**Biscuit: A Framework for Near-Data Processing of Big Data Workloads**

主机处理器➕固态硬盘内嵌入式处理器

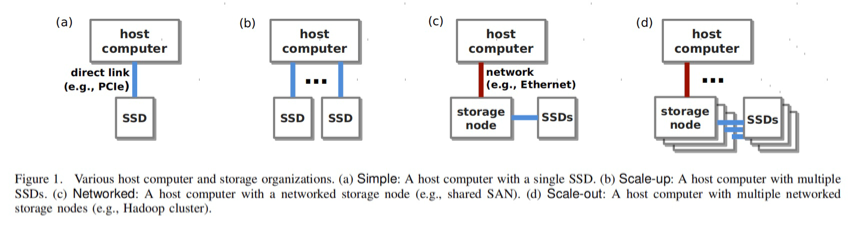
（也有用到专用硬件加速器（即硬件模式匹配器），但主体由嵌入式处理器实现）

应用领域：一般性（计算功能可由用户编程）数据密集型应用

数据紧凑型查询所需检查的数据量是很庞大的，一个提速方法是，减少由存储网络传输到主机系统的数据量，这可以通过过滤存储中的外来数据做到，这是一种近数据处理（NDP）的形式。

早期的NDP工作存在以下限制：1）主要聚焦于NDP概念的证明，很少关注到可用于全数据处理的实践框架的设计和实现；2）其中用到的硬件已经过时，所得数据也不再适用。

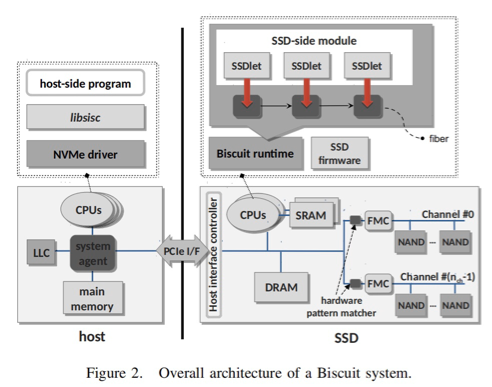
本文提出了用户可编程的NDP框架Biscuit。



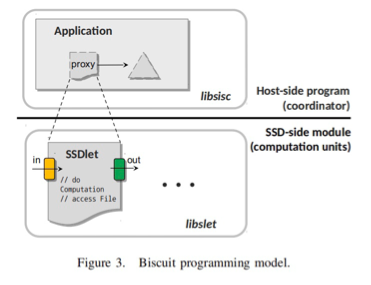
NDP系统组织有图1中的两类四种，各种结构，其主机均可将工作卸载给存储节点，存储节点也可将工作卸载给其本地SSD(s)。

Biscuit结构可以是上述四种中的任一种。

图2是Biscuit系统的总体结构。

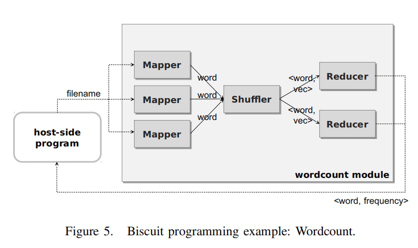


Biscuit借助SSD runtime和库提供的一组高级API使得主机和SSD能够共同分担执行应用任务。SSD runtime后续介绍，API则整体反应了编程模型，下面将详细介绍。

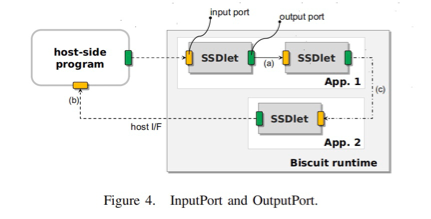
 图3是编程模型，其分为SSD端的计算模型和主机端的协作模型。计算模型使用输入输出数据描述了任务的计算功能，协作模型创建管理任务（任务执行的协调）并建立数据来源（生产者）和数据接受者（消费者）的联系。

计算模型和协作模型分别由库libslet和库libsisc实现并向用户提供了相应接口，实现了用户编程性。库libslet被用于SSD端计算模块的建立，库libsics除前述协作模型的功能代码外，还包括主机端计算的代码。

库libslet和库libsisc均包含有SSDLet类，SSDLet类表述有SSDlet，这是一个执行单元。库libslet中的SSDLet类用于SSDlet执行单元的定义（不同SSDlet执行单元的计算方式（即执行方式）不同，即其定义代码就不同，当主机端程序指令其开始执行后，其中的核心算法被调用），库libsisc中的SSDLet类定义面向SSD端SSDlet执行单元实例的接口。



一组协同运作的SSDlet执行单元组成一个应用，如下图所示（Mapper，Shuffler和Reducer三个SSDlet执行单元组成Wordcount应用）。主机端通过建立其中连接并命令其开始执行以协调各SSDlet执行单元的操作。



主机端协作模块和SSD端的计算模块以及计算模块中的SSDlet执行单元实例（计算模块建立后，其中的执行单元也随之建立）都建立好后，其间的数据是如何传递的呢？

数据传递是通过输入输出端口实现的，如图4所示。

另外，Biscuit要求读写数据操作要在文件系统下完成。文件由库libslet和库libsisc中均包含的File类表述。主机端程序使用库libsisc中的File类创建文件实例后通过引用参数或端口将该文件传递给SSD端。SSD端使用库libslet中的File类即可读写已被传递的文件实例。

File类支持异步操作，提高了性能。

下面介绍SSD runtime，其传达对于SSD资源的访问并全局性的控制架构中发生的事件。Biscuit的主要特点有支持多进程、高效数据传递、动态模块上载和动态内存分配。

1）**支持多进程：**在主机端的请求下，runtime上载SSD端计算模块、实例化SSDlet并协调执行（主机端写协作模块的执行协调功能是借助runtime完成的）。其支持多个SSDlet执行单元同时执行，即支持多进程。对于支持多核而言，是支持多应用同时执行（为硬件实现的简便和安全性，需要保证同一应用的SSDlet执行单元于同一处理器下执行）。

**2）高效数据传递：**使用有界队列实现输入输出端口，传递接收数据即可简单的通过入队列和出队列来实现。

**3）动态模块上载：**SSDlet执行单元必须被编译连接到SSD端的计算模块中，库libslet是连接runtime和SSDlet执行单元实例的桥梁。SSDlet执行单元实例建立后，其从runtime获取功能表以执行任务。

硬件上添加了一个硬件模式匹配器（Hardware Pattern Matcher）以实现对IO的过滤与匹配。

**Summary：**

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。