
Fault Recovery Designs for Processor-Embedded Distributed Storage Architectures with I/O- Intensive DB Workloads

**22nd IEEE/13th NASA Goddard Conference on
Mass Storage Systems and Technologies**

Steve C. Chiu[†], Alok N. Choudhary[‡], Mahmut T. Kandemir[§]
Benedict College[†], Northwestern University[‡], Penn State University[§]

Outline

Motivation

Related Work

Architecture

Device Model

Recovery Schemes

Workloads and Scenarios

Preliminary Results

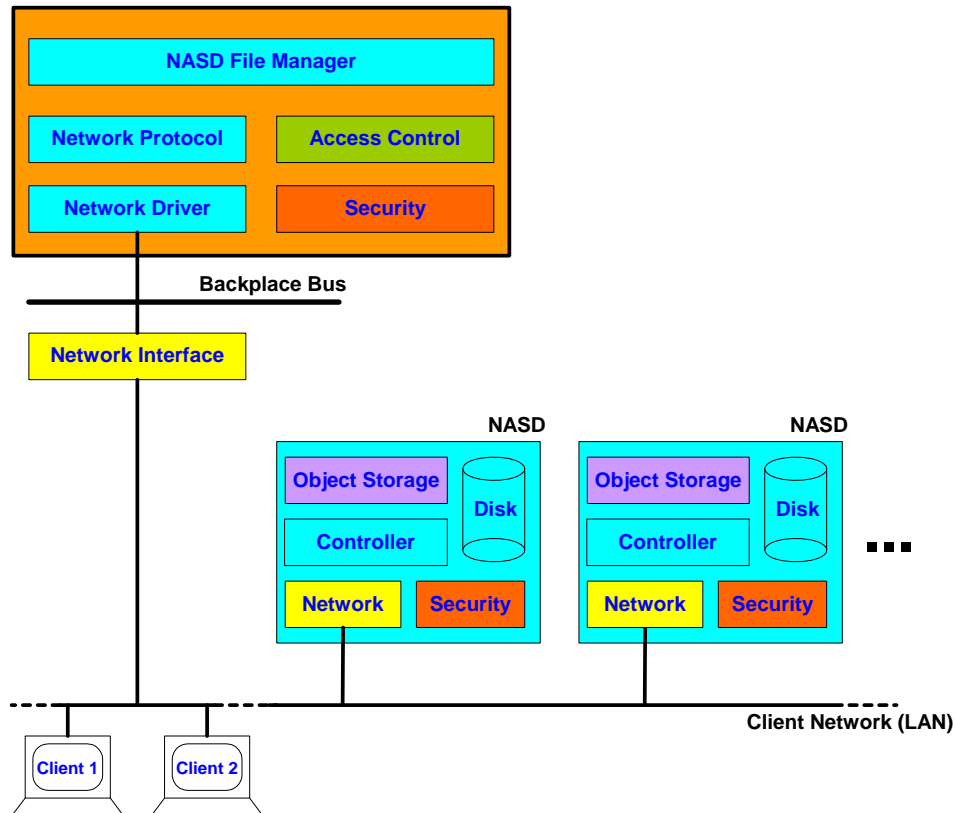
Summary

Future Work

Motivation

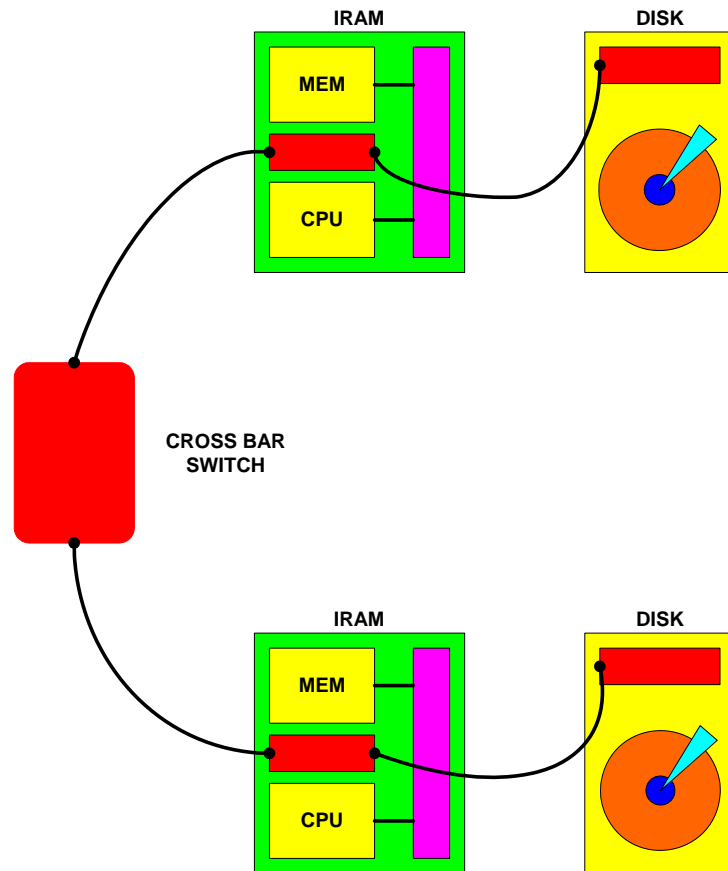
**Computation in storage
brings processing closer to data
and reduces interconnect traffic,
thus higher system performance.
However, efficient and scalable
fault tolerance capabilities
are equally important!**

Related Work



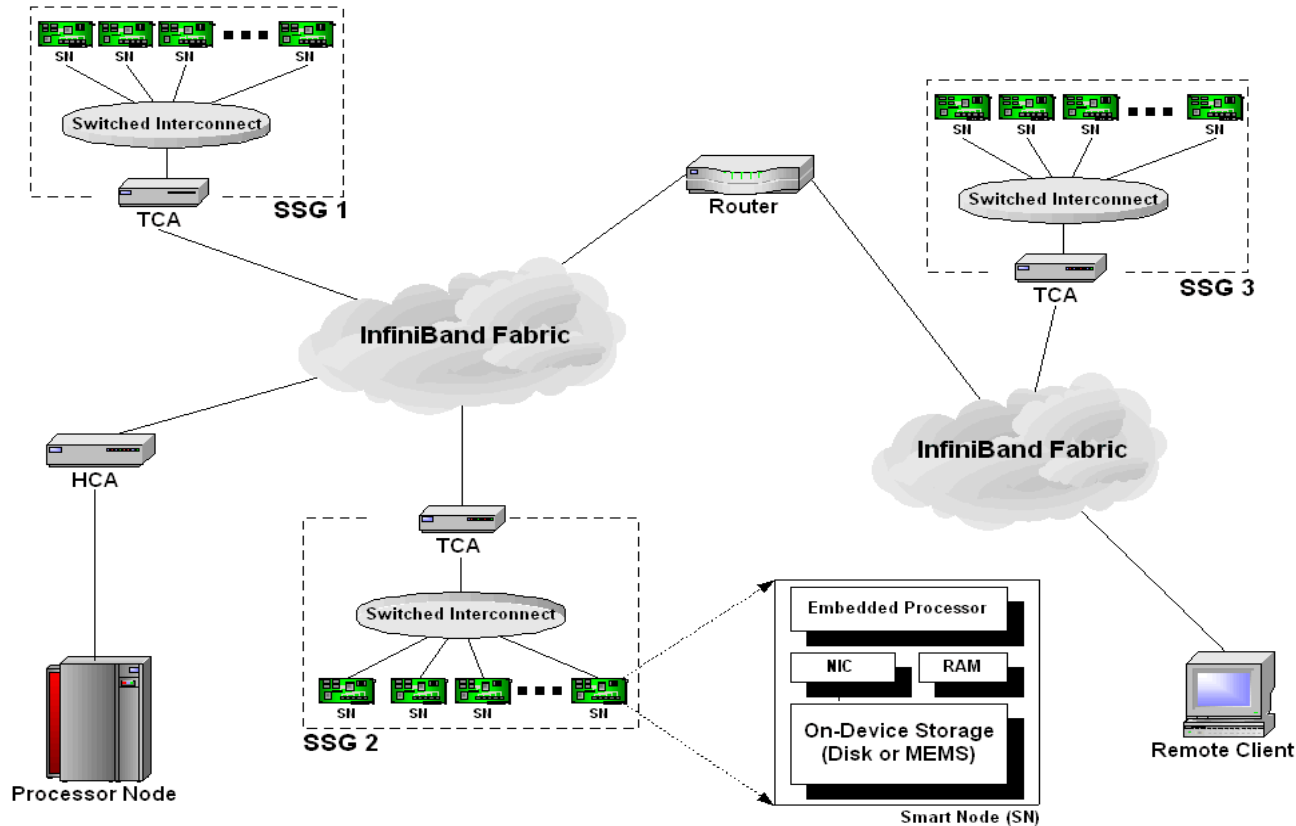
Network-attached secure disks (NASD) for offloading certain file system's performance-critical operations

