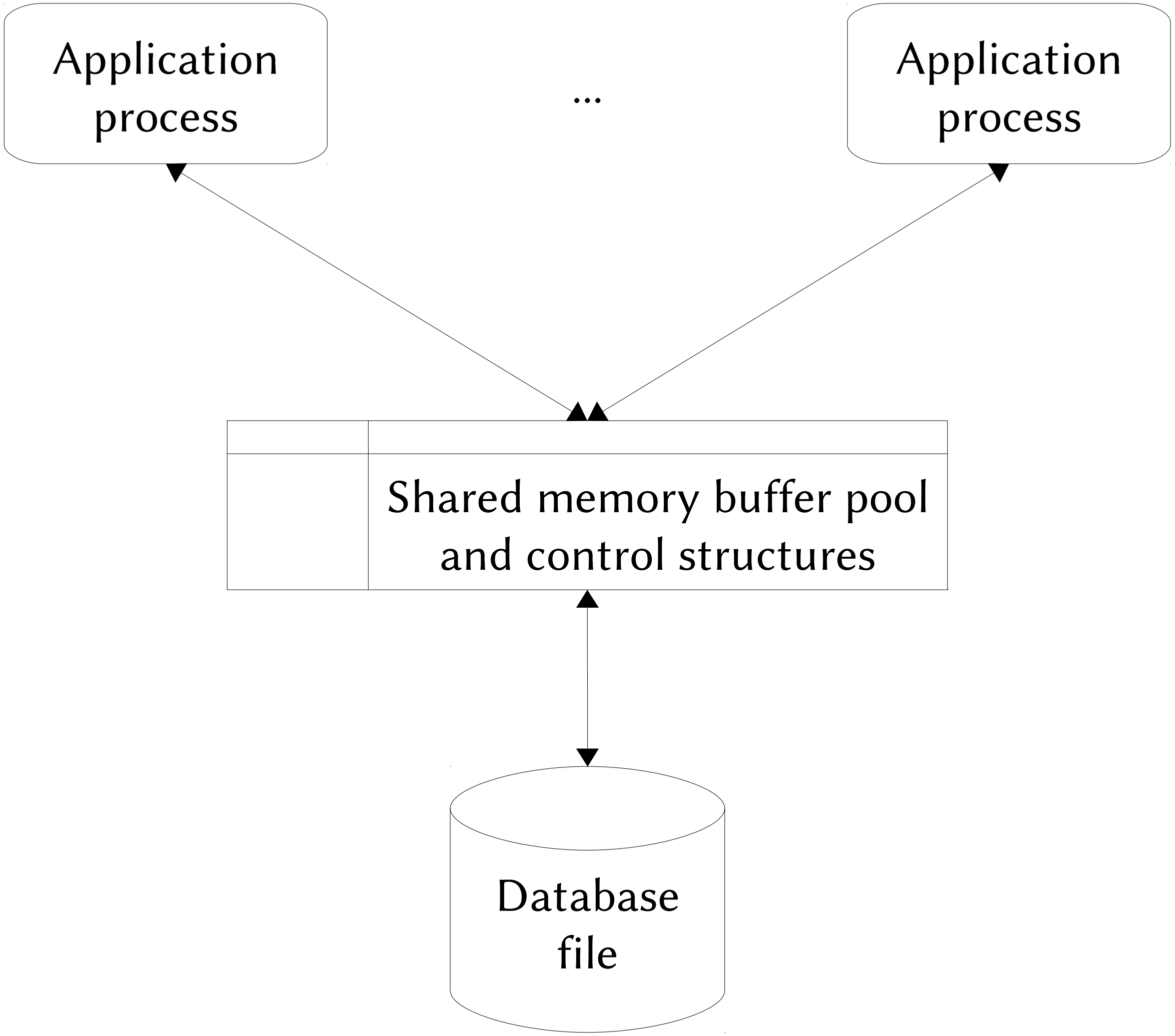
Benchmarking Linux Filesystems for Database Performance – Revisited

K.S. Bhaskar
Development Director, FIS
ks.bhaskar@fisglobal.com
+1 (610) 578-4265

