BSDCan 2014

FreeBSD's Ext2 Implementation

Features and Status Report

Pedro Giffuni <pfg@FreeBSD.org>

Why Ext2fs is still important

- Performance (FIS 2010):
  - Ext2 is the fastest filesystem in linux.
  - ext4, jfs and xfs are similar (ext4 has a small edge)
  - ext3 is much slower
  - btrfs is slowest
- Compatible
  - Linux, Windows, *BSD, MacOSX (deprecated), Haiku, Hurd Masix (?), OS2.
  - Recommended for USBmem, SSD.
  - "Lightweight"

How this came out to be …

- 1992/04 – Initial implementation of the ext2fs in Linux
- 1995/11/09 - Initial port of GNU ext2fs (SVNr12115).
- 1998/06/12 – NetBSD has a BSD licensed ext2fs.

- 2009 GSoC - Improving Second Extended File system (ext2fs) and making it GPL free.
- 2010 GSoC - Enhance ext2fs to support preallocation and read ext4 file systems.
- 2012 GSoC - HTree directory indexing for Ext3

The Linux ext2fs (1992) – Rémy Card

- Created to overcome limitations of linux original minix-like fs: Max fs size extended to 4 TB max file size 2 G.
- Conceptually inspired on UFS but generally simpler. Defined by it's superblock and inode structures. No geometry considerations, smaller block sizes, no fragments.
- Ext3 (1999): journalling.
- Ext4 (2008): Extents.
- Future is btrfs.
- License: GPLv2

BSD-lites port (1995) – Godmar Back

- First approach: take the linux code and add glue code (FAILED): Buffer cache and VFS differences
- Second approach: Start from UFS with new directory format. Bring allocation policies from linux. Minimal glue code (ext2_blkpref)
- Some linux specifics not ported (resuid/resgid)
- Some UFS specifics not ported: cluster_write and reallocblks nor miplemented.
- License: GPLv2 + BSD.

Initial FreeBSD 2.2 port (1995) –John Dyson

- Ripped from BSD-lite + update the UFS specifics.
- Notably slower than the linux version in async mode.
- Only maintainance changes, no development. No attempt to follow upstream.
- Code was ported to NetBSD and later MacOS X (sourceforge).
- Userland code (mkfs and fsck) removed. Async removed
- Due to license, code is isolated from UFS. Not linked by default.
- License BSD + GPL = ?.

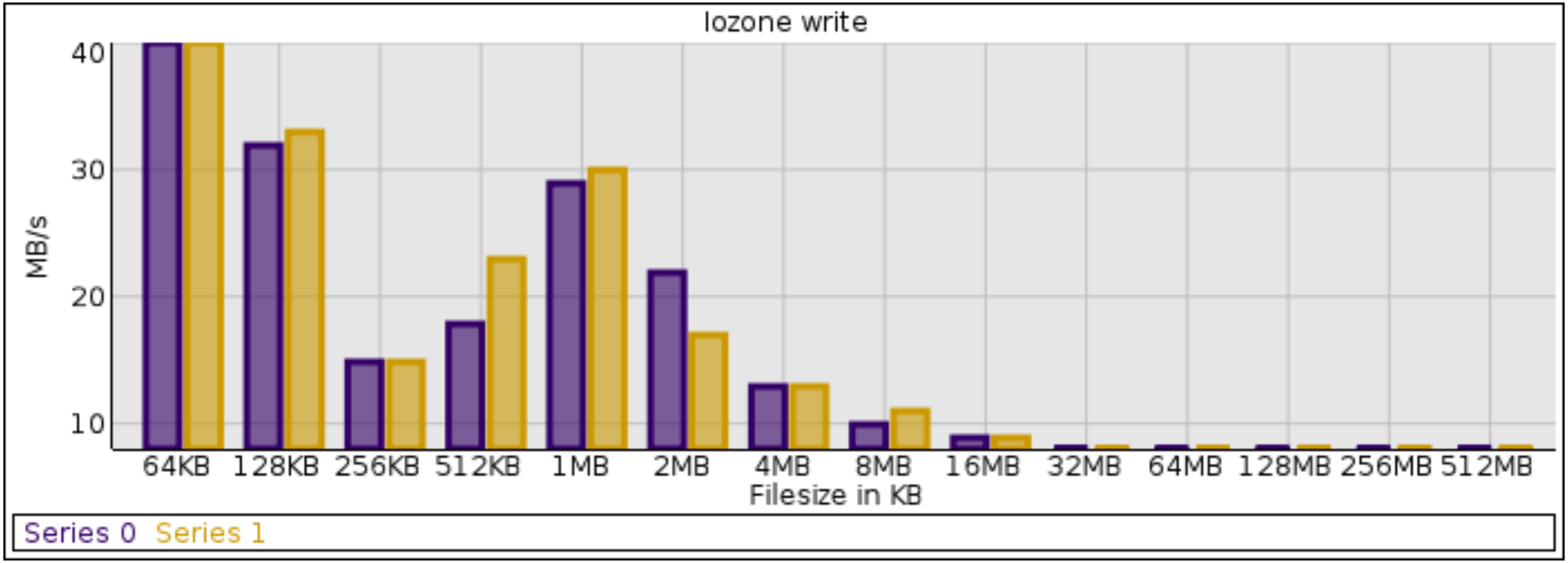
NetBSD's reimplementation (1998) – Manuel Buoyer

