Intrinsyc Software

Android on eMMC
Optimizing for Performance

Tom Foy
tfoy@intrinsyc.com

What is eMMC?

* Solid state storage device on MMC bus
* Chip on PCB
* NAND flash based

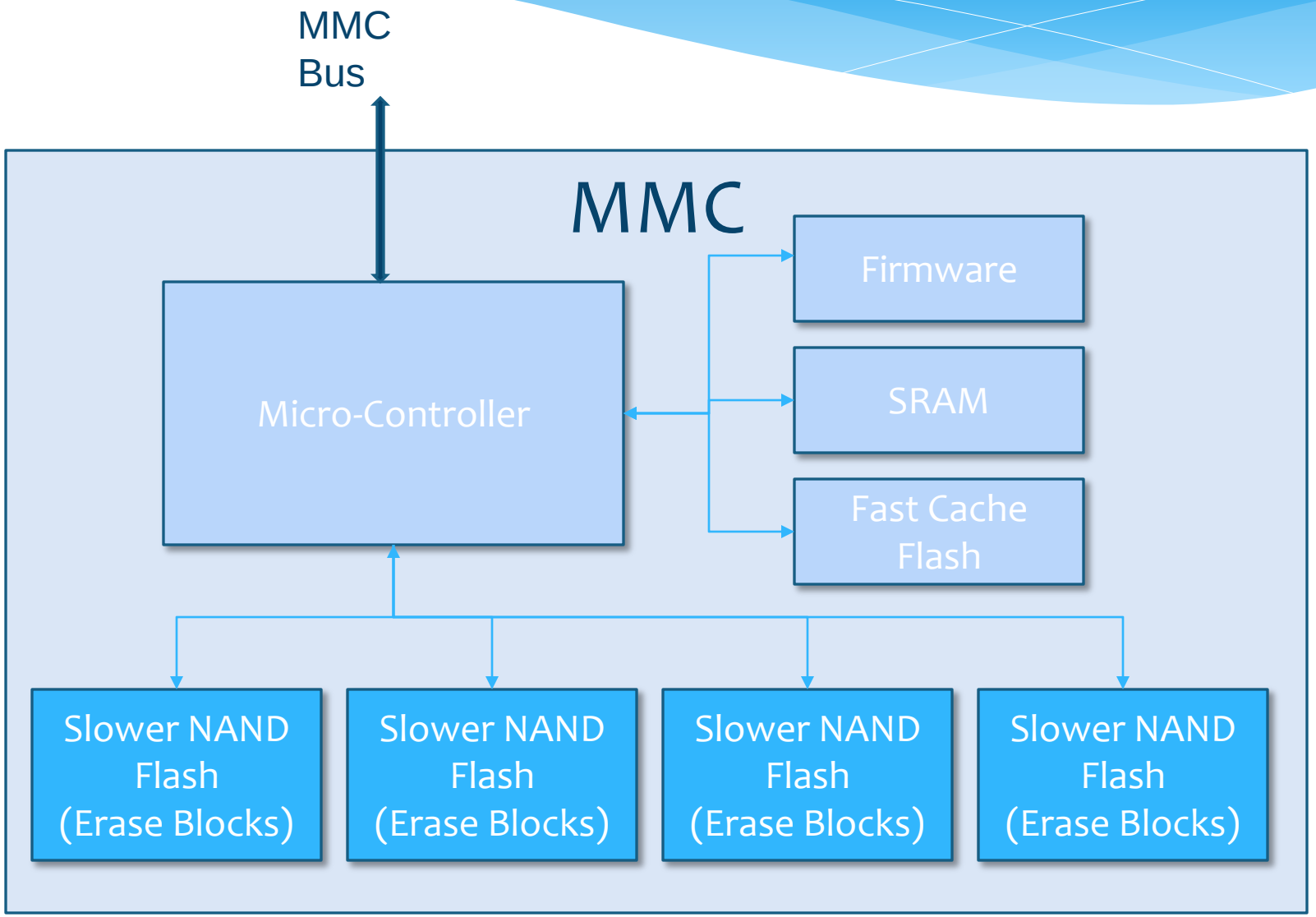
Why eMMC matters

* Popular on embedded devices
* Cheap
* Flexible

eMMC characteristics

* Fast read access
* Fast read seek times
* Acceptable sequential write performance
* Poor random write performance

Inside


MMC
Bus
MMC
Micro-Controller
Firmware
SRAM
Fast Cache
Flash
Slower NAND
Flash
(Erase Blocks)
Slower NAND
Flash
(Erase Blocks)
Slower NAND
Flash
(Erase Blocks)
Slower NAND
Flash
(Erase Blocks)

Inside the eMMC

* NAND flash arranged in pages
* Controller with temporary storage
* Wear levelling
* Free space management

