Coded Data Rebalancing

Abhinav Vaishya

Introduction

- Data is stored in a distributed fashion in the storage nodes with some replications in distributed storage systems.
- Data Replication provides:
  - Easy availability/maintenance of data
  - Protection of data upon node failure

Data Skew

Non-uniform distribution of data across the storage nodes.

Reason: Node failures or additions

Problems that arise:

- Load imbalance
- Delay in task completions

Solution: Data Rebalancing!

Data Rebalancing

Balancing the distribution of data and reinstating the replication factor.

How?

Communication of data symbols between nodes.

Coded Data Rebalancing

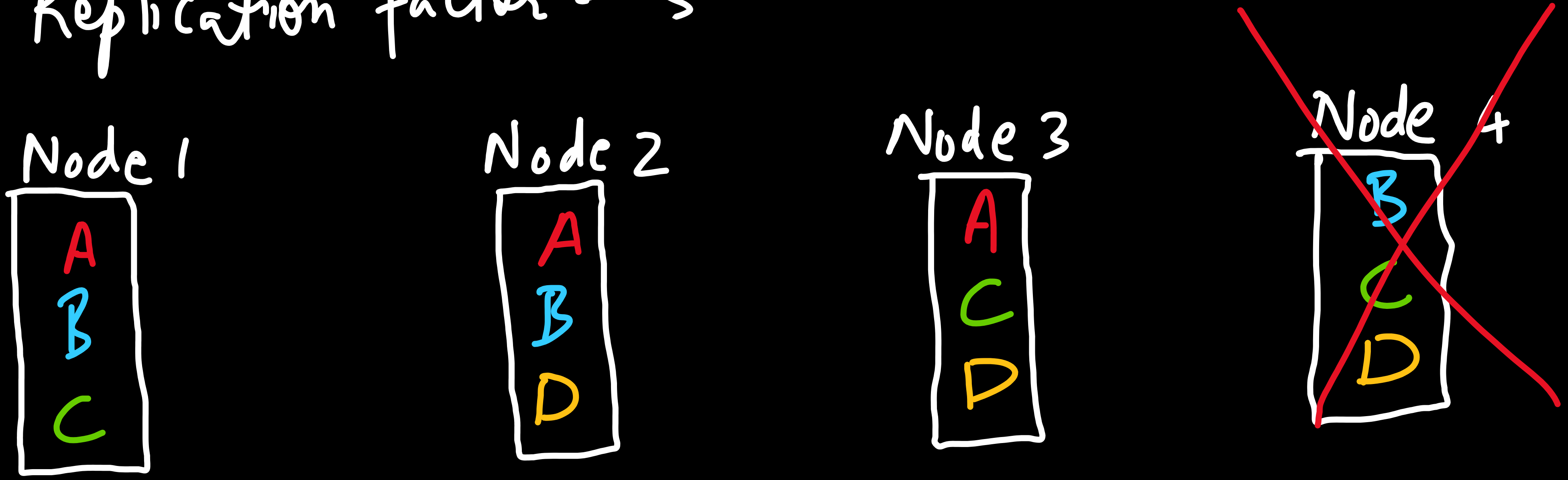
- Communication done using broadcast coded transmissions (linear combination of data symbols).
- The communication cost is reduced by a multiplicative factor.
- The time to rebalance is also reduced.

