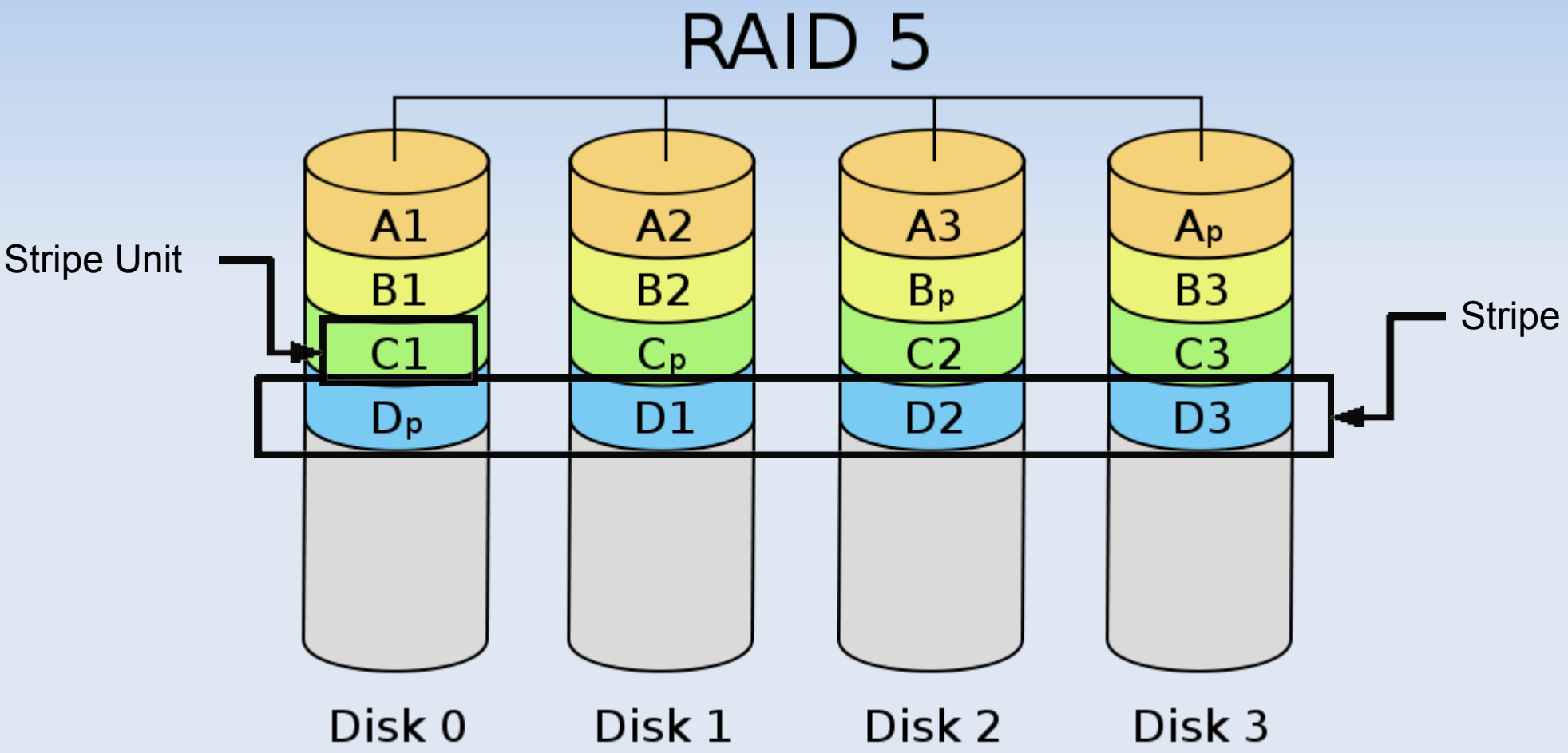
Motivation

- Redundant arrays improve reliability and performace.
- Striping unit greatly affect the RAID performance.
- Studies for striping unit RAID-0 made but not for RAID-5
- RAID-5 is very used.

Paper Goals

- Check parameters that affect the stripe size.
- Create simple rules to determine the stripe size.
- Quantify how # of disks affect the stripe size.

