

NVRAMOS 2013

Methods of eMMC performance benchmark

이 성우

andrew@elixirflash.com

Elixir Flash Technology

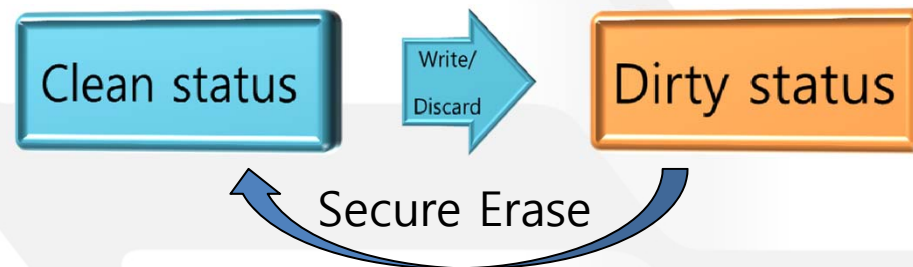
Proposal by . . .



- 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 frequency, count and allocation of valid pages, ...



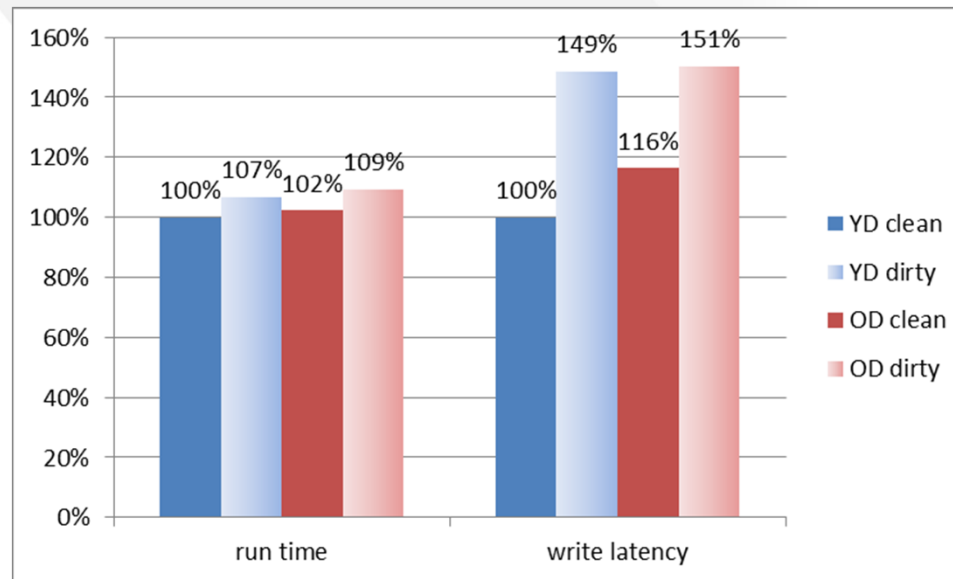
- Aging level => not cleared by secure_erase
 - the accumulated FTL-meta like bad blocks...



- Young Device(YD) v.s. Old Device(OD)
 - Old Device (OD) made by 16GX300 write-traffic
- Clean status v.s. Dirty status
 - Dirty status made by 19G write-traffic, after secure_erase

