Truth about eMMC performance benchmark

Andrew Lee

andrew@elixirflash.com

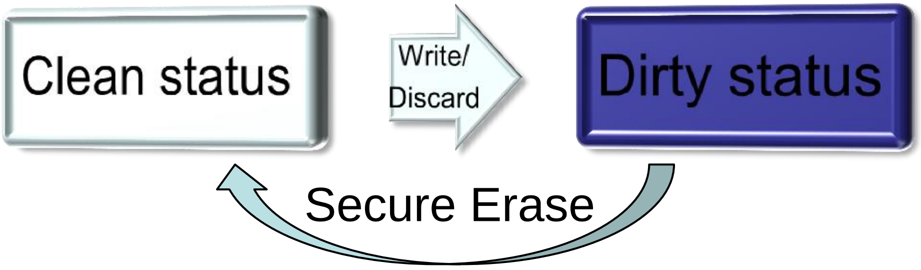
CEO, Elixir Flash Technology

Agenda

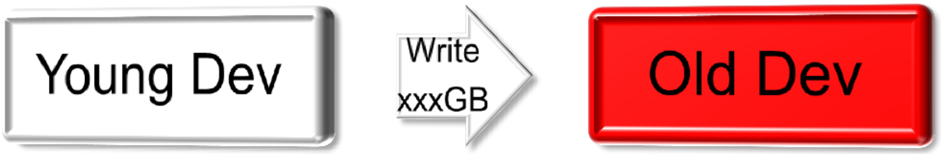
- Preconditions for eMMC performance
- Limitations of existing benchmarks
- App-based benchmark, eBench : Storage's impact on UX
- Summary

Two pre-conditions for NAND storage

- Dirty level of NANDs
  - GC freq., count and allocation of valid pages, …



- Aging level => not cleared by secure_erase
  - the accumulated FTL-meta like bad blocks…



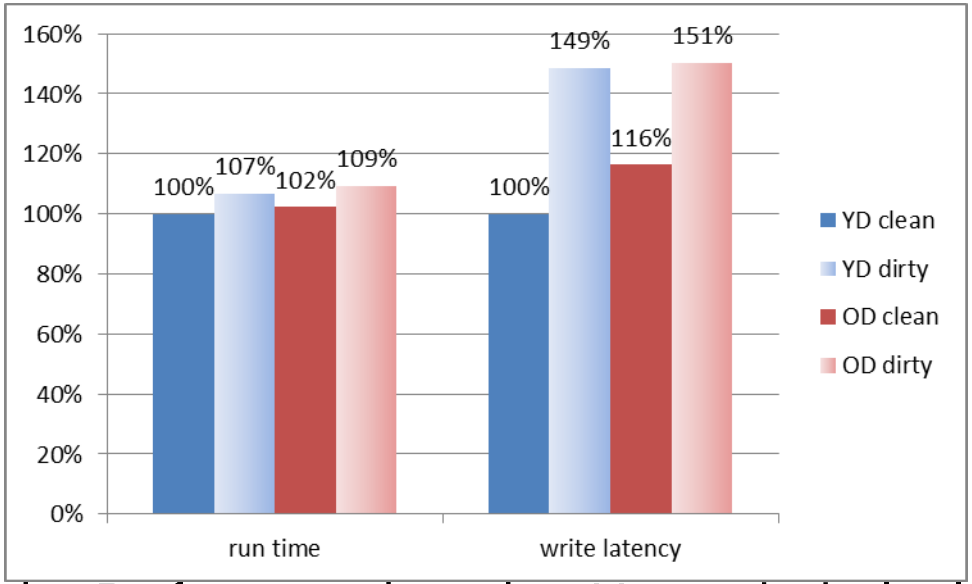
Our Targets

- Young Device(YD) v.s. Old Device(OD)
  - Old Device (OD) made by 16GX30 write-traffic

