Modern Erasure Codes for Distributed Storage Systems

Storage Developer Conference, SNIA, Bangalore

Srinivasan Narayanamurthy
Advanced Technology Group, NetApp

May 27th 2016

Everything around us is changing!

- The Data Deluge
  - Disk capacities and densities are increasing faster than the disk transfer rates
  - Increased delay to recover using classical techniques lead to availability exposure
- Changing Storage Technologies
  - Architectures: Scale-out, Distributed Storage, Cloud, Converged
  - Media: Flash, NVM, SMR, Tape, et al.
  - Features: Geo-distribution, Security, Use of commodity hardware (Failure is a norm!)
- Newer Dimensions of Erasure Codes
  - Optimality tradeoffs redefined
  - More about this inside…

> "Erasure coding usage is growing, and is now available in an increasing number of newer object, file and block storage arrays, but not in traditional general purpose disk arrays."
> – Gartner

Organization

Background

- Erasure Codes Timeline
- Classical Codes - (n, k) code

Modern Codes

- Codes on Codes
- Network Codes

Technical Analysis

- Optimality Tradeoff and Reliability Analysis
- System Requirements and Codes

Literature & Key Players

Background

Timeline – Classical (n, k) codes

Timeline – Overview

Classical Codes

Fountain Codes

Codes on Codes

Network Codes

Tradeoff against "*Storage Overhead*"