- cp -R sys/ufs/ffs sys/ufs/ext2
- Renamed data structures dropped fragments, other hacks.
- Re-implemented allocation policies (similar to ffs)
- Copied the directory lookup code from FreeBSD's port.(yes, I noticed !)
- Very clean implementation but slower than the FreeBSD port.
- License: BSD

GSoC 2009: Improving extfs and making it GPL free – Aditya Sarawgi

- Mentor: Ulf Lilleengen
- Start from NetBSD or FreeBSD? Headers or code?
- Hint: coders are lazy.
- Result: Performance halved. Pre-allocation lost. Coding style oops.
- The code was made MPsafe in an attempt to compensate for lost performance.
- UFS1 pre-softupdates became important as a reference. “Orlov” allocator.

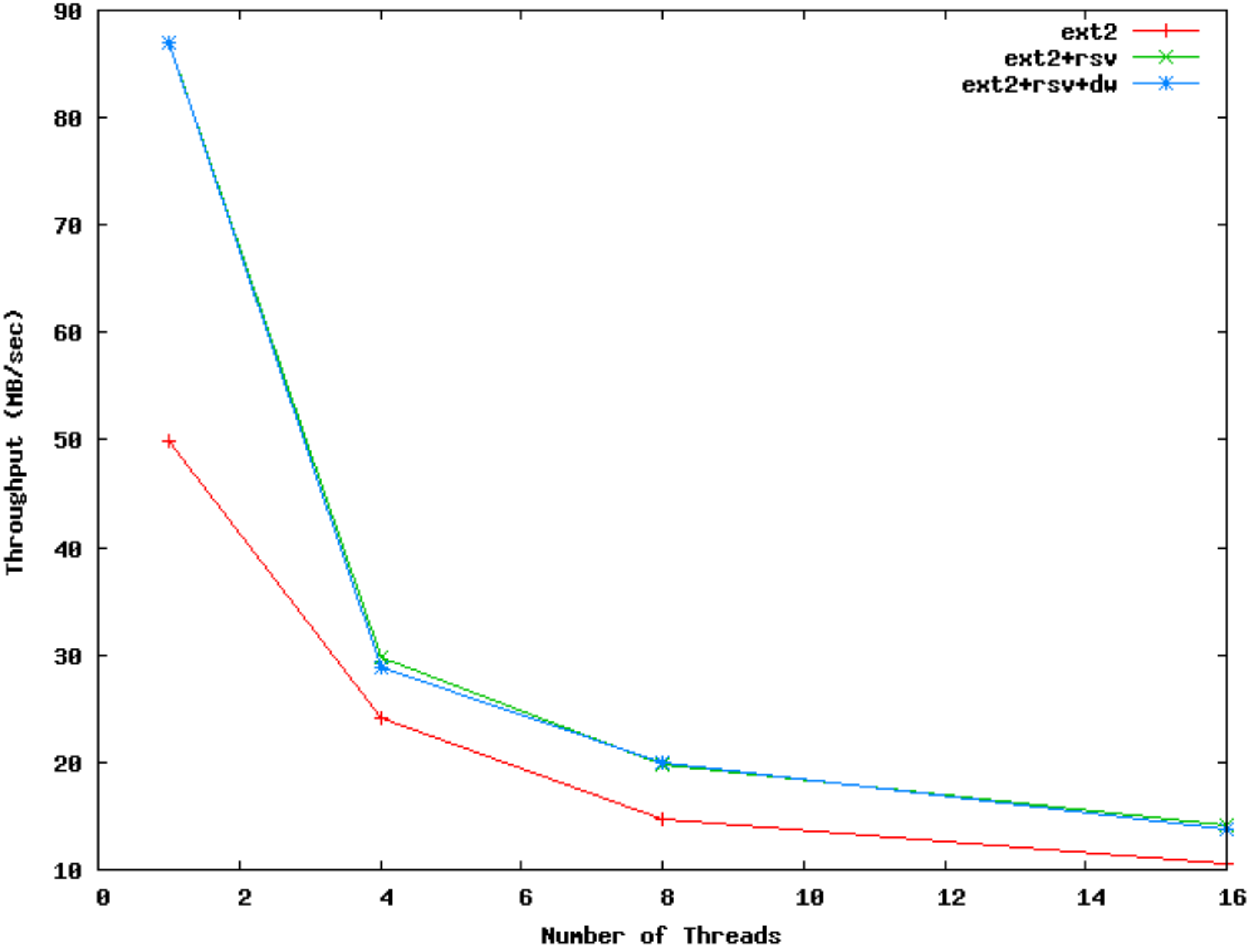
Results after GSoC 2009



GSoC 2010: Support pre-allocation and ext4 support – Zheng Liu

- Mentor: John Baldwin
- Many issues left from previous GSoC but GPL clean. Lots of work in parallel:
  - Interesting research papers related to ext2fs: development contrasts with UFS
  - Fixed async mode, added O_DIRECT.
  - bde@ had some research found bug in NetBSD’s code. Fixed by jhb@. pfg@ becomes committer.
- Project was successful but it took a lot of time to get things into shape in the tree.

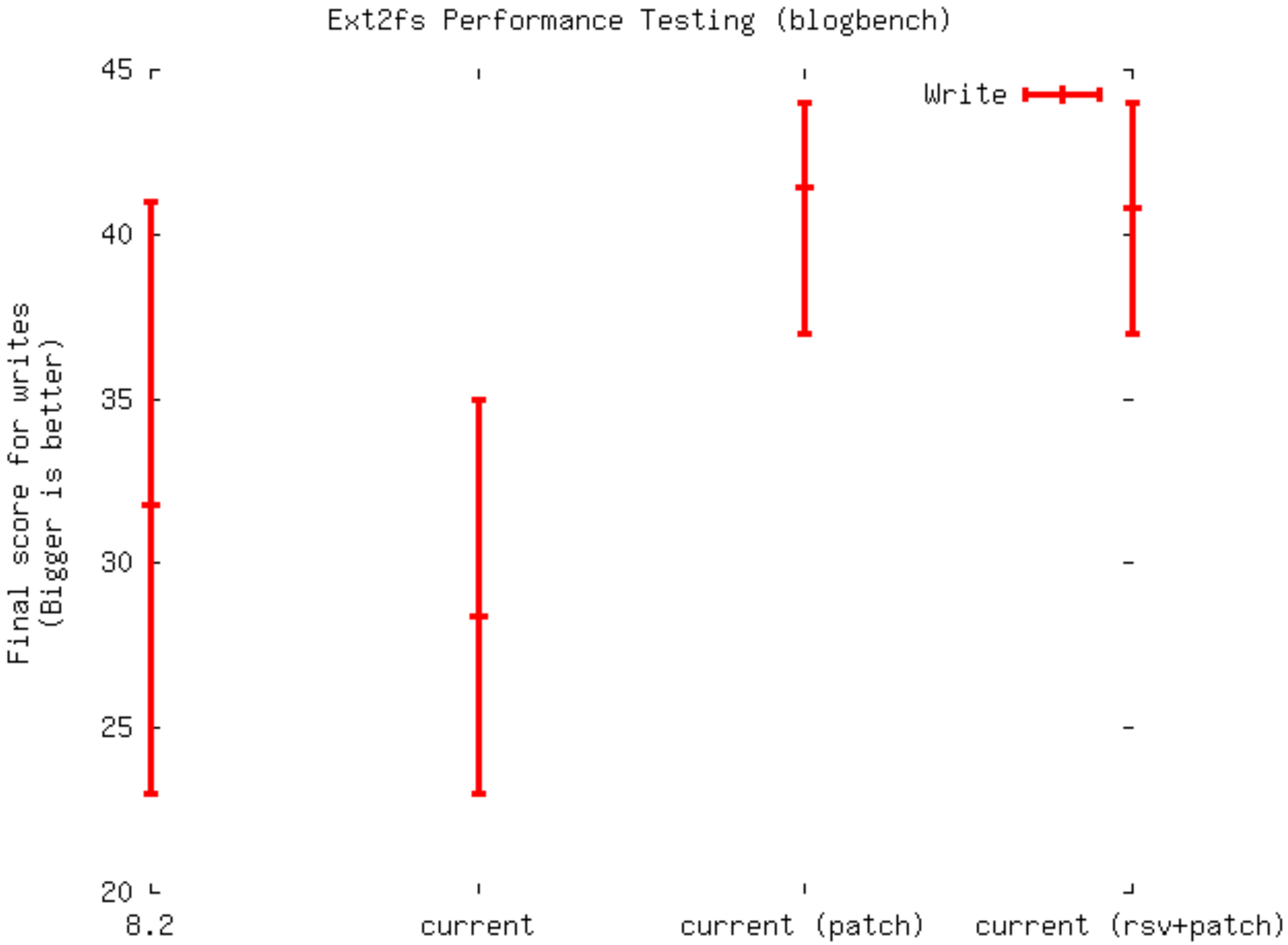
Results GSoC 2010 (Throughput/thread - async