- Clean status v.s. Dirty status
  - Dirty status made by 19G write-traffic

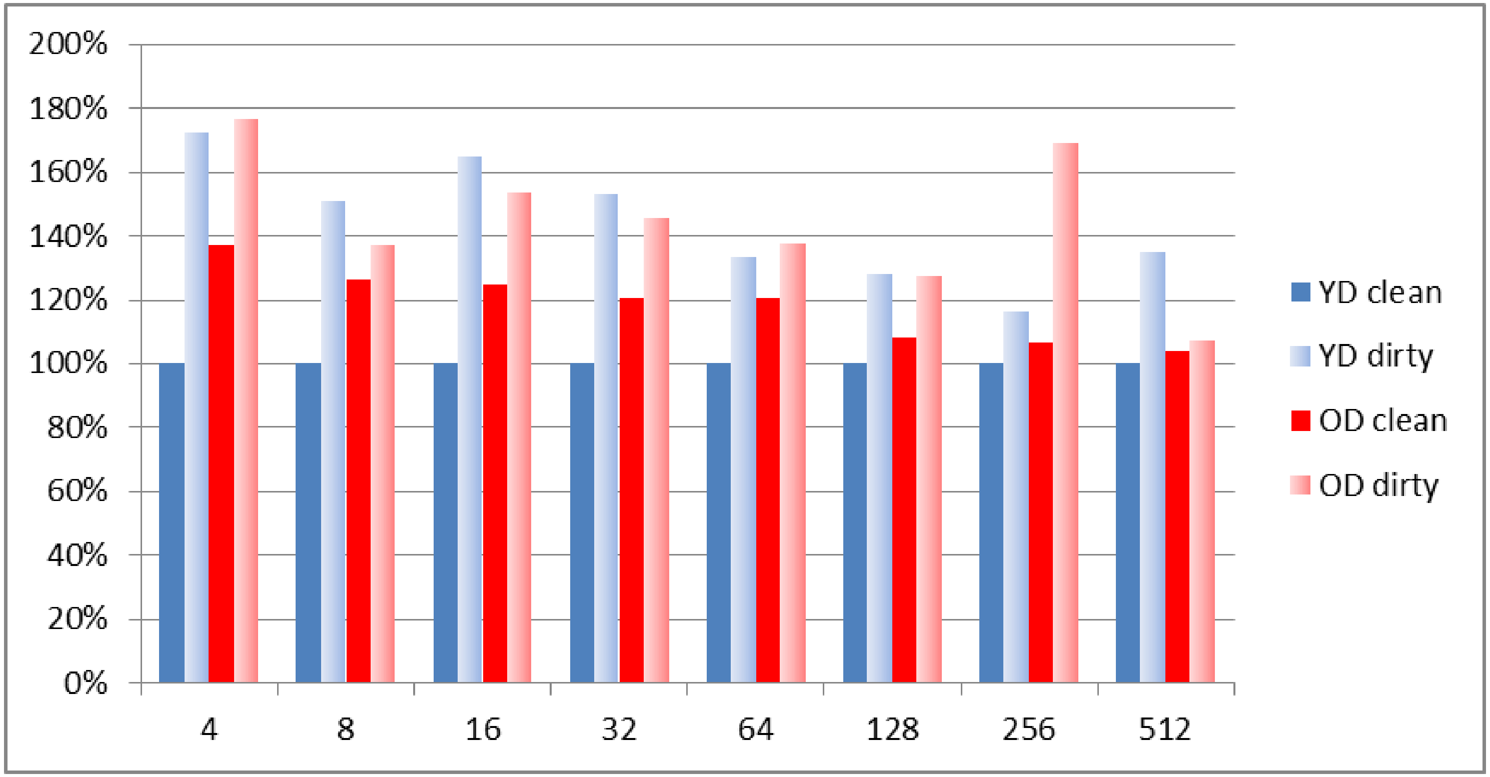
Two pre-condition's effect

- Clean status >> Dirty status : Big diff
  - 50% longer write-latency, 7~9% longer runtime
- Young dev(YD) > Old dev(OD) : relatively small diff
  - clean status:16% longer latency and 2% longer runtime



Relative Performance based on Young-device's clean status

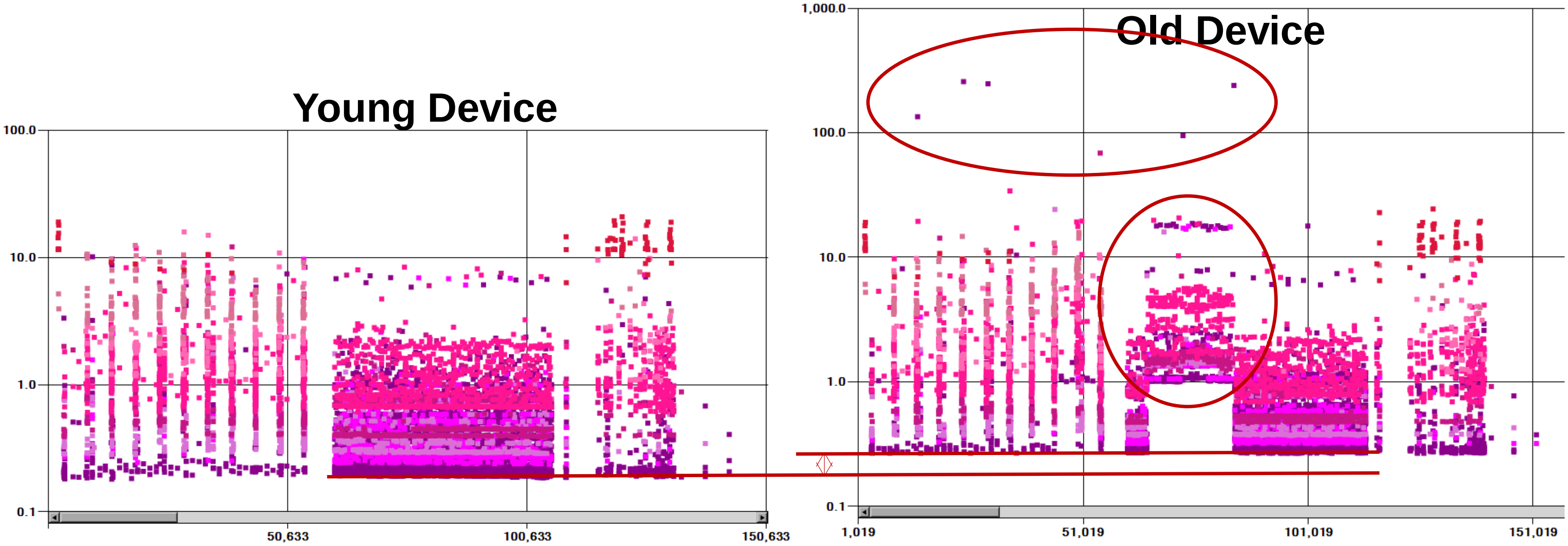
Write latency by Chunk



Write Chunks (KB)