RAID Composition

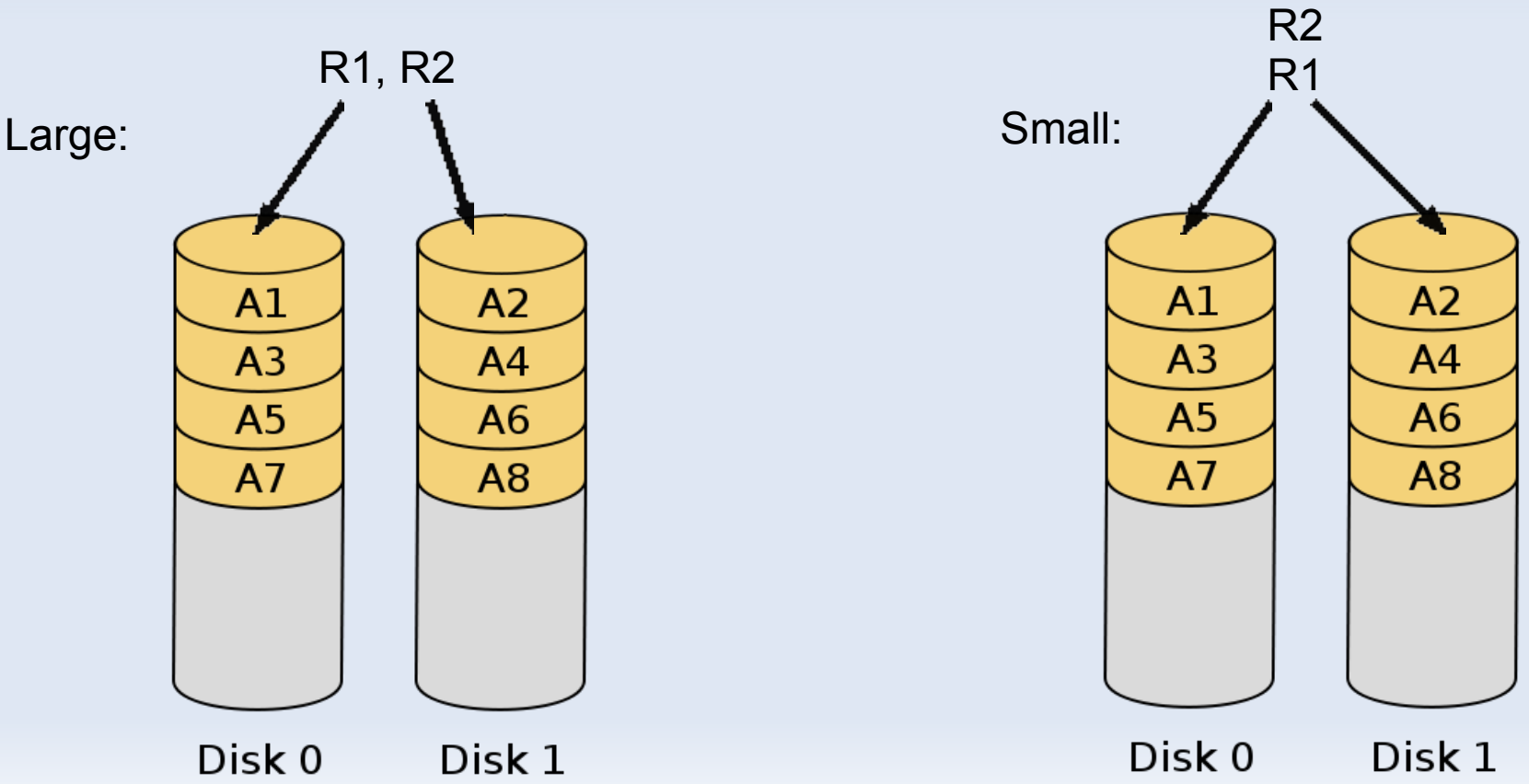


Definitions

- Stripe Unit
  - Large: One file located in few disks.
  - Small: One file distributed in several disks.
- Parallelism: number of disks that are servicing one disk.
  - Large: increment the bandwidth usage BUT spend more time in positioning.
- Concurrency: number of pending I/Os in the system.

Parallelism vs Concurrency

- Strip Unit
  - Large: support more concurrency.
  - Small: increase parallelism.



Strip Unit and Positioning time

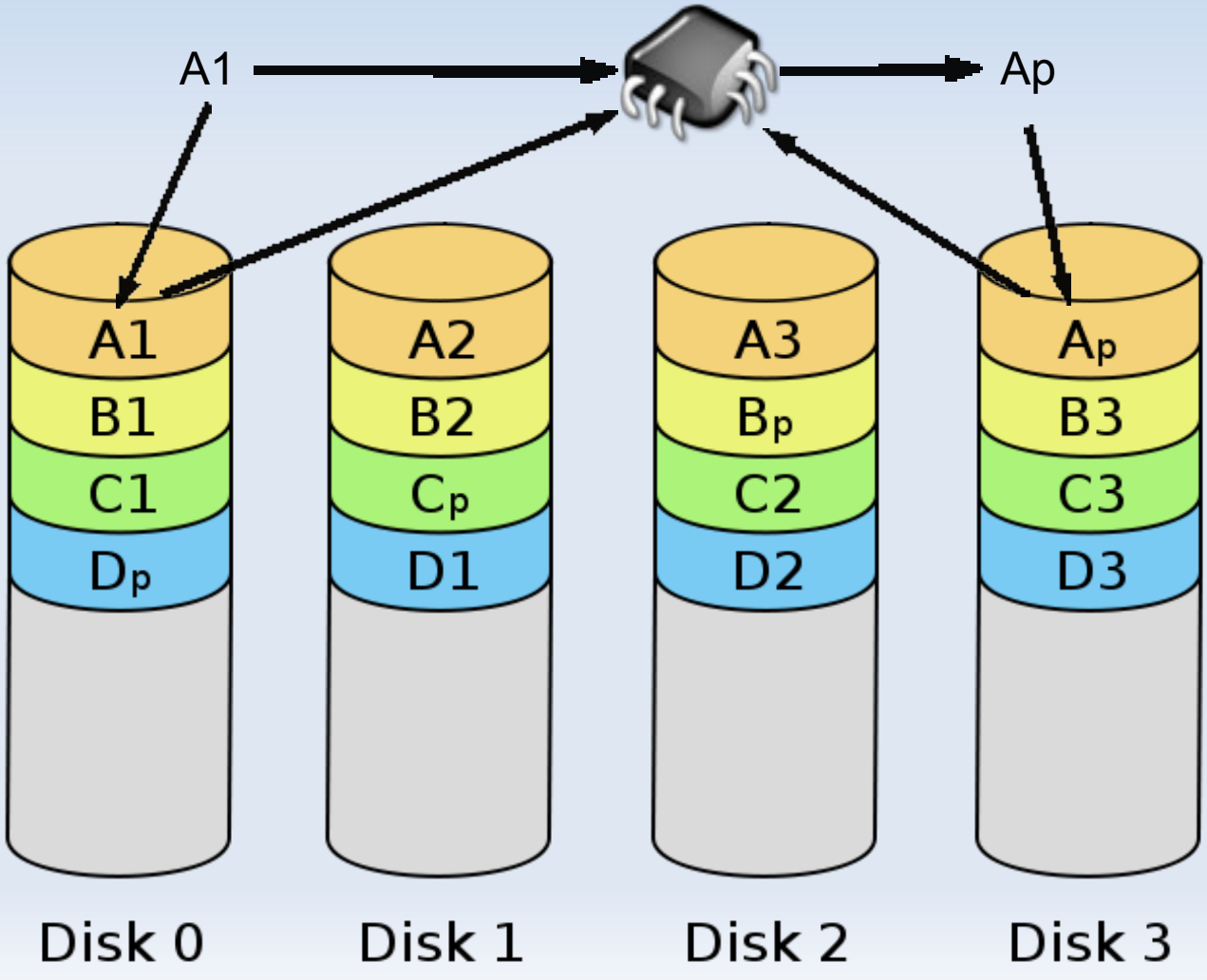
- Large stripe size increase the data transfered befor repositioning.
- However, requires more concurrency to make usage of all disks.

