# 文章列表（Processing in SSD）

1. A Case for Intelligent Disks (IDISKs) (SIGMOD’98)
2. Active flash: Out-of-core data analytics on flash storage (MSST'12)
3. Enabling cost-effective data processing with smart SSD (MSST'13)
4. Active Disk Meets Flash: A Case for Intelligent SSDs (ICS'13)
5. Active Flash: Towards Energy-Efficient, In-Situ Data Analytics on Extreme-Scale Machines (FAST'13)
6. Minerva: Accelerating Data Analysis in Next-Generation SSDs (FCCM'13)
7. Query Processing on Smart SSDs: Opportunities and Challenges (SIGMOD'13)
8. XSD: Accelerating MapReduce by Harnessing the GPU inside an SSD (1st Workshop on Near-Data Processing 2013)
9. Willow: A User-Programmable SSD (OSDI'14)
10. Scalable Multi-Access Flash Store for Big Data Analytics (FPGA’14)
11. BlueDBM: an appliance for big data (ISCA'15)
12. CoDEN: A Hardware Software CoDesign Emulation Platform for SSD-Accelerated Near Data Processing (ISCA'15)
13. Design space exploration for efficient computing in Solid State drives with the Storage Processing Unit (NAS'15)
14. Self-Sorting SSD: Producing Sorted Data Inside Active SSDs (MSST'15)
15. Energy Efficient Scale-In Clusters with In-Storage Processing for Big-Data Analytics (MEMSYS’15)
16. Biscuit: A Framework for Near-Data Processing of Big Data Workloads (ISCA'16)
17. Morpheus: Creating Application Objects Efficiently for Heterogeneous Computing (ISCA'16)
18. In-storage processing of database scans and joins (Information Sciences 2016)
19. YourSQL: a high-performance database system leveraging in-storage computing (VLDB'16)
20. Summarizer: Trading Communication with Computing Near Storage (Micro'17)
21. SSD In-Storage Computing for Search Engines (TC 2017)
22. High-performance data mining with intelligent SSD (Cluster Computing 2017)
23. GraFBoost: Using Accelerated Flash Storage for External Graph Analytics (ISCA'18)
24. CISC: Coordinating Intelligent SSD and CPU to Speedup Graph Processing (ISPDC'18)
25. HODS: Hardware Object Deserialization Inside SSD Storage (FCCM’18)
26. Programmable solid-state storage in future cloud datacenters (ACM 2019)
27. REGISTOR: A Platform for Unstructured Data Processing Inside SSD Storage (TOS 2019)
28. REACT: Scalable and High-Performance Regular Expression Pattern Matching Accelerator for In-Storage Processing (TPDS 2019)

# 文章分类

**根据体系架构对文章进行分类：**

|  |  |  |
| --- | --- | --- |
| **存算融合的体系架构类型** | | **文章序号** |
| 主机处理器 + 固态硬盘 | 固态硬盘内：嵌入式处理器 | 2, 3, 5, 7, 9, 11, 12, 13, 14, 15, 17, 20, 21 |
| 固态硬盘内：专用硬件加速器 | 4, 6, 10, 16, 18, 19, 24, 23, 25, 27, 28 |
| 固态硬盘内：图形处理器 | 8 |
| 主机处理器 + 主机图形处理器+ 固态硬盘 | | 17, 22 |
| 分布式计算 | | 5, 10, 11 |

**根据应用领域对文章进行分类：**

|  |  |  |
| --- | --- | --- |
| **应用领域** | | **文章序号** |
| **一般性数据密集型应用** | | 3, 8, 9, 11, 12, 13, 15, 16, 20 |
| **某类应用** | 数据库 | 7, 14, 18, 19 |
| 高性能计算（科学仿真） | 2, 5, 17 |
| 图计算 | 23, 24 |
| 搜索引擎 | 21 |
| 数据挖掘 | 22 |
| 数据分析 | 6, 10, |
| **特定功能** | 正则表达式匹配 | 27, 28 |
| 反序列化 | 17, 25 |
| 自定义 | 4 |

**备注：**

**一般性：**计算功能由用户可编程；

**特定功能：**固态硬盘内实现特定计算功能，面向特定类型的应用。

**根据编程框架对文章进行分类：**

|  |  |  |
| --- | --- | --- |
| **编程框架** | | **文章序号** |
| **通用框架** | Session-based | 7, … |
| RPC | 9, … |
| Event-driven | 3, … |
| Flow-based | 16, … |
| **自定义接口** | | 17, 24, 25, 27, … |

