Flash & File System Co-Design

Thomas W. McCormick
Chief Engineer/Technologist

swissbit®

Santa Clara, CA
August 2018

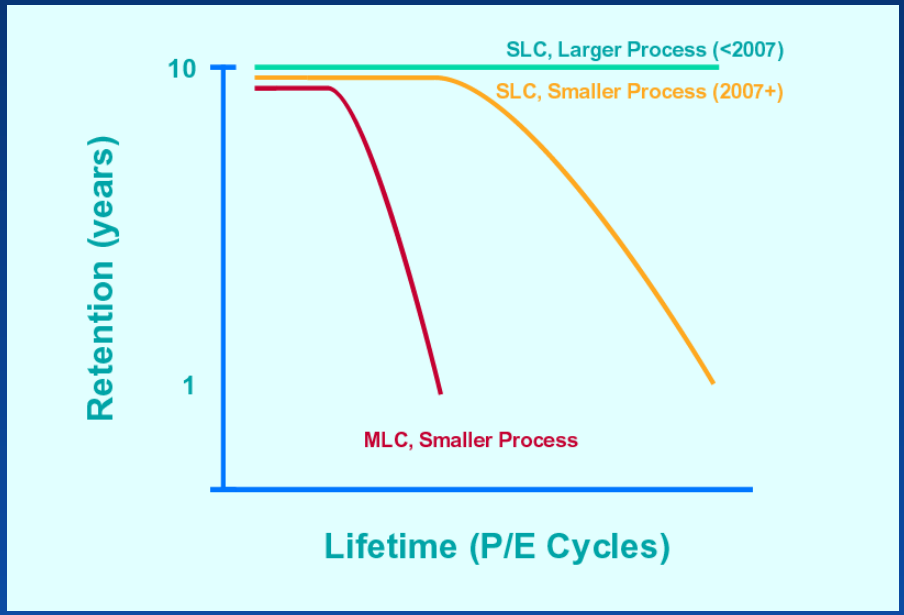
Background - Embedded Systems

- "Fixed function system"
  - Telecom, automotive, industrial control systems, medical equipment ...
- Commonality: Flash Storage
  - Code & data

[Wikipedia – Creative Commons]

Motivation

- Smaller process, more bits per cells
  - Endurance & ECC
  - Retention
- Challenge: Maintain acceptable service-life for embedded systems



Lifetime & Write Amplification

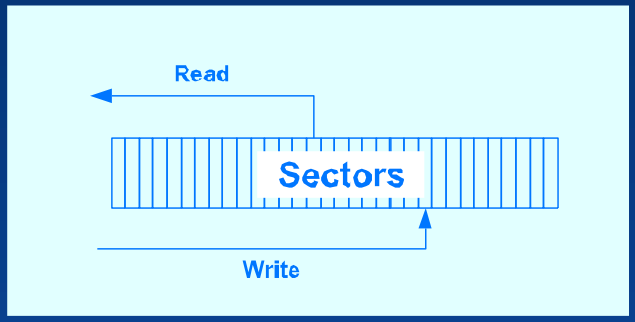
$$\text{Lifetime} = \frac{(\text{Capacity})(\text{Endurance})}{(\text{Data Rate})(\text{Write Amplification})}$$

$$\text{Write Amplification} = \frac{\text{Data Written to Flash}}{\text{Data Written by Host}}$$

- Write Amplification Factor (WAF) is a coupled function of the Flash Translation Layer (FTL) and nature of workload

Sector-Based "Contract"

