RAID Level 5: Left Symmetric Data Layout

disk 0	disk 1	disk 2	disk 3	disk 4
A.0	A.1	A.2	A.3	B.0
B.1	B.2	B.3	C.0	C.1
C.2	C.3	D.0	D.1	D.2
D.3	E.0	E.1	E.2	E.3
PE	PD	PC	PB	PA

Access functions

user addresses: a $\quad 0 \leq a < n(r\text{-}1) \quad$ (full disks)

n disks, r stripe units per disk, $b = r$ stripes

$$p = a/(n(n\text{-}1))$$

$$disk(a) = a \ \% \ n$$

$$offset(a) = a/n + p$$

$$parityDisk(a) = (\, a/(n\text{-}1) + 1\,)(n\text{-}1) \ \% \ n$$

$$parityOffset(a) = p(n\text{-}1)$$

Disk Array Architectures II

Work Load: *R reads* **and** *W writes*

- **Small Operation Fault Free Performance:**
  the total array workload is $R + 4W$ operations for small operations and each disk gets $1/n$ th the total.

Degraded Mode designates single disk failure

- **Small Operation Degraded Mode Performance:**
  Both the reads and writes increase the workloads. The total read load for the surviving n-1 disks is $2\ R\ (n\text{-}1)/n$ operations. The total write load for the survivors is $4W + W(n\text{-}7)/n$ operations.

- **Performance Consequences:**
  Since the workload increases for each surviving disk in degraded mode, we must utilize less than the full array capability in fault free operation. Otherwise, we will not be able to meet the needs of our clients during the inevitable degraded operation. For read only loads, we can use only one-half the capability.

Disk Array Architectures III

Declustering improves the degraded mode performance.

- **Disk Array Declustering**
  The client data is spread over additional disks; n disks with stripe width k, typically $n > k$.

Example: $n = 6$, $k = 5$.

disk 0	disk 1	disk 2	disk 3	disk 4	disk 5
A.0	A.1	A.2	A.3	B.0	B.1
B.2	B.3	C.0	C.1	C.2	C.2
E.1	D.0	D.1	D.2	D.3	E.0
F.3	E.2	E.3	F.0	F.1	F.2
PC	PF	PB	PE	PA	PD

- **Data Layout Goals**

Error Correction	stripe units of each stripe mapped to distinct disks.
Distributed Parity	each disk contains the same number of parity stripe units.
Distributed Reconstruction	the incremental workload is evenly distributed over the surviving disks.
Efficient Mapping	run-time efficient requiring little additional memory to represent the data layout.
Parallel Reads	contiguous client data mapped to distinct disks as much as possible.
Large Writes	contiguous client data mapped to single stripes as much as possible.

Data Layout

n	disks	r	stripe units/disk
k	stripe width	b	stripes/iteration

Complete block designs declustering

Layout calculated via binomial number system.

A complete block design contains all possible k-combinations of n items. There are $\binom{n}{k}$ such combinations. Each combination designates a set of disks for a stripe. The position/row of the combination designates the offsets on each disk.

A:	0	1	2
B:	0	1	3
C:	0	2	3
D:	1	2	3
E:	0	1	4
F:	0	2	4
G:	1	2	4
H:	0	3	4
I:	1	3	4
J:	2	3	4

$n = 5$, $k = 3$. The A stripe resides on disks 0, 1, & 2 at offsets 0, 0, & 0, the B stripe on disks 0, 1, & 3 at offsets 1, 1, & 0, etc.

- **Example:** $n = 4\ ,\ k = 3$

disk 0	disk 1	disk 2	disk 3
PA	A.0	A.1	B.1
PB	B.0	C.0	C.1
PC	PD	D.0	D.1
E.0	PE	E.1	F.1
F.0	PF	PG	G.1
G.0	H.0	PH	H.1
I.0	I.1	PI	PJ
J.0	J.1	K.1	PK
K.0	L.0	L.1	PL

A	0	1	2
B	0	1	3
C	0	2	3
D	1	2	3

- scheme

- **Layout Parameter Details:**

  $n = 4$, $k = 3$. There are $b = 4$ stripes per iteration. Each disk contains $r = 3$ stripe units per iteration. The parity units are rotated among the columns of the scheme.

- k-combinations $\binom{n}{k}$ possible k-combinations.

$\langle\, X_0, \ldots, X_{k-1} \,\rangle$ assuming $X_{i-1} < X_i$ for $0 < i < k$.

- total order $\succ$

$$\langle\, X_0, \ldots, X_{k-1} \,\rangle \succ \langle\, Y_0, \ldots, Y_{k-1} \,\rangle$$

if there is a $j < k$ such that X_i equals Y_i for $j < i < k$ and $X_j > Y_j$.

