Fault Tolerance & Reliability  
CDA 5140

Chapter 3 – RAID & Sample Commercial FT Systems

- basic concept in these, as with codes, is redundancy to allow system to continue operation even if some components fail

- **hot standby** refers to components that can fail, being associated with components that run in parallel (i.e. powered up) and can take over upon a failure

- **cold standby** refers to having just one component powered up and others available but powered down (standby) that can be powered up upon detection of failure

- once one standby takes over upon failure, either repair the failed unit or replace it, before the new operating component can fail

- various techniques for improving system/component reliability:
  - improve manufacturing or design process to decrease component failure rate
  - parallel redundancy – sometimes synchronization, if multiple parallel components, can be difficult
  - cold standby can require complex switching
  - stage of repairing or replacing components is important

### RAID

- increase of performance for processors & main memory has received considerable attention, unlike secondary storage

- secondary storage is primarily a mechanical process & hence performance can only be pushed so far without using multiple parallel components

- RAID, redundant arrays of independent disks, provides arrays of disks that can operate independently and in parallel, and is an industry standard so can be used on different platforms and operating systems

- if data is on separate disks, then I/O requests can be done in parallel
levels of RAID do not imply a hierarchy, but simply different design architectures with the following 3 characteristics:

- set of physical disk drives viewed by operating system as single logical drive
- data are across physical drives of the array
- redundancy introduced by parity information in order to recover data upon a disk failure

- note that RAID 0 does not include the redundancy characteristic

- also, other levels of RAID have been introduced by researchers but are not part of the industry standard

- another important point is that by introducing multiple disks, the overall probability of failure is increased, but this is compensated for by adding in the parity which allows data to be recovered upon a disk failure

- bandwidth for disk I/O is considered the reciprocal of the read-time, so if data broken into “chunks” (amount varies with RAID level) and read/written in parallel chunks then the effective bandwidth increases

- all user and system data viewed as being stored on a logical disk, and this data is divided into strips which can be physical blocks, sectors, or some other unit

- a stripe then is considered to be a set of logically consecutive strips that are mapped one strip to each array member, and this operation is referred to as striping

- thus, for an n-disk array, the first n logical strips are stored as the first strip on each of n disks to form the first stripe, etc.

- Table 3.6 gives a summary of the RAID approaches

Raid Level 0

- no redundancy included so not a true member of RAID, but striping is used on multiple disks

- reliability is less than if no additional disks used – why?
- rarely used except for example, on supercomputers where performance and capacity are important as well as low cost, and reliability is not as crucial

- are still advantages over single large disk, since if data are distributed over the array and have two I/O requests for different blocks, can be done in parallel

**RAID Level 1**

- redundancy is through complete replication of the data, still striped, over multiple disks, unlike the rest of the RAID levels which all use some form of parity check

- still, several advantages to this arrangement:
  
  o reads can be serviced by either disk that contains required data, in particular the one with shortest seek time plus rotational latency

  o writes require that both disk locations be updated but this can be done in parallel and the time is determined by the larger of the seek time plus rotational latency

  o error recovery is simple in that when one drive fails, data is taken from the second

- main disadvantage of RAID 1 is the cost, thus is only used when storing highly critical files such as system software and data

- for such critical data, this arrangement particularly good as the other data is immediately available

- in transaction environments, if most requests are reads, can have almost double the performance of RAID 0, but no improvement for writes as they must go to both disks

- mirrored disks can significantly increase the cost of a system

- if one disk fails, it can be repaired/replaced while system continues working with the functioning disk

- in following levels, there are differences in type of error detection, size of the “chunk” and pattern of striping
RAID Level 2

- uses Hamming error-correcting codes to both detect and correct errors, i.e. SECSED or SECDED

- error-correcting codes added to “chunks” of data and striped across the disks

- in parallel access array, all member disks must participate in every I/O request, so typically spindles of individualized drives synchronized so disk heads are in same position on each disk

- strips are very small – byte or word, typically

- error-correcting code is calculated across corresponding bits on each data disk, and then corresponding check bits stored on multiple parity disks

