**Enabling Cost-effective Data Processing with Smart SSD**

主机处理器➕固态硬盘嵌入式处理器

应用领域：一般性（计算功能可由用户编程）数据密集型应用

图1是几种计算模型，（a）存在能耗大开销大的问题，（b）（c）在机械硬盘内部或利用FPGA配置丰富的计算和内存资源会导致设计复杂和制造成本高昂等问题。随着SSD的广泛应用，在固态盘内部利用ISP应运而生。

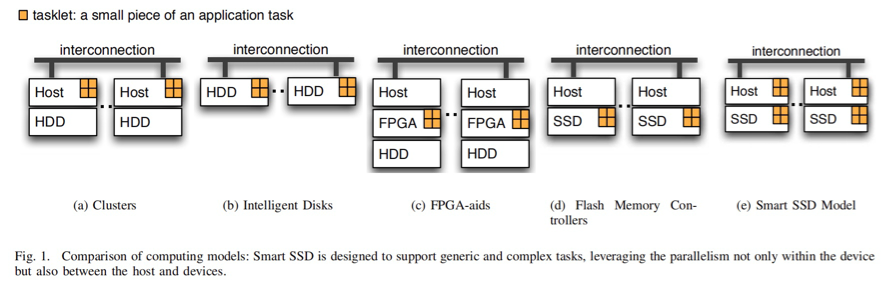
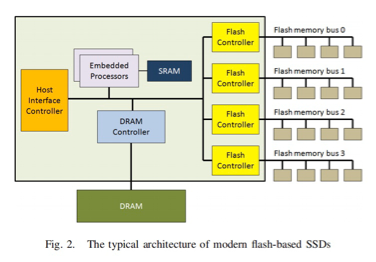


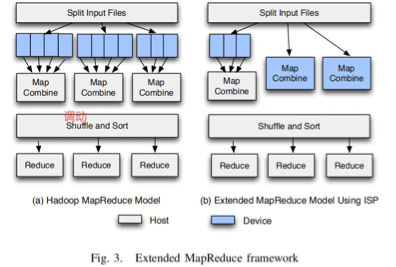
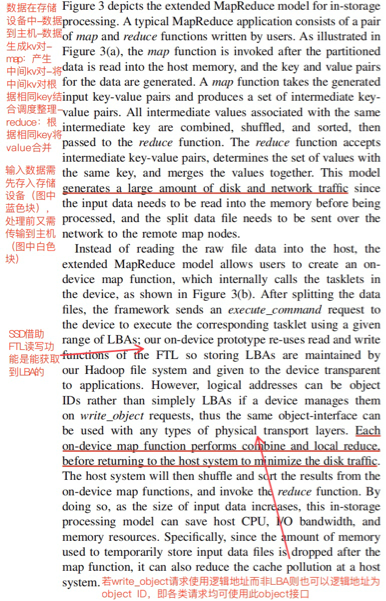
图2是SSD的结构，其主要元件包含SSD控制器、DRAM和闪存阵列。其中SSD控制器包含有主机接口控制器、嵌入式处理器和闪存控制器（FMC）。

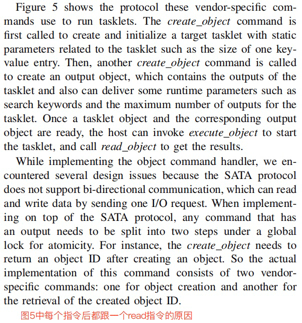
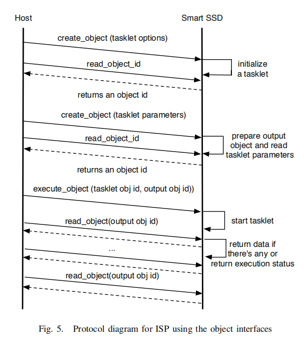
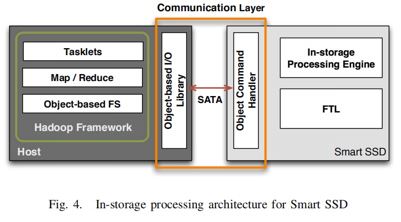


由于闪存阵列具有极高的并行性，SSD内部带宽远远高于主机设备硬件接口所能提供的带宽，若不改变硬件，则无法完全利用SSD高性能的优势。另外，利用主机进行数据处理时，需要知道存储设备内配置的信息才能进行高效的请求调度，而暴露配置信息具有一定的安全性问题。然而，利用SSD控制器进行数据处理，无需将数据由存储设备中转移到主机中，即无需扩宽主机-设备间数据通道，即无需硬件改变，就可以利用闪存阵列的高并行性实现高性能的数据处理，同时也无需暴露SSD配置信息即可进行调度。另外，SSD控制器相较于主机处理器能耗更低。综上，在SSD内部利用ISP具有高性能、低能耗的优点。

Smart SSD结构如图1（e）所示，这一计算模型性能优于（d），因为在进行复杂计算时，也可利用主机处理器完成。（SSD控制器处理能力有限）

在主机-设备原本的数据通道上，还需额外的IO指令用于管理和执行任务的传递。考虑到MapReduce编程模型可以将输入数据直接映射到任务，Smart SSD基于MapReduce模型（MapReduce编程模型用于大规模数据集的并行运算。概念"Map（映射）"和"Reduce（归约）"是主要思想，极大地方便了编程人员在不会分布式并行编程的情况下，将自己的程序运行在分布式系统上）实现了一个扩展的object-based接口。该接口运行于现存设备接口的顶层，这样主机端和设备端均可应用同一套API。扩展的MapReduce结构将Map移入设备中完成，实现ISP，具体如下图3和论文截图所示。

综上，图4是Smart SSD的ISP结构图。设备端ISP引擎即SSD控制器，因是object-based接口，用户视角视任务为目标且具有唯一的ID以供识别。设备端Object Command Handler，其API（具体如下图5及论文截图所示）由固件实现，接收主机指令并触发设备中任务的执行。主机端在Hadoop MapReduce结构上添加设备内Map功能及object-based文件系统以实现ID与LBA的联系。主机端object-based IO library处理请求并将其传递给FS（联系ID与LBA以便设备端的后续处理）。



Summary：

美国加州大学圣克鲁兹分校和三星半导体、三星微电子于2013年提出Smart SSD概念。在不对操作系统做更改的情况下实现高性能、低功耗的ISP系统，以一套API实现object-based接口以供交互，并提供一套SSD内部运行的Hadoop MapReduce系统。