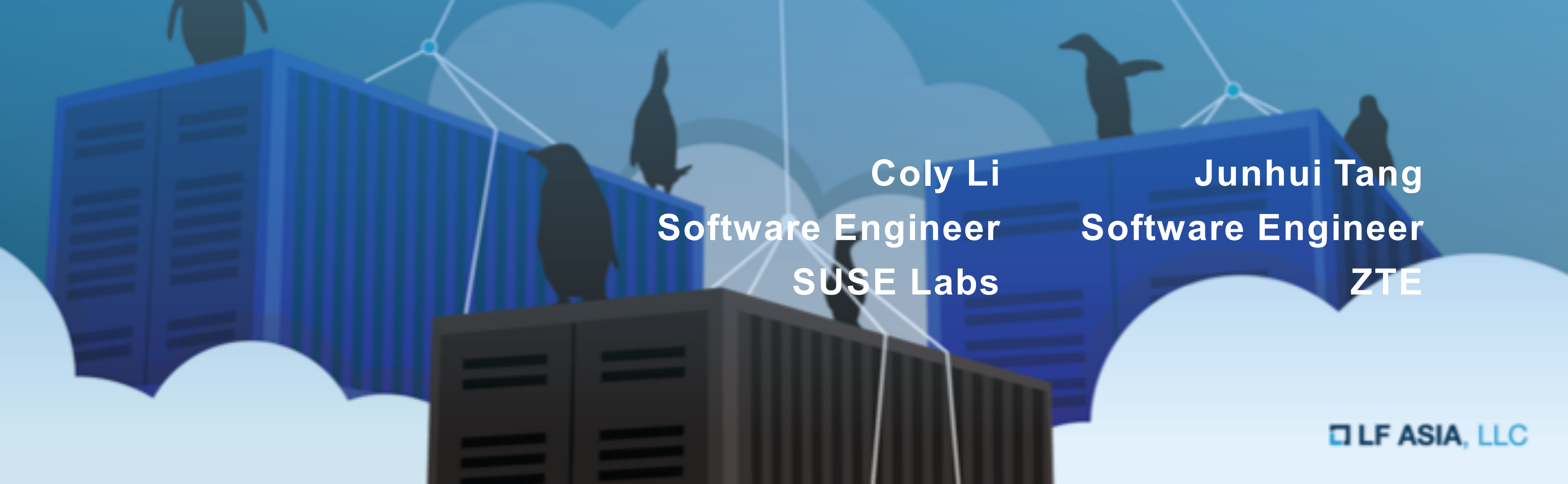
LINUXCON containercon CHINA 中国 CLOUDOPEN

THINK OPEN
开放性思维

Bcache Stability
Improvement for Cloud Infrastructure



Coly Li
Software Engineer
SUSE Labs

Junhui Tang
Software Engineer
ZTE

About Speakers

Coly Li

- Software Engineer, SUSE Labs
- Maintains md/dm/bcache for SUSE Linux
- Bcache maintainer upstream Linux kernel

Junhui Tang

- Software Engineer, ZTE
- Maintains bcache for ZTE storage products
- Active developer and important contributor to bcache of upstream Linux kernel

Bcache Overview

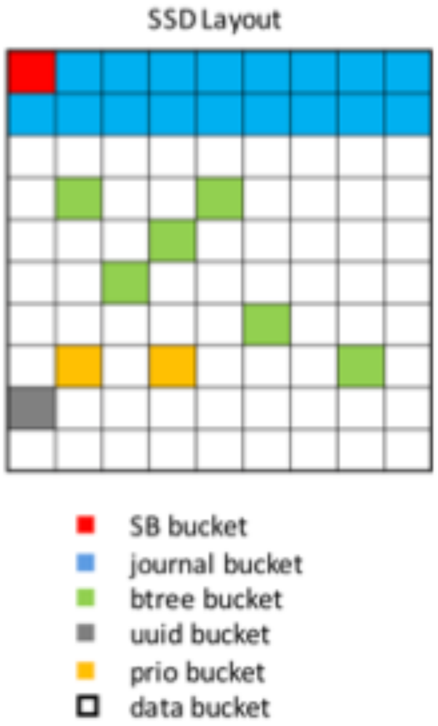
A high performance block layer cache, exists in Linux kernel for 5+ years.

