Codes for Distributed Storage

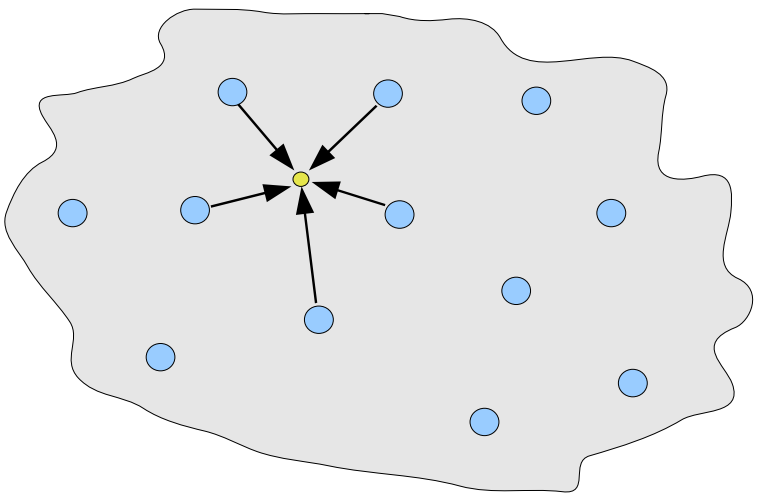
(Introduction to Some Representative Results from Ph.D. Thesis)

Birenjith Sasidharan and P. Vijay Kumar

Department of Electrical Communication Engineering,
Indian Institute of Science, Bangalore.

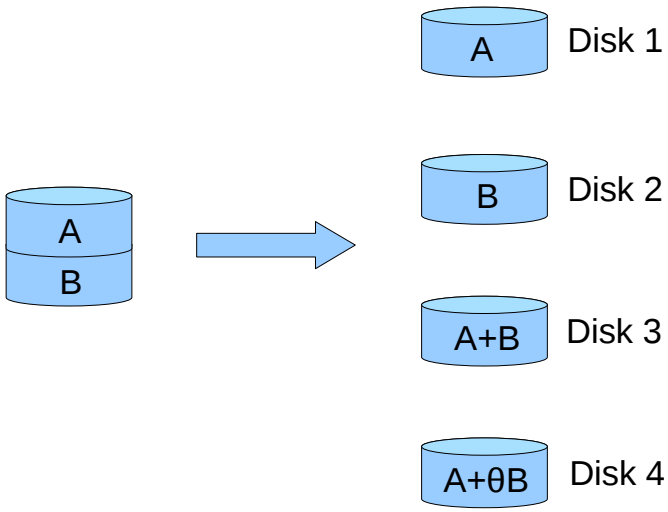
EECS Research Students Symposium - 2016
April 28-29, 2016

Distributed Data Storage Network



- Information pertaining to a file is dispersed across nodes (one or more disks) in the data center
- Nodes are prone to failure.
- Error-correcting codes (such as Reed-Solomon codes) are employed to combat erasures.

