CERN School of Computing

3.1.1

HINT 1 (1)

Parity of the bits B1, B2, B3 is computed as:

P = B1 ^ B2 ^ B3

If you don't have a good vectorizing compiler it is faster to use the largest word operation supported by your OS to compute the parity of many bits at a time and to detect wrong words first. Once you have the word you can identify faulty bytes within the word.

Otherwise just do a byte loop !

3.1.4

HINT 1 (1)

If a byte in stripe 1 is faulty by definition the same byte in all other stripes has to be considered faulty.

Parity alone is not sufficient to detect which stripe is damaged.

3.1.4

HINT 1 (2)

The sha1sum program allows you to compute SHA1 sums for files.

3.1.5

HINT 1 (1)

If you have only stripes 1,2,4,P you can recalculate stripe 3 as
3 = 1^2^4^P

3.2.1

HINT 1 (1)

You might just compare the number of operations needed for byte wise XOR or for word XOR operation ....

3.2.2

HINT 1 (1)

Strategy is to recover every row first using diagonal parity, than row-parity aso....

3.2.3

HINT 1 (1)

If X is the number of words in a stripe and N the number of stripes, then (N-2) is the number of stripes over which the original data gets distributed.

Write down a formula with this symbols or consider the two cases where N=4 and N=1.000.000

3.2.4

HINT 1 (1)

Consider two cases:

- we read the required stripe data and then compute
- we can read and compute in parallel using a sort of read-ahead algorithm for sequential reads