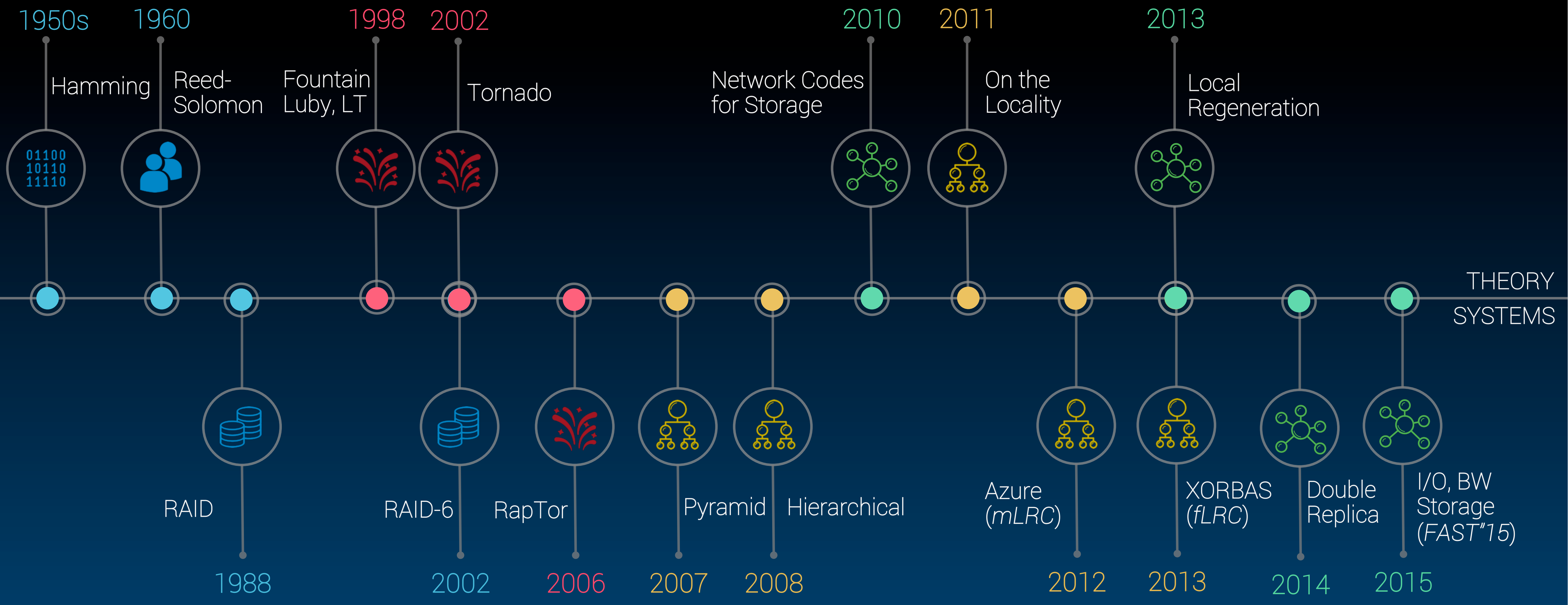
Reliability

Performance

Repair Degree

Repair Bandwidth

Erasure Codes Timeline



THEORY

1950s Hamming

1960 Reed-Solomon

1998 Fountain Luby, LT

2002 Tornado

2010 Network Codes for Storage

2011 On the Locality

2013 Local Regeneration

SYSTEMS

1988 RAID

2002 RAID-6

2006 RapTor

2007 Pyramid

2008 Hierarchical

2012 Azure (*mLRC*)

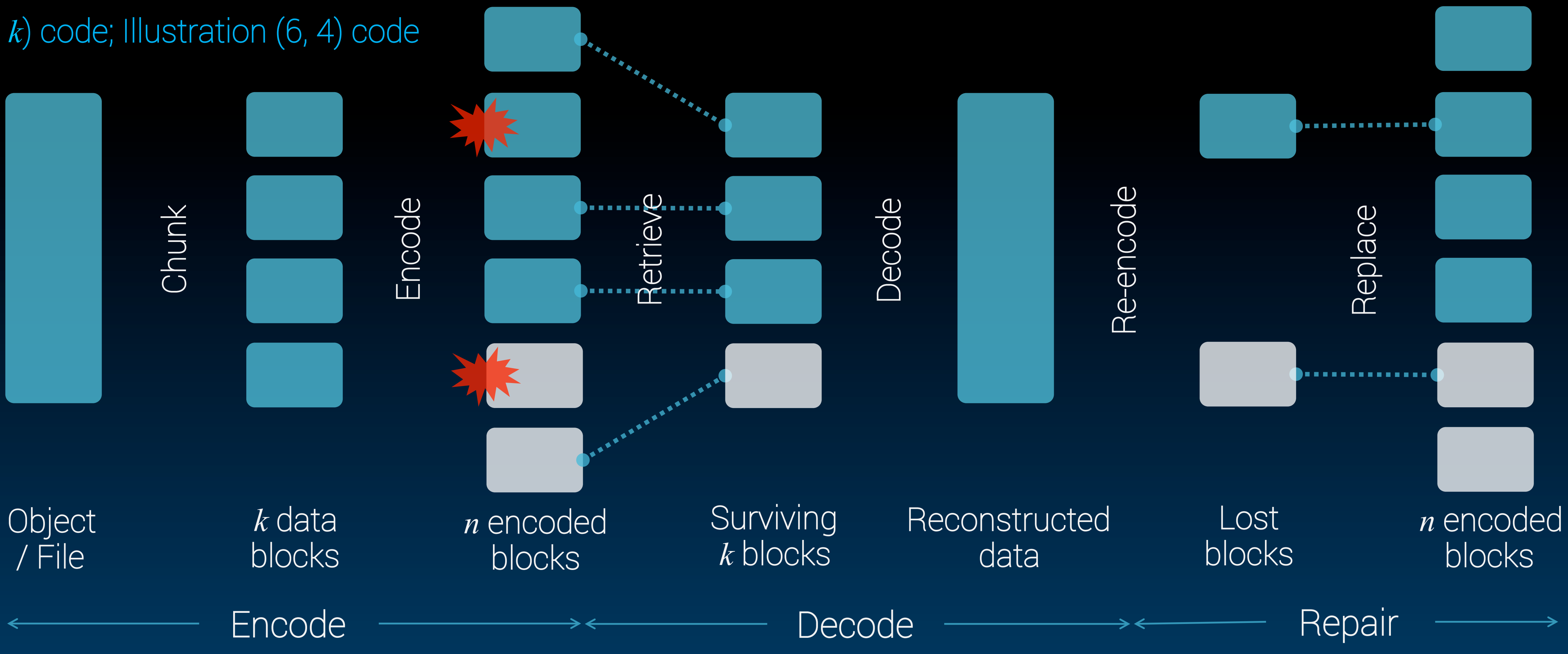
2013 XORBAS (*fLRC*)

2014 Double Replica

2015 I/O, BW Storage (*FAST"15*)

Classical Codes

(n, k) code; Illustration (6, 4) code



Think distributed systems; repairs are expensive !