Discard

* eMMC TRIM command
* Tells controller what is free
* TRIM blocks on format

eMMC scenarios

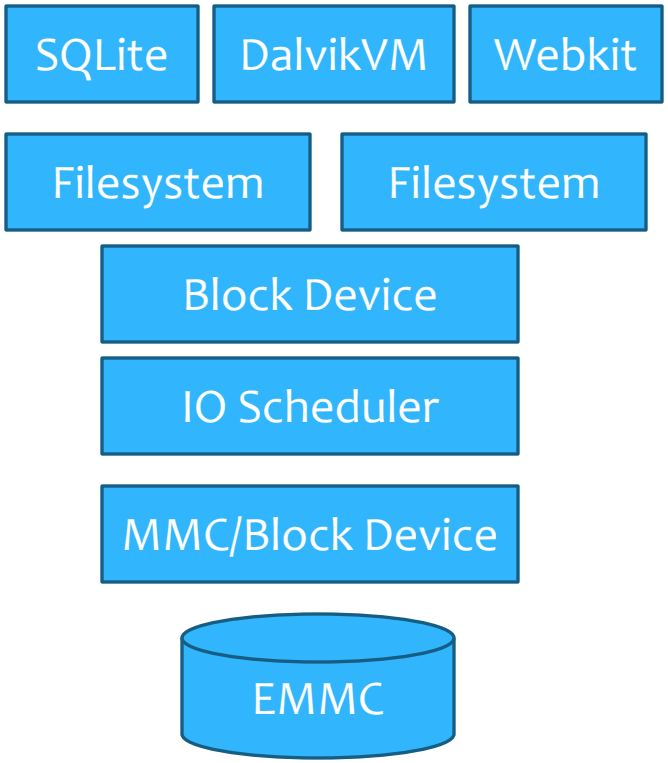
* Tablets with lots of DRAM
* Smartphones
* LCD-based eReaders
* Electronic Paper (eInk) eReaders
* LCD-based navigation devices
* Industrial devices, Loggers

DRAM is good

* Alleviates write performance issues
* Improves read times even further
* Reduces NAND wear

Areas

* User space
* Filesystem type
* Filesystem layout
* IO Scheduler
* Block IO & Cache
* MMC bus driver



Android challenges

* Vendors claim mostly sequential reads/writes
* Content Providers
* SQLite activity
* Many sync writes from userspace

Android system

* Boot
* Fixed battery
* Clean shut-down
* Warm reboots
* IO scheduler

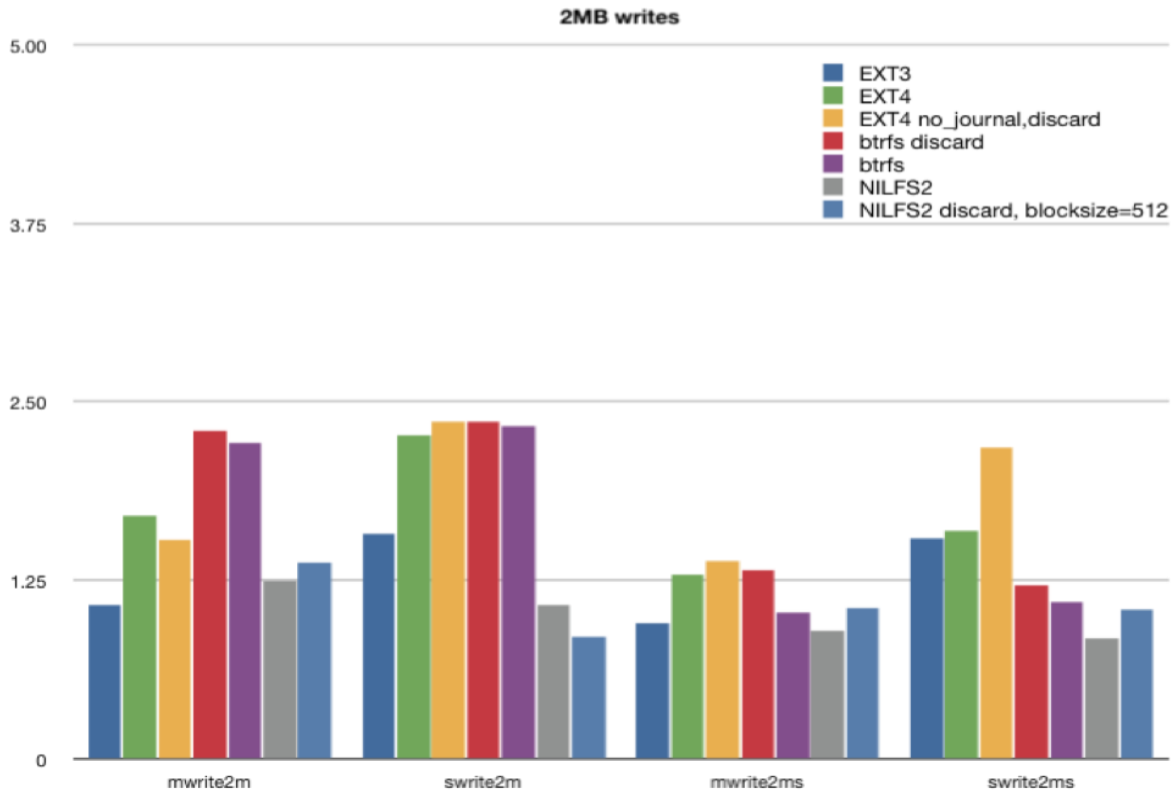
Cache is king!