Related Work



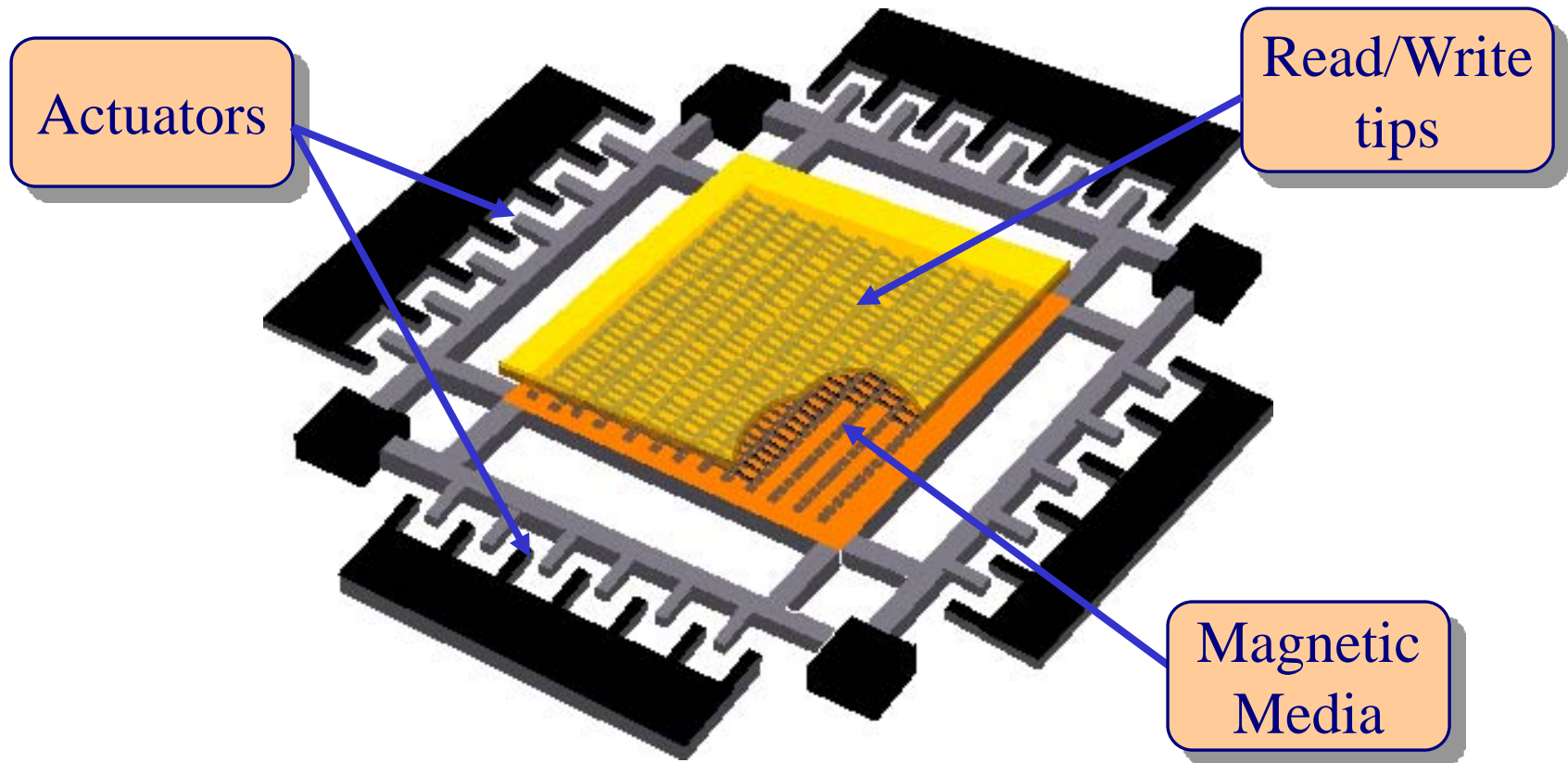
Intelligent Disk (IDISK) with IRAM and cross bar switch for full connectivity – each IDISK supports complete OS and DBMS functionality

Architecture



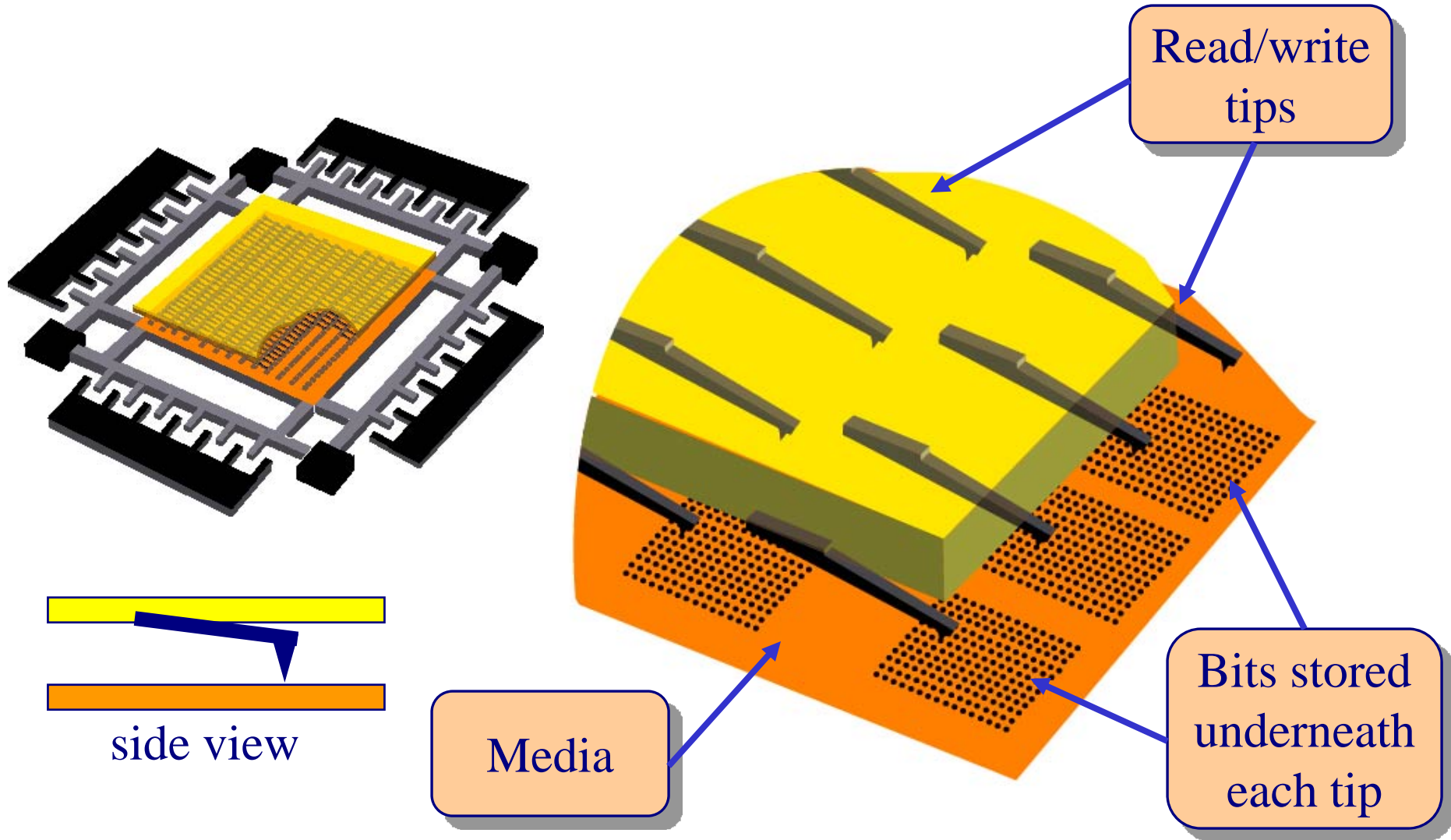
Active storage architecture with smart nodes (SN) organized into multiple smart storage groups (SSG) for disk- and MEMS-based systems interconnected via the InfiniBand® storage network.