RAID 5 operations

- Read: Same as RAID-0
- Write: Updating of the parity block needed. Three types of writes:
  - Full Stripe
  - Reconstruct
  - Read-Modify

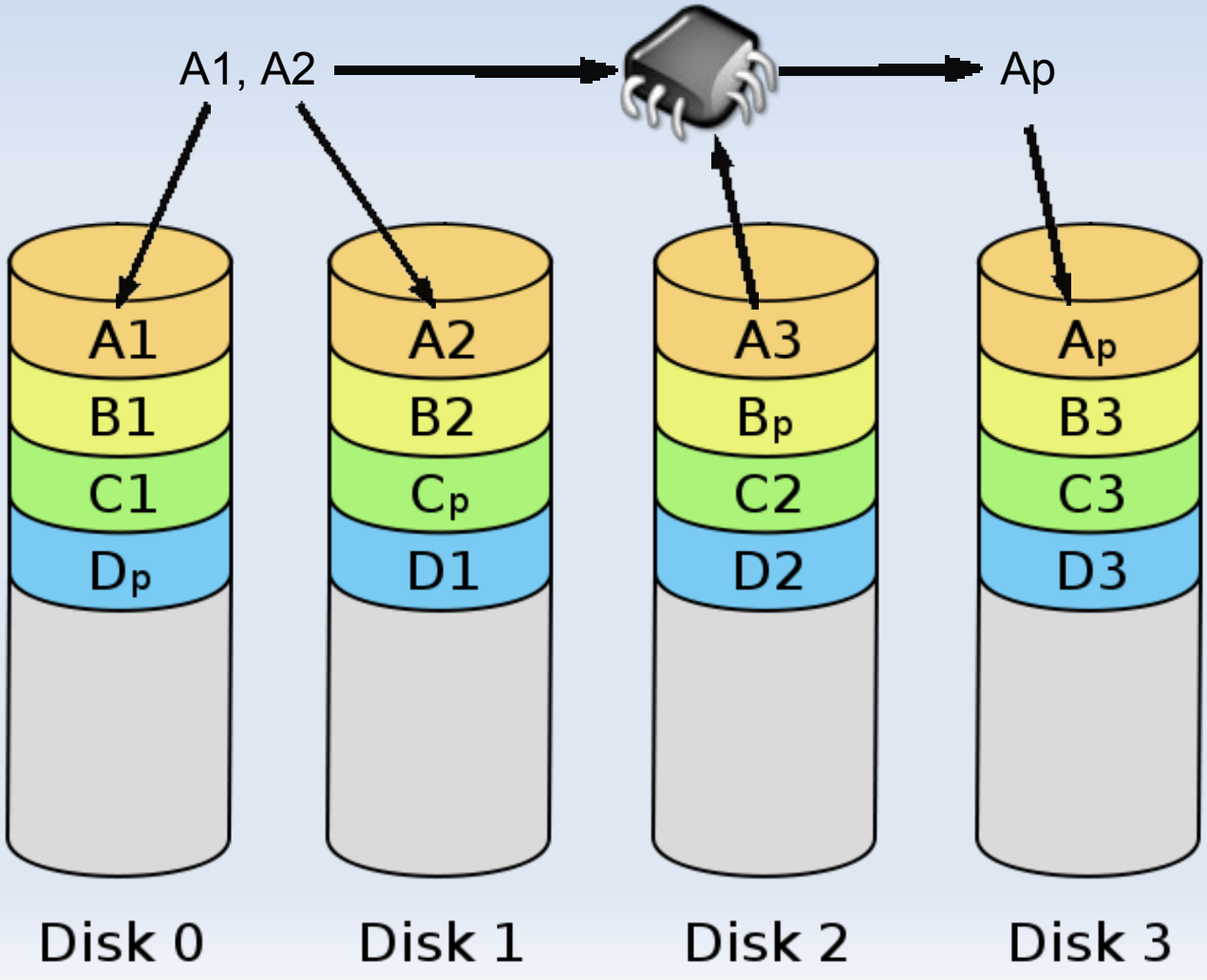
Read-modify Write

- Read only the block to be updated and the parity. Then write both blocks.