Modern Codes

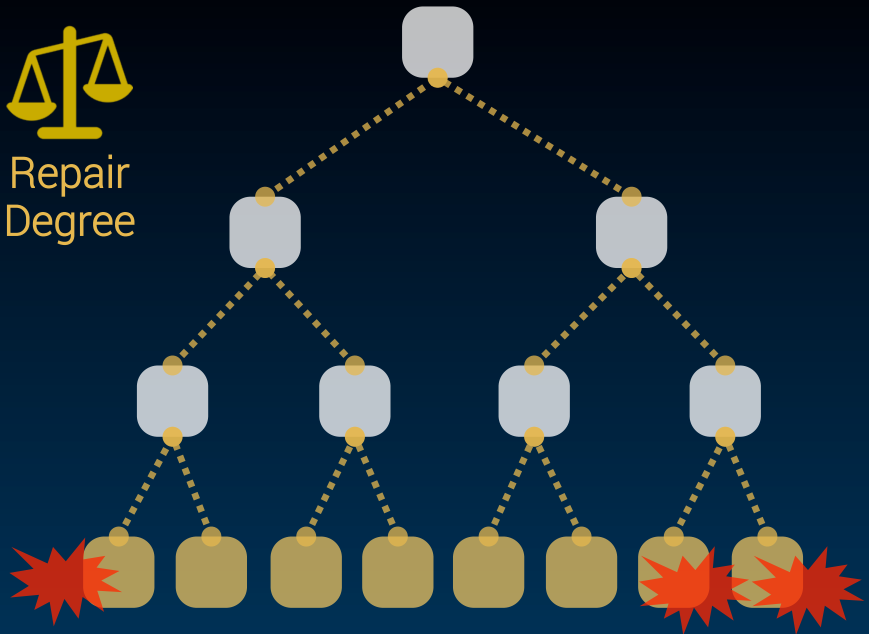
Codes on Codes – Network Codes

Codes On Codes

$(n_1, k_1) + (n_2, k_2)$ & (k, l, r) codes

Hierarchical & Pyramid Codes

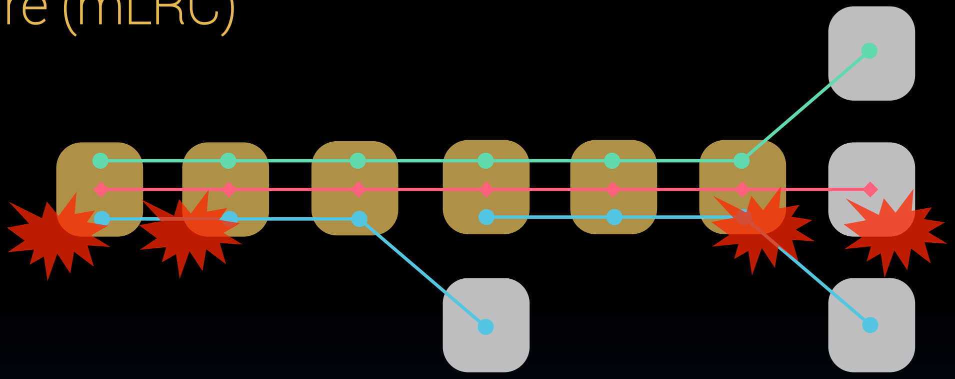
Repair Degree



Hierarchical – Bottom Up
Pyramid – Top Down

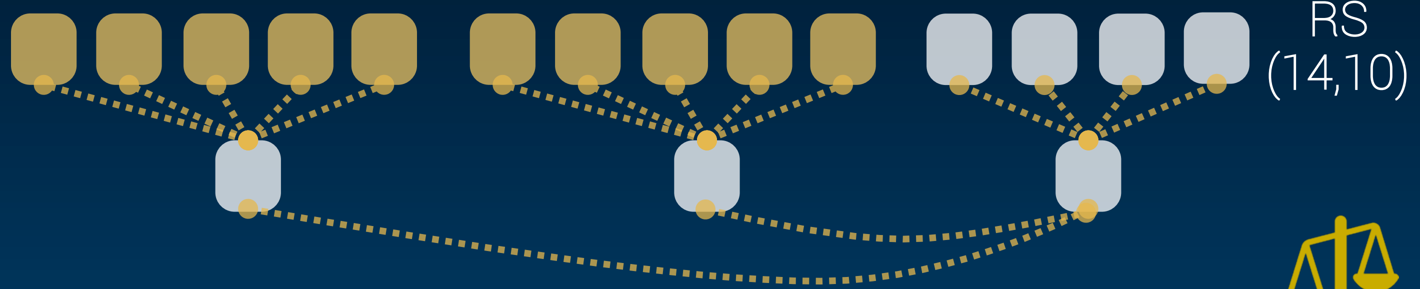
Azure (mLRC)

Locality/
Max. Recoverability



k=6 data fragments, l=2 local parities and r=2 global parities

Decoding 3 and 4 failures in mLRC



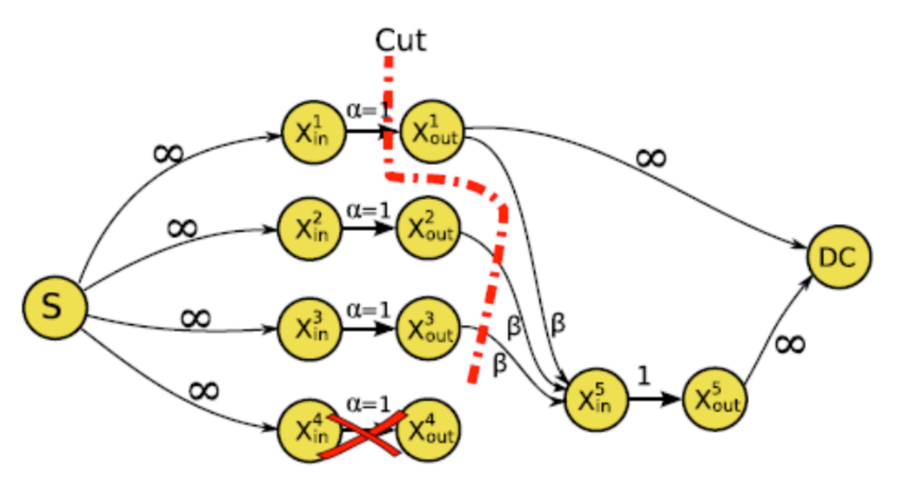
Facebook (XORBAS)