Architecture

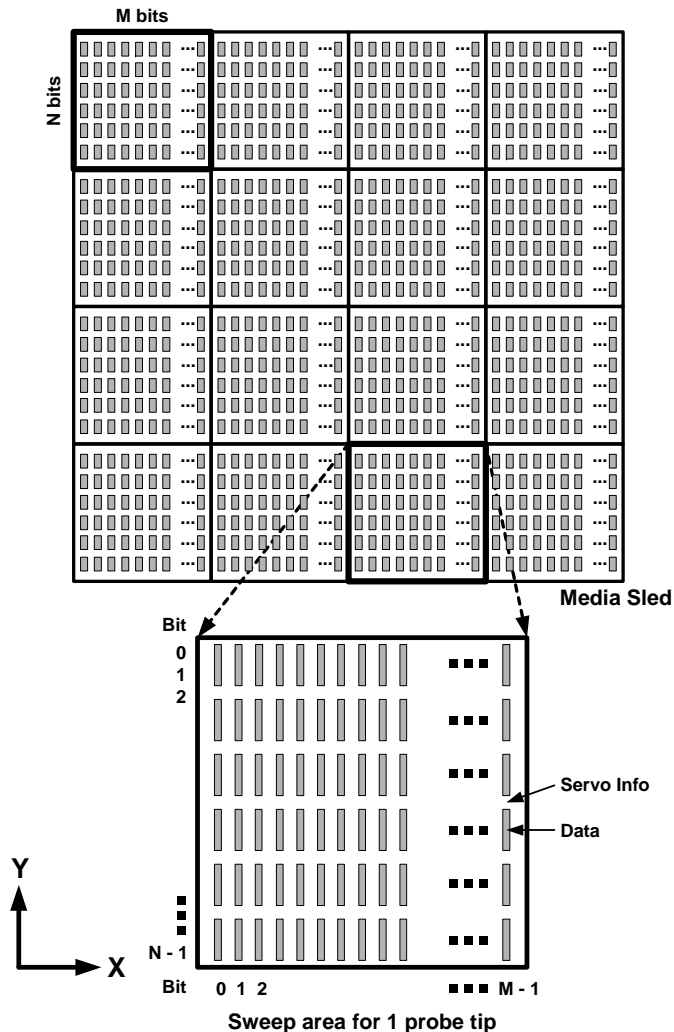


Courtesy of the Carnegie Mellon University CHIPS Research Project,
URL: <http://www.lcs.ece.cmu.edu/research/MEMS>

Architecture



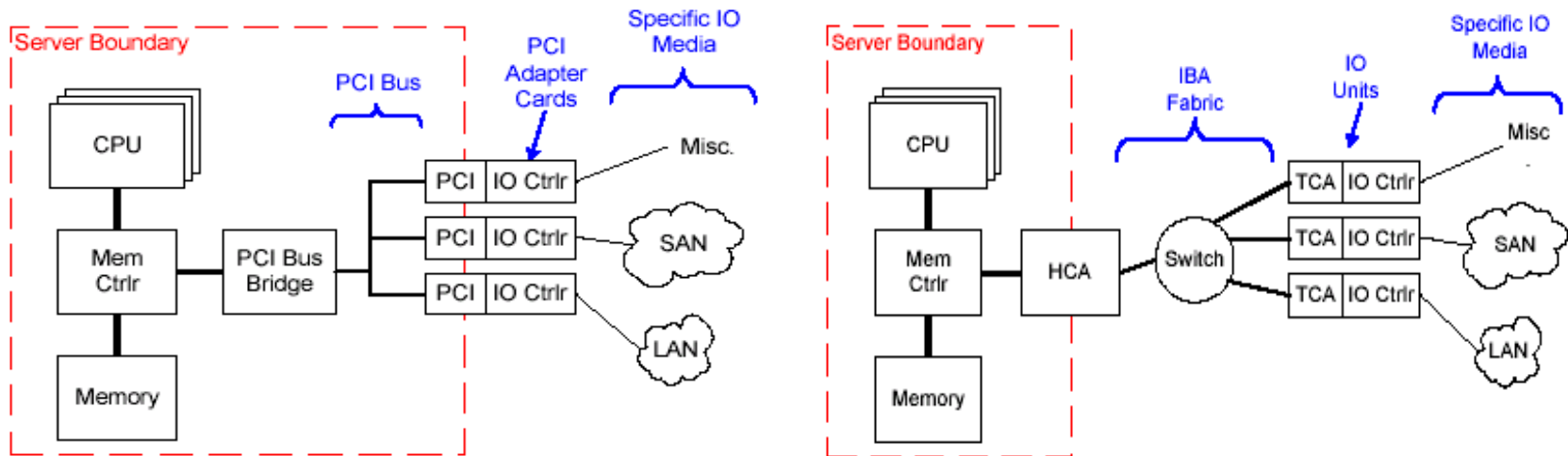
Architecture



MEMS-based storage devices:

- Made from photolithographic processes
- Moving rectangular media sled
- Array of READ/WRITE tips
- Seek in X direction, access in Y direction
- Room for new data placement designs
- Room for new I/O scheduling algorithms

