本文是作者在ACMUG 2016 MySQL年会上的演讲内容，版权归作者所有。

中国MySQL用户组（China MySQL User Group）简称ACMUG。
ACMUG是覆盖中国MySQL技术爱好者的一个技术社区，是Oracle User Group Community和MairaDB Foundation共同认可的MySQL技术社区。

我们关注MySQL，MariaDB，以及其他一切周边的开源数据库和开源工具，我们交流使用经验，推广开源技术，为开源贡献力量。

我们是开放社区，欢迎任何关注MySQL及其相关技术的人加入，我愿意跟其他任何技术组织和团体保持沟通和展开合作。

我们期望在我们的活动中大家都能以开心的、轻松的姿态交流技术，分享技术，形成一个良性循环，从而每个人都可以有一份收获。

ACMUG的口号：开源，开放，开心

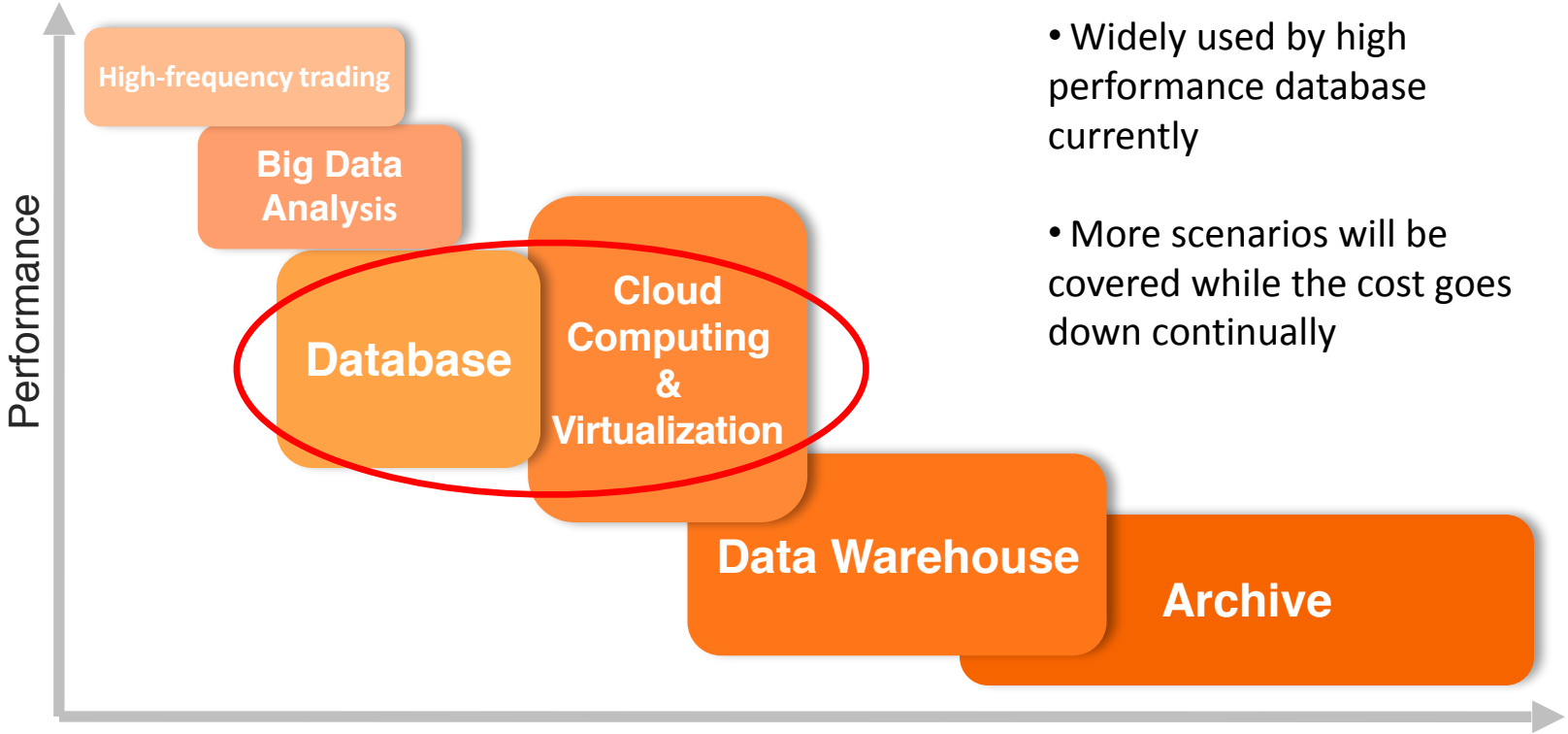
关注ACMUG公众号，参与社区活动，交流开源技术，分享学习心得，一起共同进步。