Background & Motivation

- Database workloads interest us, as the developers of FIS GT.M™
  - Especially transactional workloads
    - Platform for the three largest real-time core-banking systems in the world that we know of – databases of a few TB, 10,000 concurrent users (plus web users, ATM networks, voice response units, etc.)
    - Increasingly used for electronic health records
  - Especially NoSQL ("Not Only SQL") data
  - Uses POSIX APIs
- Benchmarking complete applications is hard
  - Not widely available
    - Licenses often require permission to publish benchmark results
  - Typically complex, requiring expertise to configure & operate
    - Benchmarking requires repeatability
- Ideally "drop dead simple"
- (Update work presented at Linux Enterprise End User Summit 2010)

GT.M Daemonless Database Engine



Concurrent Multi-process Workloads

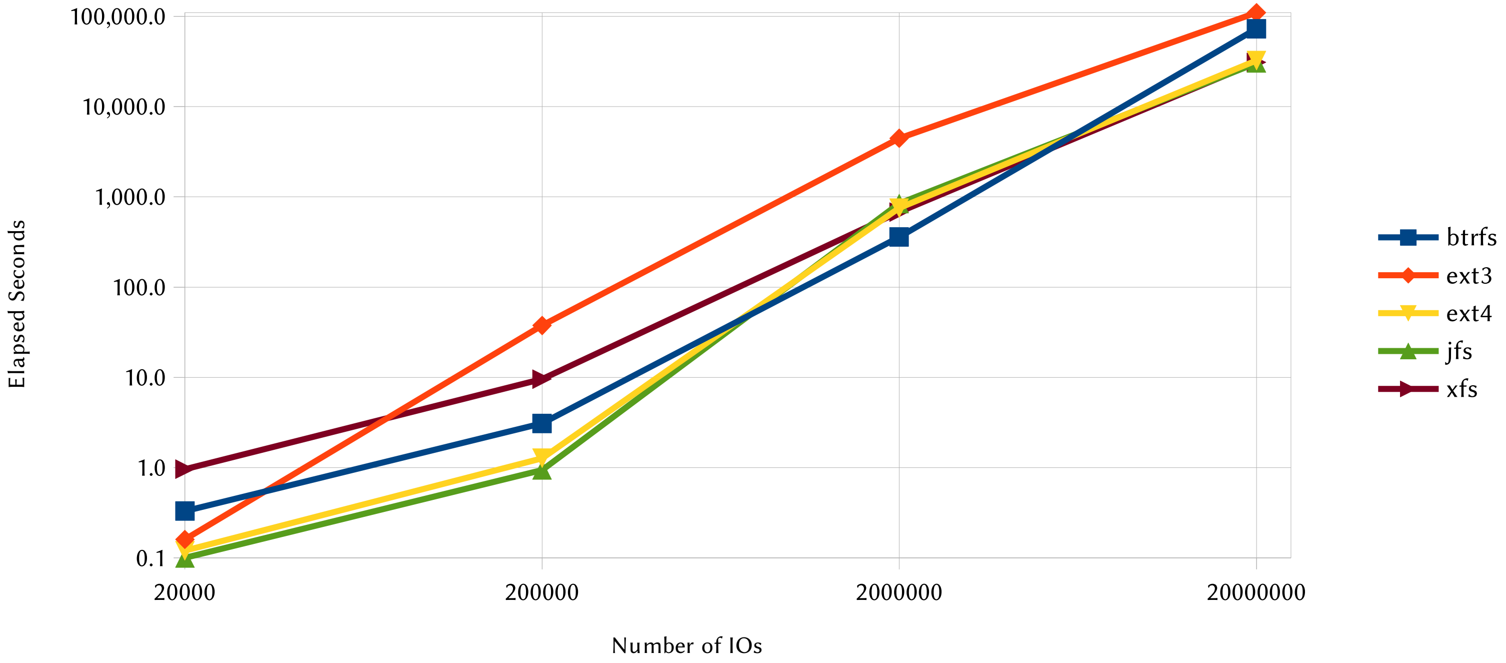
- io_thrash
  - “Download, compile, run”
  - ANSI C – originally developed in 2004; updated for current gcc releases
  - Publicly released in 2008
- threeen1f – 3n+1 sequence lengths
  - “Download, install, run”
    - (But we did change default parameters a little)
  - Developed as benchmark and sample program for Wikipedia page
  - Publicly released in 2010

