Improving Lifetime Estimates for Embedded Flash Memory Systems

Thomas W. McCormick

Chief Engineer/Technologist

swissbit®

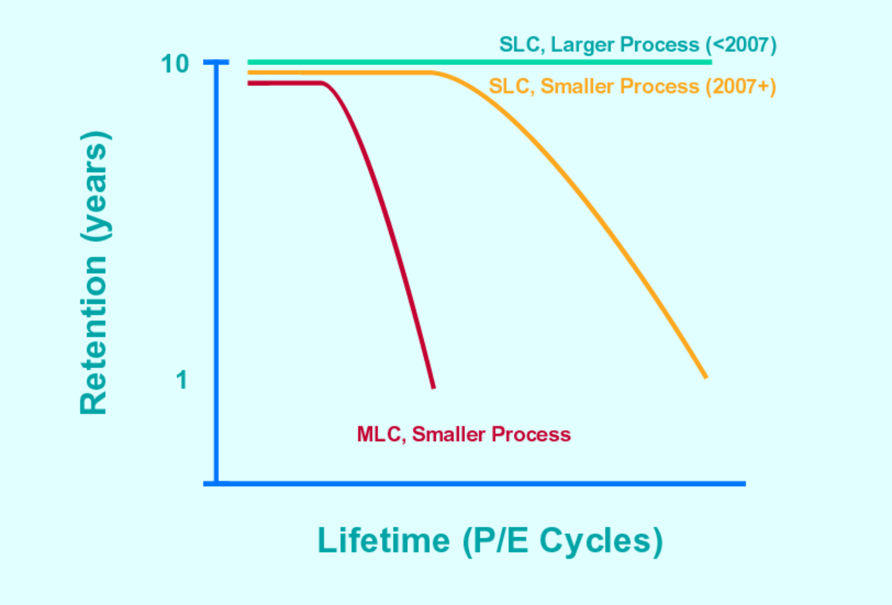
Background - Embedded Systems

- "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



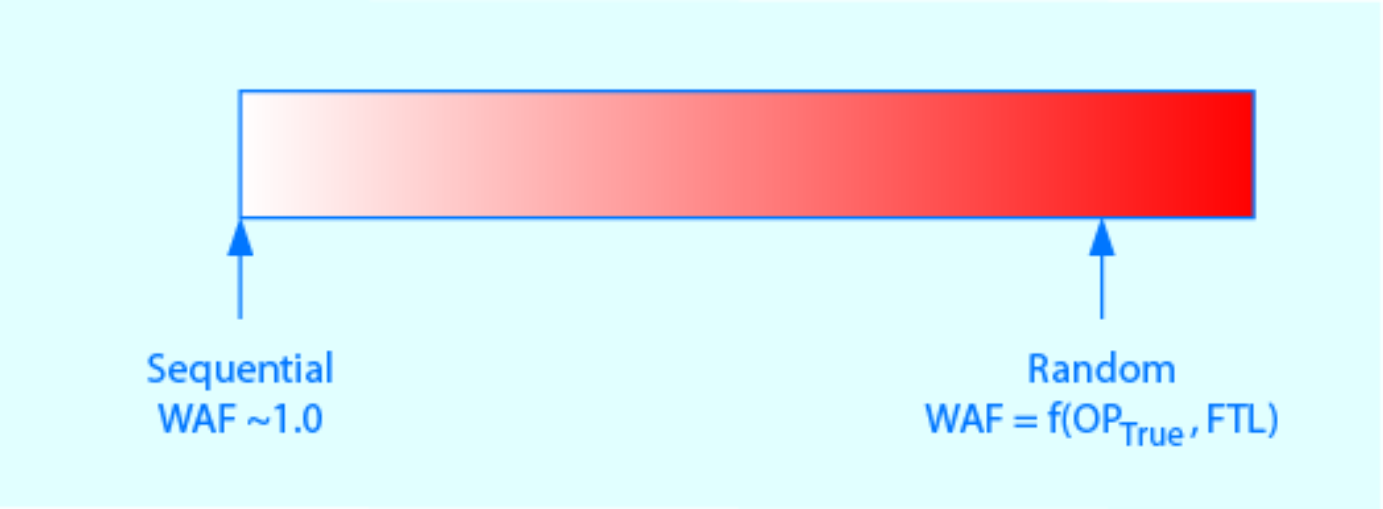
Lifetime & Write Amplification

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

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