Single block failures

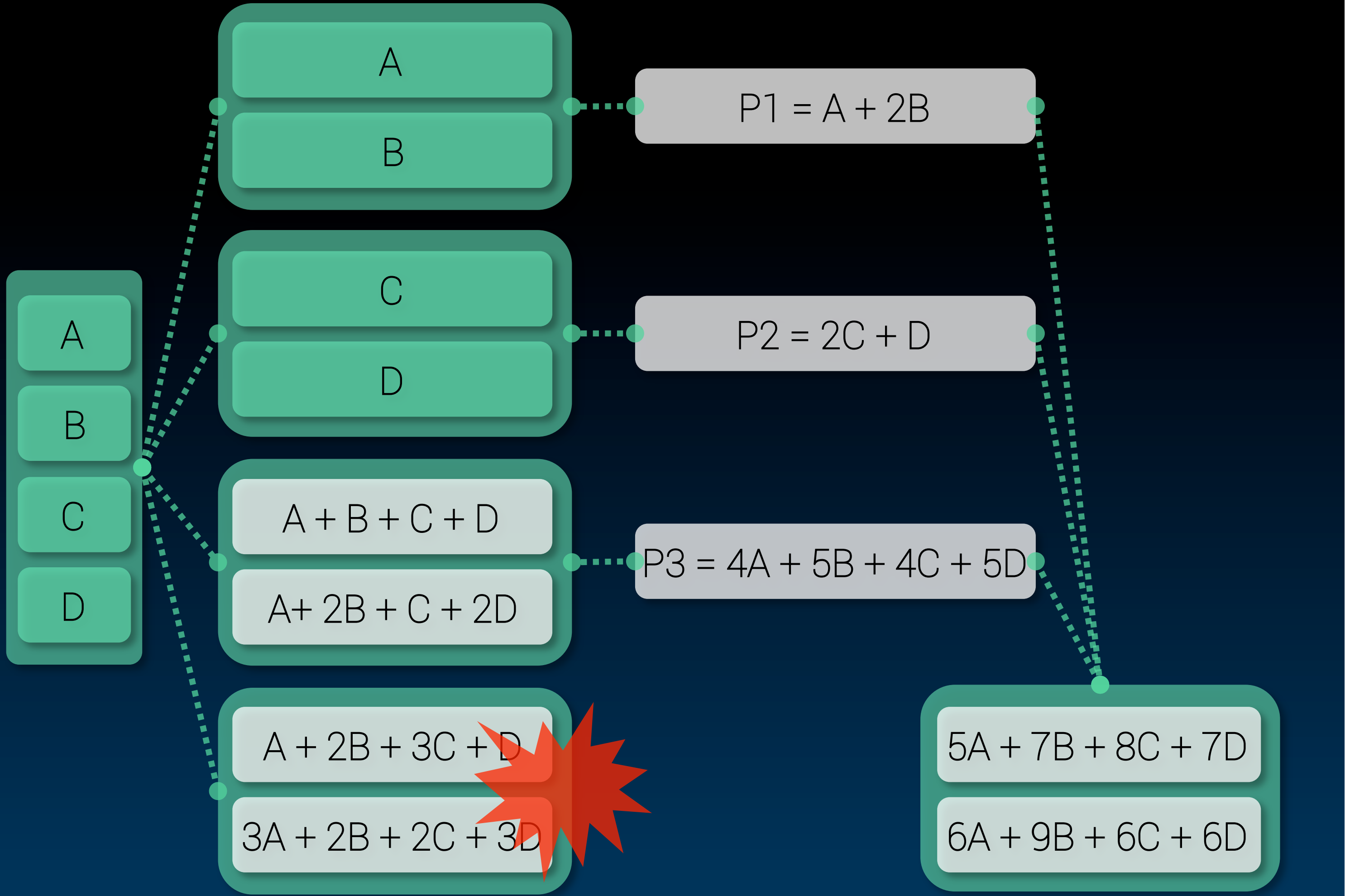
Locality/
Min. Distance

Regenerating Codes

Inspired by Network Codes



An Information Flow Graph & Min-Cut Bound


A
B
C
D

A
B

P1 = A + 2B

C
D

P2 = 2C + D

A + B + C + D
A+ 2B + C + 2D

