Data Technologies – CSC 2009

Lectures: Alberto Pace, Bernd Panzer-Steindl

Exercises: Andreas-Joachim Peters

Email Andreas.Joachim.Peters@cern.ch

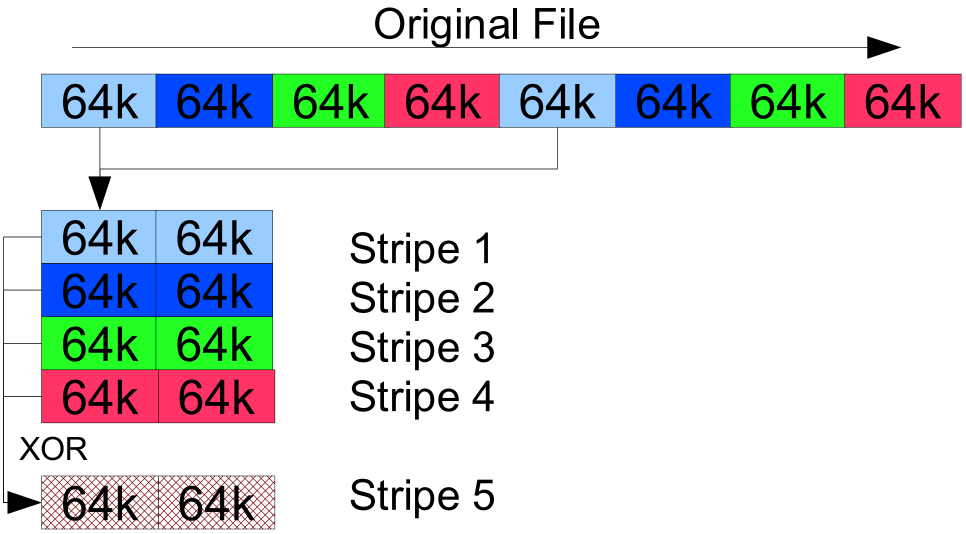
Exercises Data Technologies

3 Understanding RAID & Redundancy

3.1 RAID-4

In your /data directory you can find 5 files named ***stripe-<n>*** [n=1-5].

File 1-4 contain each the data of one 64k sized stripe in RAID-4 format, file 5 contains the parity stripe of a RAID-4 format. The format is explained here:



3.1.1 Write a program that recalculates and checks the parity information of these RAID-4 files. Choose a fast implementation! You can make the assumption that the error rate is low.

I have provided a framework program for this written in C which misses only the computational part. You find it on your desktop in the /data directory: `/data/readstripe/readstripe.c`

```
> cd /data/readstripe
> ./compile.sh # to compile readstripe.c with -O3
> time readstripe /data/stripe
> real      0m0.021s
> user      0m0.001s
> sys       0m0.020s
```

Extend the programm with the missing parity verification!

3.1.2 Print all faulty byte offsets in stripe 1.

3.1.3 Measure the execution time after many concurrent executions.

If you have it correctly running (e.g. you detect the right bytes), I will add you to the ranking score *(I verified that my algorithms run equally fast on csc01-40)*

Top Implementations (+- 2ms) :

0	Andreas	AlgSSE2	30ms	out of competition	1.74 GB/s
1	Andreas	Alg64	39ms		1.34 GB/s
2	Andreas	Alg8.1	55ms		949 MB/s
3	Andreas	Alg8.2	84ms		621 MB/s
4	Andreas	Alg8.3	98ms		532 MB/s

3.1.4 Which additional information would you need to decide which stripe has faulty information? I have added this information in the extended attributes of each stripe.

You can read it with `getfattr` ! There is a command on Linux to compute this information: take the name of the attribute (without user.) and add a 'sum' and you have the command to do it!

For the fast ones who have time left ... in any case please do it after exercise 3.2