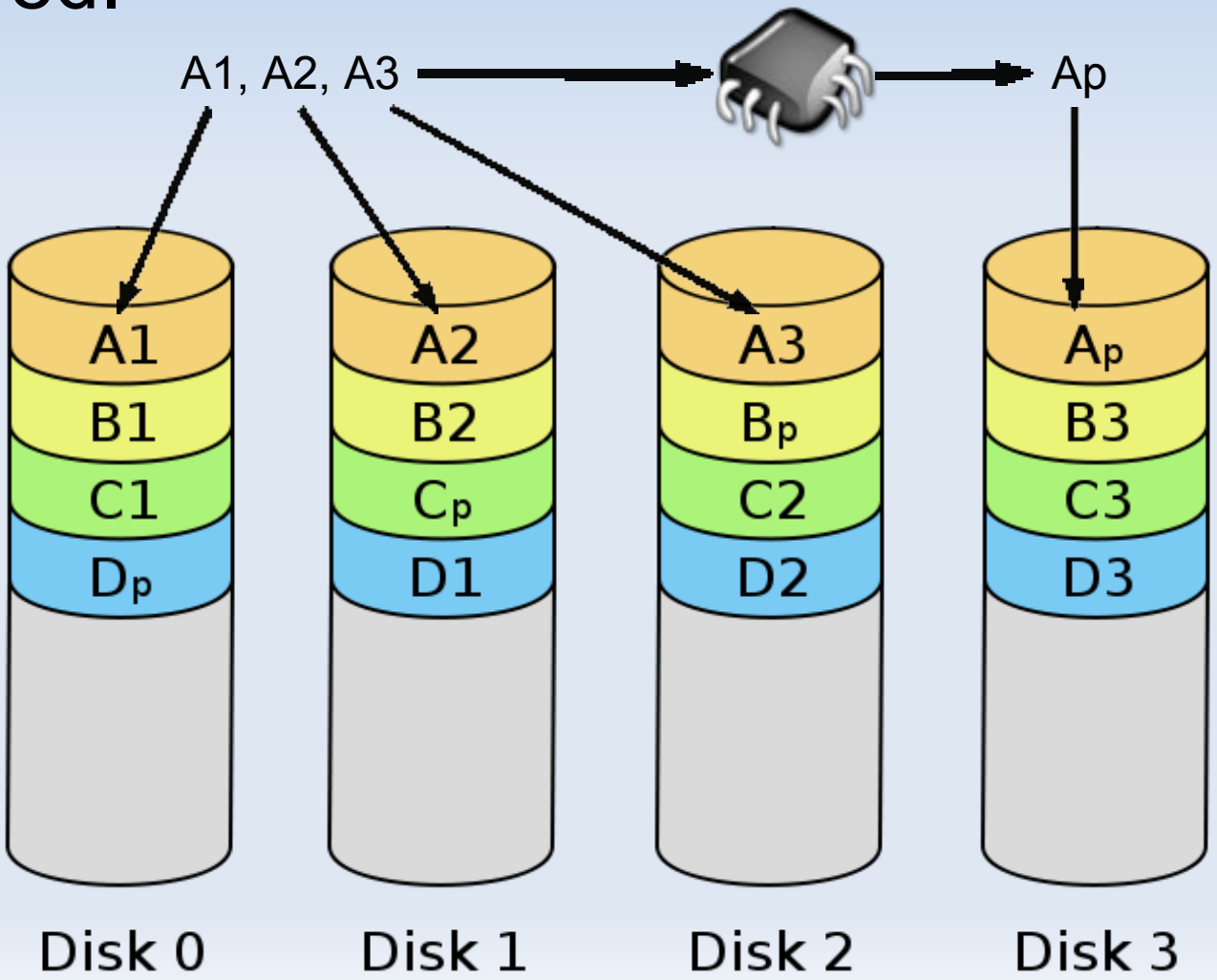
Reconstruct Write

- Read the rest of the stripe to calculate the new parity.



Full Stripe Write

- The whole stripe is written, no extra operation is required.



Reconstruct vs. Read-Modify

- Note that the write could be done with Reconstruct or Read-Modify.
- Paper assumption: the raid software choose the one that minimize the I/Os

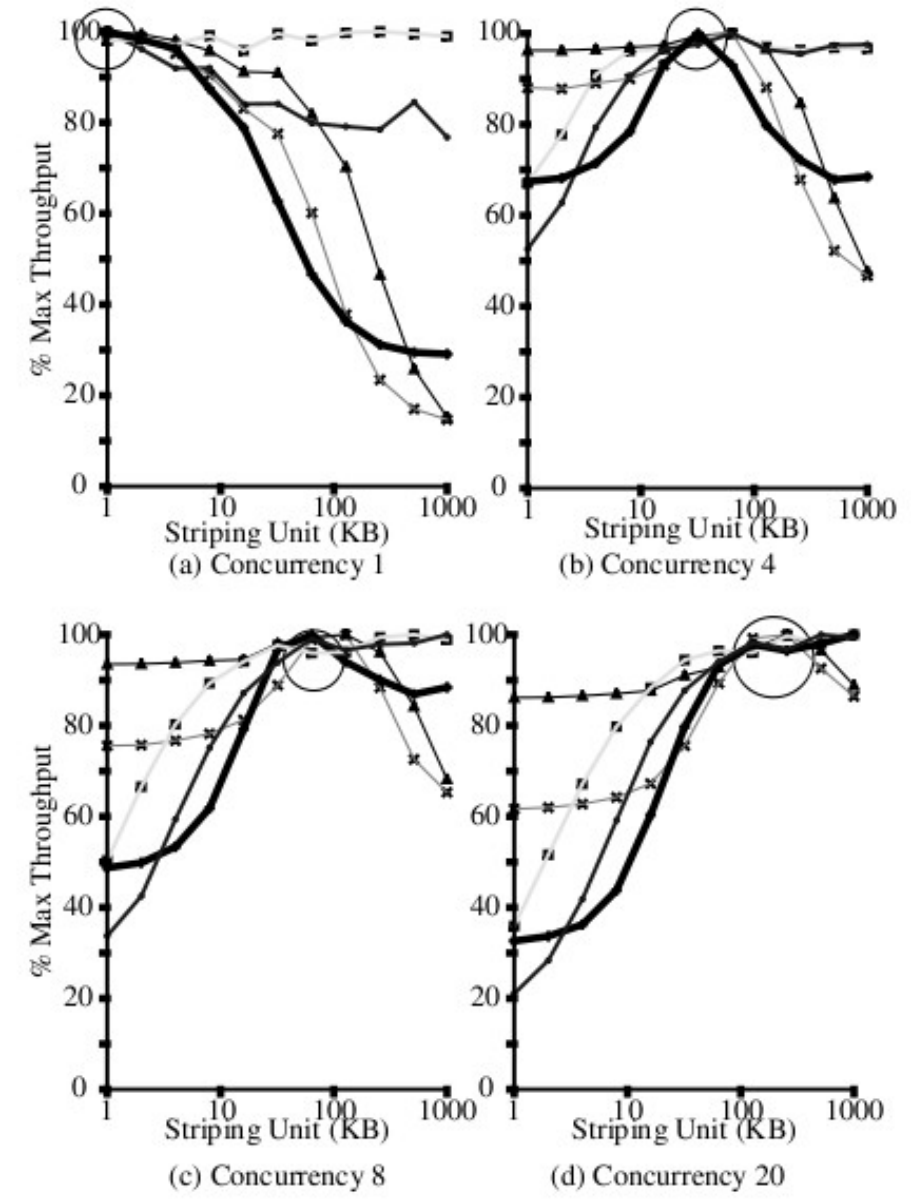
Experimental Setup

- Simulation (raidSim)
- Max capacity set to 300MB.
- 3 different disks modeled.
- Workload: three parameters
  - Concurrency
  - Request Size
  - Read/Write mix.

