Btrfs

Cesta k modernímu souborovému systému v Linuxu

Red Hat

Lukáš Czerner (lczerner@redhat.com)

May 29, 2017

Agenda

redhat.

Section 1

Souborové systémy

redhat.

Z čeho se skládá souborový systém

- **Superblock**
  - základní kámen souborového systému
  - popisuje strukturu a parametry
- **Inode** = "index node"
  - obsahuje metadata souboru
- **Mapování bloků v souboru**
  - Mapuje virtuální prostor souboru na fyzický prostor média
  - direct/indirect, stromy
- **Správa volného místa**
  - jedna ze základních funkcí souborového systému
  - historicky spojový seznam, bitmapy, rozsahy/extenty, stromy
- **Ochrana před výpadkem**
  - chrání zejména metadata, nikoliv uživatelská data
  - žurnálování, soft updates, copy-on-write

redhat.

Z čeho se skládá souborový systém

- **Superblock**
  - základní kámen souborového systému
  - popisuje strukturu a parametry
- **Inode** = "index node"
  - obsahuje metadata souboru
- **Mapování bloků v souboru**
  - Mapuje virtuální prostor souboru na fyzické bloky (pro sekvenční čtení/zápis stromy)
- **Správa volného místa**
  - jedna ze základních funkcí souborového systému
  - historicky spojový seznam, bitmapy, rozsahy (extenty) ve stromech
- **Ochrana před výpadkem**
  - chrání zejména metadata, nikoliv uživatelská data
  - žurnálování, soft updates, copy-on-write

redhat.

Z čeho se skládá souborový systém

- **Superblock**
  - základní kámen souborového systému
  - popisuje strukturu a parametry
- **Inode** = "index node"
  - obsahuje metadata souboru
- **Mapování bloků v souboru**
  - Mapuje virtuální prostor souboru na fyzický prostor média
  - direct/indirect, stromy
- **Správa volného místa**
  - jedna ze základních funkcí souborového systému
  - historicky spojový seznam, bitmapy, rozsahy/extenty, stromy
- **Ochrana před výpadkem**
  - chrání zejména metadata, nikoliv uživatelská data
  - žurnálování, soft updates, copy-on-write

redhat

Z čeho se skládá souborový systém

- **Superblock**
  - základní kámen souborového systému
  - popisuje strukturu a parametry
- **Inode** = "index node"
  - obsahuje metadata souboru
- **Mapování bloků v souboru**
  - Mapuje virtuální prostor souboru na fyzický prostor média
  - direct/indirect, stromy
- **Správa volného místa**
  - jedna ze základních funkcí souborového systému
  - historicky spojový seznam, bitmapy, rozsahy uložené ve stromech
- **Ochrana před výpadkem**
  - chrání zejména metadata, nikoliv uživatelská data
  - žurnálování, soft updates, copy-on-write

redhat.

Z čeho se skládá souborový systém

- **Superblock**
  - základní kámen souborového systému
  - popisuje strukturu a parametry
- **Inode** = "index node"
  - obsahuje metadata souboru
- **Mapování bloků v souboru**
  - Mapuje virtuální prostor souboru na fyzický prostor média
  - direct/indirect, stromy
- **Správa volného místa**
  - jedna ze základních funkcí souborového systému
  - historicky spojový seznam, bitmapy, rozsahy uložené ve stromech
- **Ochrana před výpadkem**
  - chrání zejména metadata, nikoliv uživatelská data
  - žurnálování, soft updates, copy-on-write

redhat.

Z čeho se skládá souborový systém

- **Superblock**
  - základní kámen souborového systému
  - popisuje strukturu a parametry
- **Inode** = "index node"
  - obsahuje metadata souboru
- **Mapování bloků v souboru**
  - Mapuje virtuální prostor souboru na fyzický prostor média
  - direct/indirect, stromy
- **Správa volného místa**
  - jedna ze základních funkcí souborového systému
  - historicky spojový seznam, bitmapy, rozsahy uložené ve stromech
- **Ochrana před výpadkem**
  - chrání zejména metadata, nikoliv uživatelská data
  - žurnálování, soft updates, copy-on-write

redhat.