3.1.5 As you might have verified before, all errors are located in only one stripe.

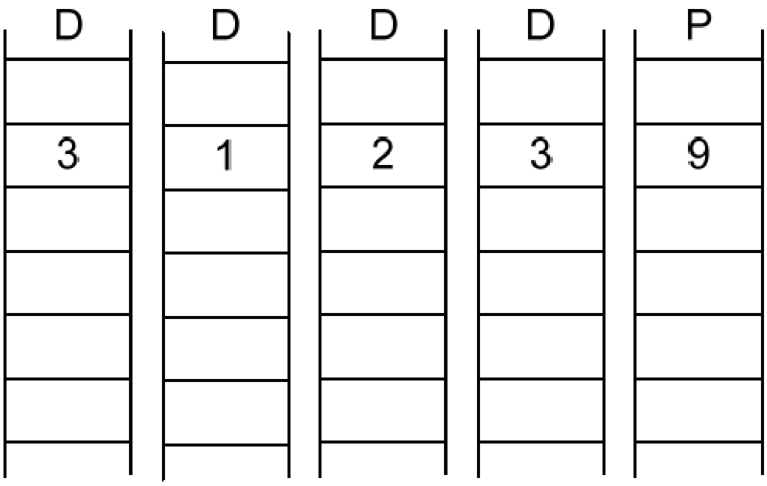
Try to recover and recreate the original file with this information using the 4 correct stripes. To do this correctly you need also the information about the original file size. Can you explain why?
In any case, it was 324675210 bytes! If you did the right job, you can list the contents of the file using
'tar tvzf <merged file>` ! If not, it will complain that it is not readable!

3.2 *Raid-4 with row-diagonal parity (Raid-DP)*

Reed-Solomon Codes help to improve the redundancy and protect against several device failures. However the computational effort is growing fast with higher redundancy and typically implementations are better done in dedicated hardware.

There is an alternative implementation to protect against double disk failures based on simple parity operations called RAID-DP. The following figures shows how this implementation works.

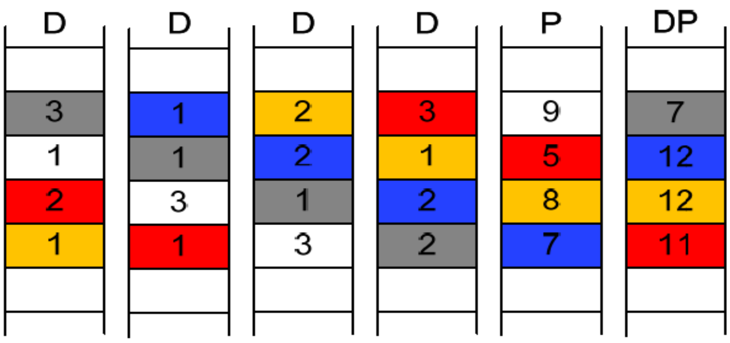
We start with a figure explaining a normal RAID-4. Instead of XOR the SUM is actually computed for illustration purposes.



D	D	D	D	P
3	1	2	3	9

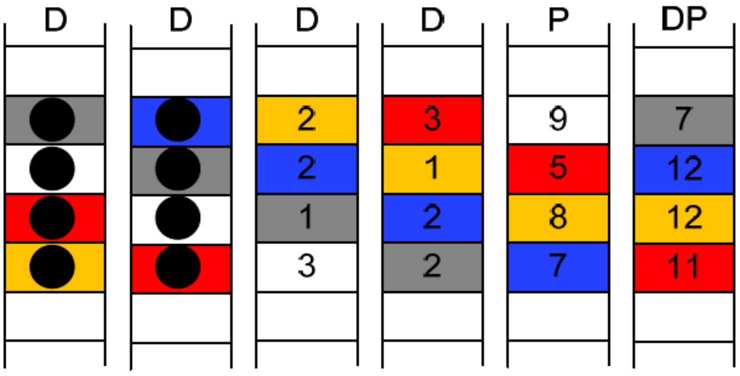
3.2.1 *Which influence has the XOR unity (bit/word length) on the operation itself and on the performance?*

The computation of the second parity column is illustrated in the following picture:



D	D	D	D	P	DP
3	1	2	3	9	7
1	1	2	1	5	12
2	3	1	2	8	12
1	1	3	2	7	11

3.2.2 ***Try to write the required order of necessary parity operations to recover from a double disk failure and count the total number of parity operations (# of XOR). The faulty blocks are illustrated as black circles in the following picture:***



3.2.3 ***Imagine you have 1MB of data. Try to scetch how the number of parity operations evolves in Raid-DP if you increase the number of stripes from 4 to 1 Million.***

3.2.4 ***We have a Raid-DP setup where each stripe is written on an identical individual harddisk. Do you have a 'feeling' how fast read operations can be recovered if the first two stripes (D-Stripes) are lost? If we can read non-degraded 100 MB/s and the XOR code run's at 600 MB/s, how fast can the degraded operation be?***

3.2.5 ***What would be the effective usable space in percent for a RAID-DP with 10 stripes if you include an SHA1 checksum for each 4k chunk? Are 4k chunks and SHA1 good choices?***