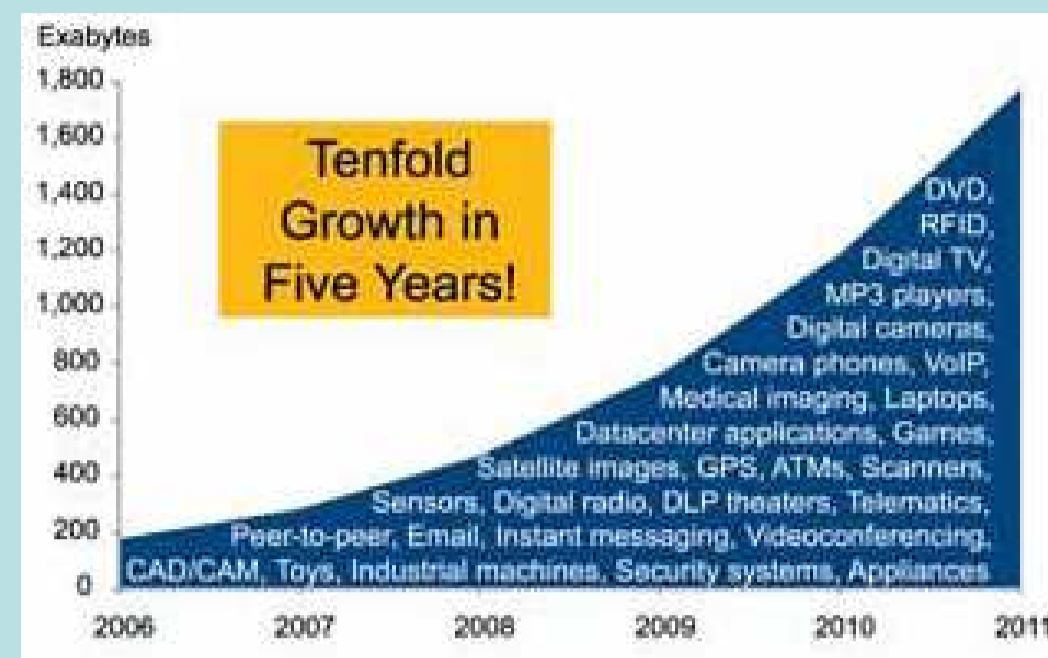
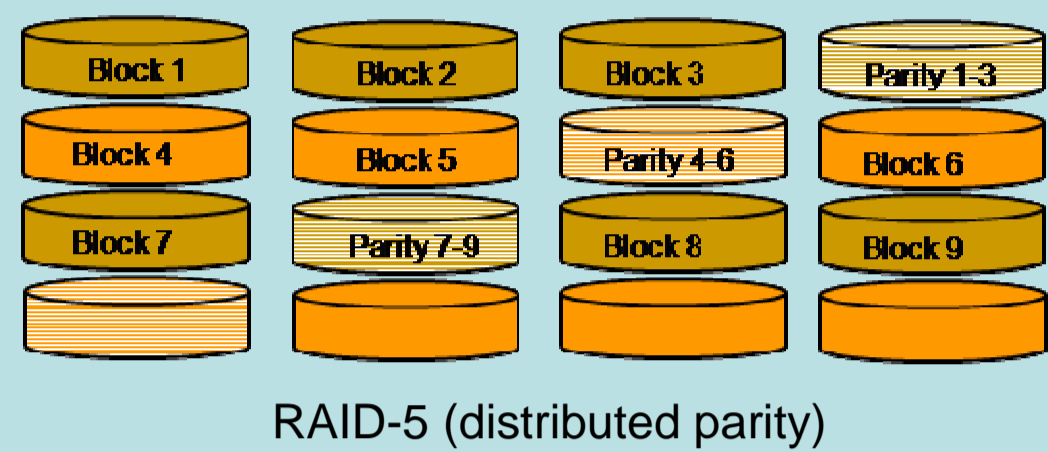