Nedávná historie a současnost

- Správa diskových oddílů
- Vzniká ext4 (2006)
- B-strom s počítadlem referencí a podporou copy-on-write (2007)
- Zrození Btrfs
  - 2009 - experimentální verze začleněna do linuxového jádra

redhat.

Nedávná historie a současnost

- Kapacity moderních disků dosahují obrovských velikostí
  - Některé souborové systémy ještě stále používají 32-bit adresování
  - Vyšší pravděpodobnost poškození dat
  - Oprava chyb za běhu
- Požadavky na pokročilé funkce souborových systémů
  - snapshotting, šifrování, komprese dat
  - data/metadata checksumming, scrubbing
- Nové technologie SSD, SMR, NVM vyžadují zvláštní přístup
- Kladeny nové nároky na moderní souborový systém

redhat.

Section 2

B-stromy

redhat.

B-stromy

- Jsou již dlouho používány v různých souborových systémech
- Mezi jejich hlavní přednosti patří
  - Strom je vyvážený, všchny listy stromu jsou na stejné úrovni
  - Logaritmická časová složitost hledání, vkládání a odebírání položek
  - Minimalizuje počet diskových operací při průchodu stromem
- V souborových systémech se obvykle používá varianta B+stromu
  - Data jsou uložena pouze v listech stromu
  - V ostatních uzlech jsou pouze klíče

redhat.

B-trees, Shadowing, and Clones

- Ohad Rodeh (IBM Haifa Research Labs), 2007
- Vyvinul algoritmy pro b-stromy s podporou copy-on-write
- Nastínil souborový systém zakládající se na těchto stromech
- Chris Mason na tomto základě začal vytvářet souborový systém Btrfs

redhat.

Copy-on-write, stínová kopie, shadowing

#1

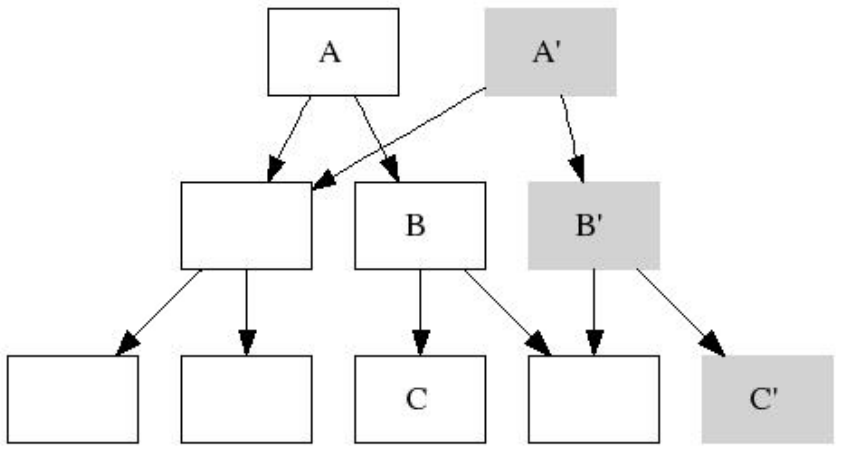
Při změně v uzlu C se vytvoří stínová kopie celé cesty až ke kořeni stromu, což dá vzniknout stromu s kořenem A'. Uzly A, B a C můžou být následně uvolněny.

#2

Aby souborový systém podporoval snapshoty (obraz souborového systému), musí umět pracovat s více než jedním kořenem stromu.

#3

Pokud bychom před modifikací C vytvořili snapshot, uzel A by zůstal jako kořen snapshotu.



redhat.