Nuts and Bolts

- CPU – AMD Phenom II X4 965 Processor @ 3.4GHz
- RAM – 8GiB DDR3 @1.6GHz in 2 banks of 4GiB
- Disk – 2x Seagate Barracuda ST1000DM003; benchmark filesystems in logical volumes striped across both drives
- OS – 64-bit Ubuntu 12.10
- Filesystems – default mount options except nodatacow for btrfs
- Results – usually the median of at least three runs, except
  - btrfs & ext3 io_thrash (two runs for 20,000,000 IOs)
  - Jfs io_thrash (one run for 20,000,000 IOs)

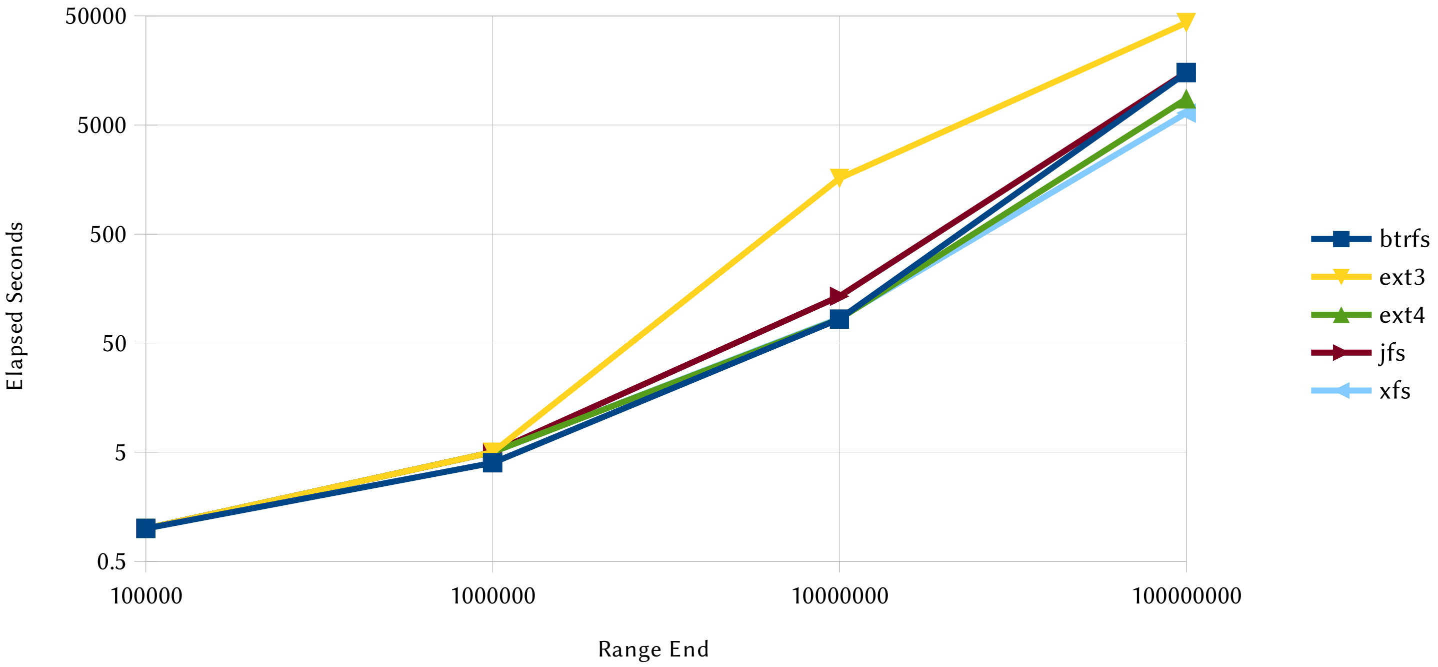
Elapsed Seconds – io_thrash

nIo	btrfs	ext3	ext4	jfs	xfs
20,000	0.33	0.16	0.12	0.10	0.96
200,000	3.09	37.51	1.27	0.94	9.55
2,000,000	359.15	4,455.76	751.93	844.11	678.58
20,000,000	72,919.42	109,799.26	32,417.50	30,317.61	30,915.41