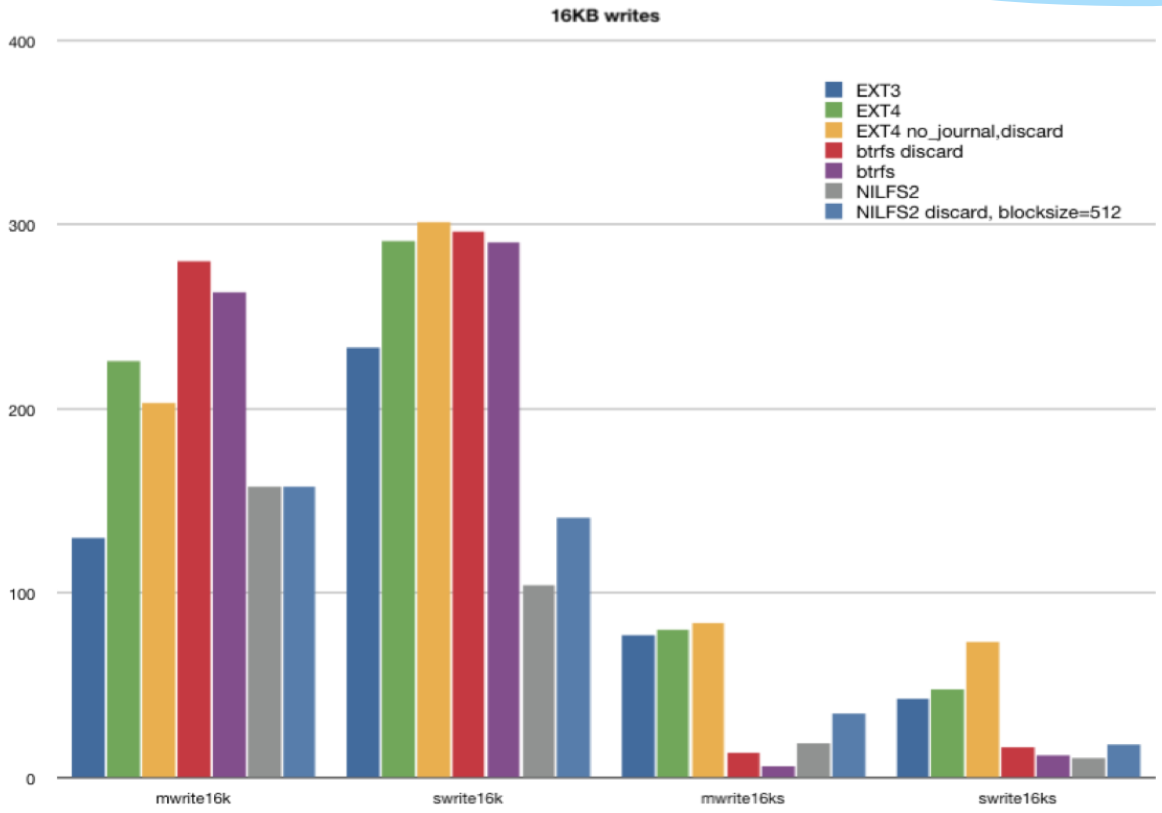
* Plenty of RAM allows large block cache
* Low memory killer
* Tune flushing thresholds

