学校编码：10384　　分类号____密级____
学号：23020141153161　　UDC____

厦门大学

硕 士 学 位 论 文

# 固态盘存储系统的性能优化研究

# Research on Performance Optimization for SSD-Based Storage Systems

杨伟健

指导教师姓名：吴素贞 副教授
专 业 名 称：计算机科学与技术
论文提交日期：2017 年 月
论文答辩时间：2017 年 月
学位授予日期：2017 年 月

答辩委员会主席：__
评阅人：__

# 厦门大学学位论文原创性声明

本人呈交的学位论文是本人在导师指导下,独立完成的研究成果。本人在论文写作中参考其他个人或集体已经发表的研究成果，均在文中以适当方式明确标明，并符合法律规范和《厦门大学研究生学术活动规范（试行）》。

另外，该学位论文为（　　）课题（组）的研究成果，获得（　　）课题（组）经费或实验室的资助，在（　　）实验室完成。（请在以上括号内填写课题或课题组负责人或实验室名称，未有此项声明内容的，可以不作特别声明。）

声明人（签名）：

年　　月　　日

# 厦门大学学位论文著作权使用声明

本人同意厦门大学根据《中华人民共和国学位条例暂行实施办法》等规定保留和使用此学位论文，并向主管部门或其指定机构送交学位论文（包括纸质版和电子版），允许学位论文进入厦门大学图书馆及其数据库被查阅、借阅。本人同意厦门大学将学位论文加入全国博士、硕士学位论文共建单位数据库进行检索，将学位论文的标题和摘要汇编出版，采用影印、缩印或者其它方式合理复制学位论文。

本学位论文属于：

（　　）1.经厦门大学保密委员会审查核定的保密学位论文，于　　年　　月　　日解密，解密后适用上述授权。

（　　）2.不保密，适用上述授权。

（请在以上相应括号内打“√”或填上相应内容。保密学位论文应是已经厦门大学保密委员会审定过的学位论文，未经厦门大学保密委员会审定的学位论文均为公开学位论文。此声明栏不填写的，默认为公开学位论文，均适用上述授权。）

声明人（签名）：

年　　月　　日

# 摘 要

数据量的激增给计算机存储系统带来严峻的性能挑战。固态盘作为新型存储介质，已被企业级存储系统和数据中心广泛使用，但简单地将传统存储软件应用于固态盘将带来一些新的问题，如将磁盘阵列技术应用于固态盘会加剧固态盘的垃圾回收操作所引发的性能波动性问题，将传统键值（Key Value，简称 KV）存储系统部署于固态盘上也可能造成存储效率低下等问题。本文分别以黑盒和白盒的使用角度，分别对固态盘阵列和基于固态盘的 KV 存储系统进行研究。

首先，对固态盘阵列的最优分块大小进行研究，理论分析和真实设备的性能测试结果表明，固态盘阵列的分块大小会对固态盘阵列的读写性能产生较大影响。基于分块大小与真实应用负载的读写密集特性分析结果，提出一种基于负载访问特性的多分块大小固态盘阵列（Multi-Chunk RAIS，简称 MC-RAIS），根据负载访问特性对数据进行布局，综合多种分块大小的优势，提高了固态盘存储系统的性能。性能测试结果表明，相对于固定分块大小的固态盘阵列，MC-RAIS 有效提高了 10%-20%的实时响应性能。

其次，以白盒使用角度研究了基于固态盘的 KV 存储系统。Open-Channel SSDs 是一种解决垃圾回收、负载均衡等问题的固态盘，即把固态盘白盒化，闪存块管理、数据布局、负载均衡以及垃圾回收等功能均可以由上层应用软件自主定制。在基于 LightNVM 的 Open-Channel SSDs，提出一种提供部分文件接口功能和基于应用特性定制的闪存管理中间件（Open-Channel based File System middleware，简称 OC-FS），连通上层 KV 数据库与底层存储设备，减少冗余 I/O 栈并整合上下层软件的垃圾回收操作，提高存储效率和性能。性能测试结果表明，OC-FS 可以提升 KV 存储系统约 50%的系统性能。

上述性能优化的两个关键技术分别以黑盒和白盒的角度对固态盘存储系统进行了深入研究，详实的性能测试展示了其有效性和可用性，为业界解决类似问题提供了思路和启发。

**关键词：**固态盘；键值存储；性能优化

# Abstract

With rapidly increasing of data, computer storage systems encounter serious performance challenge. As new storage media, Solid State Drives (short for SSDs), have been widely used in enterprise storage systems and data centers. However, simply applying traditional storage software to SSDs will bring some new problems, such as the performance variability issue caused by Garbage Collection (short for GC) operations on Redundant Array of Independent SSDs (short for RAIS) and the low storage efficiency issue caused by the dual-blind design in traditional key-value stores(short for KV stores) which are deployed on SSDs. From the point of view of black-box and white-box, this paper conduct researches on RAIS and Open-Channel SSDs based K-V store.

First, we study the optimal chunk size of RAIS. The theoretical analysis and performance evaluations on real devices showed that the chunk size of RAIS have a great impact on the read and write performance of RAIS. Based on the analysis of the optimal chunk size and the study on the read/write intensity of real workload, we propose a Multi-Chunk RAIS (short for MC-RAIS), which layouts data according to the workload characteristics and takes advantages of multiple chunk sizes, to improve the performance of the SSD-based storage systems. Performance evaluation result shows that MC-RAIS outperforms the existing fix-chunk-size RAIS in the I/O performance measure by 10%-20%.