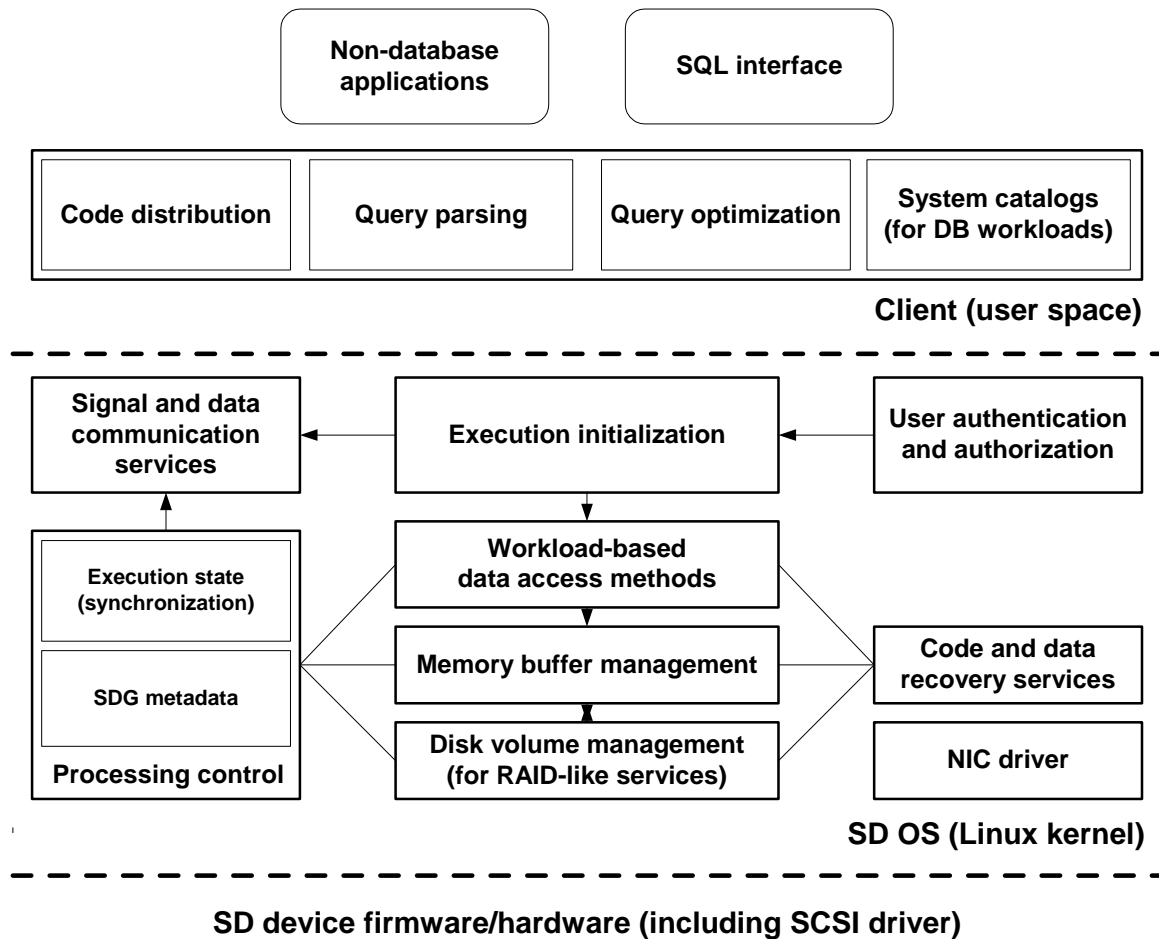
Architecture: Interconnect



InfiniBand switched fabric replaces PCI shared bus for higher interconnect bandwidth and lower latency.

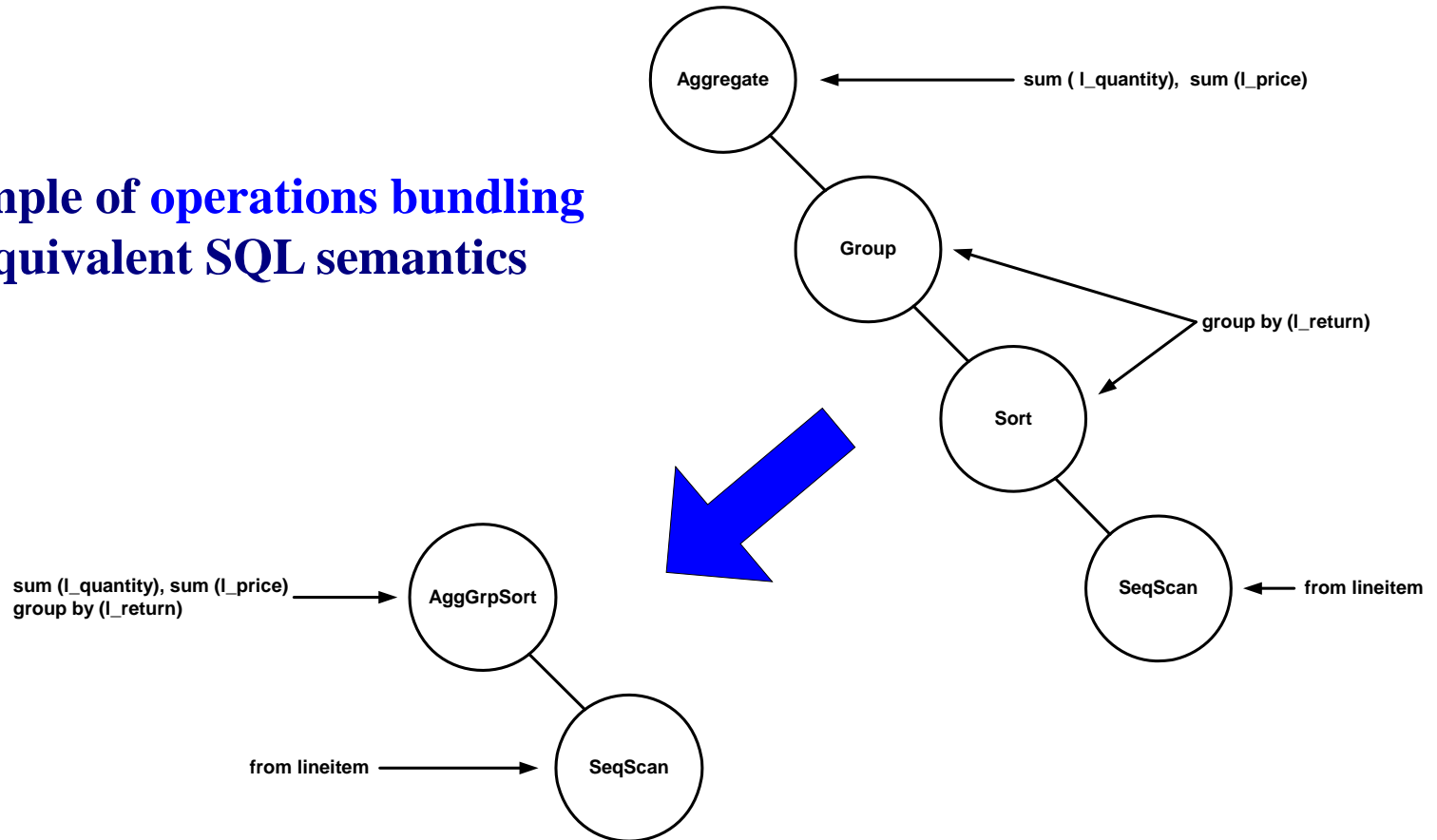
Cascaded switches provide better scalability than PCI buses since TCA is no longer within the server.

Architecture: Software

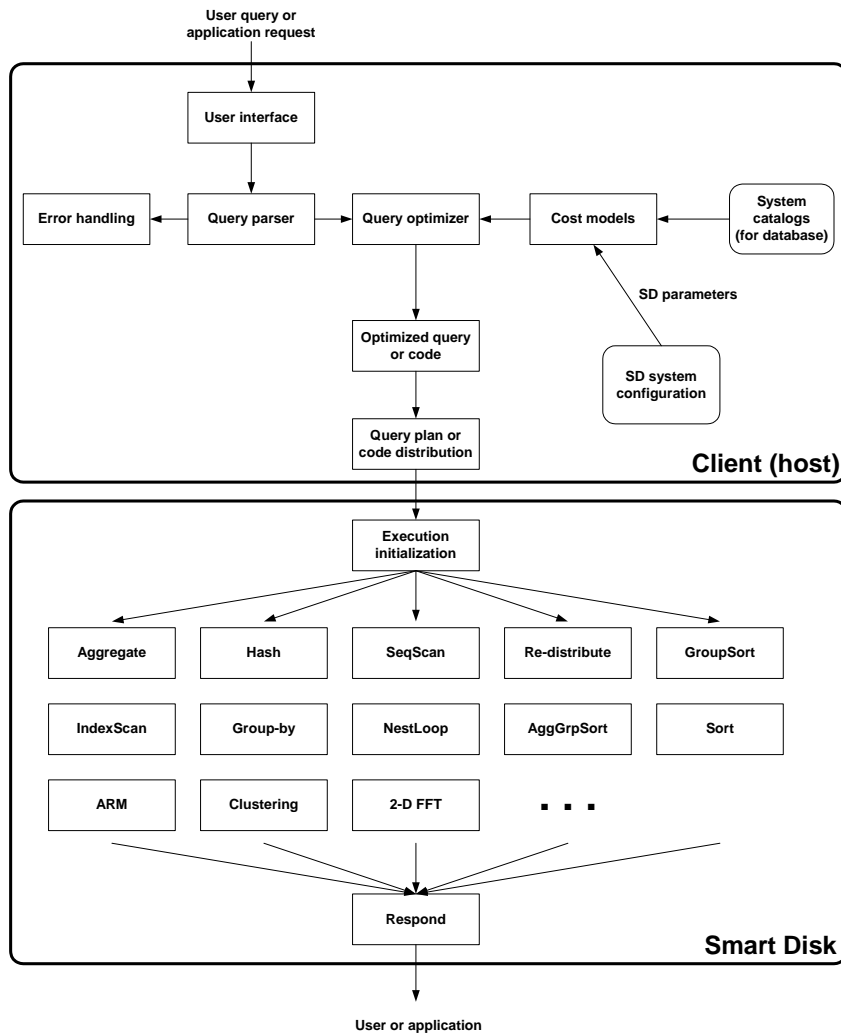


Architecture: Optimized Processing

Example of operations bundling for equivalent SQL semantics



Architecture: Software



Processing model:

- Based on the SD software architecture
- For DB and more general workloads

Device Model

MEMS-Based vs. Disk-Based Storage Systems

Processing model for SD system remains applicable, except for device-specific parameters R_{read} and R_{write} . Replace with MEMS values.

