Active Data Systems

Narasimha Reddy

Narasimha Reddy
reddy@ece.tamu.edu

Presentation overview

- Experience from Multi-view Storage System (MVSS)
- Plan for future work

Narasimha Reddy
reddy@ece.tamu.edu

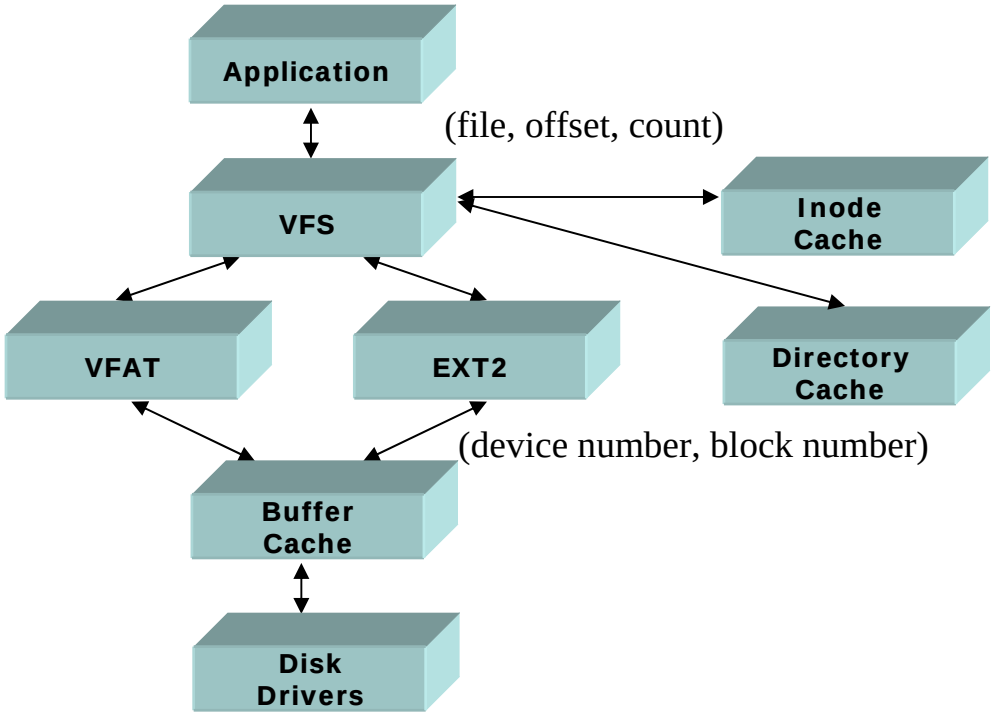
Active Storage Systems

- Proposed to move processing closer to storage
  - To reduce the I/O interconnect bottlenecks
  - To improve parallelism in data processing by exploiting cheap processors
  - Scale processing power with storage

Narasimha Reddy
reddy@ece.tamu.edu

Data access in traditional systems

- Traditional storage systems using block addresses to access data



Narasimha Reddy
reddy@ece.tamu.edu

Problem

- More information needs to be passed to device for supporting application specific services, includes:
  - Type of service
  - Parameters
    - Key for encryption
    - Query criteria for database operations
- Lack of file level information on disk

Narasimha Reddy
reddy@ece.tamu.edu

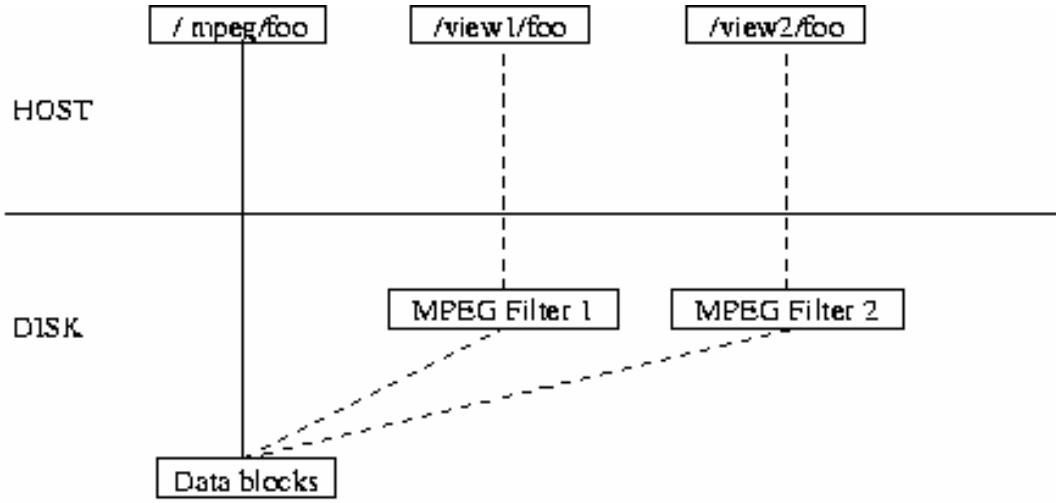
Multi-View Storage System (MVSS)

- Key concepts
  - Multiple views of a file (i.e., virtual files)
    - flexible user interface
    - transparent service deployment
  - Virtual block addresses
    - compatible with block interface

Narasimha Reddy
reddy@ece.tamu.edu

Virtual files

- A view represents a file associated with certain services
  - Views are provided through name space
    - Each view has its own pathname, in-core inode
  - Views are generated dynamically
    - Similar to multi-view database (MVDB) system



