# Data Allocations in a Heterogeneous Disk Array (HDA)

Alexander Thomasian & Jun Xu
athomas@cs.njit.edu
Integrated Systems Laboratory
Computer Science Department
New Jersey Institute of Technology
Newark, NJ 07102

# Outline

- Motivation for HDA
- Description of HDA
- Allocation methods
- Simulation results
- Preliminary conclusions

# Motivation

- Varied storage requirements
- One disk array for all applications
- High data management costs
- Efficient disk utilization

# Diverse Application Requirements

- Match RAID levels to requirements:
  - ✓ Datasets requiring high protection – RAID6
  - ✓ Cost-effective reliable storage – RAID5
  - ✓ Reliability and high performance – RAID1
  - ✓ Low integrity or temporary data –RAID0
- No single RAID level meets all requirements.

# High storage management cost

- Storage management tasks:
  - Configuring disk arrays
  - Monitoring performance
  - Dealing with disk failures
- Storage managers highly paid.
- Disk prices dropping.
- Disk management cost >> storage cost.
- Disk array as self-managed as possible.

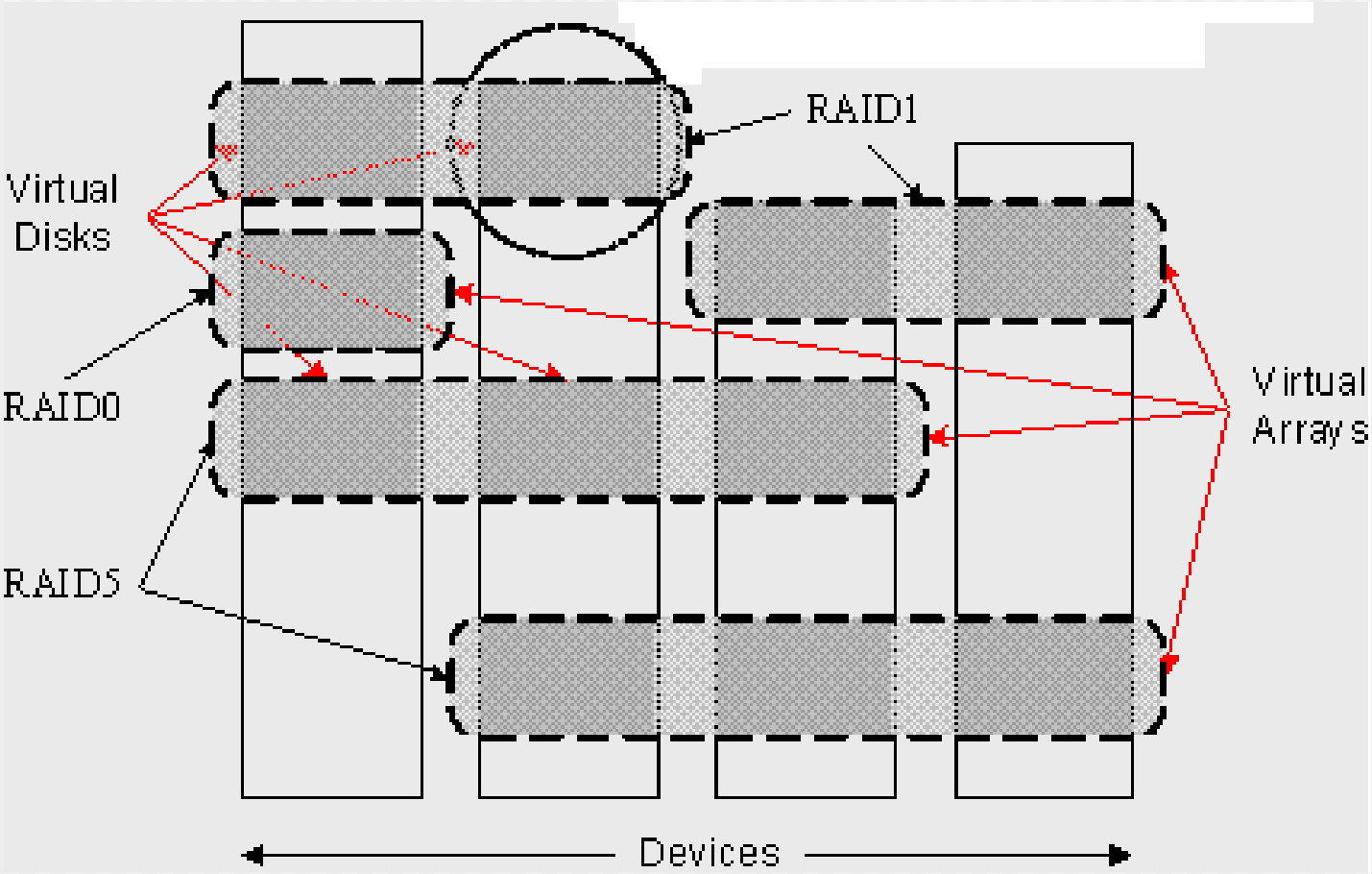
# Desirable storage system

- New disk drives/bricks utilized well (heterogeneous devices).
- Combine multiple RAID levels in same physical array (heterogeneous RAID levels).
- Roughly balance disk loads.
- Provide acceptable response time.

# Heterogeneous Disk Array-HDA

- Dataset attributes used to map requests to a VA with appropriate RAID level.
- HDA can store Virtual Arrays – VA's at different RAID levels.
