

## The Fusion-io Difference

### THE PROBLEM

Most storage vendors today are incorporating SSDs behind controllers, replacing disk drives. This approach sacrifices most of the inherent speed and latency benefits of the raw NAND flash to slow storage protocols, network latency, and unnecessary layers of complexity.

Figure 1 shows all of the layers of abstraction and complexity in between the application and the underlying storage in a common SSD-enhanced storage architecture. The numbers represent context switches along the data path, which result in latency build-up, and act as barriers to application acceleration.

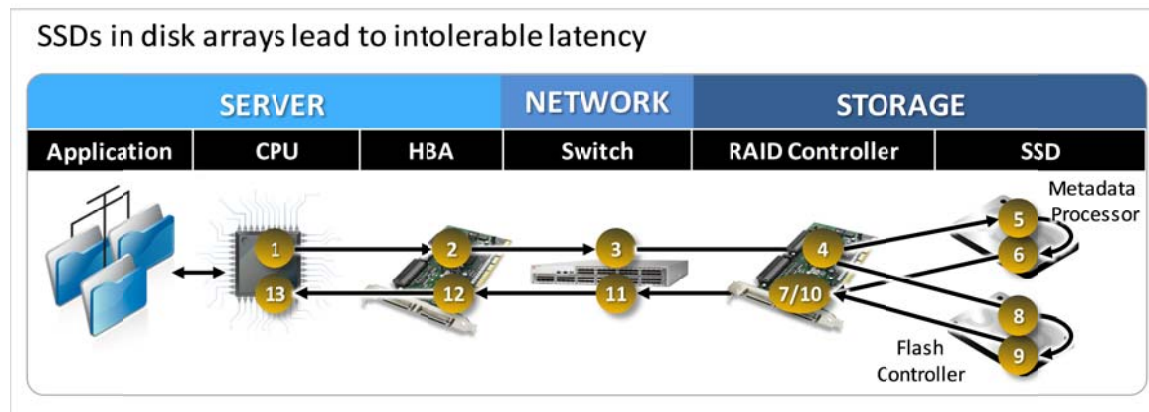


Figure 1

Fusion-io has a fundamental architectural advantage, which is a completely different and superior approach to flash integration than any storage vendor. Fusion-io presents flash to host systems as a new memory tier, integrating flash as close to the host CPU as possible, based on the most efficient, reliable, and scalable architecture.

FUSION-IO®

## ARCHITECTURAL DIFFERENCES

### 1) FORM FACTOR

Some vendors put flash on a printed circuit board form factor that can attach to the PCIe bus, rather than in hard drive slots.

While PCIe vendors have eliminated the storage and network abstraction layers (four context switches), the underlying embedded RAID controller still results in too many context switches and latency build-ups. Each additional SSD module in Figure 2 adds two additional context switches and PCIe SSD cards generally have between two and eight modules.

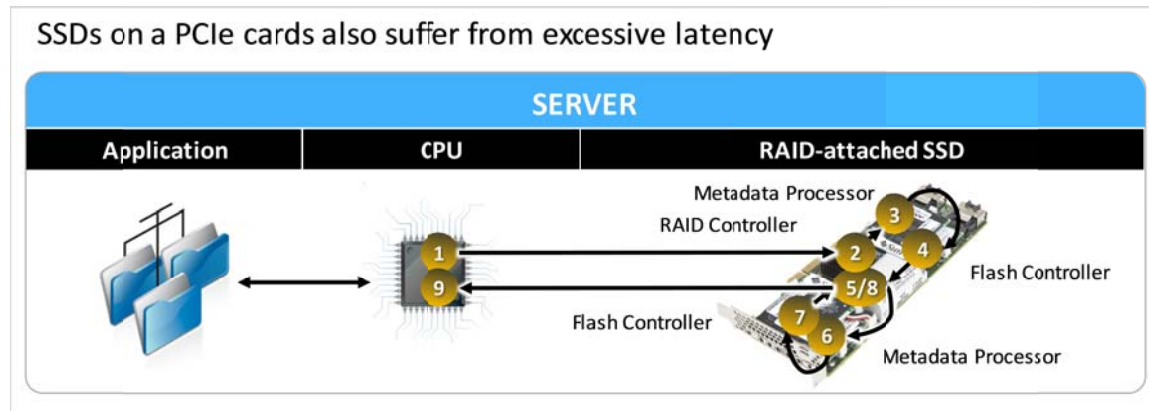


Figure 2

There is no performance advantage to this approach.

The similarity between Fusion-io and other imitation products stops here. The best description for these look-a-likes is a RAID controller with SSDs directly attached, which eliminates cabling and frees up drive bays. Even though these solutions are generally called PCIe SSDs, this is a misnomer, because it is still just a RAID controller with SSDs physically attached/integrated. These embedded SSDs communicate with the RAID card using native storage protocols on an internal bus, such as SAS or SATA. Bottom-line, the only advantage gained, for other vendors, by moving flash to a PCIe card is freeing up a few drive bays.