SSD Optimization for MySQL

- Shannon Systems
- 2016.12.10

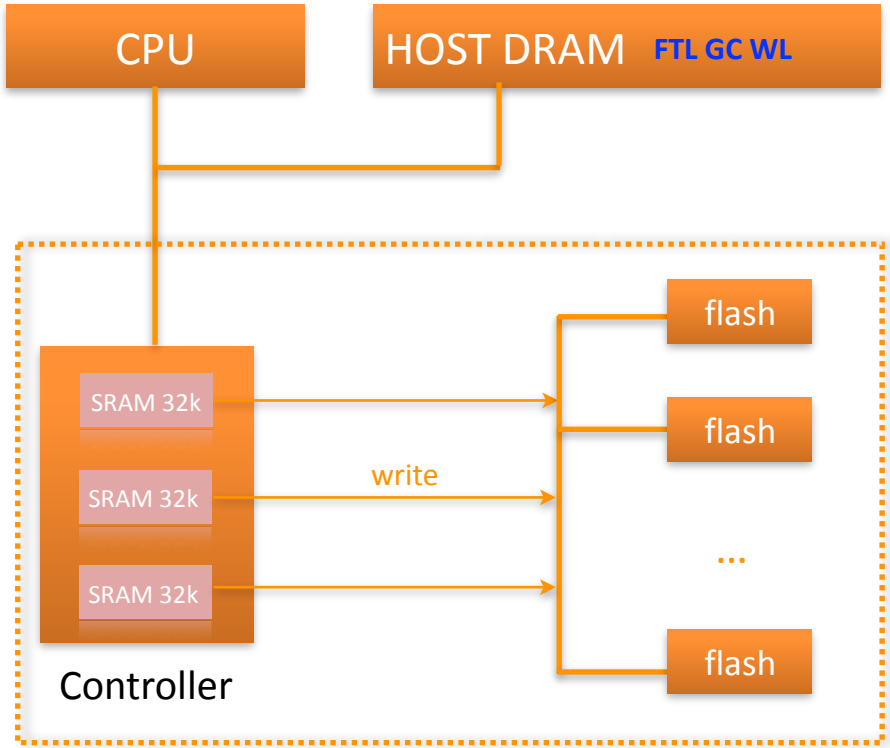
Shannon Systems

Application Scenarios of PCIe SSD



- Widely used by high performance database currently
- More scenarios will be covered while the cost goes down continually

Possibility — The Architecture



Host-Based Architecture

- FTL/GC/WL in host memory
- Write cache corresponding to flash page in size
- Software defined storage