- mapping: $\langle\, X_0, \ldots, X_{k-1} \,\rangle$ is in position

$$\binom{X_0}{1} + \binom{X_1}{2} + \cdots + \binom{X_{k-1}}{k}$$

- inverse mapping

exists since each k-combination is in a unique position according of $\succ$.

Datum VIII

- inverse mapping calculation $\mapsto$

```
inverseMapping ( int position )

    for ( int i = k-1 ; i >= 0 ; i-- ) {
        ll = i ;
        while ( binomial( ll , i+1 ) <= position )
            ll = ll + 1 ;
        c[i] = ll - 1 ;
        position = position - binomial ( c[i] , i+1 ) ;
    }
```

- c array contains the k-combination associated with `position` X_i is in slot `c[i]`.

Datum IX

- **Binomial number system** for $0 \leq s < \binom{n}{k}$

$$s = \binom{d_0}{1} + \binom{d_1}{2} + \cdots + \binom{d_{k-1}}{k}$$

with $0 \leq d_0 < d_1 < d_2 < \cdots < d_{k-1} < n$.

For $0 \leq s < \binom{n}{k}$ with $i \leq d_i < d_{i+1} < n$,

the binomial number system mapping

$$s \mapsto \{ d_0, d_1, \cdots, d_{k-1} \}$$

is a bijection.

