一种高性能和可靠的混合客户端缓存

李 楚，冯 丹，王 芳

光电国家实验室，武汉，430074

华中科技大学计算机学院，武汉，430074

www.wnlo.cn

研究背景和动机

固态盘被广泛地应用为磁盘阵列的缓存层以提升存储性能。然而单一固态盘的可靠性远低于常用的冗余磁盘阵列 (如RAID5/6)，因此一旦固态盘缓存发生故障，将可能造成数据丢失以及破坏数据一致性等问题。

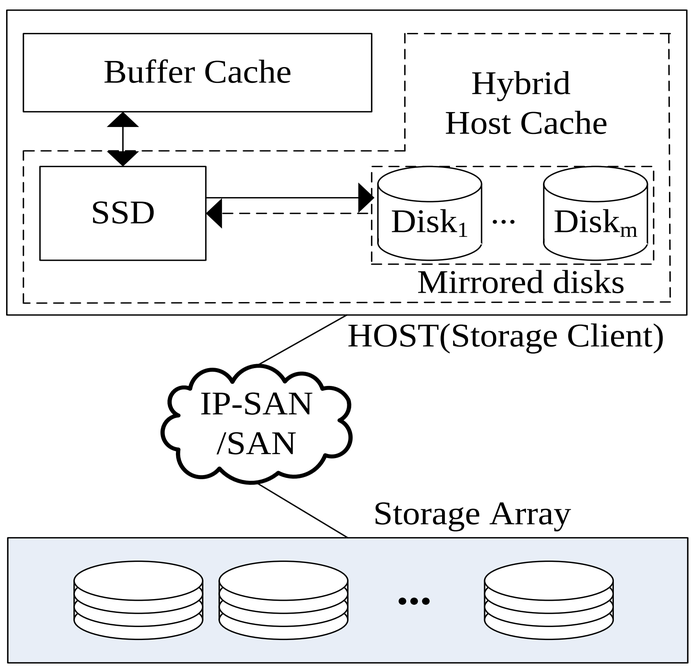
现有方案存在的问题

- 构建固态盘阵列提高缓存层可靠性[1,2]。然而这种方式将导致成本大幅增加，且冗余数据的写入加速了固态盘的磨损。
- 使用新的缓存策略保证数据一致性[3,4]。然而[3]无法保证数据的持久性；[4]产生频繁的缓存刷回操作，对缓存性能产生不利影响。

缓存设计目标

		WT	WB	WB-ordered	WB-flush	HHC
一致性		√	×	√	√	√
持久性		√	×	×	√	√
性能	读密集	高	高	高	高	**高**
	写密集	低	高	高	中	**高**
	同步密集	低	高	高	低	**高**

