On Coding for Distributed Networked Storage Systems

Frédérique Oggier,
Joint work with Anwitaman Datta

Nanyang Technological University, Singapore

IMS-NTU Workshop on Coding and Cryptography,
Singapore, May 2011

Outline

Distributed Networked Storage

- A data owner wants to *store* data over a network of nodes (e.g. data center, back-up or archival in peer-to-peer networks).

Distributed Networked Storage

- A data owner wants to *store* data over a network of nodes (e.g. data center, back-up or archival in peer-to-peer networks).
- Redundancy is essential for resilience

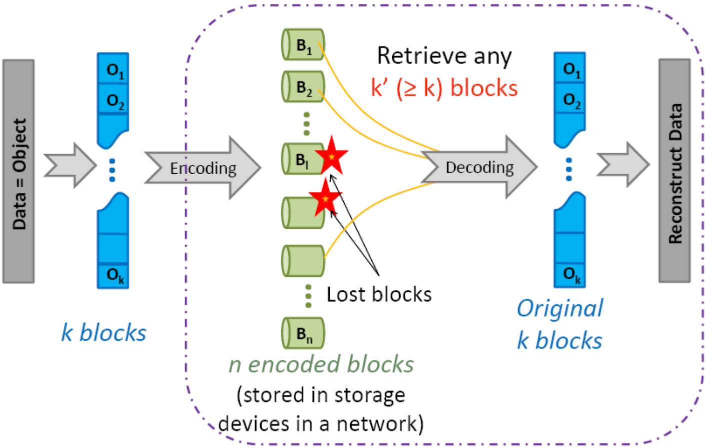
Distributed Networked Storage

- A data owner wants to *store* data over a network of nodes (e.g. data center, back-up or archival in peer-to-peer networks).
- Redundancy is essential for resilience
    - *Replication*: good availability and durability, but very costly.

Distributed Networked Storage

- A data owner wants to *store* data over a network of nodes (e.g. data center, back-up or archival in peer-to-peer networks).
- Redundancy is essential for resilience
  - *Replication*: good availability and durability, but very costly.
  - *Erasure codes*: good trade-off of availability, durability and storage cost.

Erasure codes for storage systems



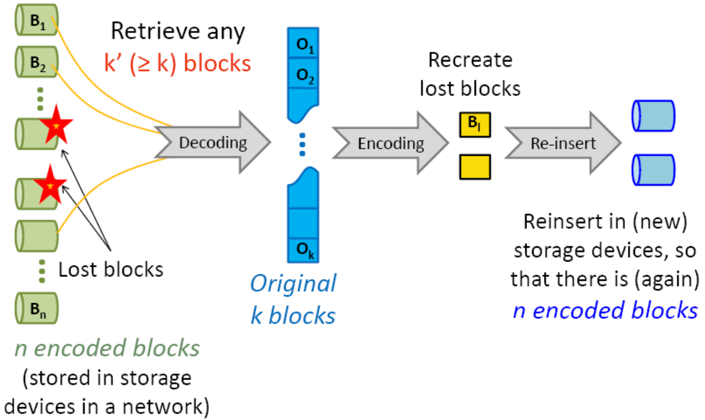
Repair

- Nodes may go offline, or may fail, so that the data they store becomes *unavailable*.

Repair

- Nodes may go offline, or may fail, so that the data they store becomes *unavailable*.
- Redundancy needs to be *replenished*, else data may be permanently lost over time (after multiple storage node failures)

Repair process using traditional Erasure Codes



Related work

1. J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao. OceanStore: An Architecture for Global-Scale Persistent Storage, ASPLOS 2000.
2. H. Weatherspoon, J. Kubiatowicz. Erasure Coding Vs. Replication: A Quantitative Comparison, Peer-to-Peer Systems, LNCS, 2002.