Experiment Parameters

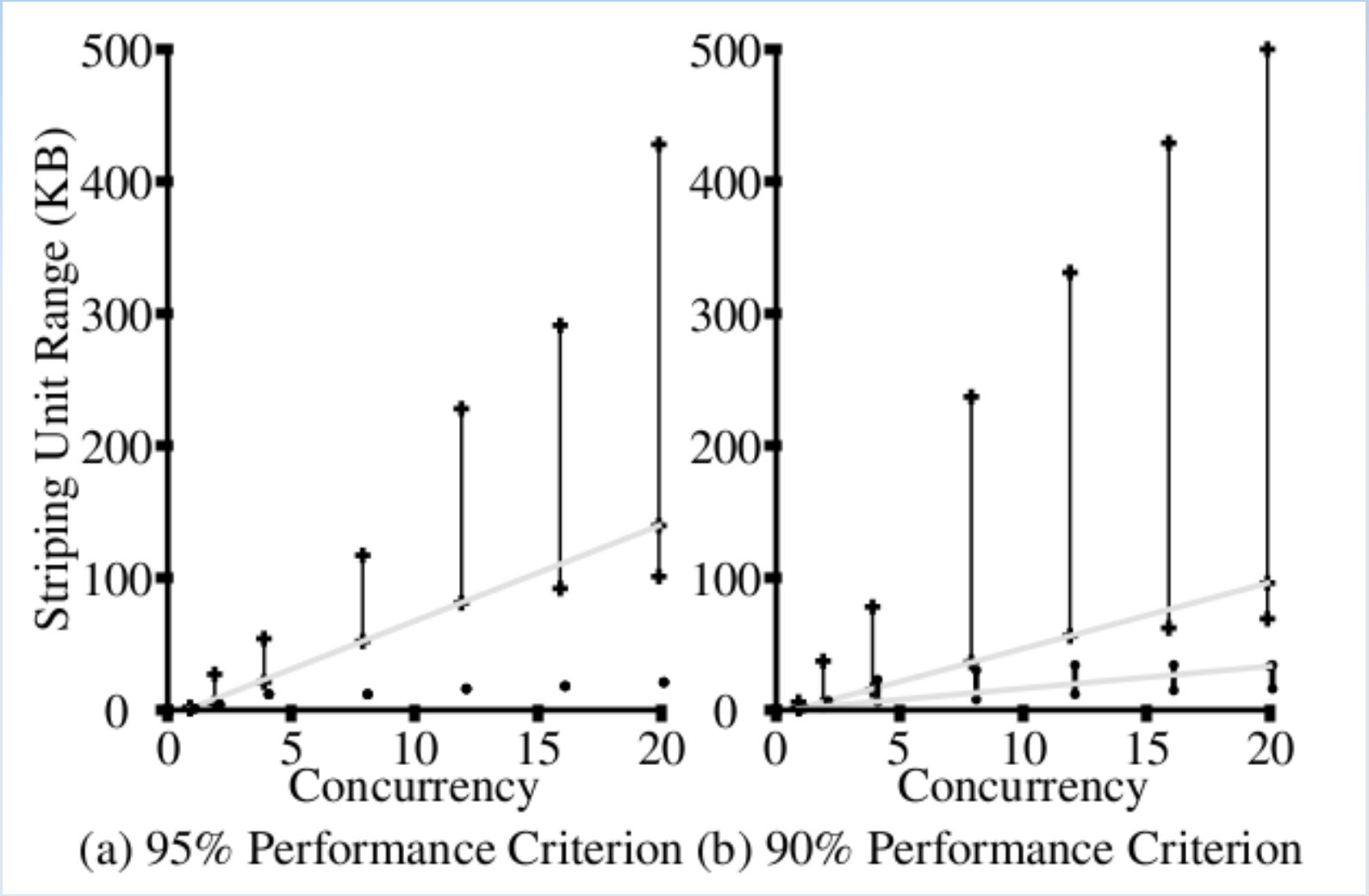
- Concurrency: number of processes issuing synchronous requests. values: 1-20
- Request Sizes:
  - exp4KB: exponential distribution, mean 4KB
  - exp16KB: exponential distribution, mean 16KB
  - norm100/400/1500KB: normal distribution, mean of 100, 400 and 1500 KB respectively.
- Request position uniformily distributed.
- Metrics: Percentage of the maximun throughput in the array.

Choosing striping unit, Reads



- Concurrency more imporant than request size.
- Knowin concurrency, we can guarantee 95% of max throughput.