Parameter	G1	G2	G3
bit width (nm)	50	40	30
sled acceleration (g)	70	82	105
access speed (Kbits/s)	400	700	1000
X settling time (ms)	0.431	0.215	0.144
total number of tips	6400	6400	6400
number of active tips	640	1280	3200
max throughput (MB/s)	25.6	89.6	320
number of media sleds	1	1	1
per-sled capacity (GB)	2.56	4.00	7.11
bi-directional access	no	yes	yes

MEMS G1, G2 and G3

Parameter	Value
RPM	6,400
max bandwidth (MB/s)	10
avg seek time R/W (ms)	9.2/17
number of data surfaces	18
number of cylinders	2630
sector size (bytes)	512
diameter of disk	3.5"
height of disk	1.63"

HP C2490A Disk

Recovery Schemes

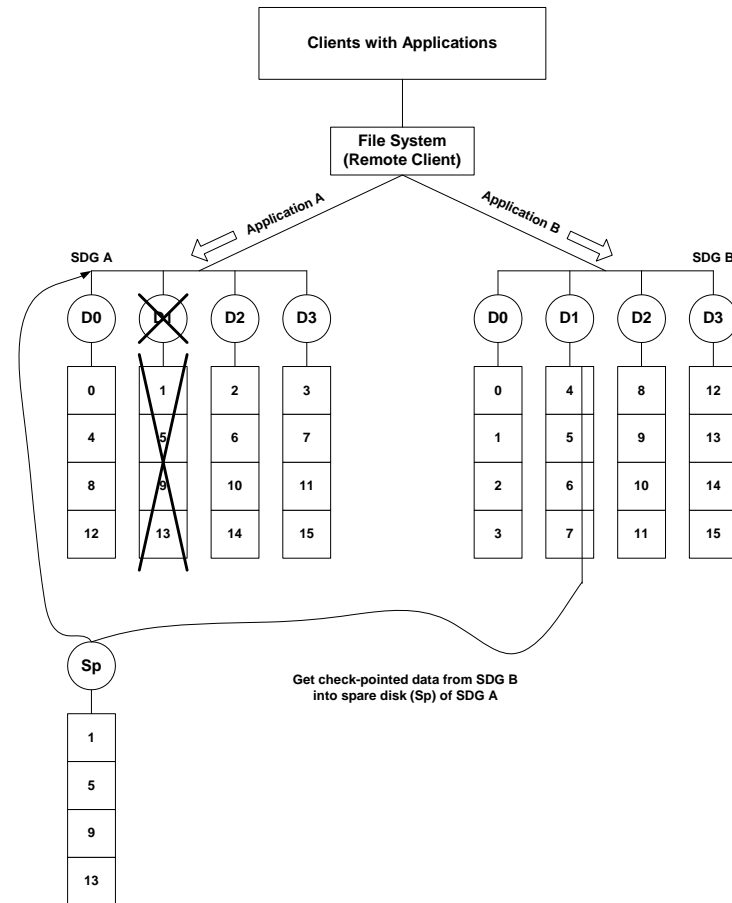
Mirroring with Spare Storage

Reliability:

- D1 of SDG A faults
- Send last check-point data from D1 of SDG B to Sp of SDG A
- Processing resumes with D0, D2, D3 and Sp of SDG A

Performance:

- SDG A and SDG B each processes different access pattern workloads
- READ from either SDG A or SDG B for faster I/O access
- WRITE must be performed to both SDG A and SDG B



Recovery Schemes

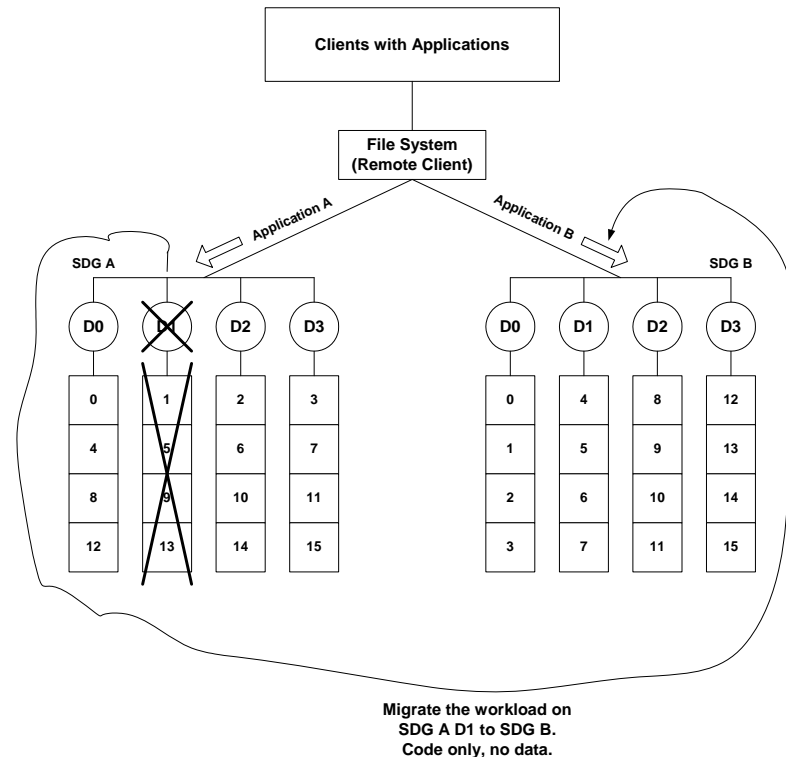
Mirroring with Workload Migration

Reliability:

- D1 of SDG A faults
- Migrate the workload (code only) from SDG A to SDG B
- Processing continues with SDG B while D1 of SDG A is repaired

Performance:

- SDG A and SDG B each processes different access pattern workloads
- READ from either SDG A or SDG B for faster I/O access
- WRITE must be performed to both SDG A and SDG B
- No need to migrate check-point data so recovery cost is reduced



Recovery Schemes

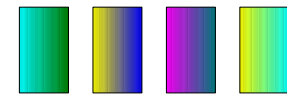
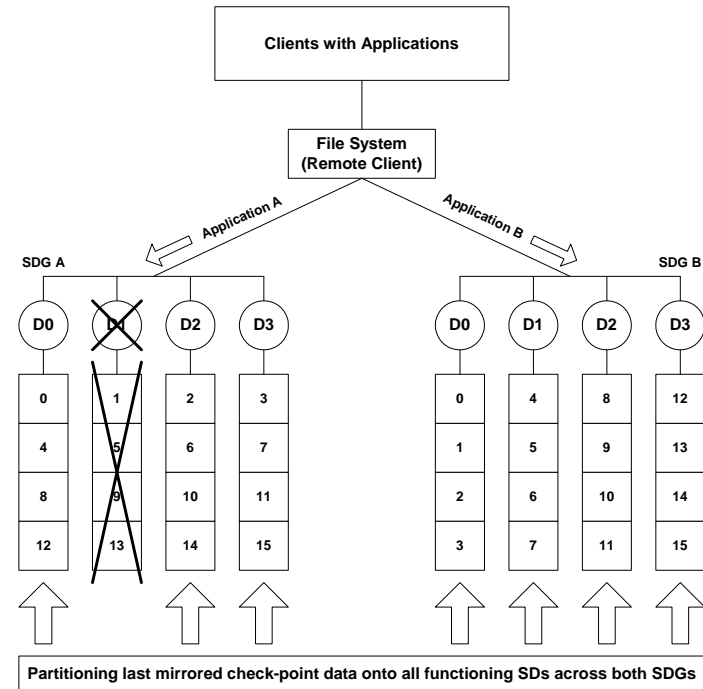
Mirroring with Workload Re-distribution

Reliability:

- D1 of SDG A faults
- Re-distribute the last check-point data among SDG A and SDG B
- Processing resumes with remaining SDs within SDG A and SDG B

Performance:

- SDG A and SDG B each processes different access pattern workloads
- READ from either SDG A or SDG B for faster I/O access
- WRITE must be performed to both SDG A and SDG B
- Amortize cost of recovery with reduced workload on every SD



Re-distribute data from the last checkpoint on SDG A and SDG B except D1 of SDG A. Then re-send the code and resume work.