What We Are Trying

- Atomic Write

  Partial page write causes redo log can’t be applied

- Prior Write

  Implementation of IO priority

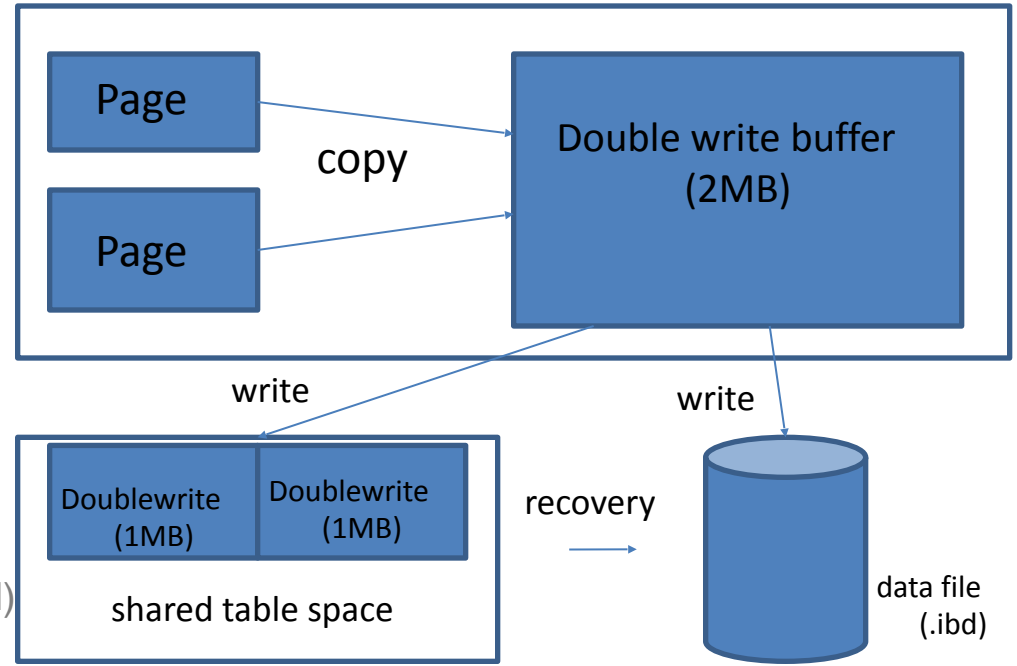
- Huge Capacity

  Up to 80TB usable space per single server

Double Write

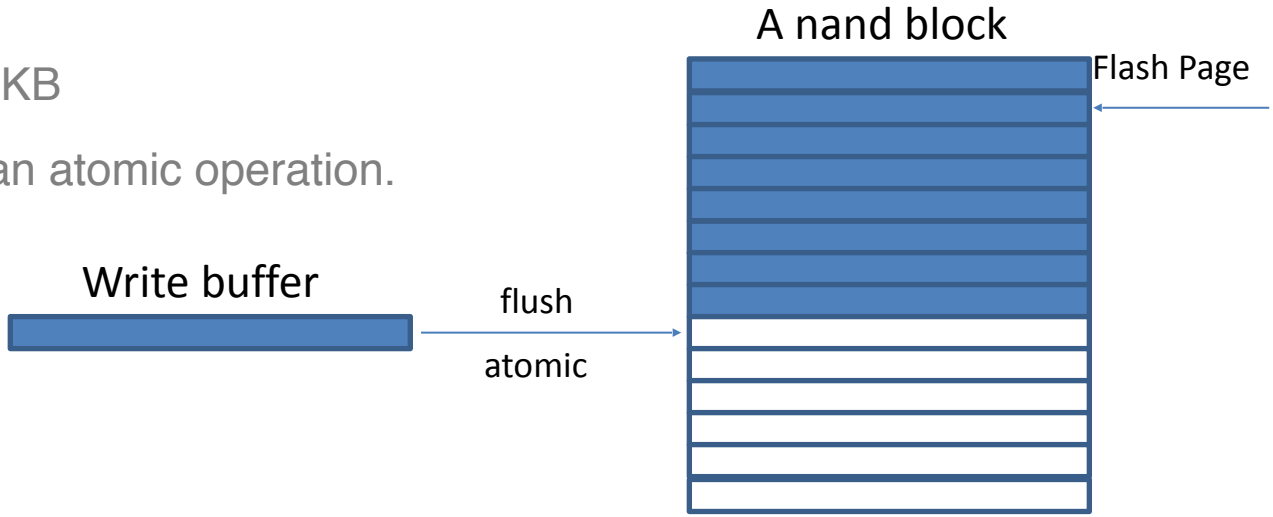
- The atomicity of InnoDB page write can't be insured by traditional hardware and OS.
- InnoDB uses double write to avoid partial page write.

