Current Challenges in UBIFS

Richard Weinberger
sigma star gmbh

/me

- Richard Weinberger
- Co-founder of sigma star gmbh
- Linux kernel developer and maintainer
- Strong focus on Linux kernel, lowlevel components, virtualization, security

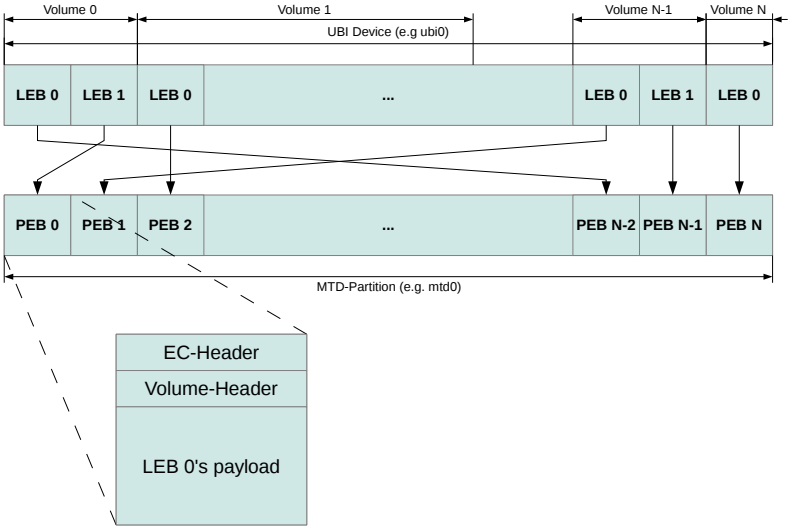
UBI overview

- Not a FTL (flash translation layer)
- A block layer for flash memory (NAND and NOR)
- Mapps physical blocks to logical blocks
- Offers volume management (static and dynamic volumes)
- Performs wear-leveling (across all volumes!)
- Allows atomic writes
- Offers a generic interface, most popular user: UBIFS
- Does not use OOB (out of band data)

UBI nomenclature

PEB	Physical erase block
LEB	Logical erase block
Image	UBI on your MTD partition
Device	Runtime representation of your UBI image (i.e. **/dev/ubi0**)
Volume	A volume within the UBI image (i.e. **/dev/ubi0_0**)
Attach	Process of loading an UBI image (i.e. attach mtd0)

UBI example


Volume 0
Volume 1
Volume N-1
Volume N
UBI Device (e.g ubi0)
LEB 0
LEB 1
LEB 0
...
LEB 0
LEB 1
LEB 0
PEB 0
PEB 1
PEB 2
...
PEB N-2
PEB N-1
PEB N
MTD-Partition (e.g. mtd0)
EC-Header
Volume-Header
LEB 0's payload

UBIFS overview

- Journaling filesystem on top of an UBI volume
- Successor of JFFS2
- Depends heavily on UBI semantics
- Offers transparent compression (lzo and zlib)
- Powercut tolerant (unlike most MMC, eMMC, SD, SSD, and other FTL devices)
- Very stable and mature

NAND these days

- Current NAND flashes are much bigger
- Cheap SLC NAND is often not that good as UBI and UBIFS expect
- MLC and TLC are commonly used
- ...UBI and UBIFS have to deal with that

Fastmap overview

- At attach time UBI learns for each volume the LEB to PEB mapping
- It has to inspect every PEB and reads EC and VID headers
- On a large flash this can take up to seconds
- Unacceptable for fast boot setups
- Fastmap stores LEB to PEB mappings into a special PEB
- No need to scan the whole flash