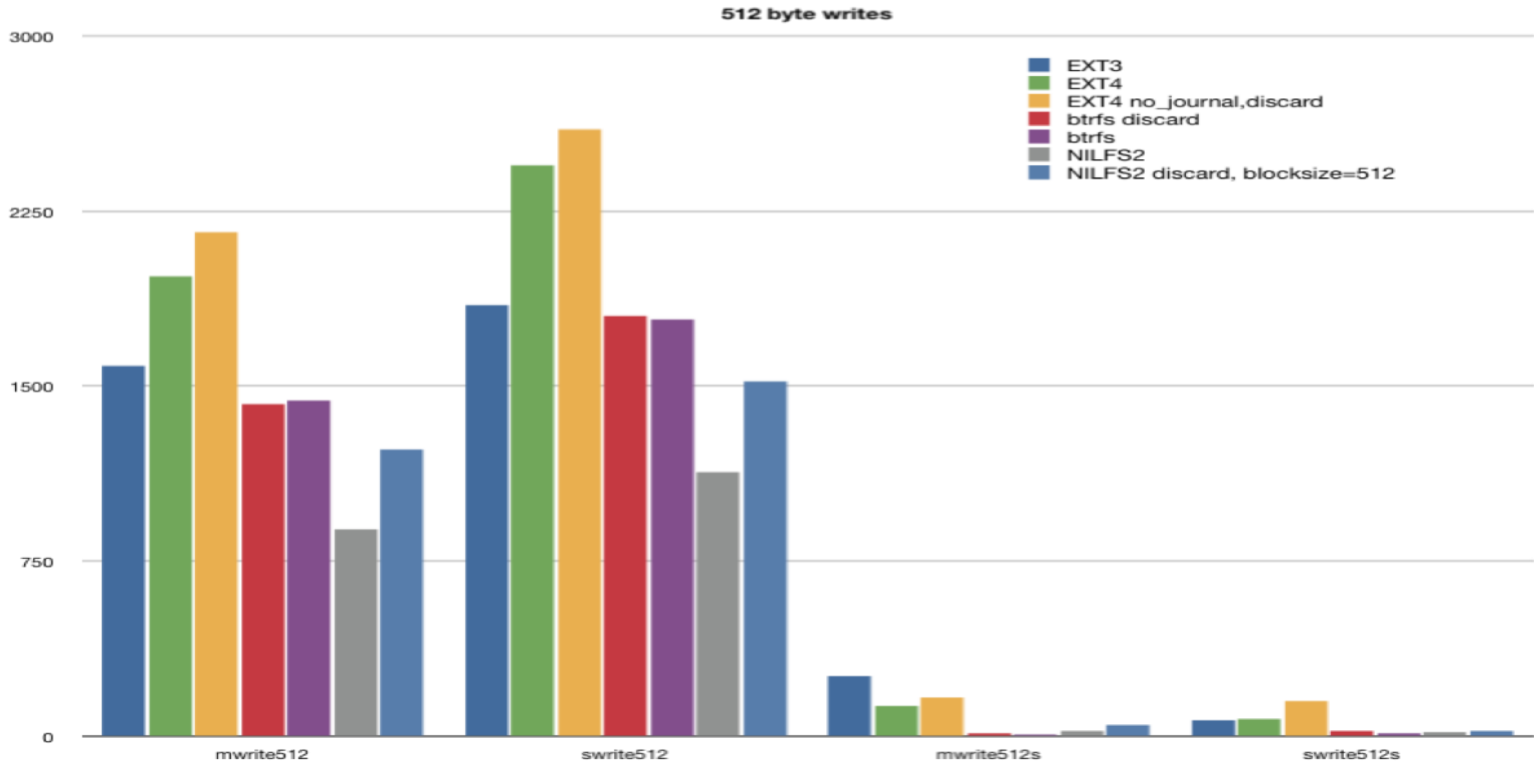
Filesystems

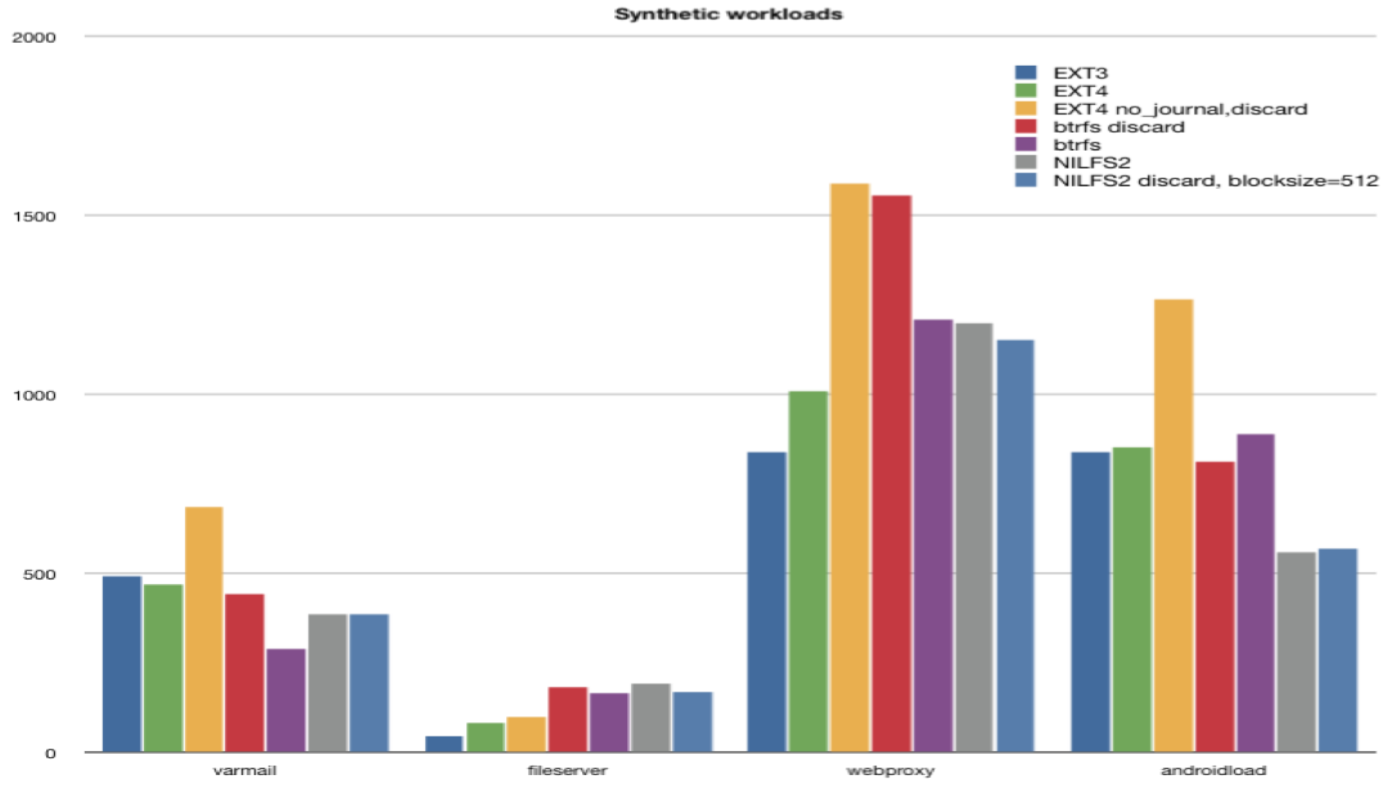
* Focus on write performance
* Tests run using fsbench (3.0 kernel, OMAP3 aka Nook Color)
* EXT4, BTRFS,NILFS2