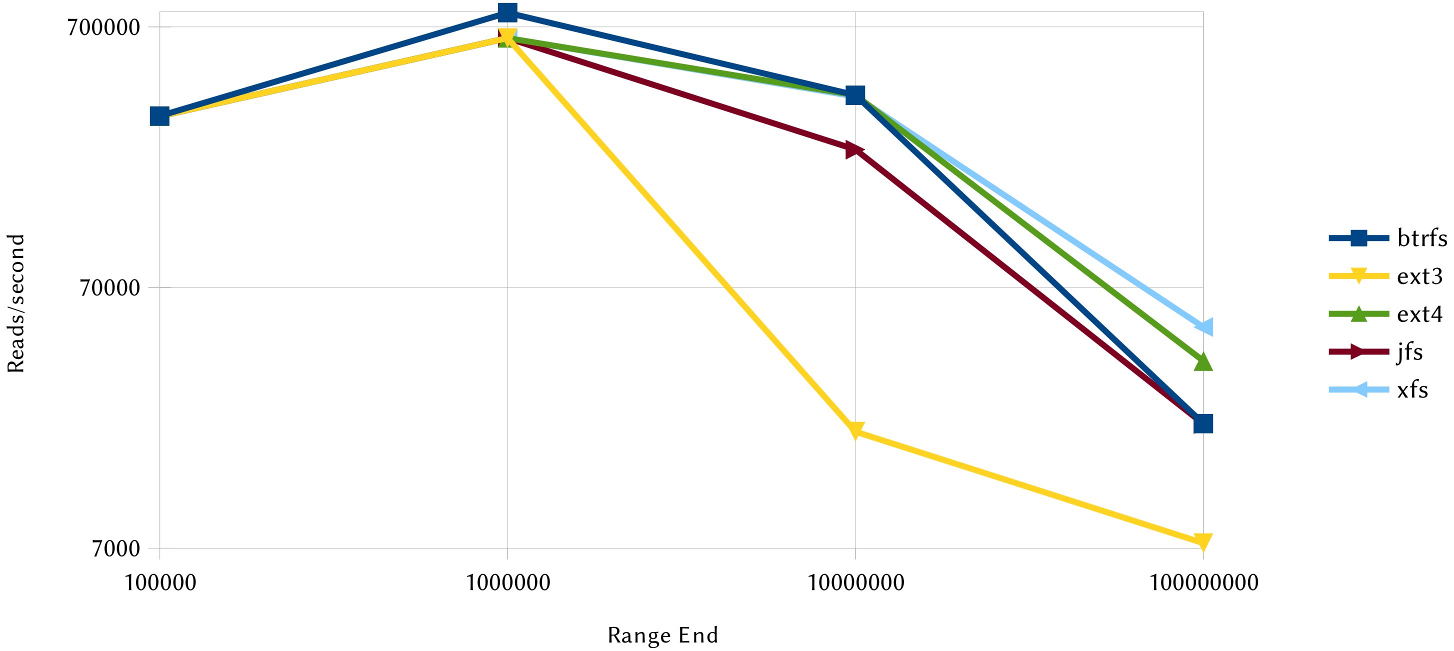
Elapsed Seconds – 3n+1

Range end	btrfs	ext3	ext4	jfs	xfs
100,000	1	1	1	1	1
1,000,000	4	5	5	5	5
10,000,000	83	1,620	83	135	85
100,000,000	15,114	43,354	8,695	15,150	6,439