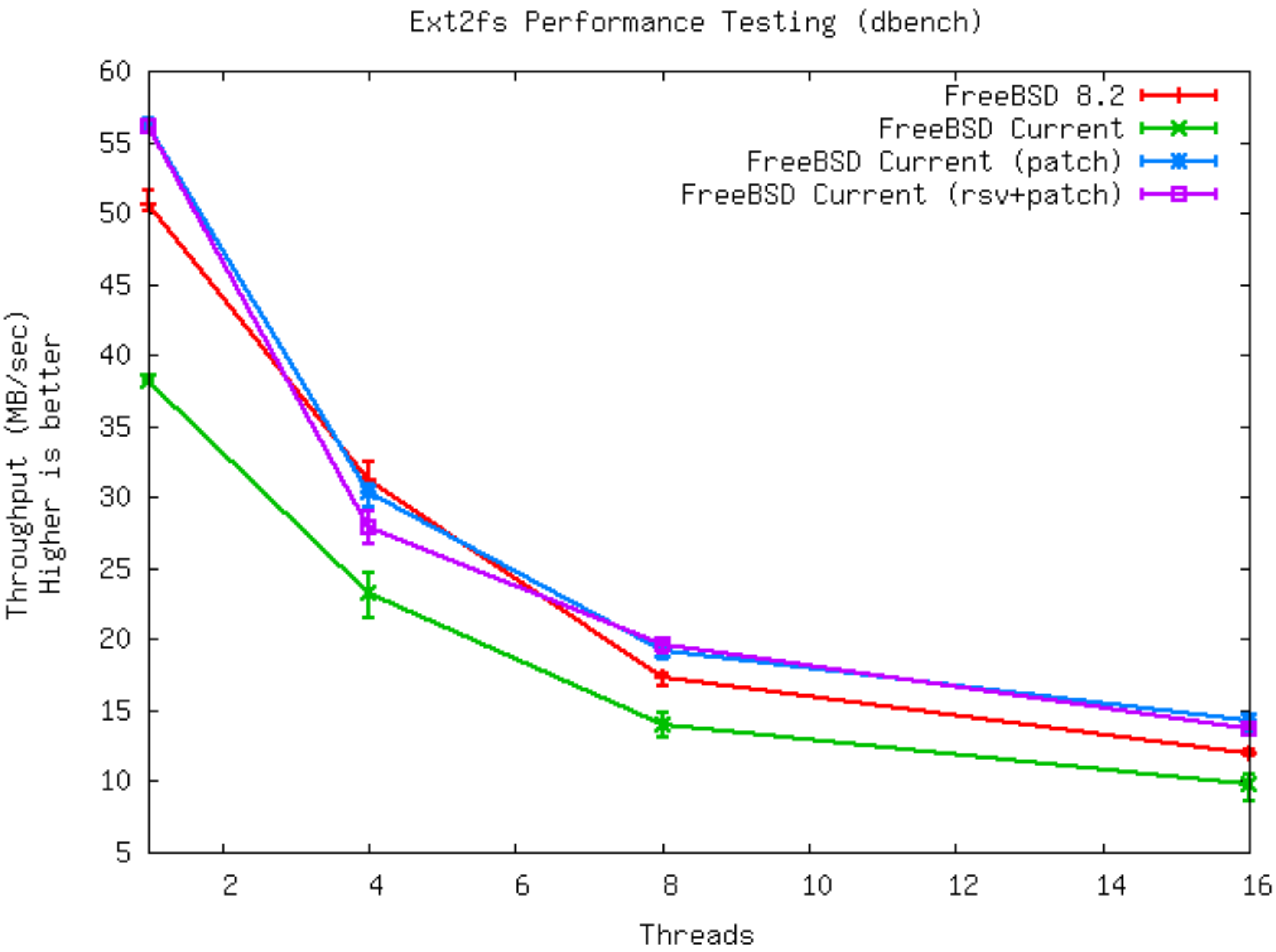
Linux Reservation Windows – Mingming Cao 2005

