Ext4: The Next Generation of Ext2/3

Theodore Ts'o
IBM

What's good about ext3

- Very large user community
- Very large developer community
  - From a large number of companies:
    - Red Hat, IBM, Bull, Clusterfs, Google, NEC, others
- Emphasis on robustness above all else
  - Simple filesystem format
  - “PC Class hardware sucks”

What's not so good about ext3

- 16TB filesystem size limitation (32-bit block numbers)
- Second resolution timestamps
- 32,768 limit on subdirectories
- Performance limitations

Why fork Ext4?

- No development 2.7 tree
  - ... and changes take longer than the 2-3 months between 2.6 releases
- Large userspace community
  - Kernel developers like Linus Torvalds and Andrew Morton get really cranky if their source trees get trashed
- Many changes on-deck require format changes
- Allows more experimentation than if the work is done outside of mainline
  - Make sure users understand that ext4 is risky: mount -t ext4dev

Features

- Abillity to use > 16TB filesystems (going beyond 32-bit block numbers)
- **Support files larger than 2TB**
- **Replacing indirect blocks with extents**
- **More efficient block allocation**
- **Allow greater than 32k subdirectories**
- **Nanosecond timestamps**
- **Metadata checksumming**
- **Uninitialized groups to speed up** mkfs/**fsck**
- **Persistent file allocation**
- **Inode table readahead**
- Online defragmentation

Extents

- Traditional indirect block maps are incredibly inefficient
  - One extra block read (and seek) every 1024 blocks
  - Really obvious when deleting big CD/DVD image files
- Extents are an efficient way to represent large file
- An extent is a single descriptor for a range of contiguous blocks