- **Examples:**

14 $\mapsto \{4 , 5 \}$ k = 2

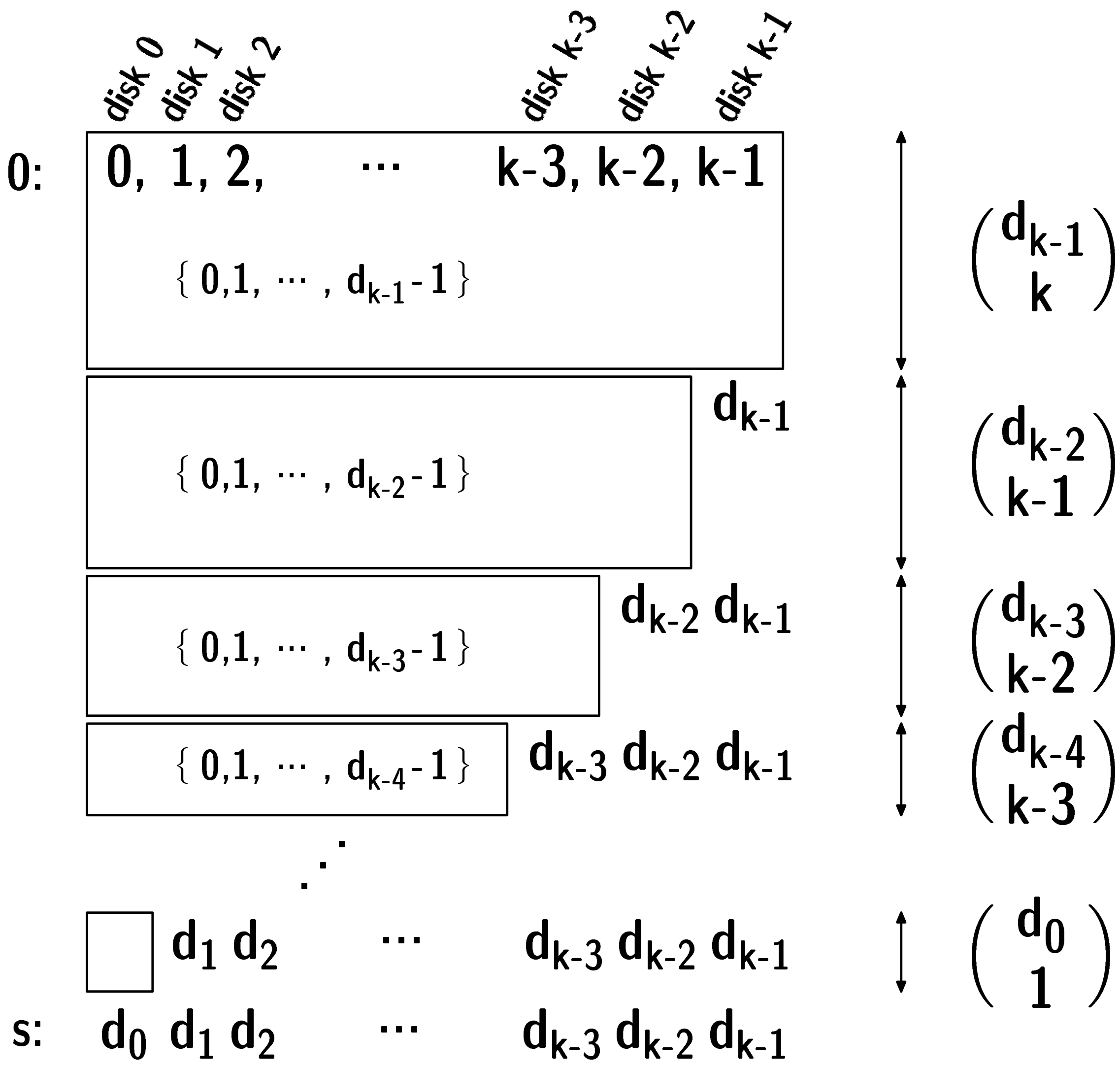
14 $\mapsto \{1 , 3 , 5 \}$ k = 3

15 $\mapsto \{0 , 6 \}$ k = 2

15 $\mapsto \{2 , 3 , 5 \}$ k = 3

Datum X

- Position of $\langle d_0, d_1, \cdots, d_{k-1} \rangle$



Binomial Number System

- **Datum Access functions**

n disks

r stripe units per disk

k stripe width

b stripes / iteration

$bk = nr$ stripe units per iteration. $m = k - 1$

user addresses: a $0 \leq a < b(k\text{-}1)$ (full disks)

$$stripe = \left\lfloor a/m \right\rfloor \bmod \binom{n}{k}$$

$$stripe \mapsto \{ d_0, d_1, \cdots , d_{k\text{-}1} \}$$

$$iterations = \left\lfloor \left\lfloor a/m \right\rfloor / \binom{n}{k} \right\rfloor$$

$$parity = iterations \bmod k$$

$$index = a \bmod m$$

if ($index \geq parity$) $index = index + 1$;

- **Datum Access functions**

$$dataDisk(a) = d_{index}$$

$$parityDisk(a) = d_{parity}$$

$$dataOffset(a) = iterations \times \binom{n-1}{k-1} + offset(a, index)$$

$$parityOffset(a) = iterations \times \binom{n-1}{k-1} + offset(a, parity)$$

$$offset(a,i) = \begin{cases} \sum_{j=1}^{k-1} \binom{d_j-1}{j} & i = 0 \\ \sum_{j=0}^{i-1} \binom{d_j}{j+1} + \sum_{j=i+1}^{k-1} \binom{d_j-1}{j} & 0 < i < k-1 \\ \sum_{j=0}^{k-2} \binom{d_j}{j+1} & i = k-1 \end{cases}$$

- $\mapsto$ returns `d[i]` values.

```
for ( i = k ; i > 0 ; i-- ) {
    for ( l = i ; choose[l][i] <= stripe ; l++ ) ;
    d[i-1] = l-1 ;
    stripe -= choose[l-1][i] ;
}
```

- Examples $n = 6$, $k = 3$:

$s = 4$

$\binom{4}{3} = 4$: d[2] is 4

$\binom{1}{2} = 0$; d[1] is 1

$\binom{0}{1} = 0$; d[0] is 0

$s = 5$

$\binom{4}{3} = 4$: d[2] is 4

$\binom{2}{2} = 1$; d[1] is 2

$\binom{0}{1} = 0$; d[0] is 0

$s = 7$

$\binom{4}{3} = 4$: d[2] is 4

$\binom{3}{2} = 3$; d[1] is 3

$\binom{0}{1} = 0$; d[0] is 0

$s = 8$

$\binom{4}{3} = 4$: d[2] is 4

$\binom{3}{2} = 3$; d[1] is 3

$\binom{1}{1} = 1$; d[0] is 1

Datum XIV

- **Example:** $n = 8\ ,\ k = 4\ ,\ a = 354.$

$$stripe = \left\lfloor 354 \,/\, 3 \right\rfloor \ \mathbf{mod} \ \binom{8}{4} = 118 \ \mathbf{mod}\ 70 = 48$$

$$stripe \ \mapsto \ \{0\ ,\ 3\ ,\ 5\ ,\ 7\ \}$$

$$iterations = \left\lfloor \left\lfloor 354 \,/\, 3 \right\rfloor \,/\, 70 \right\rfloor = 1$$

$$parity = 1 \ \ \mathbf{mod} \ \ 4 = 1$$

$$index = 354 \ \ \mathbf{mod} \ \ 3 = 0$$

$$disk(354) = d_0 = 0$$

$$parity(354) = d_1 = 3$$

Datum XV

$$
\begin{aligned}
\mathit{offset}(354,0) &= \sum_{j=1}^{3} \binom{d_j\text{-}1}{j} \\
&= \binom{2}{1} + \binom{4}{2} + \binom{6}{3} = 28
\end{aligned}
$$