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

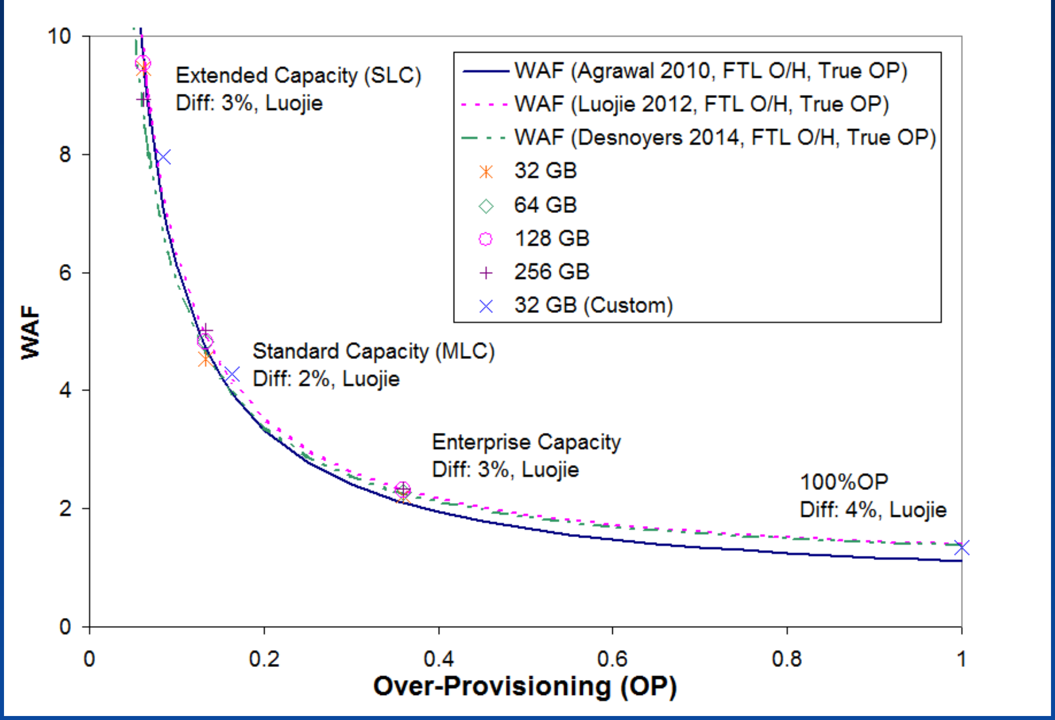
WAF Measurements - Workload



Sequential
WAF ~1.0

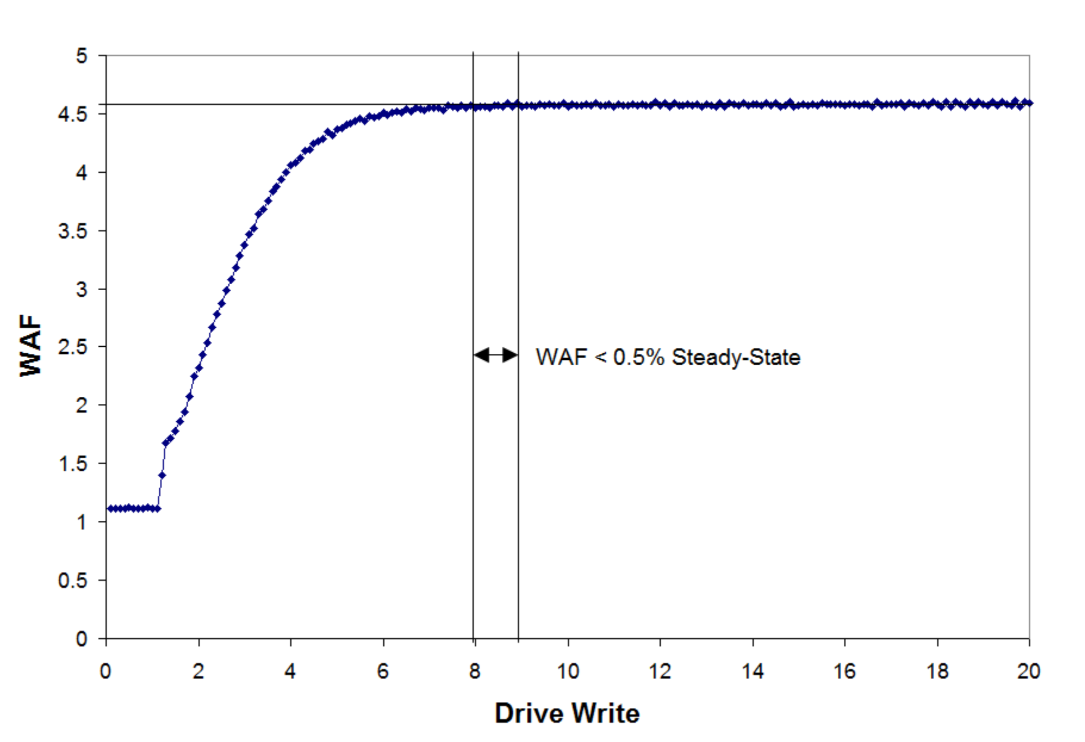
Random
WAF = $f(OP_{True}, FTL)$

WAF – Analytic & Measurements



[McCormick FMS2016]

WAF Measurements – Steady-State



WAF Measurements - Workload

- Sequential
  - WAF ~1.0
- Random
  - WAF = f(OP_{True}, FTL)
- Enterprise & Client (JEDEC)
- Embedded?