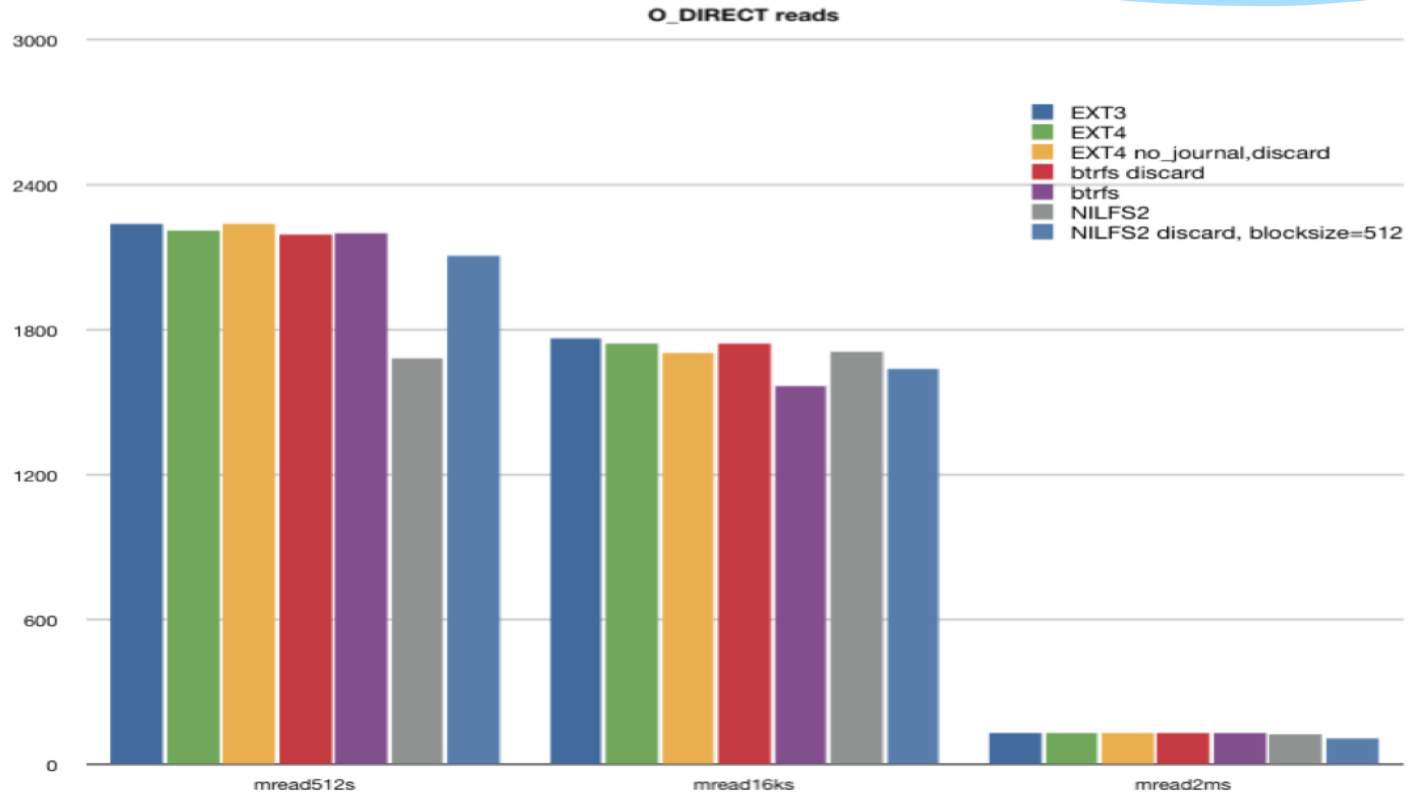
Benchmarks


2MB writes
5.00
3.75
2.50
1.25
0
EXT3
EXT4
EXT4 no_journal,discard
btrfs discard
btrfs
NILFS2
NILFS2 discard, blocksize=512
mwrite2m
swrite2m
mwrite2ms
swrite2ms


16KB writes
400
300
200
100
0
EXT3
EXT4
EXT4 no_journal,discard
btrfs discard
btrfs
NILFS2
NILFS2 discard, blocksize=512
mwrite16k
swrite16k
mwrite16ks
swrite16ks


512 byte writes
3000
2250
1500
750
0
EXT3
EXT4
EXT4 no_journal,discard
btrfs discard
btrfs
NILFS2
NILFS2 discard, blocksize=512
mwrite512
swrite512
mwrite512s
swrite512s


Synthetic workloads
2000
1500
1000
500
0
EXT3
EXT4
EXT4 no_journal,discard
btrfs discard
btrfs
NILFS2
NILFS2 discard, blocksize=512
varmail
fileserver
webproxy
androidload


O_DIRECT reads
3000
2400
1800
1200
600
0
EXT3
EXT4
EXT4 no_journal,discard
btrfs discard
btrfs
NILFS2
NILFS2 discard, blocksize=512
mread512s
mread16ks
mread2ms

EXT4 - a write

* Journal write (usually ~16K)
* inode update (usually 4K)
* Data goes into page cache

EXT4 w/o journal

* Not too dangerous on embedded systems with battery
* Good performance due to improved sequentiality

BTRFS

* If not using a lot of fsync/fdatasync
* Great large write performance
* Terrible on small/mediun sync writes
* Good performance on multiple writes

NILFS2

* Consistent performance
* Potentially much faster if eMMC part has fast sequential performance
* Should theoretically be the fastest :-)

EXT4 with journal

* If journaling is needed, consider RAM journal device
* Again RAM journal not as dangerous as you think
* Better than BTRFS on small/medium sync writes

I/O schedulers

* CFQ, noop, deadline
* Results are similar within ~10% range
* QOS considerations are more important than throughput

User space

* Avoid synchronization on files
* Avoid sync/fsync/fdatasync/etc
* Avoid small writes to files, better to buffer
* Don’t be afraid to read, be afraid to write!