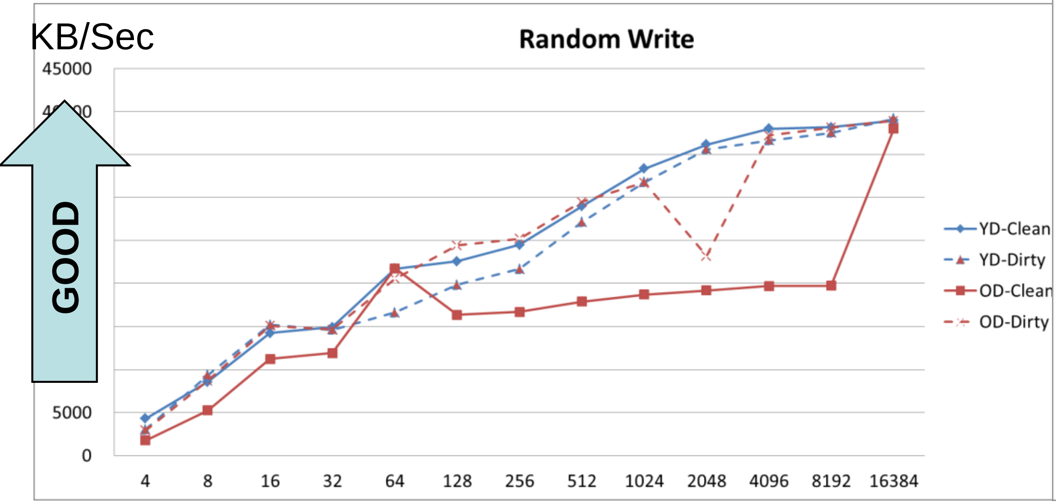
Write latency of Old device

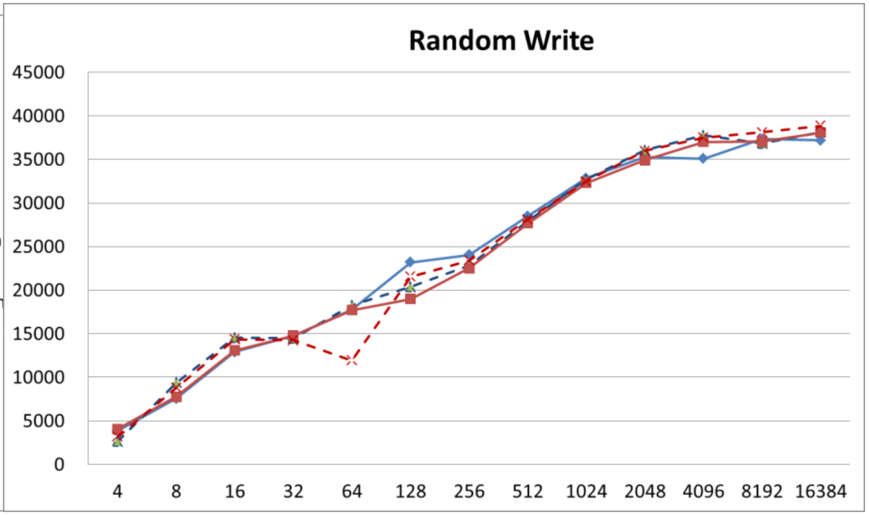
- High Write-latencies over 100ms
- The fastest latency group is higher than YD’s
- Strange “slow period”



Benchmarks (1) : IOZONE 16M

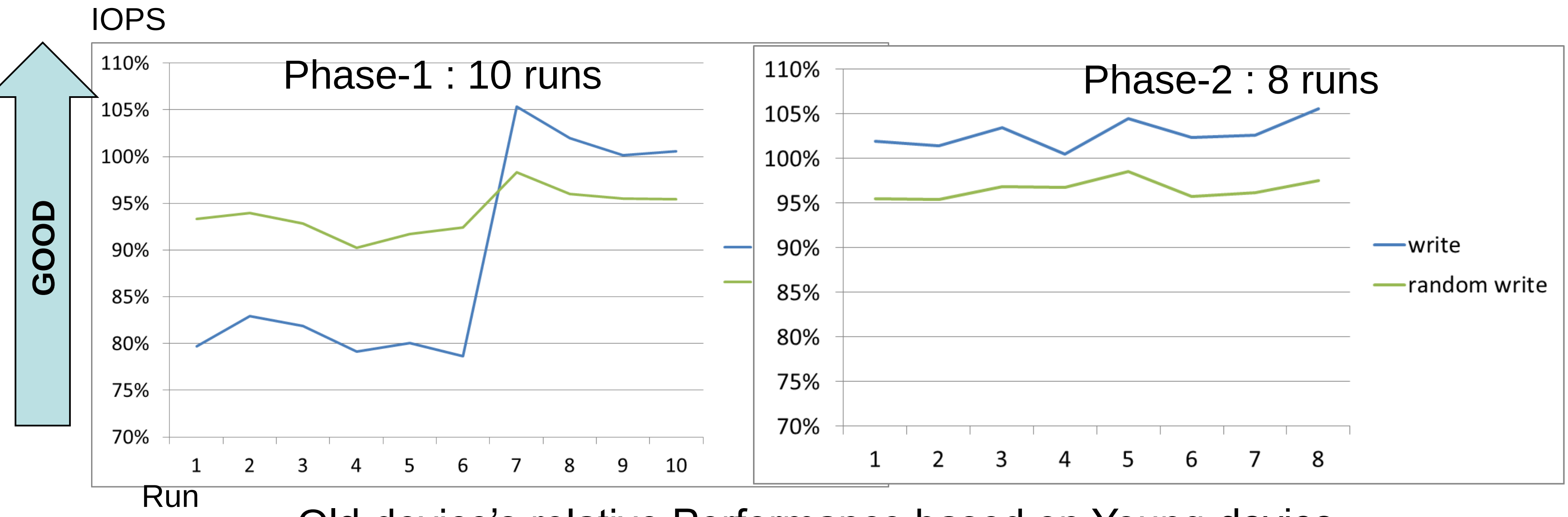
- OD-Clean of every run shows different patterns