Copy-on-write, stínová kopie, shadowing

- Copy-on-write na B-stromu přináší nesporné praktické výhody
  - Ochrana před výpadkem bez použití žurnálu
  - Vytvoření snapshotu celého souborového systému
  - Klonováni souborů
- Nese sebou ale i některé problémy
  - Listy stromu nemohou být spojeny
  - Při změně v listu musí být uzamčeny předcházející uzel
  - Při vyvážení stromu, může být modifikace několika cest drahá

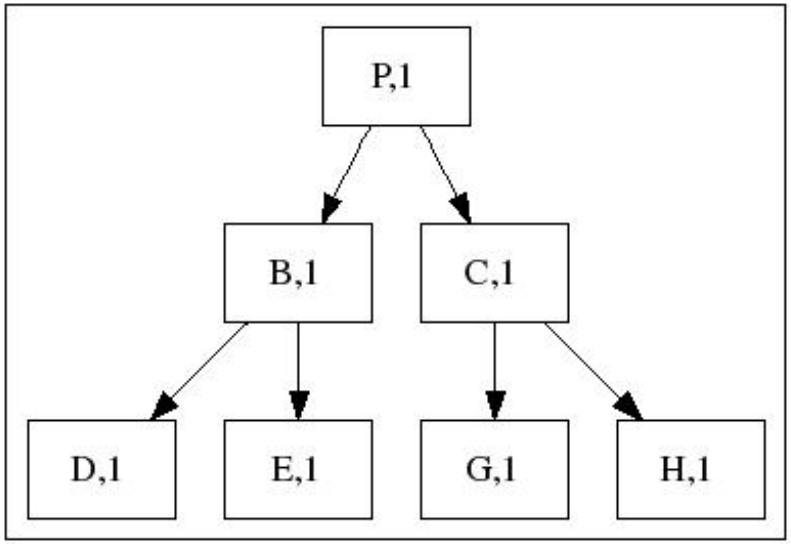
redhat.

Klonování

#1

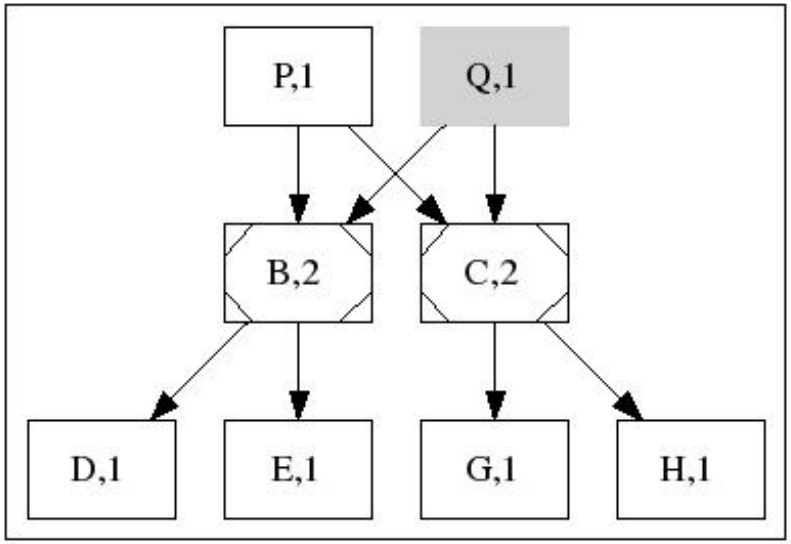
Klonováním stromu umožníme vytvoření snapshotu stromu.

- Rychlost
- Prostorová úspornost
- Možnost vytvořit velké množství klonů
- Možnost klony dále klonovat



#2

Mapa volného místa musí obsahovat čítač referencí pro každý blok. To umožní, místo vytvoření fyzické kopie stromu, pouze zvýšit referenci dotčených uzlů. To znamená, že uzly patří do dvou místo do jednoho stromu.



#3

Další optimalizací je zvyšování reference pouze u potomků kořene uzlu. Minimalizuje se tím množství změn v mapě volného místa.

