Large-Scale
Enterprise Flash Storage Reliability

It's Not Just About The Chips!

Erik Eyberg, Sr. Analyst

Texas Memory Systems, Inc.

Disclaimers

- IBM recently announced their intent to acquire TMS (exciting!)
- But we are completely separate companies until the transaction closes
- This presentation is <u>not</u> intended to convey product plans, strategic directions, or any forward-looking statements for TMS or IBM
- It is just industry commentary based on a variety of sources

Agenda

- Failure and Reliability Fundamentals
- Metrics
- Example Reliable Flash System Design

Defining “Failure”

- Enterprise Flash storage systems include a variety of components, typically at least:
  - **Data path infrastructure:** drives, modules, RAID controllers, external interfaces
  - **Management infrastructure:** functional units for provisioning, monitoring, maintenance
  - **Environmental infrastructure:** power, cooling
- **“Failures” that matter most: *failures that impact the user*** *(especially with data loss)*

Performance “Failures”?

- “Write cliff” phenomenon
- Performance drops as Flash systems age
- *Is diminished performance a failure?*

It depends!

Main factor: most customers don’t push systems to their absolute limit.

Possible Storage Stack Failures

- At the Flash chip level: inability to reliably read and/or write data
- At the storage module/“SSD” level: problems with enough Flash chips or infrastructure (controllers, etc) to make modules inoperable or degraded
- At the system level: problems with enough modules to make system inoperable or degraded

Current Failure Mitigations

- Chip level: error correcting codes and other advanced techniques to boost the number of P/E cycles that can be sustained*
- Module level: adaptive Flash management plus RAID and/or sparing techniques across sets of chips
- System level: RAID and/or sparing techniques across sets of modules + other infrastructure (interfaces, power, etc)

* There are other important things to worry about besides P/E cycles (e.g. disturb errors) but the industry seems to focus on P/E cycles as the primary chip-level reliability metric.

User Perspective on Reliability

- All data storage systems eventually fail
- Two key questions:

1. **How long should you expect between failures?**
2. **How gracefully are failures handled?**

MTBF, MTTF, etc

For most datacenter Flash customers:

- MTBFs and MTTFs measured in hours are either not meaningful or misleading (*alone*)
- Flash chip reliability drops due to #P/E cycles before it drops due to #operating hours
- MTTFs measured in bytes (PBW metrics) are more relevant
- Time-based “full drive write” and “full write performance” metrics are probably better

“Full Write Performance” Metric

$$\frac{\text{Maximum number of P/E cycles}}{\text{Maximum write bandwidth}}$$

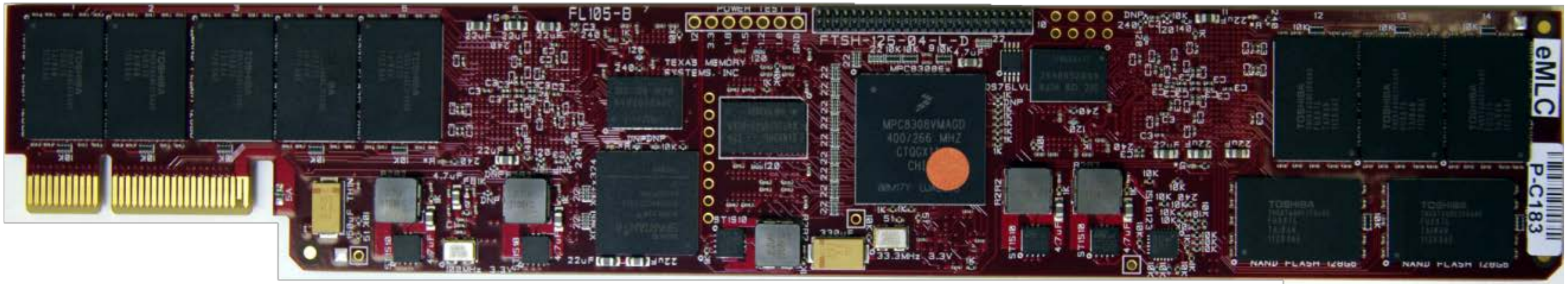
(in P/E cycles/time)