Recovery Schemes

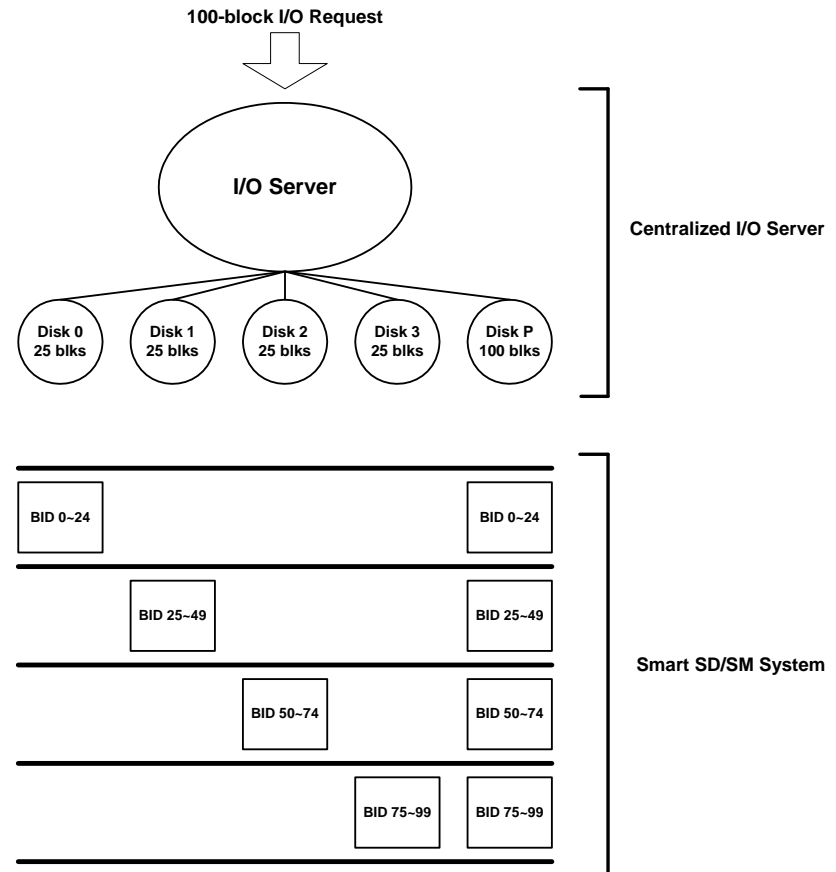
Parity with Dedicated Single Parity Device

Reliability:

- D1 of SDG faults
- Recover data by XORing D0, D2, D3 and DP (onto a spare, Sp)
- Processing resumes with D0, D2, D3 and Sp after recovery

Performance:

- Optimal speedup of 4 with more conservative system size
- Communication cost is $50*4$ compared to $75*4$ for servers
- Parity disk becomes the bottleneck for I/O and communication



Workloads and Scenarios

Fault Recovery for DB Workloads:

Scenario 0 for database *scan*: normal operation

Scenario 1 for database *scan*: mirroring with spare storage

Scenario 2 for database *scan*: parity with single parity device

Scenario 0 for TPC-H Q_1 : normal operation

Scenario 1 for TPC-H Q_1 : mirroring with spare storage

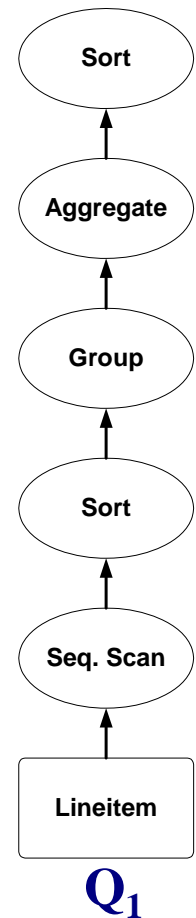
Scenario 2 for TPC-H Q_1 : mirroring with load migration

Scenario 3 for TPC-H Q_1 : mirroring with load re-distribution

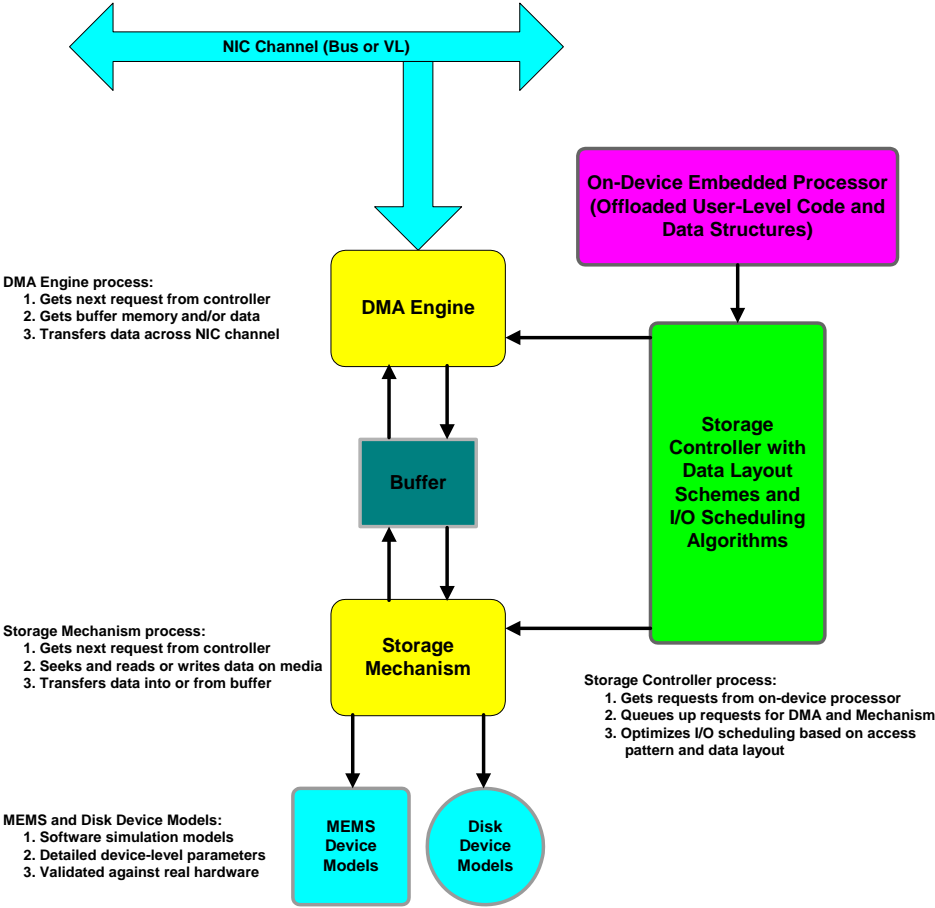
Scenario 4 for TPC-H Q_1 : parity with single parity storage

Special Considerations for TPC-H Q_1 :

Scenarios for Q_1 incorporate **check-pointing**, i.e. updating mirror or parity SDG/SMG with intermediate results at the granularity of **one primitive**. Thus, post-recovery processing backtracks by **1 stage** only.



