


## Trading Capacity for Performance in a Disk Array

Xiang Yu,  
Ben Gum, Yuqun Chen, Randolph Wang, Kai Li  
*Princeton University*


Arvind Krishnamurthy  
*Yale University*

Thomas Anderson  
*University of Washington*

## Gigabites and Gigabytes





Twice the meat,  
same price!





Twice the bytes,  
same price!

2 Randy Wang

## Is Bigger Better?

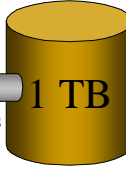

← Bottlenecks →


3 Randy Wang

## The “Absurd” Disk (stolen from Jim Gray)

100 MB/s

200 Kaps





1 TB




- 2.5 hr scan time (Poor sequential access)
- 1 access per second/5 GB (VERY cold data)
- It's a tape!

Talk at NASA Goddard on storage trends and on interesting applications (TerraServer, EOS/DS, Sloan Sky Survey), 9/21/00, Jim Gray

4 Randy Wang

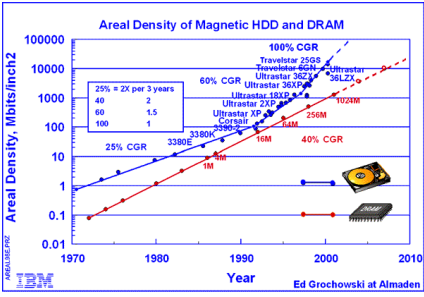
## McLean Storage

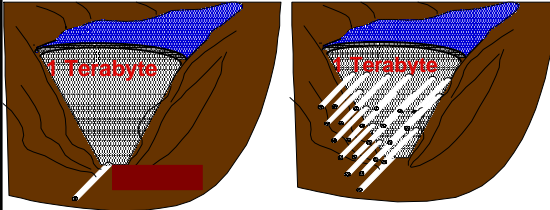
5 Randy Wang

## Can We Afford to Waste Food? Yes!



6 Randy Wang

## Many Small Devices (stolen from Jim Gray)



Talk at NASA Goddard on storage trends and on interesting applications (TerraServer, EOS/DIS, Sloan Sky Survey), 9/21/00, Jim Gray

7

Randy Wang

## Hard Questions

- So didn't RAID answer all the questions? Not quite...
- Bandwidth is easy, but how about latency and throughput?
- Given  $D \times X$  disks, what's the best way of using the extra capacity?
  - So many choices: striping, mirroring, "RAID-10", ...
  - Even better alternatives?
  - What performance can I expect on all these alternatives?
  - Guidance on configuration given workload parameters?

8

Randy Wang

## Non-contributions and Contributions

- Non-contributions
  - The big "revelation" that more disks give better performance
  - Striping, rotational data replication, and RAID-10
- Contributions
  - A new disk array design (SR-Array)
  - Analytical models and evaluations for different configurations
  - Software-only implementations of disk head location prediction and SATF disk array scheduling

9

Randy Wang

## Outline

- Introduction
- **SR-Array design and models**
- Implementation
- Experimental results
- Conclusions

10

Randy Wang

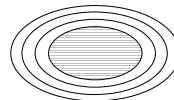
## SR-Array Overview

- Sacrifice capacity to improve **both** seek and rotational delay
  - Use **S**triping to improve seek time
  - Use **R**otational data replication to improve rotational delay
- **Balance** seek and rotational delay improvements
  - Consider disk parameters
  - Consider workload parameters