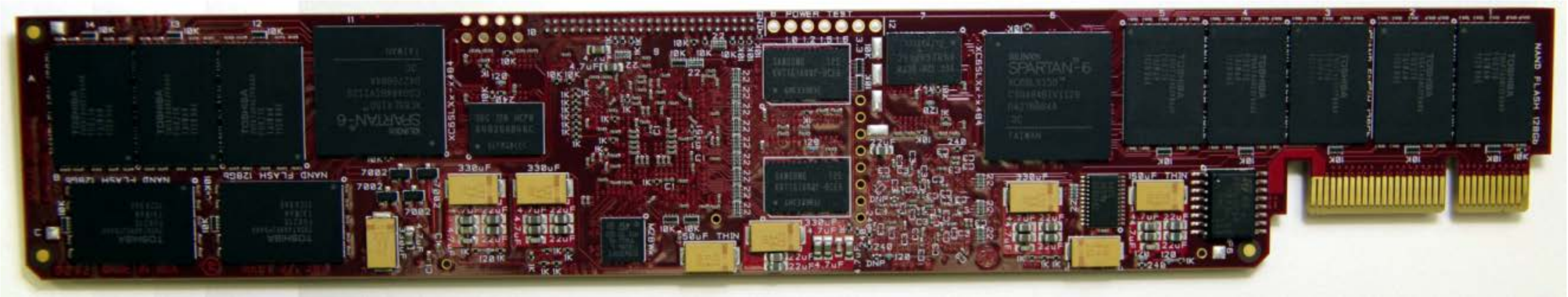
- Estimates amount of time system/module will be able to deliver full write performance
- Suggestions for better acronym than yFWP?

Out There: MTPI

- MTPI: Mean Time to Performance Impairment
- System-level metric that goes beyond yFWP
- Must be calculated @ specific performance
- RAID and sparing techniques coupled with internal performance governors should mean that MTPI for system > ∑ module MTPIs
- MTPI and MTBF provide fairly complete reliability picture for most customers

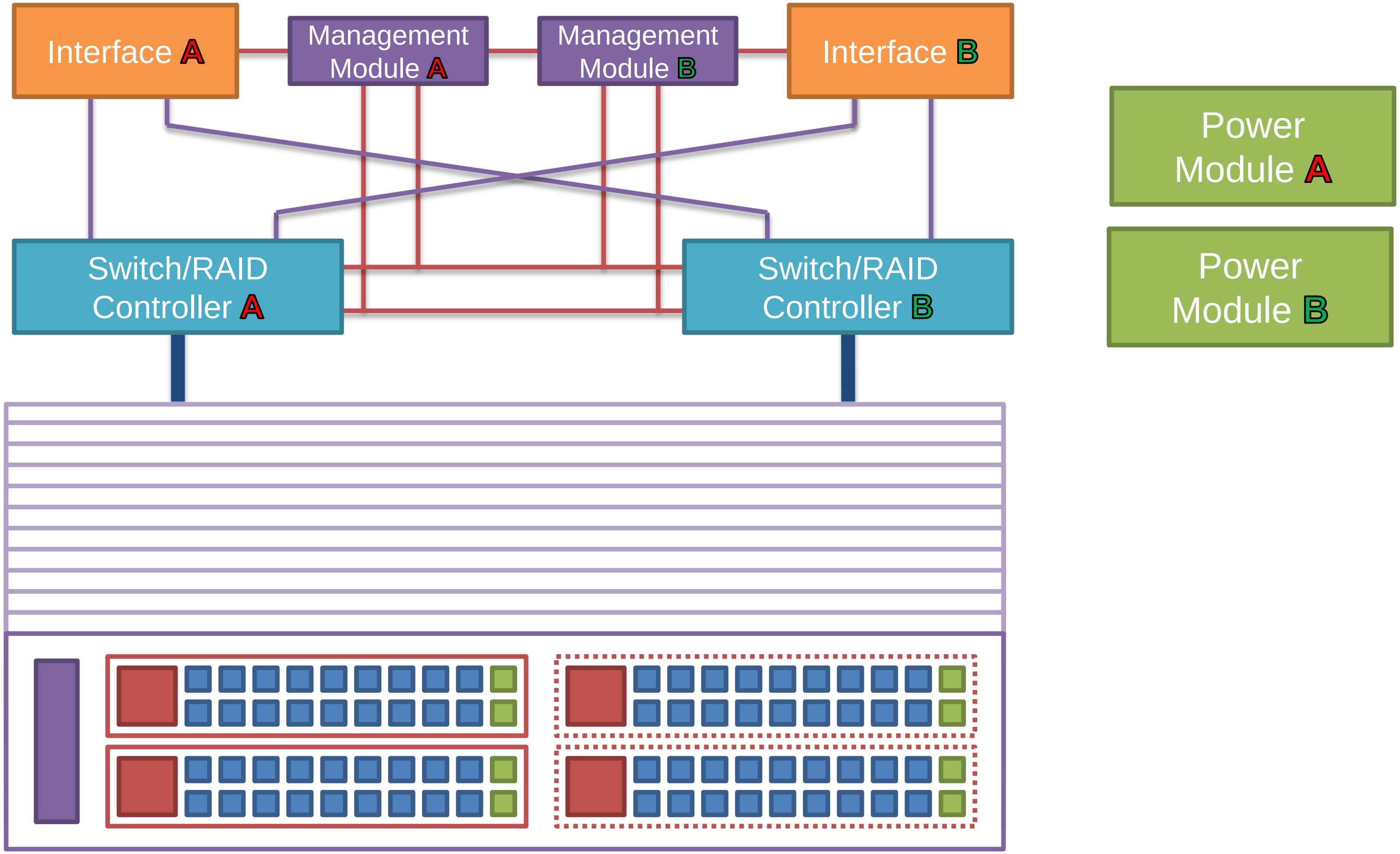
Example HA Flash System Design
Storage Modules