Section 3

Btrfs design

redhat.

Obecná implementace b-stromu pro všechny objekty

- Postaven na základě práce Ohad Rodeh
- Nehraje roli jaké objekty jsou uloženy ve stromu
- Každý uzel stromu nese hlavičku a klíč
- Bez ohledu na operaci btrfs používá stejný kód
  - čtení
  - zápis
  - alokace dat/metadat

redhat.

Všechno je uloženo v b-stromech

- Stejný kód pro manipulaci se stromem
- Několik typů stromů
  1. *root tree* - kořeny ostatních stromů
  2. *chunk tree* - mapování logických bloků na fyzické
  3. *device allocation tree* - mapuje části fyzického média
  4. *extent allocation tree* - veškeré alokace
  5. *fs tree* - inody, soubory, adresáře
  6. *checksum tree*
  7. *data relocation tree*
  8. *log root tree*
- Každý strom má přidělené ID
- Pouze listy stromu obsahují samotná data

redhat.

Transakce v Btrfs

- Btrfs nespoléhá na klasický žurnál
- COW (copy-on-write) je zárukou konzistence
- Při změně *fs tree*, nebo *extent tree*
  1. Vytvoří se kopie stromu, změněná větev se zkopíruje a upraví
  2. Nové kořeny stromů jsou přidány do *root tree*
  3. Nový kořen *root tree* přidán do superbloku
  4. Čeká se až budou všechny změny dat a metadat zapsány na disk
  5. Nový superblok je zapsán
  6. Původní kořeny stromů můžou být uvolněny
- V případě výpadku je použit původní *root tree* (ten v nejaktuálnějším superbloku)

redhat.

Snapshoty v Btrfs

- Mohou být pouze pro čtení, nebo zapisovatelné
- Jsou vytvářeny stejným způsobem jako v případe transakcí
  - Původní strom však není automaticky uvolněn
- Může s nimi být zacházeno stejně jako se zbytkem souborového systému
  - Pouze sdílí části původního souborového systému
- Mohou být vytvořeny okamžitě v jakoukoliv chvíli

redhat.

Další vlastnosti Btrfs

- Správa disků
- Správa svazků
- Inkrementální zálohy pomocí send/receive
- Data/metadata checksumming
- Filesystem scrub
- Online defragmentace
- Komprese, šifrování a deduplikace
- Instantní kopírování (reflink)

redhat.

Section 4

Problémy a budoucnost Btrfs

redhat.

Problémy a budoucnost Btrfs

- Stabilizace souborového systému trvá velmi dlouhou dobu
  - Stále probíhá dokončování slibovaných vlastností
  - Na stabilizaci se stále pracuje
  - Většinou není považován za stabilní
- Btrfs udělal obrovský pokrok, ale stále se potýká s problémy
  - Problémy v případě nedostatku místa
  - Vysoká zátěž zámků v b-stromech ve specifických workloadech
  - Špatný výkon na souborech s náhodným zápisem
  - Historicky špatně fungující nástroj na kontrolu a opravu

redhat.

Problémy a budoucnost Btrfs

- Konkurence
  - ZFS portován na Linux - nekompatibilní licence
  - Podobné vlastnosti je možné dosáhnou v kombinaci s device mapper
  - Implementace některých pokročilých vlastností v tradičních souborových systémech
- Přesto vývoj stále pokračuje vysokým tempem
- Od myšlenky k reálnému použití vede dlouhá cesta

The end.

Thanks for listening.

redhat.

Odkazy

Dave Chinner:
Linux Filesystems: Where did they come from ?
2014.
`https://www.youtube.com/watch?v=DxZzSifuV4Q`

Ohad Rodeg:
B-trees, Shadowing and Clones
2007.
`https://liw.fi/larch/ohad-btrees-shadowing-clones.pdf`

`http://btrfs.wiki.kernel.org`

`http://lwn.net`