Introduction to BTRFS

--- focus on usage

王坤山、王少岩
Taobao R&D Core-Sys

Content

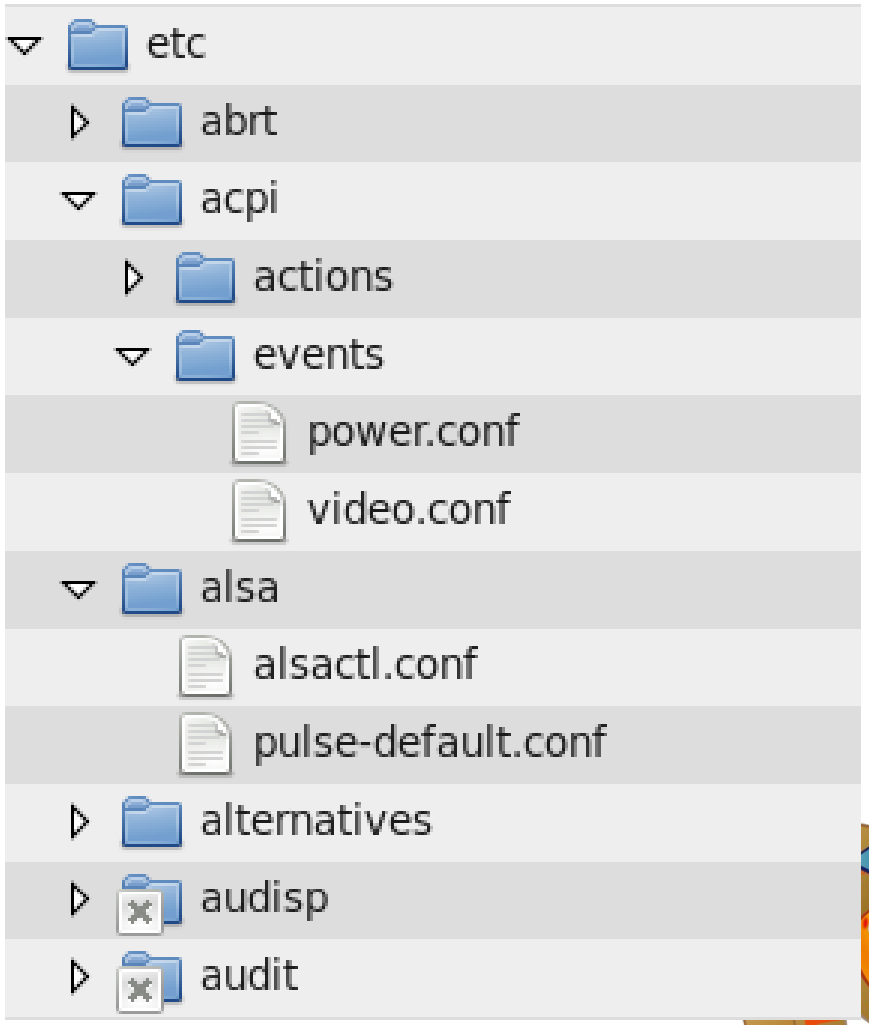
The Fundamental Functionality of File System

From end users' view

– File
The data constructed by DATA

– Directory
Contains other files and sub-directories

– Files creation/deletion, access permission management

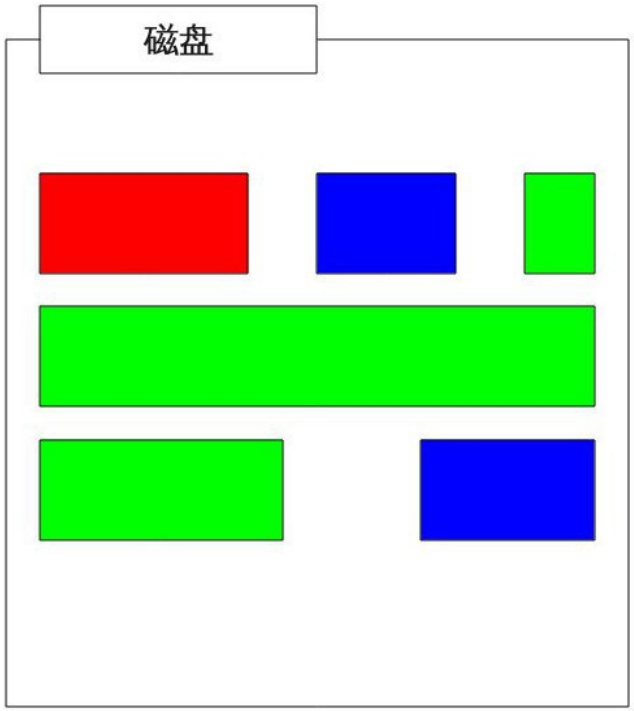


The Fundamental Functionality of File System (Cont.)