P3 = 4A + 5B + 4C + 5D

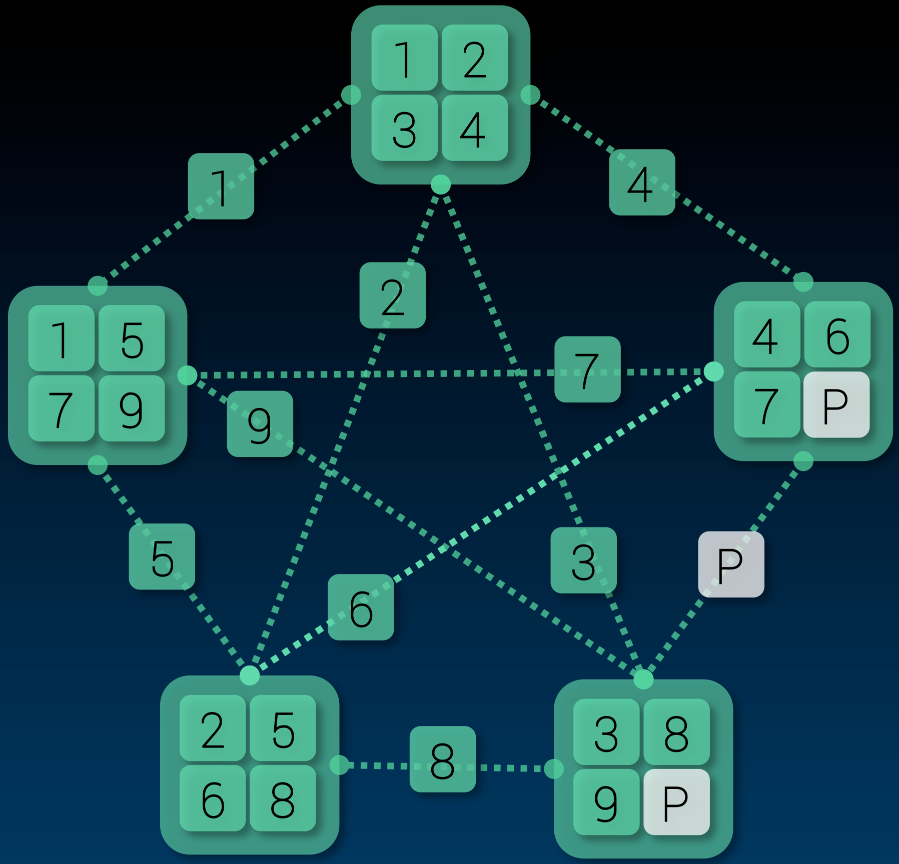
A + 2B + 3C + D
3A + 2B + 2C + 3D

5A + 7B + 8C + 7D
6A + 9B + 6C + 6D

Functional Repair

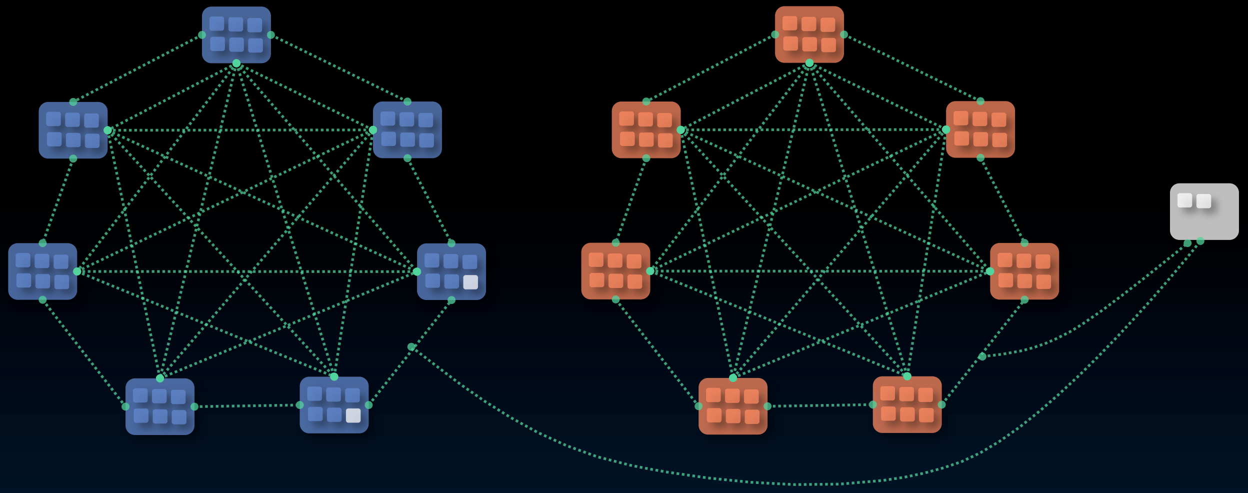
Regenerating Codes

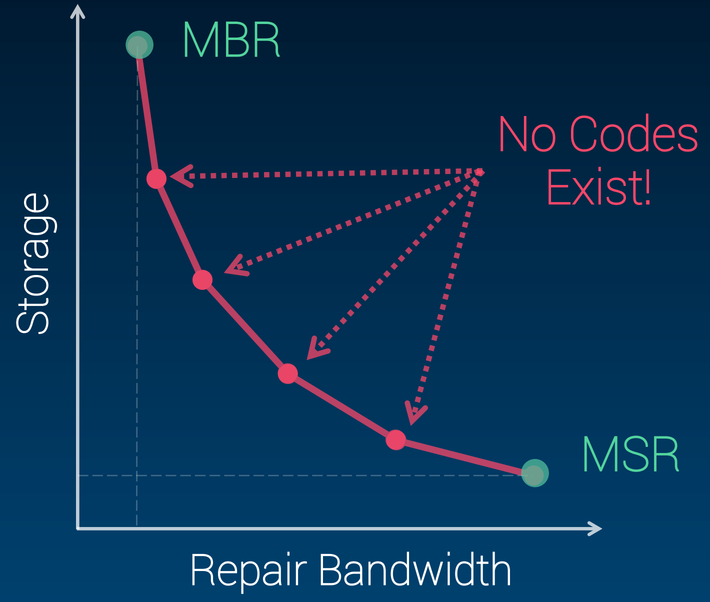
