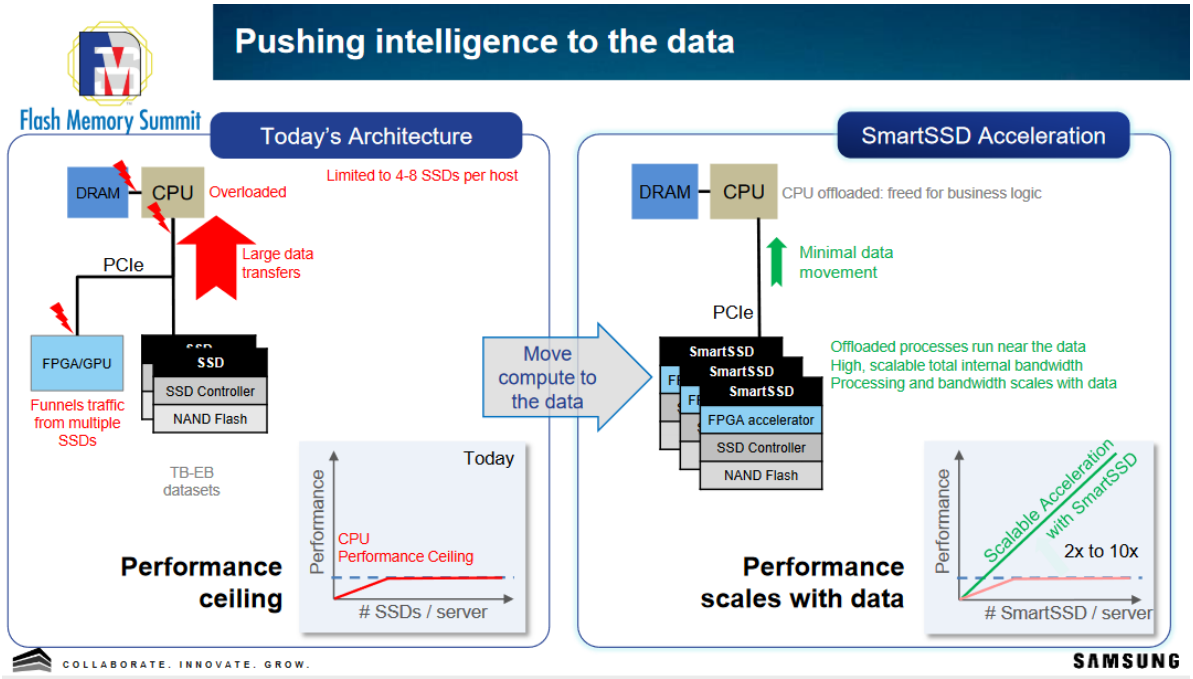
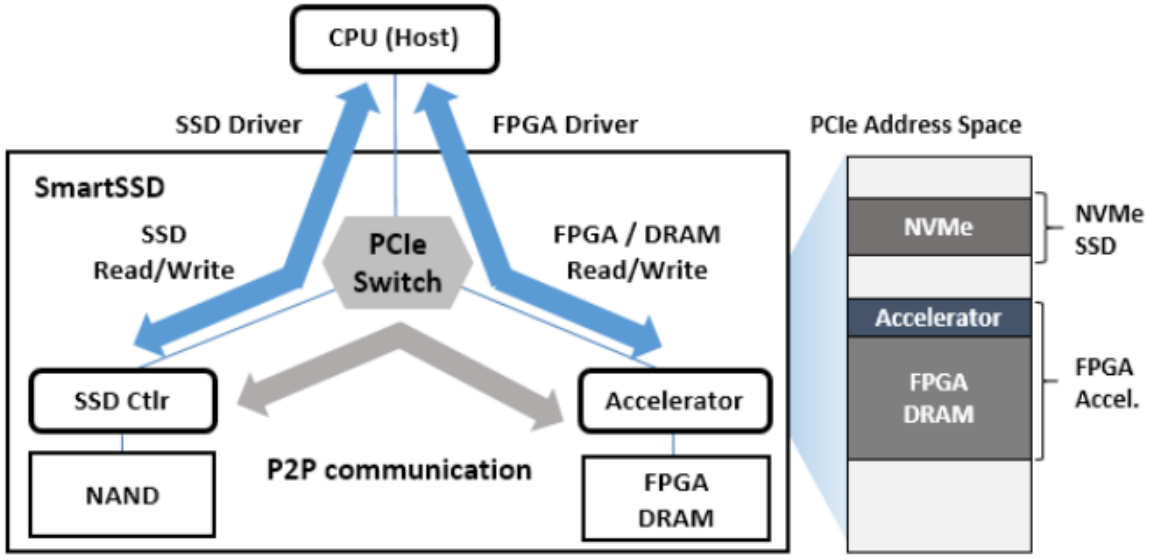
SmartSSD调研

Presentation



The primary advantage of using a computing storage device is the direct integration of the SSD and FPGA within a compact device. This integration allows a large amount of data transfer without going through the long PCIe lanes in the main board of the system. This physical advantage reduces the power consumed in the entire system, and it makes the whole system more energy-efficient.