11

Randy Wang

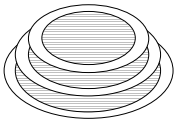
## Start with a Single Disk of Data



12

Randy Wang

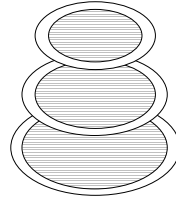
**“Split” It into Three Pieces...**



13

Randy Wang

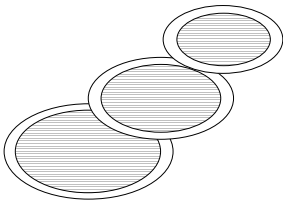
**“Split” It into Three Pieces...**



14

Randy Wang

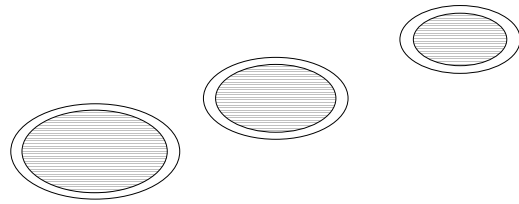
**“Split” It into Three Pieces...**



15

Randy Wang

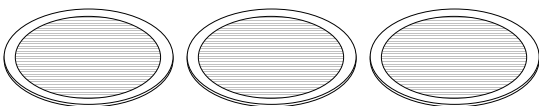
**“Split” It into Three Pieces...**



16

Randy Wang

**“Split” It into Three Pieces...**

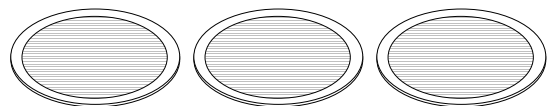


17

Randy Wang

**3-Way Striping**

- Sacrifice space in the middle
- Restrict arm movement to reduce seek delay

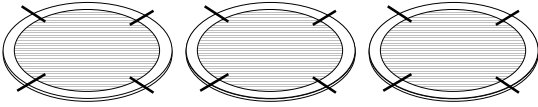


18

Randy Wang

## Rotational Delay

- Higher degree of striping hits diminishing return
- Rotational delay takes over

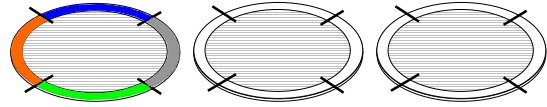


19

Randy Wang

## Rotational Delay

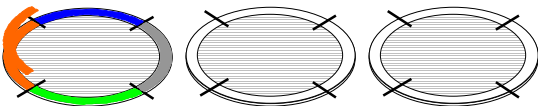
- Consider the 4 sectors in a track...



20

Randy Wang

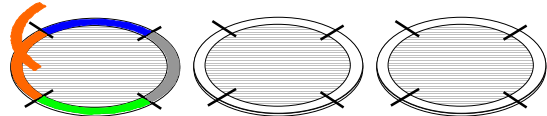
## Replicating a Sector...



21

Randy Wang

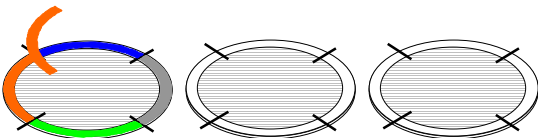
## Replicating a Sector...



22

Randy Wang

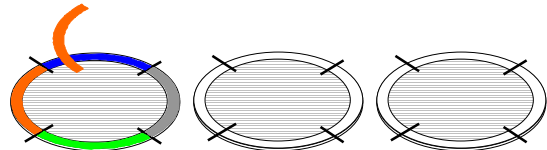
## Replicating a Sector...



23

Randy Wang

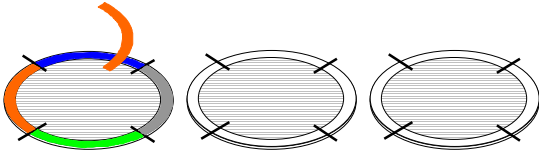
## Replicating a Sector...



24

Randy Wang

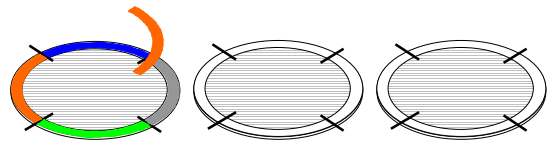
### Replicating a Sector...



25

Randy Wang

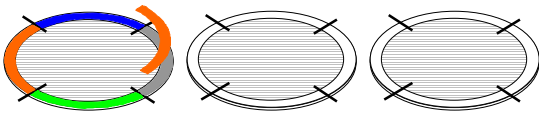
### Replicating a Sector...