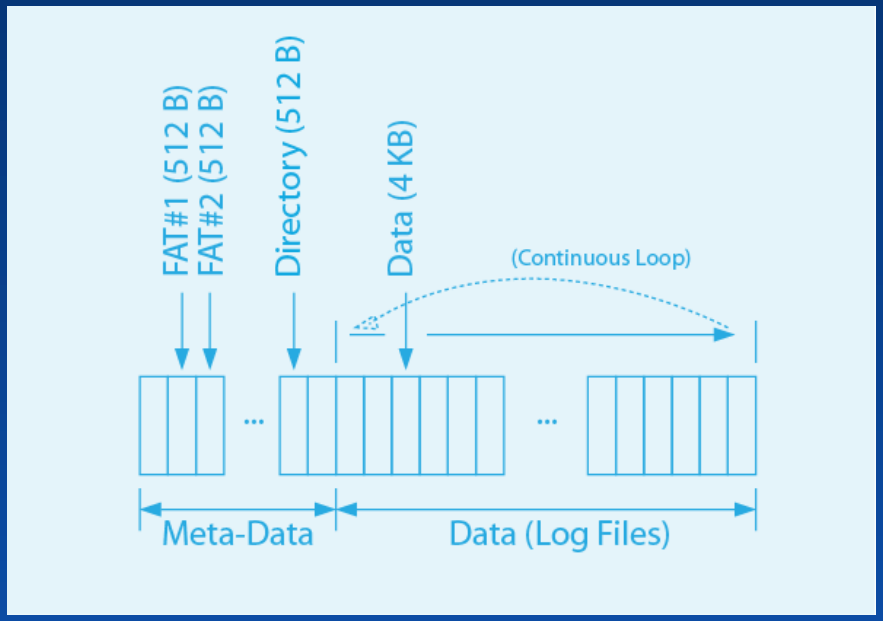
- Read/Write field of addressable sectors



- Trend: Shift FTL details to Host
  - Not backwards compatible. Not for Embedded Apps

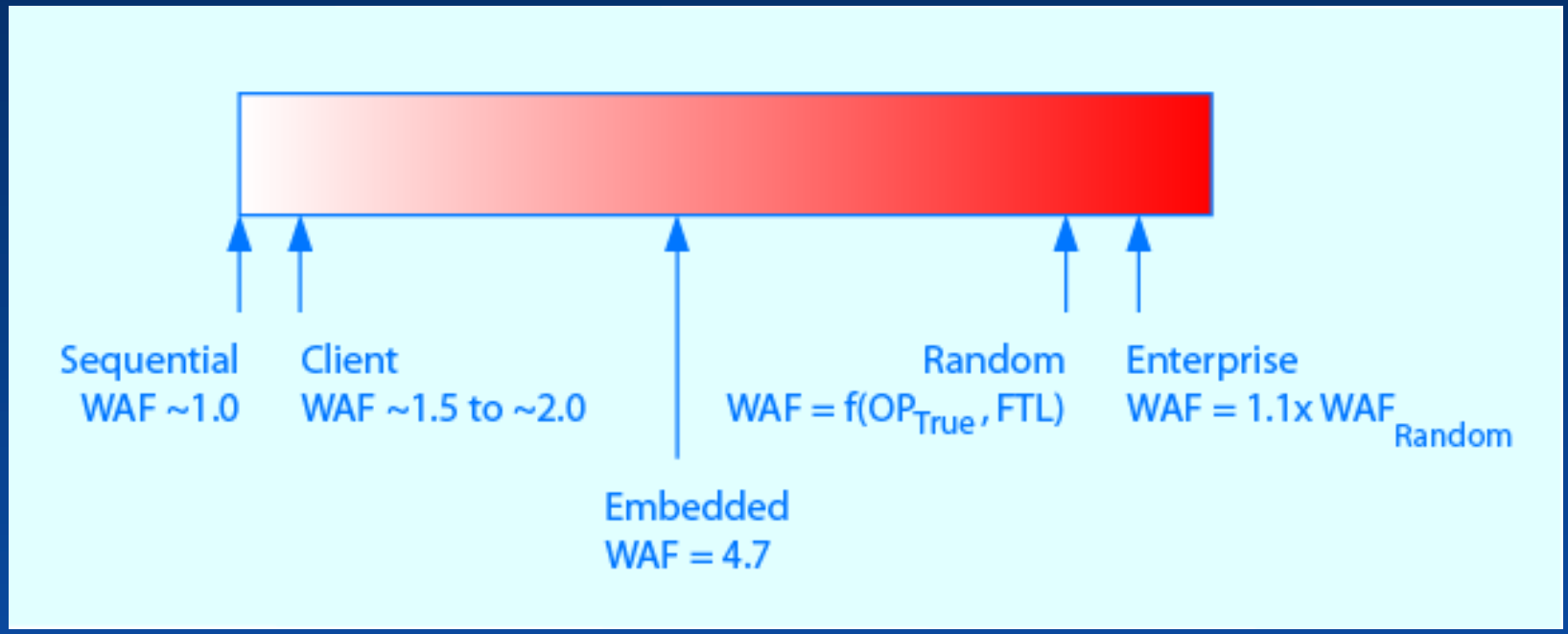
$WAF_{Embedded}$ Workload Model

- Embedded Workload: Sequential & "Repeated"