Related work

1. J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao. OceanStore: An Architecture for Global-Scale Persistent Storage, ASPLOS 2000.
2. H. Weatherspoon, J. Kubiatowicz. Erasure Coding Vs. Replication: A Quantitative Comparison, Peer-to-Peer Systems, LNCS, 2002.
3. A. G. Dimakis, P. Brighten Godfrey, M. J. Wainwright, K. Ramchandran, The Benefits of Network Coding for Peer-to-Peer Storage Systems, Netcod 2007.
4. A. Duminuco, E. Biersack, Hierarchical Codes: How to Make Erasure Codes Attractive for Peer-to-Peer Storage Systems, Peer-to-Peer Computing (P2P), 2008.
5. K. V. Rashmi, N. B. Shah, P. V. Kumar and K. Ramchandran, Explicit Construction of Optimal Exact Regenerating Codes for Distributed Storage, Allerton Conf. on Control, Computing and Comm., 2009.

Outline

Self-Repairing Codes (SRC)

- Motivation: *minimize* the number of nodes necessary to repair a missing block.

Self-Repairing Codes (SRC)

- Motivation: *minimize* the number of nodes necessary to repair a missing block.
- Gain: lower bandwidth consumption, lower computational complexity of repair, possibility for faster and parallel replenishment of lost redundancy.

Self-Repairing Codes (SRC)

- Motivation: *minimize* the number of nodes necessary to repair a missing block.
- Gain: lower bandwidth consumption, lower computational complexity of repair, possibility for faster and parallel replenishment of lost redundancy.
- Self-repairing codes are (n, k) codes such that

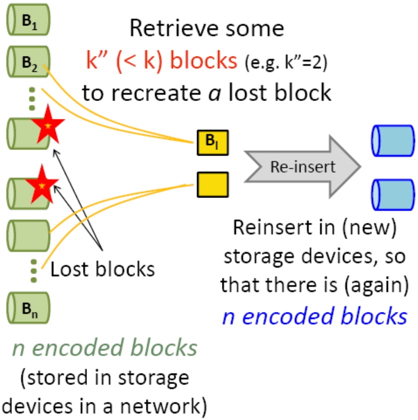
Self-Repairing Codes (SRC)

- Motivation: *minimize* the number of nodes necessary to repair a missing block.
- Gain: lower bandwidth consumption, lower computational complexity of repair, possibility for faster and parallel replenishment of lost redundancy.
- Self-repairing codes are (n, k) codes such that
    - encoded fragments can be repaired directly from other subsets of encoded fragments.

Self-Repairing Codes (SRC)

- Motivation: *minimize* the number of nodes necessary to repair a missing block.
- Gain: lower bandwidth consumption, lower computational complexity of repair, possibility for faster and parallel replenishment of lost redundancy.
- Self-repairing codes are (n, k) codes such that
  - encoded fragments can be repaired directly from other subsets of encoded fragments.
  - a fragment can be repaired from a fixed number of encoded fragments, independently of which specific blocks are missing (analogous to erasure codes supporting reconstruction using any $n - k$ losses, independently of which).

Self-Repairing Codes (a black-box view)



Homomorphic SRC (HSRC)

- A first instance of self-repairing code.

Self-repairing Homomorphic Codes for Distributed Storage Systems
F. Oggier, A. Datta, *INFOCOM 2011*

Preliminaries: Weakly linearized polynomials

- A *weakly linearized polynomial* $p(X)$ over $\mathbb{F}_q$, $q = 2^m$, has the form

$$p(X) = \sum_{i=0}^{k-1} p_i X^{2^i}, \; p_i \in \mathbb{F}_q.$$

- Let $a, b \in \mathbb{F}_{2^m}$ and let $p(X)$ be a weakly linearized polynomial given by $p(X) = \sum_{i=0}^{k-1} p_i X^{2^i}$. We have

$$p(a+b) = p(a) + p(b).$$

HSRC: Encoding

1. Take an object $\mathbf{o}$ of length M:

$$\mathbf{o} = (\mathbf{o}_1, \ldots, \mathbf{o}_k),\ \mathbf{o}_i \in \mathbb{F}_{2^{M/k}}.$$

