**Query Processing on Smart SSDs: Opportunities and Challenges**

主机处理器➕固态硬盘嵌入式处理器

应用领域：数据库

【Abstract】

Smart SSD将CPU处理能力与DRAM存储能力共同包含在内，并使得在其中运行用户程序成为可能。（SmartSSD具体见论文Enabling Cost-effective Data Processing with Smart SSD）

本文旨在研究将上述功能运用于相关的解析查询处理（其作用是：通过各级处理缓存，将数据从主存中移出）之中所需面临的机会和挑战。

研究表明，将所选的查询处理原件植入Smart SSD中能够获得客观的性能提升和能耗下降。同时，本文还为SSD设备制造商确定了一些改变，以使其能够更好的在数据处理应用中利用Smart SSD；本文同时也为数据库界提供了可能的研究机会。

【Introduction】

将程序代码移近数据相较于将数据移近程序代码来说是更加有效的。早期硬件边界严格，为使数据流更高效，常应用预取等机制。现今存储与处理边界愈加模糊，故提出Smart SSD的想法。

Smart SSD的机会：

① SSD的内部带宽远远大于IO接口（SAS/SATA）所支持的带宽，且这一差距将越来越大，这意味着将程序代码移近数据将带来性能的提升。目前所应用的技术均是在IO子系统上附加处理硬件，而Smart SSD是直接在SSD内部包含有处理部件。

② 若在Smart SSD内部进行计算，则会降低主机端的处理能耗（换句话说，这提高了服务器/应用有效计算所消耗的能耗占比），由此，使用Smart SSD的服务器可以使用价格较低的ARM作为处理器，同时收获价格及性能的优待。

③ 通过降低查询处理的运行时间（代码靠近数据，运行时间自然会短）或在Smart SSD中能耗较低的处理器中运行查询处理，能够有效降低应用能耗。

文章在Samsung Smart SSD上扩展Microsoft SQL Server，编写了简单的选择和聚合操作符并将其编译到SSD的固件中，还扩展了SQL Server的执行框架，以开发一个简单(但功能有限)的工作原型，使其可以运行简单的选择和聚合查询。实验结果表明，性能提升2.7x，能耗节省3.0x。同时也揭示了以下挑战：文章添加到Smart SSD中的处理功能还太简略（更多的功能添加还需硬件制造商和软件开发商的进一步考察认可，本文或可提供有效参考）；支持Smart SSD的固件开发流程需随之更新；DBMS内部必须进行修改，以便在生产环境中使用Smart SSD。（挑战与前景相见第六章第二段）

【Smart SSD】

Communication Protocol 指令传递

为使处理功能能够得以运行，需要以会话为基础的协议，其包含3个指令：OPEN/CLOSE/GET。

OPEN/CLOSE：会话以OPEN开始，以CLOSE结束。会话开始后，运行用户程序所需的运行时资源（如线程和存储空间）会被授权，且会返回给主机一个唯一的会话ID；CLOSE关闭会话后，所有程序终止运行且其所用资源均被释放。需要注意的是，OPEN后，再使用GET/CLOSE指令时，需根据会话ID进行匹配，匹配成功后该指令才可执行。

GET：主机检测程序状态，且若有输出的结果，也同时进行检索。通过各自不同的会话ID，多个用户程序可并行执行。需要注意的是，若无可用资源，该程序可能被阻塞。

API 指令/线程/内存/数据管理

指令API：在协议中各指令传递至设备端时，Smart SSD运行时系统唤醒由用户程序注册的对应的回调函数以执行相应功能。

线程API：会话开启时即创造所需线程，指令到达后才会排定线程（线程是非先发的）。线程一旦被排定，其上便会唤醒相应的回调函数。

内存API：会话开启时会为会话分配预定份额的内存空间，会话结束时将其收回。

数据API：多个数据页可并行由flash上载至DRAM。一旦上载至DRAM即被固定，待处理后才会被释放，以储备足够后续操作进行的内存空间（否则后续操作会被阻塞）。

Summary：

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。

后附另一篇Smart SSD上的应用：（list intersection）

SSD In-Storage Computing for List Intersection