# 文章内容总结

（按照发表时间的顺序依次对每篇文章的主要内容进行总结）

存储内计算的概念早在20世纪70年代就被提出并应用于机械硬盘中，比如美国加州大学伯克利分校于1998年提出基于智能机械硬盘的体系结构，将多个内置有嵌入式处理器和内存的智能机械硬盘通过高速串行总线连接组成集群系统，运行数据库服务或其中的部分功能（比如扫描、排序和连接）。相比于利用昂贵的桌面处理器进行数据处理，该体系结构能够减少I/O总线上的数据传输，降低系统的能耗和总体拥有成本，并提高系统的性能和扩展性。

然而，由于在机械硬盘内部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，智能机械硬盘没有得到广泛地应用。近年来，基于NAND型闪存的固态硬盘技术快速发展，并被广泛用在运行数据密集型应用的系统中。固态硬盘内部不仅具有高并发的多通道存储结构，能够提供丰富的I/O带宽资源（超过固态硬盘外部接口的带宽资源），而且内置有强大的嵌入式处理器和相应的内存资源，提供了低功耗和低成本的计算资源。因此，固态硬盘非常适合用于实现存储内计算技术，相关问题成为学术界和工业界的研究热点。

美国东北大学和橡树岭国家实验室于2012年提出在高性能计算领域中利用主机处理器和固态硬盘内的嵌入式处理器共同进行数据分析（比如边缘探测、寻找局部极值、心跳检测和数据压缩等），并分析了该混合计算模型在性能和能耗方面存在的权衡。当更大比例的计算任务在硬盘内进行时，整体计算性能会下降，但能耗效率更高。仿真结果表明，相比于完全利用主机处理器进行数据分析，该混合计算模型能够以较小的性能代价大幅度提高能耗效率。

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。

韩国三星微电子和美国匹兹堡大学于2013年设计了专用SSD（非同时兼顾高性能+可编程），在SSD内集成更多嵌入式CPU和在每个闪存控制器中增加计算单元。并基于对性能分析与能耗模型对Intelligent SSD的收益进行了量化统计。

美国北卡罗来纳州立大学、东北大学和橡树岭国家实验室于2013年对数据分析任务进行了建模，包括SSD内的分期比率、性能、能耗模型。并应用这些模型将SSD设计成一个专用的数据分析的节点，定义数据分析命令。由Host端下发命令经SSD内调度，实现数据分析的盘内加速。

美国加州大学圣地亚哥分校和劳伦斯利福摩尔国家实验室于2013年提出了Minerva。将数据密集型的应用下放到基于FPGA设计的NVM controller中执行，硬件实现的FPGA controller显著地降低了主机和SSD之间的数据拥塞，对Host端提供一个的数据处理负载引擎。Minerva还提供一个runtime库供下放计算任务到SSD。

美国威斯康星大学麦迪逊分校，三星微电子和微软于2013年在Smart SSD基础上，利用Smart SSD实现对微软SQL Server查询功能的加速，同时降低了功耗开销。

韩国延世大学于2013年提出了XSD，XSD在SSD内新増GPU，以实现在并行I/O负载访问的处理。XSD提供了一个SSD+GPU实现的MapReduce框架API集合，从而允许用户充分利用其并行性加速应用。

美国加州大学圣地亚哥分校于2014年提出了一个用户可编程的SSD框架——Willow，使用户可以针对自己的应用进行编程以优化存储系统的高延迟、低带宽等问题。Willow在主机端实现了HRE(Host RPC Endpoints)以供主机/应用与SSD交互，在SSD端实现了一套微操作系统SPU-OS（存储处理单元-操作系统），指定部分I/O请求(SSDApps)在SSD内部完成。与运行在Host端的应用相比，采用Willow设计的SSD端的可编程的功能SSD Apps可以获得低延迟、高带宽的性能。Willow的编程模型也非常灵活，它支持多SSD Apps之间的并发运行，并且保证了SSD Apps能够安全可靠得运行。

美国麻省理工学院和Intel于2014年提出了一个可扩展的分布式闪存存储架构，该架构通过低延迟的网络接口共享闪存芯片，进而提供高性能、大容量的可扩展存储系统。此外，该架构通过基于FPGA的可编程闪存控制器对近数据处理任务加速，从而提升系统性能。

美国麻省理工学院于2015年针对大数据分析设计了一套分布式闪存存储系统BlueDBM。BlueDBM是一个基于闪存的可扩展的系统，该系统包括一个全局地址空间，一个灵活的Inter-Controller网络以及一个In-Store的计算单元，其中In-store处理引擎提供了一个支持用户自定义的软硬件协同设计环境。软件接口提供给用户文件系统接口、块设备接口、加速接口等三个可编程接口。最终实现一个复杂数据分析器，与同样功能的云系统相比，其能耗少一个数量级。