Example

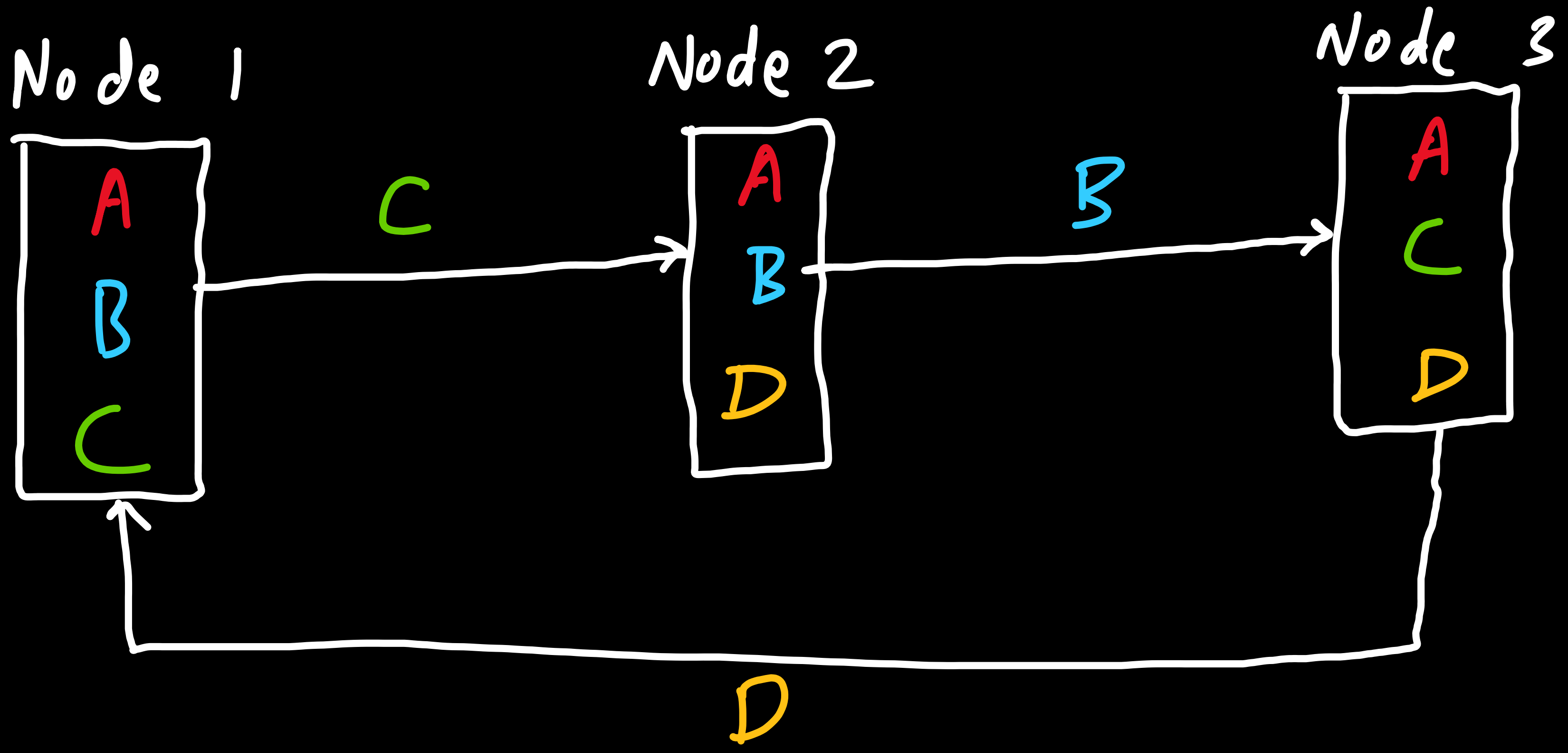
File : ABCD

Nodes : 4

Replication factor = 3

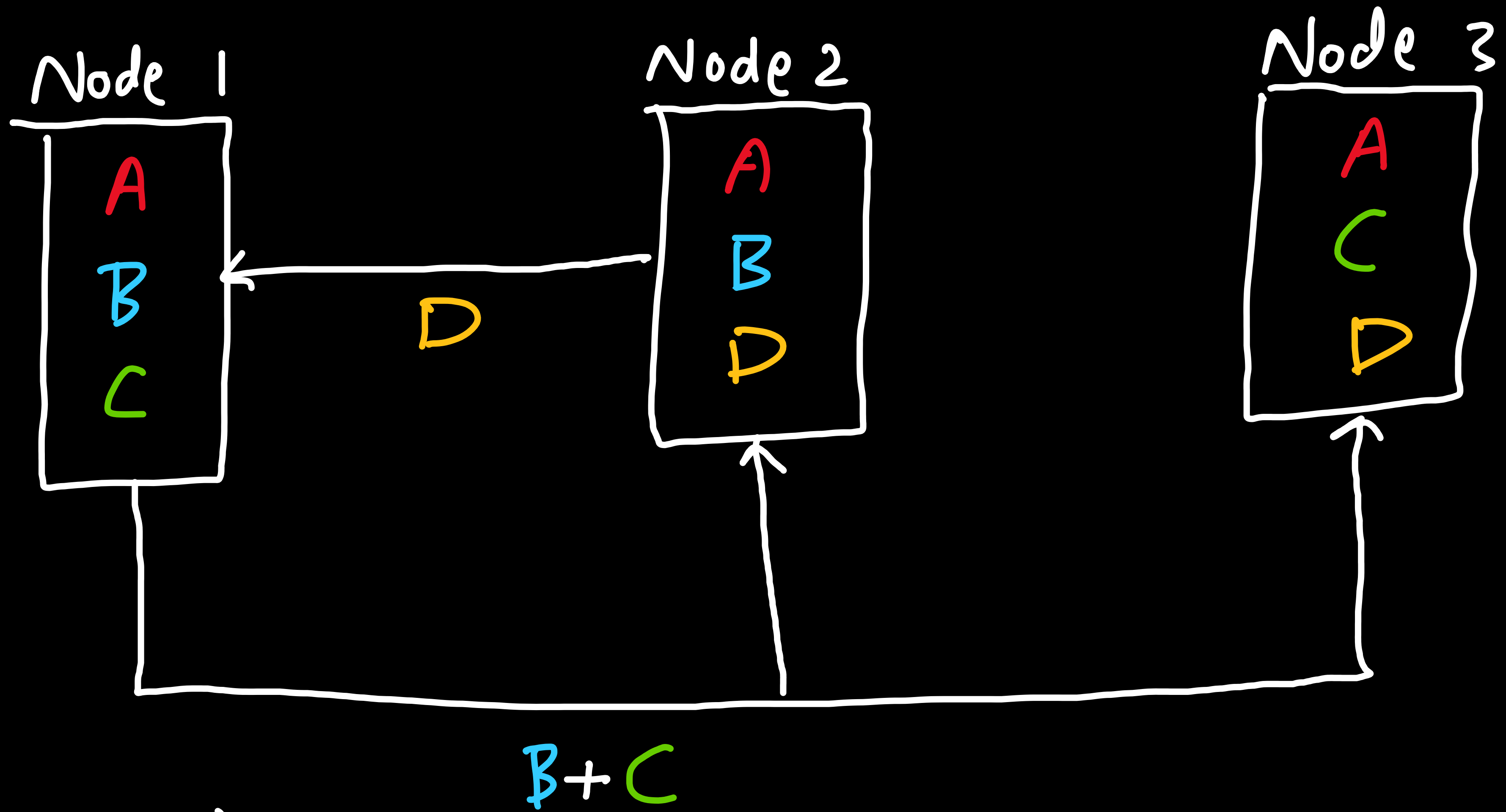


Normal Data Rebalancing



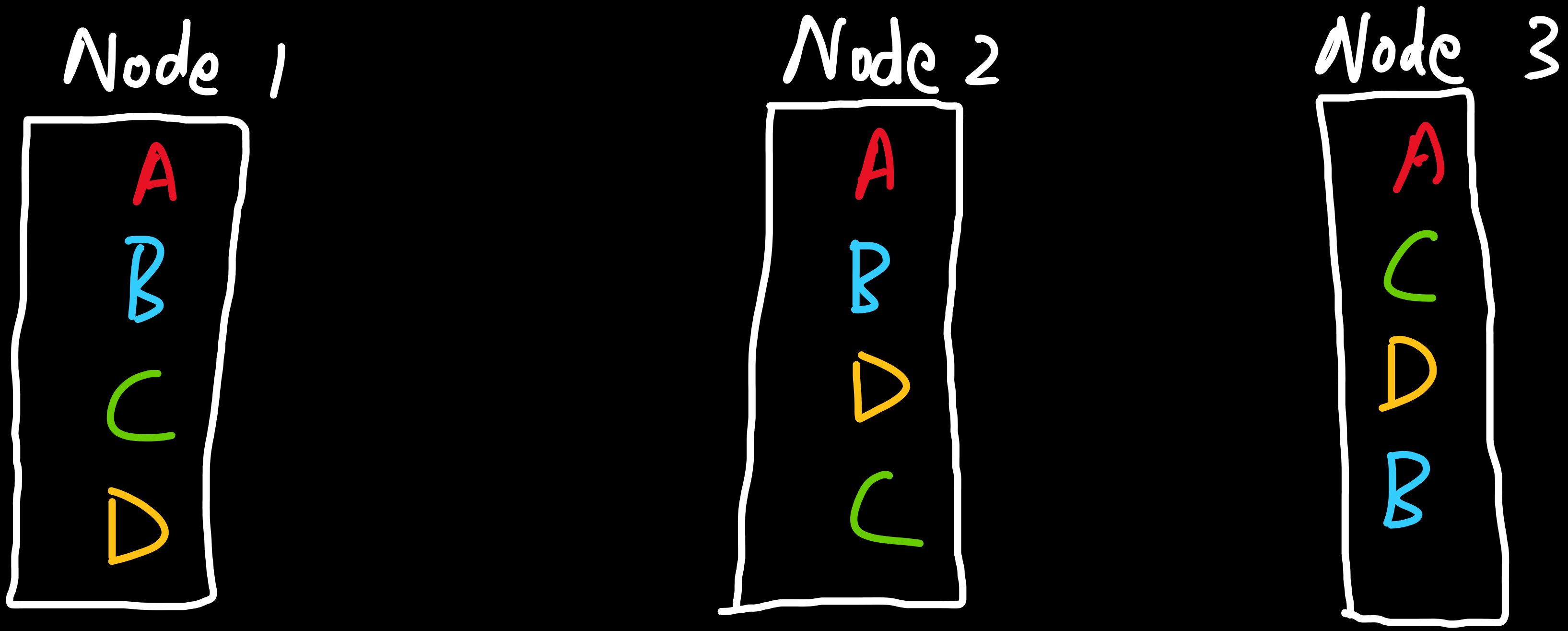
3 transmissions

Coded Data Rebalancing



2 transmissions

System after Rebalancing



System Model and Definitions

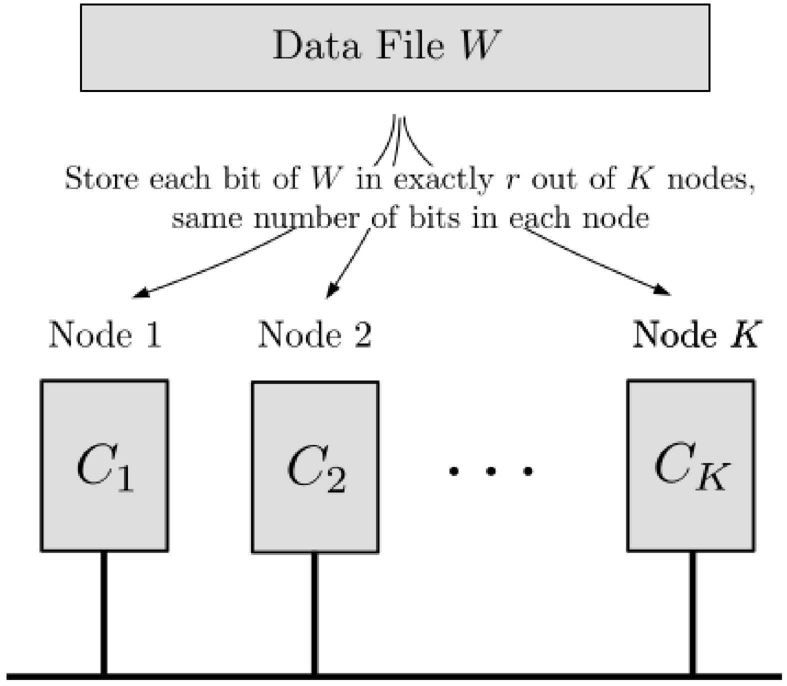


Fig. 1. An r-balanced distributed database across K nodes. The storage nodes are connected by a shared communication link.

Data File $W \rightarrow N$ bits