Conclusion

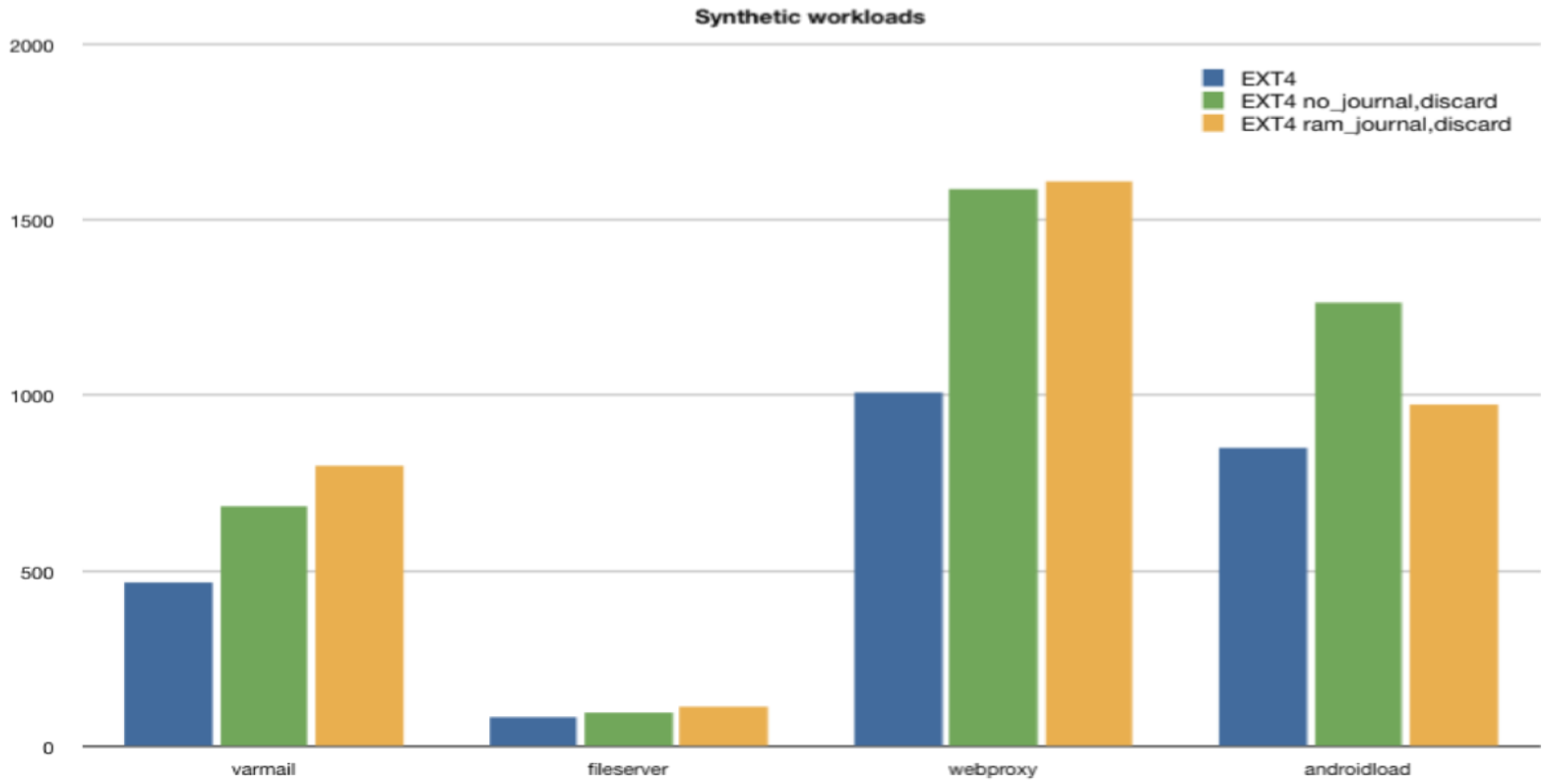
* EXT4 (discard, ram/no journal) is probably your best bet
* Try out a couple of configurations for the eMMC you are targeting
* Avoid writes! :-)

