Venti: a new approach to archival storage[1]

Presented By: 陈子旸

Fudan University

13307130148@fudan.edu.cn

January 8, 2015

[1] powered by pandoc and XƎLaTeX

Outline

1. Overview
   - Abstract
   - Background
   - Venti
2. Data Organization
3. Application Example
4. Implementation
5. Performance
6. Conclusion

Abstract

- Venti: A network storage system intended for archival data
- A building block for a variety of storage applications
  1. logical backup
  2. physical backup
  3. snapshot file systems
- A block is identified by a unique hash of it's contents
- Enforce a write-once policy
- Duplicate copies of a block can be coalesced

Archival Storage

- Purpose
  - Store data for long periods of time (forever)
  - Data may not be needed frequently, but when it is needed it is often crucial

Prevalent Form

- Tape backup
  - Backup data to magnetic tape
  - (tar, ufsdump...)
  - Full backup vs Incremental backup
  - To provide backup as a central service for a number of client machines

Prevalent Form

- Snapshot
  - A snapshot is a consistent read-only view of the file system at some point in the past.
  - Each snapshot is a complete file system tree, much like a full backup.
  - A snapshot only requires additional storage for the blocks that have changed, like a incremental backup.
  - Always available and easy to access
  - Plan 9, WAFL, AFS...

Venti Archival Storage

- Goal: To provide a write-once archival reponository than can be shared by mutiple client machines and applications.
- Block level network storage system
  - Actually a backend storage for client apps
- Blocks addressed by hash of their contents
  - Use SHA-1 algorithm
  - Use hash value as its unique ‘fingerprint’
- Write-Once policy
  - Block once written, never modified
  - Modified blocks will have new address

Venti

Presented By:
陈子旸

Overview
Abstract
Background
Venti
Organization
Application Example
Vac
Phy Bak
Plan 9
Implement
Performance
Conclusion
Q&A
End

Why SHA-1?

- SHA-1 hash function is developed by NIST
- Output 160 bit hash values(20 bytes)
- Probability that there will be one or more collisions:

$$p \leq \frac{n(n-1)}{2} \times \frac{1}{2^b}$$

- Consider a large storage system contains 10^{18} byte of data stored as 8 Kbyte blocks($\sim 10^{14}$ blocks), the probability is less then 10^{-20}.
- Variants of SHA-1 can produce 256, 384 and 512 bit results for future use.

Venti Archival Storage

- Multiple clients can Share a Venti server
  - Hash function gives an unversal namespace
  - Duplication increases the utility rate of space
- Inherent integrity checking for data
- Caching is simplified
- Uses magnetic disk as storage technology
  - Access time comparable to non-archival data

Outline

1. Overview
2. Data Organization
3. Application Example
4. Implementation
5. Performance
6. Conclusion

Data Organization

- Data is divided into blocks and written to the server
- Pack the fingerprints into additional blocks, called pointer blocks, that are also written to the server
- Until a single fingerprint is obtained
- Applications can use such a structure to store a single file or to mimic the behavior of a physical device such as a tape or a disk drive

Data Organization

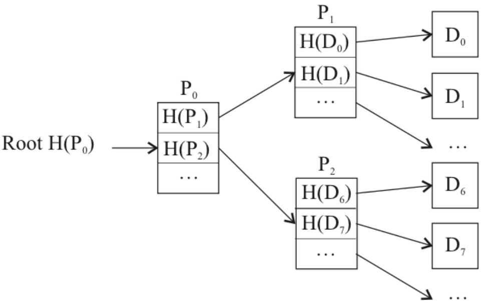


Figure 1. A tree structure for storing a linear sequence of blocks

Data Organization

- Venti does not allow such a tree to be modified
- But new versions of the tree can be generated efficiently by storing the new or modified data blocks and reusing the unchanged sections
- By mixing data and fingerprints in a block, more complex data structures can be constructed.
- For example, a structure for storing a file system may include three types of blocks:
  - Directory
  - Pointer
  - Data.

Data Organization

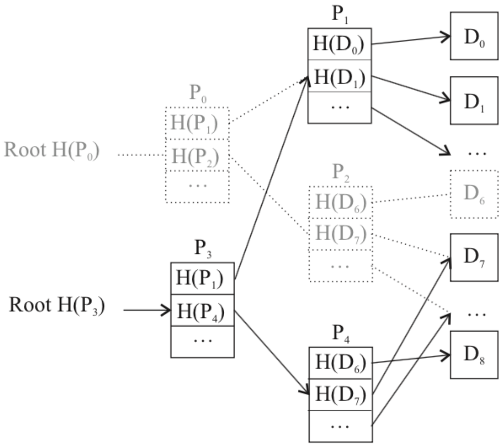


Figure 2. Build a new version of the tree.