HHC(Hybrid Host Cache) 设计与实现



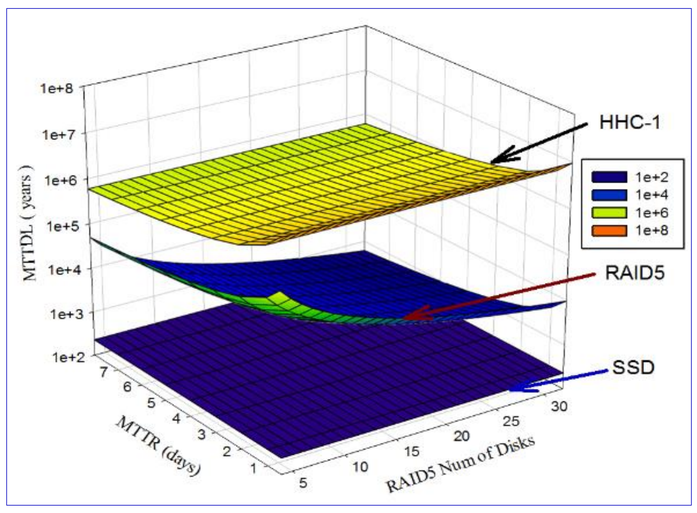
HHC基本架构

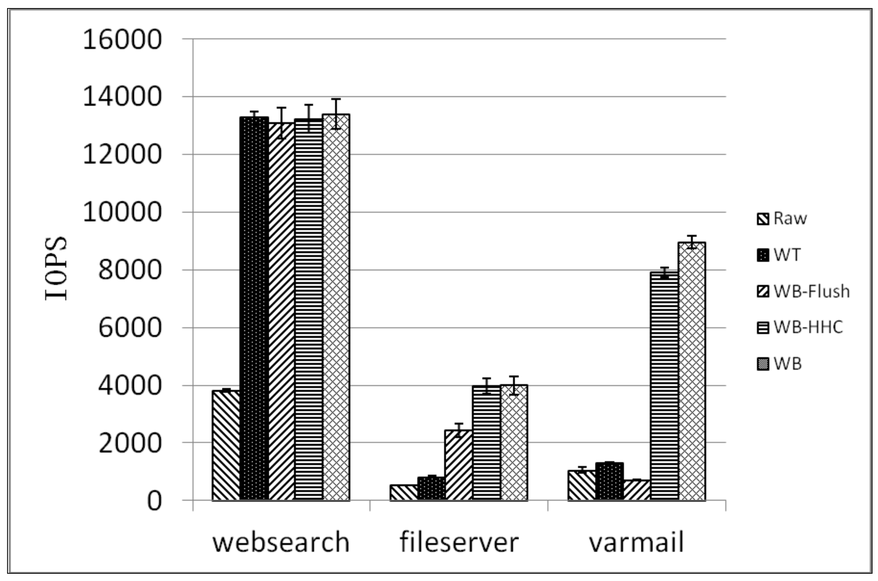
1 SSD + *n* HDDs (e.g., *n* =1, 2)

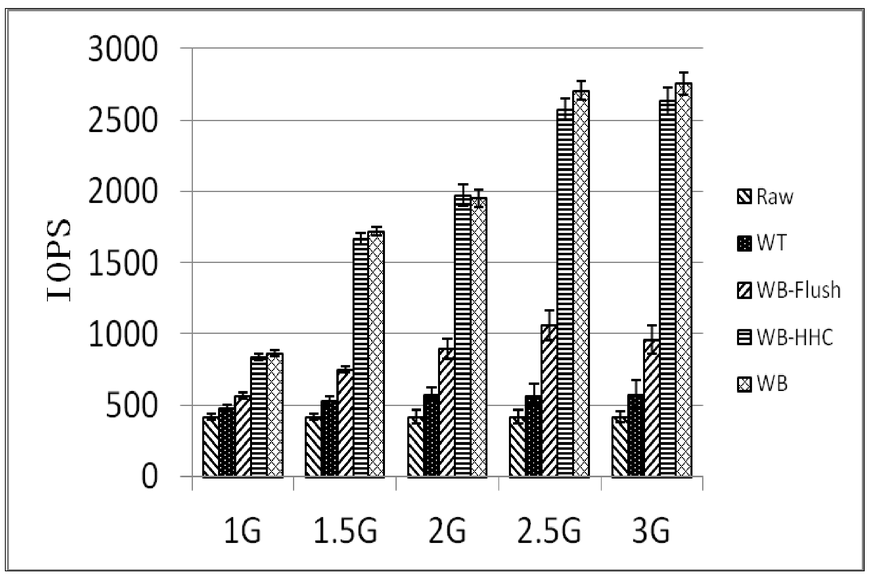
HHC主要模块

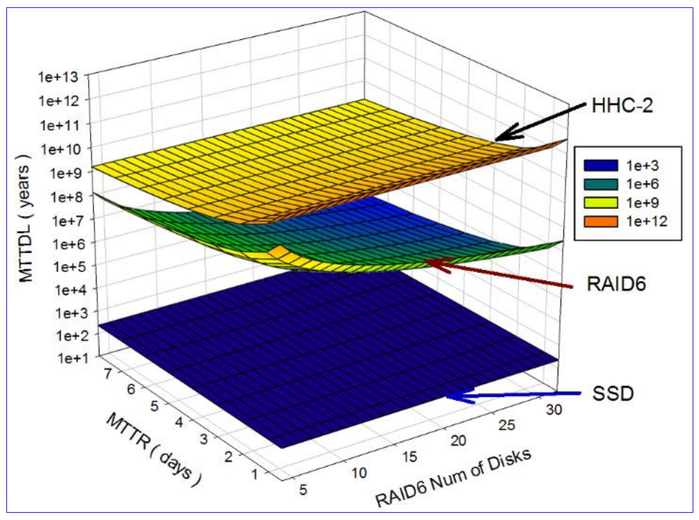
- **缓存管理：**在写回法和传统的置换算法(如LRU)的基础上，增加写屏障语义，选择性的将脏数据写入磁盘已保证固态盘缓存中脏数据的可靠性。为减少脏数据写回的突发性，HHC首先将脏数据写入临时缓存并批量的持久化到磁盘。
- **日志管理：**多个磁盘互为镜像，且以日志的方式持久化保存写入的数据，以最大限度发挥磁盘的顺序写性能。当需要日志回收时，从固态盘中读取相应的脏数据写回阵列，利用固态盘高随机读性能的特性大大降低垃圾回收的开销。
- **故障处理：**当发生设备故障时，将缓存置为只读模式，并将所有脏数据刷回到磁盘阵列，从而保证数据的可靠性。通过设置缓存脏数据的阈值，以及日志盘的垃圾回收频率，从而减少故障处理的时间。