- Downside of double write:
  - Doubled write amount(bad for flash)
  - Heavier write load(less than doubled)



NAND Page Write

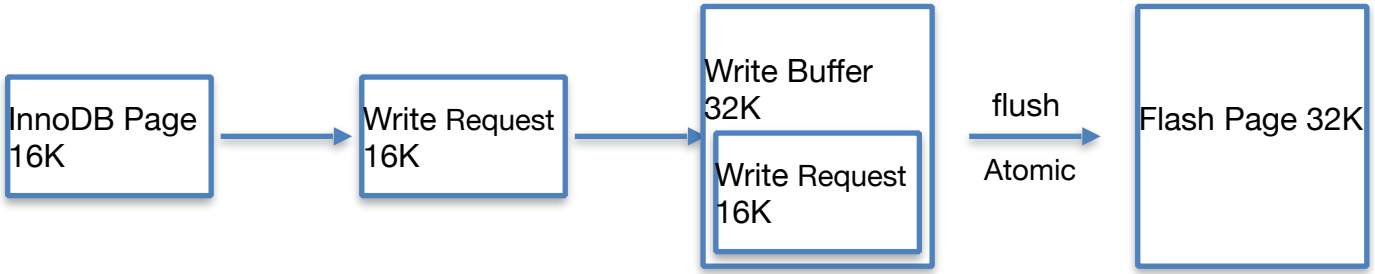
- Flash Page Size: 32KB
- NAND flash write is an atomic operation.



Atomic Write

- InnoDB Page Size: 16KB(default)
- Flash Page Size: 32KB

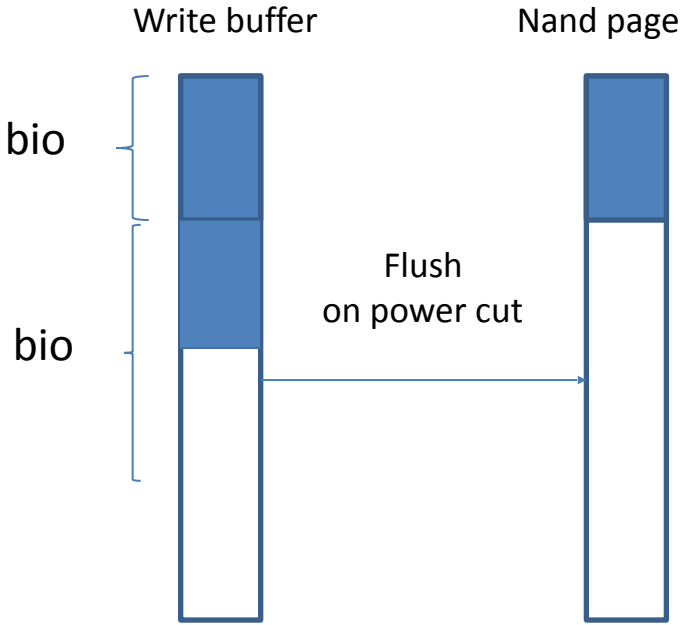
- If every flash page contains one or more InnoDB page(InnoDB page size<= flash page size), InnoDB page write will be atomic.



Atomic Write

- Request size <= Flash write buffer(32KB)
- Flash write buffer is empty

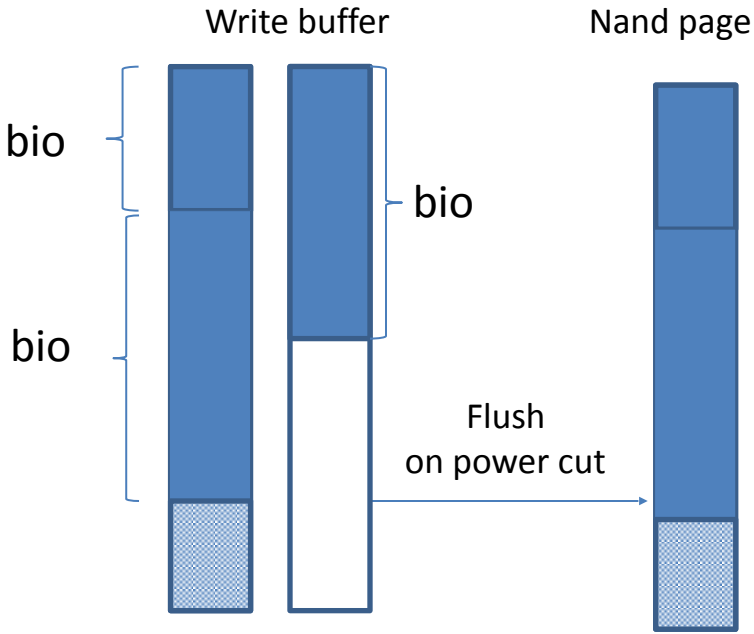
- Only fully written bio will be flushed into NAND page
- Partial written bio will be dropped