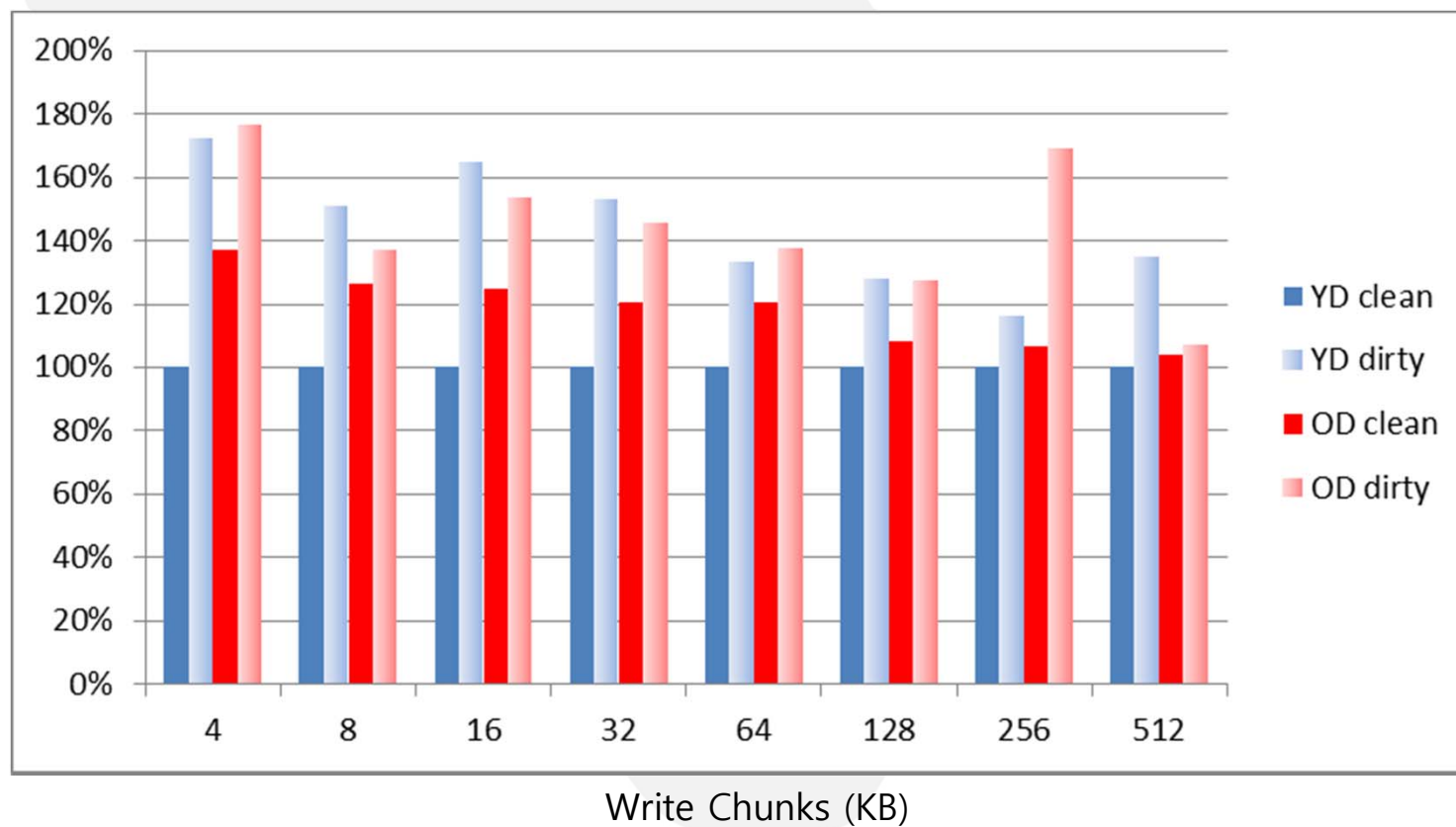
Two pre-condition's effect

- 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



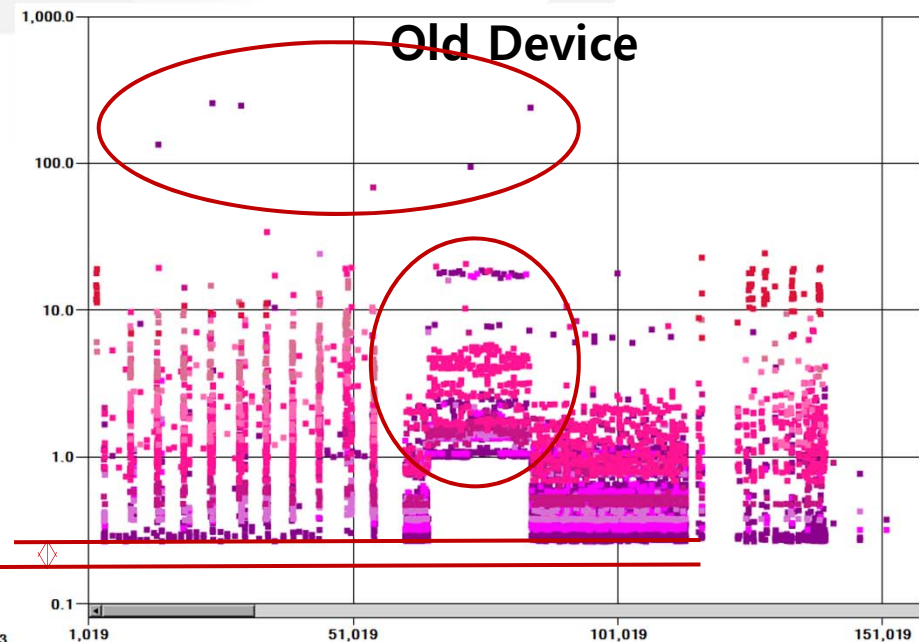
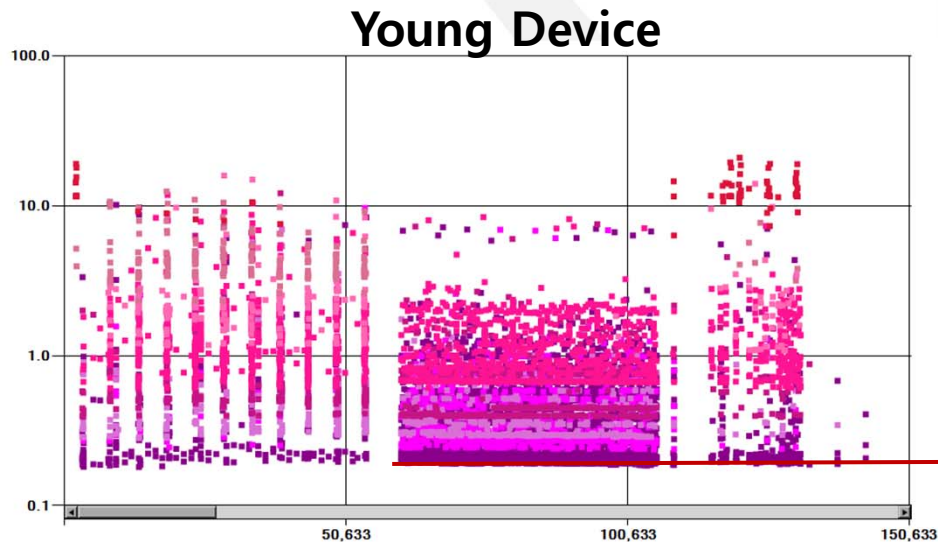
Relative Performance based on Young-device's clean status

Write latency by Chunk



Write latency of Old device

- High Write-latencies over 100ms
- The fastest latency group is higher than YD's
- Strange "slow period"