Motivation

- Amount of data created each year is increasing exponentially; **need systems that can scale out**
- Disk capacity is increasing every year; **sector errors more probable**
- Number of disks in a system is increasing; multiple disk failures more probable, but **RAID-5 or RAID-6 designed to sustain only up to one or two disk failures**
- New storage technologies like flash, phase-change memory; **different error patterns, speed, etc.**
- Coding theory traditionally used in increasing reliability; how can it be used in large-scale storage systems where **not only reliability, but performance, storage efficiency, and energy efficiency matter?**



Growth of newly created data each year (Source: International Data Corporation, 2008 [5])

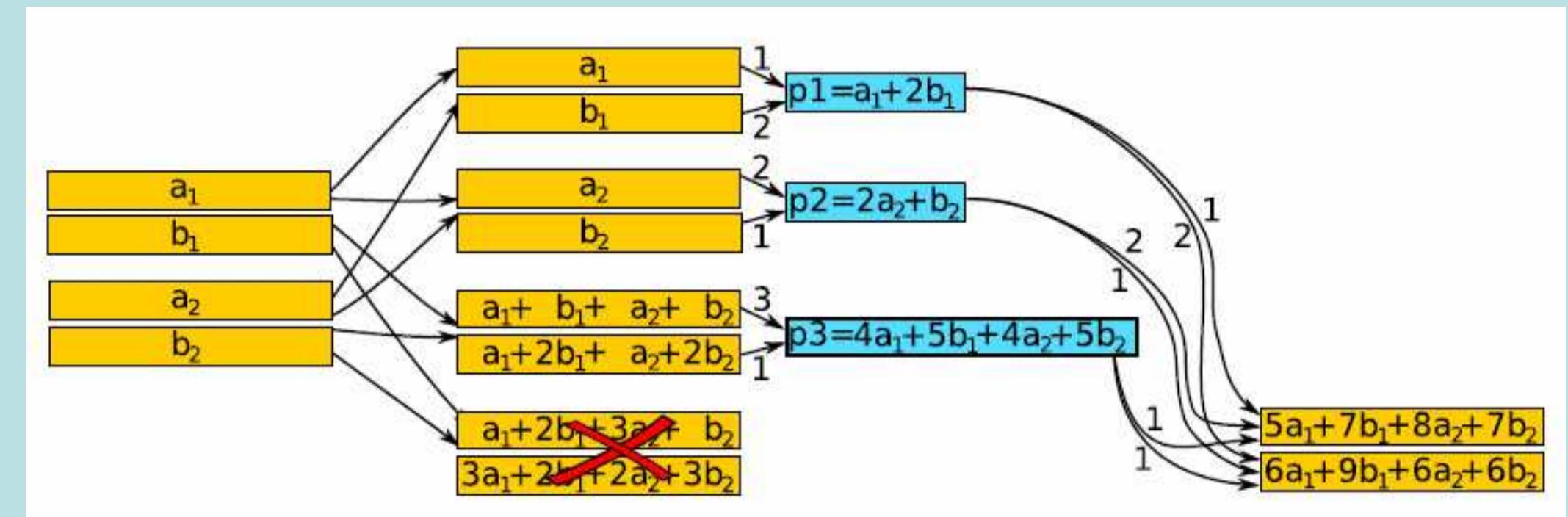


RAID-5 (distributed parity)

Coding Scheme	Storage Efficiency (ρ)	IOPS _{disk}	MBPS _{disk}	MTTDL
RAID-5	$\frac{n-1}{n}$	$\frac{1}{2} \text{IOPS}_{\text{disk}}$	$(n-1) \text{MBPS}_{\text{disk}}$	$\frac{\mu'}{n(n-1)\lambda^2}$
RAID-6	$\frac{n-2}{n}$	$\frac{1}{3} \text{IOPS}_{\text{disk}}$	$(n-2) \text{MBPS}_{\text{disk}}$	$\frac{\mu'}{n(n-1)(n-2)\lambda^3}$
Distributed Replication	$\frac{1}{2}$	$\frac{1}{2} \text{IOPS}_{\text{disk}}$	$\frac{1}{2} \text{MBPS}_{\text{disk}}$	$\frac{\mu'}{n\lambda^2}$
Regenerating Codes	$\frac{k}{n}$	$\frac{k}{n} \text{IOPS}_{\text{disk}}$	$k \text{MBPS}_{\text{disk}}$	$\frac{\mu'^k}{n(n-1)\dots(n-k)\lambda^{k+1}}$