### 2) NO STORAGE PROTOCOLS

It is not enough to physically locate the flash on the system bus via a PCIe card. The storage bus and storage protocols themselves such as SATA and SAS remain unnecessary layers of complexity which add latency in between the host processor and the underlying data. Fusion-io has removed these bottlenecks, enabling the host to directly access the flash using direct memory access (DMA), instead of going through some mediator like an HBA or RAID controller.

These storage protocols and controllers were originally designed to accommodate slower, rotating, mechanical disk drives and are no longer relevant in the world of flash media.

### 3) USE THE HOST CPU LIKE RAM

After eliminating the storage bus and protocols, one physical barrier to realizing the full performance potential of NAND flash remains: the embedded processor. Off-loading to embedded processors is an architecture built to accommodate slower, rotating, mechanical disk drives. Traditional storage operates this way to free host CPUs from having to wait for I/O requests from slow disk drives. NAND flash is fast enough to avoid this pain point. Putting an embedded processor on NAND flash would be akin to using embedded processors on DRAM modules—it would only slow down the raw speed of the underlying medium.

By using embedded processors, other flash integration methods are actually re-creating the performance gap between processors and storage. Putting flash behind an infrastructure designed for hard drives is itself a

FUSION-io®

bottleneck preventing flash from keeping pace with Moore's law, wasting the full potential of the underlying flash medium. Fusion-io removes embedded processors used for off-loading, and instead uses the host CPU in the same way as the virtual memory subsystem — for caching metadata tables so the host can access them independently.

#### 4) VIRTUALIZE THE STORAGE LAYER

Today, SSD vendors are running on operating systems that use a 30 year old subsystem designed for disk drives, which unfortunately can't take advantage of the capabilities of flash. For this reason, Fusion-io developed the Virtual Storage Layer (VSL), a flash-optimized OS subsystem software layer. VSL allows the flash to interact independently with each host CPU core, in parallel, as a new tier of memory, similar to L1, L2, L3 or in DRAM. VSL simultaneously presents the flash to the application layer as a normal block-based storage device so that existing applications do not require modification.

The impact of treating NAND flash as a new memory tier leads to only two context switches, regardless of how many ioMemory modules are aggregated together. This is why Fusion-io is able to accelerate throughput in a linear fashion without latency build-ups preventing application acceleration.

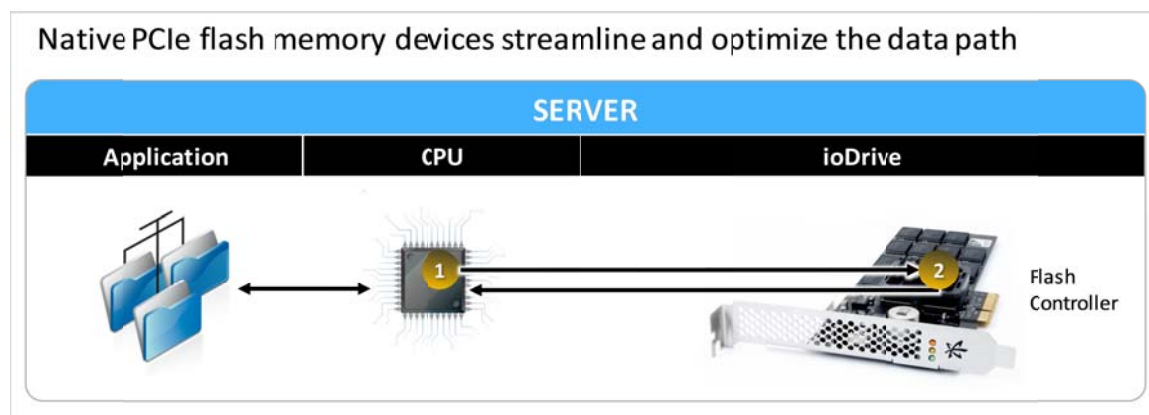


Figure 3

## Summary

Following these steps, Fusion-io integrates flash as a new tier of memory with greater density than RAM and orders of magnitude performance increases over disk. This new memory tier is nearly as fast as memory, non-volatile, with orders of magnitude more capacity for far less power consumption and cost. When compared to storage systems, with or without SSDs, Fusion-io has a dramatic latency advantage driven by fewer context switches and a parallel construct eliminating bottlenecks. Without this architecture, other solutions can never achieve the same levels of application acceleration and infrastructure consolidation.

#### FOR MORE INFORMATION:

Visit <http://www.fusionio.com>

To speak with a Fusion-io specialist in the U.S. call: +1 (801) 424 5500

To speak with a Fusion-io specialist outside the U.S.: <http://fusionio.com/company/contact>

WPCH031511

FUSION-IO®