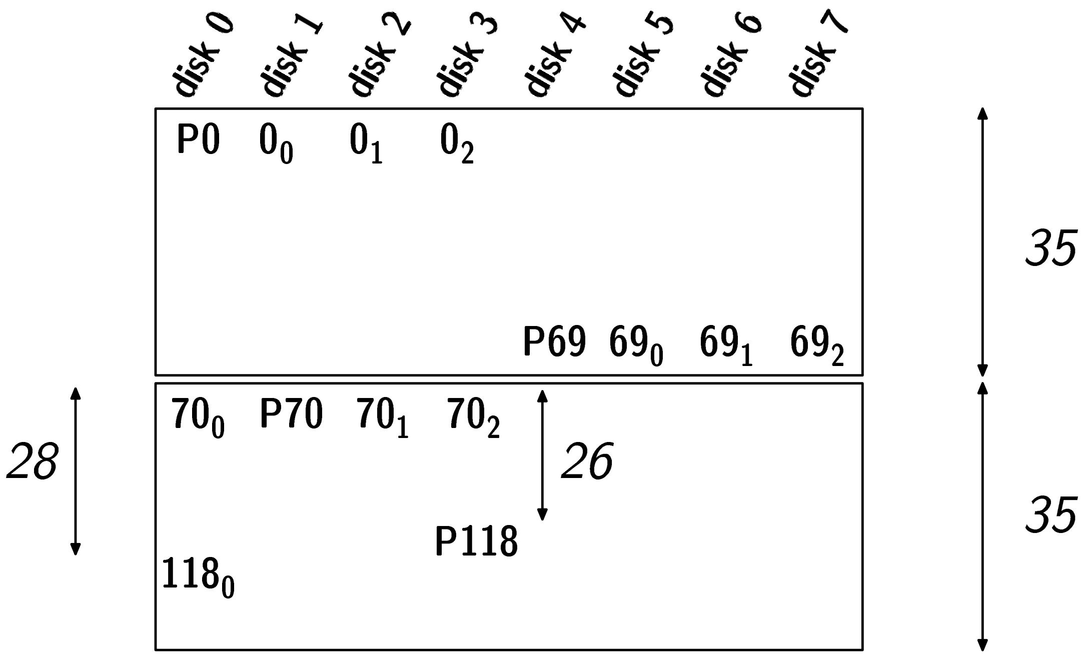
$$
\begin{aligned}
\mathit{offset}(354,1) &= \sum_{j=0}^{0} \binom{d_j}{j+1} + \sum_{j=2}^{3} \binom{d_j\text{-}1}{j} \\
&= \binom{0}{1} + \binom{4}{2} + \binom{6}{3} = 26
\end{aligned}
$$

$$
\begin{aligned}
\mathit{diskOffset}(354) &= \mathit{offset}(354,0) + \binom{7}{3} \\
&= 28 + 35 = 63
\end{aligned}
$$

$$
\begin{aligned}
\mathit{parityOffset}(354) &= \mathit{offset}(354,1) + \binom{7}{3} \\
&= 26 + 35 = 61
\end{aligned}
$$

Datum XVI

- **Example details:** $n = 8 \; , \; k = 4 \; , \; a = 354.$



$$stripe = \left\lfloor 354 \,/\, 3 \right\rfloor \textbf{ mod } \binom{8}{4} = 118 \textbf{ mod } 70 = 48$$

$$stripe \mapsto \{0 \, , \, 3 \, , \, 5 \, , \, 7\}$$

$$parity = 1 \qquad index = 0$$

$$disk(354) = d_0 = 0 \quad parity(354) = d_1 = 3$$

$$diskOffset(354) = 63 \qquad parityOffset(354) = 61$$

- **Structural Properties:** n disks & stripe width k

Number of stripes per iteration: $\binom{n}{k}$

Number of stripes per period: $k\binom{n}{k}$

Number of stripe units per disk per iteration: $\binom{n-1}{k-1}$

- **Data Layout Goals revisited**

Error Correction

Stripe units distributed over k distinct disks.

Even Parity Distribution

Parity stripe units per disk per period: $\binom{n-1}{k-1}$

Even Reconstruction Workload Distribution

Number of times a pair of disks occurs together per iteration: $\binom{n-2}{k-2}$

Large Write Optimization

Every stripe contains a contiguous interval of client data.

Parallel Reads

Contiguous client data is not mapped to distinct disks. We do not meet all the data layout goals.

- ### Degraded Mode Small Operation Performance

Work Load: *R read* and *W writes*

The total read workload for the surviving n-1 disks is $R(n+k\text{-}2)/n$ operations and the total write workload for the survivors is $4W + W(k\text{-}7)/n$ operations.