Comparison of various coding schemes across different metrics; here, n is the number of disks, $\text{IOPS}_{\text{disk}}$ is the maximum number of I/Os per disk, $\text{MBPS}_{\text{disk}}$ is the write bandwidth of the disk, MTTDL stands for mean time to data loss, $1/\mu'$ is the mean time to repair a disk, and $1/\lambda$ is the mean time to failure of a disk, and k/n is the rate of the regenerating code [3].

Network Coding for Storage Systems

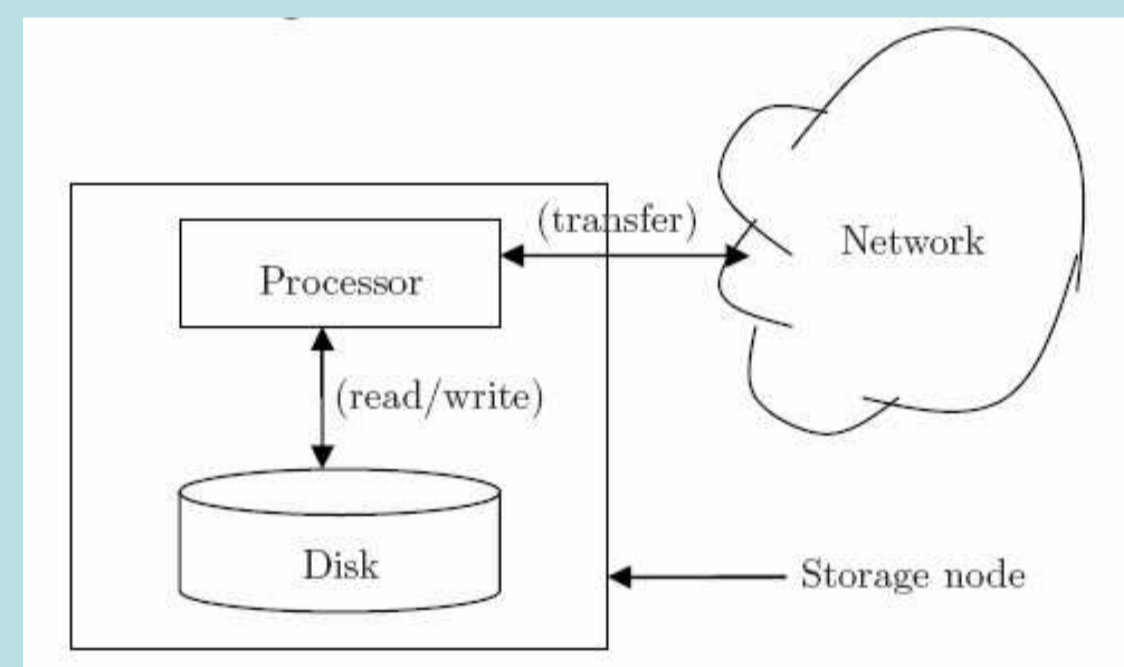


Network coding can reduce the amount of data transferred over the storage network from the surviving nodes to the new node during rebuild process (Source: [1])