For hard disk file systems,

– <logical, physical> location mapping

– Hard disk space management



Limitation and Challenge of Existed Linux File Systems

Flexibility Limitation
- Max physical media capacity
- Max files number
- Max file size

Storage efficiency
- Big/small files

Access efficiency
- large directory

Challenge
– High concurrent workload for random I/O
– Huge storage space
– Application bound I/O patterns
– Security, stability, disaster recovery
– Etc

Limitation and Challenge of Existed Linux File Systems (Cont)

Status of existing Linux hard disk file systems,

– FAT
  Single link list storage
  Fragment and performance bottleneck

– Ext2
  Multiple level indirect block pointers
  – Faster random I/O
  – Linear directory lookup

– Ext3
  Journaling
  Faster file system check/recovery

– Ext4
  Extent based blocks management
  More efficient for big files storage

Limitation and Challenge of Existed Linux File Systems (Cont)

Challenges of existing file systems,

– Max file system size (PB level)
– Max individual file size (PB level)
– Higher I/O and storage efficiency for (small) files
– Online file system check
– Online high efficient file system defragmentation
– Faster directory lookup
– More flexible data/metadata allocation
– Cross physical volumes file system
– File system level data concurrency and redundancy
– Data and metadata check-sum

New File Systems Breakthrough

There are some breakthrough from new file system development efforts

– ZFS
  Copy-on-write, snapshot, EB (1024P) level large capacity

– LogFS
  Special optimization for flash based storage (SSD, Flash disk)

– ReiserFS
  Tree structure on disk
  Optimized for small files I/O and storage

– BTRFS
  The most promising candidate for next generation Linux hard disk file system
  The one this talk is focused on

BTRFS, Next Generation Linux File System Candidate

Larger storage capacity
- Max volume size 16EB
- Max file size 16EB
  (comparing Ext4 max file size 16TB)

Faster directory lookup
- Btree structure
  - for hashed directory entries order
  - inode order

Higher storage effiency
- COW (Copy On Write)
- Extent for large files
- Inline data for small files

Multiple device support
- Increase storage capacity and stability

BTRFS, Next Generation Linux File System Candidate (Cont.)

Snapshot
- Backup all files of a file system at a given moment
- No extra space consumed
- Faster, more convenient backup/restore

File compress
- For text based files, save more disk space

Checksum
- Data corruption detection

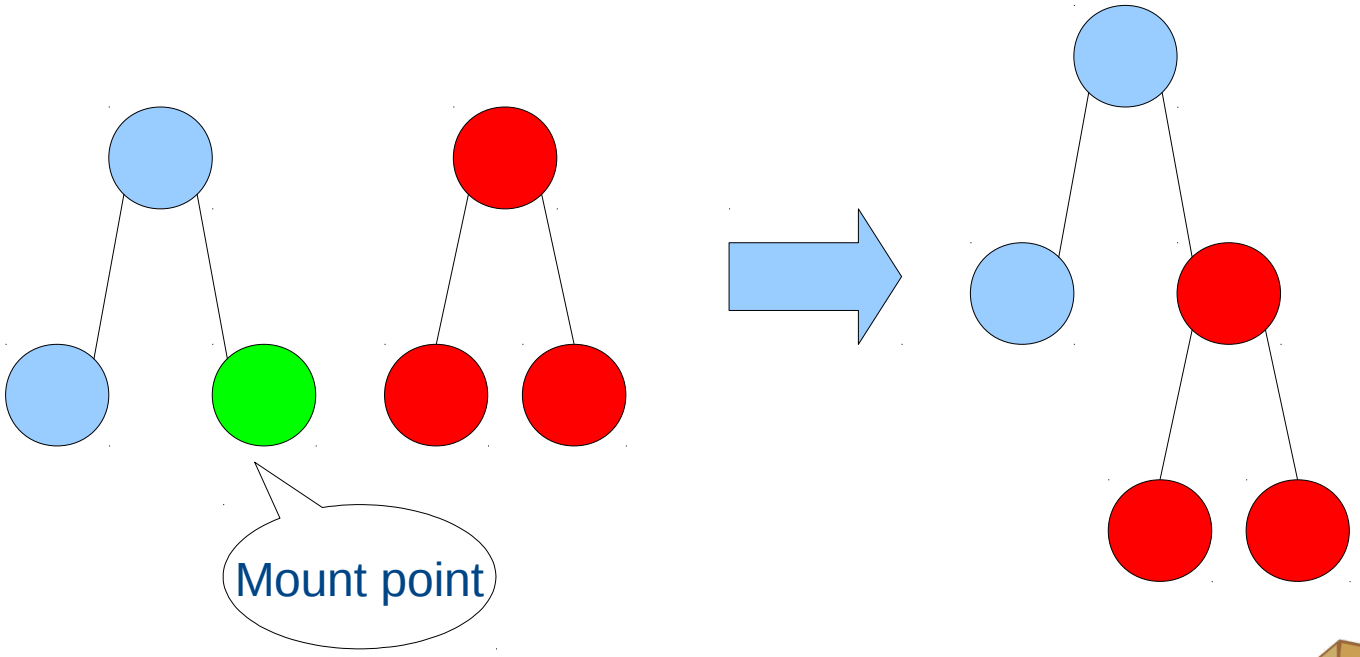
Online file system check
- No mandatory offline
- More flexible task schedule
- Run time data consistency check

