

BASICS OF RAID

A couple of the recent Tech Tips have made mention of RAID, but the level of detail required in those tips didn't shed much light on what RAID actually is. The number of e-mail responses and comments in the Readers Digress section was convincing enough that an introduction to the basics of RAID would be an appropriate Tech Tip, so here it is.

Introduction

The word RAID sounds like it might describe something Marines conduct in Fallujah, or a can of what all roaches fear, but it is simply an acronym that stands for Redundant Array of Independent (or Inexpensive) Disks. Depending on who you talk to, the letter "I" can stand for either independent or inexpensive, but in my opinion independent is more appropriate, and far less subjective.



RAID generally allows data to be written to multiple hard disk drives so that a failure of any one drive in the array does not result in the loss of any data, as

well as increasing the system's fault tolerance. I say RAID generally does this, as there are several RAID configurations that provide different approaches to redundancy, but some RAID configurations are not redundant at all. Fault tolerance refers to a system's ability to continue operating when presented with a hardware (or software) failure, as should be experienced when a hard drive fails in one of the redundant configurations of RAID.

The Hardware



The basic hardware required to run RAID includes a set of matched hard drives and a RAID controller.

RAID can be run on any type of hard drive, including SCSI, SATA, and ATA. The number of hard drives required is dependent on the particular RAID

configuration chosen, as described later. I mention the need for matched hard drives, and although this is not absolutely necessary, it is recommended. Most arrays will only be able to use the capacity of the smallest drive, so if a 250GB Hitachi drive is added to a RAID configuration with an 80GB Hitachi drive, that extra 170GB would probably go to waste the only time that this doesn't

apply is in a RAID configuration called JBOD "Just a Bunch Of Disks"; which really "isn't a RAID configuration" but just a convenient thing that a RAID controller can do – see “Basic RAID Configurations” below for more information. In addition to matching capacities, it is highly recommended that drives match in terms of speed and transfer rate as the performance of the array would be restricted by the weakest drive used. One more area that should be considered while matching is the type of hard drive. RAID controllers are generally for either SCSI, SATA, or ATA exclusively, although some systems

allow RAID arrays to be operated across controllers of different formats.



The RAID controller is where the data cables

from the hard drives are connected, and conducts all of the

processing of the data, like the typical drive connections found on a

motherboard. RAID controllers are available as add on cards, such as

this Silicon Image PCI ATA RAID controller, or integrated into motherboards,

such as the SATA RAID controller found on the Asus K8V SE

Deluxe. Motherboards that include RAID controllers can be operated without

the use of RAID, but the integration is a nice feature to have if RAID is a consideration. Even for systems without onboard RAID, the relatively low cost of add on cards makes this part of the upgrade relatively pain free.

Another piece of hardware that is not required, but may prove useful in a RAID array is a hot swappable drive bay. It allows a failed hard drive to be removed from a live system by simply unlocking the bay and sliding the drive cage out of the case. A new drive can then be slid in, locked into place, and the system won't skip a beat. This is typically seen on SCSI RAID arrays, but some IDE RAID cards will also allow this such as this product manufactured by Promise Technology.

The Software

RAID can be run on any modern operating system provided

that the appropriate drivers are available from the RAID

controller's manufacturer. A computer with the operating

system and all of the software already installed on one drive can be easily be

cloned to another single drive by using software like Norton Ghost. But it is

not as easy when going to RAID, as a user who wants to have their existing



system with a single bootable hard drive upgraded to RAID must start from the beginning. This implies that the operating system and all software needs to be re-installed from scratch, and all key data must be backed up to be restored on the new RAID array.

If a RAID array is desired in a system for use as storage, but not as the location for the operating system, things get much easier. The existing hard drive can remain intact, and the necessary configuration can be made to add the RAID array without starting from scratch.

Basic RAID Configurations

There are about a dozen different types of RAID that I know of, and I will describe five of the more typical configurations, and usually offered on RAID controller cards.