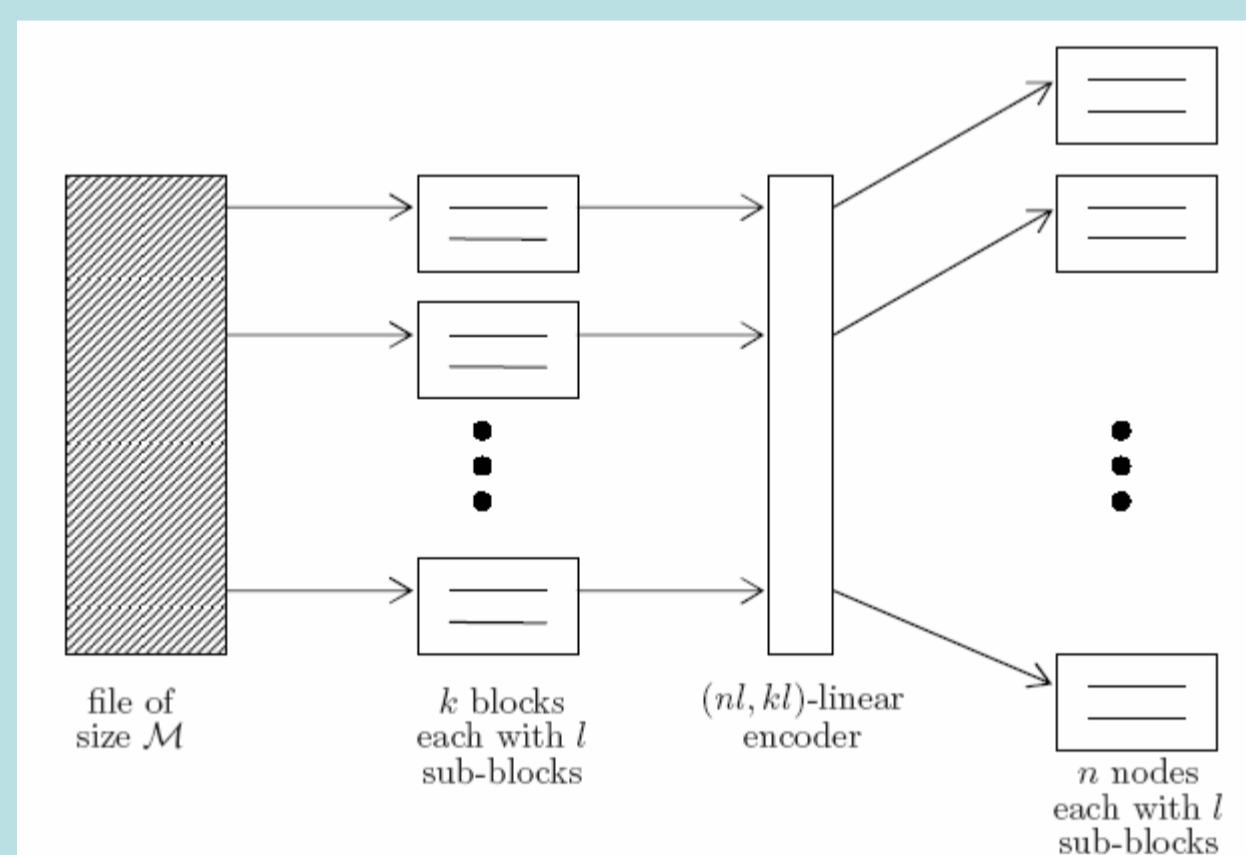
- When a storage node fails, a rebuild process is initiated
- Rebuild involves replacing the failed node with a new node and recreating lost information from the surviving nodes
- Network coding can help minimize the amount of information transferred during rebuild thereby using minimum network bandwidth [1]

Read as much as you transfer

- Codes described in [1], called **regenerating codes**, require to **potentially read all the information in a node** in order to transfer the required information to the new node
- In [2], it is shown that there exists a subset of regenerating codes, where **only what needs to be transferred needs to be read** thereby **reducing read time**, and **avoiding processing** at the each storage node
- In the proof, a storage system is considered where a file of size M is split into k blocks each with l subblocks and is encoded using a (n, k, l) linear code that has a block-wise MDS property
- It is then shown that there exists a code under certain conditions such that, when a node fails, the new node can be rebuilt from just the first subblock of each surviving node.