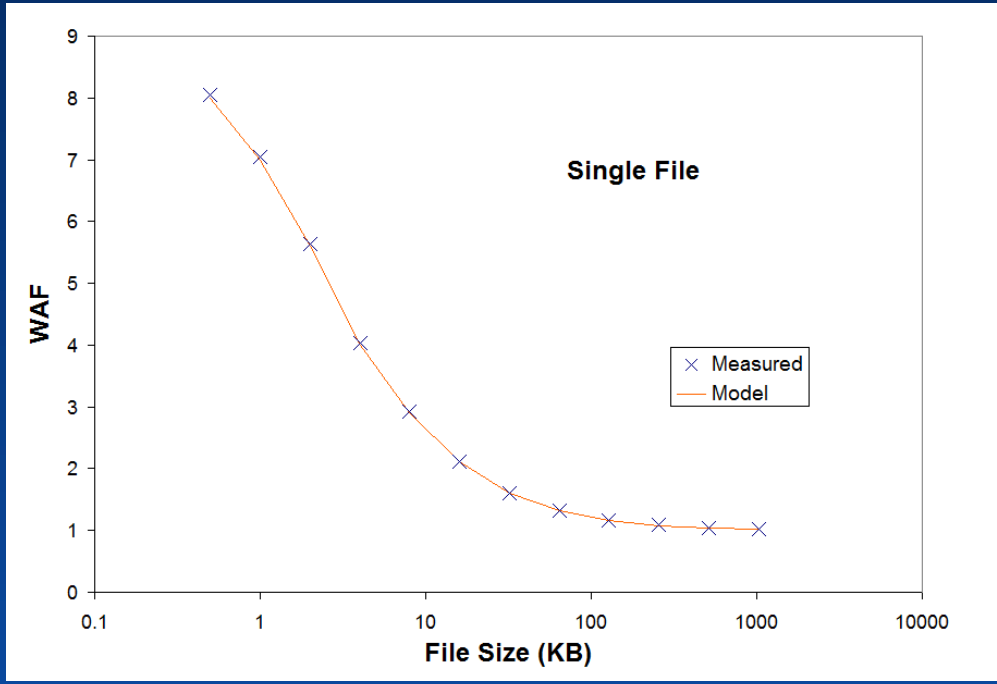
[McCormick FMS2017]

WAF Measurements - Workload



[McCormick FMS2017]

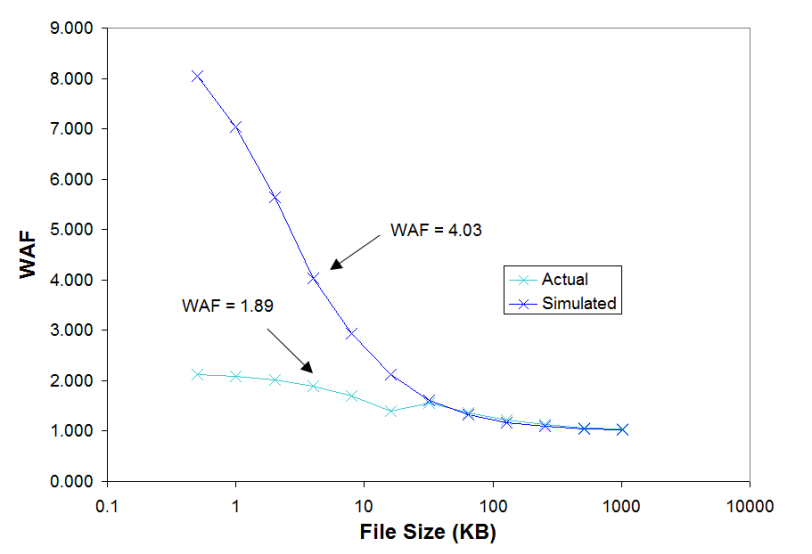
$WAF_{Embedded}$ Measurements & Modeling



[McCormick FMS2017]

$WAF_{Embedded}$ FAT File System

Metadata: 6 sectors (simulated) vs 60 sectors (actual)



Conclusion: WAF alone isn't complete

WAF_{File}

- Proposal: WAF_{File} to consider both WAF and file system design

$$WAF = \frac{\text{Data, Flash}}{\text{Data, Host}}$$

$$WAF = \frac{\text{File Data, Flash + Metadata, Flash}}{\text{File Data, Host + Metadata, Host}}$$

$$WAF_{File} = \frac{\text{File Data, Flash + Metadata, Flash}}{\text{File Data, Host}}$$