BTRFS, Next Generation Linux File System Candidate (Cont.)

Optimization for Solid State Disk
- Wear-out leveling
- Delay write-back for large (2M) SSD data chunk

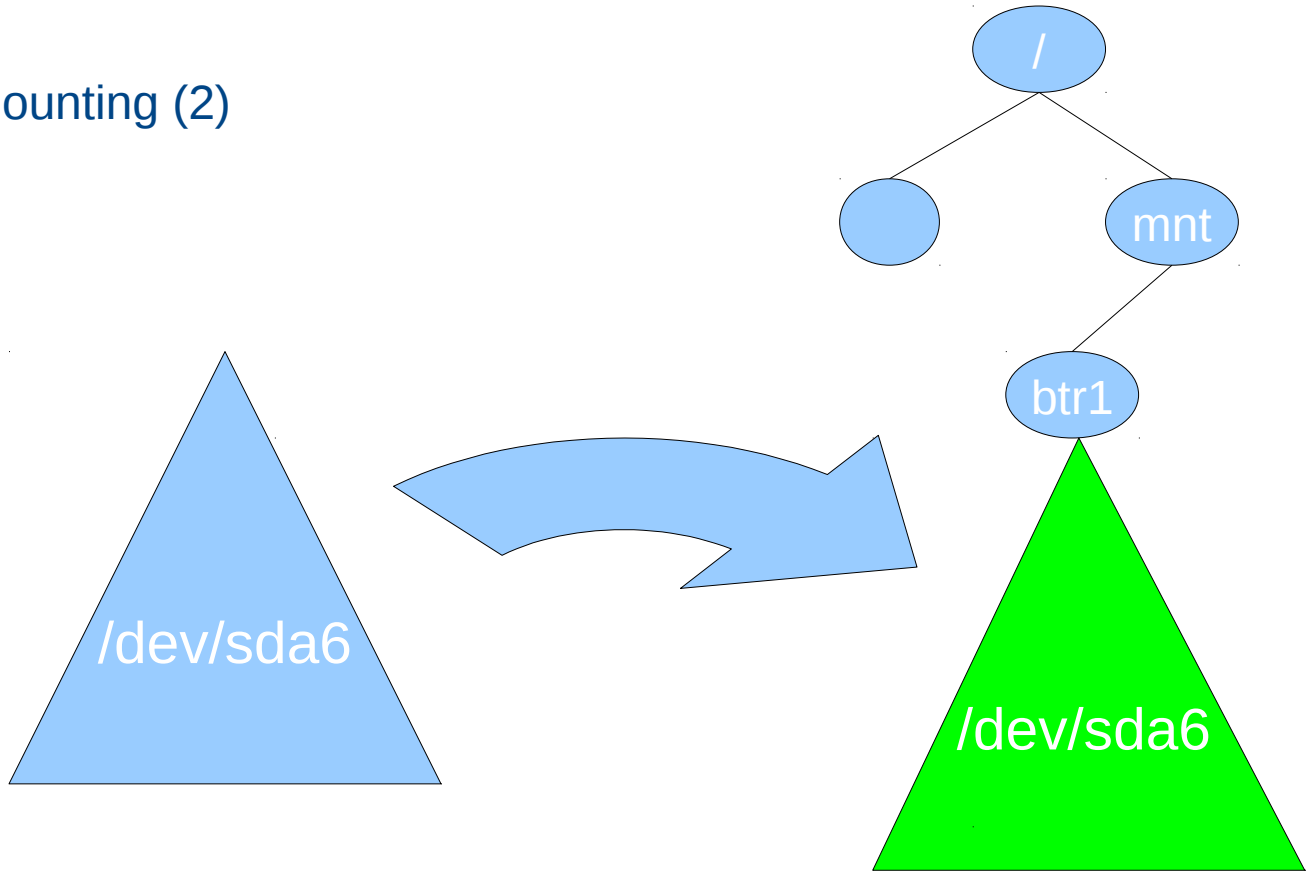
Concepts of BTRFS

Mount point



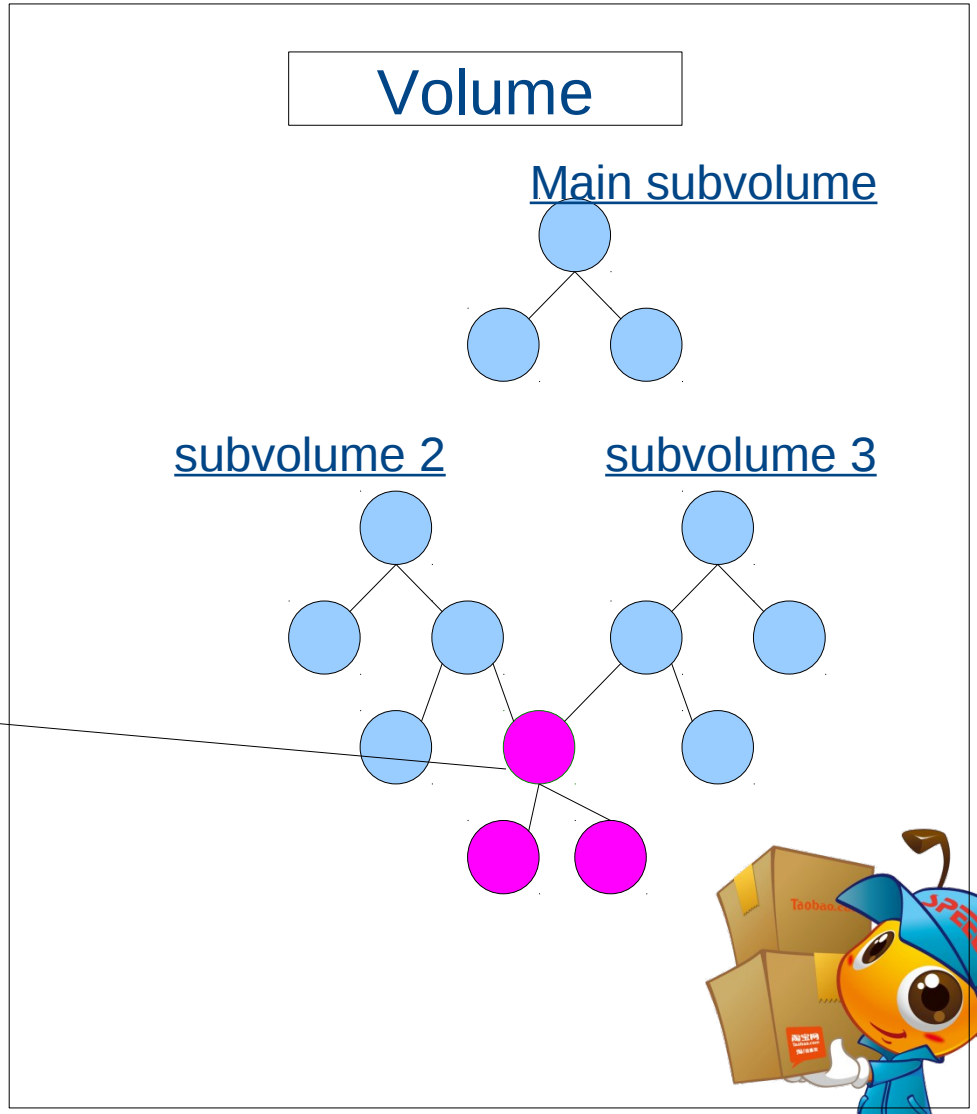
Concepts of BTRFS (Cont.)

Mounting (2)



Concepts of BTRFS (Cont.)