- Support for erasure coding and replication to handle VAs in different categories
- Disk array controllers supporting VAs in different categories:RAID0/1/5/6 feasible.

# Entities in HDA



# Allocation Requests for Virtual Arrays

- VA allocations become available one at a time & processed immediately (no batching).
- RAID level deduced from dataset attributes.
- Allocation size depends on RAID level.
- Replication and parity cost additional.
- Required disk bandwidth depends on workload:
  - Rate of requests to read/write data blocks.
  - Fraction of reads/writes.
  - Size of data blocks being accessed.
  - Distribution of requests.

# VA Allocation Requests

- Given RAID level, 2 characteristics:
  - Disk bandwidth utilization
  - Disk capacity requirement
- VA width determines number of Virtual Disks (VDs), includes disks for replication and parity.
- Two parameters: max bandwidth and capacity per VA at each disk

# Allocation Methods for VDs

1. **Round Robin**: Allocate on disk drives sequentially.
2. **Random**: Select disk drives randomly.
3. **Best Fit**: Select disk drive with minimum remaining bandwidth.
4. **First Fit**: Allocate VD on first disk that can hold it.
5. **Worst Fit:** Allocate starting with disks sorted in non-increasing order of bandwidth utilization.
6. **Minimize F1**: minimize the maximum utilizations of disks throughput and capacity.
7. **Minimize F2**: minimize the variance for utilizations of throughput and capacity

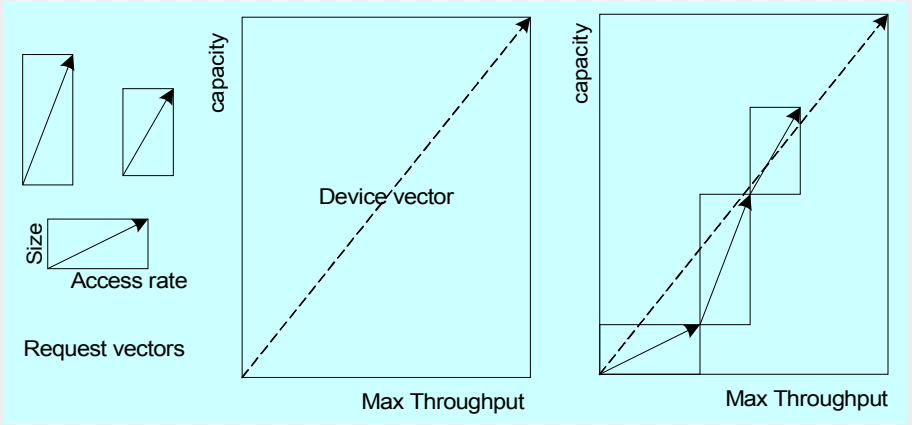
# Algorithm

Starting with first allocation request (i=1).