26

Randy Wang

### Replicating a Sector...

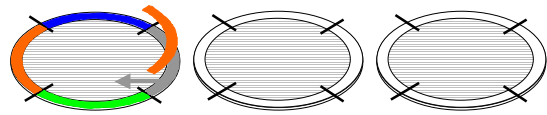


27

Randy Wang

### Replicating a Sector...

- But what about the gray sector that is about to be replaced?

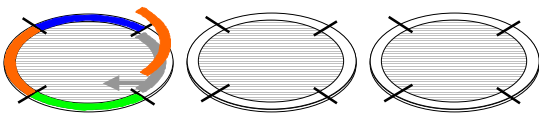


28

Randy Wang

### Replicating a Sector...

- Move the gray sector to a different track

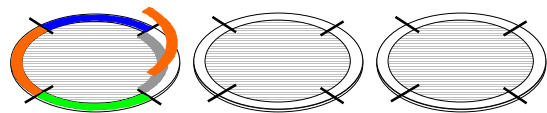


29

Randy Wang

### Replicating a Sector...

- Move the gray sector to a different track

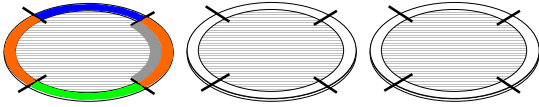


30

Randy Wang

## Rotational Data Replication

- Pick the closest red sector to read
- Average rotational delay cut in half

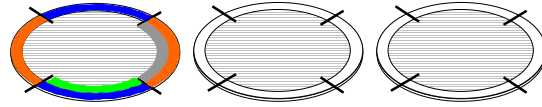


31

Randy Wang

## Rotational Data Replication

- Do the same to the blue and green sectors

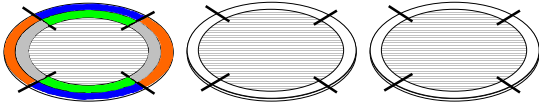


32

Randy Wang

## 2-Way Rotational Data Replication

- If we rotationally replicate every sector 2x...
- Then we occupy 2x tracks on a single disk

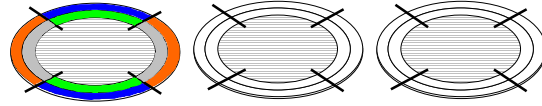


33

Randy Wang

## Rotational Data Replication

- And we do this to all disks...
- Now our seek time has gotten worse again

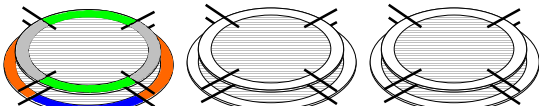


34

Randy Wang

## Splitting the Disks Again...

- To bring back down the seek time.

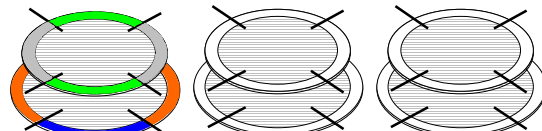


35

Randy Wang

## Splitting the Disks Again...

- To bring back down the seek time.

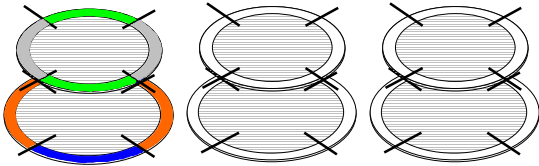


36

Randy Wang

### Splitting the Disks Again...

- To bring back down the seek time.

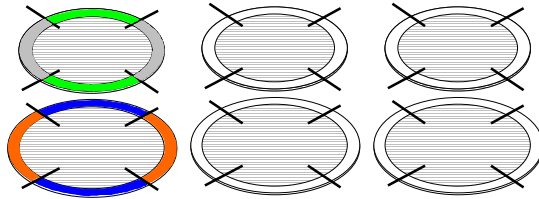


37

Randy Wang

### Splitting the Disks Again...

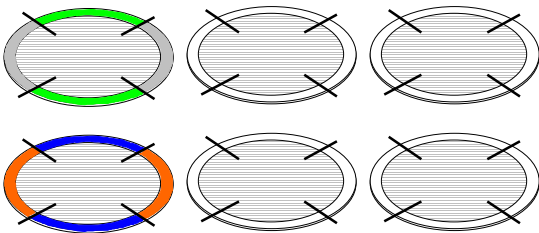
- To bring back down the seek time.