- Meant as a replacement for preallocation in ext3 but reserves in memory instead of disk.
- Each inode has it's own reservation window, windows cannot overlapped, indexed by a per-filesystem redblack tree.
- “Reducing fsck time for ext2 file systems”, Valerie Aurora et al. (2006): “*The results for the reservations-only versions of ext2 are even more puzzling; we suspect that our port of reservations is buggy or suboptimal*”.

Results GSoC 2010 – Blogbench write – Jan 2010



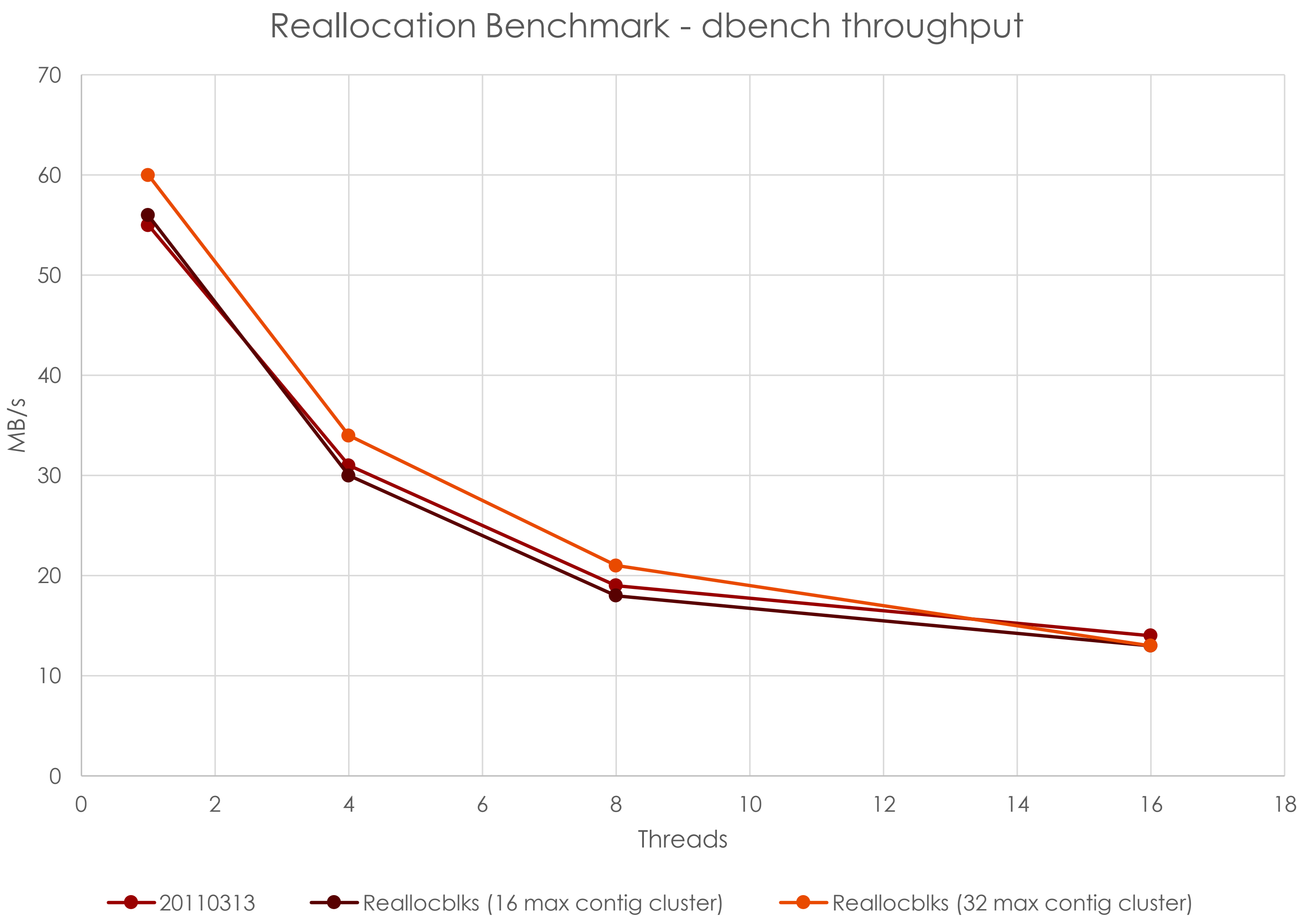
Dbench – Jan 2011



Reallocblk — McKusick 1994

- "old time classic" for BSD users: fragmentation.
- "A Comparison of FFS Disk Allocation Policies", Keith A. Smith and Margo Seltzer: "*The improved file layout achieved by the realloc algorithm improved read and write performance for large files by up to 16%. Read performance for files up to 96 kilobytes improved by as much as 20%.*"
- Makes allocation somewhat complex (NetBSD disabled it) but ext2 is simpler than UFS : no fragments. Similar to Ext4 "delayed allocation".

Reallocblk – Zheng Liu 2011



Status after GSoC2011

- Ext4 read-only was done but no feedback. Eventually rusted.
- Reallocblk started showing issues in other parts of the filesystem. FSX and benchmarks were showing new bugs. Very slow adoption but tried to keep the code in FBSD 10 and 9 in sync.