- Determine VA RAID level (RAID1 or RAID5).
- Determine VA size.
- Determine access rate (depends on RAID level, VA size, bandwidth boundedness).
- Determine VA width based on disk capacity and bandwidth constraints.
- Make sure disk failures do not cause overload.
- Stop if a successful allocation is not possible.
- Increment the utilization of devices to which the VDs of a VA are assigned.
- Increment counter i and return to Step 1.

# Simulation configuration

- 12 “IBM18ES” disk drives, 9.17GB, max 85 1/sec. to small (4 KB) blocks, uniformly distributed
- RAID1:RAID5 = 1:3
- Read/Write ratio = 1:0
- Three cases:
  - Bandwidth bound
  - Capacity bound
  - Balanced



# Results—Bandwidth bound

| | | *Utilizations* | | *Allocations* | |
|---|---|---|---|---|---|
| *Method* | *Best of 100* | *BW %* | *Cap %* | *R1* | *R5* |
| Best Fit | 0 | 53.00 | 24.00 | 7 | 25 |
| Worst Fit | 73 | 83.33 | 39.00 | 9 | 39 |
| Round Robin | 2 | 70.67 | 32.67 | 8 | 33 |
| Random | 35 | 80.33 | 36.67 | 10 | 37 |
| First Fit | 0 | 55.33 | 25.33 | 6 | 26 |
| Min F1 | 75 | 84.00 | 39.00 | 9 | 39 |
| Min F2 | 75 | 84.00 | 39.00 | 10 | 39 |

# Results—Balanced

| | | *Utilizations* | | *Allocations* | |
|---|---|---|---|---|---|
| *Method* | *Best of 100* | *BW %* | *Cap %* | *R1* | *R5* |
| Best Fit | 0 | 51.00 | 45.00 | 13 | 45 |
| Worst Fit | 73 | 86.00 | 76.00 | 20 | 76 |
| Round Robin | 0 | 74.67 | 66.00 | 18 | 66 |
| Random | 16 | 81.00 | 71.00 | 20 | 72 |
| First Fit | 0 | 51.67 | 45.67 | 12 | 46 |
| Min F1 | 76 | 86.00 | 76.00 | 20 | 76 |
| Min F2 | 83 | 86.00 | 76.67 | 20 | 77 |

# Results—Capacity bound

| | | *Utilizations* | | *Allocations* | |
|---|---|---|---|---|---|
| *Method* | *Best of 100* | *BW %* | *Cap %* | *R1* | *R5* |
| Best Fit | 1 | 32.00 | 59.00 | 17 | 60 |
| Worst Fit | 1 | 46.00 | 86.67 | 23 | 87 |
| Round Robin | 4 | 49.00 | 91.00 | 25 | 91 |
| Random | 9 | 49.33 | 91.67 | 26 | 92 |
| First Fit | 0 | 32.33 | 60.33 | 17 | 60 |
| Min F1 | 84 | 53.00 | 98.00 | 27 | 98 |
| Min F2 | 86 | 53.00 | 98.00 | 27 | 98 |

# Conclusion

- Need consider disk bandwidth utilization and capacity to get robust allocations.
- Minimize F1 and F2 consistently the best in terms of the number of allocations. .
- First Fit is the worst among all methods.
- Worst Fit comparable with F1 & F2 when balanced or bandwidth bound, but worse when capacity bound.
- Much more work remains, e,g., the effect of max bandwidth/ capacity limits!

# THANK YOU!
# Any Questions?