2. Take a linearized polynomial with coefficients in $\mathbb{F}_{2^{M/k}}$

$$p(X) = \sum_{i=0}^{k-1} p_i X^{2^i},$$

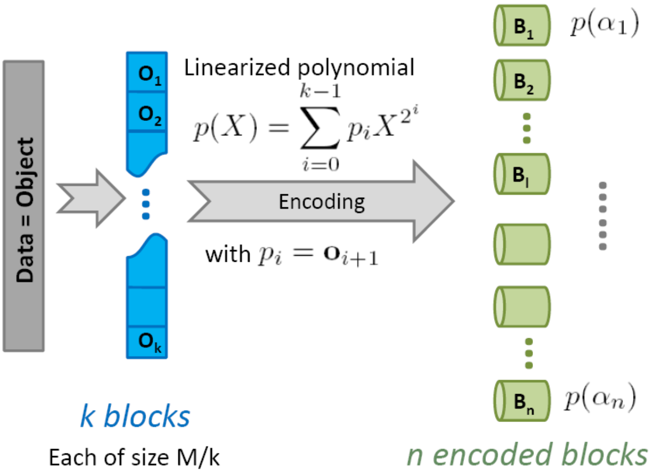
and encode the k fragments $p_i = \mathbf{o}_{i+1}$, $i = 0, \ldots, k-1$.

3. Evaluate $p(X)$ in n non-zero values $\alpha_1, \ldots, \alpha_n$ of $\mathbb{F}_{2^{M/k}}$ to get a n-dimensional codeword

$$(p(\alpha_1), \ldots, p(\alpha_n)),$$

and each $p(\alpha_i)$ is given to node i for storage.

HSRC: Encoding Illustration



HSRC: Decoding and Repair

1. *Decoding* is ensured by Lagrange interpolation.

HSRC: Decoding and Repair

1. *Decoding* is ensured by Lagrange interpolation.
2. *Repair*: Express α_i in a $\mathbb{F}_2$-basis $B = \{b_1, \ldots, b_{M/k}\}$ of $\mathbb{F}_{2^{M/k}}$, then

$$\alpha_i = \sum_{j=1}^{M/k} \alpha_{ij} b_j,\ \alpha_{ij} \in \mathbb{F}_2 \Rightarrow p(\alpha_i) = \sum_{j=1}^{M/k} \alpha_{ij} p(b_j).$$

HSRC: Decoding and Repair

1. *Decoding* is ensured by Lagrange interpolation.
2. *Repair*: Express α_i in a $\mathbb{F}_2$-basis $B = \{b_1, \ldots, b_{M/k}\}$ of $\mathbb{F}_{2^{M/k}}$, then

$$\alpha_i = \sum_{j=1}^{M/k} \alpha_{ij} b_j,\ \alpha_{ij} \in \mathbb{F}_2 \Rightarrow p(\alpha_i) = \sum_{j=1}^{M/k} \alpha_{ij} p(b_j).$$

3. Computational cost of a repair: XORs.

HSRC: A toy example (I)

- A data file $\mathbf{o} = (o_1, \ldots, o_{12})$ of $M = 12$ bits is cut into $k = 3$ fragments $(M/k = 4)$

$$\mathbf{o}_1 = (o_1, \ldots, o_4),\ \mathbf{o}_2 = (o_5, \ldots, o_8),\ \mathbf{o}_3 = (o_9, \ldots, o_{12}) \in \mathbb{F}_{2^4}.$$

HSRC: A toy example (I)

- A data file $\mathbf{o} = (o_1, \ldots, o_{12})$ of $M = 12$ bits is cut into $k = 3$ fragments ($M/k = 4$)

$$\mathbf{o}_1 = (o_1, \ldots, o_4),\ \mathbf{o}_2 = (o_5, \ldots, o_8),\ \mathbf{o}_3 = (o_9, \ldots, o_{12}) \in \mathbb{F}_{2^4}.$$

- $\mathbb{F}_{2^4}^* = \langle w \rangle$, with $w^4 = w + 1$. Encode with the polynomial

$$p(X) = \sum_{i=1}^{4} o_i w^i X + \sum_{i=1}^{4} o_{i+4} w^i X^2 + \sum_{i=1}^{4} o_{i+8} w^i X^4.$$

HSRC: A toy example (I)