Enterprise Workload

JEDEC 219A Workload:

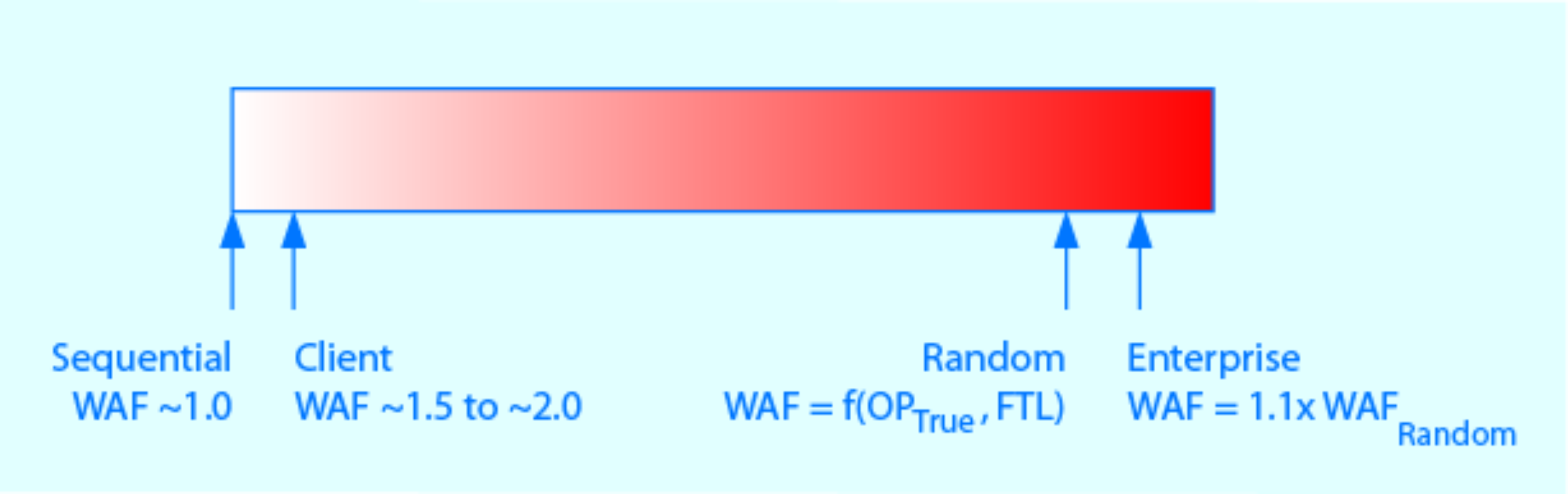
- 512 bytes (0.5K) 4%
- 1024 bytes (1K) 1%
- 1536 bytes (1.5K) 1%
- 2048 bytes (2K) 1%
- 2560 bytes (2.5K) 1%
- 3072 bytes (3K) 1%
- 3584 bytes (3.5K) 1%
- 4096 bytes (4K) 67%
- 8192 bytes (8K) 10%
- 16,384 bytes (16K) 7%
- 32,768 bytes (32K) 3%
- 65,536 bytes (64K) 3%

$$\mathrm{WAF}_{\mathrm{Enterprise}} = f(\mathrm{WAF}_{\mathrm{Random}})$$

Configuration	Enterprise WAF	Random WAF	Scale
Extended (SLC)	11.39	9.94	1.15
Standard (MLC)	5.33	4.80	1.11
Enterprise	2.59	2.28	1.13

$$\mathrm{WAF}_{\mathrm{Enterprise}} = 1.1 \text{ x } \mathrm{WAF}_{\mathrm{Random}}$$