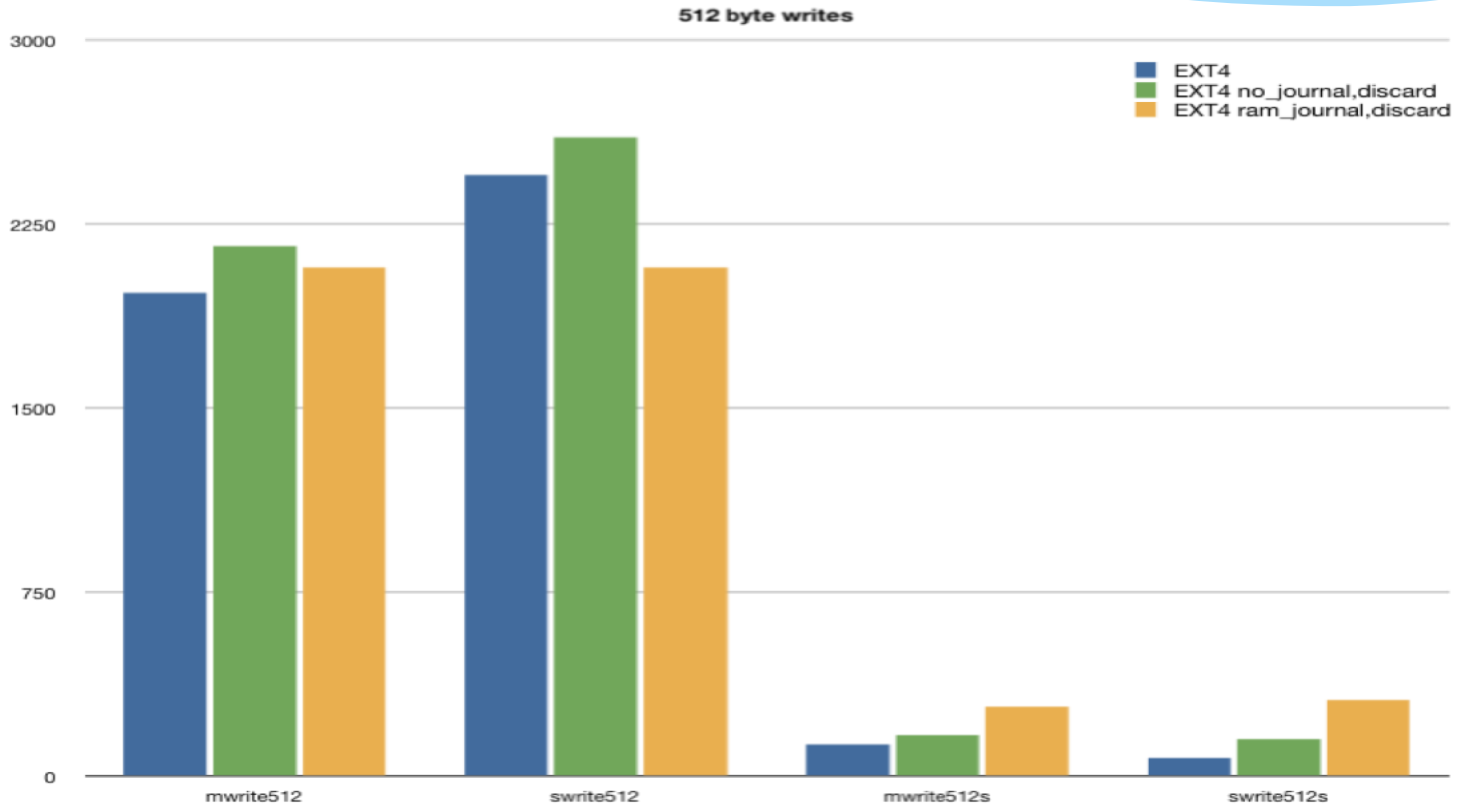
Questions?