n^{th} bit $\rightarrow w_n \in \{0, 1\}$

$$W = \{w_n : n \in [N]\}, \quad [N] = \{1, \ldots, N\}$$

Nodes $\rightarrow [K] = \{1, 2, \ldots, K\}$

Database $\rightarrow C = \{C_i \subseteq W, i \in [K]\}, \quad \bigcup_{i \in [K]} C_i = W$

r-balanced database

$$C(r, [K]) = \{ C_i \subseteq N, \ i \in [K] \}, \text{ such that}$$

(i) $r_i[K] = r, \ \forall \ i \in [N]$

(ii) $|C_1| = |C_2| = \dots \dots = |C_K|$

any $|C_i| = \lambda N$, where $\lambda = \frac{r}{K}$ → storage fraction

$|C_i|$ ↓ number of bits in each node

Node Removal

Suppose k^{th} node is removed / failed.

Target Database : $C(r, [K] \setminus k)$

New Storage Fraction: $\lambda_{rem} = \frac{r}{K-1}$

$$|C_i| = \lambda_{rem} N \quad , \quad i \in \{1, \ldots, K-1\}$$

Rebalancing Load

Total number of bits transmitted normalized by the number of bits in the removed node.

Rebalancing Load (Uncoded)

Rebalancing Load will be at least 1, as all the symbols in the failed node must be communicated directly between the nodes.