WAF Measurements - Workload



Sequential
WAF ~1.0

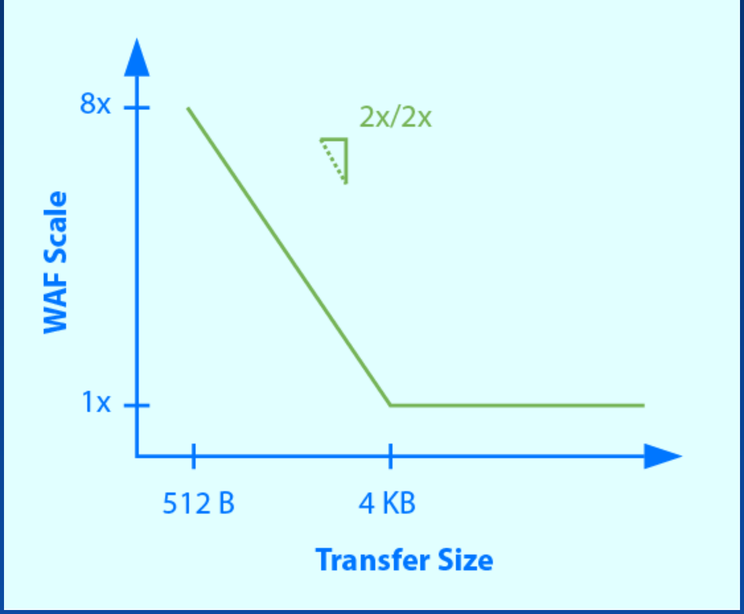
Client
WAF ~1.5 to ~2.0

Random
WAF = $f(OP_{True}, FTL)$

Enterprise
WAF = $1.1x\ WAF_{Random}$

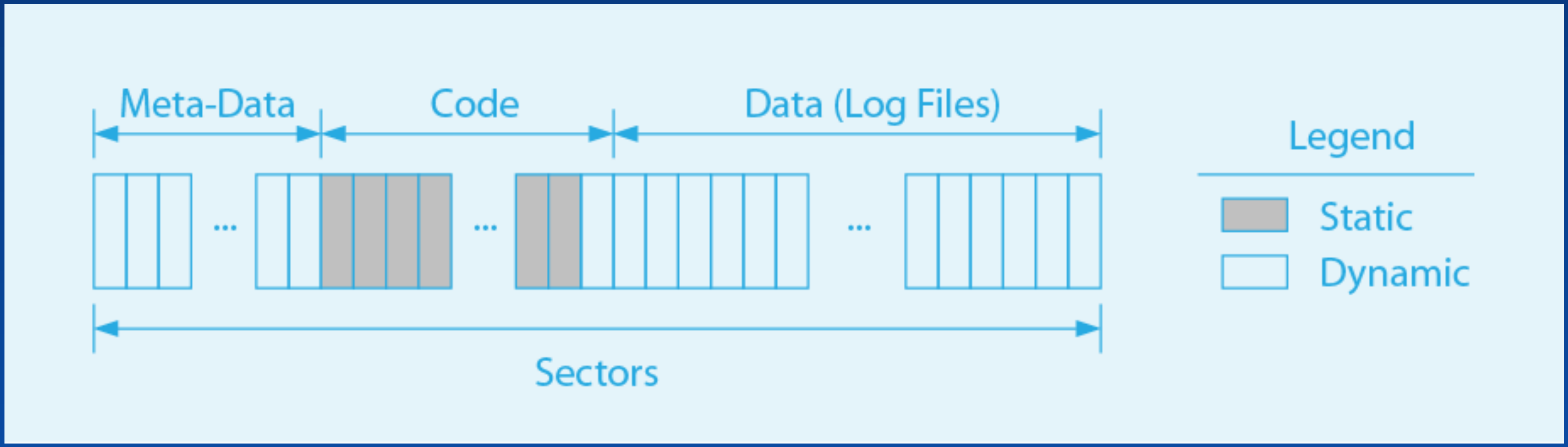
WAF Measurements – Write Size

- WAF is for random writes to Flash Allocation Unit (4 KB)



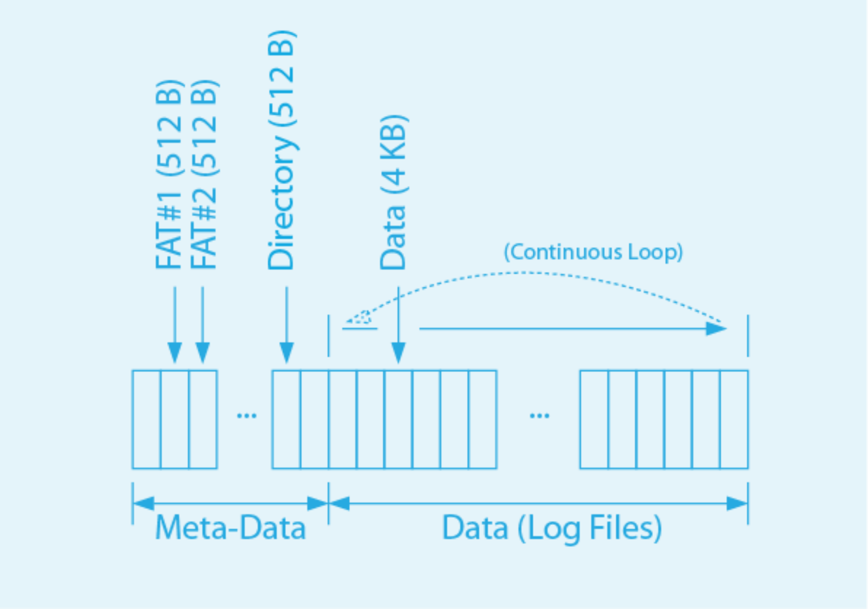
$WAF_{Embedded}$ Workload

- Embedded Workload
  - File System: Code and Logs



$\text{WAF}_{\text{Embedded}}$ Workload Model

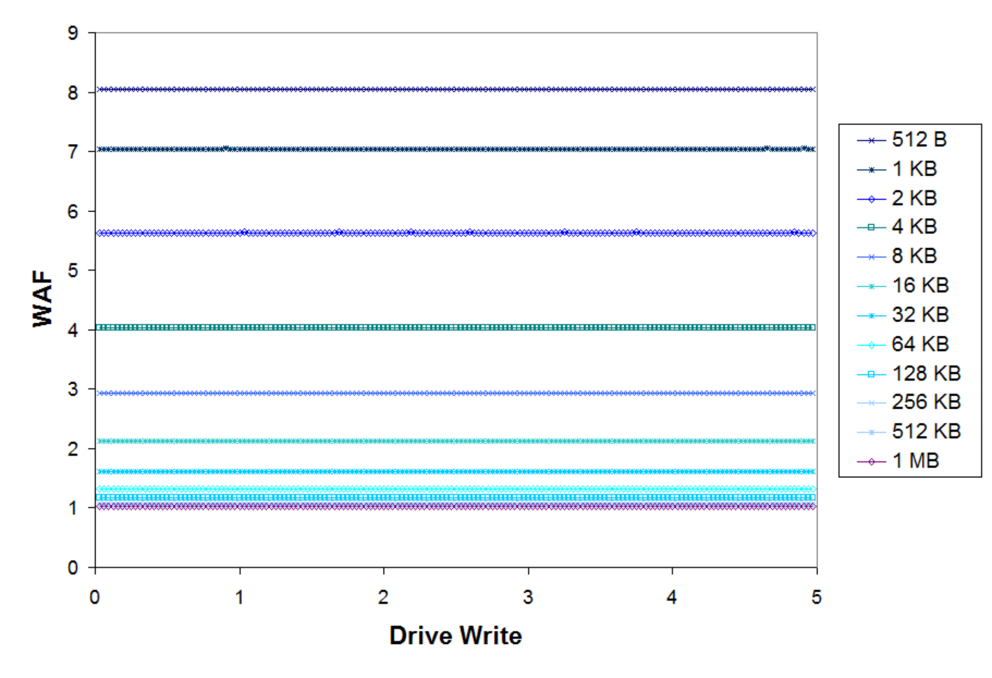
- Embedded Workload: Sequential & “Repeated”