Narasimha Reddy
reddy@ece.tamu.edu

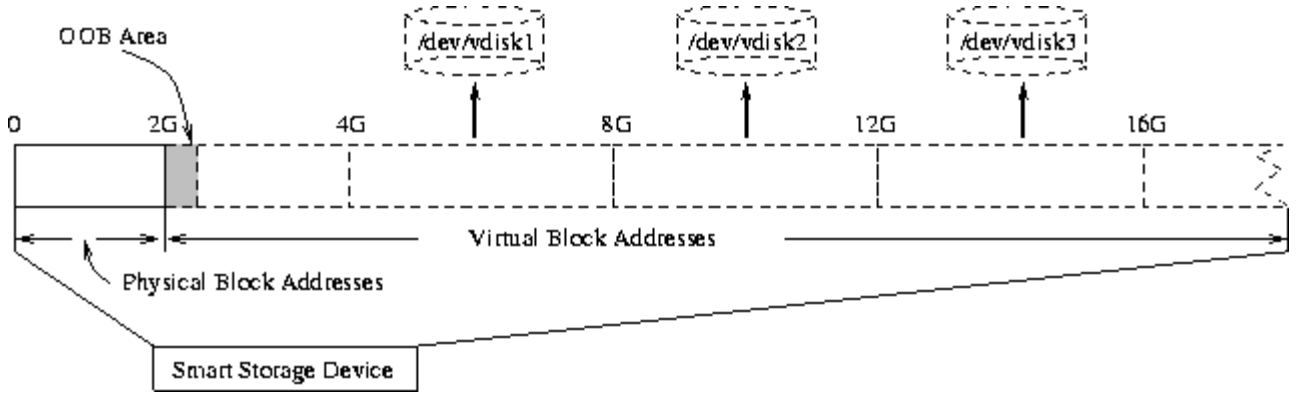
Virtual block addresses

- Problems:
  - How should different views of the same file be cached?
    - let each virtual file have its own cache buffers
  - How to pass service binding information to device level through existing interfaces?
- Virtual block addresses — block addresses beyond the physical device capacity

Narasimha Reddy
reddy@ece.tamu.edu

Virtual block addresses

- Observation
  - normal disk capacities are much smaller than what OS can support
- Out-of-band communication through OOB area (similar to memory mapped I/O)



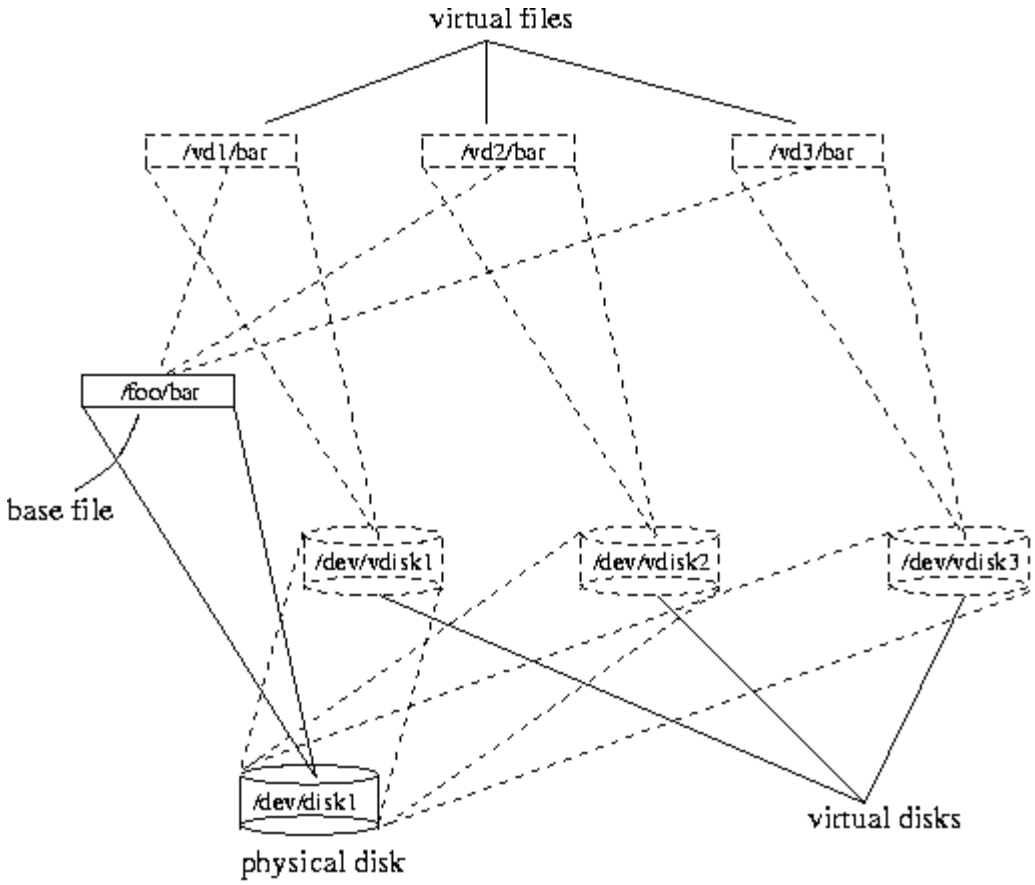
Narasimha Reddy
reddy@ece.tamu.edu

Virtual disks

- Place holders for virtual blocks and virtual files
  - Virtual disk has no corresponding physical device
- Hooking a virtual disk to a block device
  - Causes all I/Os to it be forwarded to the underlying device
  - Allows mounting specified directory on the device under new location in the namespace