Repair By Transfer (RBT), MBR Code



Pentagon Code

Local Regenerating Code





Storage /
Repair BW

NetApp

Technical Analysis

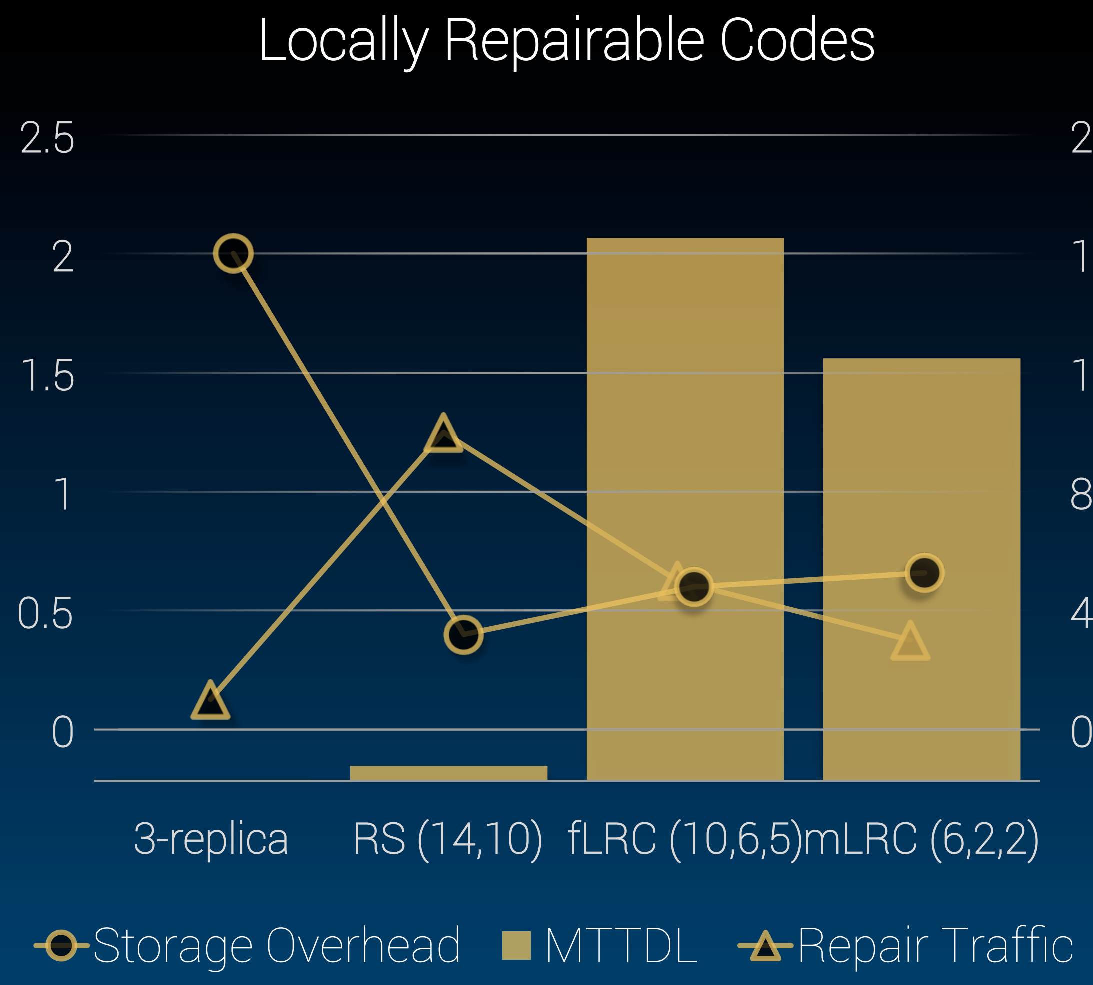
Optimality Tradeoffs – Reliability Analysis – System Requirements