WAF$_{\text{Embedded}}$ FAT File System

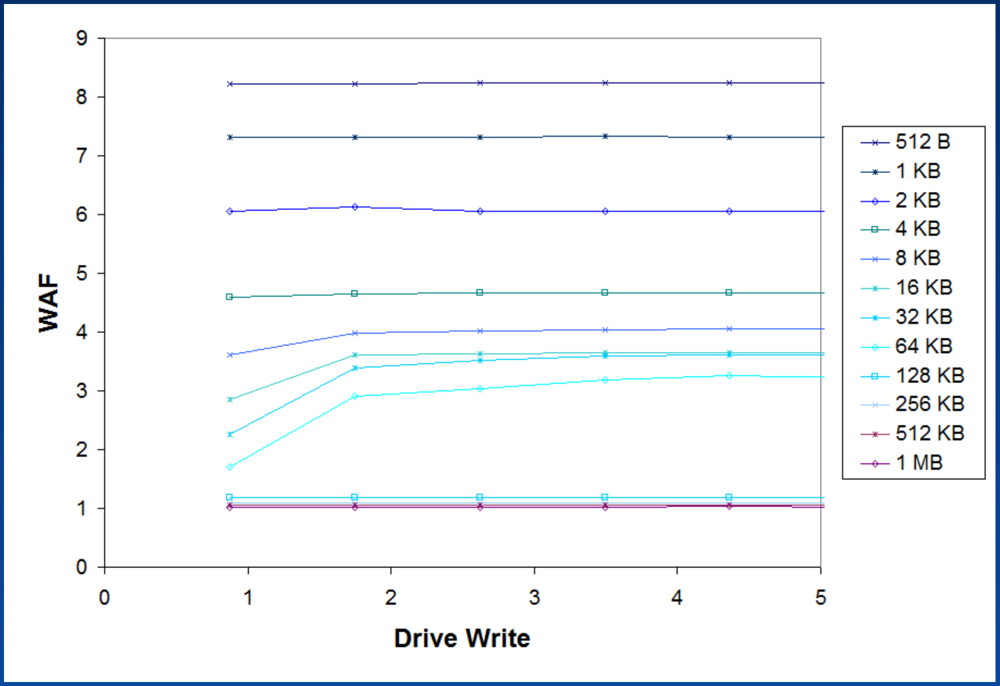
- FAT File System
  - FAT #1, FAT #2, Directory metadata sectors
  - Expect 6 sectors of metadata per file create/delete
  - Measure ***60 sectors*** of metadata per file create/delete (Windows 7)
- Tests with simulated file system (ideal)

WAF$_{\text{Embedded}}$ Measurements (Single)



WAF for file system operations (single) is only first-order effects

WAF$_{\text{Embedded}}$ Measurements (Fill)



WAF for file system operations (fill) is only first-order effects (mostly)

WAF$_{\text{Embedded}}$ Analytic

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

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

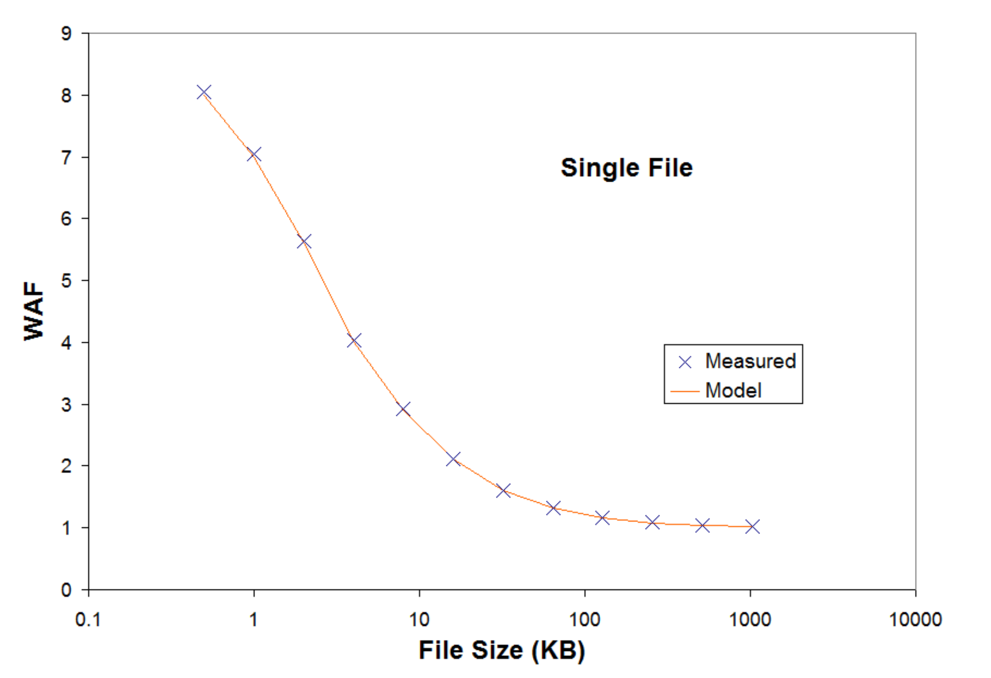
$$\text{WAF} = \frac{\text{File Data, Flash} + (3 * 4\ \text{KB}) + (3 * 4\ \text{KB})}{\text{File Data, Host} + (3 * 0.5\ \text{KB}) + (3 * 0.5\ \text{KB})}$$

