

Have a Seat on the ErasureBench: Easy Evaluation of Erasure Coding Libraries for Distributed Storage Systems

2016 Workshop on Planetary-Scale Distributed Systems

Sébastien Vaucher, Hugues Mercier, Valerio Schiavoni

Institute of Computer Science
Université de Neuchâtel, Switzerland

sebastien.vaucher@unine.ch

Budapest, Hungary, 26 September 2016



Motivation

More and more data needs to be stored reliably on online servers.
Reliability can be provided through:

- Replication
- Erasure coding

Motivation

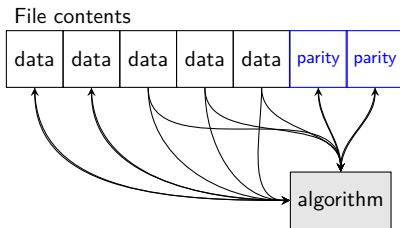
The characteristics of an erasure coding algorithm are difficult to evaluate.

Evaluation is often done theoretically or by simulation.

Erasure coding

Goal: add redundancy to cope with data loss/corruption

Example using a $(5, 2)$ Reed-Solomon code:



Key features

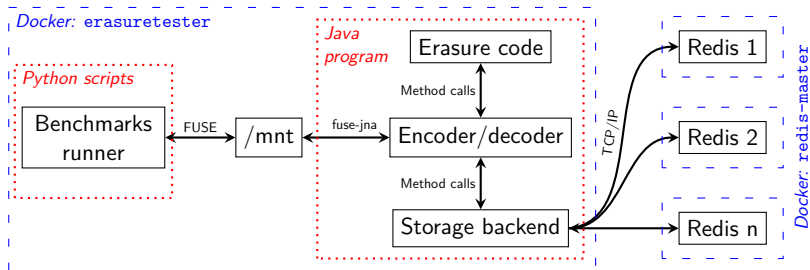
- Compatible with existing benchmark programs
- Automated benchmarks execution
- Containerized storage nodes (> 1 per physical node)
- Replay fault traces

Evaluation example

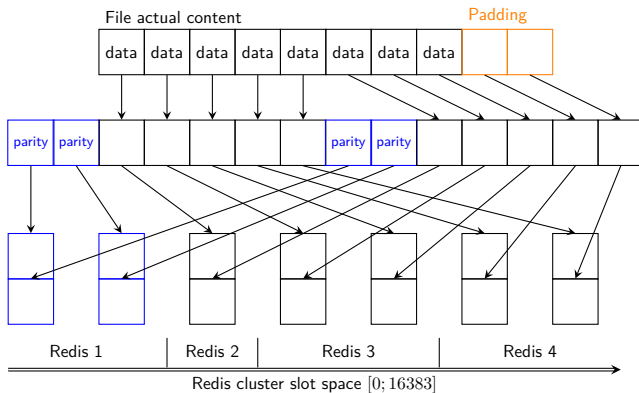
How to evaluate a new algorithm.

1. Program the algorithm as a Java class
2. Write benchmarks as Python functions
 - Debian-compatible programs can be launched as sub-processes
3. Configure the evaluation
 - e.g. algorithm parameters, fault trace, ...
4. Easily deploy the solution to a Docker cluster
5. Collect results

Technical components



Blocks distribution



Metadata management

Each block is identified by a 32-bit key. Using it, we derive:

1. Key of the blocks aggregation stored in Redis
2. Offset within that aggregation

The list of all block keys is kept in memory.