SmartSSD platform



组成：**SSD controller, NAND array, FPGA accelerator, FPGA DRAM and PCIe switch**

SSD： **4TB PM1733**

SmartSSD和主机之间的数据通路：**PCIe Gen3 x4**

FPGA加速器： **Xilinx Kintex FPGA (KU15P)**:

- LUTs / FFs (K): 523 / 1045
- BRAM / URAM (Mb): 34.6 / 36
- DSP slices: 1968

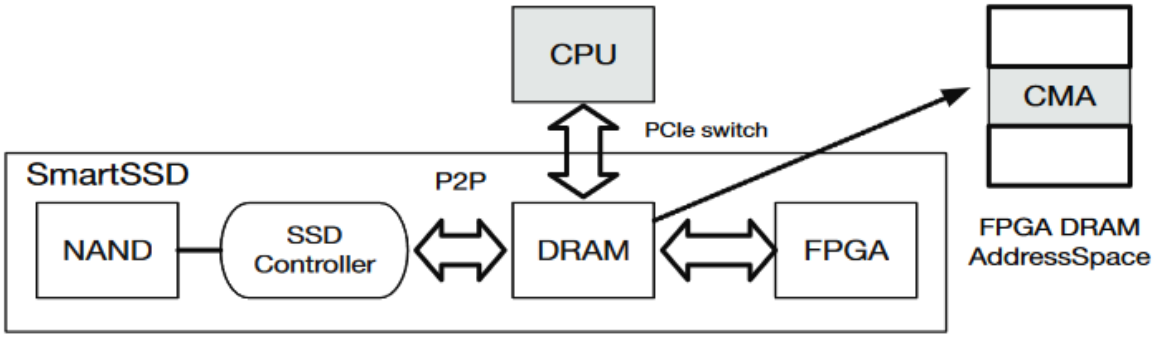
CPU可以通过SSD驱动向SSD控制器发出SSD读写请求，也可以通过FPGA驱动发出FPGA计算请求和FPGA DRAM读写请求。

P2P：使用FPGA DRAM和板载PCIe Switch，SmartSSD支持NVMe SSD和FPGA之间内部数据路径的数据移动。

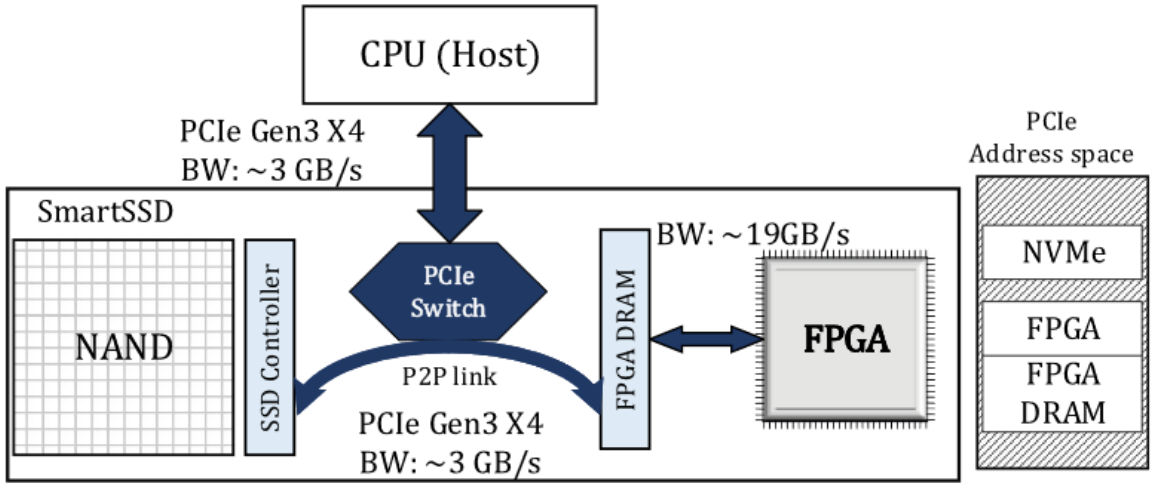
FPGA DRAM被暴露在硬件FPGA内核以及公共内存区，公共内存作为PCIe基本地址寄存器（BAR）暴露在主机地址空间，可以被NVMe SSD访问，用于P2P数据传输。