$$\text{WAF} = \frac{\text{File Data, Flash} + 24\ \text{KB}}{\text{File Data, Host} + 3\ \text{KB}}$$

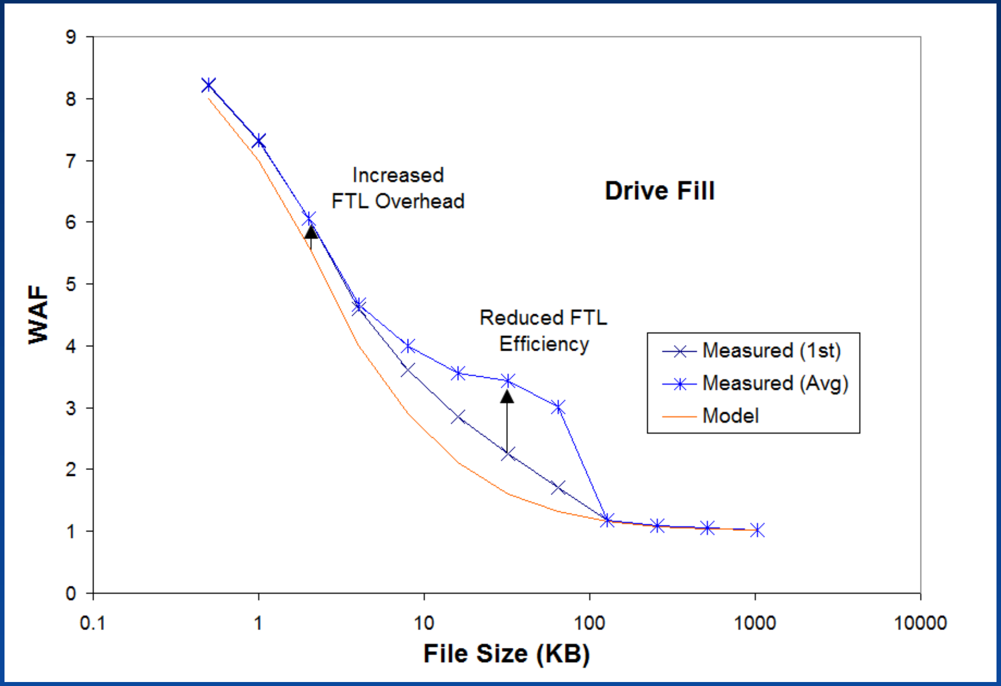
where,

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

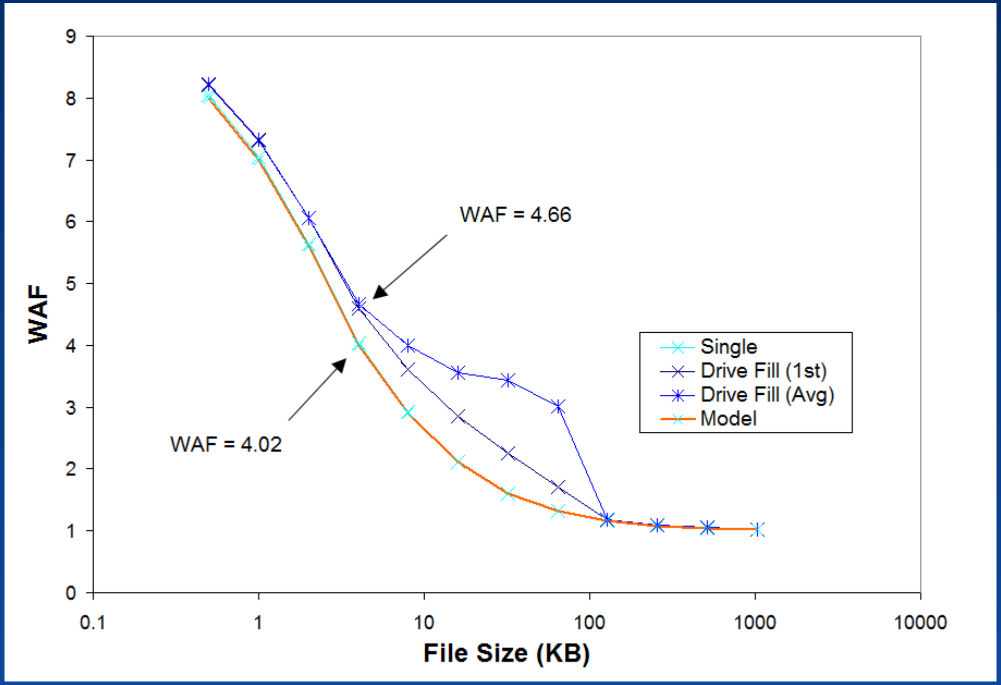
$\text{WAF}_{\text{Embedded}}$ Measurements & Modeling