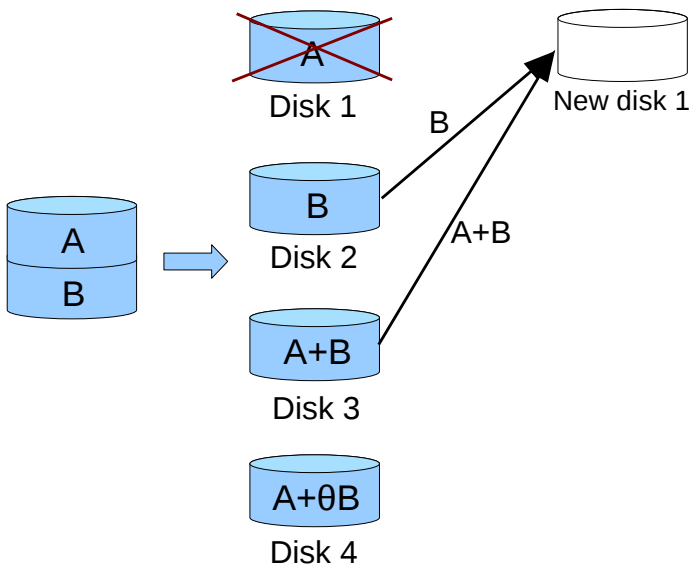
An Example: The RAID Code



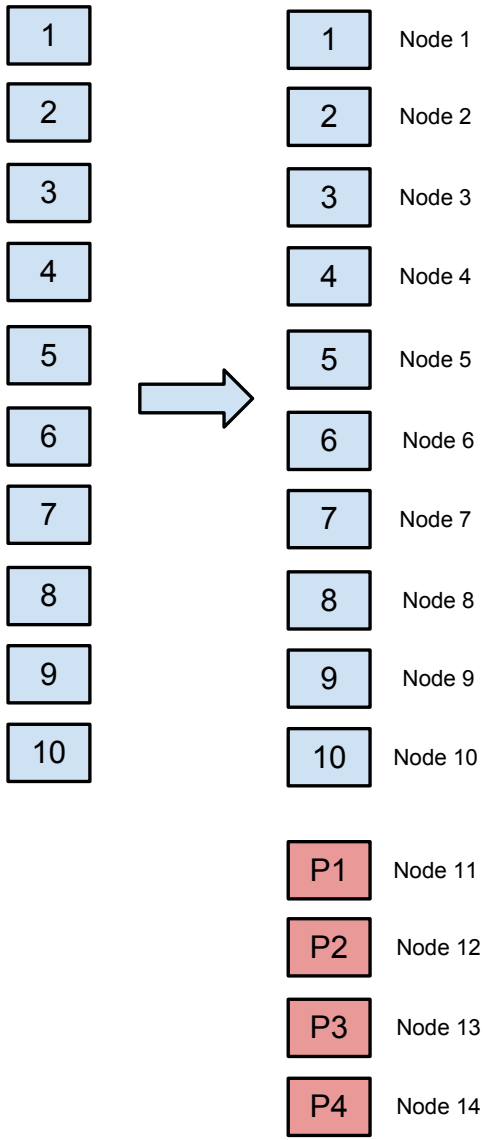
- $[4, 2]$ Reed-Solomon code
- Can recover data by connecting to any 2 nodes.

Handling node repair:

- Reconstruct entire data;
- And then, reconstruct node-data.
- Download 2 blocks to repair 1 block.



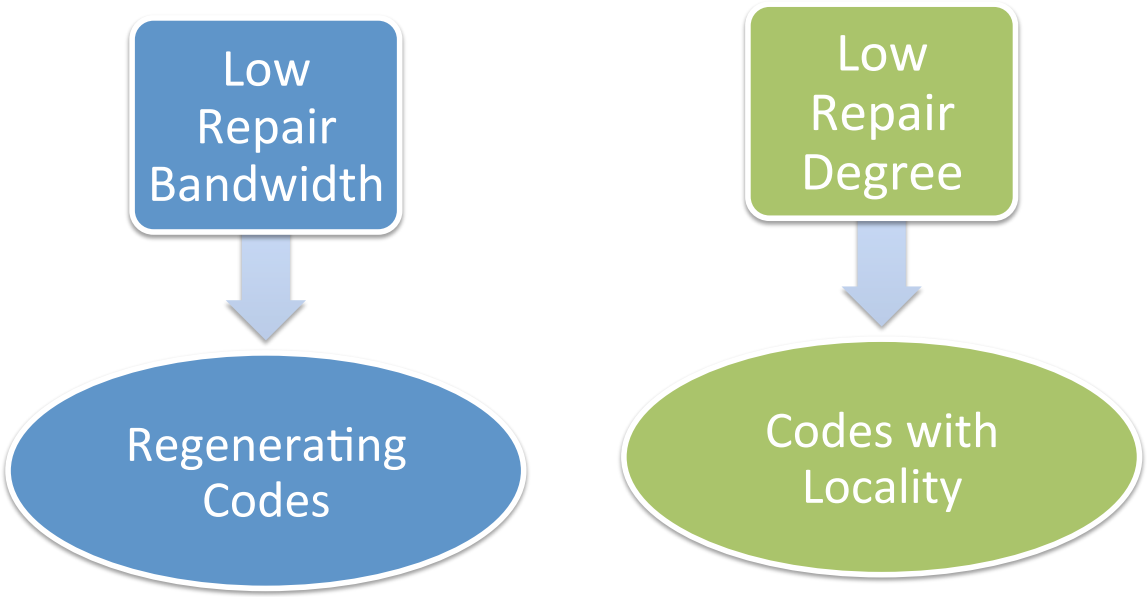
A Second Example: Facebook's HDFS-RAID Code