Concurrency vs Stripe size



(a) 95% Performance Criterion (b) 90% Performance Criterion

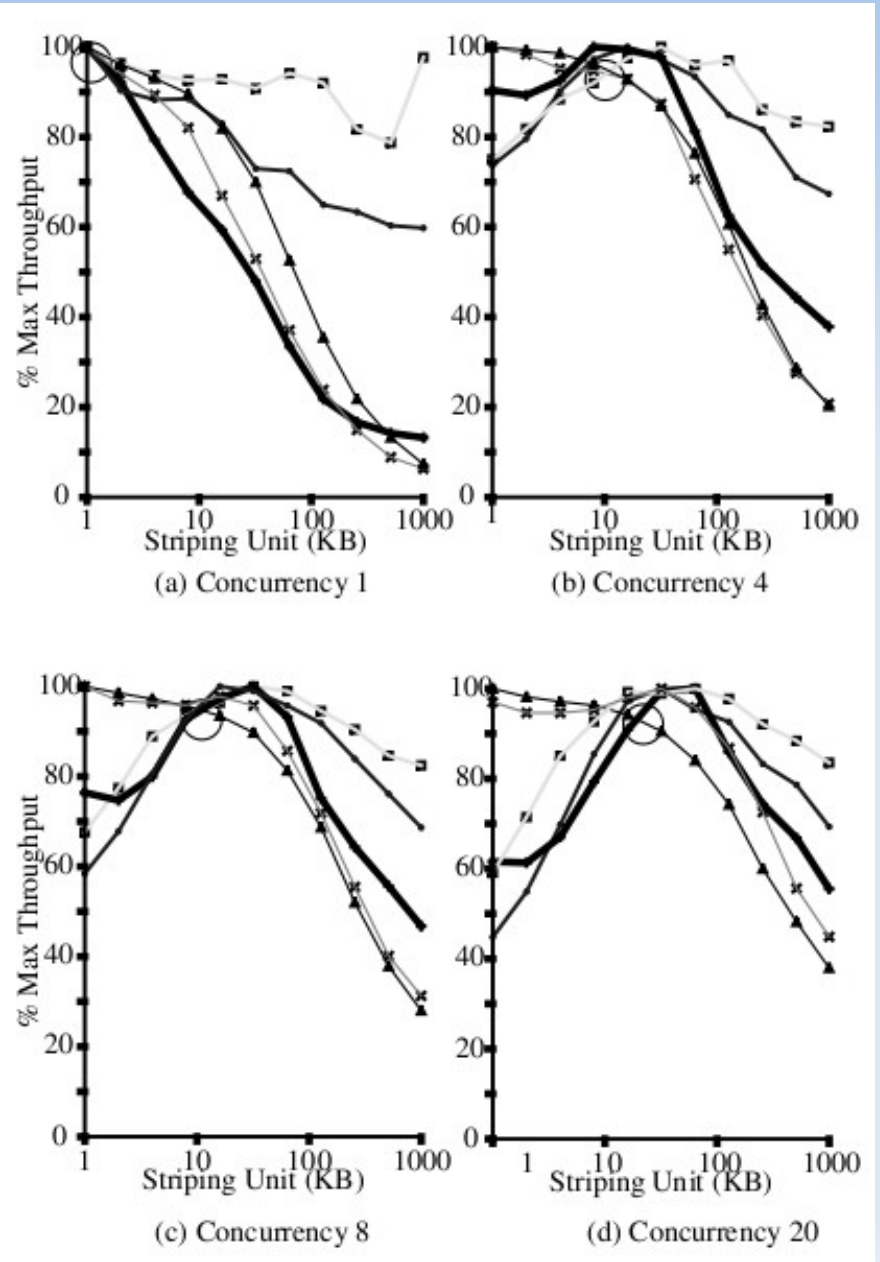
Finding a stripe size function

- Stripe Size could be expressed as a linear function of the concurrency.

$$\begin{aligned} f(x) &= a \cdot x + b \\ f(concurrency) &= Slope \cdot (concurrency - 1) + 1 \end{aligned}$$

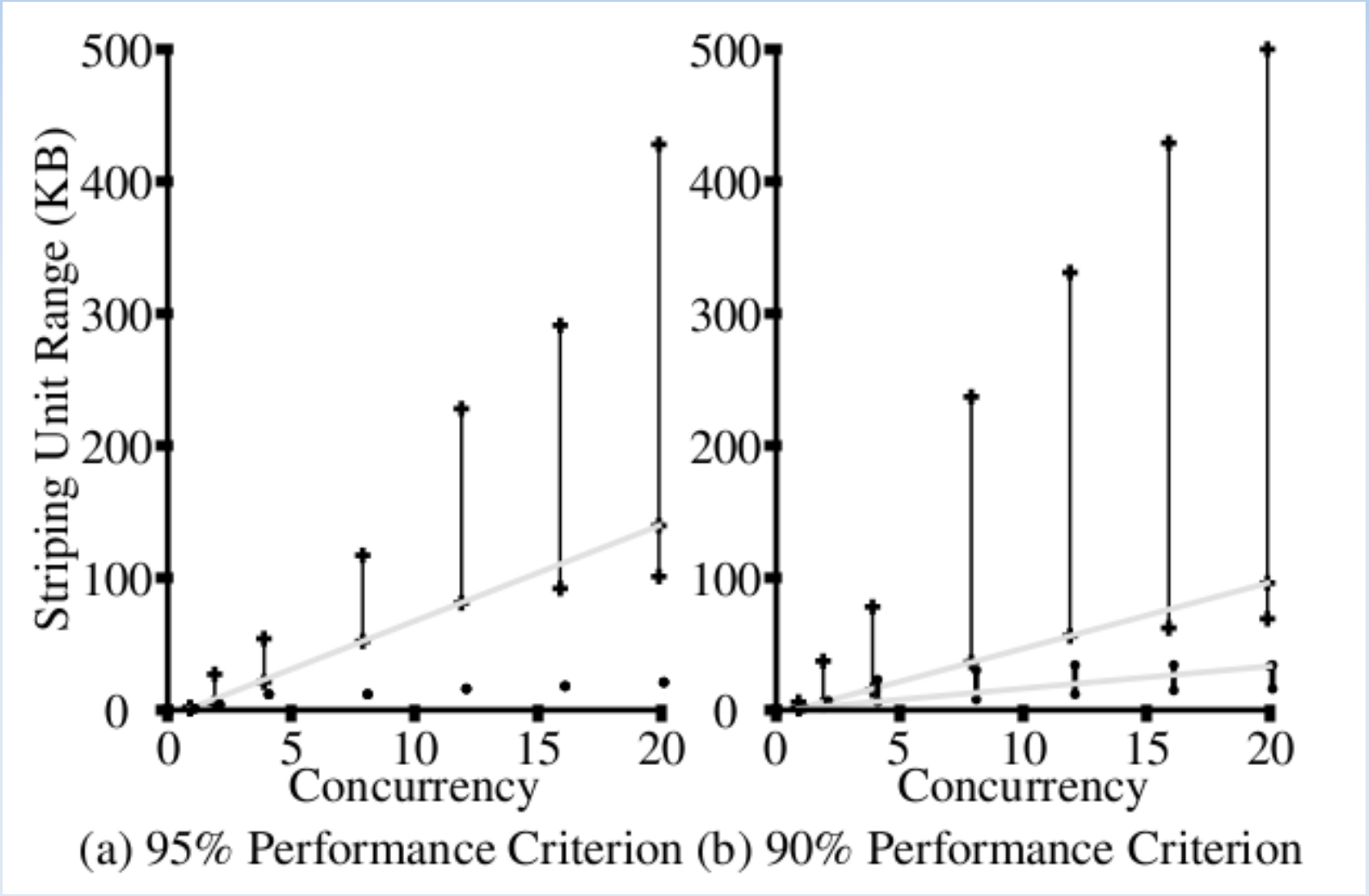
- Where Slope is dependent on the disk characteristics and number of disks in the array.
- Superposing all reads figure, we could find a value that guarantee 70% of performance.