Outline

1. Overview
2. Data Organization
3. Application Example
   - Vac
   - Physical Backup
   - Plan 9 File System
4. Implementation
5. Performance
6. Conclusion

Vac

- Vac is an application similar to tar and zip
  - With vac, Selected files will be stored as a tree of blocks on Venti server.
  - The output is always 45 bytes long, included a 20 byte root fingerprint.
  - ‘unvac’ enables user to estore files from a vac archive.
- Vac writes each file as a seperated collection of Venti blocks, which can coalesce duplicate copies of a file
- Incremental backups options can improve performance

Physical Backup

- Vac archive data at the file or logical level
- Alternative approach: block-level or physical backup
- Copy the raw contents of disk drives to Venti
- Coalescing duplicate blocks is the main advantage
- Can even mount a backup file system image from Venti
- Full restore can be done in a lazy fashing

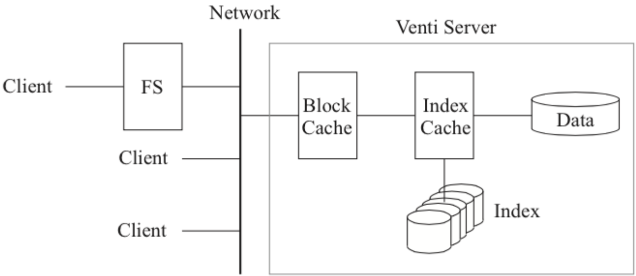
Plan 9 File System

- When combined with a small amount of read/write storage, Venti can be used as the primary location for data
- Plan 9 file system store snapshot on optical jukebox
- magnetic disks act as a cache for the jukebox
- New version of the Plan 9 file system uses Venti instead of an optical jukebox as its storage device

Outline

1. Overview
2. Data Organization
3. Application Example
4. **Implementation**
5. Performance
6. Conclusion

Implementation



Venti

Presented By:
陈子旸

Overview
Abstract
Background
Venti
Organization
Application Example
Vac
Phy Bak
Plan 9
Implement
Performance
Conclusion
Q&A
End

Implementation

- For data block
  - Use Append-only log
  - Blocks store on a RAID — 5 array of IDE disk drives
- For Index
  - Using a disk-resident hash table
  - Index is diveided into fixed-size buckets
  - Index store on 8 SCSI drives
- Additional work
  - caching, striping, write buffering

Format of Data Log

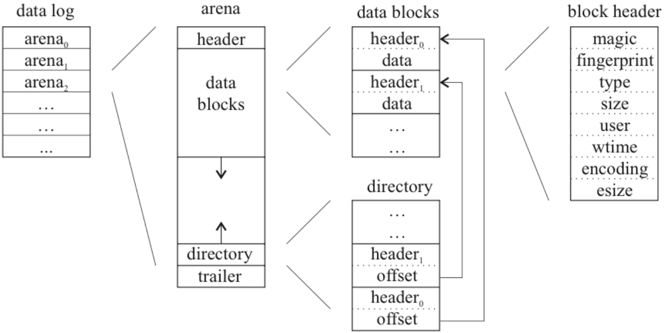


Figure 4. The format of the data log.

Format of Index

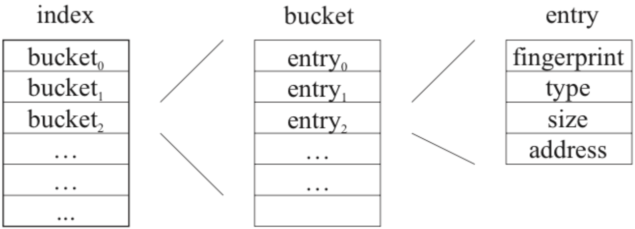


Figure 5. Format of the index.

Outline

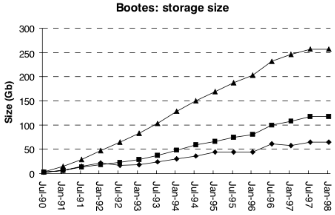
1. Overview
2. Data Organization
3. Application Example
4. Implementation
5. Performance
6. Conclusion

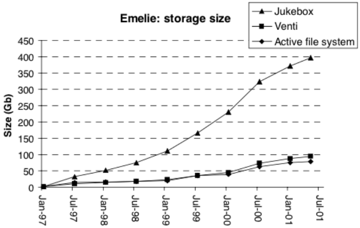
Performance

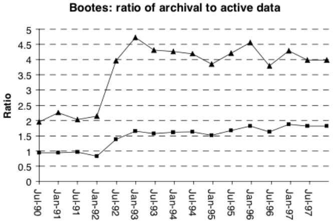
The performance of read and write in Mbytes/s :

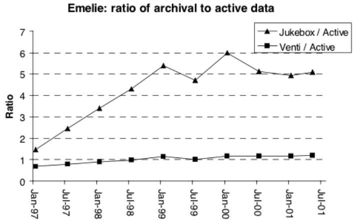
	sequential reads	random reads	virgin writes	duplicate writes
uncached	0.9	0.4	3.7	5.6
index cache	4.2	0.7	-	6.2
block cache	6.8	-	-	6.5
raw raid	14.8	1.0	12.4	12.4

Performance


Bootes: storage size


Emelie: storage size


Bootes: ratio of archival to active data


Emelie: ratio of archival to active data

Performance

The percentage reduction in the size of data stored on Venti :

	bootes	emelie
Elimination of duplicates	27.8%	31.3%
Elimination of fragments	10.2%	25.4%
Data Compression	33.8%	54.1%
Total Reduction	59.7%	76.5%

Outline

1. Overview
2. Data Organization
3. Application Example
4. Implementation
5. Performance
6. Conclusion

Conclusion

- Approach of identifying a block by SHA-1 hash is a well suited to archival storage
- Write-once policy of a block and ability to coalesce duplicate copies of a block makes Venti a useful building block for many interesting storage application
- By rapid groth in capacity of magnetic disks, it seems unlikely that archival data will be deleted to reclaim space

$$\Downarrow$$

Venti provides an attractive approach to archive data

Q&A

Any Questions?

End

Thanks For Attention!