Benchmarks (2) : IOZONE 1G-4K

- Even iozone's big area is unstable
  - Old device's "write" is superior on Phase-2, and not bad on the rear part of Phase-1



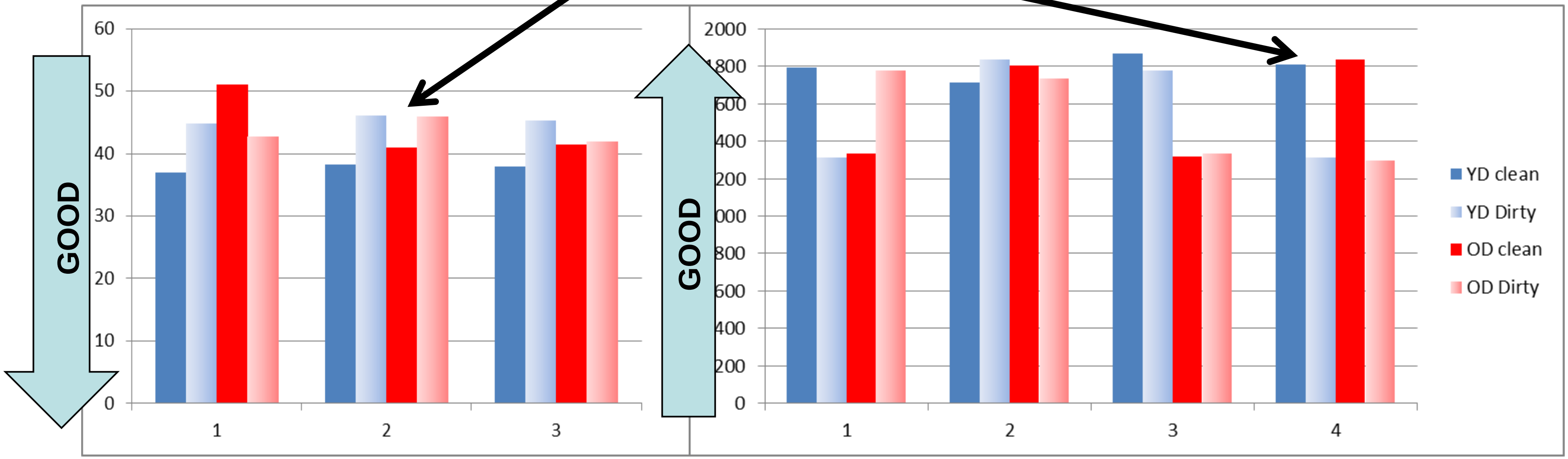
Old-device's relative Performance based on Young-device

Benchmarks (3):RLBench,Qurdrant

Only two Expected Results

RL Bench

Qurdrant



Run

Run

The limitations of benchmark

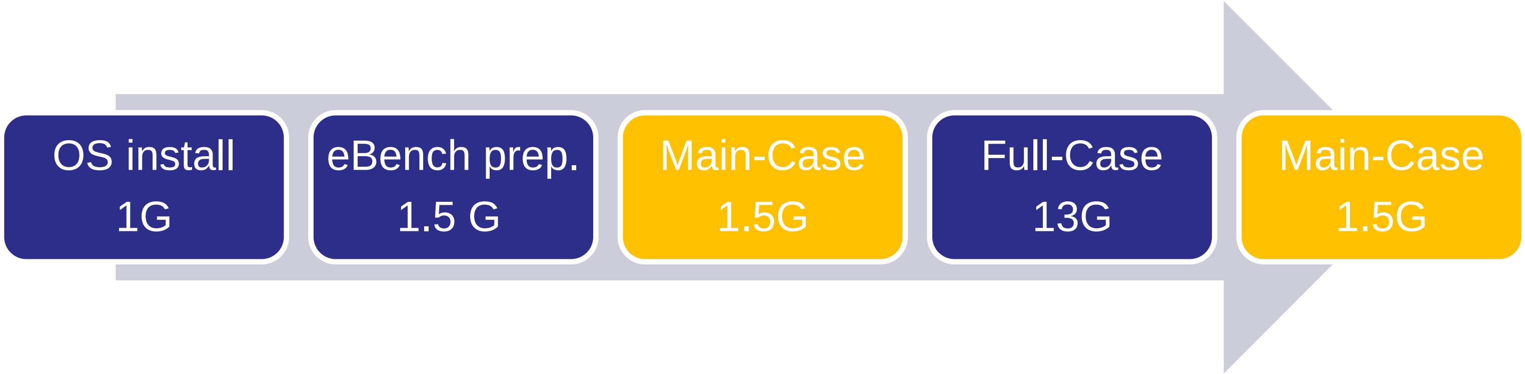
- Small amount of write-traffic
- Simple and synthetic write-workload
- Can’t show storage’s impacts on UX