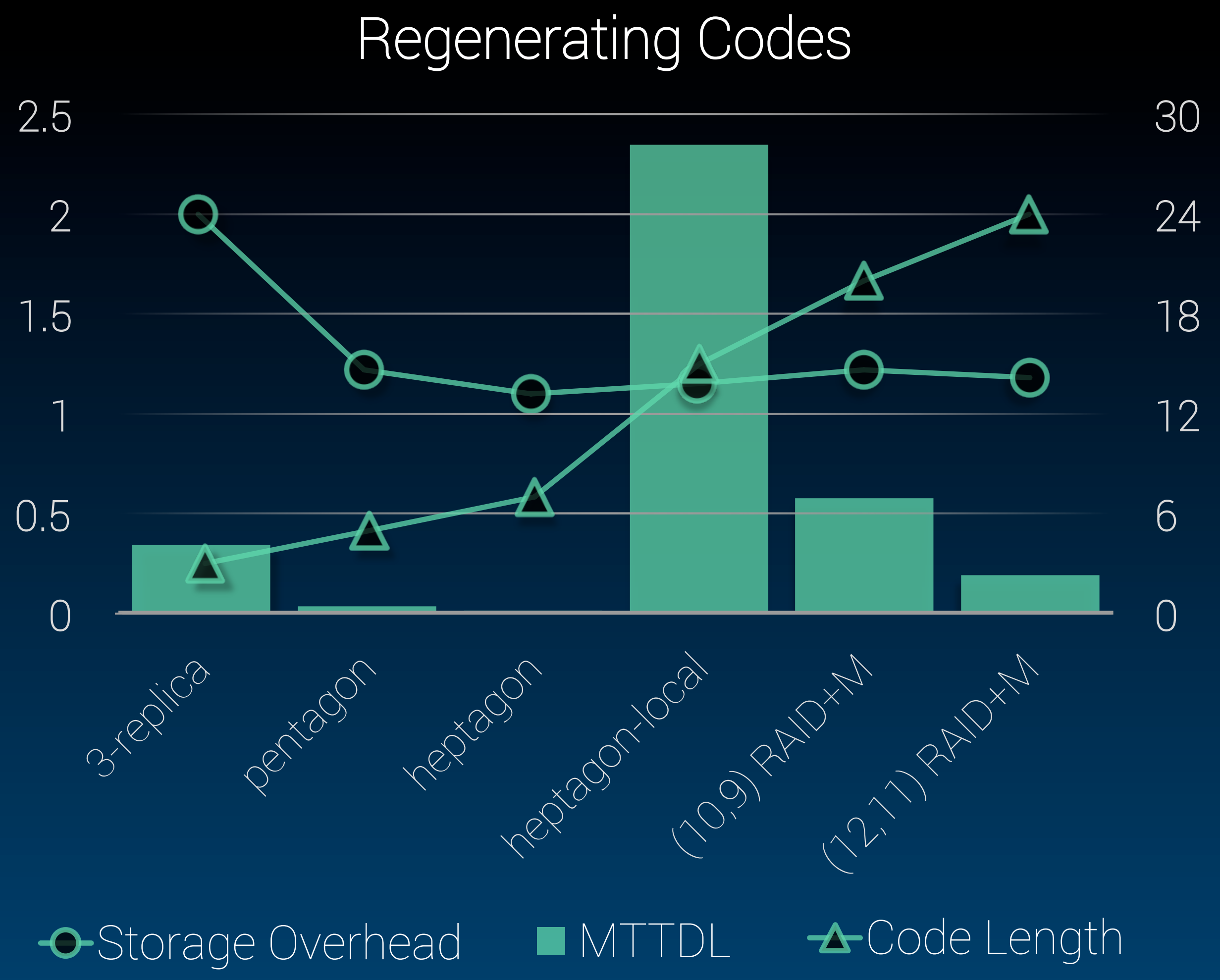
Summary of Codes and their Tradeoff

Code/Family	Tradeoff	
MDS	Storage overhead	Reliability
Replication & Parity (RAID)	Storage overhead	Reliability
Reed-Solomon	Storage overhead	Reliability
Near-Optimal	Correction capabilities	Computational Complexity
Fountain	Rate	Probability of Correction
Codes on Codes	Storage overhead	Repair Degree (Fan-in)
Azure (mLRC)	MDS	Maximum Recoverability
XORBAS (fLRC)	Locality	Minimum Distance
Regenerating	Storage overhead	Repair Bandwidth
Local Regenerating	Storage overhead	Reconstruction Cost

Reliability Analysis

MTTDL is in the order of E+12 (for LRC) and E+09 (for Regenerating)