- $[14, 10]$ RS code
- Can recover data by connecting to any 10 nodes
- Used in Facebook data centres
- HDFS $\equiv$ Hadoop Distributed File System
- Again, repair of a single node requires to access 10 nodes.

D. Borthakur, R. Schmit, R. Vadali, S. Chen, and P. Kling. "HDFS RAID." Tech talk. Yahoo Developer Network, Nov. 2010

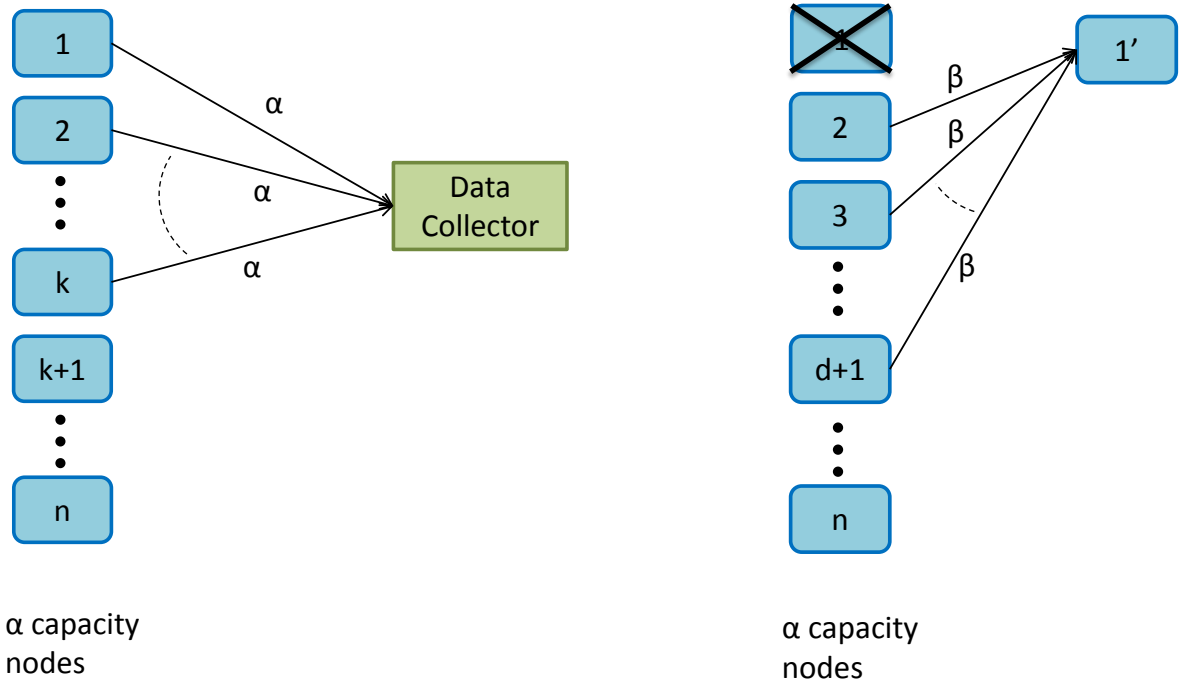
Two Problems – Two Solutions



- A. G. Dimakis, P. B. Godfrey, Y. Wu, M. Wainwright, and K. Ramchandran, "Network Coding for Distributed Storage Systems," *IEEE Trans. Inform. Th.*, Sep. 2010.
- **P. Gopalan, C. Huang, H. Simitci, and S. Yekhanin, "On the Locality of Codeword Symbols," IEEE Trans. Inf. Theory, Nov. 2012.**