Ebench : app-based bench

- Showing storage’s impact on UX
- App-based bench: not synthetic workload
  - web, contacts, install, camera, gallery so on
- full test-case
  - Using camera and install-app, make file system to be full-status (dirty-status)
- Including FS and app’s behavior
- Dirty status is more important in terms of real-world

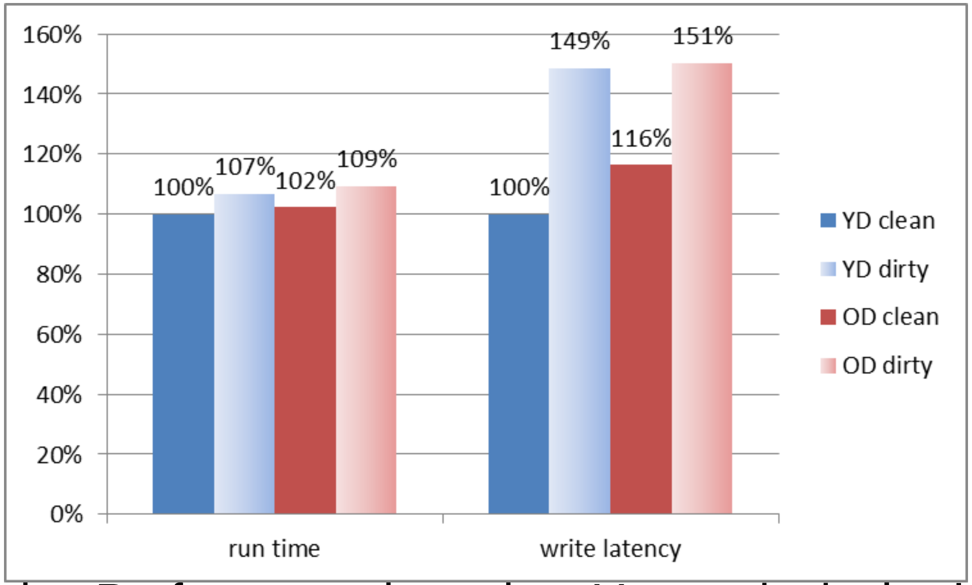
Ebench : example

- Configurations
  - Main-case : many accesses of web & DB
  - Full-case : make full-status of file system (device dirty)
- Sequence
  - Main-case -> Full-case -> Main-case



Ebench : main-case

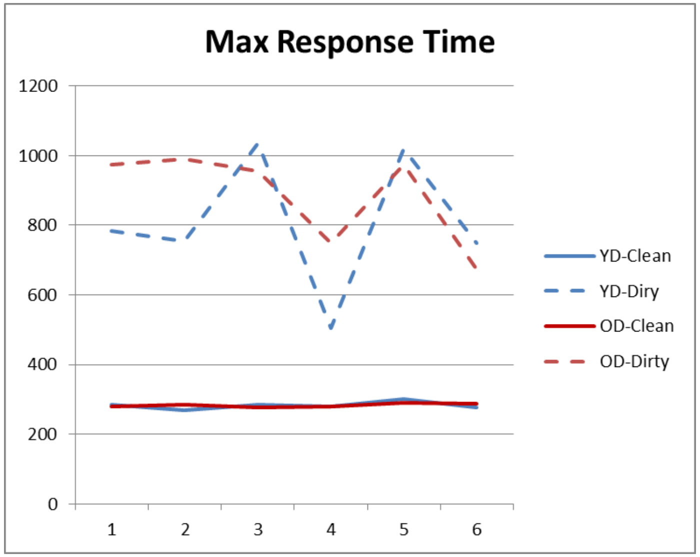
- Main-case results



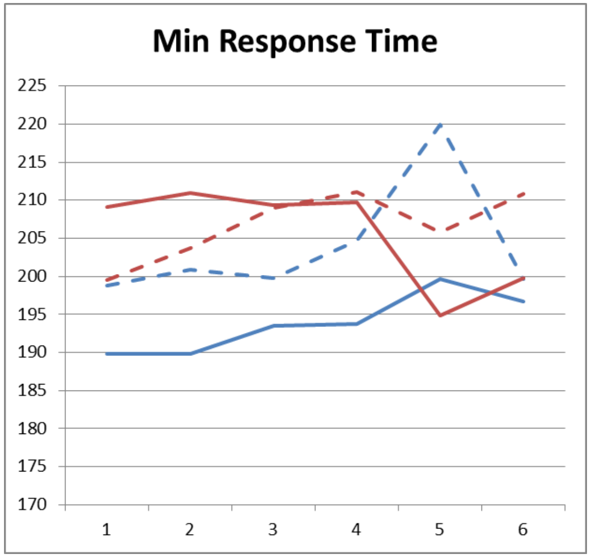
Relative Performance based on Young-device’s clean status