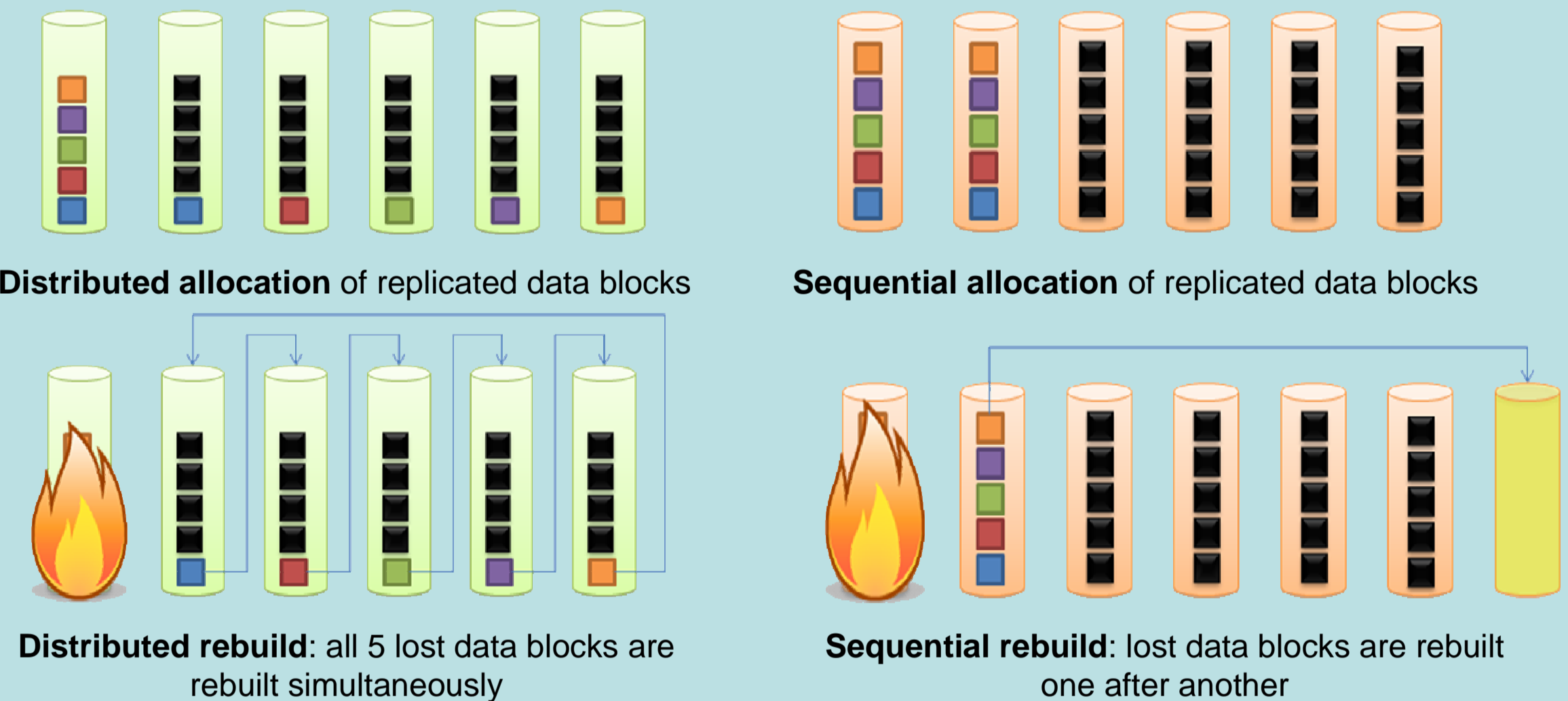


A storage node with disk, processor, and connection to the storage network



System model used in proof of existence of codes that require to read only as much as needed to transfer from each surviving node during rebuild