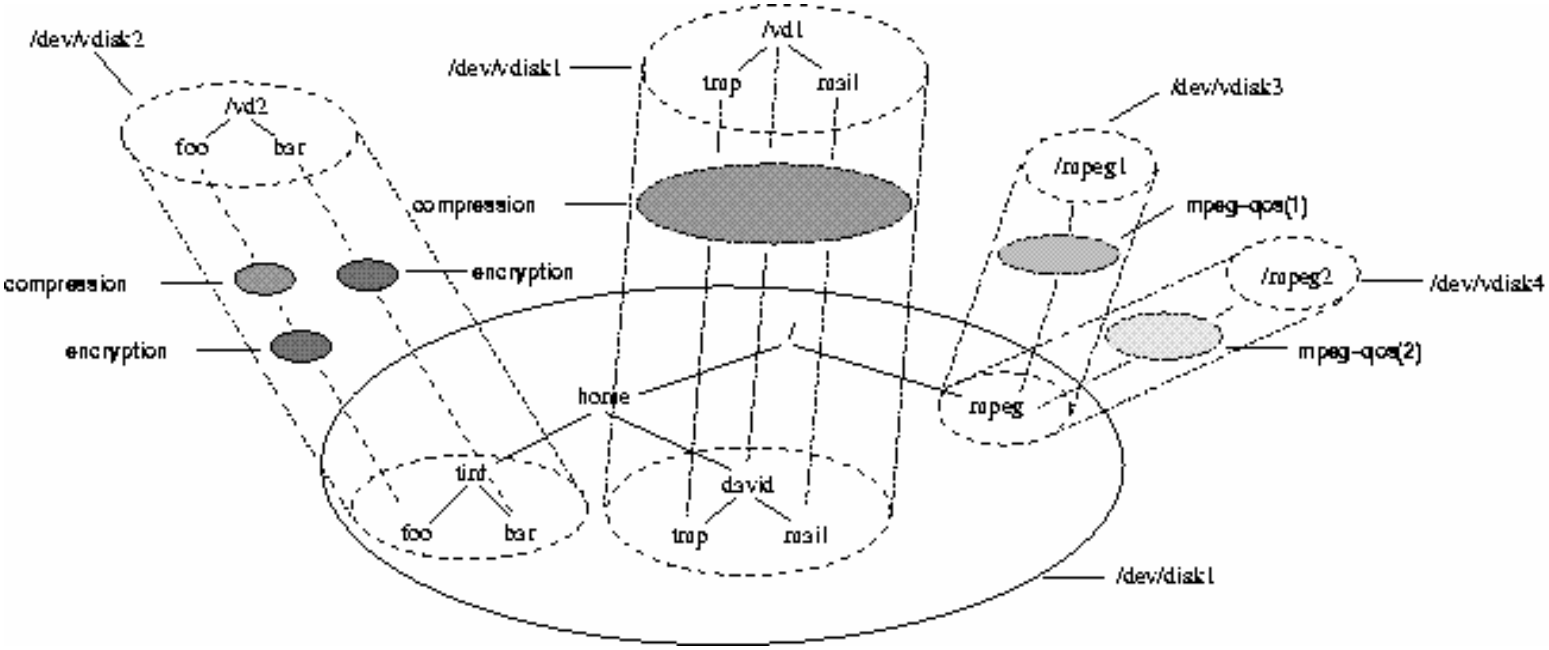
Narasimha Reddy
reddy@ece.tamu.edu

Virtual files and virtual disks



Narasimha Reddy
reddy@ece.tamu.edu

Example of views



Narasimha Reddy
reddy@ece.tamu.edu

Service binding at device level

- Virtual blocks are allocated to virtual files
- Virtual blocks of a virtual file associated with service information through virtual block maps
- Virtual block maps and service binding information replicated on disk

Narasimha Reddy
reddy@ece.tamu.edu

Virtual block map

- Let disk know how to provide required service based solely on block addresses
- One virtual block map for each hooked virtual disk
- Efficient virtual block map management through segments and zones
  - Efficient binding
  - Small memory footprint

Narasimha Reddy
reddy@ece.tamu.edu

Application Interface

- Attach: *attach(virtual file, service, parameters)*
  - Example: *attach(/view1/foo, encryption, key)*
- Separates service binding from other file operations
- Supports binding at the granularity of a single file
  - Attach a directory affects all files under it

Narasimha Reddy
reddy@ece.tamu.edu

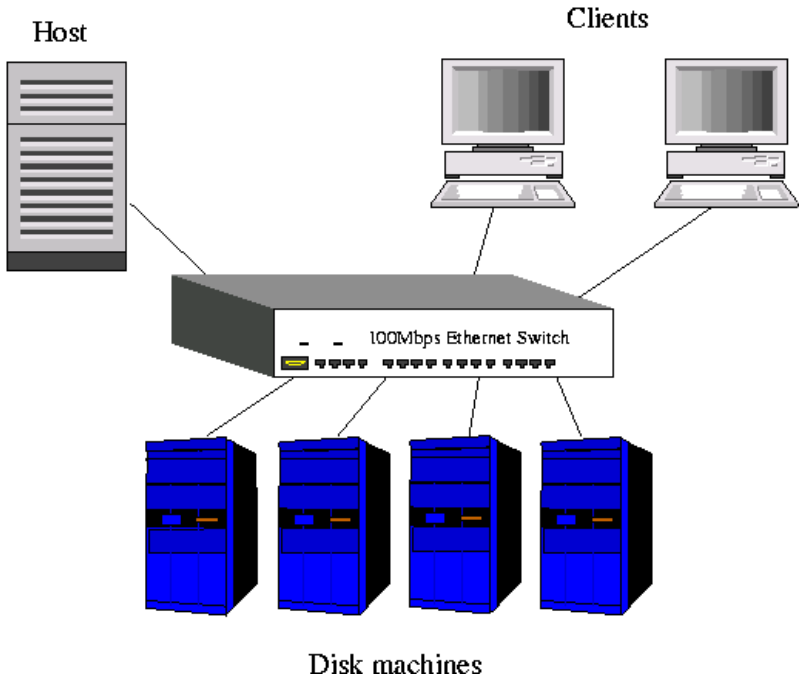
Programming model

- Filter applets
- Security issues for filter applets
  - Filter applets are executed at process level
  - Restricted access to disk resources
  - Can only initiate IO requests in a limited way