美国德克萨斯大学达拉斯分校于2015年提出一个固态硬盘仿真平台CoDEN，用于探索在存储内计算架构下的软硬件设计空间。该平台既支持调整各个硬件模块的配置参数，如嵌入式处理器的核心数量、内存的层次结构、以及存储介质的访问速度等，也包含完整的软件栈，可以实现不同的固件算法和计算功能。基于20种计算功能的实验结果表明，该平台的数据处理带宽高达2GB/s。

美国明尼苏达大学于2015年采用建模和仿真的方法探索了固态硬盘内实现存储内计算技术的设计空间，比如应用类型、数据布局、处理器核心数量、内部I/O带宽等因素对系统性能和能耗的影响，仿真结果显示存储内计算技术能够实现11 ~ 423倍的能耗下降和4 ~ 66倍的性能提升。

韩国成均馆大学和韩国科学技术院于2015年提出了自排序SSD的架构，将排序操作完全卸载到SSD中处理，在需要大量数据传输的数据密集型和数据库环境中减轻了排序操作的负担。基本思想是在SSD中构建一个B+树索引，后续通过遍历索引来获取记录，可以直接得到排序后的输出。该体系结构的主要优点是完全消除了用于计算排序结果的数据传输开销，从而减少了运行时间和能耗，并提高了SSD的寿命。他们使用Jasmine OpenSSD平台评估，利用该平台SSD控制器集成的单核87.5MHz ARM7TDMI CPU和64 MB DRAM，没有额外硬件开销，结果表明性能比传统的外部归并排序方案高出60.75%，同时还减少了58.86%的功耗。

韩国三星半导体于2016年提出了MapReduce框架，SSD内实现Spark。

韩国三星微电子于2016设计了Biscuit框架，基于NVMe协议或Ethernet协议（视Biscuit结构组织类型而定）并增加了一个硬件模式匹配器（hardware pattern matcher），实现对I/O的过滤与匹配。软件上设计并实现了Biscuit的runtime和一套API实现应用执行环境，在SSD端建立计算模块并实例化执行单元，协同管理控制任务的执行，辅助以过滤匹配、I/O通信、异步操作、API封装等，最终实现一套可编程的NDP框架，主要用以提升MySQL的查询速度。

美国加州大学圣地亚哥分校于2016年提出了Morpheus模型，将数据的反序列化下放到存储设备中进行，解决高性能计算中对象反序列化带来的瓶颈，避免不必要的系统开销，为计算密集型工作负载释放稀缺的CPU和主内存资源，节省I/O带宽，并降低功耗。通过拓展NVMe实现了Morpheus-SSD，将对象反序列化的性能提高了1.66倍，降低了7%的功耗，减少了42%的能耗，并将总执行时间加速了1.32倍。通过使用NVMe-P2P实现了Morpheus-SSD与GPU之间的点对点通信，在异构计算平台中，Morpheus-SSD可以将总执行时间加速1.39倍。

韩国全北国立大学于2016年提出在固态硬盘内增加专用的硬件计算模块，实现数据库的扫描和连接功能，减少硬盘到主机端的数据传输。仿真实验结果表明，相比于利用主机处理器进行数据处理，该方法能够将扫描操作和连接操作的性能最多提高7倍，将系统能耗最多降低45倍，而额外的硬件开销只占固态硬盘的1%。

韩国三星公司于2016年实现了一个利用存储内计算技术加速查询的数据库系统原型，具体方式是在一个用户可编程的固态硬盘内（利用其内置的模式匹配硬件模块）实现数据过滤功能，减少硬盘到主机端的数据传输。实验结果表明，相对于传统的数据库系统，该系统能够将查询性能最多提高15倍，将能耗降低近24倍。

美国南加州大学、北卡罗莱纳州立大学和加州大学圣地亚哥分校于2017年发现并不是所有任务在SSD内处理都能提升系统性能。其将Host端下发的请求分为计算型和I/O型，在SSD内进行计算，其实只对I/O密集而计算任务较为简单的任务能取得收益，因为SSD的处理器本身计算能力十分有限，将计算型任务放SSD内执行反正会拖慢系统性能。因此Summarizer在SSD内实现一个任务队列（Task Queue, TQ），如果TQ未满，指定在SSD内运行的任务会进入TQ并在SSD内部完成计算。当TQ满时，表示SSD处于高负载状态，所有任务会送回Host端进行处理。与手动指定I/O任务在SSD端执行相比，Summarizer简单地平衡了数据传输与SSD的计算能力，避免了SSD端的计算能力成为瓶颈而导致的系统性能下降。