- A data file $\mathbf{o} = (o_1, \ldots, o_{12})$ of $M = 12$ bits is cut into $k = 3$ fragments ($M/k = 4$)

$$\mathbf{o}_1 = (o_1, \ldots, o_4),\ \mathbf{o}_2 = (o_5, \ldots, o_8),\ \mathbf{o}_3 = (o_9, \ldots, o_{12}) \in \mathbb{F}_{2^4}.$$

- $\mathbb{F}_{2^4}^* = \langle w \rangle$, with $w^4 = w + 1$. Encode with the polynomial

$$p(X) = \sum_{i=1}^{4} o_i w^i X + \sum_{i=1}^{4} o_{i+4} w^i X^2 + \sum_{i=1}^{4} o_{i+8} w^i X^4.$$

- For $n = 7$, evaluate $p(X)$ at say $1, w, w^2, w^4, w^5, w^8, w^{10}$. We get:

$$(p(1), p(w), p(w^2), p(w^4), p(w^5), p(w^8), p(w^{10}))$$

HSRC: A toy example (II)

missing fragment(s)	pairs to reconstruct missing fragment(s)
$p(1)$ $p(w)$ $p(w^2)$	$(p(w), p(w^4))$;$(p(w^2), p(w^8))$;$(p(w^5), p(w^{10}))$ $(p(1), p(w^4))$;$(p(w^2), p(w^5))$;$(p(w^8), p(w^{10}))$ $(p(1), p(w^8))$;$(p(w), p(w^5))$;$(p(w^4), p(w^{10}))$
$p(1)$ and $p(w)$	$(p(w^2), p(w^8))$ or $(p(w^5), p(w^{10}))$ for $p(1)$ $(p(w^8), p(w^{10}))$ or $(p(w^2), p(w^5))$ for $p(w)$
$p(1)$ and $p(w)$ and $p(w^2)$	$(p(w^5), p(w^{10}))$ for $p(1)$ $(p(w^8), p(w^{10}))$ for $p(w)$ $(p(w^4), p(w^{10}))$ for $p(w^2)$

Self-Repairing Codes from Projective Geometry (PSRC)

- A second instance of self-repairing code.

Self-Repairing Codes for Distributed Storage - A Projective Geometric Construction, F. Oggier, A. Datta, *preprint 2011*

Preliminaries: Spreads

- Consider a vector space of dimension m over $\mathbb{F}_q$, namely, a projective space $PG(m-1,q)$.
- Let $\mathcal{P}$ be a projective space. A t-*spread* of $\mathcal{P}$ is a set $\mathcal{S}$ of t-dimensional subspaces of $\mathcal{P}$ which partitions $\mathcal{P}$.

Preliminaries: Spreads

- Consider a vector space of dimension m over $\mathbb{F}_q$, namely, a projective space $PG(m-1,q)$.
- Let $\mathcal{P}$ be a projective space. A t-*spread* of $\mathcal{P}$ is a set $\mathcal{S}$ of t-dimensional subspaces of $\mathcal{P}$ which partitions $\mathcal{P}$.

Theorem (André)

In $PG(m-1,q)$, a t-spread exists if and only if $t+1|\ m$.

Spreads from Field Extensions

- Suppose that $t+1 \mid m$. Consider the finite fields $F_0 = \mathbb{F}_q$, $F_1 = \mathbb{F}_{q^{t+1}}$ and $F_2 = \mathbb{F}_{q^m}$.

Spreads from Field Extensions

- Suppose that $t+1|\ m$. Consider the finite fields $F_0 = \mathbb{F}_q$, $F_1 = \mathbb{F}_{q^{t+1}}$ and $F_2 = \mathbb{F}_{q^m}$.
- Then $F_0 \subseteq F_1 \subseteq F_2$. The field F_2 is an m-dimensional vector space V over F_0.

Spreads from Field Extensions

- Suppose that $t+1 \mid m$. Consider the finite fields $F_0 = \mathbb{F}_q$, $F_1 = \mathbb{F}_{q^{t+1}}$ and $F_2 = \mathbb{F}_{q^m}$.
- Then $F_0 \subseteq F_1 \subseteq F_2$. The field F_2 is an m-dimensional vector space V over F_0.
- The subspaces of V form the projective space $\mathcal{P}$=PG(m, q). The field F_1 is a $(t+1)$-dimensional subspace of V and hence a t-dimensional (projective) subspace of $\mathcal{P}$.