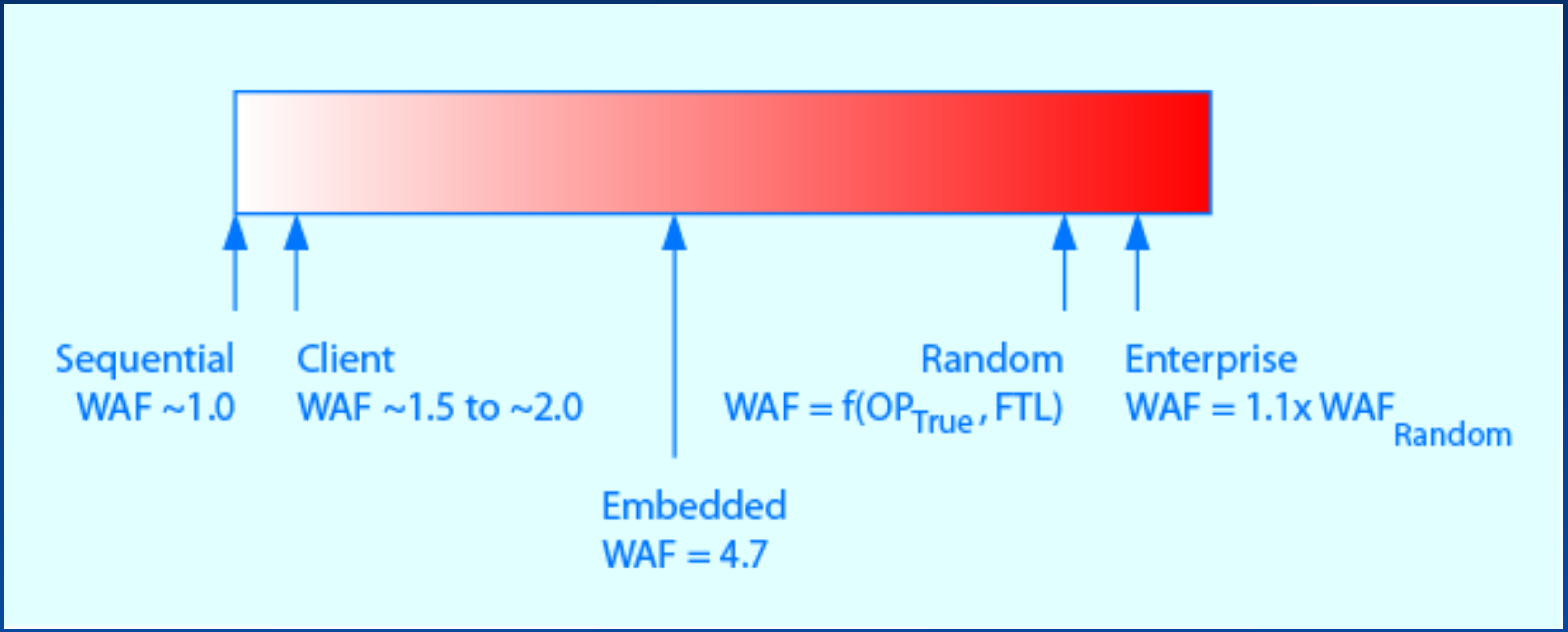
WAF$_{\text{Embedded}}$ Measurements & Modeling