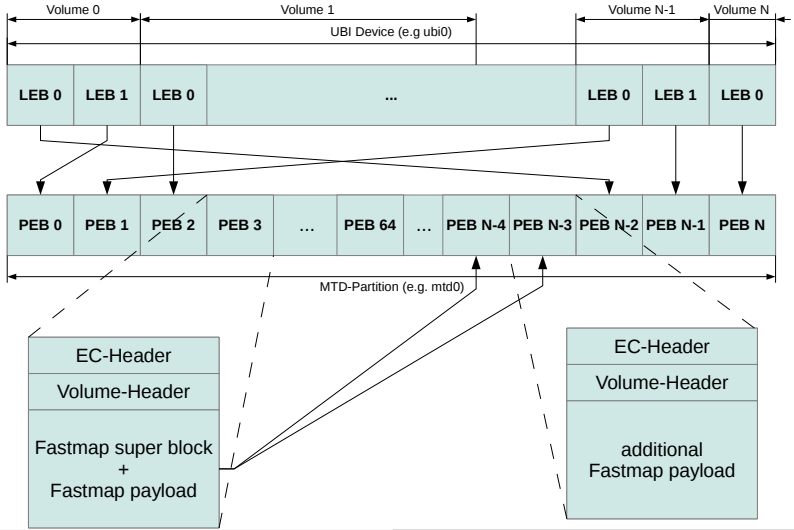
Fastmap timeline

- **2010:** PoC by Samsung
- **2011/2012:** PoC implemented by me under supervision of Thomas Gleixner
- **2012 - 2014:** New job, development stalled, ...
- **2014/2015:** Got founding to continue with development and as of Linux v4.1 all patches are mainline

Fastmap details

- At runtime fastmap manages a pool of PEBs
- Everytime all PEBs out of the pool are used a new checkpoint is written
- Initial name of fastmap was UBI checkpointing. I'm bad at naming things BTW.
- Fastmap data structure consists of two parts, the super block (also known as anchor) and the actual data.
- The super block has to be placed within the first 64 PEBs and will point to the PEB which contains all data
- If super block and data are small enough they will be squashed into the same PEB
- At attach time only pool PEBs have to be scanned

Fastmap example



How to use it?

- Enable **CONFIG_MTD_UBI_FASTMAP** in your Linux config
- Set **fm_autoconvert** UBI module parameter
- Watch out for **attached by fastmap** log message at bootup
- TODO: ubinize support

We want you!

- Fastmap is still marked as experimental
- Please give it a try and report issues
- I consider it as stable and would like to mark it so soon
- None of my customers managed to brick it, maybe you can :-)

UBI block

- Implements a read-only block device on top of an UBI volume
- Very handy for running squashfs on your flash
- Mainline since v3.15, by Ezequiel Garcia

Rename of busy volumes

- UBI supports rename of volumes
- But not for busy volumes
- Now you can, mainline since v4.0
- Very handy for atomic firmware upgrades
- e.g. you have two volumes, **main** and **test**. Bootloader tries **main** first
- Write new firmware to **test**, swap volume names using **ubirename** and reboot

Data retention overview

- Was not a big deal for SLC NAND at the time when UBI/FS was designed
- These days it is. Escpecially with MLC NAND
- Do deal with read disturb blocks have to be read and if bitflips happen it has to be re-written
- Often suggested (for each volume): **dd if=/dev/ubi0_X of=/dev/null**
- Running that in a cronjob is not sufficient to deal with read disturb
- It won't catch UBI meta data (pages with EC and VID headers and internal volumes)
- If you re-attach (or reboot) from time to time you'll catch these too
- . . . unless you're using fastmap
- We also have to re-write blocks which have not erased for a long time

ubi-healthd

- Userspace daemon
- Collects read/write/erase statistics from UBI
- User defined policy
- Statistics can be stored in a regular file or an UBI volume
- No absolute values needed, relative are enough
- Idea should materialize soon ;-)

Paired pages overview

- On MLC NAND pages are paired
- If you're writing page X and a powercut happens X's partner can get corrupted
- Possible solutions are currently evaluated
- It needs to be addressed in UBI and UBIFS, maybe in generic MTD core too
- Boris Brezillon is working on it

Paired pages possible solution

- UBI EC and VID headers need protection
- UBIFS journal too
- Skip paired pages and if enough data is written pack pages

Bitflips on erased pages

- UBIFS assumes that empty space is **0xff**
- Sadly some NFC still don't have ECC support for erased pages
- Upon a single bitflip UBIFS will complain

UBIFS general purpose features

- Part of a project to use UBIFS on general purpose servers
- Custom MTD at terabyte scale
- New UBIFS features: atime, quota
- Dongsheng Yang is working on it
- I don't have much details so far

UBI unpacker

- A tool to unpack UBI and UBIFS, like unsquashfs
- Scans the UBIFS index, replays journal
- Can operate on nanddumps

UBIFS fsck

- Based on the unpacker
- Scans all UBI and UBIFS metadata and is able to repair issues
- Fuzzy unpack mode

FIN

- ⋆ Questions?

Misuses: one UBI image per UBIFS

- Don't create an UBI for each UBIFS
- Make the UBI image as large as possible and use UBI volumes
- UBI wear leveling work across volumes!
- . . . incldudes also read-only volumes

Misuses: monster boot

- Don't boot from UBIFS
- Your boot loader does not have to know UBIFS nor have full UBI support
- Use static volumes, you can read out a static UBI volume with 100 LoC
- Thomas Gleixner sent patches for u-boot