Main Result

Rebalancing load $= \frac{1}{r-1}$

An example illustrating the scheme

$K = 5, \quad r = 3$

$W \rightarrow$ divided into $P(K, K-r)$ subfiles $\quad \left(P(K, r) = \frac{K!}{(K-r)!} \right)$

$P(5, 2) = \frac{5!}{3!} = 20$ subfiles

Indexing of subfiles $\rightarrow$ $(K-r)$ sized subsets of $\{1, 2, \dots, 5\}$

Subfiles: $W_{[12]}$, $W_{[13]}$, $W_{[14]}$, $W_{[15]}$,

$W_{[21]}$, $W_{[23]}$, $W_{[24]}$, $W_{[25]}$,

$W_{[31]}$, $W_{[32]}$, $W_{[34]}$, $W_{[35]}$,

$W_{[41]}$, $W_{[42]}$, $W_{[43]}$, $W_{[45]}$,

$W_{[51]}$, $W_{[52]}$, $W_{[53]}$, $W_{[54]}$

Node i contains all the subfiles that do not have 'i' in the index vector.

For example, node 1 contains:

$W_{[2\,3]}$, $W_{[2\,4]}$, $W_{[2\,5]}$

$W_{[3\,2]}$, $W_{[3\,4]}$, $W_{[3\,5]}$

$W_{[4\,2]}$, $W_{[4\,3]}$, $W_{[4\,5]}$

$W_{[5\,2]}$, $W_{[5\,3]}$, $W_{[5\,4]}$

Total subfiles in each node : $P(K-1, K-r) = \frac{(K-1)!}{(r-1)!} = \frac{4!}{2!} = 12$

On removing node 5

- In uncoded case, number of transmissions required will be the number of subfiles present in node 5, i.e., 12.
- For coded rebalancing, the same will be 6.

$$\frac{1}{n-1} \times 12 = \frac{1}{2} \times 12$$

Divide the subfiles in node 5 into 4 disjoint groups:

$$G_{[1]} = \{ W_{[21]}, W_{[31]}, W_{[41]} \}$$

$$G_{[2]} = \{ W_{[12]}, W_{[32]}, W_{[42]} \}$$

$$G_{[3]} = \{ W_{[13]}, W_{[23]}, W_{[43]} \}$$

$$G_{[4]} = \{ W_{[14]}, W_{[24]}, W_{[34]} \}$$

- For any group, the set of nodes associated with that group will be the first elements of the index vectors.
  - For example, in case of node 4, this set will be {1,2,3}.
- Every subfile in a group is present in exactly 2 nodes and is missing at the node corresponding to the first index of that subfile.
  - For example, $W_{[1\ 4]}$ is present in nodes 2 and 3.
- According to the scheme, a particular subfile is to be sent to a node in which it is not present, i.e., the first index of that subfile.
- For group 4, the subfiles $W_{[1\ 4]}$, $W_{[2\ 4]}$, $W_{[3\ 4]}$ will be sent to nodes 1, 2, and 3 respectively.

Data Exchange Protocol

Split the subfiles of a group into 2 parts
↳ $(r-1)$ parts

$$W_{[1\ 4]} \rightarrow W_{[1\ 4],2}\ ,\ W_{[1\ 4],3}$$

$$W_{[2\ 4]} \rightarrow W_{[2\ 4],1}\ ,\ W_{[2\ 4],3}$$

$$W_{[3\ 4]} \rightarrow W_{[3\ 4],1}\ ,\ W_{[3\ 4],2}$$

Broadcasting

Node 1 broadcasts: $W_{[2\ 4],1} \oplus W_{[3\ 4],1}$

Node 2 broadcasts: $W_{[1\ 4],2} \oplus W_{[3,4],2}$

Node 3 broadcasts: $W_{[1\ 4],3} \oplus W_{[2,4],3}$

Node 1 has: $W_{[24],1}$, $W_{[24],3}$, $W_{[34],1}$, $W_{[34],2}$

Node 2 has: $W_{[14],2}$, $W_{[14],3}$, $W_{[34],1}$, $W_{[34],2}$

Node 3 has: $W_{[14],2}$, $W_{[14],3}$, $W_{[24],1}$, $W_{[24],3}$

Node 1 receives: (i) $W_{[14],2} \oplus W_{[34],2}$

(ii) $W_{[14],3} \oplus W_{[24],3}$

Total size of transmissions for group 4:

$$3 \times \frac{1}{2} = \frac{3}{2}^{\text{rds}} \text{ of the size of a subfile}$$