Simulation Setup



Simulation Structure for a Single SN

Simulation Platform & Tools

Platform:

Cluster of 9 500-MHz Pentium III Linux PC, 64 MB on-device RAM, 9 GB local disk space, and Ethernet

Input data generator:

dbgen: populates TPC-H tables at scale factor 1.0 (~ 1 GB size)

Performance Components:

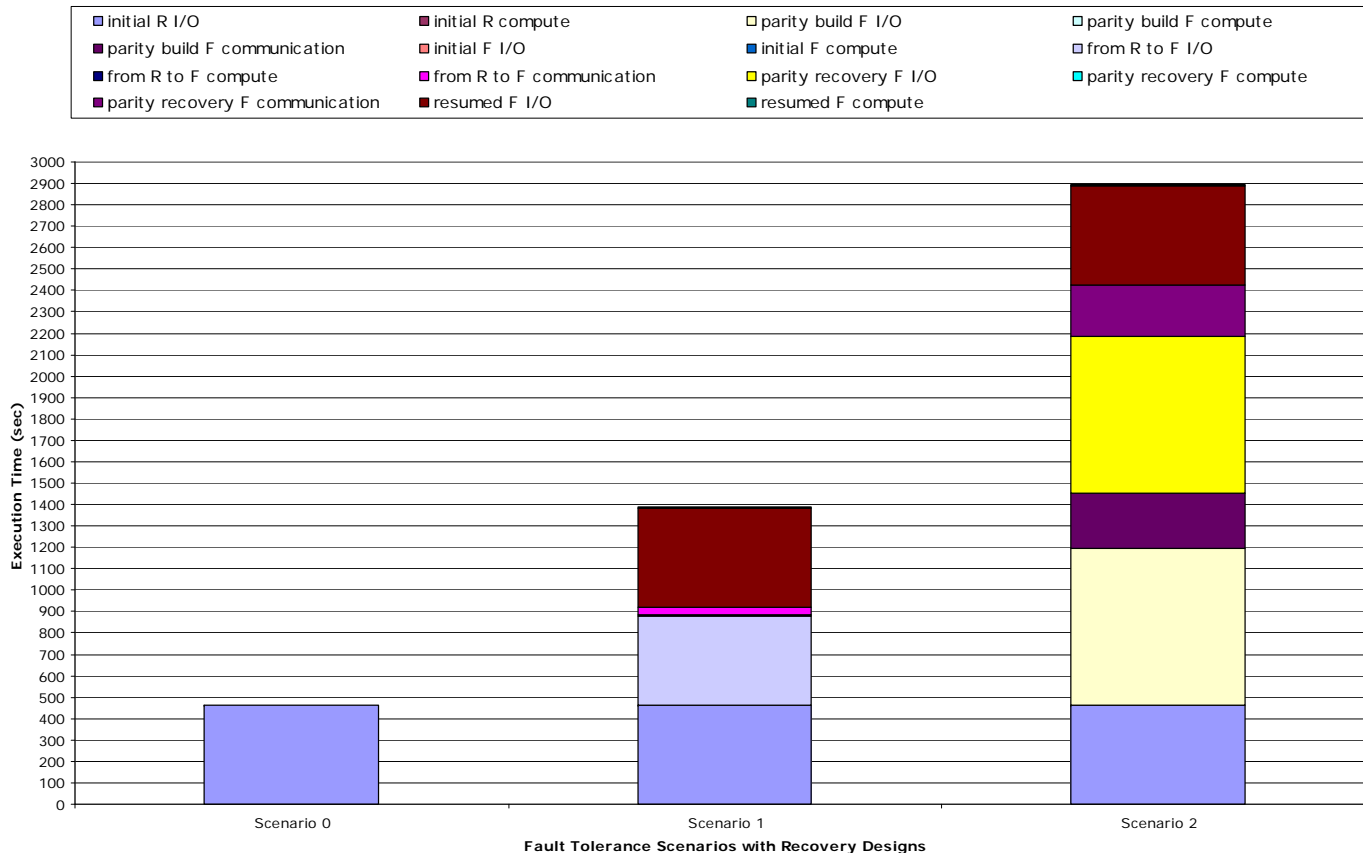
I/O: simulated with *DiskSim 3.0* for disks and MEMS storage

Computation: measured with timers with MPI_Wtime() calls

Communication: measured with timers with MPI_Wtime() calls

Preliminary Results

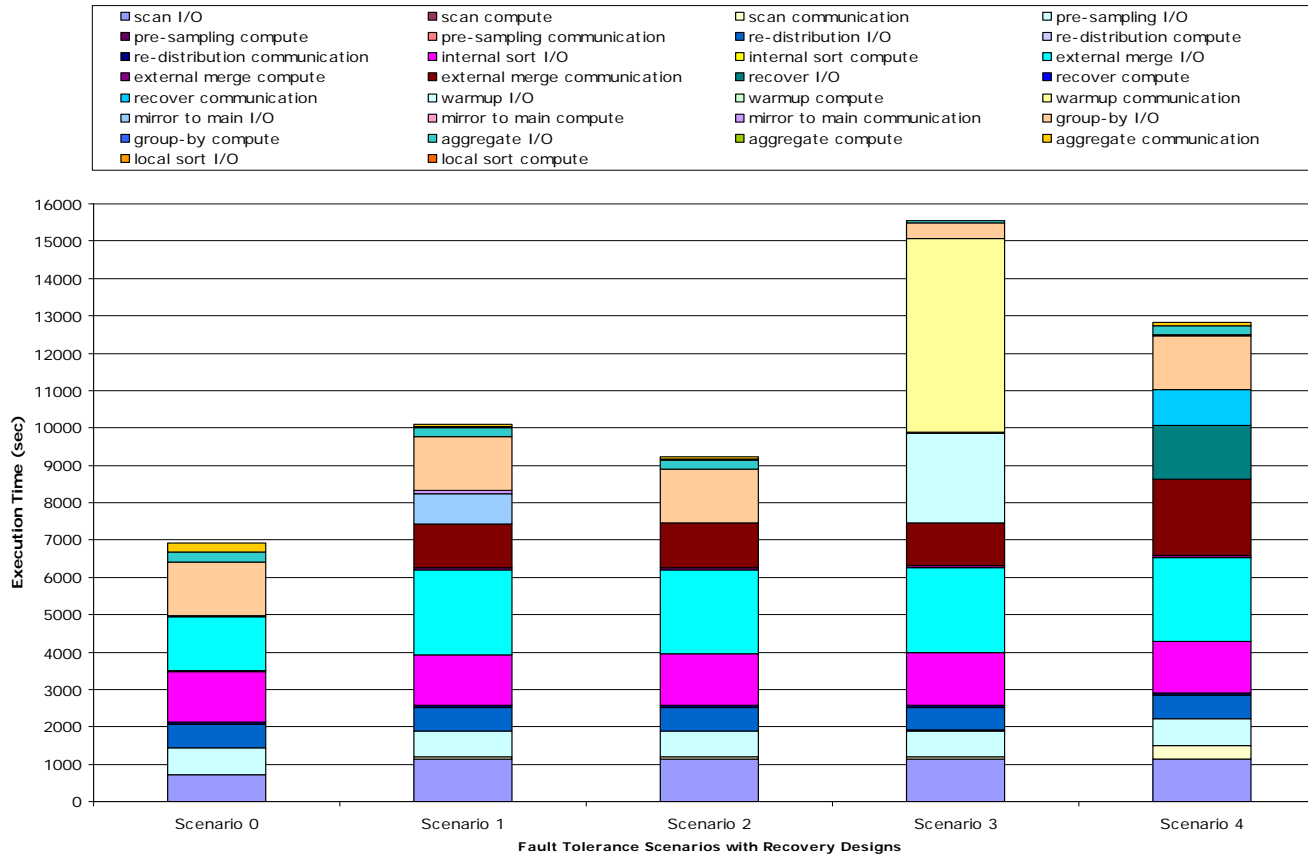
Scan Scenarios 0, 1 and 2 for HP C2490A Disk-based Systems, SF=1.0



Scenario 2 (parity scheme) incurs higher I/O and communication costs but requires smaller system size than Scenario 1 (mirroring scheme).

