Evolution of low-cost RAID requirements

RAID is a form of virtualization of hard drives, in which the client sees a single drive despite the multiplicity of physical drives. There are two primary reasons for using RAID: performance and reliability. By reading from multiple drives, RAID can mask the seek time delays for each physical drive. With RAID, the speed of the I/O technology and IOP efficiency, rather than the access speed of individual drives, binds the overall storage system performance.

From a reliability standpoint, RAID allows data to be replicated across multiple physical drives, thereby reducing the single-point-of-failure risks associated with conventional storage.

RAID implementations started in servers, enclosures attached to servers, and large storage arrays attached to mainframes using the SCSI protocol. Because it is a shared bus technology, SCSI made RAID possible by allowing connection of multiple disk drives to a single compute node. Given the difficulty of creating a Storage Area Network (SAN) out of SCSI, first-generation RAID storage typically was behind either servers or mainframes. The Fibre Channel protocol created a network underneath SCSI, enabling the development of SANs. As SAN technology has matured, a new standard called iSCSI is also bringing block storage into TCP/IP networks, including Ethernet LANs and the Internet. The new SAS standard is supporting a SAN based on the SCSI protocol over a simpler topology than Fibre Channel. Meanwhile, the SAS standard is enabling SATA drives to work with the SAS protocol so that low cost drives can be attached to SANs.

Design approaches for embedded RAID solutions

RAID comes in various forms (RAID 1, RAID 0, RAID 5, and RAID 6). RAID 1 mirrors data on one drive on another drive.
This is the simplest form of RAID and the most costly because it takes two drives to store one drive worth of data. RAID 0 stripes data across multiple drives, which boosts performance, but doesn’t provide fault tolerance. RAID 5 goes the next step by maintaining a parity byte for every group of data bytes. RAID 6 extends the parity concept. In addition to the normal parity (P), RAID 6 computes a second parity (Q), which allows data to be reconstituted even if two drives fail.

Note that there are two types of computations:

1. Creating tasks for drives from the virtual task that is targeting a single virtual drive
2. Performing the parity computations

The first computation is complex and therefore belongs in a processor; however, the second can be streamlined and thus implemented efficiently in hardware. The basic approaches to RAID implementation include:

- Software solutions
- External hardware solutions
- IOP hardware accelerated solutions

In the software solution, the processor on the motherboard performs both parsing the tasks for drives and computing the parities. This allows for the use of general-purpose hardware, but burdens the processor with a relatively high level of complex software overhead. Figure 1 shows the steps of software-based RAID 5 parity calculation. An external DMA in the I/O controller fetches the results into the hard drives. This approach, called RAID On Mother Board (ROMB), takes more processor, bus, and DRAM steps and also keeps the CPU on the motherboard busy.

In the external hardware accelerator approach, an embedded CPU performs the task-parsing calculations and hands off the parity calculations to the hardware accelerator. This approach overcomes the performance penalty, but can impose a significant cost penalty for the additional hardware.

The RAID-enabled IOP approach provides a more optimal solution. In addition to being a programming facility with one or more embedded processors, it includes acceleration hardware necessary for RAID computations. As shown in Figure 2, an IOP with integrated RAID 6 hardware-assist capabilities for computing dual parity (P + Q) can manage multiple stripes across multiple physical drives efficiently, while appearing to the host as a single unified storage system.

To complete the controller solution, one needs to attach an I/O uplink and one (or more) I/O downlink(s) to the processor’s PCI Express ports. For example, AMCC’s PowerPC440SPe has three PCI Express ports (one x8 and two x4) and a PCI-X port. This configuration provides flexibility for a connection to the host processor or an attachment of an Ethernet device for iSCSI uplink, SAS/SATA controllers for the downlink drives, and in some cases an interprocessor connection for redundancy between two controllers.

The IOP approach using a PowerPC440SPe offers a choice of two exclusive OR (XOR) engines for computing the RAID 5 parity and multiple Galois Field (GF)-based polynomials for performing the P and Q parity calculations for RAID 6, as well as various alternatives for balancing load and performance. The integrated
memory controller is dual-ported and supports up to 16 GB of 64-bit DDR667 SDRAM. To support the high-bandwidth PCI Express interfaces and memory controller, the processor local bus features a nonblocking two-way crossbar structure. The Low Latency (LL) segment implements smaller data buffers and is optimized for LL access, while the High Bandwidth (HB) segment has larger data buffers and is optimized for maximum throughput (see Figure 3).

Summary
Demand for RAID-based reliability and performance to extend into a widening range of network-based, higher performance, and lower cost storage systems has thrust the need for embedded RAID control solutions to the forefront. Faced with these trends, designers cannot afford the performance penalty of software-based solutions or the cost penalty of external hardware accelerators. The use of highly integrated IOP devices optimized for high data throughput and RAID parity calculations and nonblocking access to standards-based I/O interfaces offers the path toward meeting stringent demands for high performance and reliability at affordable costs.

John Fakiris is a senior marketing manager with AMCC’s Integrated Communications Products group. He previously held positions with IBM Microelectronics Embedded Products and Network Processor groups, and Harris Semiconductor Wireless Products. John holds BSEE and MSEE degrees from Purdue University.

Haluk Aytac is a senior solutions architect at AMCC currently concentrating on storage. He has worked at National Semiconductor, 3Taus, Hewlett-Packard, and Velio in various engineering and marketing roles. He holds three patents and has an MA in Physics from UC Berkeley.

To learn more, contact John at:

AMCC
4000 Centre Green Way, Suite 200
Cary, NC 27513
Tel: 919-678-4544 • Fax: 919-678-4501
E-mail: jfakiris@amcc.com
Website: www.amcc.com