WAF_{File} - Ideal

$$WAF_{File} = \frac{\text{File Data, Flash + Metadata, Flash}}{\text{File Data, Host}}$$

$$WAF_{File\ (FAT)} = \frac{\text{File Data, Flash + (3 * 4 KB)}}{\text{File Data, Host}}$$

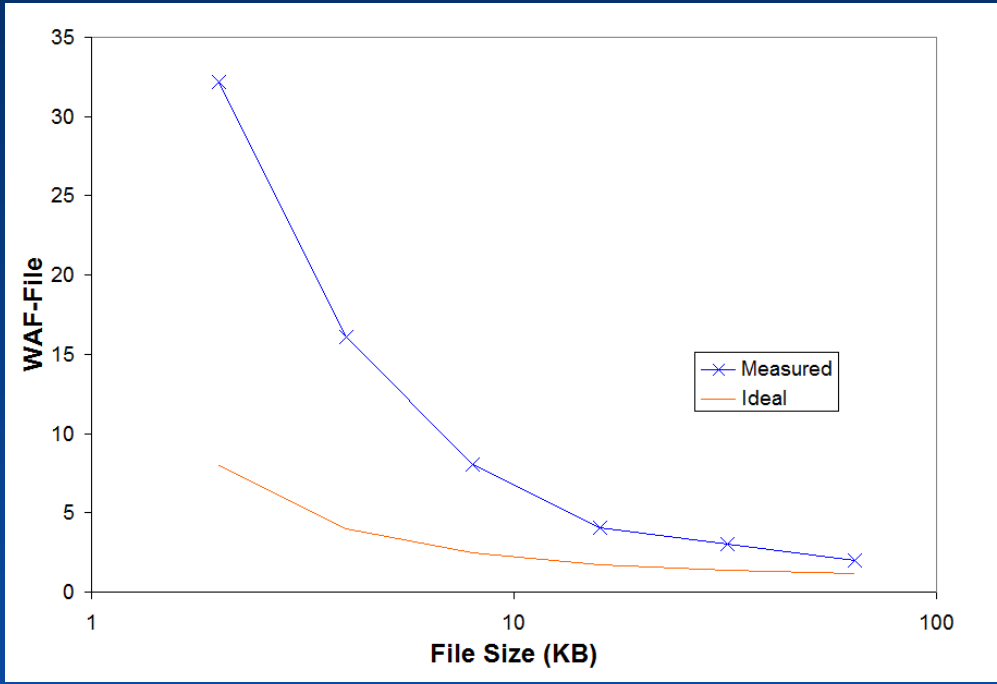
$$WAF_{File\ (ext)} = \frac{\text{File Data, Flash + (2 * 4 KB)}}{\text{File Data, Host}}$$

where,

$$\text{File Data, Flash} = \begin{cases} \text{4 KB, File Data when Host < 4 KB} \\ \text{File Data, Host when File Data Host >= 4 KB} \end{cases}$$