Spreads from Field Extensions

- Suppose that $t+1|\ m$. Consider the finite fields $F_0 = \mathbb{F}_q$, $F_1 = \mathbb{F}_{q^{t+1}}$ and $F_2 = \mathbb{F}_{q^m}$.
- Then $F_0 \subseteq F_1 \subseteq F_2$. The field F_2 is an m-dimensional vector space V over F_0.
- The subspaces of V form the projective space $\mathcal{P}$=PG(m, q). The field F_1 is a $(t+1)$-dimensional subspace of V and hence a t-dimensional (projective) subspace of $\mathcal{P}$.
- The same holds for all cosets aF_1, $(a \in F_2)$. These cosets partition the multiplicative group of F_2. Hence they form a t-spread of $\mathcal{P}$.

PSRC: Encoding

1. For an object **o** of size M, consider the finite field $\mathbb{F}_{2^M}$.
2. Consider a t-spread $\mathcal{S}$ formed of t-dimensional subspaces of $\mathcal{P}$ such that $t+1|M$. Set $\alpha = t+1$. Assign to each node an $\mathbb{F}_2$-basis containing α vectors. The number of storage nodes is (at most)

$$n = \frac{2^M - 1}{2^\alpha - 1}.$$

3. The ith node will actually store

$$\{\mathbf{o}v_{i\alpha+1}^T, \ldots, \mathbf{o}v_{(i+1)\alpha}\}$$

for a total storage of α.

PSRC: Decoding and Repair

1. *Decoding* is solving a system of linear equations.

PSRC: Decoding and Repair

1. *Decoding* is solving a system of linear equations.
2. *Repair* The lth node N_l stores $\nu^l\mathbb{F}^*_{2^\alpha}$, $l = 1, \dots, n$. Let us assume this lth node fails, and a new comer N_i joins. Contact the jth node N_j such that $\nu^j = \nu^i + \nu^l$. By combining the data stored at node N_i and N_j, we get

$$\nu^i\mathbb{F}^*_{2^\alpha} \coprod (\nu^i + \nu^l)\mathbb{F}^*_{2^\alpha}$$

which contains $\nu^l\mathbb{F}^*_{2^\alpha}$.

Lemma

For any choice of node N_i among the remaining $n-1$ live nodes, there exists at least one node N_j such that N_l can be repaired by downloading the data stored at nodes N_i and N_j.

PSRC: A toy example

node	basis vectors	data stored
N_1	$v_1 = (1000)$, $v_2 = (0110)$	$\{o_1, o_2 + o_3\}$
N_2	$v_3 = (0100)$, $v_4 = (0011)$	$\{o_2, o_3 + o_4\}$
N_3	$v_5 = (0010)$, $v_6 = (1101)$	$\{o_3, o_1 + o_2 + o_4\}$
N_4	$v_7 = (0001)$, $v_8 = (1010)$	$\{o_4, o_1 + o_3\}$
N_5	$v_9 = (1100)$, $v_{10} = (0101)$	$\{o_1 + o_2, o_2 + o_4\}$

Outline

Static resilience

- There is at least one pair to repair a node, for up to $(n-1)/2$ simultaneous failures
- Static resilience of a distributed storage system is the probability that an object stored in the system stays available without any further maintenance, even when a fraction of nodes become unavailable.