Choosing striping unit, Writes



- Same pattern as Reads, concurrency is very important.
- However, for writes stripe units tend to be smaller.

Concurrency vs Stripe size



(a) 95% Performance Criterion (b) 90% Performance Criterion

Finding a stripe size function

- For writes, the stripe size could be expressed as a function as well.

$$\begin{aligned} f(x) &= a \cdot x + b \\ f(concurrency) &= Slope \cdot (concurrency - 1) + 1 \end{aligned}$$

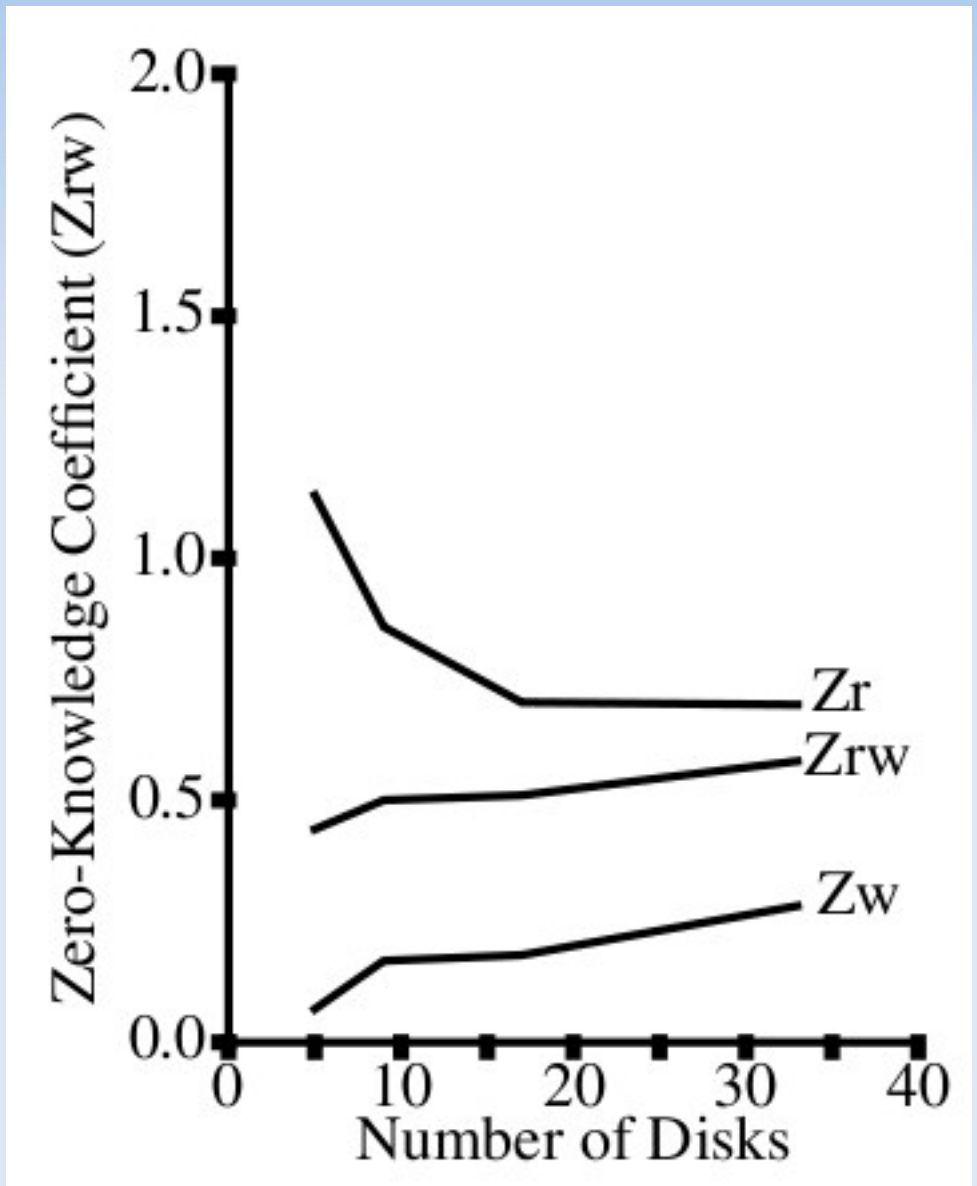
- However, now the slope is different.
- Increase slower than reads.
- Again, we could garantee 73% with no knowledge of the concurrency.

Without information?

- What about read/writes mix?
- Superimpose all figures!
- It is possible to find a stripe size that guarantees 60% of the max throughput.

And the number of disks?

- Reads: More disks, smaller striping size to use all disks.
  - Hence Zr decreases
- Writes: Less disks, smaller striping size to ensure full stripe writes.