美国加州大学圣地亚哥分校和韩国三星公司于2017年提出将存储内计算技术用于加速搜索引擎，求交集、求并集和求差集等五项搜索引擎中常用的功能被卸载到固态硬盘内部执行，基于一个开源搜索引擎的实验结果表明，存储内计算技术能够将搜索引擎的查询延时和能耗分别降低2~3倍和6~10倍。

韩国汉阳大学于2017年首先基于gem5仿真器开发了一个支持存储内计算的固态硬盘仿真器，然后提出综合利用主机端的处理器和图形处理器、以及固态硬盘内的嵌入式处理器三种异构计算资源，加速数据挖掘算法（比如k-means和PageRank）。主机图形处理器具有大量的并行计算核心，固态硬盘具有丰富的内部I/O带宽资源，因此它们能够进行高效的并行数据处理；相比而言，主机处理器更适合进行数据合并或高时间复杂度的计算。提出基于异构计算资源的静态调度方法。采用图形建模的方法将数据挖掘算法抽象为同步数据流（即一张描述函数之间关系的图）。仿真实验结果表明，相比于只利用单一或两种计算资源的架构，综合利用三种计算资源能够将性能最多提高10倍。

美国麻省理工学院于2018年提出基于硬件加速和闪存存储的图分析系统GraFBoost，它在固态硬盘内基于FPGA实现了一个数据排序和合并的硬件加速器，用于顺序化和合并对图数据结构的细粒度随机更新。GraFBoost只需要消耗很少的内存，就能够在闪存存储上进行高性能的大规模图分析。

美国罗德岛大学、深圳大普微电子科技有限公司于2018年提出了基于一个全面的软硬件协同设计框架CISC(协调智能SSD和CPU)来加速图计算中的MST计算。其思想是利用SSD内部的控制器逻辑对图形边缘进行预处理，将大量的图边缘划分为chunk，并使用硬件对每一chunk的边缘进行排序。在Open-SSD平台上，使用Xilinx Zynq-7000系列FPGA实现了CISC原型，对于串行版本实现2.2∼2.7倍加速，对于96核的并行版本达到了11.47~17.2倍加速。对于相同数量的核心，并行CISC比传统软件MST的性能高出35%。

美国罗德岛大学、深圳大普微电子科技有限公司于2018年提出了一种提供线速度对象反序列化的硬件方法，在SSD存储中对硬件对象反序列化HODS。基于NVMe拓展实现了包含FPGA模块和I/O总线的PCIe SSD卡，其中FPGA模块用来执行数据转换，与所有存储操作并行工作，并以流水线结构进行实时计算。在高端主机CPU上，与传统的反序列化相比，HODS有8\~12倍反序列化速度提升。对于浮点数据集，加速可以达到17\~21倍。根据反序列化时间占应用程序总运行时间的比例，应用程序级别的总体性能提升达到1.1\~4.3倍。

微软研究院、亚马逊AWS和美国加州大学圣地亚哥分校于2019年提出了在云数据中心中完全可编程存储基板的设想，允许应用程序开发人员以云的速度创新存储基础设施，就像软件应用程序/OS基础设施一样。SSD的可编程性演化提供了将它们作为云数据中心中一流的可编程平台的机会，使软件硬件创新成为可能，从而弥合应用程序/操作系统需求和存储能力/限制之间的差距。他们希望阐明软件定义存储的未来，并帮助为云数据中心设计、构建、部署和利用软件定义存储架构指明方向。

美国罗德岛大学、深圳大普微电子科技有限公司于2019年提出了一种用于在存储中抓取正则表达式的平台Registor，通过加速正则表达式(regex)在存储大数据集的存储器内的搜索，从而消除I/O瓶颈问题。Registor在SSD中利用FPGA设计并增强了一种用于regex搜索的特殊硬件引擎，在寄存器硬件中设计了一个深流水线结构，能够实时处理数据。他们还在Linux中开发了一组API和库，易于高级应用程序使用Registor。在实际的硬件平台测试结果表明，Registor实现了高吞吐量，减少了高达97%的I/O带宽需求，并且在大型数据集中减少了高达82%的CPU使用率，而硬件功耗极低，不到SSD原型总功耗的1%。

韩国延世大学于2019年提出在固态硬盘内配置一个正则表达式匹配的硬件加速器，用于并行处理多个数据流，并优先调度可能阻塞匹配操作的数据访问。相比于传统的闪存控制器调度算法，它将正则表达式匹配的性能最多提高了22.6%。