CS 444 Operating Systems

Hamming Code

J. Holly DeBlois

August 19, 2024

Richard Hamming

- He studied error correction code in the late 1940s to early 1950s
- Hamming code can correct 1 bit error and detect 2 bit errors
- Code nomenclature uses the number of bits per word and the number of data bits
- Hamming(7, 4) code: 7 bits per word, 4 data bits, 3 parity bits
- Hamming(15, 11) code: 15 bits per word, 11 data bits, 4 parity bits
- Hamming(31, 26) code: 31 bits per word, 26 data bits, 5 parity bits

Hamming(7, 4) Code

- 4 data bits — a nibble
- Typically, the least significant bit is on the right
- Hamming code is the opposite — endianness matters
- The following is the code for the nibble 0110_2

```
001  010  011  100  101  110  111   //address
 p1   p2   D1   p4   D2   D3   D4   //designation
-------------------------------------------------
 1    1    0    0    1    1    0    //code word
```

- D1, D2, D3, D4 are data bits
- P1: parity of D1, D2, D4
- P2: parity of D1, D3, D4
- P4: parity of D2, D3, D4

Hamming Distance

- Take 2 bit strings
- Calculate their bitwise XOR
- Count the number of 1 bits in the XOR'd string
- This count is the Hamming distance of the 2 bit strings

Hamming Distance of Hamming(7, 4) Codes

- Data of 4 bits
  - 16 strings
  - Hamming distance is at least 1 between data strings
- Code of 7 bits
  - 128 possible strings
  - Only 16 strings are correct codes
  - Hamming distance is at least 3 between correct codes
- When 1 bit is flipped, we can correct it — change the incorrect code to the nearest correct code
- When 2 bits are flipped, we can detect them but can't correct them, because the incorrect code is equidistant to 2 correct codes
- When 3 bits are flipped, it may actually become another correct code

Error Correction When 1 Bit is Flipped

- Let's flip D2

```
001 010 011 100 101 110 111  //address
 p1  p2  D1  p4  D2  D3  D4  //designation
---------------------------------------------------
 1   1   0   0   1   1   0   //correct code

 1   1   0   0   0   1   0   //D2 flipped
 0   1       1               //check parity
 x           x               //discrepancy
```

- The disagreeing parity bits give the address of the incorrect bit
- P1 and P4 form the address 101, so we know D2 is flipped

Flip a Different Bit

- Let's flip D3

```
001 010 011 100 101 110 111  //address
 p1  p2  D1  p4  D2  D3  D4  //designation
--------------------------------------------
 1   1   0   0   1   1   0   //correct code

 1   1   0   0   1   0   0   //D3 flipped
 1   0       1               //check parity
     x       x               //discrepancy
```

- P2 and P4 form the address 110, so we know D3 is flipped

Flip a Parity Bit

- Let's flip P2

```
001  010  011  100  101  110  111   //address
 p1   p2   D1   p4   D2   D3   D4   //designation
--------------------------------------------------
 1    1    0    0    1    1    0    //correct code

 1    0    0    0    1    1    0    //P2 flipped
 1    1         0                   //check parity
      x                             //discrepancy
```

- P2 is its own address, so we know P2 is flipped

Hamming(15, 11) Code

- 11 data bits, 4 parity bits

p1	p2	D1	p4	D2	D3	D4	p8	D5	D6	D7	D8	D9	D10	D11	
1	0	0	1	1	1	0	0	1	0	0	1	1	1	0	
1		0		1		0		1		0		1		0	p1
	0	0			1	0			0	0			1	0	p2
			1	1	1	0					1	1	1	0	p4
							0	1	0	0	1	1	1	0	p8

- P1: XOR every other bit
- P2: XOR every other couple of bits
- P4: XOR every other quadruple of bits
- P8: XOR every other octuple of bits
- Hamming(31, 26) code has P16: XOR every other 16-tuple

Connection Machine CM-2, circa 1980s

- A massively parallel supercomputer
- Hamming(38, 32) code
  - 32 data bits
  - 6 parity bits
  - Shortened from Hamming(63, 57)
- Add 1 extra bit for word parity, 39 bits in total
- 4 bytes of data become 39 bits of code, spread over 39 identical drives
- The drives are totally synchronized
- 32 times of data throughput than a single drive
- Overhead 7/39 (about 18%) of capacity for error detection and correction

ECC Memory

- Hamming code is used in ECC RAM
- Circuit widths on chips keep shrinking
- Circuit voltage is dropping — to be more power efficient
- Fewer atoms at lower energized state to keep a bit
- A high energy particle from space can flip the bit
- ECC RAM uses a small amount of memory for the parity bits
- Servers can be configured with ECC on or off
    - ECC is required for scientific computing