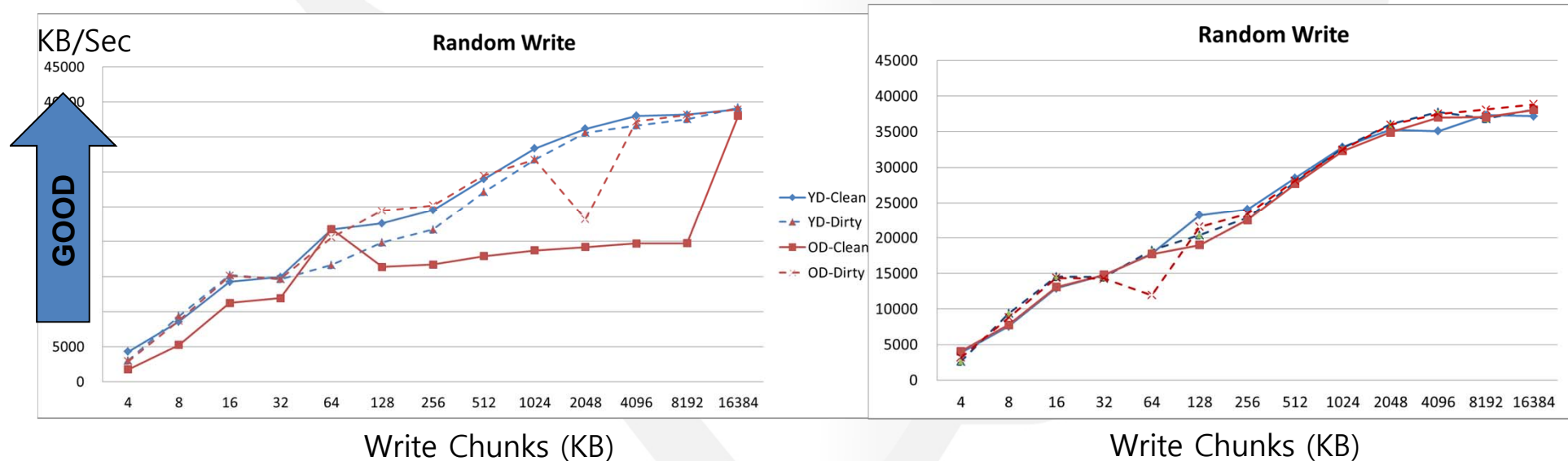


Benchmarks's limitation

Benchmarks (1) : IOZONE 16M



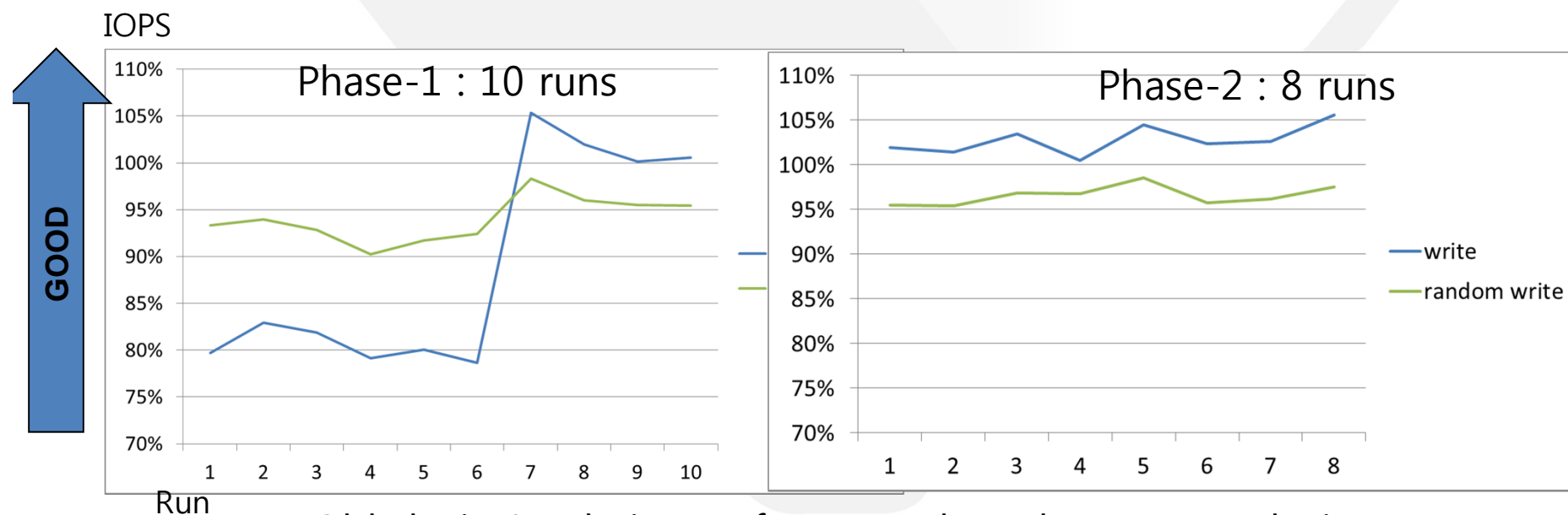
- OD-Clean of every run shows different patterns