Framework of Regenerating Codes

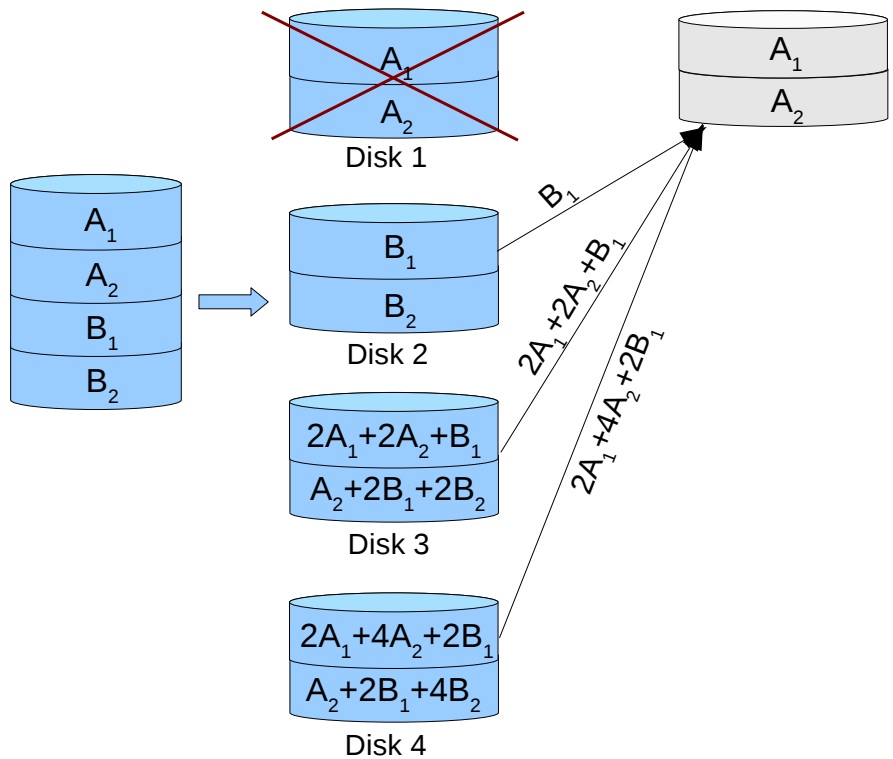
Parameters: $([n, k, d], [\alpha, \beta], B, \mathbb{F}_q)$



- Data Collection: Connect to any k nodes to retrieve file of size B.
- Nodes Repair: Connect to d nodes, download β symbols from each, to recover any failed node.
- The characterization was first done for "functional repair" – in which the data in the replacement node need not exactly match with that of failed node, but with potentially different symbols so that data collection & node repair still hold good. This made the theory easier, but we will stick to "exact repair" here.