- requires fewer disks than Level 1, but still costly

- on single read, all disks accessed simultaneously, and data and parity checks given to array controller which can identify and correct single errors instantly, so read access not slowed

- on write, all data disks and parity disks must be accessed

- only really useful where many single disk errors occur, vs complete failure, and since individual disks and drives are highly reliable, this more than necessary and seldom used

RAID Level 3

- organized much like level 2, except only one parity drive

- parallel access with data in small strips as shown in Figure 3.17

- simple parity bit for set of individual data bits in same position on all data disks calculated using $\text{XOR}$ over the strips, as for example:

$$P(0-3) = \text{strip 0 XOR strip 1 XOR strip 2 XOR strip 3}$$
assume that Disk 2 fails corrupting strip 1 data and is replaced, and then the data is regenerated by adding P(0-3) to the non-corrupted disks as follows:

\[
\text{REGEN}(1) = P(0-3) \text{ XOR } \text{strip 0 XOR strip 2 XOR strip 3}
\]
\[
= (\text{strip 0 XOR strip 1 XOR strip 2 XOR strip 3}) \text{ XOR (strip 0 XOR strip2 XOR strip 3)}
\]
\[
= \text{strip 1}
\]

this same approach is used for levels 4 through 6 and this is accomplished “on the fly” using the XOR operation

return to full operation requires failed disk be replaced and entire contents of failed disk rebuilt

since data striped in very small strips, very high data transfer rates can be obtained since this involves parallel transfer of data

however only one I/O request can be executed at a time so for transaction oriented tasks performance is reduced

**RAID Level 4**

strips are at the block level and each disk operates independently so that separate I/O requests can be handled in parallel so good for situations where there are high I/O requests

not as applicable where there are high data transfer rates

bit-by-bit parity strip calculated across corresponding data strips

whenever there is a write, the array controller must not only update the data disk, but also the parity disk

consider that strip 1 is being updated and is represented now as strip 1'

the original parity is given as

\[
P(0-3) = \text{strip 0 XOR strip 1 XOR strip 2 XOR strip 3}
\]

and after the update as:
P(0-3)' = strip 0 XOR strip 1' XOR strip 2 XOR strip 3

= strip 0 XOR strip 1' XOR strip 2 XOR strip 3
XOR (strip 1 XOR strip 1)

= P(0-3) XOR strip 1 XOR strip 1'

- this implies that to update the parity strip, the old parity strip must be read and updated with the new and old data strip

- this implies for each update, two reads and two writes

- in large I/O writes, if all data disks are accessed, then the parity drive can be updated in parallel with the new data

- every update does involve the parity drive which can then be a bottleneck

**RAID Level 5**

- similar to level 4 except that the parity strips are across all disks in a round-robin style, repeating, for n disks, after n stripes

- avoids potential bottleneck noted for level 4

**RAID Level 6**

- there are two levels of parity with separate calculations with the result stored on separate disks

- thus for n data disks there are n+2 total disks

- one parity calculation is the same as for levels 4 and 5, and the second is a distinct parity calculation

- thus, two data disks can fail and the data can still be regenerated

- provides extremely high data availability, but an associated high penalty on writes since each one affects two other disks

- one model is to have a horizontal parity and a vertical parity
Representative Commercial Fault-tolerant Systems

Tandem Systems

- began in 1980s as “non-stop” computer systems which was very attractive to on-line transaction businesses such as airlines, banks, credit card companies etc.

- in 1997 Tandem was purchased by Compaq which maintained the emphasis on fault tolerance

- survey in 1999 indicated that 66% of credit card transactions, 95% of securities transactions, 80% of ATM transactions were processed by Tandem computers – indicating fault tolerance is a very important concept in the business world

- now called NonStop Himalaya computers

- in a 1985 survey determined that a typical well-managed transaction-processing system failed about once every two weeks for about an hour, which, in comparison to an auto, which requires one repair a year, is significant

- original objectives of Tandem were:
  
  o no single hardware failure should stop the system
  o hardware elements should be maintainable with on-line system
  o database integrity necessary
  o should be modularly extensible without changing software

- last point important so that companies can expand without taking whole system down

- original Tandem was combination of both hardware & software fault tolerance, i.e. hot standby, with parallel units for CPUs, disks, power supplies, controllers, etc.