Benchmarks (2) : IOZONE 1G-4K

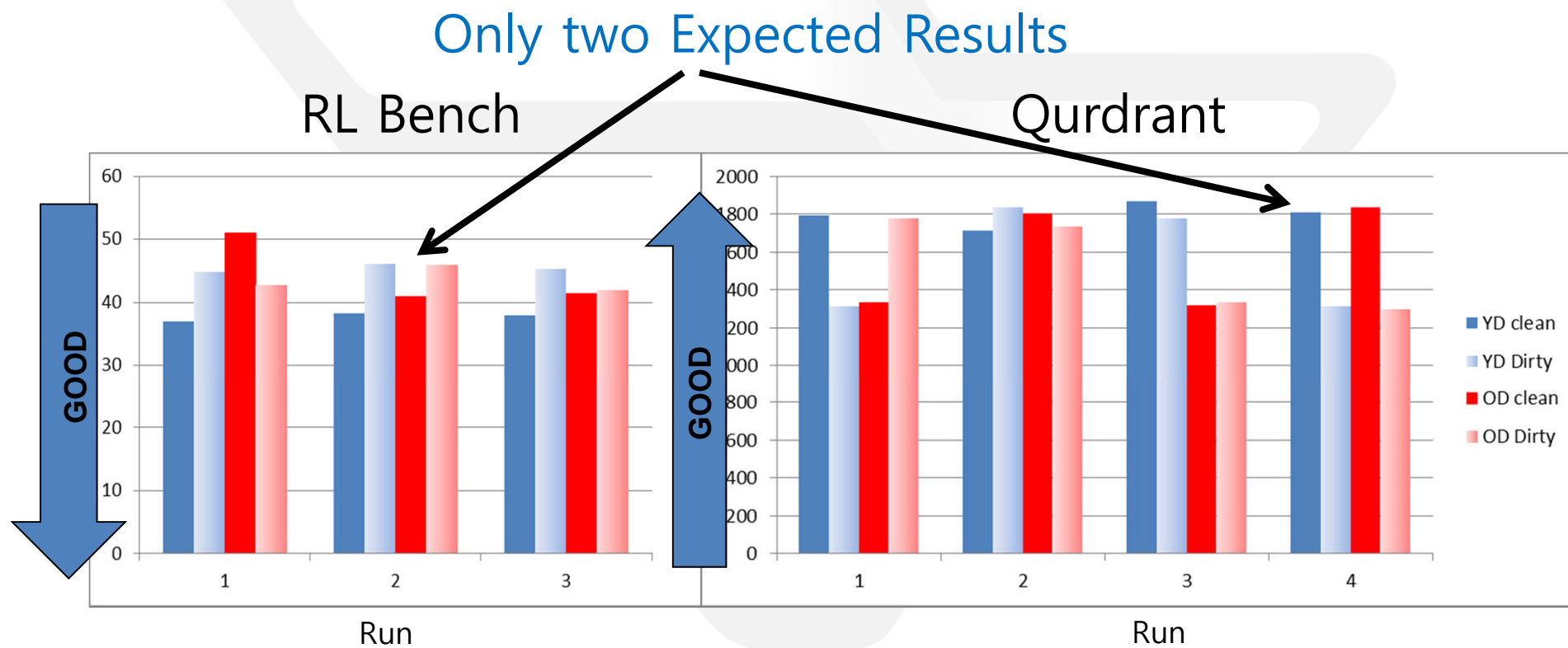


- Even iozone with big area is unstable
 - Old device's "write" is superior on Phase-2, and not bad on the rear part of Phase-1



