Reliability Comparison of Various Regenerating Codes for Cloud Services

Yonsei Univ. Seoul, KOREA
Jung-Hyun Kim, Jin Soo Park, Ki-Hyeon Park,
Inseon Kim, Mi-Young Nam, and Hong-Yeop Song
Contents

Introduction
  - Why we need regenerating codes for clouds?
  - How to regenerating failed nodes?

Background of Regenerating Codes
  - Regenerating code framework
  - Tradeoff between storage size and repair bandwidth

Various Regenerating Codes
  - Minimum Storage Regenerating (MSR) codes
  - Minimum Bandwidth Regenerating (MBR) codes
  - Local Reconstruction Codes (LRC)
  - LT Regenerating codes

Simulation Results

Conclusion
Introduction

- Why we need regenerating codes for clouds?
  - To repair *node failure*
    - At *Facebook*, it is quite typical to have 20 or more node failures per day.

Number of failed nodes over a single month period in 3000 nodes

Introduction

- How to regenerate failed nodes?
  - Node repair using codes for erasure channel

![Diagram showing node repair using codes]

- Repetition code
- MDS code
Introduction

- How to regenerate failed nodes?
  - MDS codes have higher reliability than repetition codes
Background of Regenerating codes

- Regenerating Codes Framework

\[
\begin{align*}
  n & : \text{# of storage nodes} \\
  k & : \text{# of storage nodes for data collection} \\
  \alpha & : \text{storage size} \\
  \beta & : \text{download size} \\
  d & : \text{# of storage nodes for node repair (read cost)} \\
  d\beta & : \text{repair bandwidth} \\
  M & : \text{data size}
\end{align*}
\]
Background of Regenerating codes

- Reducing *storage size and repair bandwidth*
  - Based on the min-cut bound:
    \[ \sum_{i=0}^{k-1} \min\{(d-i)\beta, \alpha\} \geq M \]

< Tradeoff between storage size and repair bandwidth \((M=7000, n=15, k=7, d=7)\) >

Background of Regenerating codes

- Reducing *repair read cost*
  - Repair read cost: the minimum number of nodes for repair

Efficient Repair
Low Reliability

High Reliability
Inefficient Repair

Q. Tradeoff between “repair read cost” and “reliability”?
Various Regenerating codes

Regenerating codes

- **Tradeoff storage size and repair bandwidth**
  - Minimum Storage Regenerating (MSR) Codes
  - Minimum Bandwidth Regenerating (MBR) Codes

- **Repair read cost**
  - Local Reconstruction Codes (LRC)

- **Rateless codes**
  - (Good performance for erasure channel)
  - LT Reconstruction Codes
Various Regenerating codes

- Minimum Storage Regenerating (MSR) codes
  - Using a Maximum Distance Separable (MDS) code

- Code construction methods
  - Interference Alignment method, Product-Matrix method, etc.


Various Regenerating codes

- Minimum Bandwidth Regenerating (MBR) codes
  - Using a Fractional Repetition (FR) code

![Diagram of FR code](image)

- Code construction methods
  - Repair-by-product method, Product-Matrix method, etc.


Various Regenerating codes

- Local Reconstruction Codes (LRC)
  - Extending an MDS code

Various Regenerating codes

- Local Reconstruction Codes (LRC)
  - Repair read cost comparison between MSR code and LRC

**MSR code**
- Repair read cost = 8

**LRC**
- Repair read cost = 4
Various Regenerating codes

- LT Regenerating Codes
  - Using the ideal/robust soliton distribution

< (8, 4) LT regenerating code >

Simulation Results

- Better cost and overhead trade-off

![Graph showing the trade-off between storage overhead and repair read cost for different schemes: LT, MSR_{RS}, LRC, and MBR_{REP}.

- **Storage overhead**: the ratio of all storage nodes, n, to storage nodes for data collection, k.
- **Repair read cost**: the number of storage nodes for node repair.

Better cost and overhead trade-off is achieved with lower repair read cost and comparable storage overhead in the graph shown.
Simulation Results

- Repair failure probability for different node failure probability

Node failure prob. : the probability that a node is unavailable

Repair failure prob. : the probability that any newcomer nodes cannot repair the original data symbol from coded data symbols of surviving storage nodes
Conclusion

Through the trade-off between repair read cost and storage overhead, we can expect that the optimal coding scheme might be different according to system requirements.

Although LRC is not an MDS code, it achieves both low repair read cost and low storage overhead by relaxing MDS property. Hence LRC can be a good candidate for practical systems and it should be studied more as a future coding scheme for cloud services.