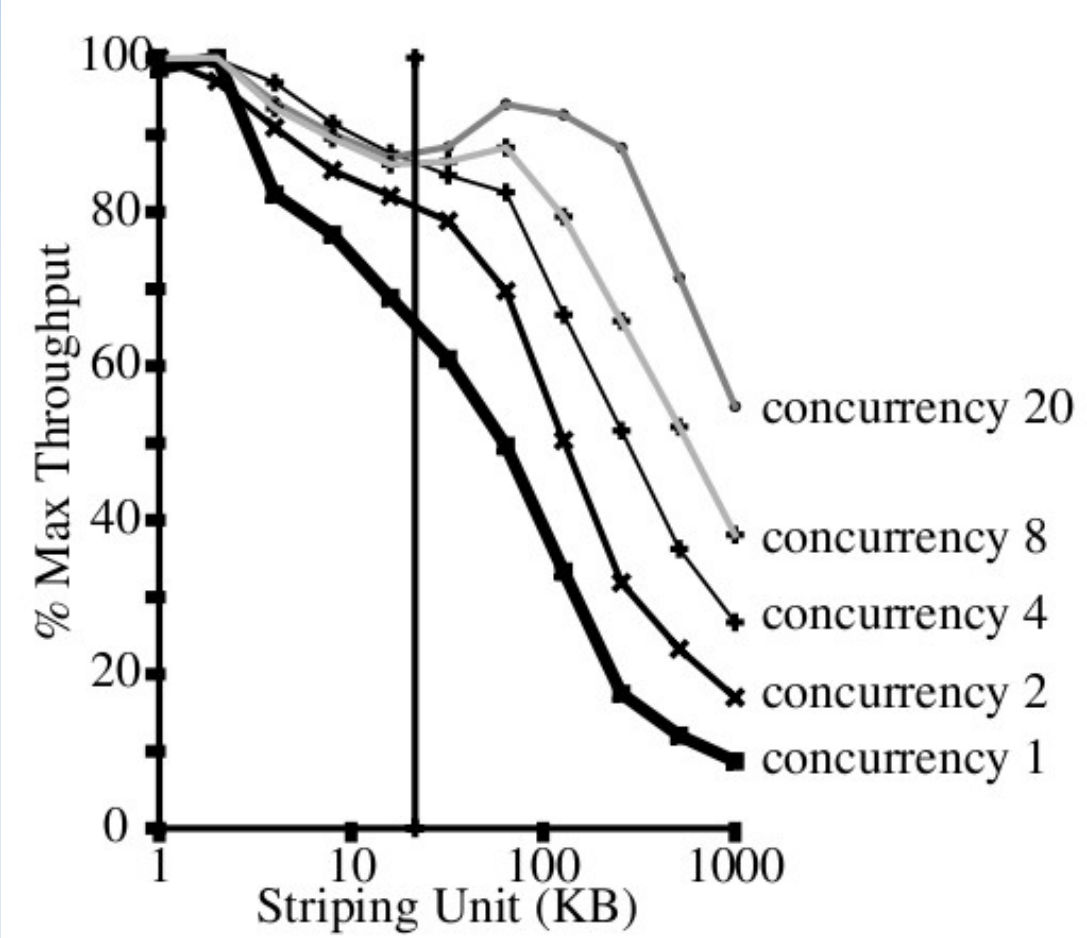


Verification

- They verified the results with real traces.
- Scientific applications
- Replay traces
- Two traces:
  - Venus: 400KB request sizes
  - CCM: 16-32KB request sizes

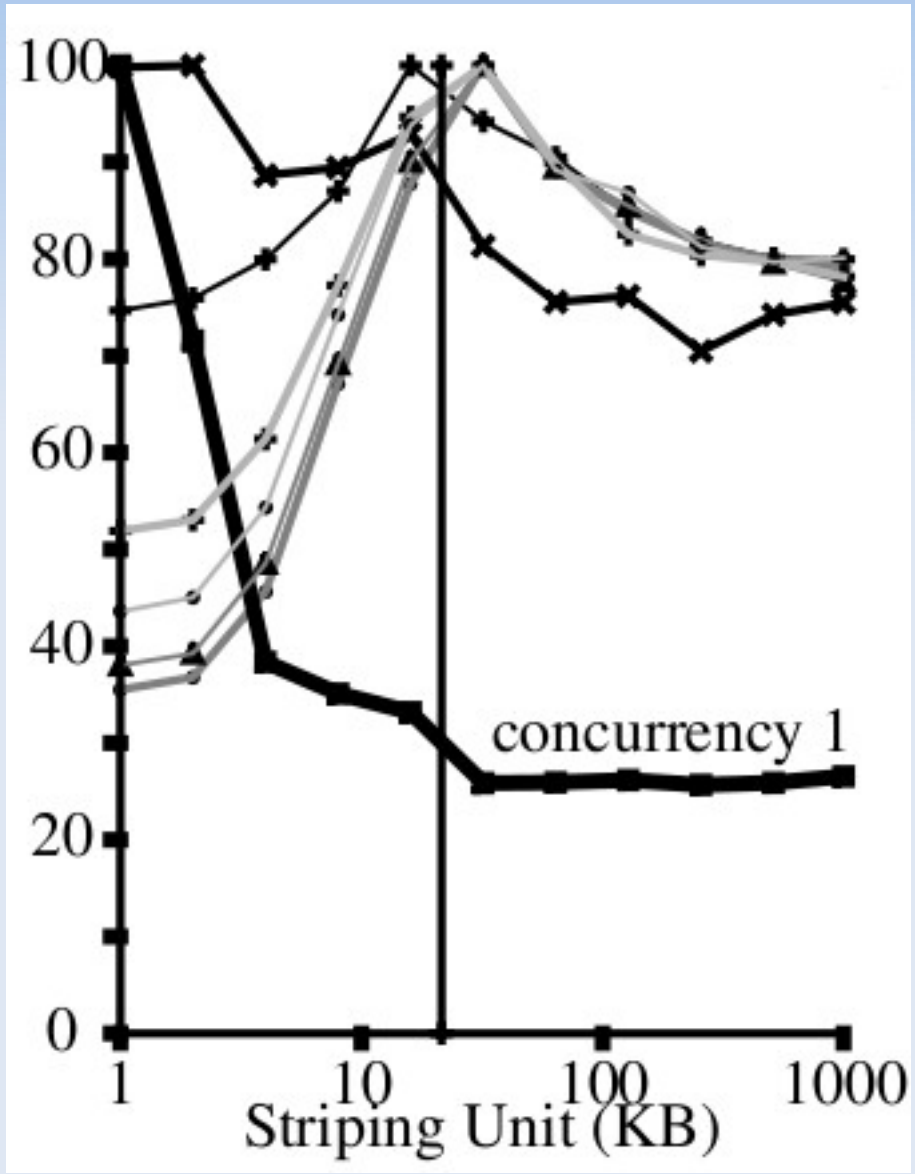
Venus Traces

- With no knowledge, 66% of max throughput guaranteed.
- Best stripe sizes are small.



CCM Traces

- Recomended size works very well except for concurrency=1
- Having concurrency=1, small stripes ensure the use of all disks.



Conclusions

- Reads behaves as non-redundat RAID.
- Writes perform better with smaller stripe size.
- Concurrency is the most important factor.
- The number of disks does not matter since the constants of Read and Write cancel out. (0.5)
- The system was evaluated with real workloads and corroborate the behavior in the artificial workloads.