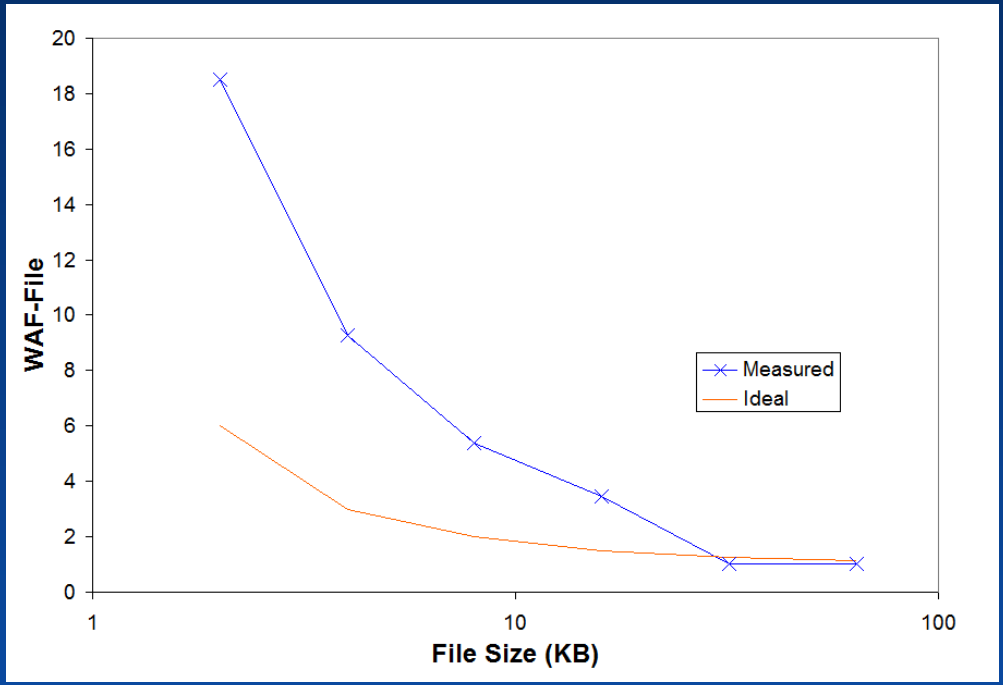

Flash Memory
SUMMIT

WAF_{File} - FAT


35
30
25
20
15
10
5
0
WAF-File
1
10
100
File Size (KB)
Measured
Ideal

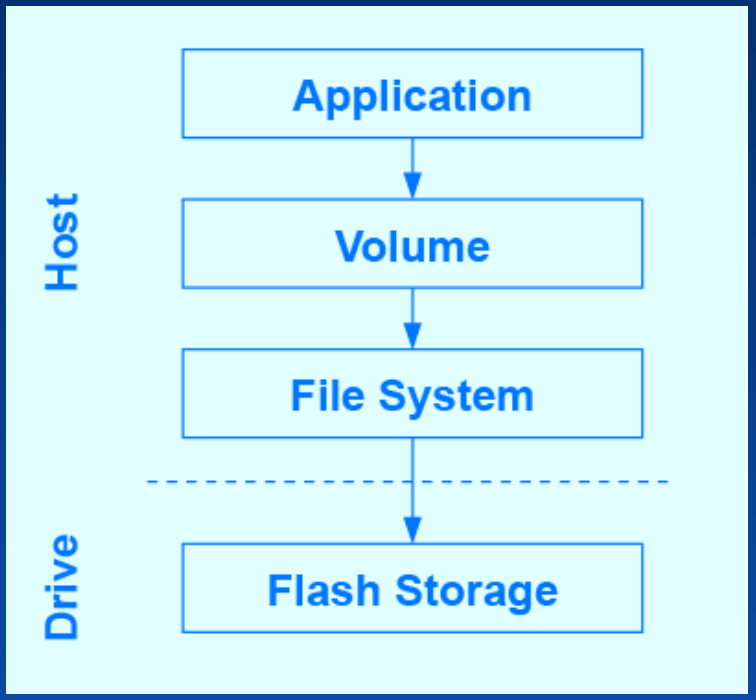

Flash Memory
SUMMIT

WAF_{File} - ext


20
18
16
14
12
10
8
6
4
2
0
WAF-File
1
10
100
File Size (KB)
Measured
Ideal

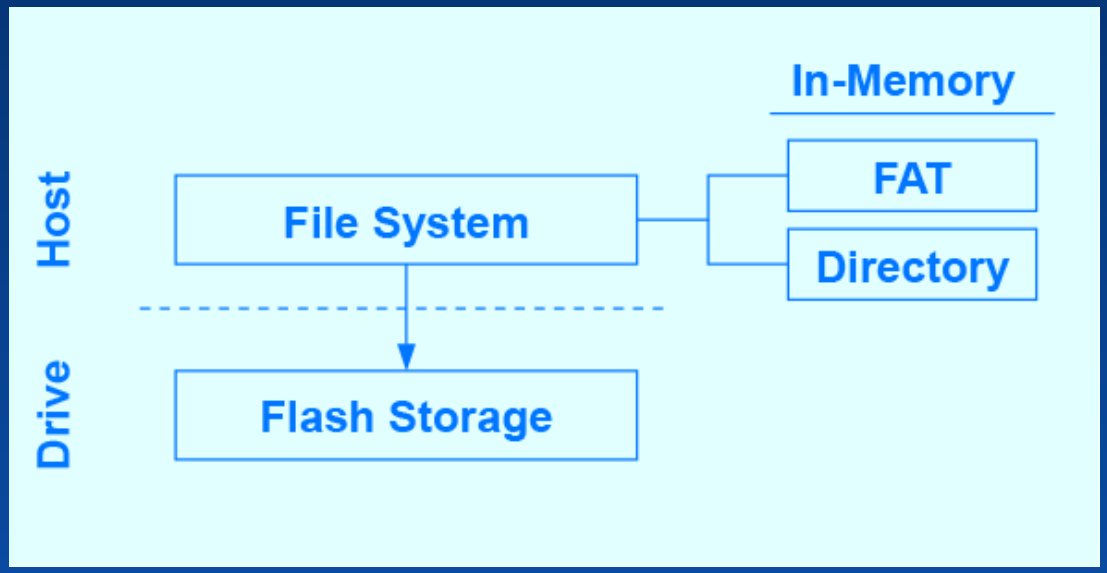
How to Improve?

