Background

Erasure Coding

Defined: a technique used in computer systems to handle data loss and corruption
Applications:
- network data transmission
- file archiving
- bar code reliability

Start with \( K \) blocks of data.
Encode to get \( M \) additional blocks of coding data.
Decoding recovers the data when up to \( M \) blocks are lost.

Jerasure - an open-source erasure coding library

Reed-Solomon coding:
- Data is organized in blocks.
- PacketSize (PS) \( \times \) PacketSlices (PPS) = BlockSize
- Performs matrix multiplication over Galois field \( GF(2^W) \)

\[ K = 3 \quad PPS = 2 \quad M = 2 \]

Performance tests and results

Testing Methods

- Encoded 100 MB with Reed-Solomon and Cauchy Reed-Solomon
- -C3 compiler optimizations
- Averaged 5 runs where each run averaged 10 encodes

Coding Parameters:

- Reed-Solomon
  - \( W = 8, 16, 32 \)
  - \( M \) varied from 1 to 8
  - NumberOfCores increased from 1 to 4

- Cauchy Reed-Solomon
  - \( WPD = 5 \), the smallest valid value for all tested values of \( K \) and \( M \)

Parallel programming

In recent years, CPU speeds have leveled off due to device limitations such as power consumption.

- Modern processor advancements focus heavily on multi-core CPUs.
- Software must be specially programmed to take advantage of the multiple cores.

Multi-threaded erasure coding in Jerasure

Michael Jugan
mjugan1@utk.edu

Goals

Update Jerasure to utilize multiple processor cores while coding
- Increase performance
- Provide an intuitive user interface

Implementation

Four methods for splitting the work among threads:

- disks (M)
- packet_rows (WP M)
- packet_cols (PPS / PPS)
- packets (PPS x M)

Each color represents the data coded by a single thread.

Two new public variables were added to class JER_Slices.
- string MultiThreadMethod

Users add two additional lines of code before calling Encode().

- slices->MultiThreadMethod = "disks";
- slices->MultiThreadMethod = "packets";

Performance tests and results

Testing Methods

- Encoded 100 MB with Reed-Solomon and Cauchy Reed-Solomon
- -C3 compiler optimizations
- Averaged 5 runs where each run averaged 10 encodes

Coding Parameters:

- Reed-Solomon
  - \( W = 8, 16, 32 \)
  - \( M \) varied from 1 to 8
  - NumberOfCores increased from 1 to 4

- Cauchy Reed-Solomon
  - \( WPD = 5 \), the smallest valid value for all tested values of \( K \) and \( M \)

Conclusion

- Increased performance
  - 2 cores \( \rightarrow \) almost 2x speedup
  - depends heavily on the chosen coding parameters

- Easy to use
  - requires two additional lines of code.

- Future work
  - automate the process of finding optimal settings

References