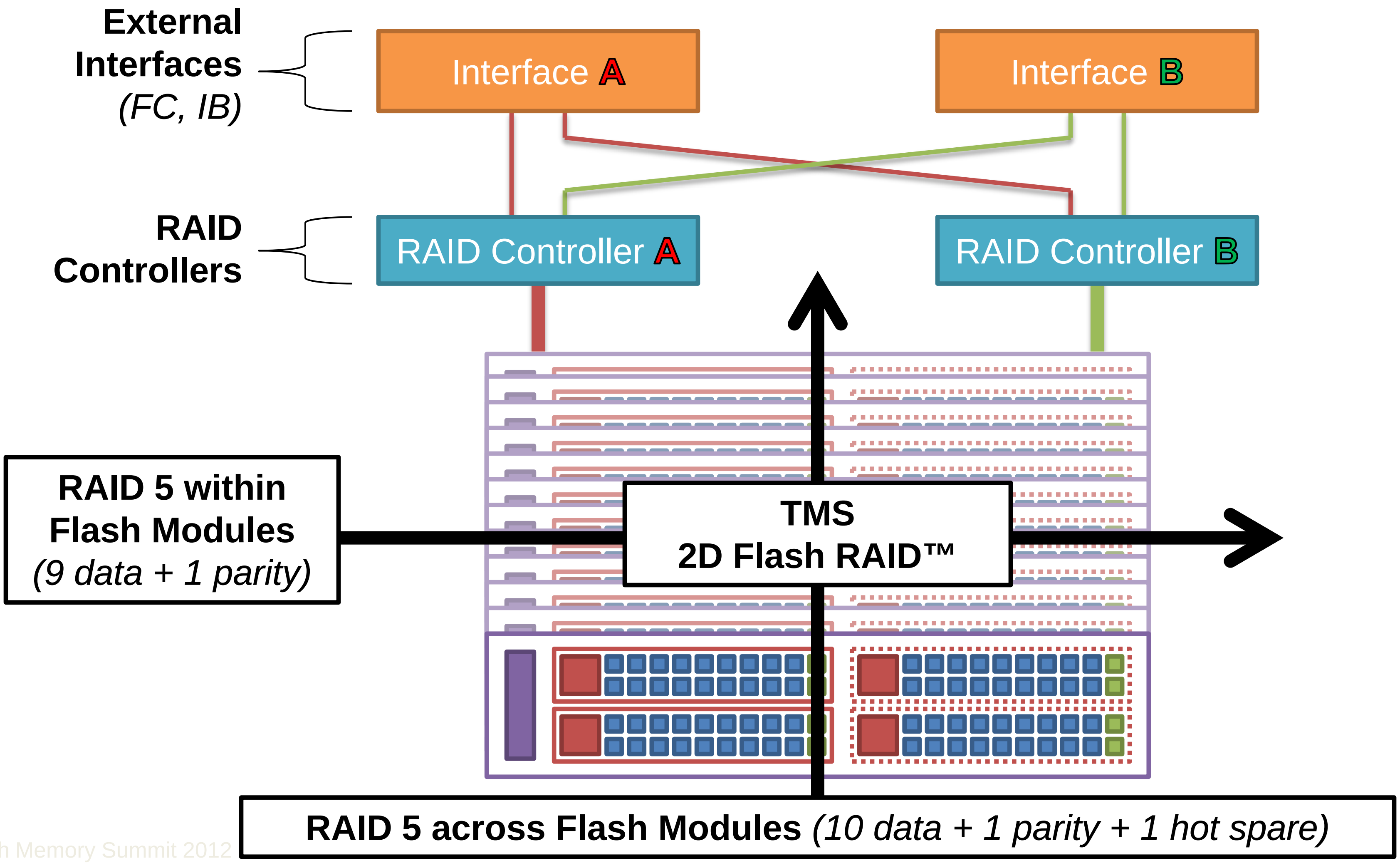


Example HA Flash System Design

System Architecture



Example HA Flash System Design
Data Protections



Conclusions

- Design to avoid failures *in the entire stack*
- Need to start at the chip and module level, but can’t ignore the system level for truly large scale deployments
- Mirroring isn’t efficient, and sparing isn’t effective enough alone
- MTBF + MTPI = good picture of *meaningful* reliability for most customers