Static resilience: HSRC versus EC

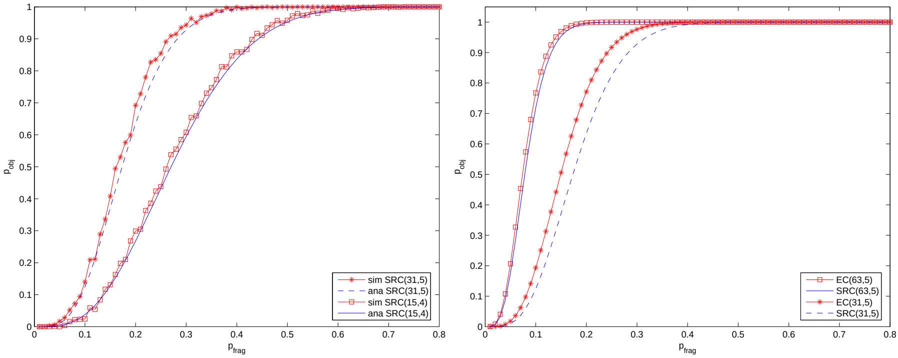
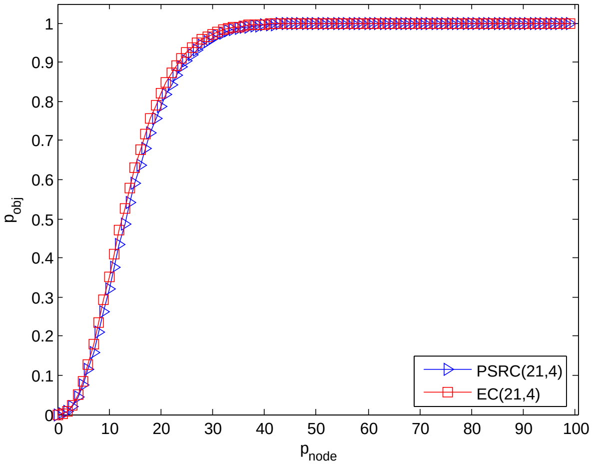
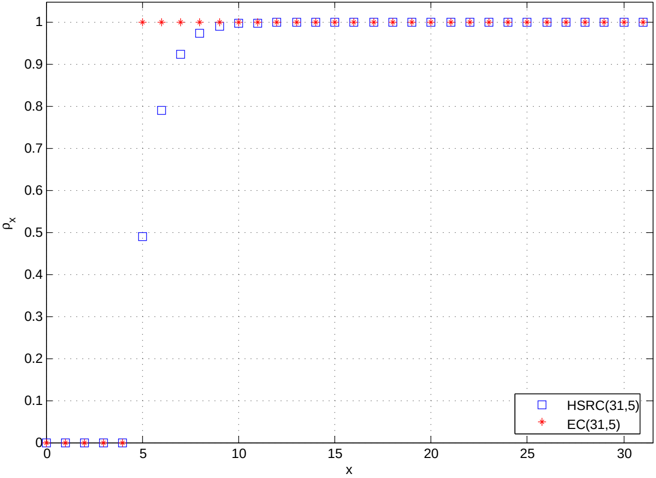


Figure: Static resilience of self-repairing codes (SRC): Validation of analysis, and comparison with erasure codes (EC)

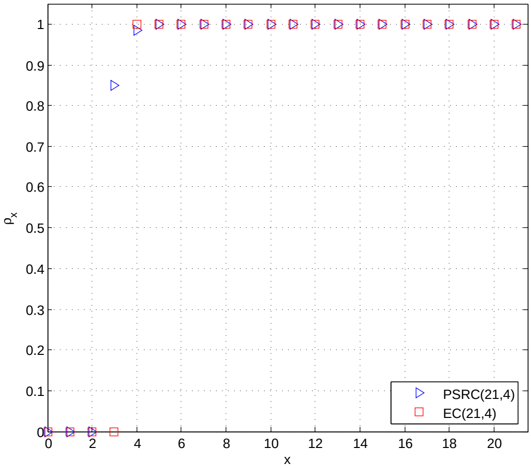
Static resilience: PSRC versus EC



More on Resilience: HSRC versus EC



More on Resilience: PSRC versus EC



Fast & parallel repairs using HSRC: A toy example

- Consider:
  - (15,4) code, nodes storing $p(w^i)$ for $i = 0, 1, 2, 3, 4, 5, 6$ are missing
  - Nodes have upload/download bandwidth limit: one block per time unit