Subvolume
– like directory
– can contain files and directories
– can be created as directory



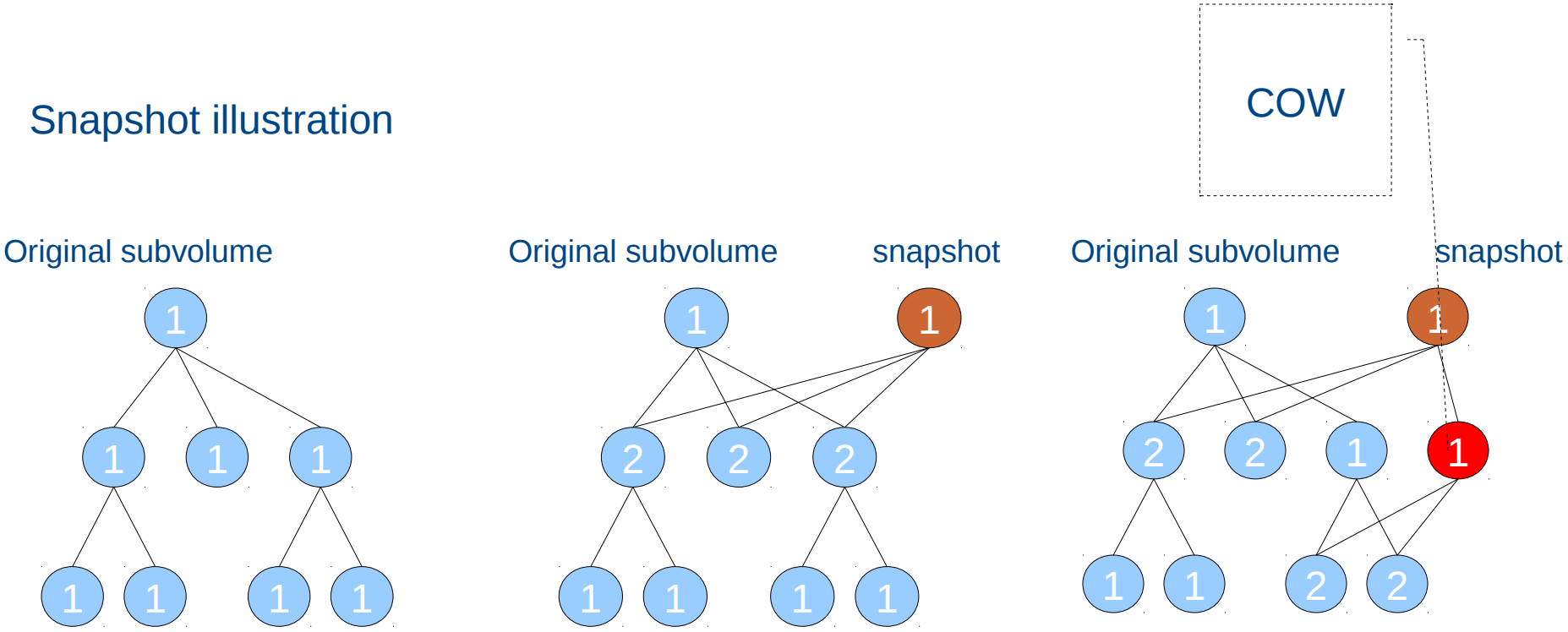
Concepts of BTRFS (Cont.)

Snapshot
- What we want ?
  - Complete backup
  - Accessible and recoverable
  - Frequently access

BTRFS' Implementation
- Snapshot is a subvolume
- Created by copying another subvolume
- Share most data and metadata
- Writable snapshot (Copy On Write)

Concepts of BTRFS (Cont.)

Snapshot illustration



BTRFS Usage & Live Demo

– Create file system
– Mount & umount
– Subvolume create, delete
– Snapshot create
– Writable snapshot
– Compress
– Multiple device support

Exploring BTRFS from Taobao

Most promising Linux disk file system for enterprise environment.

We are working on a research between existing BTRFS features and application requirement from Taobao's deployment.

More upstream efforts to make BTRFS to be a enterprise usage ready file system.

Exploring BTRFS from Taobao (Cont.)

Something we can improve to make BTRFS fit Taobao's requirement

Subvolume Compress
- Requirement
  - automatic compress when place data in a specific direcotry
  - friendly usage experience for end user and developers

- Current status
  - global compress option provided
  - no per-subvolume compress supported
  - OPPORTUNITY

Exploring BTRFS from Taobao (Cont.)

Something we can improve to make BTRFS fit Taobao's requirement

Metadata on SSD
- Requirement
  - Metadata allocation on dedicated SSD

- Current status
  - Metadata on first device of file system
  - Can not prevent data allocation on a specific device
  - Single point of failure problem
  - OPPORTUNITY

Exploring BTRFS from Taobao (Cont.)

Something we can improve to make BTRFS fit Taobao's requirement

Underlying Media Topology Aware allocation
- Requirement
  - RAID, 4K sector hard disk ...

- Current status
  - not implementated yet
  - OPPORTUNITY

Q & A

Thank you