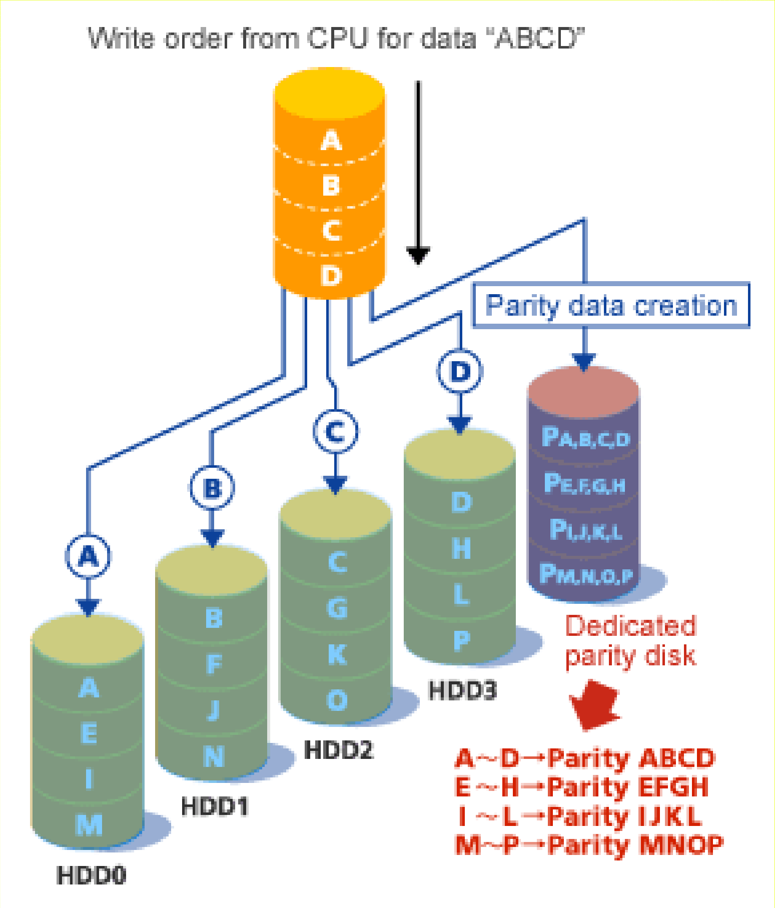
CERN
School of Computing

Data Technologies – Exercise 3
Fastest Parity Check Implementation ...



Dedicated parity disk

A~D→Parity ABCD
E~H→Parity EFGH
I~L→Parity IJKL
M~P→Parity MNOP

$$
\begin{aligned}
p \oplus q &= (p \wedge \neg q) &&\vee && (\neg p \wedge q) \\
&= ((p \wedge \neg q) \vee \neg p) &&\wedge && ((p \wedge \neg q) \vee q) \\
&= ((p \vee \neg p) \wedge (\neg q \vee \neg p)) &&\wedge && ((p \vee q) \wedge (\neg q \vee q)) \\
&= (\neg p \vee \neg q) &&\wedge && (p \vee q) \\
&= \neg(p \wedge q) &&\wedge && (p \vee q)
\end{aligned}
$$

p	q	$\oplus$
F	F	F
F	T	T
T	F	T
T	T	F

The Benchmarking ....

- I downloaded all your existing binaries to the same desktop machine
- I created 4 stripe files and parity for a 886 MB archive file
- I modified a single byte in the middle of the parity stripe and verified your code
- I benchmarked your code in 20 concurrent iterations and measured the average

THE TOP
10
9
4
8
1
5
6
2
3
7

The Results ….

Desktop	[s]	login	Team	
CSC24	0.440	*gpuccian*	Gianni Pucciani	64 INCPTR ^-CON
CSC02	0.442	*sbukowie*	Sebastian Bukowiec	64 CALPTR !=CON
CSC09	0.443	*mmeijer*	Melvin Meijer	64 CALPTR !=CON
CSC38	0.443	*dsinuela*	David Sinuela Pastor	64 INCPTR ^-CON
CSC33	0.444	*jmaes*	Joris Maes	64 CALPTR !=CON
*CSC34**	0.456	*mnuhn*	Malte Nuhn	64 CALPTR !=CON
CSC06	0.460	*jmacedo*	Jose Macedo	128 SSE2 CALPTR !=CON
CSC39	0.512	*caguado*	Carlos Aguado Sanchez	64 CALPTR ^-CON
CSC41	0.667	*mborodin*	Maskym Borodin	64 CALPTR ^-CON
CSC26	0.909	*cbrachem*	Carsten Brachem	8 CALPTR ^-CON

An easy solution ....

... was slow ... (2x)

```
{
        int i;

        for (i=0; i<65536; i++){
          if (  stripebuffer[0][i] ^
                stripebuffer[1][i] ^
                stripebuffer[2][i] ^
                stripebuffer[3][i] ^
                stripebuffer[4][i]) {
            printf(" ERROR: %d\n",i);
          }
        }
}
```

A fancy solution ....

... is even slower .. (3x).

```
{
  int k=65536;
  int i;
  char* ptr[5];
  for (i=0; i< 5; i++) ptr[i] = stripebuffer[i];
  while (k--) {
    if ( (*ptr[0]++) ^
         (*ptr[1]++) ^
         (*ptr[2]++) ^
         (*ptr[3]++) ^
         (*ptr[4]++) )
      printf("error ...);
  }
}
```

The Fastest Code ....

is simple ....

```
{
      int i;
      long *p0 = (long*)stripebuffer[0];
      long *p1 = (long*)stripebuffer[1];
      long *p2 = (long*)stripebuffer[2];
      long *p3 = (long*)stripebuffer[3];
      long *p4 = (long*)stripebuffer[4];

      for (i=0; i<8192; i++){
        if ( *p0 ^ *p1 ^ *p2 ^ *p3 ^ *p4) {
          printf(" ERROR: %d\n",i);
          ....
        }
        p0++;p1++;p2++;p3++;p4++;
      }
}
```

But ....

I was hacking last night ....

and ....

```
{
    double *tr0,*ptr1,*ptr2,*ptr3,*ptr4;
    register int k,i;
    double out[2];
    __m128d a;
    __m128d b;
    __m128d c;
    ptr0 = (double*) stripebuffer[0];
    ptr1 = (double*) stripebuffer[1];
    ptr2 = (double*) stripebuffer[2];
    ptr3 = (double*) stripebuffer[3];
    ptr4 = (double*) stripebuffer[4];

    for (k=0; k< 4096; k++) {
      a = _mm_load_pd(ptr0);
      b = _mm_load_pd(ptr1);  c = _mm_xor_pd(a,b);
      a = _mm_load_pd(ptr2);  c = _mm_xor_pd(c,a);
      a = _mm_load_pd(ptr3);  c = _mm_xor_pd(c,a);
      a = _mm_load_pd(ptr4);  c = _mm_xor_pd(c,a);
      _mm_store_pd(out, c);

      if (out[0] || out[1]) {
        printf("error in block %d 128word %d\n", nreadchunk, k);
        ....
      }
      ptr0++; ptr0++; ptr1++; ptr1++; ptr2++; ptr2++; ptr3++; ptr3++; ptr4++; ptr4++;
    }
}
```

... produced this using only -O2

.... and ...

I will have to keep the cup until 2010 with 0.430s (-10ms) and 2.06 Gb/s parity validation!

Last Information:

I will provide a tar ball asap with all exercise materials and solutions for you to download from the CSC web site!

I am happy to answer further questions by mail:
Andreas.Joachim.Peters@cern.ch

Thank you for your attendance!