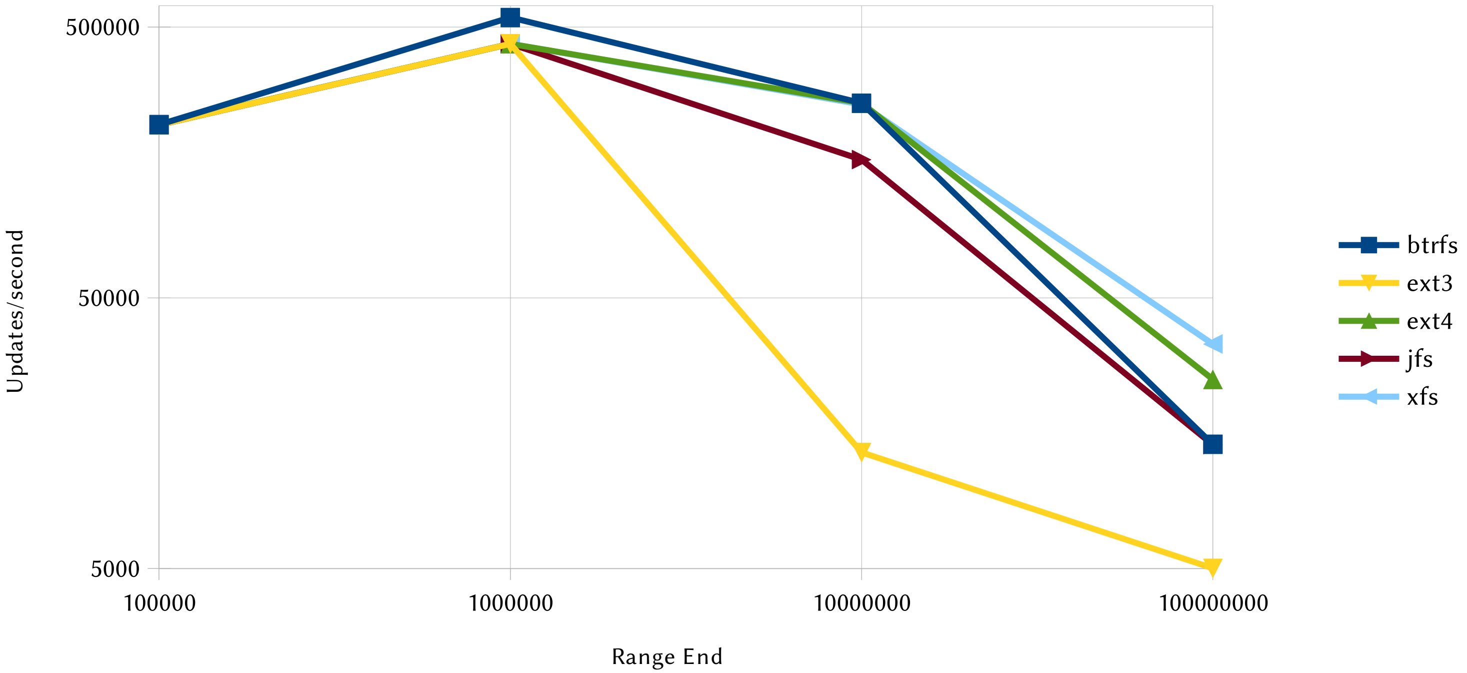
Reads/Second – 3n+1

Range end	btrfs	ext3	ext4	jfs	xfs
100,000	318,012	317,718	317,838	317,712	317,777
1,000,000	792,516	633,988	634,027	633,972	634,016
10,000,000	382,329	19,589	382,323	236,493	377,783
100,000,000	20,993	7,319	36,492	20,949	49,287



Updates/Second – 3n+1

Range end	btrfs	ext3	ext4	jfs	xfs
100,000	218,012	217,718	217,838	217,712	217,777
1,000,000	542,516	433,988	434,027	433,972	434,016
10,000,000	261,847	13,416	261,841	161,969	258,736
100,000,000	14,377	5,012	24,991	14,347	33,754



Results

- xfs is best
- ext4 is a good choice
- jfs met expectations
- btrfs was a pleasant surprise
- Avoid ext3

Links

- FIS GT.M home page: http://fis-gtm.com
- This presentation: http://tinco.pair.com/bhaskar/gtm/doc/misc/130510-1LinuxFileSystemBenchmarks.pdf
- How To: http://tinco.pair.com/bhaskar/gtm/doc/misc/130512-1LFSBenchmarkHowTo.pdf
- Raw Data: http://tinco.pair.com/bhaskar/gtm/doc/misc/130423-1FilesystemBenchmarkData.ods
- lshw of platform: http://tinco.pair.com/bhaskar/gtm/doc/misc/130512-2LFSBenchmarklshw.txt
- K.S. Bhaskar / ks.bhaskar@fisglobal.com / +1 (610) 578-4265

Questions / Discussion