NetApp

System Requirements & Example Codes

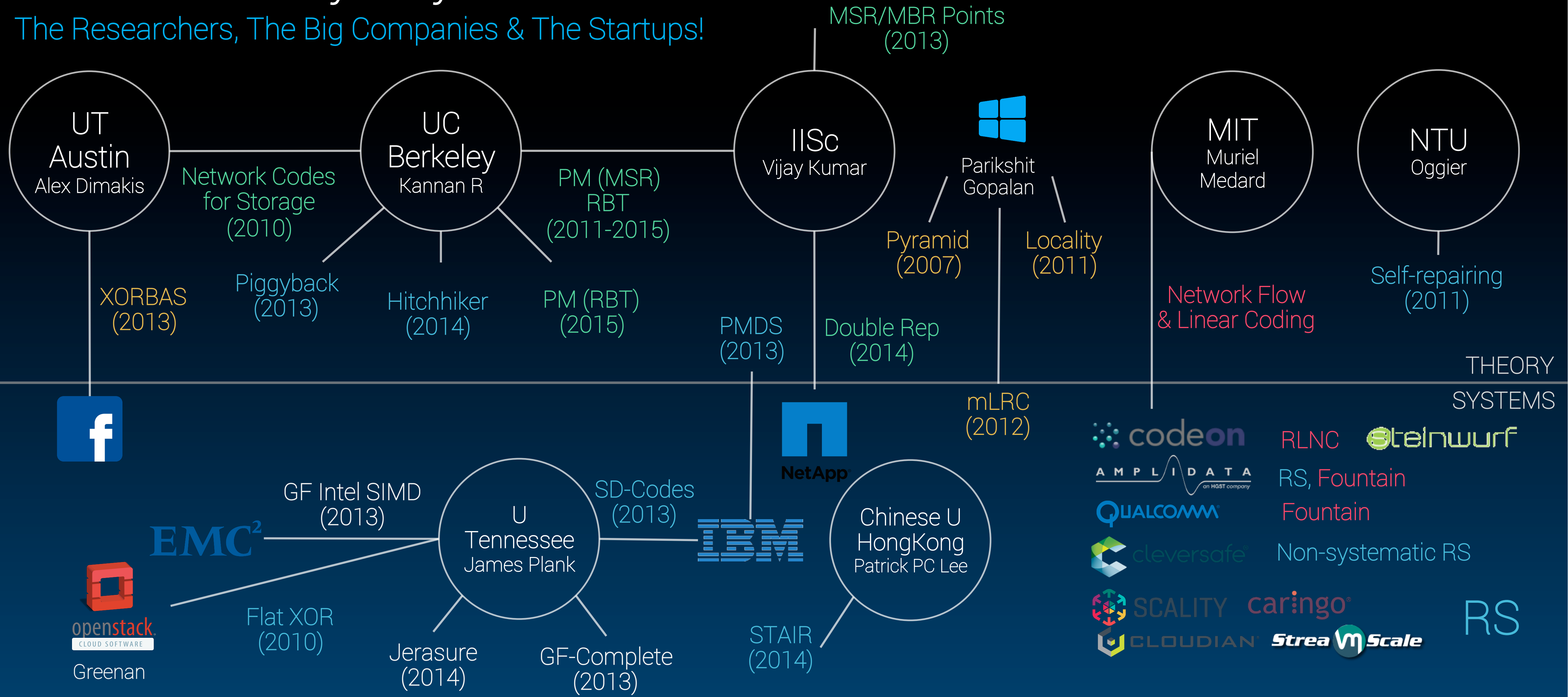
	System	Properties of the System		Requirements for a Code		Example family/ code
		Most Important	Least Important	Most Important	Least Important	
Architecture	General-purpose storage array	Reliability & Performance	Cost	Reliability	Complexity	MSR, SD/STAIR Codes
	Geo-distributed storage	Repair over WAN is expensive	Storage overhead across DR sites	Local repair	Storage overhead	LRC
	Secure Storage	Security	Storage overhead	Faster degraded reads	Repair time	Non-systematic codes; MBR
	Distributed Systems	Parallelism & Availability	Storage overhead	Systematic	Storage overhead	Replication
Workload	Big Data (say, Hadoop)	Large volumes of data	Write latency	Storage overhead	Repair bandwidth	Regenerating (MSR/MBR), systematic

Literature & Key Players

Theory & Systems

Literature & Key Players

The Researchers, The Big Companies & The Startups!



UT Austin
Alex Dimakis

Network Codes for Storage (2010)

XORBAS (2013)

UC Berkeley
Kannan R

Piggyback (2013)

Hitchhiker (2014)

PM (RBT) (2015)

PM (MSR) RBT (2011-2015)

IISc
Vijay Kumar

MSR/MBR Points (2013)

PMDS (2013)

Double Rep (2014)

Parikshit Gopalan

Pyramid (2007)

Locality (2011)

mLRC (2012)

MIT
Muriel Medard

Network Flow & Linear Coding

NTU
Oggier

Self-repairing (2011)

THEORY

SYSTEMS

NetApp

IBM

EMC²

openstack CLOUD SOFTWARE

Greenan

GF Intel SIMD (2013)

Flat XOR (2010)

U Tennessee
James Plank

SD-Codes (2013)

Jerasure (2014)

GF-Complete (2013)

Chinese U HongKong
Patrick PC Lee

STAIR (2014)

codeon

RLNC

steinwurf

AMPLIDATA an HGST company

RS, Fountain

QUALCOMM

Fountain

cleversafe

Non-systematic RS

SCALITY

caringo

CLOUDIAN

StreamScale

RS

Other Relevant Areas

- Cross-object Coding
  - Sector & Disk failures – PMDS, SD, STAIR Codes
- Other media:
  - Flash: LDPC, WOM, Multi-write codes; NVM
- Security
  - Dispersal, AONT-RS
- Cloud
  - NC-Cloud
- Transformational Codes: Transform encoded data to different parameters as they become hot/cold without decoding and re-encoding

Thank you.