Managing High-Bandwidth Real-Time Data Storage

David Bigelow, Scott Brandt, John Bent, HB Chen

Systems Research Laboratory
University of California, Santa Cruz

Los Alamos National Laboratory
High-Bandwidth Streaming Data

MB/s
High-Bandwidth Streaming Data

GB/s
High-Bandwidth Streaming Data

TB/s
TiVo on Big Data

Attach THIS...

...to THIS
Questions

- High-Bandwidth Streaming Data
  - How to capture it in real time?
  - How to manage it?
  - How to search it?
  - How to guard against hardware failure?
  - How to control a large system?
  - How to reach high performance?
Goals

- Develop techniques to store, retrieve, manage, and safeguard high-bandwidth real-time data
- Integrate data management into core storage system functionality
- Extend quality of service guarantees in the storage domain and show how disk drives can be used to make hard performance guarantees
Standard Storage System

Client

Interface Layer

File System

Storage Subsystem

Physical Hardware

Database Network

RAID
Mirroring
Remapping

Rotational Drives
SSDs
Integrated Storage System

- Client
- Interface Layer
- File System
- Storage Subsystem
- Physical Hardware

Integrate into one unit
Existing Systems

- Modular storage design
- Poor quality of service abilities
- Little hardware integration
- **Different design goals:**
  - Long-term storage
  - Flexible file structure
  - Support for multiple concurrent clients
Existing Approaches

- Available Bandwidth
- Standard Filesystem
- Database

Graph showing bandwidth usage over the position in the data cycle.
Ring Buffer Concept

- Fixed size
- Limited lifetime
  - Data automatically expires
- Highly predictable
- Limited indexing
- In-place preservation
Requirements

- Data management and information flow
- Performance guarantees with commodity hardware
- Real-time indexing and search
- Reliability mechanisms
- Control mechanisms
Data Management

- Handle small and large data elements
  - 100 MB chunks of binary sensor data
  - 20 byte IP packets

- Variable indexing complexity
  - Simple sequence (time) indexing
  - Multiple indices for each small element

- Manage data relationship
  - Related data from multiple sources
Disk Formatting and Consistency

- Minimal indexing and metadata requirements
- System never shuts down
- “File” structure: large and simple
- No need for on-disk index
Constraining Minimum Data Chunk Size

- **Definitions:**
  - A **Data Element** is a single piece of data to be treated as an indivisible unit
  - A **Data Chunk** is the minimum unit of the storage system itself, and may be composed of any number of data elements

- **Large data chunks are ideal**
  - Data aggregated into large blocks produce best I/O performance
  - “File” fragmentation is minimized
  - Disk head movement is minimized
Random and Continuous I/O

Random Blocks
- Multiple seeks
- Many rotations

Contiguous Blocks
- One seek
- One rotation at most
## I/O Example

### Hypothetical Drive Statistics
- 100 MB/s bandwidth (more or less)
- 10 ms seek time

<table>
<thead>
<tr>
<th></th>
<th>512 KB</th>
<th>1 MB</th>
<th>5 MB</th>
<th>10 MB</th>
<th>50 MB</th>
<th>100 MB</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>I/O Time</strong></td>
<td>15 ms</td>
<td>20 MS</td>
<td>60 ms</td>
<td>110 ms</td>
<td>510 ms</td>
<td>1010 ms</td>
</tr>
<tr>
<td><strong>Efficiency</strong></td>
<td>33%</td>
<td>50%</td>
<td>83%</td>
<td>90.9%</td>
<td>98%</td>
<td>99 %</td>
</tr>
<tr>
<td><strong>Time to complete</strong>&lt;br&gt;100 MB of I/O</td>
<td>3 s</td>
<td>2 s</td>
<td>1.2 s</td>
<td>1.1 s</td>
<td>1.02 s</td>
<td>1.01 s</td>
</tr>
<tr>
<td><strong>Time to complete</strong>&lt;br&gt;2 TB of I/O</td>
<td>17.47 hr</td>
<td>11.65 hr</td>
<td>6.99 hr</td>
<td>6.41 hr</td>
<td>5.94 hr</td>
<td>5.88 hr</td>
</tr>
</tbody>
</table>
Mechanical Disk Drives

- **Factors in I/O time (assuming a small I/O)**
  - Data Transfer Rate
  - Rotational Delay
  - Seek Time
  - Spin-Up Time

One revolution will bring this length under the disk head:

- Inner
- Outer
Disk Profiling

- Each disk has a unique performance curve
- Performance degrades over the course of the disk
- Understanding these curves allows one to tune performance

![Graph showing read bandwidth in MB/s vs position in disk in GB for different drives.](chart.png)
Individual Performance Points

Disks may have regions of poor performance which impact performance goals.

(Likely) Remapped Region
Small Element Indexing

- Data sorting and movement problem
  - Minimize movement on and off disk
  - Arrange indexing elements efficiently
- Actual search process is fast
- General case: indexing information approaches 100% of data
Multiple Indexes

- Index along multiple simultaneous vectors
  - Categorization by the most-used search aspects and stored in multiple formats
Architecture

Source -> Raw Data -> Processing

Processing -> Individual Elements -> Index

Index -> Chunk Assignment

Indexing Data -> Chunk Return

Indexing Data -> Query

Query -> Chunk Request

Chunk Request -> Chunk Return

Chunk Return -> Communicator

Communicator -> Chunk Request

Communicator -> Chunk Return

Chunk Return -> Outside Process

Outside Process -> Communication with Other Instances

One Instance

Disk

Data Chunk

I/O Manager

Writes

Reads

I/O Manager
Evaluation

- Two comparison systems
  - Flat filesystem (ext2)
  - Database (mysql)

- Multiple systems and hard drives
  - Results here are from one system/drive for comparison purposes

- Tests tweaked to suit base system ability
  - Synchronizing constraints looser on comparison systems
  - Data per cycle was about 80% of the drive's maximum capacity
Large Elements, Standard Filesystem

![Graph showing bandwidth performance over data processed.]
Large Elements, Valmar
Small Elements, Filesystem

![Graph showing bandwidth over data processed.]
Small Elements, Database
Small Elements, Valmar

![Graph showing bandwidth in MB/s vs data processed in GB, with two lines representing write and read bandwidths.](image-url)
Small Elements, Comparison

![Graph showing bandwidth over data processed, with lines representing different operations: Valmar, Writes; Valmar, Reads; Filesystem, Writes; Database, Writes.](image)
Query Times

![Query Time Graph]

- **Y-axis:** Seconds
- **X-axis:** Data Processed, GB

The graph shows the query times over different data processed, with two peaks at around 2000 GB and 3000 GB, indicating high query times during these data processing intervals.
Current and Future Work

- Automated Disk Profiling and Hardware-Aware Decision Making
- Real-Time Appropriate Reliability Mechanisms
- Control Systems
- Data Duplication
Acknowledgments and Questions

- Questions?

This work was carried out under the auspices of the National Nuclear Security Administration of the U.S. Department of Energy at Los Alamos National Laboratory under Contract No. DE-AC52-06NA25396. This work received funding from Los Alamos National Laboratory LDRD Project #20080729DR.