Second, we conduct study on SSD-based KV store from the point of view of white-box. Open-Channel SSDs (short for OCSSDs) are new kind of SSDs which can solve the internal problems of SSDs, such as garbage collection and wear-leveling, by making SSDs be white boxes to the upper software. In OCSSDs, the NAND Flash block management, data layout, wear leveling, garbage collection and other mechanisms can be independently customed by the upper application software. In LightNVM based Open-Channel SSDs, we propose an Open-Channel based File

System middleware, short for OC-FS, to provide some file interface functions and enable application characteristics based customization. By connecting the upper-layer KV database and the lower-layer storage device together, OC-FS reduces the redundant I/O stack and integrates the GC operations of both the upper-layer and lower-layer software, thus improving the storage efficiency and performance. Performance evaluation result shows that OC-FS outperforms the existing traditional KV store system by approximately 50%.

From the perspective of black-box and white-box, we deeply study two key technologies of SSDs to improve the performance of SSD-based storage systems. Extensive performance evaluations demonstrate the effectiveness and usability of our proposed optimization technologies, thus providing thinking and inspiration for industries to solve similar problems.

**Key words**: Solid State Drives; Key Value Store; Performance Optimization.

# 目录

# Content

# 第一章 绪论

## 1.1 课题背景和研究意义

当代社会，信息化革命已经深入到人们生活的方方面面，数据资源已成为保证社会分工与正常运转不可或缺的资源之一。据统计表明[1]，因特网每秒新增数据量约 28,875GB，该数据的加速趋势还在不断增加。“信息爆炸”给计算机存储系统带来严峻挑战，在这一背景下，传统磁盘存储系统的性能瓶颈问题更是日趋严重。相比于磁盘（Hard Disk Drives，简称 HDDs），基于 Flash 技术的固态盘（Solid State Drives，简称 SSDs）能提供更高的 I/O 响应性能。因此，作为可能替代磁盘的持久化存储介质，固态盘逐步进入人们的视野，得到业界的广泛关注[2][3][4]。相比于磁盘，固态盘由于其内部结构没有机械部件（如磁头、转轴、盘片等），而是直接由半导体组件构成，数据传输时延中不存在由机械运动引发的寻道时间等时延，固态盘性能相对于传统机械磁盘有极大提升，具体表现为高随机读写性能，并同时具备高可靠性和低能耗[5][6]。由于固态盘优异的 I/O 性能，许多笔记本厂商如苹果、联想等已开始大量使用固态盘；另外很多数据中心，如 Google、Intel、百度、淘宝等也已部署了基于固态盘的存储系统。

固态盘主要由以下几部分组成：控制器、寄存器与缓存机构，若干持久化存储芯片以及连通控制器、缓存机构和对外数据接口的数据总线。其中，市面上绝大部分固态盘的存储芯片使用基于 NAND-Flash 的闪存颗粒作为数据存储单元（Memory Cells）。由于闪存颗粒的固有属性[5][6][7]，固态盘对数据读写的处理完全不同于磁盘。例如，闪存颗粒在写入数据（Program）时必须先进行擦除操作（Erase），该写入/擦除操作称为闪存颗粒的 P/E 操作，每个闪存颗粒的 P/E 操作存在一定上限，当该颗粒的 P/E 操作数超出该上限后该颗粒即被写穿（wear-out），无法继续使用。针对闪存颗粒的读写特性，固态盘内部提供闪存转换层（Flash Translation Layer，简称 FTL）以达到维护和管理闪存颗粒的读、写、擦除的需求，并提供一系列机制如校验、并行存取、垃圾回收、负载均衡等。针对不同场

景、不同需求， 学术界对固态盘的闪存转换层展开了不同的研究[8][9][10]。闪存转换层本质上是对固态盘存储颗粒的一套管理机制，其解决的主要问题包括以下几个方面。

（1） 数据布局

由于闪存颗粒的固有属性，在数据写入时需要对原始物理地址所在颗粒进行擦除操作，这就要求固态盘需要以异地更新的方式维护内部数据，而如何进行数据布局则是异地更新方式的核心，关系到盘内通道（channel）间、平面（plane）间的负载均衡。

（2） 垃圾回收策略

固态盘处理写更新请求时，将坏块表（Bad Block Table，简称 BBT）中原物理地址所在表项置为无效，并在其他物理地址写入新数据，根据某种策略，在某个时间点统一地将无效地址擦除，该批量化的擦除操作称为固态盘的垃圾回收（Garbage Collection，简称 GC），固态盘 GC 操作期间，对外响应性能将明显下降，因此高效、均衡的垃圾回收策略至关重要。

（3） 坏块管理与损耗均衡

当闪存颗粒擦除次数超出擦除上限时，该颗粒被写穿，后续无法继续使用该物理地址，一方面，固态盘内部需要管理某种映射关系使已损坏的闪存颗粒不被继续使用，另一方面，固态盘需要一定的损耗均衡策略，控制闪存颗粒的损坏情况。

在大部分企业与学术应用场景下，固态盘作为一个黑盒被使用。一方面，受益于操作系统对底层设备驱动的抽象，固态盘使用者（如企业级用户、学术研究机构等）无须了解底层物理设备的使用特性，在部署应用时无须根据特定底层设备进行特殊配置，例如，使用部署于传统磁盘存储软件部署于固态盘时无须进行特殊配置；但另一方面，基于传统磁盘的存储软件系统并不完全适用于固态盘，例如，将磁盘阵列技术（Redundant Array of Independent Disks，简称 RAID）直接应用于固态盘，即将若干固态盘组成固态盘阵列（Redundant Array of Independent SSDs，简称 RAIS），往往在某些场景下造成性能损失（Performance Degradation）；再如，将基于键值对的数据库（Key-Value Database，简称 KV 数