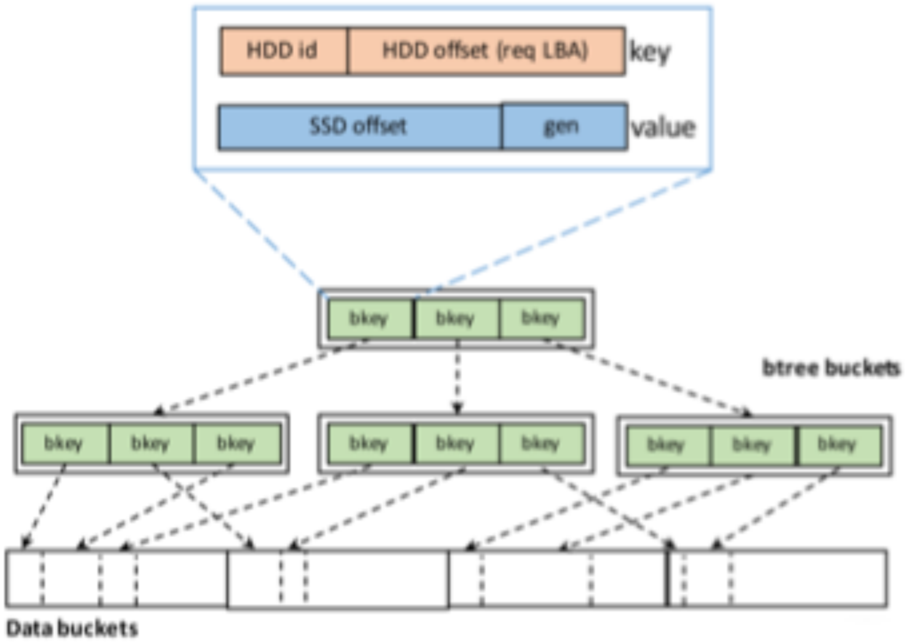
Originally developed by Kent Overstreet, and merged into Linux v3.10.

Widely deployed in industry (virtualization, data base, software define storage, etc) everywhere hot data can be accelerated.

Brief Design

- Storage space allocated in buckets
- Cache indexed by B+ tree, in-memory binary search tree used within each B+ tree node





Improvement required by Industry

- Device failure handling
- Writeback performance
- I/O latency in worst case
- Stability, don’t lock, don’t panic

Many development and improvement take place in the past 2 years.

Device Failure Handling

- Potential data corruption if device failure is not handled properly
  - Broken cache device
  - Broken backing device
    - Backing device gone without I/O
- Linux v4.18 has fundamental device failure handling for bcache
  - Fix many bugs for dependences of multiple reference counters
  - Retire cache set for broken cache
  - Stop bcache device for broken cache
  - Stop bcache device for broken backing device

Writeback Performance

- Dirty blocks on cache device are mostly randomly distributed on backing device.
- Write them back to spindle hard drive results unhappy performance (40~400KB/s)
- Even worse on SMR (Shingled Magnetic Recording) hard drives

- Michael Lyle makes situation better by re-ordering write requests before issuing to backing device.
- Throughput improved 3+ times for ideal conditions.

I/O latency in Worst Case

- There are multiple threads to access B+ tree in parallel
  - Writeback
  - Garbage collection
  - Data insert/invalidate/replacement
- Lock contention
- Dependency loop between B+ tree and journal

- Junhui Tang from ZTE contributes a lot of fixes to make things much better
  - Reduce worst latency from 120+ seconds to ~200ms
  - Well done!

Stability, don’t lock, don’t panic

- Deadlock on cache retire code path
- Panic of NULL pointer deference of gone struct block_device *bdev.
- Dependence circle within multiple locks among multiple threads

Almost all reported issues are fixed, bcache code in Linux v4.18 has improved stability with better overall performance.

Stable for Cloud Infrastructure

Positive feedback from industry partners and community,

- Virtualization
- Software Defined Storage
- Hyper-converged Infrastructure

Bcache is stable to I/O cache acceleration for your large scale could infrastructure.

Full enterprise production support from SUSE Linux Enterprise Sever 12 and 15.

Credit to Bcache Developers

Incomplete contributors list since Linux v3.10:

Kent Overstreet (164)	Kees Cook (5)	Liang Chen (2)
Coly Li (26)	Ming Lei (5)	Jan Kara (2)
Tang Junhui (24)	Ingo Molnar (4)	Al Viro (2)
Michael Lyle (13)	Mike Christie (4)	Mike Snitzer (2)
Slava Pestov (12)	Andy Shevchenko (3)	Wei Yongjun (1)
Nicholas Swenson (10)	Zheng Liu (3)	Zhai Zhaoxuan (1)
Bart Van Assche (9)	Dan Carpenter (3)	Jianjian Huo (1)
Jens Axboe (7)	Jiri Kosina (3)	Gu Zheng (1)
Christoph Hellwig (7)	NeilBrown (3)	Guoqing Jiang (1)
Eric Wheeler (7)	Surbhi Palande (2)	Greg Kroah-Hartman (1)
Yijing Wang (5)	Rui Hua (2)	Chengguang Xu (1)
Gabriel de Perthuis (5)	Michal Hocko (2)	

Development Roadmap

- Big endian support
- Reduce lock contention on B+ btree iteration
- User space tool enhancement
- SMR & 4K native hard drive support

We have active developers and users community, wildly used in cloud industry. Welcome to join us for a better block layer cache!

linux-bcache@vger.kernel.org

LINUXCON
containercon
CLOUDOPEN
CHINA 中国

THINK OPEN
开放性思维

Q & A

LF ASIA, LLC

LINUXCON

containercon

CLOUDOPEN

CHINA 中国

THINK OPEN

开放性思维

LF ASIA, LLC