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

Benchmarks (3):RLBench,Qurdrant



The limitations of benchmark



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



Right approach

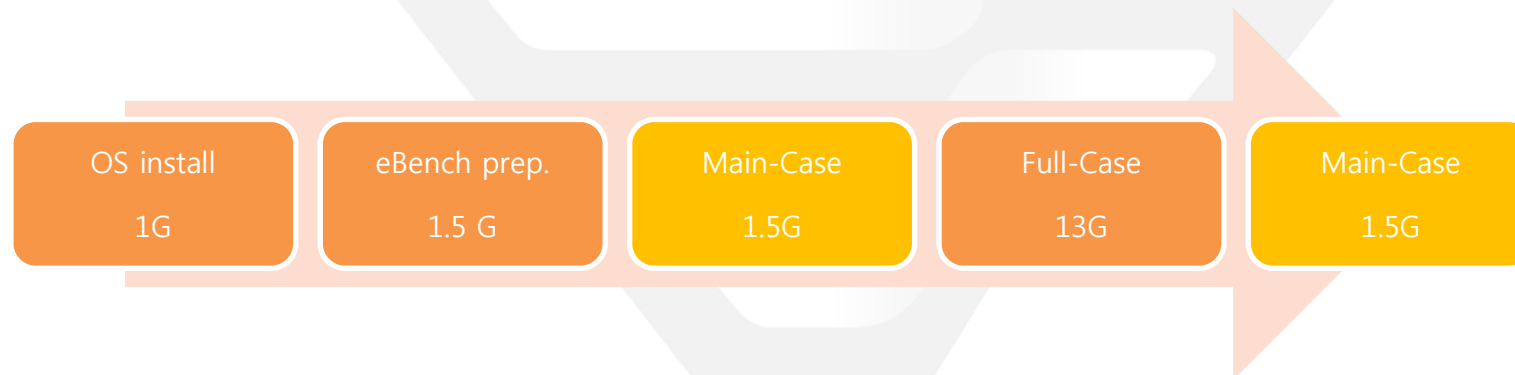
Ebench : app-based bench



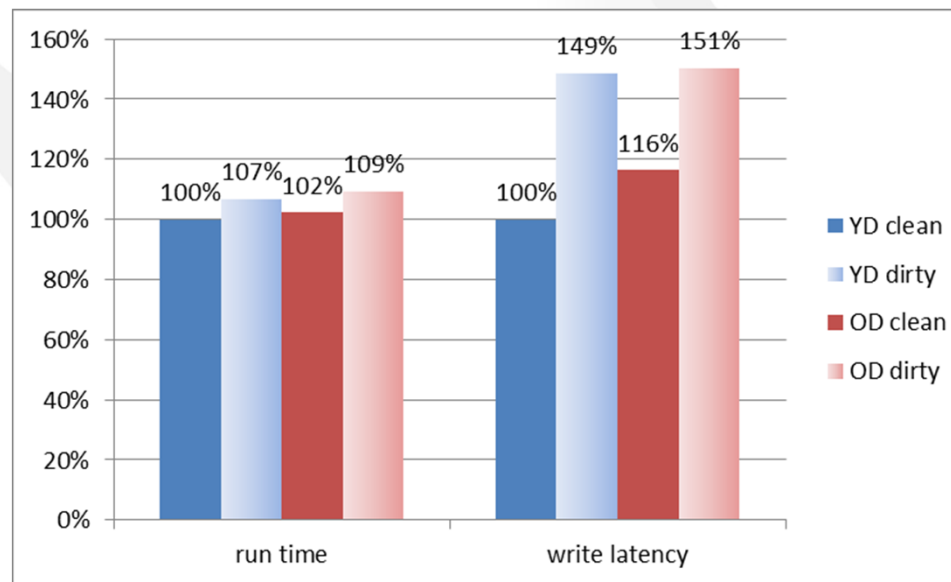
- Showing storage's impact on UX
- App-based bench: not synthetic workload
 - web, contacts, install, camera, gallery so on
- full test-case
 - Using camera and install-app, make file system to be full-status (dirty-status)
- Including FS and app's behavior
- Dirty status is more important in terms of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