Effect of distributed rebuild on reliability

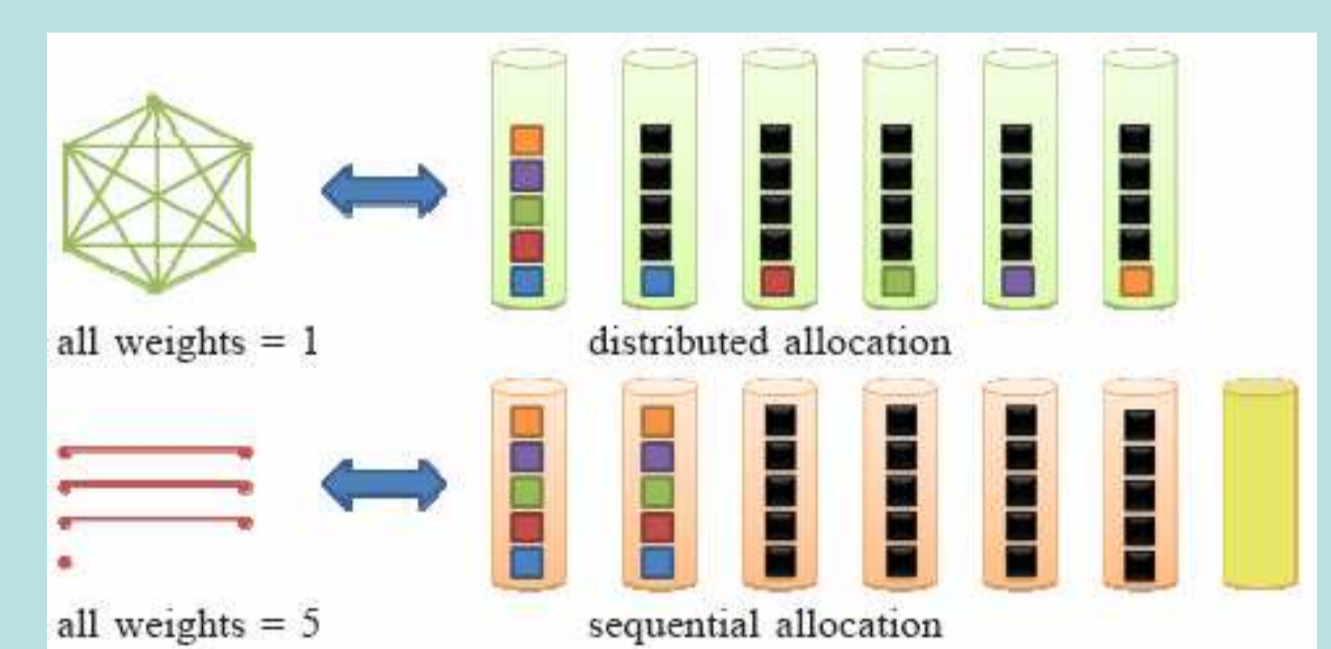


Distributed rebuild: all 5 lost data blocks are rebuilt simultaneously

Sequential rebuild: lost data blocks are rebuilt one after another

Even though distributed rebuild is several times faster than sequential rebuild, it can be shown that **both systems have the same probability of data loss during rebuild in a two-way replication based storage system** [4].

In [3], a graphical model for a replication based storage system is developed, where vertices represent storage nodes, and weights on edges represent number of shared data blocks between the nodes



Using this model and its extensions (in terms of hypergraphs), it can be shown that the probability of data loss during rebuild for distributed rebuild is lesser than the probability of data loss for sequential rebuild for replication factors greater than 2.

Future Directions

- Replicated systems: does replica placement affect reliability?
- What about placement of coded blocks in a general erasure coding scheme?
- Can hybrid storage systems with various storage technologies like flash, magnetic drives, and other future technologies like phase-change memories be used to deliver reliability without compromising on performance?
- A 2006 study by IDC [5] found that power consumption that was 1 kW per server rack in 2000 was closer to 10 kW in 2006 and that customers building new data centers are planning for 20 kW per rack. There is a need for energy efficient storage systems.

References

- [1] A. Dimakis, P. B. Godfrey, Y. Wu, M. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," in *Proc. of IEEE INFOCOM*, 2007.
- [2] V. Venkatesan, "Fast rebuilds in distributed storage systems using network coding," in *IBM Research Report*, RZ3741, 2009.
- [3] V. Venkatesan, "Performance metrics for distributed storage systems," in *Semester Project Report*, EPFL, 2009.
- [4] I. Iliadis and R. Haas, "Replication versus RAID for distributed storage systems," in *IBM Research Report*, RZ3733, 2009.
- [5] International Data Corporation, "The diverse and exploding digital universe," *White Paper*, 2008