Atomic Write

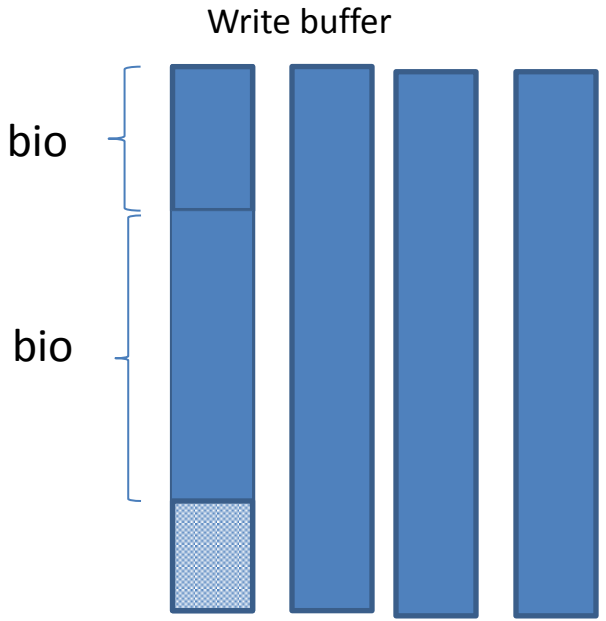
- Flash write buffer is half full.
- Case 1: Write request size <= Free space of flash write buffer

- Case 2: Write request size > Free space of flash write buffer
  - Fill free space with dummy data and open a new write buffer to store bio.



Atomic Write

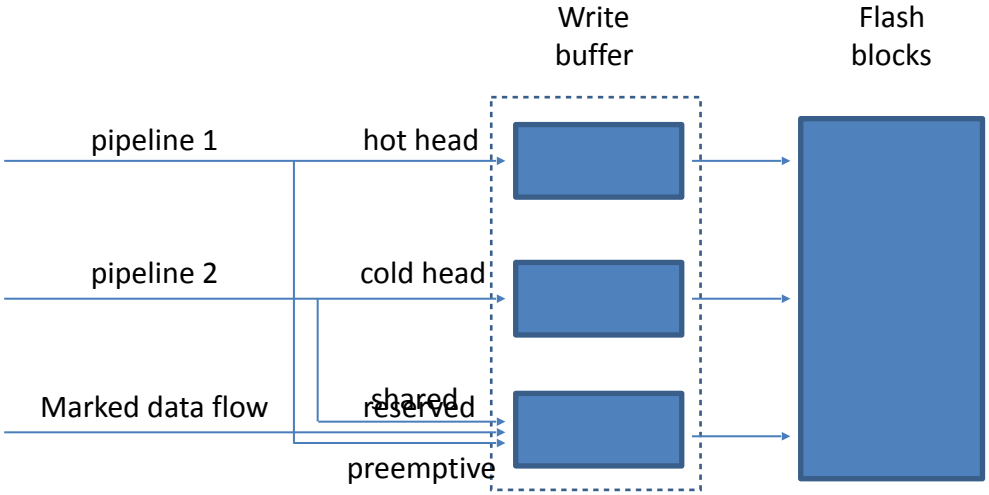
- Big write request(not InnoDB request) > Flash page size(32K)
- No buffer write(higher latency).