Ebench : Response time

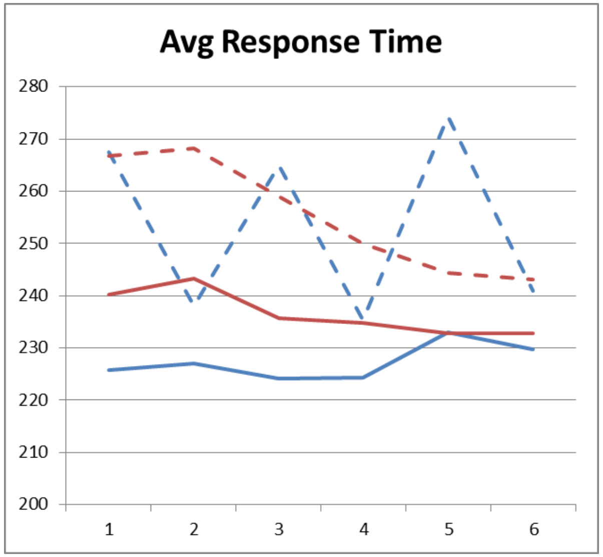
- Response means UX’s “done”
- Contact’s min/max/avg
  - longest Max is 1 sec to insert a record into contact



Max Response Time



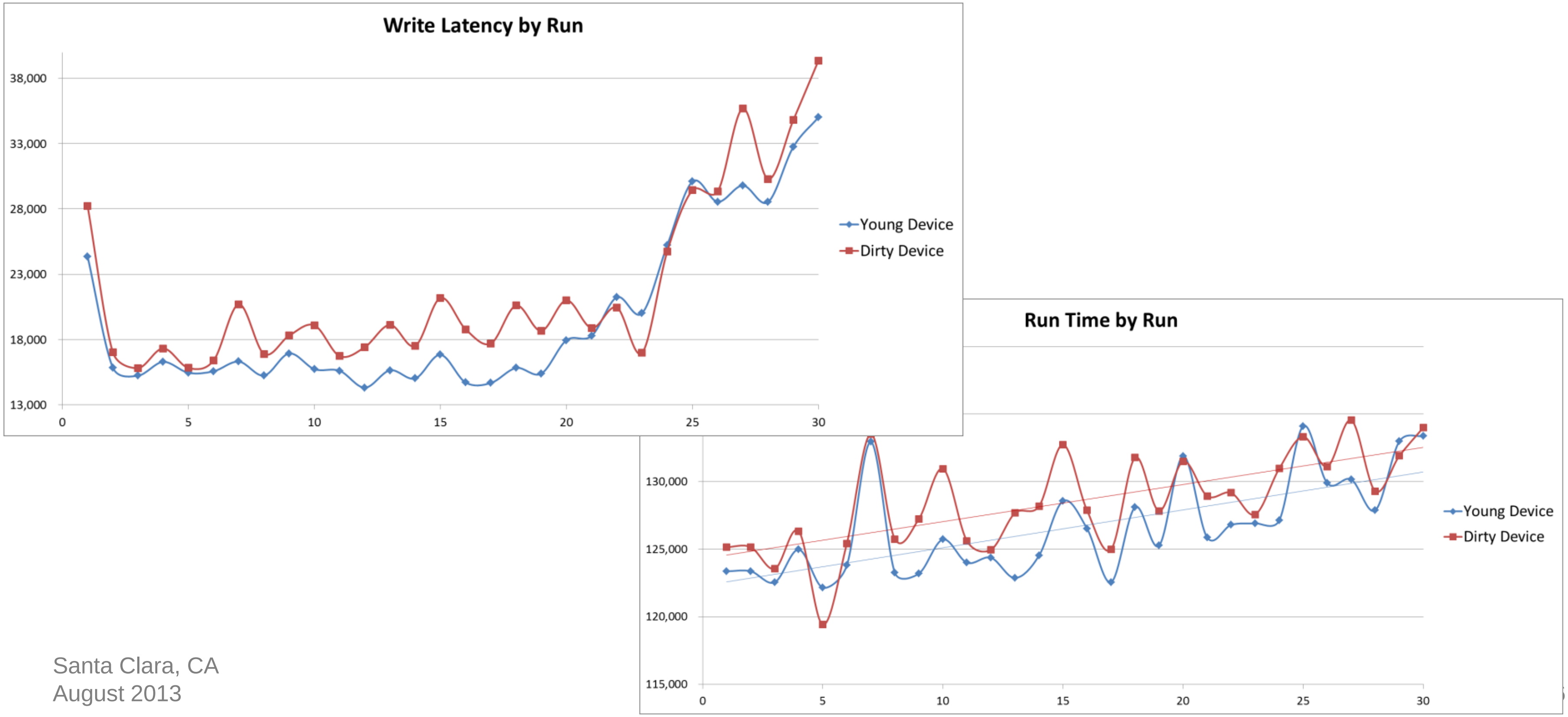
Min Response Time



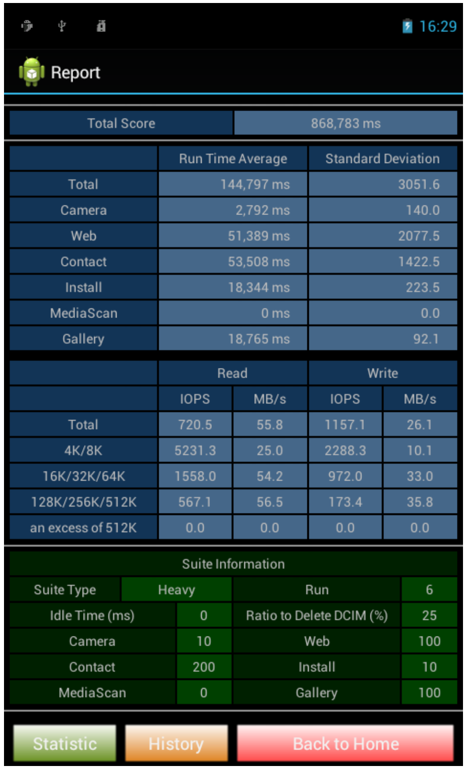
Avg Response Time

Ebench : Full-case

- Make 95% full of file system by 30 runs
- YD shows better about 10% latency/2% run-time



eBench : Final Report View



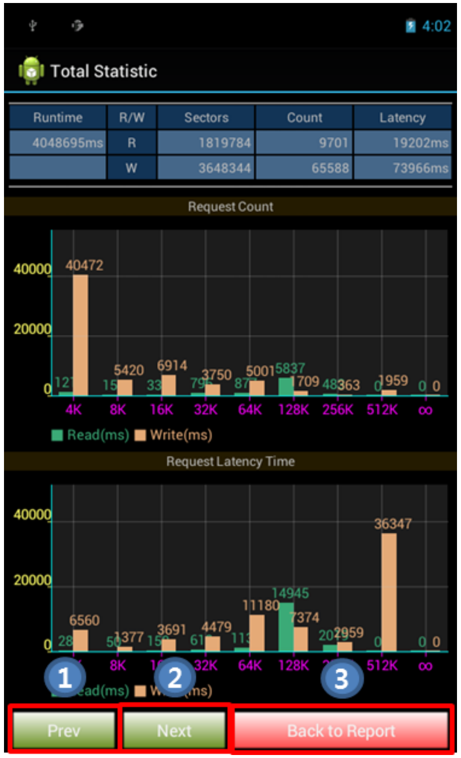
Total Score	868,783 ms

	Run Time Average	Standard Deviation
Total	144,797 ms	3051.6
Camera	2,792 ms	140.0
Web	51,389 ms	2077.5
Contact	53,508 ms	1422.5
Install	18,344 ms	223.5
MediaScan	0 ms	0.0
Gallery	18,765 ms	92.1