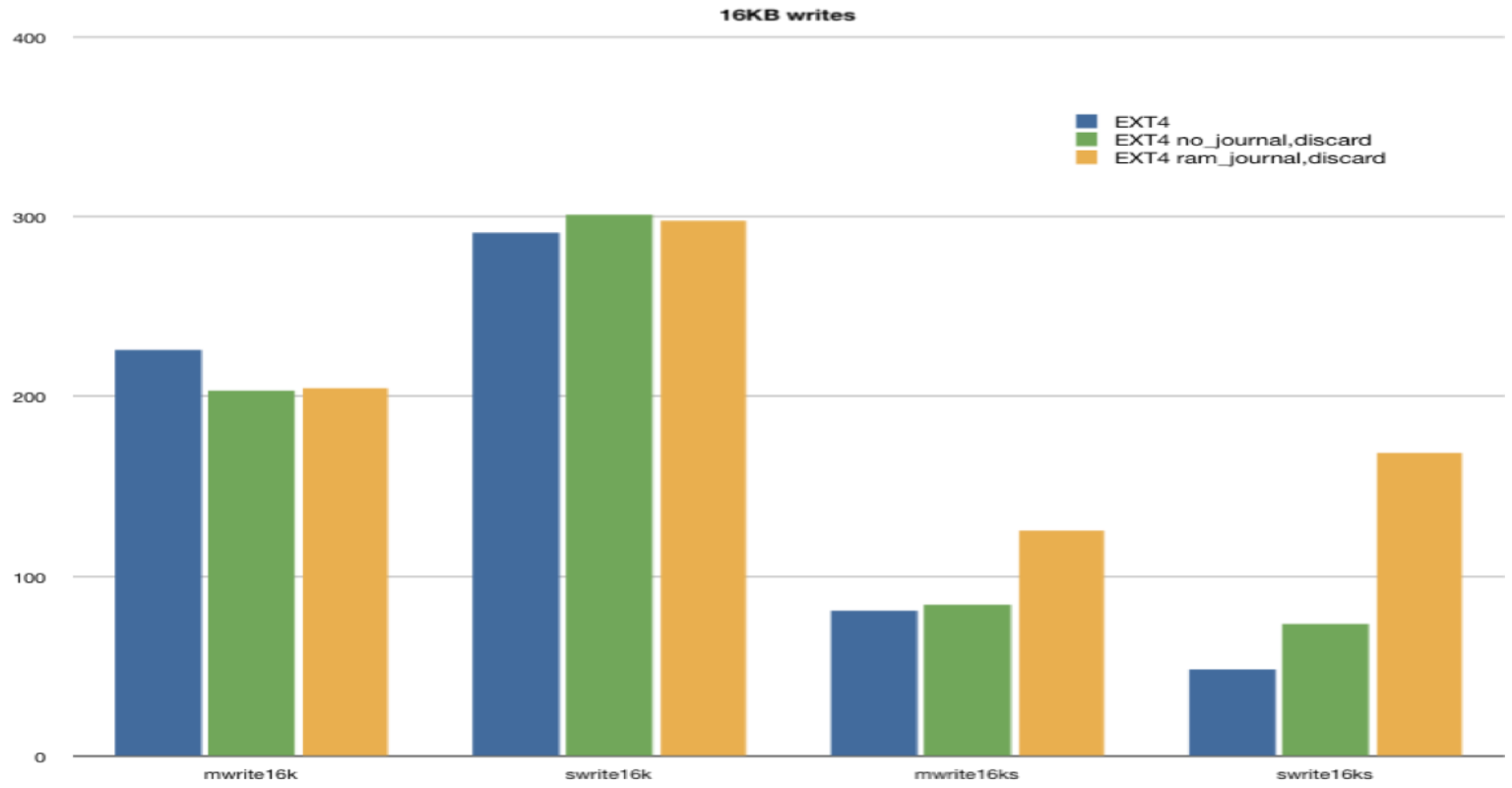
Appendix

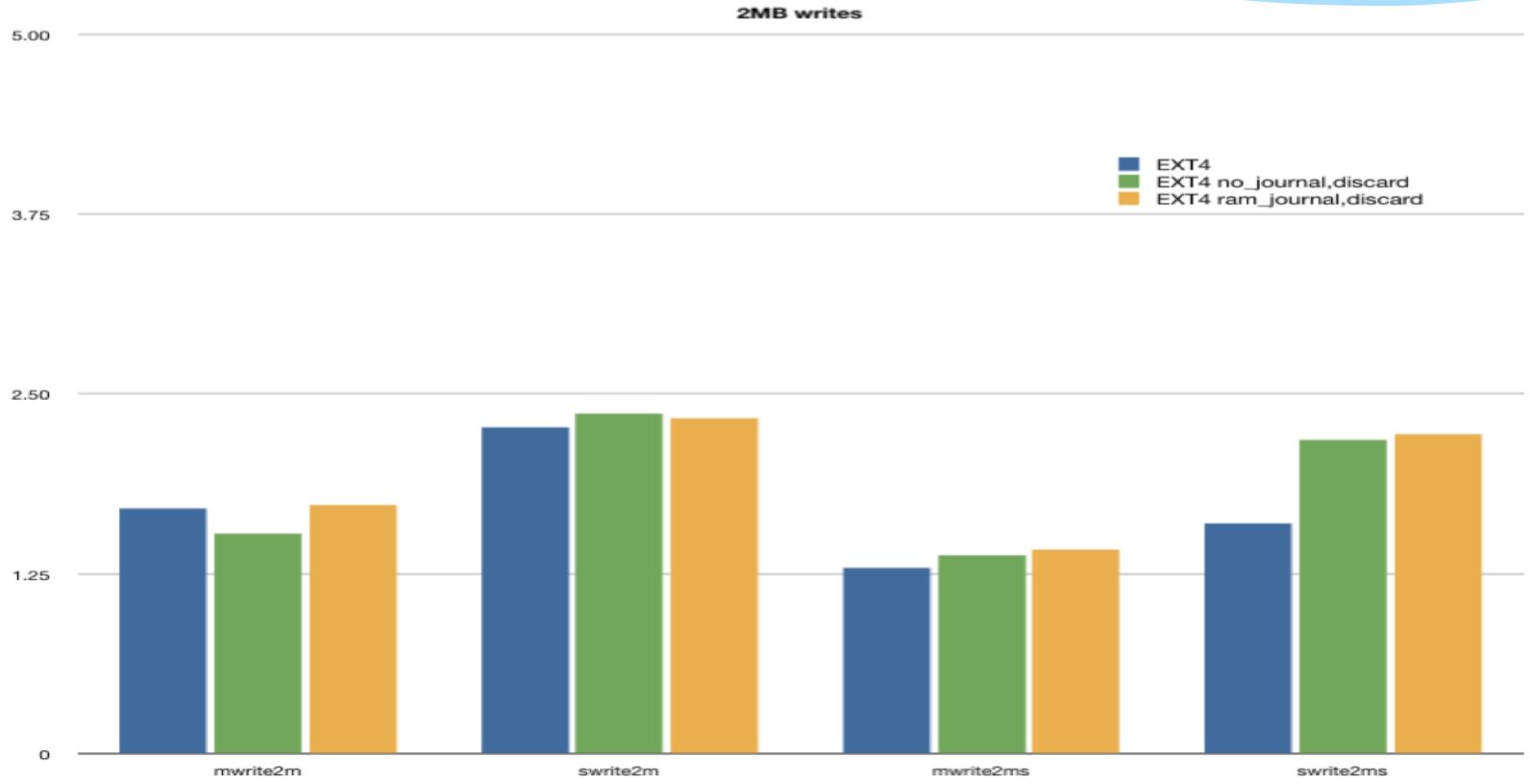
CONFIDENTIAL INFORMATION

EXT4 with RAM journal




512 byte writes
3000
2250
1500
750
0
EXT4
EXT4 no_journal,discard
EXT4 ram_journal,discard
mwrite512
swrite512
mwrite512s
swrite512s


16KB writes
400
300
200
100
0
EXT4
EXT4 no_journal,discard
EXT4 ram_journal,discard
mwrite16k
swrite16k
mwrite16ks
swrite16ks


2MB writes
5.00
3.75
2.50
1.25
0
EXT4
EXT4 no_journal,discard
EXT4 ram_journal,discard
mwrite2m
swrite2m
mwrite2ms
swrite2ms