WAF$_{\text{Embedded}}$ Measurements & Modeling



WAF Measurements - Workload



Sequential
WAF ~1.0

Client
WAF ~1.5 to ~2.0

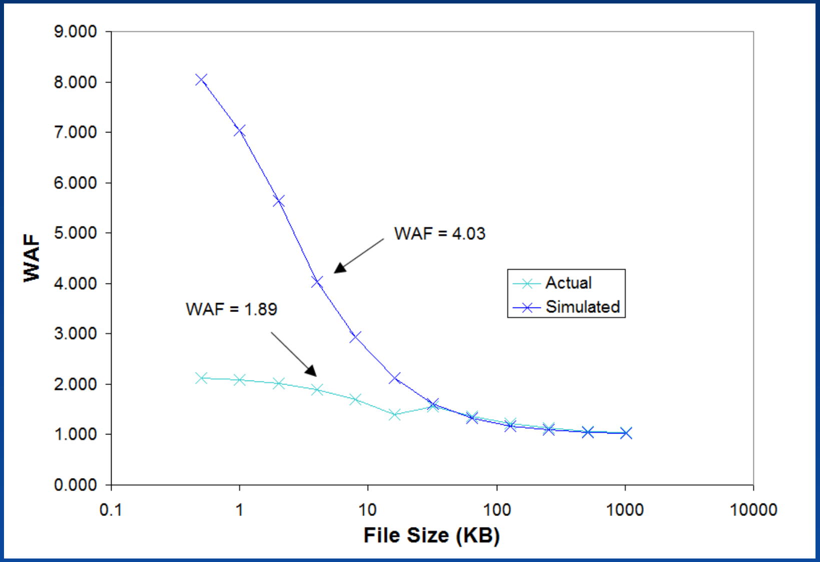
Embedded
WAF = 4.7

Random
$\text{WAF} = f(\text{OP}_{\text{True}}, \text{FTL})$

Enterprise
$\text{WAF} = 1.1\text{x}\ \text{WAF}_{\text{Random}}$

$\text{WAF}_{\text{Embedded}}$ FAT File System

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



Conclusion: WAF alone isn’t complete

WAF_{File}

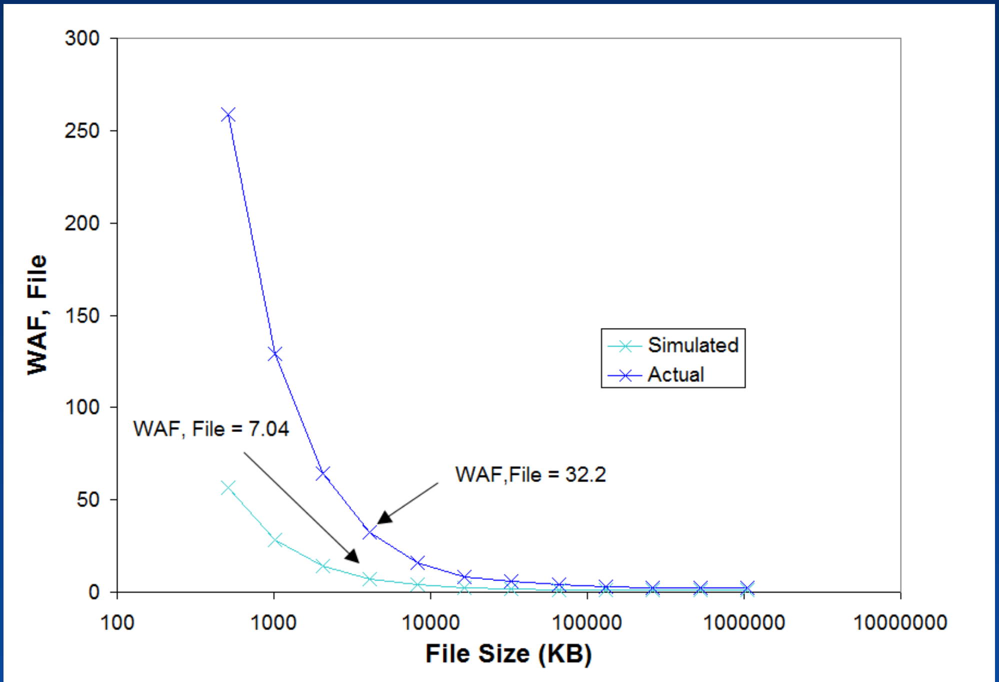
- Proposal: WAF_{File} to consider both WAF and file system design

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

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

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

WAF$_{\text{File, Embedded}}$ FAT File System



Summary & Conclusions (1)

- $\text{WAF}_{\text{Embedded}}$ Model: File Create/Delete
- $\text{WAF}_{\text{Embedded}}$ first-order effects
- $\text{WAF}_{\text{Embedded}} \neq f(\text{OP})$
- Lifetime of file is lesser effect
  - $\text{WAF}_{\text{Embedded, Single}} \approx \text{WAF}_{\text{Embedded, Fill}}$
- $\text{WAF}_{\text{Embedded, 4 KB}} = 4.7$

Summary & Conclusions (2)

- Design Suggestions
  - Conservative commitment (avoid flush)
  - Consider both WAF and $\mathrm{WAF}_{\mathrm{File}}$
    - – WAF alone hides poor file system design

More Embedded Sessions

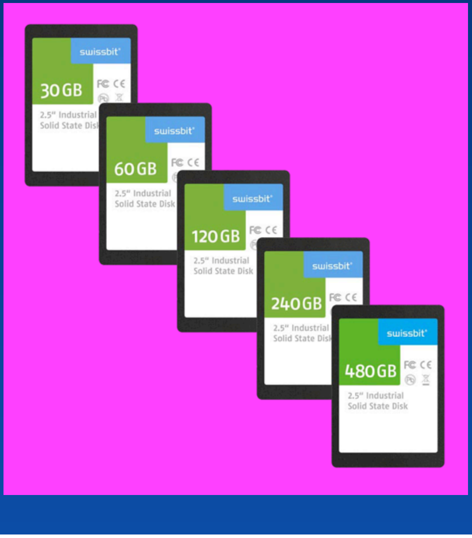
- Flash Memory System Embedded Events:
  - Embedded Applications, Part 1 (101-B)
    - Tues 8:30 – 9:35 AM
  - Embedded Applications, Part 2 (102-B)
    - Tues 9:45 – 10:50 AM
  - Beer, Pizza, and Chat with the Experts
    - Tues 7:00 – 8:30 PM

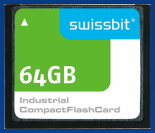
Questions?

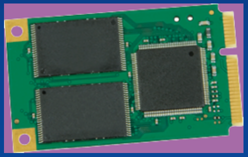
Tom McCormick - Chief Engineer/Technologist

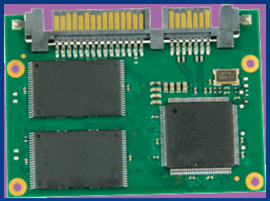
Swissbit

tom.mccormick@swissbit.com









swissbit®