Narasimha Reddy
reddy@ece.tamu.edu

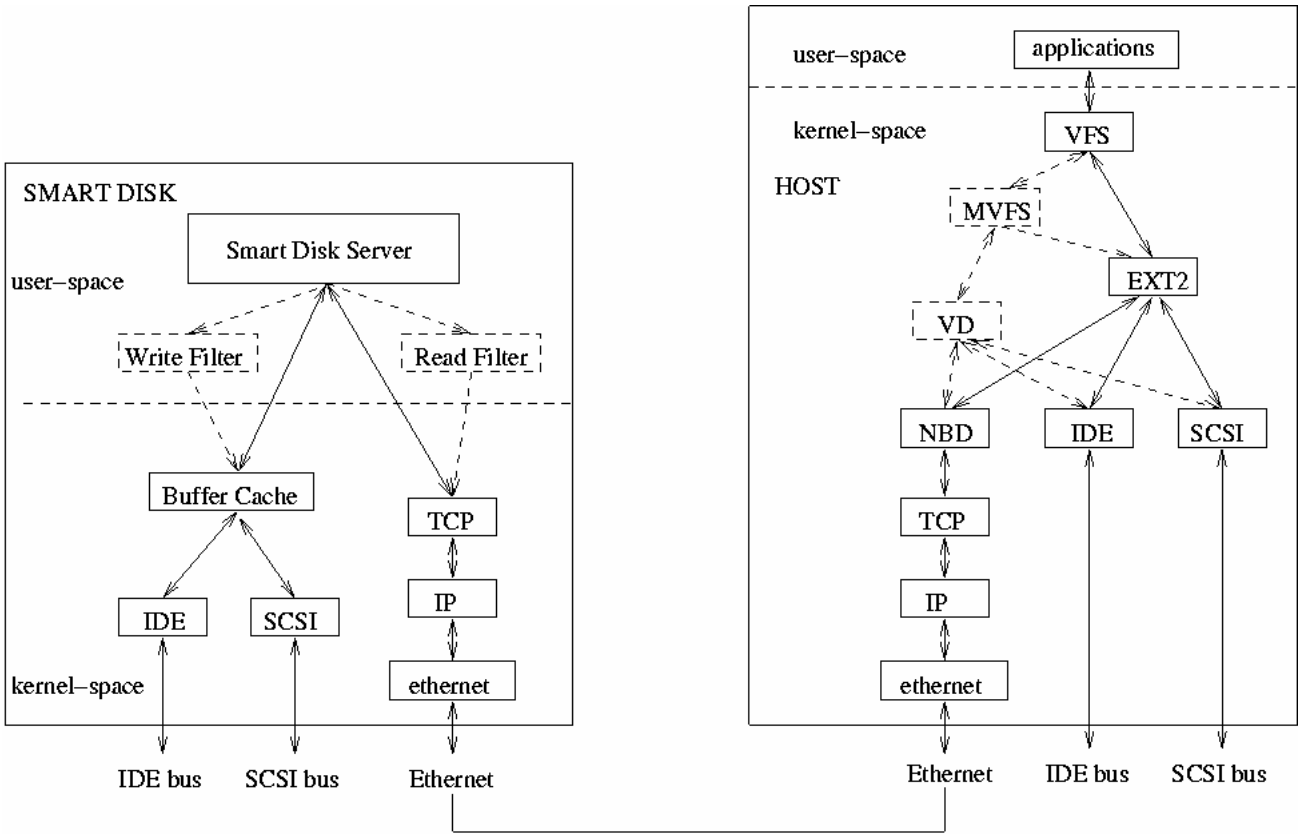
Testbed configuration

- Implemented on PCs running Linux



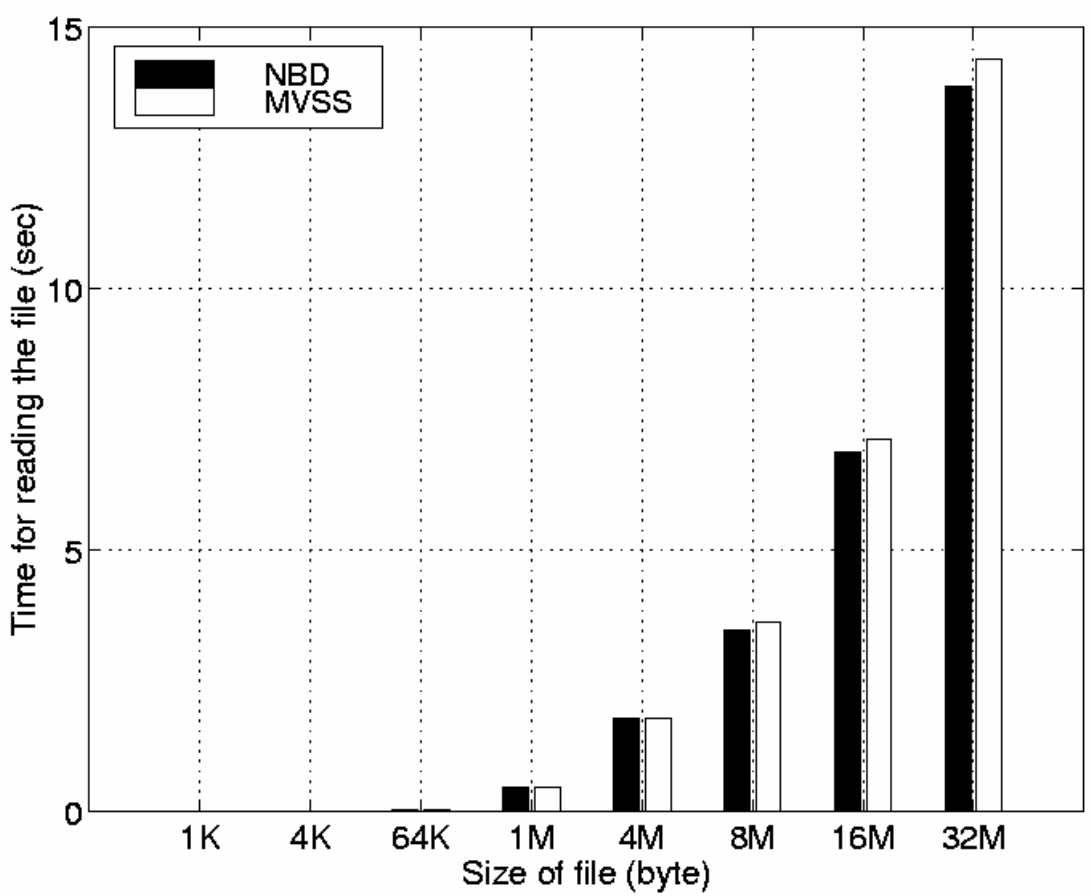
Narasimha Reddy
reddy@ece.tamu.edu

Structure of the prototype



Narasimha Reddy
reddy@ece.tamu.edu