P2P实现了近存储的数据计算，从而减少主机-SSD和主机-FPGA的PCIe流量及往返延迟。

SmartSSD支持OpenCL编程模型，使用OpenCL API进行内核启动、内存分配和数据传输，FPGA内核通过HLS或RTL实现。



SmartSSD Architecteture



FPGA直接与存储阵列NAND进行通信，即所谓的点对点（P2P）数据传输，消除SSD到CPU、CPU到FPGA之间不必要的往返流量，实现近存储计算。

SmartSSD上的DRAM可以被FPGA访问，在DRAM中有一个特定的内存区域，称为common memory area(CMA)，FPGA和CPU主机都可以访问。

CMA区域用于在存储部件和FPGA之间直接传输数据，虽然CPU主机不参与从SSD到FPGA的点对点传输的数据移动，但是它启动了数据传输。

此外，SmartSSD上的主机和FPGA可以通过将CMA区域映射到主机的地址空间来与CPU通信。

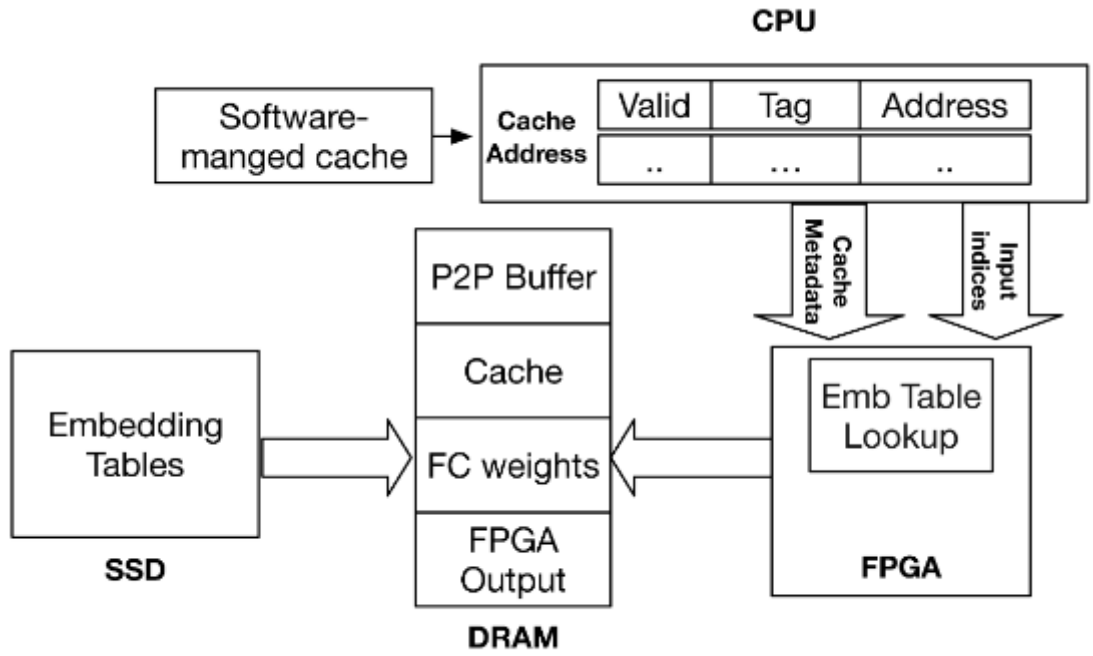
计算卸载到SmartSSD上的FPGA，FPGA的计算结果可以直接写回SSD或CPU可访问的主存储器。

Software-managed Cache for SmartSSD

虽然SSD增加了存储能力，但SSD比DRAM内存有更高的延迟和更低的带宽，降低了性能。

在基于推荐系统的近存储计算中，为SmartSSD设计一种新型缓存技术，使用FPGA的DRAM来缓存SSD上的一些频繁访问的数据。

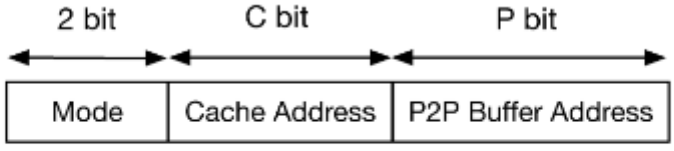
允许FPGA上的内核使用其外部DRAM作为SSD上数据的缓存，缓存是由CPU上的主机管理的，FPGA上的内核使用主机提供的信息来定位其外部DRAM中的数据。



(a) The overall architecture of SmartRec for SSD-based embedding table operations.

