HEWLETT PACKARD

Mime: a high-functionality shadow writing disk system — Slides presented at BASS–29

David M. Jacobson

Hewlett-Packard Laboratories
Concurrent Systems Project

HPL–CSP–~~91~~92–15, April 29, 1992

Mime is a disk system using shadow writing to both improve performance and enable new features. One such feature is "rollback groups." Rollback groups provide a means to specify points at which a transaction is consistent. After a failure, the state will be restored to the last such point. In addition, roll back groups can be either committed or aborted, and thus provide support for the transactions.

Mime: a high-functionality shadow-writing disk system

You must love Mime! — *Richard Wagner*

Chia Chao, Robert English, David Jacobson,
Alexander Stepanov and John Wilkes

Hewlett-Packard Laboratories
Palo Alto, CA

Outline

- Technical summary
- Functions
- Design
- Status and future work

Technical Summary

"Update in place is a poison apple" — Jim Gray

"One can solve any computer science problem with an extra level of indirection" — ancient proverb

"Keep an extra copy of your metadata with the data" — Butler Lampson

"Do not invent new abstractions" — John Ousterhout

Technical summary—reprise

Mime is a shadow writing disk array.

- ❑ It keeps a map of logical to physical addresses.
- ❑ It stores logical addresses at physical address.
- ❑ It allows multiple logical blocks to be mapped to the same physical location.
- ❑ It provides careful atomicity
- ❑ It provides provisional command streams and UNDO
- ❑ It allows for multiple versions of the same logical block.

Basic interface

- ❑ Mime looks like a (SCSI) disk
- ❑ It has two main commands:
  - ◆ write (block#, data)
  - ◆ read (block#) -> data

Provisional command streams

- Transactions - locking = rollback groups
- Data isolation
- Atomic updates
- Save points
- Undo capability

Failure Semantics

- All writes are atomic - no partial overwrites;
- Permanent command stream:
  - prefix semantics
  - best effort.
- Provisional command streams:
  - Transaction support
  - Consistent views

Data links

- "Copy-on-write" for data
- Multiple pointers
- Consistent data views
  - Backup
  - DBMS queries

Sparse address space

- Simplifies host data management
- Preserves locality information
- Requires space management primitives

Competitive performance

- 15% better than normal disks on typical OS trace
- 2 to 4 times faster on random writes
- Read optimizations

Basic algorithm

Data Structures

- Index
- Free map

Read

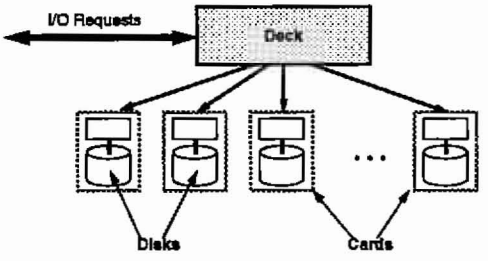
- Looks up address in index; read data

Write

- Select free segment
- Write data, installing mapping in index.
- If segment was previously mapped, free old physical segment.

System Structure

- Deck contains index
- Card contains free map and disk



Two stage recovery

- Incremental checkpoints of index
  - Recovers to a known state
- Scan of free segments
  - Detects operations since last checkpoint

Support for recovery

On every write:

- Writes tag with data
  - Tag is <logical address, sequence number>
- Free operations are delayed
  - Freed segments go into tentative free list
  - Prevents overwriting by subsequent writes

Periodically:

- Deck checkpoints index
- Card marks tentatively free segments free

Crash recovery

- Index restored from checkpoints
- Free map rebuilt
- Free segments scanned for recent updates
- Updates applied in sequence
- Updates after gaps ignored
  - Supports atomic updates on arrays

Incremental Checkpointing

- Index divided in k partitions
- k+1 meta-segments reserved on disk
- Memory log of index changes
- Partition and the log written together

Checkpoint Recovery

- For each meta-segment in time order
  - Restore partition
  - Apply memory log to entire index

Rollback groups: data structures

Deck:

- Auxiliary index—one per rollback group
- Global operation log

Card:

- Reference counts

Segment:

- Tag = <block number, sequence number, rollback group number>

Rollback groups: algorithms

New rollback group:

- Create group index

Read:

- Deck looks up address first in group index, then in global index

Write:

- Deck installs mapping in group index.

Commit:

- Merge group index into global index
- Put commit record into operation log

Abort:

Rollback groups: crash recovery

- At top level when recovering from free area of disk
  - Find first gap in sequence numbers
  - Recover only to here
- In rollback group when recovering from free area of disk
  - Back up to barrier

Sparse address space

- Sparse data structures
- Space management
  - Free command
  - Find block command
- Free commands go to operation log

Status

- Basic functionality working
  - Single-node Inmos T800 Transputer
  - Hewlett-Packard 5.25" SCSI disks
- Recovery being tested

Future work

- Match functionality and user needs
- Evaluate for OLTP and filesystem

Conclusion

- Rollback groups
- Prefix semantics
- Atomicity
- Consistent views
- Competitive performance