Overhead



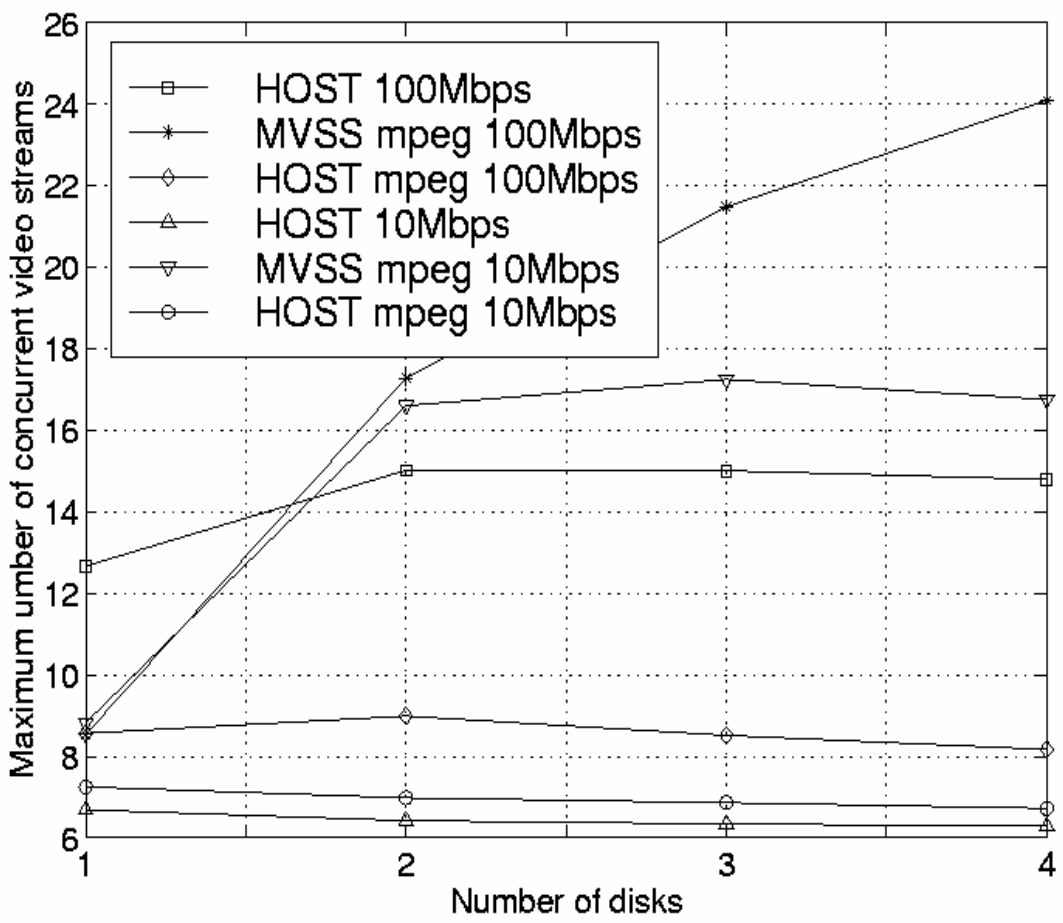
Narasimha Reddy
reddy@ece.tamu.edu

Applications

- Mpeg QoS filtering
  - Both parallelism and bandwidth reduction
- Image processing with Median filter
  - Allows processing to be flexibly split between host and disks