- Control flushing



How to Improve?

- Cache file system metadata



File System Aware FTL

$$WAF_{File\ (FAT)} = \frac{File\ Data,\ Flash + (3 * 4\ KB)}{File\ Data,\ Host}$$

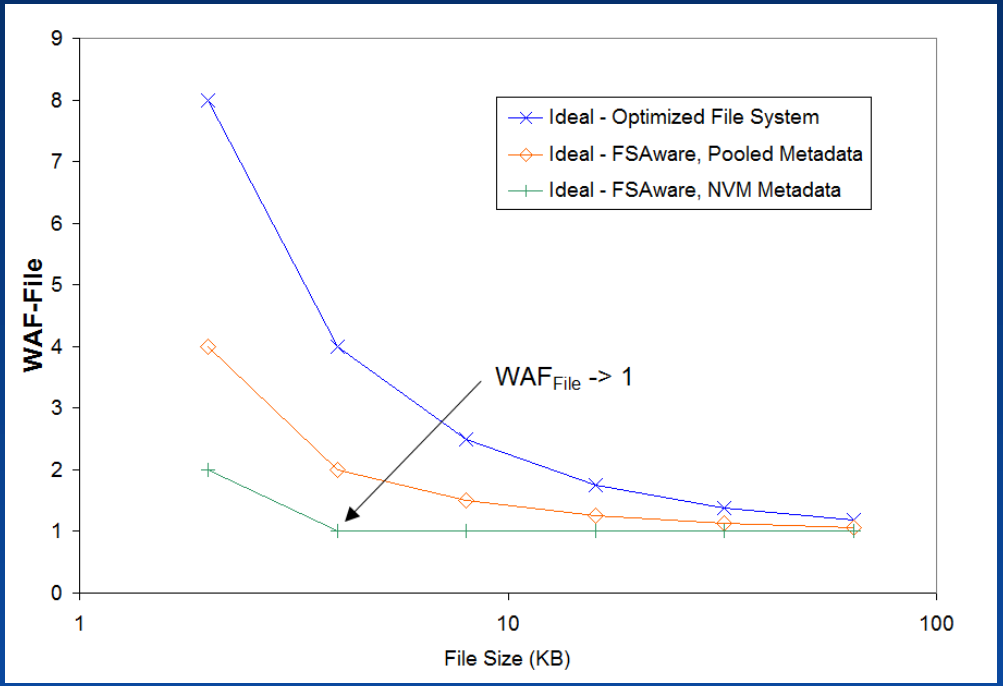
Pool Metadata:

$$WAF_{File\ (FAT)} = \frac{File\ Data,\ Flash + (4\ KB)}{File\ Data,\ Host}$$

Metadata, NVM:

$$WAF_{File\ (FAT)} = \frac{File\ Data,\ Flash}{File\ Data,\ Host}$$

File System Aware FTL



Summary & Conclusions

System Design

- Use WAF_{File} for efficient file system design
- Limit flush
- Cache file system metadata

Drive Design

- File System Aware FTL
  - Pool Metadata
  - NVM Metadata

Conclusion: Co-Design enables WAF_{File} -> 1.0

More Embedded Sessions

- Flash Memory System Embedded Events:
  - Embedded Applications, Part 1 (101-B)
    - Tues 8:30 – 9:35 AM
  - Embedded Applications, Part 2 (102-B)
    - Tues 9:45 – 10:50 AM
  - Beer, Pizza, and Chat with the Experts
    - Tues 7:00 – 8:30 PM
  - Flash and the IoT (302-B)
    - Thurs 3:40 – 5:00 PM

Questions?

Tom McCormick - Chief Engineer/Technologist
Swissbit
tom.mccormick@swissbit.com