- figure 3.19 demonstrates this with \( N \) processors where \( N \) is an even number between 2 and 16

- processor subsystem uses hardware and software fault tolerance to recover from processor failures

- operating system called Guardian which manages heartbeat signals which are sent to all other processors once a second to indicate it is ‘alive’
- if no heartbeat in 2 seconds then each operating processor goes into a system state called *regroup* to determine the unit(s) that failed

- try to avoid the *split-brain* situation where communication lost between two groups

- at end of regroup, each processor is aware of available system resources

- originally Tandem used all custom microprocessors and checking logic to test for hardware faults

- once hardware fault detected, heartbeat signal was no longer sent out & processors had to regroup

- software fault tolerance implemented in Guardian by having process pairs, a primary and a backup, on separate processors

- primary process sends checkpointing information to its backup as long as it is without failure, so that should it fail, the backup can take over

- checkpointing does not require much processing power so the "backup" is the primary for other activities

- also, software fault tolerance protects against transient software failures since backup re-executes an operation rather than simultaneously performing the same operation

- in 1997, S-series NonStop Himalaya servers were introduced which replaced the processor and I/O buses with a network architecture called ServerNet and included many new features, one of which is cyclic redundancy check (CRC) code which is a polynomial based code (see Figure 3.20)

- one of the new features was fault-tolerant power & cooling systems that are able to cool a whole cabinet if there is a failure of one system, and the working fans modify their power to compensate for a failed one

- similar workings for battery backups

- all software and hardware modules are self-checking so that it halts immediately on error so errors don’t propagate

- all memory has error-detection and correction codes to correct single-bit errors, detect double-bit errors and detect small bursts
- also has ECC on memory addresses to avoid reading from or writing to the wrong address

- data on buses are parity protected

- disk-driver software adds end-to-end checksum to end of 512-byte disk sectors

- have NonStop remote duplicate database facility (RDF) to protect against earthquakes, hurricanes, fires, floods, and now, terrorism by sending database updates to remote site thousands of miles away, and if disaster occurs, remote site takes over in matter of minutes

- also considered “coffee fault-tolerant”

**Stratus Systems**

- continuous processing systems to supply uninterrupted operation with no loss of data or degradation of performance

- in 1999 Stratus acquired by Investcorp but still operates as Stratus Computers

- customers include major credit card companies, 4 of 6 US regional securities exchanges, largest stock exchange in Asia, 15 of world’s 20 top banks, 911 emergency services, etc.

- architecture as shown in Figure 3.21

- as with Tandem, CPUs, I/O and memory controllers, disk controllers, communication controllers, high-speed buses are duplicated

- fault tolerance of hardware different from Tandem in that it has four microprocessors all running the same instruction stream configured as redundant pairs of physical processors

- on detection of a fault (continual compares) redundant pair takes over

- Stratus system is simpler, lower in cost and competes at middle and lower end of market, leaving higher end and more complex system to Tandem

- each Stratus circuit board has self-checking hardware that continuously monitors operation so that if an error is detected it takes itself out of service and the backup takes over, while Tandem has backup continuously processing
- if error determined transient, then board is put back into service

- much attention has been paid to power supply which some manufactures do not do

Clusters

- idea with cluster is that if one system is going down, operating system will be able to transfer from an on-line system to a standby without overall system going down

- Sun cluster is such an example that has following characteristics:
  
  o ECCs on all memories and caches
  
  o RAID controllers
  
  o Redundant power supplies, cooling fans
  
  o System can lock out bad components
  
  o Solaris 8 has error-capture capabilities
  
  o Solaris 8 provides recovery reboots
  
  o Sun Cluster 2.2 is add on to Solaris and allows up to 4 nodes with networking and fiber-channel interconnections and some form of nonstop processing upon failures
  
  o Sun Cluster 3.0 (released in 2000) increased number of nodes and simplified the hardware

- overall, now adding in many of fault-tolerant techniques Tandem has had for some time