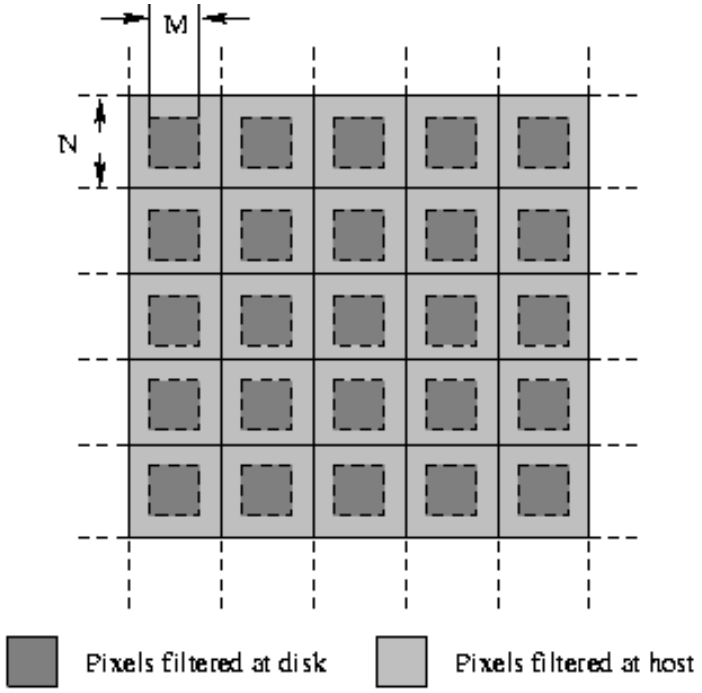
Narasimha Reddy
reddy@ece.tamu.edu

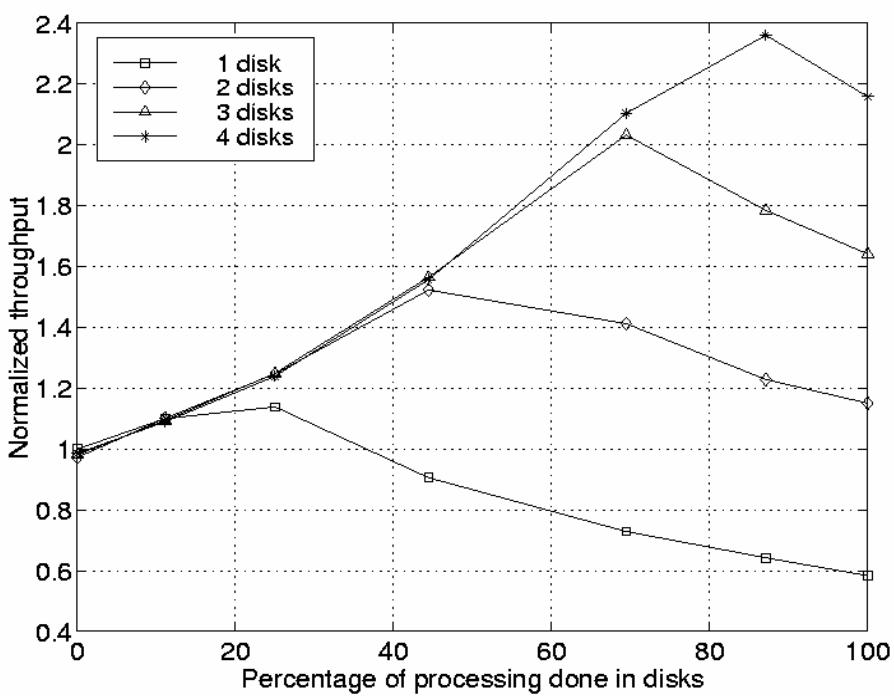
MPEG QoS filtering



Narasimha Reddy
reddy@ece.tamu.edu

Image Processing with Median filter





Narasimha Reddy
reddy@ece.tamu.edu

Mixed workload

- Different data requests require different processing
  - normal data requests and active data requests
  - active data requests from different applications
- Should not treat all data requests in the same way
- How should data requests be scheduled at the device?

Narasimha Reddy
reddy@ece.tamu.edu

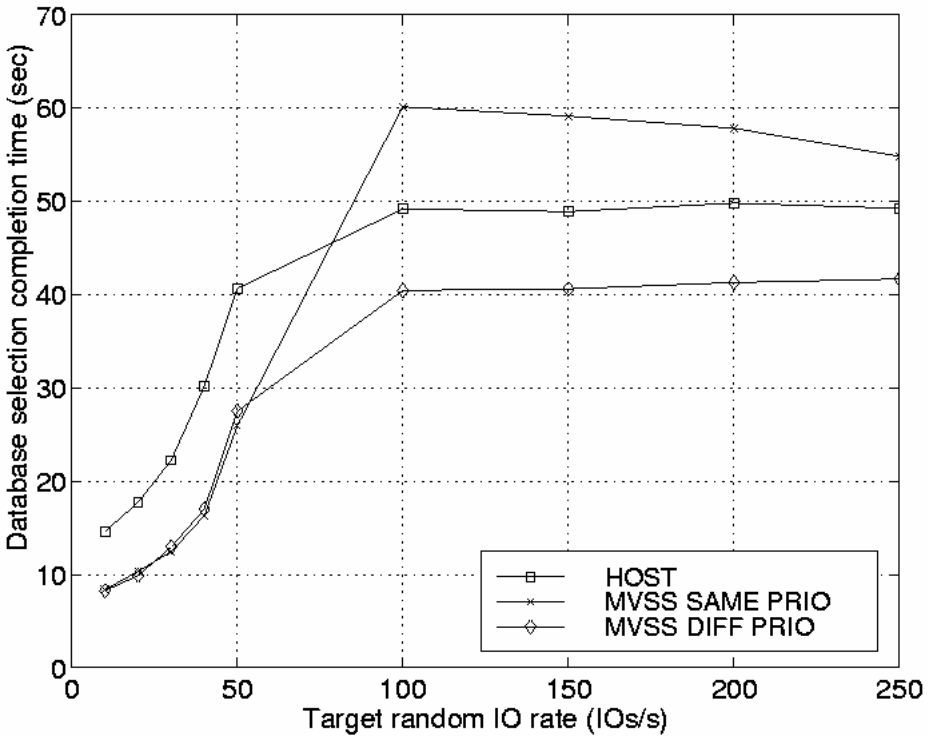
Mixed workload experiment (1)

- Active data requests + normal data requests
  - active data requests
    - database SELECT operation (selectivity = 8)
  - random normal data requests
    - 10 processes, 4KB size request, varying rates
- Two threads on disks for active and normal requests respectively
  - 1st setting: same priority
  - 2nd setting: higher priority for active thread