Automated deployment and scaling

As part of ERASUREBENCH, we provide scripts that completely automate

Evaluation

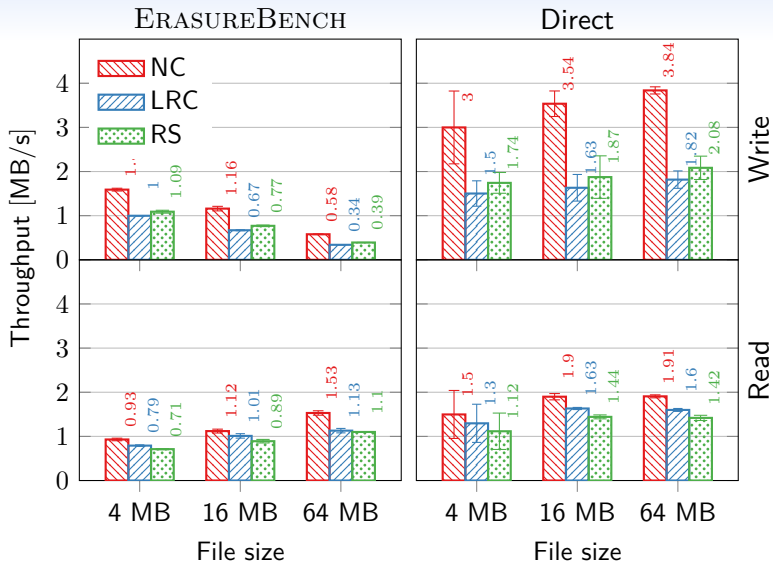
We evaluated algorithms from "XORing Elephants: Novel Erasure Codes for Big Data".

NC No erasure coding

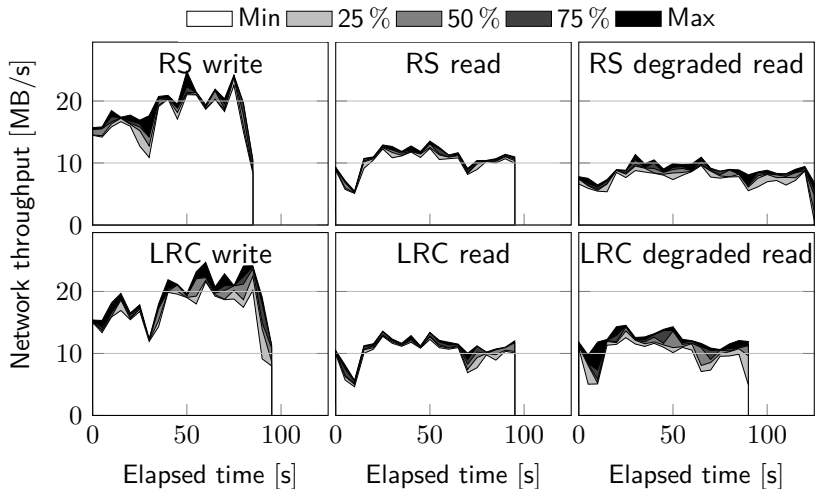
RS Reed-Solomon (10, 4)

LRC Locally Repairable Code (10, 6, 5)

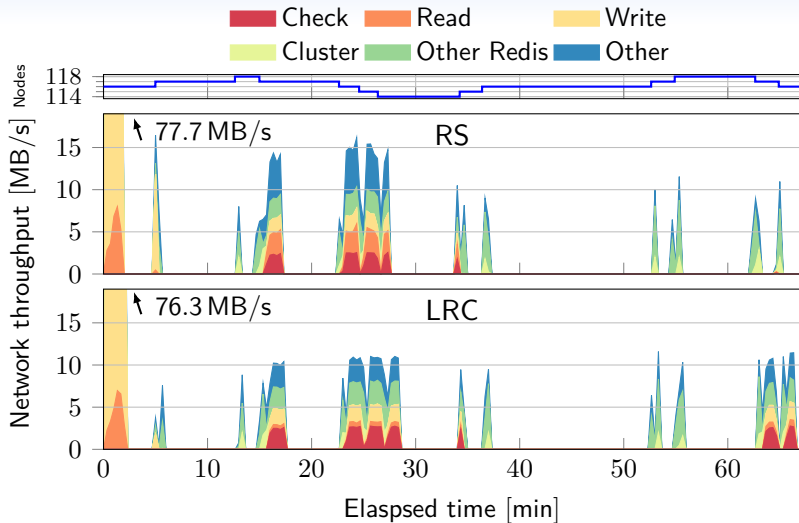
Throughput



Traffic



Trace



Conclusion

Using ERASUREBENCH, evaluating an erasure coding algorithm under real conditions is easier and cheaper.

Available open-source at
<https://github.com/safecloud-project/erasurebench>