Today's Topics

- Error Correcting Code
- The memory Hierarchy
- Secondary Storage, disks, and RAID.
- Other Secondary Storage Devices, Tape, CD, DVD, . . .

Error Correcting Codes

External storage devices are not perfect, sometimes data is corrupted. We need some way or recognising that data is corrupt.

There are two separate issues:

- Recognising is data is corrupt.
- Correcting Corrupt data.

An Error Correcting Code - English

- Deer granney Pleze send mi some moore of you're lovely foood, I really mis you're tish and nap
- fjff ajfdkl asdkl dfjkl afjk ajsdkldf;as fdjk ase

Why is the first sentence easier to correct than the second?

What is the feature of English that makes it possible to correct errors?

Redundancy

We need to build some redundancy into our representation.

- What is we simply repeated every thing twice?

Can we correct errors, can we detect errors?

Can we come up with a more efficient method?

Simple Error Detecting Code

Some memories have an extra bit the parity bit. Which is 1 (or 0) if the number of 1 bits is even and 0 (or 1) if the number of 1 bits is odd.

Parity bits can detect errors in 1-bit. But they can't correct them.

Code Distance

- To *detect* d-single-bit errors you need a distance $d + 1$ code.
- To *correct* d-single-bit errors you need a distance $2d + 1$ code.

Why?

Hamming Distance

The two binary words:

- 11110001
- 00110000

are Hamming distance 3 apart.

The Hamming distance is the number of bits in which the two words differ.

It would take three error to convert the first word into the second.

Redundancy through Hamming Distance

Instead of having one word for our data, we have many words representing the data.

Example code with four valid key-words

- 0000000000, 0000011111 , 1111100000 and 1111111111

The distance of a code is the minium Hamming distance between any two words. This code has distance 5.

The Hamming Construction

Example 16 bits and 5 parity bits.

- Bit 1 checks bits 1,3,5,7,9,11,13,15,17,19,21
- Bit 2 checks bits 2,3,6,7,10,11,14,15,18,19
- Bit 4 checks bits 4,5,6,7,12,13,14,15,20,21
- Bit 8 checks bits 8,9,10,11,12,13,14,15
- Bit 16 checks bits 16,17,18,19,20,21

Overhead

Word Size	Check Bits	Total Size	Overhead (percent)
8	4	12	50
16	5	21	31
64	7	71	11
128	8	136	6
256	9	265	4
512	10	522	2

The Hamming Construction

Have more than one parity bit. Each bit checking the parity of a different set of bits.

Number the bits starting at 1 (not 0). Each bit that is a power of 2, will be a parity bit.

Bit b is checked by parity bits $b_1, \ldots, b_k$ suck $b_1 + \cdots + b_k = b$.

Number of Parity Bits needed

To correct 1-bit errors in m-words r parity bits are needed where

$$(m + r + 1) \leq 2^r$$

Correcting Errors

Compute all the parity bits, if they are correct then the is either 0-errors or more than 1.

Otherwise add up the incorrect parity bits, the resulting sum is the position of the incorrect bit.

Example: Parity bits 1,4 incorrect (but rest correct) $\Rightarrow$ bit 5(=1+4) has been inverted.

Issues in Constructing Codes

- People want to correct more than one-bit.
- People want low overhead

Lots of theory on how to do this, a good example of applied maths which doesn't involve calculus.

Other Codes

- CRC-checks
- Reed-Solomon code

Secondary Storage

Secondary storage is not perfect, therefore you have to use error-correcting (or detecting) codes to store the data.

Often this is done the hardware of the device, the software doesn't even notice that there is an error correcting code.

RAID

Redundant Array of Inexpensive Disks

- Developed by Patterson (Same person responsible for early version of the MIPS) and . . .
- Basic Idea, couple together lots of cheap hard-disks, read data from more than one drive at a time, (faster throughput).
- Looks to the main machine just like a single hard-disk.

RAID Levels

- Divide the sectors of the disk up into strips each on a separate drive. Works well with large requests.
- RAID 1 - Duplicate the data, two sets of drives for the data. Read performance better then RAID 1, but write performance no better.
- RAID 2 - Each Drive stores each bit of a word, use Hamming code to store data. Drives must be synchronised, (always the right bit appearing at the right time)
- RAID 3 - Simplified RAID 2, with only one parity bit.
- RAID 4 and 5, back to strips, bit with parity strips (RAID 4) or RAID 5, distribute parity bits evenly over the drives.

The Bottleneck

CPUs are faster than secondary storage devices. How do we get data into the CPU faster.

Think about how they might design AltaVista . . .

Pitfall Don't Increase the failure Rate

- One Drive mean time to failure 20,000 Hours
- 4 Drives, mean time to failure for anyone drive is 5000 hours.

Message: You have to build in redundancy. Otherwise you have a less efficient system.

Other Secondary Storage Devices

- CDs - Holds 640Meg. Writable once and Write many versions.
- DVDs - Similar technology to CDs, but much more advance, higher density. 4.7GB - 17GB. Format wars, . . . who will be the next beta-max?