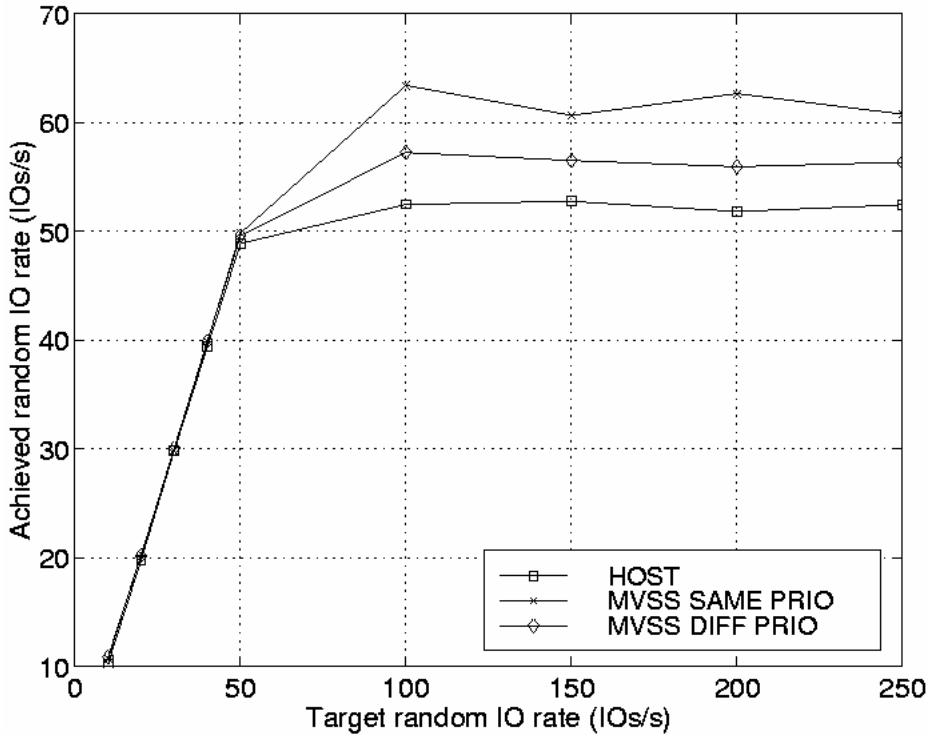
Narasimha Reddy
reddy@ece.tamu.edu

Mixed workload results (1)

Database select completion time



Achieved random I/O rate



Narasimha Reddy
reddy@ece.tamu.edu

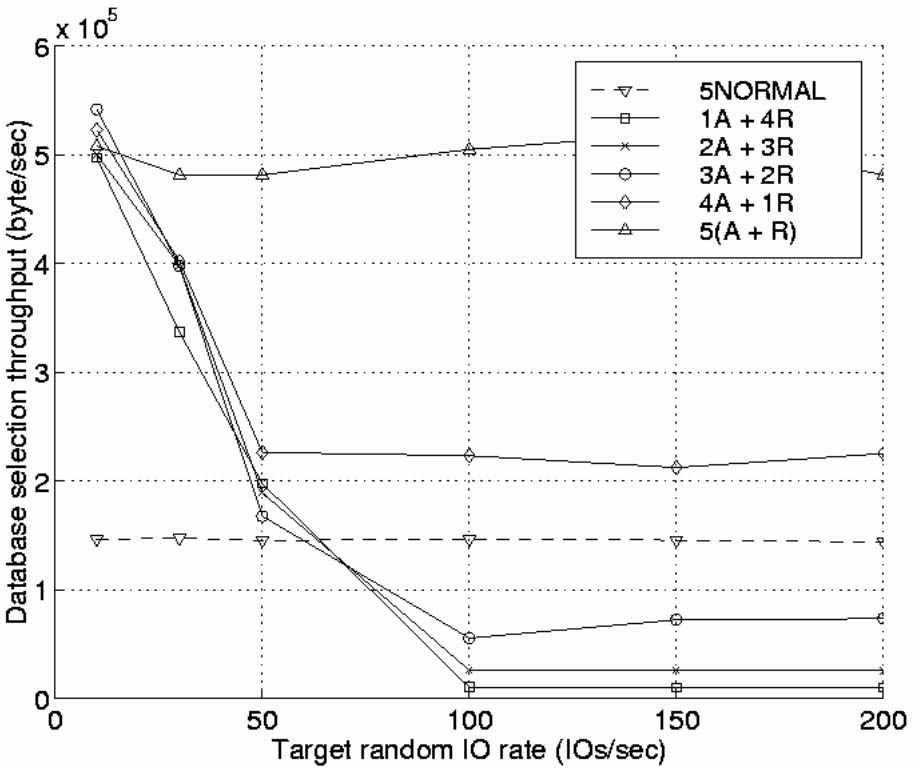
Mixed workload experiment (2)

- Totally 5 working threads
- data requests distributed in various combinations:
  - common FIFO queue (5(A+R))
  - 3 active threads and 2 normal threads (3A+2R)
  - normal system (5 NORMAL)