38

Randy Wang

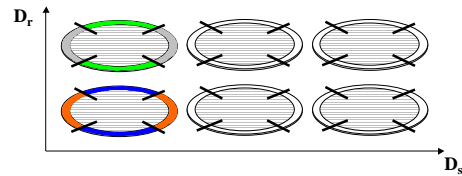
### A 3x2 SR-Array



39

Randy Wang

### A 3x2 SR-Array

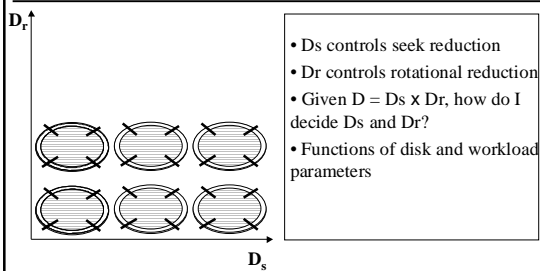


- 1/6 of original data on each disk
- < 1/3 of max/avg seek distance of original disk
- 1/2 of max/avg rotational delay of original disk

40

Randy Wang

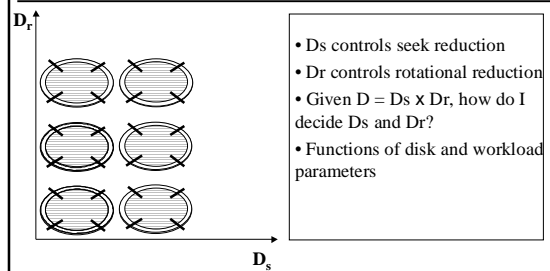
### The SR-Array "Aspect Ratio"



41

Randy Wang

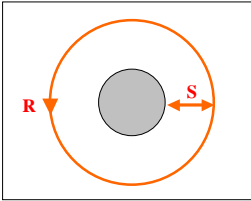
### The SR-Array "Aspect Ratio"



42

Randy Wang

## The “Square-Root Rules”



- Best configuration for random read:

$$D_s = \sqrt{\frac{2S}{3R}} D \quad D_r = \sqrt{\frac{3R}{2S}} D$$

- Best overhead-independent latency:

$$T_{best} = \sqrt{\frac{2SR}{3D}}$$

- More sophisticated models take workload parameters into account
- In general, the overhead-independent part of the latency improves by  $\sqrt{D}$

43

Randy Wang

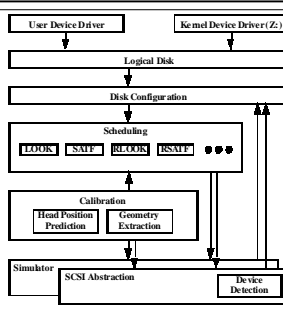
## Outline

- Introduction
- SR-Array design and models
- **Implementation**
- Experimental results
- Conclusions

44

Randy Wang

## The MimdRAID Prototype



- **SCSI**: device interaction with 9G 10K RPM Seagate disk
- **Simulator**: faithfully emulates the Seagate disks to shorten experimental time
- **Calibration**: disk layout extraction and disk head location prediction
- **Scheduling**: includes rotational positioning sensitive scheduling algorithms
- **Disk configuration**: configures an array as a striping, mirroring, or SR-Array system
- **Logical disk**: exposes the MimdRAID as drive Z: or an API

45

Randy Wang

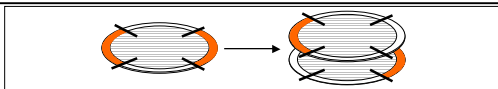
## Head Location Prediction

- Measure rotational speed by repeatedly reading the *reference sector*
- Note timestamp of last reference sector read
- Rotational position at any time can be calculated relative to the reference sector
- Re-calibrate by re-reading the reference sector every two minutes
- Error: 1% of full rotational time with 98% confidence
- Also need to extract block mapping and predict operation timing
- Use head position info to implement SATF scheduling