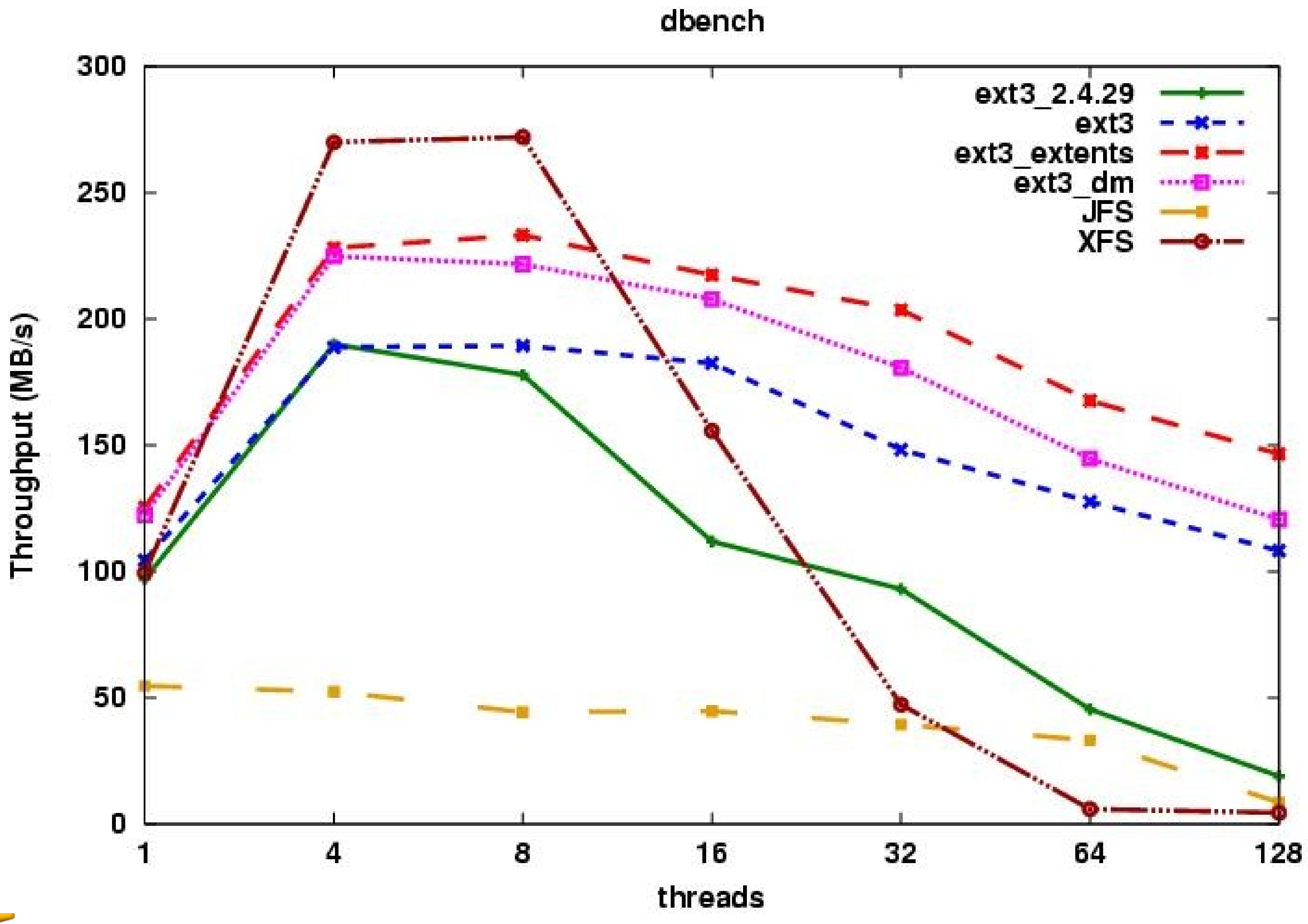
logical	length	physical
0	1000	200

Extent Related Technologies

- Multiple block allocation
  - Allocate contiguous blocks together
    - Reduce fragmentation, reduce extent meta-data
    - Stripe aligned allocations
- Delayed allocation
  - Defer block allocation to writeback time
  - Improve chances allocating contiguous blocks, reducing fragmentation
- Preallocation of file blocks without having to initialize them
  - Contiguous allocation to reduce fragmentation
    - Irrespective of order that blocks are written
    - While avoiding overhead of zeroing blocks
  - Guaranteed space allocation
  - Useful for Streaming audio/video and databases

Benefits to end-users

- Scalability
  - Support files > 2TB
  - Support Exabyte-sized filesystems
- Performance
  - For many different workloads
    - Streaming read/writes to large files
    - Random I/O to large files
    - Access to many related small files
- Better robustness
- Faster fsck times – by a factor of 6-8


dbench
300
250
200
150
100
50
0
Throughput (MB/s)
1
4
8
16
32
64
128
threads
ext3_2.4.29
ext3
ext3_extents
ext3_dm
JFS
XFS

Current status

- Ext4 is in the mainline kernel
    - Ext4 patch queue for fixes and enhancements
- Leaving development phase
    - 2.6.28 we will be renaming “ext4dev to ext4”
    - I have been using it on my laptop since July...
- Next steps
    - More performance tuning and testing
    - E2fsprogs 64-bit block number support
    - Will be showing up in distributions soon
        - First community distributions, such as Fedora 10
        - Since ext4 has a conservative design, and reuses large parts of ext3, it is easier for enterprise distributions to be confident supporting ext4

Getting involved

- Mailing list: linux-ext4@vger.kernel.org
- Wiki: http://ext4.wiki.kernel.org
  - To get started, please see: http://ext4.wiki.kernel.org/index.php/Ext4_Howto
- Weekly conference call

The Ext4 Development Team

- Alex Thomas (Sun/Clusterfs)
- Andreas Dilger (Sun/Clusterfs)
- Theodore Tso (IBM/Linux Foundation)
- Mingming Cao (IBM)
- Aneesh Kumar (IBM)
- Frédéric Bohé (Bull)
- Eric Sandeen (Red Hat)
- Val Aurora Henson (Red Hat)

- Andrew Morton
- Laurent Vivier
- Alexandre Ratchov
- Eric Sandeen
- Takashi Sato