	Read		Write	
	IOPS	MB/s	IOPS	MB/s
Total	720.5	55.8	1157.1	26.1
4K/8K	5231.3	25.0	2288.3	10.1
16K/32K/64K	1558.0	54.2	972.0	33.0
128K/256K/512K	567.1	56.5	173.4	35.8
an excess of 512K	0.0	0.0	0.0	0.0

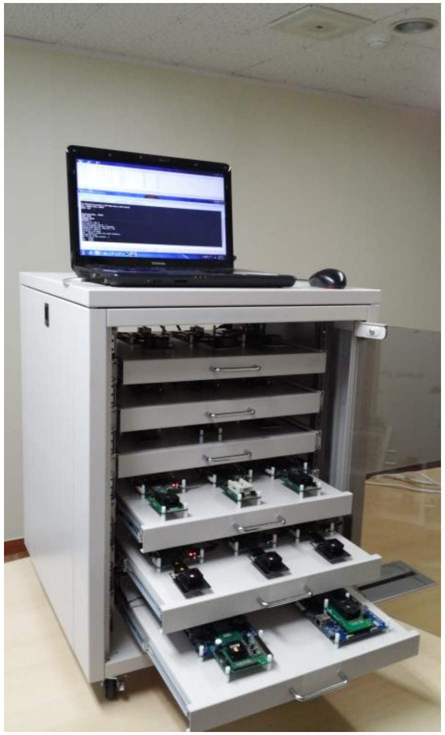
Suite Information

Suite Type	Heavy	Run	6
Idle Time (ms)	0	Ratio to Delete DCIM (%)	25
Camera	10	Web	100
Contact	200	Install	10
MediaScan	0	Gallery	100

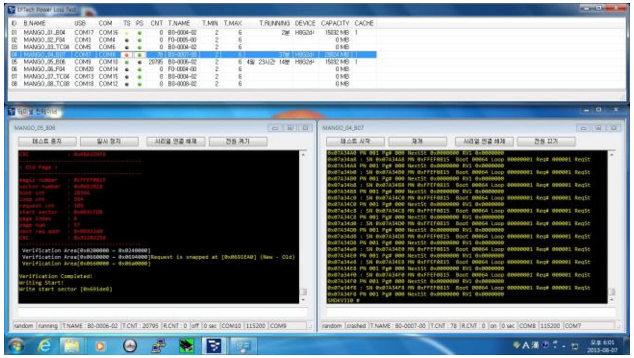


Runtime	R/W	Sectors	Count	Latency
4048695ms	R	1819784	9701	19202ms
	W	3648344	65588	73966ms

EF Storage Tester



- Power-cycle/Aging/Performance testing
- “Faster and Wider” smart test-cases
- Validation of eMMC 4.5 spec
- Supporting 64 devices simultaneously