Benefits From Atomic Write

- TPS: ~10% increase
- Latency: 50% decrease at 95% percent line
- SSD endurance: 200% increase

Prior Write — Data Flow Markup



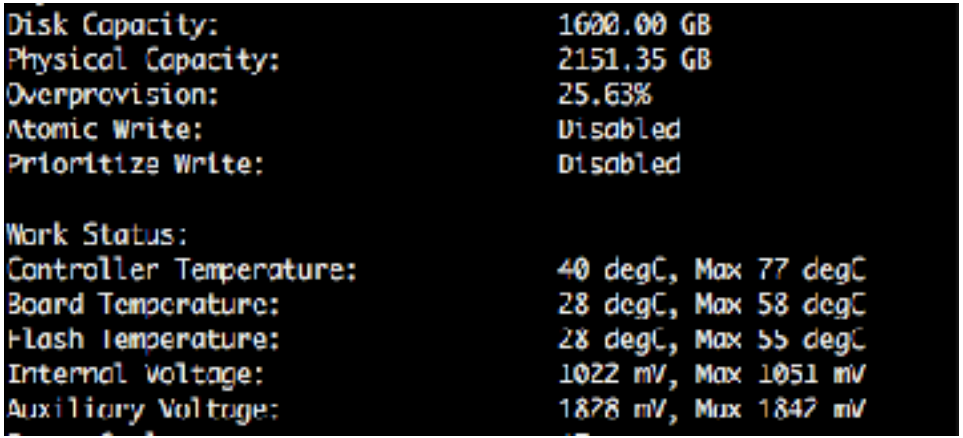
- Dedicated write buffer for marked data flow

Prior Write — Flag Delivery

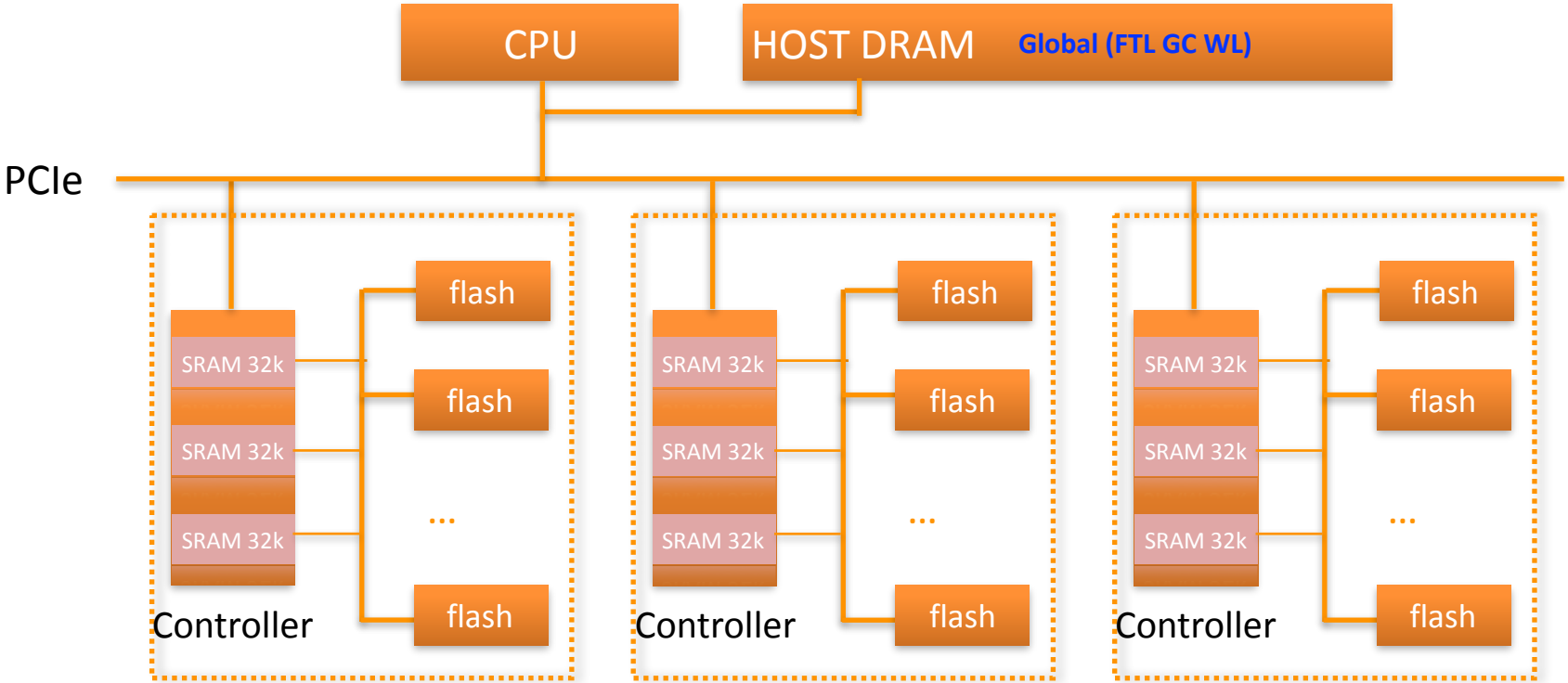
- App: generate flag in write request(eg. MySQL redo log)
- FS: get flag, set flag in BIO(Block IO)
- SSD driver: get flag, set high priority