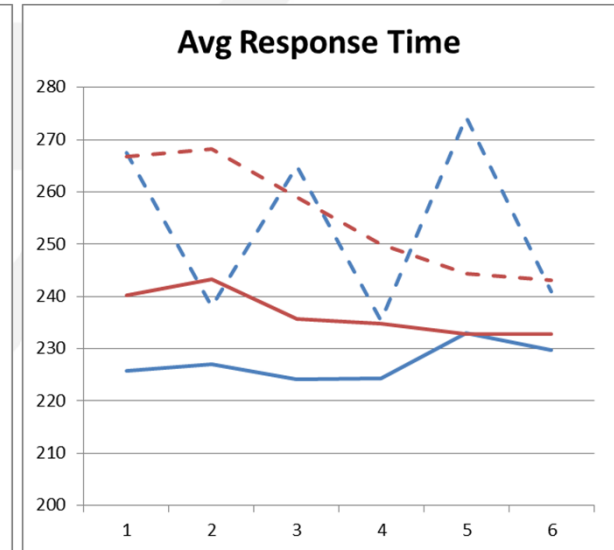
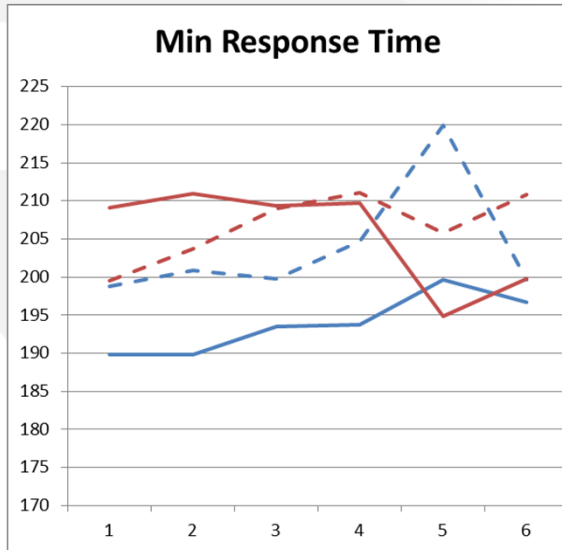
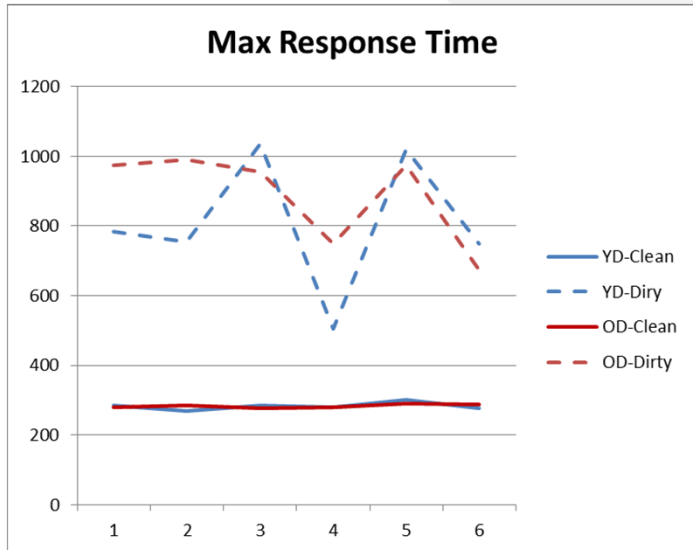
- Main-case results