将DRAM分为四个区域。第一个区域通过P2P传输从SSD带入DRAM的数据（P2P缓冲区）；第二个区域用于缓存，在DRAM中保留一些之前获取的块；第三个层存储模型中的权重等数值；第四个区域缓存FPGA的输出。

P2P区域应该足够大，考虑如果没有一个访问在高速缓存中命中的话，P2P区域要容纳一个批次大小所需的全部数据。



(b) The format of the metadata provided by the CPU to the FPGA to manage the blocks in the cache

这种通过DRAM设计的缓存是由主机CPU管理的，因此是一种软件管理的缓存。当数据不在缓存中时，CPU从SSD到DRAM发出一个P2P读取。主机针对访问向FPGA发送元数据，以传达关于每个访问的位置的信息。FPGA使用缓存地址和缓冲区来定位数据，取决于模式位：

- Mode 0：数据块不存在于cache中。FPGA使用buffer address指定的地址来读取该块。此外FPGA在cache address提供的地址上保留该块。
- Mode 1：与Mode 0类似，只是FPGA不在cache中保留该块。
- Mode 3：数据存在于cache中，地址由cache address指定。

Scalability

在传统的数据库架构中，当系统中存在一个加速器时，要么主机从存储设备中读取数据并转移到加速器，要么加速器可以与存储设备进行P2P通信，直接从存储设备上读取数据。

在前一种情况下，数据应该从主机内存到达加速器内存(FPGA DRAM),通过主机传输的延迟要比加速器直接从存储设备中读取数据的延迟大的多。在P2P通信的情况下，不会占用主机资源进行数据传输，但是数据存储在多个存储设备中时，此架构受到性能扩展的影响。

PCIe是传统计算机系统中事实上的I/O互连，它提供了对存储设备有限的同时访问，当独立操作在不同的存储设备上被并行调用时，它限制了系统的可扩展性。

传统系统不能利用存储层面的并行性，因为它们不能同时提供对所有存储设备的访问，而SmartSSD设备阵列可以对每个设备中的数据独立操作：

- 以推荐系统为例，对不同嵌入表的操作可以独立进行。嵌入表可以存储在不同的SmartSSD上，操作由每个SmartSSD内部的FPGA并行执行，然后主机收集每个FPGA的最终输出。
- 以数据库排序为例，存储数据库需要多个存储设备，数据库管理系统将数据库划分为多个分区，并将操作分解为对分区数据库的多个独立操作。对每个SmartSSD中的数据提供独立的操作，不占用存储到主机的带宽。

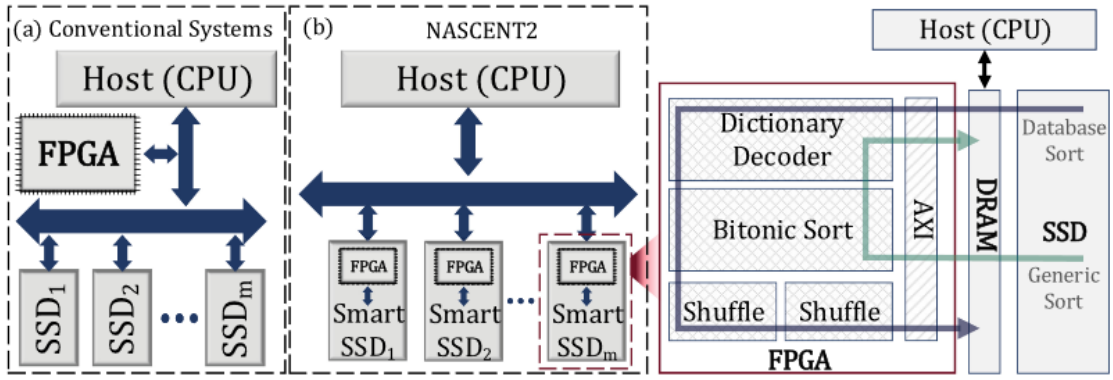


Fig. 3. The overall architecture of NASCENT2 (right) as compared to the conventional systems equipped with an FPGA accelerator (left). The blue arrow represents the data flow of the database sort, and the green arrow shows that of the generic sort.