可靠性分析及性能测试









可靠性分析（MTTDL）

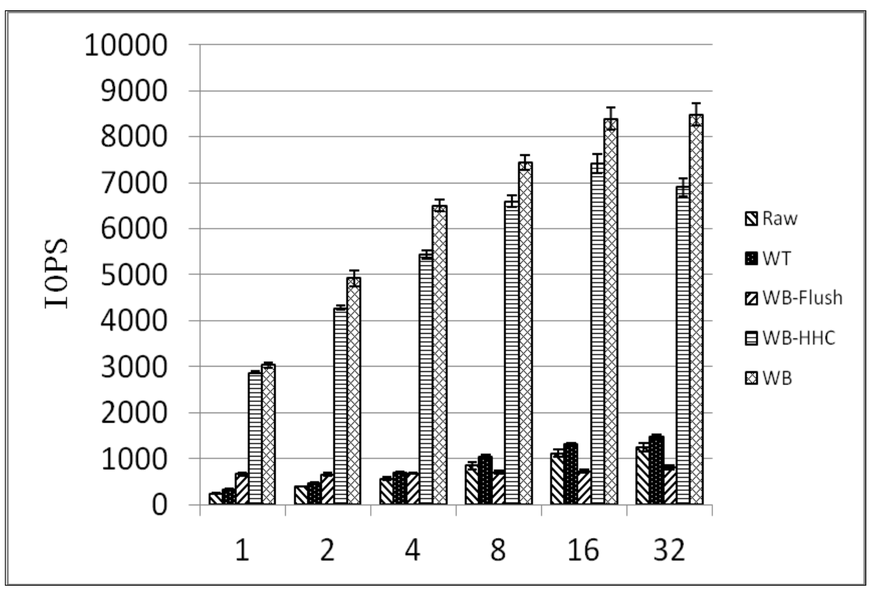
- HHC-1 > RAID5
- HHC-2 > RAID6

性能测试环境

- CentOS 5.4 + Linux Kernel 3.19.2
- Host cache: HHC-1 (SSD+HDD)
- Server RAID: 5 disks RAID-5
- Benchmark: Filebench (websearch，fileserver，和varmail)

性能测试结果

- 相比于现有缓存算法，WB-HHC可以大大提高I/O性能，尤其是在写密集型以及同步密集型负载下，HHC的优势更加明显。



参考文献

[1] D. Arteaga and M. Zhao. Client-side flash caching for cloud systems. In Proceedings of International Conference on Systems and Storage, SYSTOR 2014, pages 7:1–7:11.

[2] Y. Oh, J. Choi, D. Lee, and S. H. Noh. Improving performance and lifetime of the ssd raid-based host cache through a log-structured approach. In Proceedings of the 1st 5 Workshop on Interactions of NVM/FLASH with Operating Systems and Workloads, INFLOW '13, pages 5:1-5:8.

[3] R. Koller, L. Marmol, R. Rangaswami, S. Sundararaman, N. Talagala, and M. Zhao. Write policies for host-side flash caches. In Proc. of the 11th USENIX Conference on File and Storage Technologies (FAST'13), pages 45–58, February 2013.

[4] D. Qin, A. D. Brown, and A. Goel. Reliable writeback for client-side flash caches. In 2014 USENIX Annual Technical Conference (USENIX ATC 14).

Data Storage & Application Lab