Preliminary Results

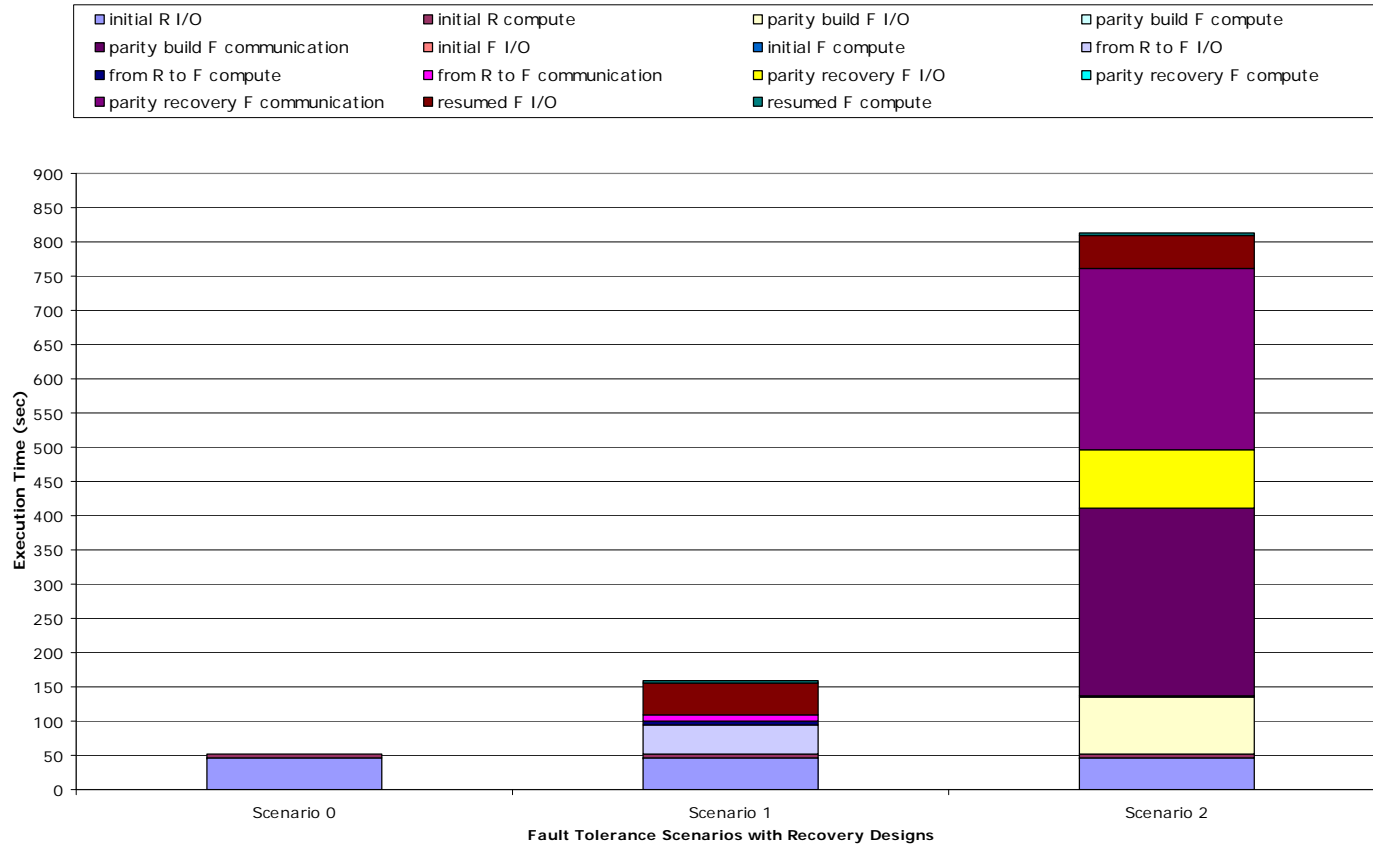
TPC-H Q1 Scenarios 0, 1, 2, 3, 4 for HP C2490A SD Systems, SF=1.0



Scenario 3 (mirroring + load re-dist) has network-limited characteristic while reducing post-recovery *Group-by I/O* by ~ 75% to 80%!

Preliminary Results

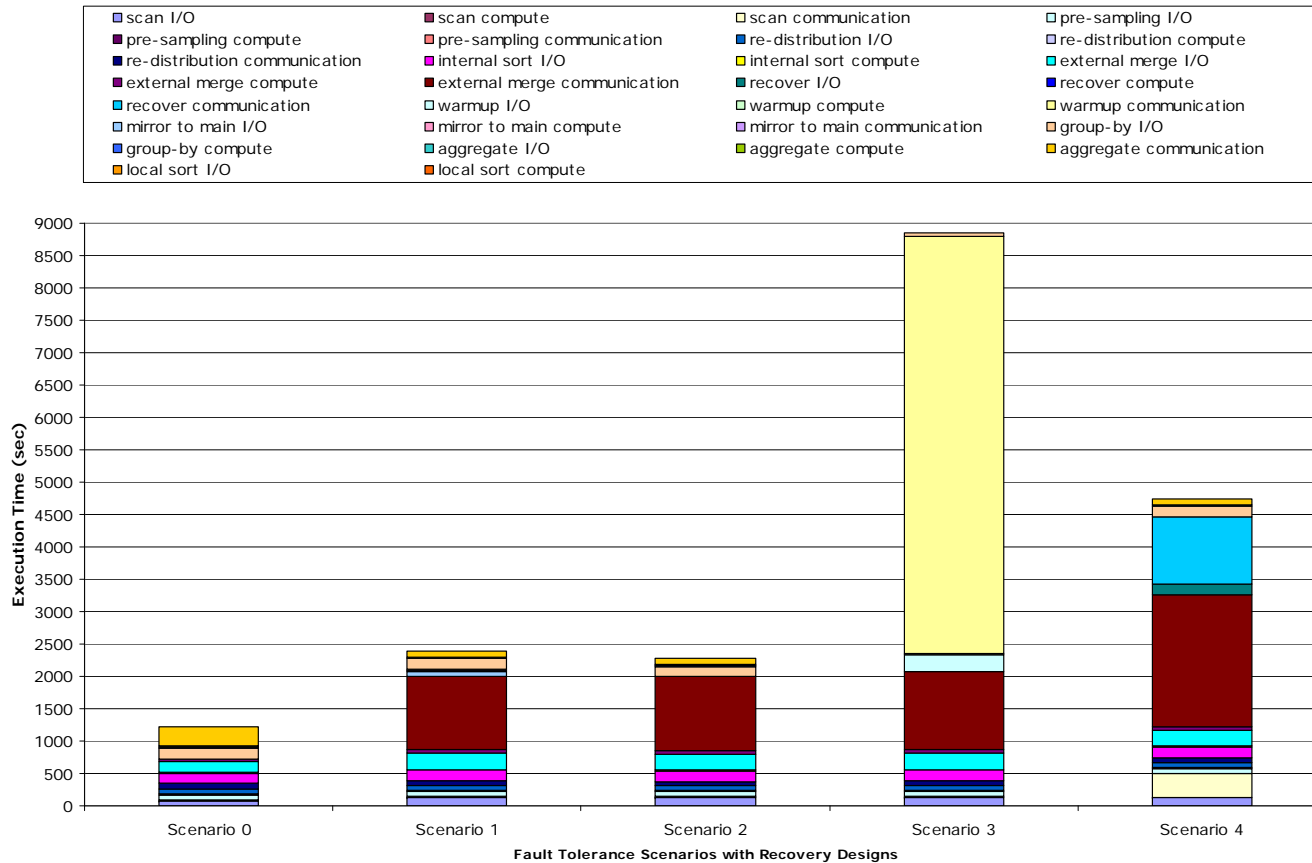
Scan Scenarios 0, 1 and 2 for G3 MEMS-based Systems, SF=1.0



Scenario 2 (parity scheme) shows network-limited characteristics due to diminishing I/O cost. Call for high-performance interconnect!

Preliminary Results

TPC-H Q1 Scenarios 0, 1, 2, 3, 4 for G3 SM Systems, SF=1.0



Scenario 3 (mirroring + load re-dist) screams for faster interconnect!
Scenario 2 appears desirable since only code (no data) is migrated.

Summary

**Computation in storage
brings processing closer to data
and reduces interconnect traffic,
thus higher system performance.
Disks and MEMS storage impact
fault tolerance capabilities for
active I/O systems significantly,
as are storage interconnects.**

Future Work

- **Code Offloading & Partitioning:**
 - An optimization problem
 - MEMS: I/O scheduling + data layout
- **Impact of InfiniBand and MEMS:**
 - Effect on the recovery schemes
 - Both device and system levels
- **Holistic Simulation Environment:**
 - Extension to *Pantheon* and *SimOS*
- **Smart Ubiquitous Computing:**
 - Integrated processor-memory-storage

Thank You! ☺