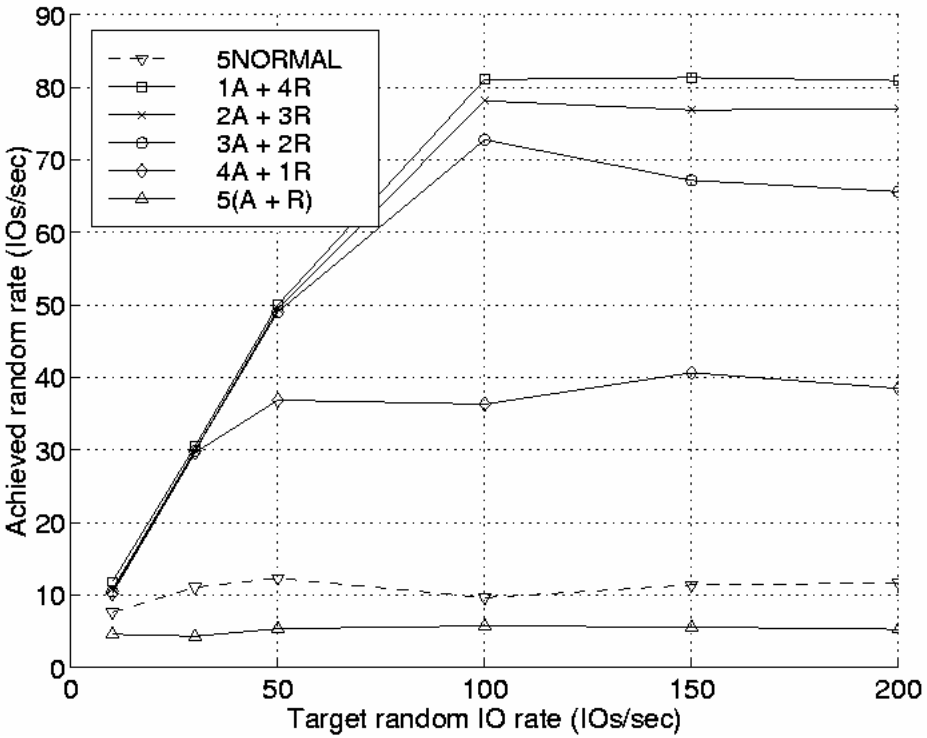
Narasimha Reddy
reddy@ece.tamu.edu

Mixed workload results (2)

Database select throughput



Achieved random IO rate



Narasimha Reddy
reddy@ece.tamu.edu

Summary of MVSS results

- Allows flexible migration of processing to disks
- Significant performance improvement
- Mixed workload experiments demonstrated the importance of scheduling policies at disks

Narasimha Reddy
reddy@ece.tamu.edu

Future Work

- Study scheduling and QOS issues related to
  - Multiple Active requests
  - Active requests vs. Normal requests
  - Processing vs. Disk bandwidth
- Much work related to Multimedia & Network QOS scheduling can be leveraged

Narasimha Reddy
reddy@ece.tamu.edu

Future Work

- Security and DOS protection
  - Active requests can lead to more problems
  - Exhaustion of processing resources
  - Potential crashes
  - Isolation and fair scheduling

Narasimha Reddy
reddy@ece.tamu.edu

Future Work

- Active allocation?
- Can we maintain mirror copies of a matrix, one in row-major, one in column-major fashion
  - Improves read accesses through allocation
  - How to extract data properties that can be exploited?

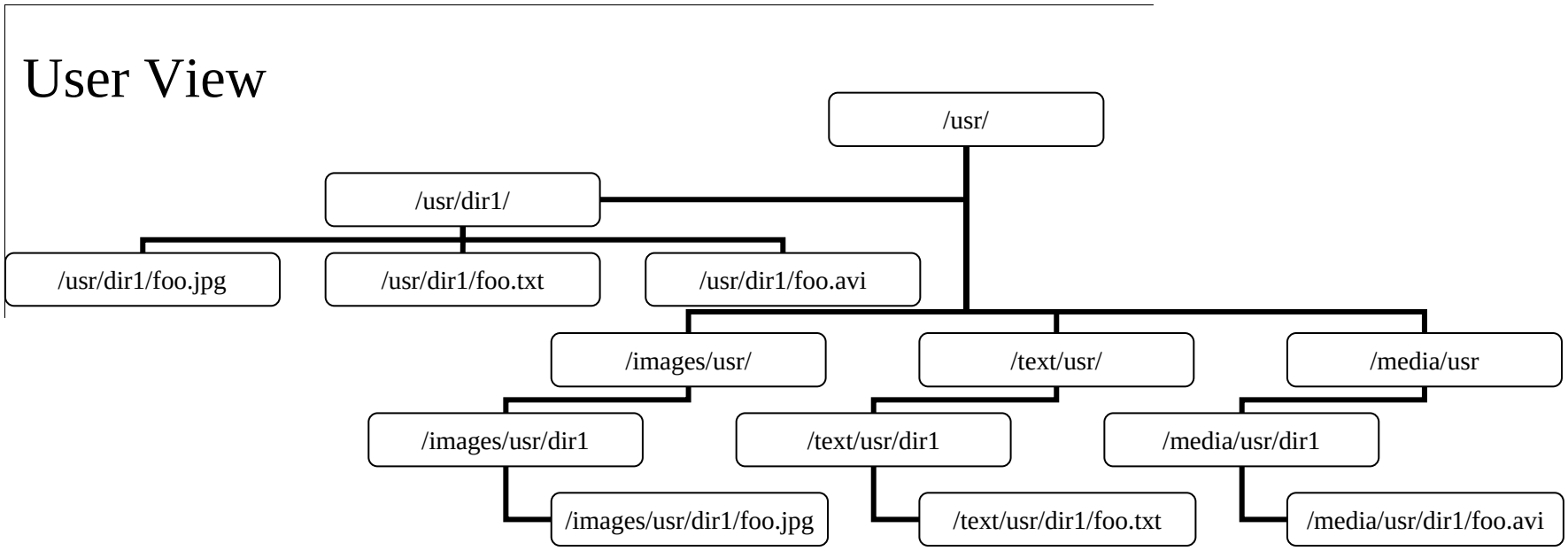
Narasimha Reddy
reddy@ece.tamu.edu

Future Work

- Can we allocate data based on its content or other policies?
- Examples:
  - Multiple levels of encryption for different levels of protection
  - Some data protected through mirroring, some through parity protection, some not protected…
- How to detect the content types?
- How to continue providing same user interface of unified directory for different types while providing different allocations for different files?

Narasimha Reddy
reddy@ece.tamu.edu

Content Type based Allocation



Underlying data organization

Narasimha Reddy
reddy@ece.tamu.edu

Conclusion

- Scheduling, QOS, security in active storage systems needs considerable work
- Much potential
  - Improved performance
  - Improved functionality
    - Policy-based storage

Narasimha Reddy
reddy@ece.tamu.edu