ID	B.NAME	USB	COM	TS	PS	CNT	T.NAME	T.MIN	T.MAX	T.RUNNING	DEVICE	CAPACITY	CACHE
01	MANGO_01_B04	COM17	COM16	●	●	0	B0-0004-02	2	6	2분	H8G2d	15032 MB	1
02	MANGO_02_F04	COM3	COM4	●	●	0	F0-0005-00	2	6			0 MB	
03	MANGO_03_TC04	COM5	COM6	●	●	0	B0-0004-02	2	6			0 MB	
04	MANGO_04_B07	COM7	COM8	●	●	70	B0-0007-00	2	6	37분	H8G2d	[illegible] MB	1
05	MANGO_05_B06	COM9	COM10	●	●	20795	B0-0006-02	2	6	4일 23시간 14분	H8G2d	15032 MB	1
06	MANGO_06_F04	COM20	COM14	●	●	0	F0-0004-00	2	6			0 MB	
07	MANGO_07_TC04	COM13	COM15	●	●	0	B0-0004-02	2	6			0 MB	
08	MANGO_08_TC08	COM18	COM12	●	●	0	B0-0008-02	2	6			0 MB	

```
CRC              : [illegible]
-----------------------------------------
< Old Page >
-----------------------------------------
magic number     : [illegible]
sector number    : [illegible]
boot cnt         : [illegible]
loop cnt         : [illegible]
request cnt      : [illegible]
start sector     : [illegible]
page index       : [illegible]
page num         : [illegible]
next req addr    : [illegible]
CRC              : [illegible]
-----------------------------------------
 Verification Area[0x0200000 - 0x0240000]
 Verification Area[0x0160000 - 0x0194000]Request is snapped at [0x0691EA0] (New - Old)
 Verification Area[0x0660000 - 0x06a0000]

Verification Completed!
Writing Start!
Write start sector [0x691de0]
```

random | running | T.NAME : B0-0006-02 | T.CNT : 20795 | R.CNT : 0 | off | 0 sec | COM10 | 115200 | COM9

```
0x07A34A0 PN 001 Pg# 000 NextSt 0x0000000 RV1 0x0000000
0x07a34a8 : SN 0x07A34A8 MN 0xFFEF0815  Boot 00064 Loop 00000001 Req# 000001 ReqSt
0x07A34A8 PN 001 Pg# 000 NextSt 0x0000000 RV1 0x0000000
0x07a34b0 : SN 0x07A34B0 MN 0xFFEF0815  Boot 00064 Loop 00000001 Req# 000001 ReqSt
0x07A34B0 PN 001 Pg# 000 NextSt 0x0000000 RV1 0x0000000
0x07a34b8 : SN 0x07A34B8 MN 0xFFEF0815  Boot 00064 Loop 00000001 Req# 000001 ReqSt
0x07A34B8 PN 001 Pg# 000 NextSt 0x0000000 RV1 0x0000000
0x07a34c0 : SN 0x07A34C0 MN 0xFFEF0815  Boot 00064 Loop 00000001 Req# 000001 ReqSt
0x07A34C0 PN 001 Pg# 000 NextSt 0x0000000 RV1 0x0000000
0x07a34c8 : SN 0x07A34C8 MN 0xFFEF0815  Boot 00064 Loop 00000001 Req# 000001 ReqSt
0x07A34C8 PN 001 Pg# 000 NextSt 0x0000000 RV1 0x0000000
0x07a34d0 : SN 0x07A34D0 MN 0xFFEF0815  Boot 00064 Loop 00000001 Req# 000001 ReqSt
0x07A34D0 PN 001 Pg# 000 NextSt 0x0000000 RV1 0x0000000
0x07a34d8 : SN 0x07A34D8 MN 0xFFEF0815  Boot 00064 Loop 00000001 Req# 000001 ReqSt
0x07A34D8 PN 001 Pg# 000 NextSt 0x0000000 RV1 0x0000000
0x07a34e0 : SN 0x07A34E0 MN 0xFFEF0815  Boot 00064 Loop 00000001 Req# 000001 ReqSt
0x07A34E0 PN 001 Pg# 000 NextSt 0x0000000 RV1 0x0000000
0x07a34e8 : SN 0x07A34E8 MN 0xFFEF0815  Boot 00064 Loop 00000001 Req# 000001 ReqSt
0x07A34E8 PN 001 Pg# 000 NextSt 0x0000000 RV1 0x0000000
0x07a34f0 : SN 0x07A34F0 MN 0xFFEF0815  Boot 00064 Loop 00000001 Req# 000001 ReqSt
0x07A34F0 PN 001 Pg# 000 NextSt 0x0000000 RV1 0x0000000
0x07a34f8 : SN 0x07A34F8 MN 0xFFEF0815  Boot 00064 Loop 00000001 Req# 000001 ReqSt
0x07A34F8 PN 001 Pg# 000 NextSt 0x0000000 RV1 0x0000000
SHDKV310 #
```

random | crashed | T.NAME : B0-0007-00 | T.CNT : 78 | R.CNT : 0 | on | 0 sec | COM8 | 115200 | COM7

Conclusion

- eBench : New storage benchmark
    - Showing storage’s impact on UX
    - Covering wide storage-status from clean to dirty(full)
    - App-based benchmark, generating real-workload
    - Including FS and app’s behavior

Thank You

Elixir Flash Technology

andrew@elixirflash.com