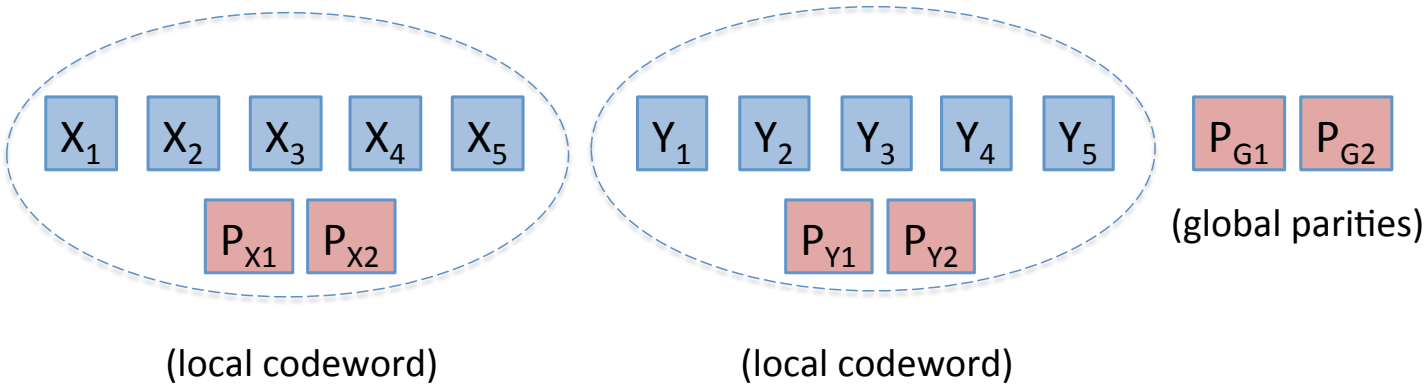
Regenerating Code: An Example

- $(n = 4, k = 2, d = 3)$,
- sub-packetization $\alpha = 2$, repair-bandwidth per-node $\beta = 1$.
- File size, $B = 4$.
- Repair Bandwidth, $d\beta = 3$.
- This is an example of "MSR" code.

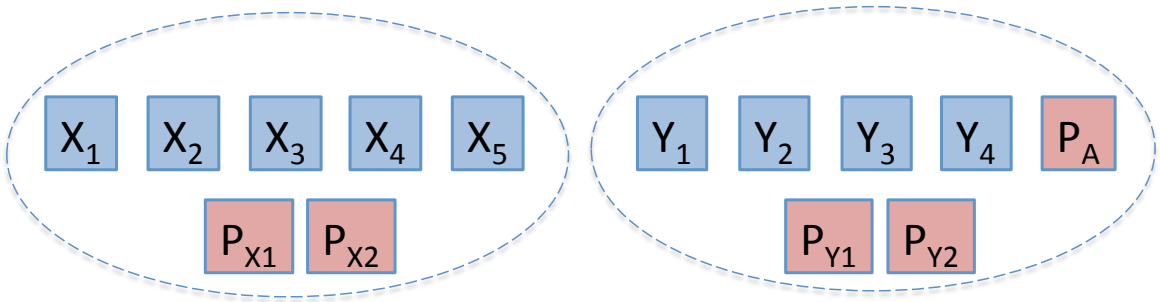


We came up with a construction having sub-packetization level that scales polynomial in k (as opposed to exponential scaling available till then.)