Relative Performance based on Young-device's clean status

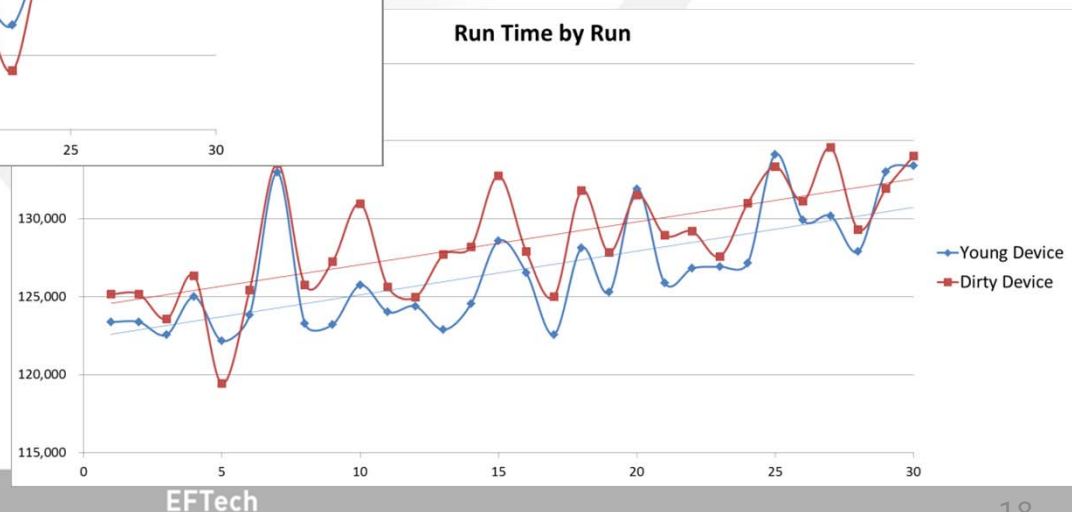
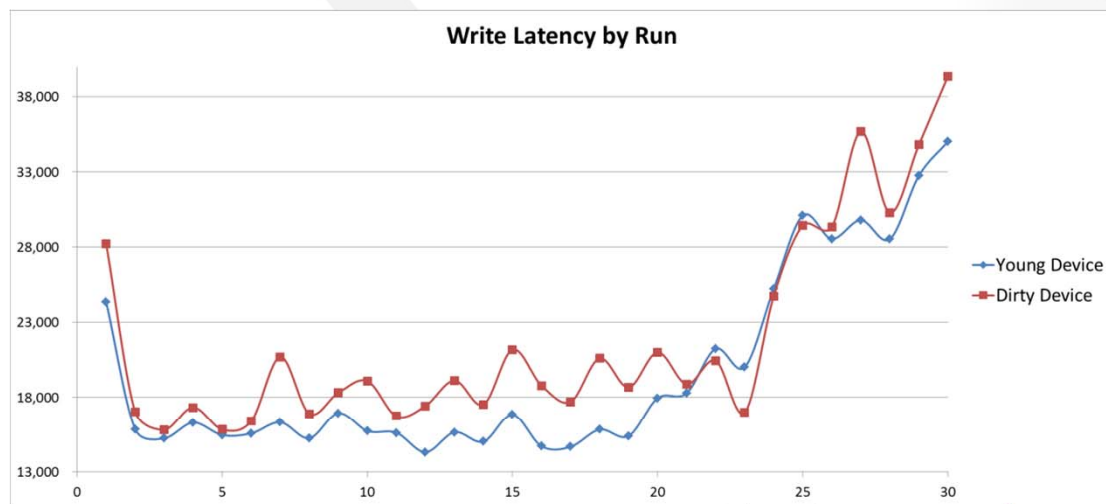
Ebench : Response time

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



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



16:29

Report

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

Statistic

History

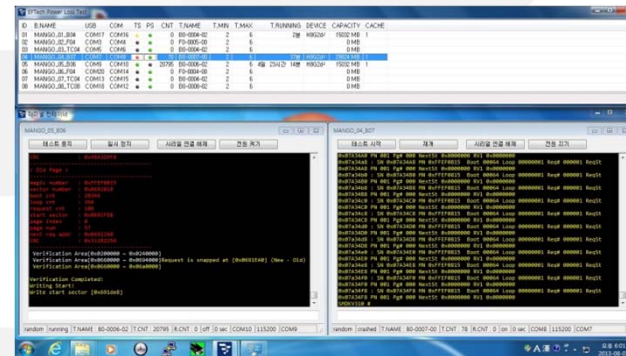
Back to Home



EF Storage Tester



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



- 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

andrew@elixirflash.com

Elixir Flash Technology

Proposal by . . .