3 → 3 nodes

$\frac{1}{2}$ → splitting into 2 parts

Total size of transmissions for all groups:

$$\frac{3}{2} \times 4 = 6 \left(= \frac{1}{2} \times 12 \right)$$

4 → No. of groups

2 → (3 - 1)

Example (Re-indexing and Structural Invariance)

- Since there are K-1 nodes now, the size of index vectors will be K-r-1 for the system after node removal and rebalancing.

$$W_{[1]} = \{ W_{[21]}, W_{[31]}, W_{[41]}, W_{[15]}, W_{[51]} \}$$

$$W_{[2]} = \{ W_{[12]}, W_{[32]}, W_{[42]}, W_{[25]}, W_{[52]} \}$$

$$W_{[3]} = \{ W_{[13]}, W_{[23]}, W_{[43]}, W_{[35]}, W_{[53]} \}$$

$$W_{[4]} = \{ W_{[14]}, W_{[24]}, W_{[34]}, W_{[45]}, W_{[54]} \}$$

Node 1 : $W_{[2]}, W_{[3]}, W_{[4]}$

Node 2 : $W_{[1]}, W_{[3]}, W_{[4]}$

Node 3 : $W_{[1]}, W_{[2]}, W_{[4]}$

Node 4 : $W_{[1]}, W_{[2]}, W_{[3]}$

General Case

File W of N bits → divided into $P(K, K-r)$ subfiles of same size $\left(\text{i.e, } \frac{N}{P(K, K-r)} \right)$

Indexing : $(K-r)$ sized vectors

Let A_K be the set of all such vectors.

Suppose, node k is removed.

The subfiles of node k are to be divided into groups.

$$G_{i'} = \{ i \in A_k \mid [i_2\ i_3 \ldots i_{K-r}] = i' \}$$

Example: $K = 5$, $r = 2$, $K - r = 3$

$$G_{[2\ 4]} = \{ W_{[1\ 2\ 4]}, W_{[3\ 2\ 4]} \}$$

$[2\ 4]$: $K-r-1$ sized

Groups indexed by vectors of size $(K-r-1)$

Total number of elements in a group:

$$\underbrace{K-1}_{\text{nodes}} - (\underbrace{K-r-1}_{\text{index vector size}}) = r$$

Total number of groups: $P(K-1, K-r)/r$

Each subfile of a group present in : $(r-1)$ nodes

Next Steps

- Each file not present at only the first index, so it needs to be sent to that node.
- Divide the subfiles of a group into (r-1) parts and index them with the nodes that they are present in. (recall from the example)
- Broadcast coded symbols.

Total Communication Cost

Splitting ← $\frac{1}{r-1}$ × r (No of nodes) × $\frac{P(K-1, K-r)}{r}$ (No. of groups) × $\frac{N}{P(K, K-r)}$ (Size of each subfile)

$$\frac{1}{r-1} \times \cancel{r} \times \frac{P(K-1, K-r)}{\cancel{r}} \times \frac{N}{P(K, K-r)}$$

$$= \frac{P(K-1, K-r)}{r-1} \times \frac{N}{P(K, K-r)}$$

$$= \frac{Nr}{(r-1)K} = \frac{N\lambda}{r-1} = \frac{1}{r-1} \times N\lambda$$

Structural Invariance

- The system after rebalancing has K-1 nodes, so the size of index vectors will be K-r-1 for the system after node removal and rebalancing.

For any $(K-r-1)$ sized vector i', there exist distinct $j_1, j_2, j_3, \ldots, j_r$ such that

$j_1, j_2, j_3, \ldots, j_r \notin i'$.

Re-indexed subfile $W_{i'}$ → concatenation of k original subfiles

$[j_1\ i'],\ [j_2\ i'],\ \ldots,\ [j_r\ i'],\ \rightarrow r$

$$\left.\begin{array}{l}[k\ i'_1\ i'_2 \ldots i'_{k-r-1}],\ [i'_1\ k\ i'_2 \ldots i'_{k-r-1}], \\ \ldots,\ [i'_1\ i'_2 \ldots i'_{k-r-1}\ k]\end{array}\right\}\ k-r$$

Conclusion

- We saw a scheme that rebalances data upon node failure with a rebalancing load of 1/r-1.
- This scheme not only maintains the r-balanced property but also preserves the same essential structure with the help of reindexing.

Questions?