Fast & parallel repairs using HSRC: A toy example

- Consider:
  - (15,4) code, nodes storing $p(w^i)$ for $i = 0, 1, 2, 3, 4, 5, 6$ are missing
  - Nodes have upload/download bandwidth limit: one block per time unit
- Possible pairs to repair each missing block:

fragment	suitable pairs to reconstruct
$p(1)$	$(p(w^7), p(w^9))$;$(p(w^{11}), p(w^{12}))$
$p(w)$	$(p(w^7), p(w^{14}))$;$(p(w^8), p(w^{10}))$
$p(w^2)$	$(p(w^7), p(w^{12}))$;$(p(w^9), p(w^{11}))$;$(p(w^{12}), p(w^{10}))$
$p(w^3)$	$(p(w^8), p(w^{13}))$;$(p(w^{10}), p(w^{12}))$
$p(w^4)$	$(p(w^9), p(w^{14}))$;$(p(w^{11}), p(w^{13}))$
$p(w^5)$	$(p(w^7), p(w^{13}))$;$(p(w^{12}), p(w^{14}))$
$p(w^6)$	$(p(w^7), p(w^{10}))$;$(p(w^8), p(w^{14}))$

Fast & parallel repairs using HSRC: A toy example

- Consider:
  - (15,4) code, nodes storing $p(w^i)$ for $i = 0, 1, 2, 3, 4, 5, 6$ are missing
  - Nodes have upload/download bandwidth limit: one block per time unit
- Possible pairs to repair each missing block:

fragment	suitable pairs to reconstruct
$p(1)$	$(p(w^7), p(w^9));(p(w^{11}), p(w^{12}))$
$p(w)$	$(p(w^7), p(w^{14}));(p(w^8), p(w^{10}))$
$p(w^2)$	$(p(w^7), p(w^{12}));(p(w^9), p(w^{11}));(p(w^{12}), p(w^{10}))$
$p(w^3)$	$(p(w^8), p(w^{13}));(p(w^{10}), p(w^{12}))$
$p(w^4)$	$(p(w^9), p(w^{14}));(p(w^{11}), p(w^{13}))$
$p(w^5)$	$(p(w^7), p(w^{13}));(p(w^{12}), p(w^{14}))$
$p(w^6)$	$(p(w^7), p(w^{10}));(p(w^8), p(w^{14}))$

- A parallelized schedule:

node	$p(w^0)$	$p(w^1)$	$p(w^2)$	$p(w^3)$	$p(w^4)$	$p(w^5)$	$p(w^6)$
Time 1	$p(w^7)$	$p(w^8)$	$p(w^9)$	$p(w^{13})$	$p(w^{11})$	$p(w^{12})$	$p(w^{10})$
Time 2	$p(w^9)$	$p(w^{10})$	$p(w^{11})$	$p(w^8)$	$p(w^{13})$	$p(w^{14})$	$p(w^7)$

Systematic Object Retrieval using PSRC: A toy example

node	basis vectors	data stored
N_1	$v_1 = (1000)$, $v_2 = (0110)$	$\{o_1, o_2 + o_3\}$
N_2	$v_3 = (0100)$, $v_4 = (0011)$	$\{o_2, o_3 + o_4\}$
N_3	$v_5 = (0010)$, $v_6 = (1101)$	$\{o_3, o_1 + o_2 + o_4\}$
N_4	$v_7 = (0001)$, $v_8 = (1010)$	$\{o_4, o_1 + o_3\}$
N_5	$v_9 = (1100)$, $v_{10} = (0101)$	$\{o_1 + o_2, o_2 + o_4\}$

Future/ongoing work

- Efficient decoding, other instances of SRC
- Implementation & integration in a distributed storage system
- Various systems/algorithmic issues: Topology optimized placement, repair scheduling

Wrap Up

- Design of codes for distributed networked storage

Wrap Up

- Design of codes for distributed networked storage
- Self-Repairing Codes

Wrap Up

- Design of codes for distributed networked storage
- Self-Repairing Codes
- New research topic in coding theory!

Q&A

- More information: http://sands.sce.ntu.edu.sg/CodingForNetworkedStorage/

Q&A

- More information: http://sands.sce.ntu.edu.sg/CodingForNetworkedStorage/
- Contact: {frederique,anwitaman}@ntu.edu.sg