- Proposed GSoC2012: dirindex and journalling. (Not acepted).
- Not a wide used fs in the BSDs. Lot's of catching-up with UFS: direct_io, seek data/hole. Huge files.

NetBSD GSoC 2012 - Vyacheslav Matyushin

- Based on paper “A Directory Index for Ext2”, Daniel Philips (2002). Not used for Tux3.
- Included in Linux ext3 but only default in ext4.
- A lot of trouble with NFS dircookies: issues due to hash order. Not recommended for UFS.
- Haiku and NetBSD haven’t adopted it. lz@ ported it to help with ext4.
- Still uncertain benchmarking but feature is important in linux.

Late Implementation Strategy and Ext4

- No control over "upstream" design: no interest in "private" fields.
- Strategy was to get ext2 in good shape, slowly adding extensions: adapt headers to maintain ext4 data fields
- Share code with UFS, KISS, try to behave as Linux.
- Ext4 performance not a priority do get all metadata we can: timestamps, huge files.
- Wild development upstream: Feature flag mess.
- Kudos to Zheng Liu.
- Ext4 read-write?

Some thoughts …

- Features we don’t have: EA, ACLs, many directory limits. Used in Lustre.
- Endianness.
- Benchmarks vs features and time testing.
- ZFS and zvols.
- Other Linux filesystems (no plans yet).
- Recomendations

Huge thanks to Ext2 team!

- Questions?