Prior Write — Patch Required

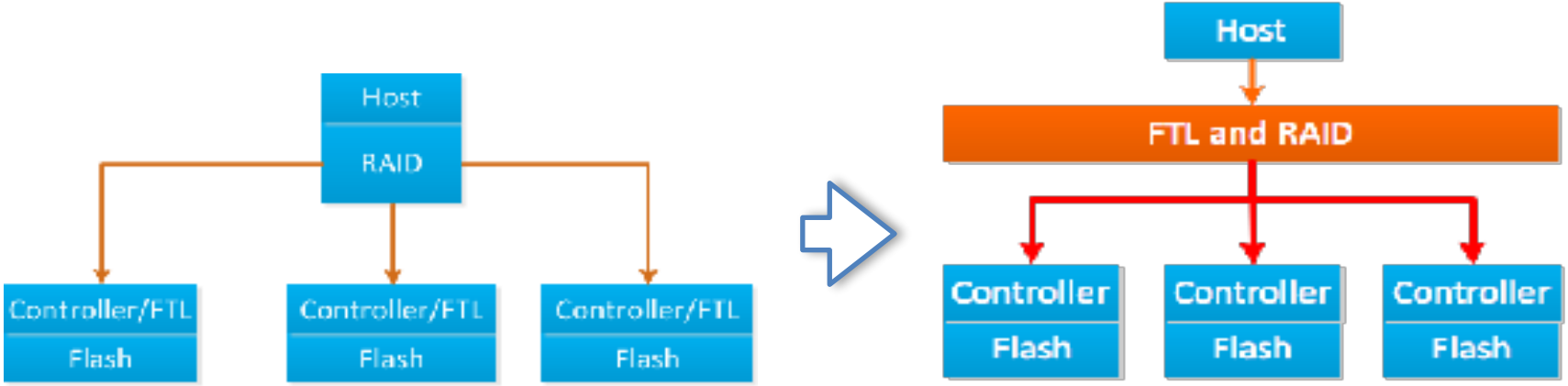
- App: set flag for particular data flow(less than 20 lines modification)
- FS: get flag, set flag in struct bio{}(less than 20 lines modification)
- SSD driver: get flag, set high priority(driver already has the toggle)



Huge Capacity— Global Redundant



Huge Capacity— Global Redundant



Benefits — waiting to be validated

- Performance
- Latency
- Capacity
- TCO

Thank you

Shannon Systems

Addr: Suit 1801, Wentong Building, 739 Kunming Road, Yangpu, Shanghai

Tel: 021-55580181

Email: contact@shannon-sys.com

Web: www.shannon-sys.com

Weibo: @宝存科技

WeChat: Shannon-Systems