RAID-0 is one of the configurations that does not provide redundancy, making it arguably not a true RAID array. Using at least two disks, RAID 0 writes data to the two drives in an alternating fashion, referred to as striping. If you had 8 chunks of data, for example, chunk 1, 3, 5, and 7 would be written to the first drive, and chunk 2, 4, 6, and 8 would be written to the second drive,

but all in sequential order. This process of splitting the data across drives allows for a theoretical performance boost of up to double the speed of a single hard drive, but real world results will generally not be nearly that good. Since all data is not written to each disk, the failure of any one drive in the array generally results in a complete loss of data. RAID 0 is good for people who need to access large files quickly, or just demand high performance across the board (i.e. gaming systems). The capacity of a RAID 0 array is equal to the sum of the individual drives. So, if two 160GB Seagate drives were in a RAID 0 array, the total capacity would be

RAID-1 320GB.

is one of the most basic arrays that provides redundancy. Using at least two hard drives, all data is written to both drives in a method referred to as mirroring. Each drive's contents are identical to each other, so if one drive fails, the system could continue operating on the remaining good drive, making it an ideal choice for those who value their data. There is no performance increase as in RAID 0, and in fact there may be a slight decrease compared to a single drive system as the data is processed and written to both

drives. The capacity of a RAID 1 array is equal to half the capacity of the sum of individual drives. Using those same two 160GB Seagate drives from above in RAID 1 would result in a total capacity of 160GB.

Raid is the name may imply, is a combination of RAID 0 and RAID 1. You have the best of both worlds, the performance boost of RAID 0 and the redundancy of RAID 1. A minimum of four drives is required to implement RAID 0+1, where all data is written in both a mirrored and striped fashion to the four drives. Using the 8 chunks of data from the example above, the write pattern would be something like this... Chunks 1, 3, 5, and 7 would be written to drives one and three, and chunks 2, 4, 6, and 8 would be written to drives two and four, again in a sequential manner. If one drive should fail, the system and data are still intact. The capacity of a RAID 0+1 array is equal to half the total capacity of the individual drives. So, using four of the 160 GB Seagate drives results in a total capacity of 320GB when configured in RAID 0+1.

RAID-5 may be the most powerful RAID configuration for the typical user, with three (or five) disks required. Data is striped across all drives in the array, and in addition, parity information is striped as well. This parity information is basically a check on the data being written, so even though all data is not being written to all the drives in the array, the parity information can be used to reconstruct a lost drive in case of failure. Perhaps a bit difficult to describe, so let's go back to the example of the 8 chunks of data now being written to 3 drives in a RAID 5 array. Chunks one and two would be written to drive one and two respectively, with a corresponding parity chunk being written to drive three. Chunks three and four would then be written to drives one and three respectively, with the corresponding parity chunk being written to drive two. Chunks five and six would be written to drives two and three, with the corresponding parity chunk being written to drive one. Chunks seven and eight take us back to the beginning with the data being written to drives one and two, and the parity chunk being written to drive three. It might not sound like it, but due to the parity information being written to the drive not containing that specific bits of information, there is full redundancy. The capacity of a RAID 5 array is equal to the sum of the capacities of all the

drives used, minus one drive. So, using three of the 160GB Seagate drives, the total capacity is 320GB when configured in RAID 5.

JBOD is another non-redundant configuration, which does not really offer a true RAID array. JBOD stands for Just a Bunch Of Disks (or Drives), and that is basically all that it is. RAID controllers that support JBOD allow users to ignore the RAID functions available and simply attach drives as they would to a standard drive controller. No redundancy, no performance boost, just additional connections for adding more drives to a system. A smart thing that JBOD does is that it can treat the odd sized drives as if they are a single volume (thus a 10GB drive and a 30GB would be seen as a single 40GB drive), so it is good to use if you have a bunch of odd sized drives sitting around – but otherwise it is better to go with a RAID 0, 1 or 0+1 configuration to get the performance boost, redundancy or both.

Source: <http://www.geeks.com/techtips/2005/techtips-012705.htm>