46

Randy Wang

## Location of Replicas

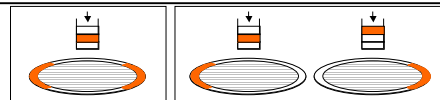


- Rotational replicas within same track hurts bandwidth
- SR-Array: replicas on different tracks in same cylinder
- RAID-10: replicas on different disks
- SR-Array vs. RAID-10
  - No “equivalent” schedule for a general request stream
  - Scheduling is more complicated on RAID-10
  - RAID-10 provides reliability
  - Can combine the two

47

Randy Wang

## Scheduling



- Reads
  - Per-drive SATF scheduling
  - For mirroring: queue request on all relevant drives, and cancel all but one
- Writes
  - Write closest replica synchronously as we do reads
  - Remaining replicas propagated in background
  - Overwritten blocks discarded from background queue
  - Location(s) of partially completed block propagation kept in NVRAM for recovery

48

Randy Wang

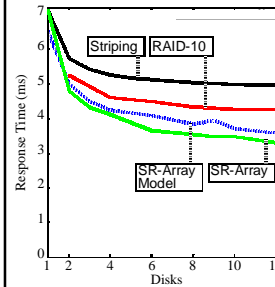
## Outline

- Introduction
- SR-Array design and models
- Implementation
- **Experimental results**
- Conclusions

49

Randy Wang

## File System Response Time (HPL Cello)

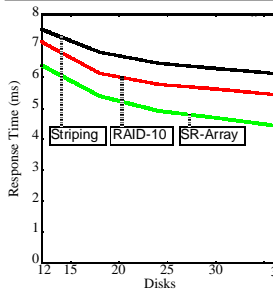


- SR-Array: best balance of seek and rotation reduction
- Stripping: diminishing return of seek reduction
- RAID-10 (2-way mirroring plus striping): not enough rotational reduction
- Response time model is a good approximation

50

Randy Wang

## TPC-C Response Time

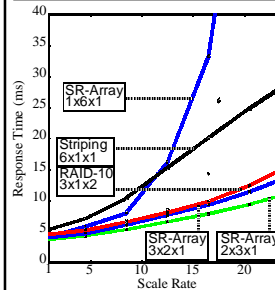


- SR-Array: best balance of seek and rotation reduction
- Stripping: diminishing return of seek reduction
- RAID-10 (2-way mirroring plus striping): not enough rotational reduction

51

Randy Wang

## Accelerate File System Trace (HPL Cello)

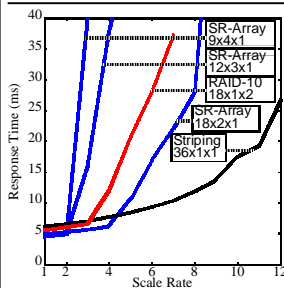


- Notation: Ds x Dr x Dm
- Even under high I/O rate, replication still desirable
- Even under high I/O rate, there is idle time
- Even when there's no idle time, benefit of reading closest replica may still outweigh replication cost

52

Randy Wang

## Accelerate TPC-C Trace



- 45% writes, no "idle" time
- Under low I/O rate, high degree of rotational replication still wins
- As I/O rate rises, progressively lower degree of rotational replication

53

Randy Wang

## Outline

- Introduction
- SR-Array design and models
- Implementation
- Experimental results
- **Conclusions**

54

Randy Wang

## On-going Work



- One  $D_s \times D_r \times D_m$  configuration



- A couple configuration levels, like AutoRAID



- A smooth continuum of configuration levels
  - Highly dynamic
  - Movement across levels while minimizing copies
  - True self-configuring storage

55

Randy Wang

## Conclusions

- Necessity, opportunity, and feasibility of trading capacity for performance in a disk array
- A new disk array design (SR-Array) for reducing seek and rotational delay, and hence, improving latency and throughput
- Models that guide its configuration based on disk and workload characteristics

56

Randy Wang

## Princeton MimdRAID



<http://www.cs.princeton.edu/~rywang/mimdraid>

57

Randy Wang