Codes with Locality

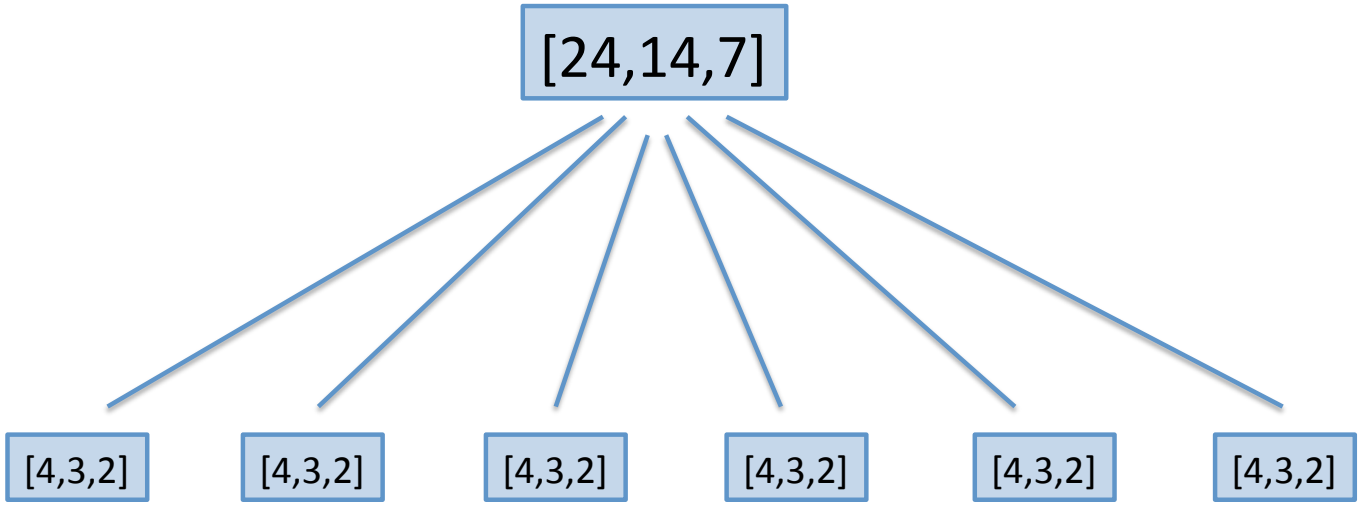


(Information-Symbol locality)



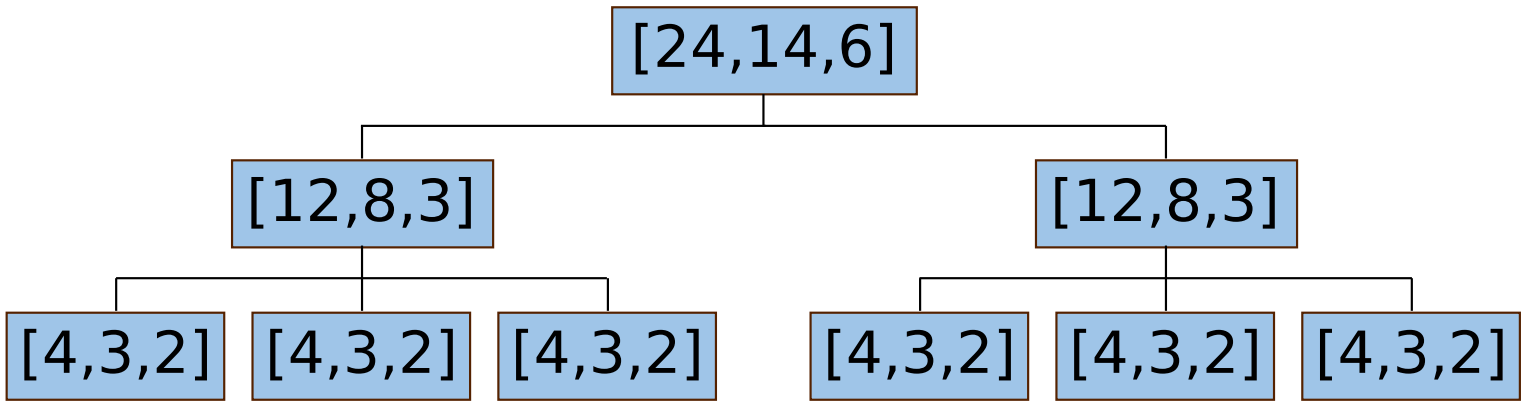
(All-symbol locality)

Codes with Locality do not Scale



- If the local code is overwhelmed, then one has to appeal to the overall code which means contacting all 14 nodes for node repair.
- Is it possible to build a code where the repair degree increases gradually as opposed to in a single jump ?

Codes with Hierarchical Locality



- Codes with hierarchical locality do exactly that by calling for help from an intermediate layer of codes when the local code fails.
- We defined such classes of codes, and gave "optimal" constructions.

Please visit the poster session for more details on these code constructions!

Thank you!