The lzip format

Antonio Díaz Díaz

antonio@gnu.org

http://www.nongnu.org/lzip/lzip_talk_ghm_2019.html
http://www.nongnu.org/lzip/lzip_talk_ghm_2019_es.html

GNU Hackers Meeting
Madrid, September 4th 2019

Introduction

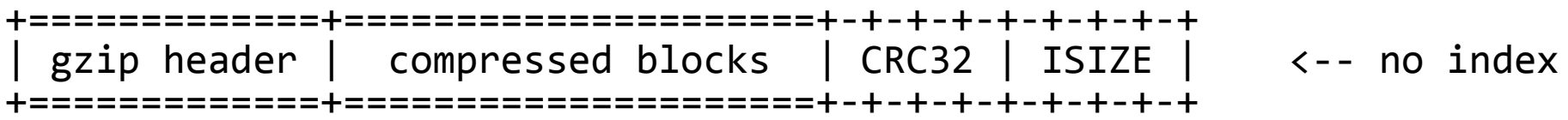
- There are a lot of compression algorithms
- Most are just variations of a few basic algorithms
- The basic ideas of compression algorithms are well known
- Algorithms much better than those existing are not probable to appear in the foreseeable future
- Formats existing when lzip was designed in 2008 (gzip and bzip2) have limitations that aren't easily fixable

Therefore...

- It seemed adequate to pack a good algorithm like LZMA into a well designed format
- Lzip is an attempt at developing such a format

Why a new format and tool?

- Adding LZMA compression to gzip doesn't work
  - The gzip format was designed long ago
  - It has limitations
    - → 32-bit uncompressed size
    - → No index
  - If extended it would impose those limitations to the new algorithm



- A new format with support for 64-bit file sizes is needed

LZMA algorithm

Features (thanks to Igor Pavlov)

- Wide range of compression ratios and speeds
- Higher compression ratio than gzip and bzip2
- Faster decompression speed than bzip2

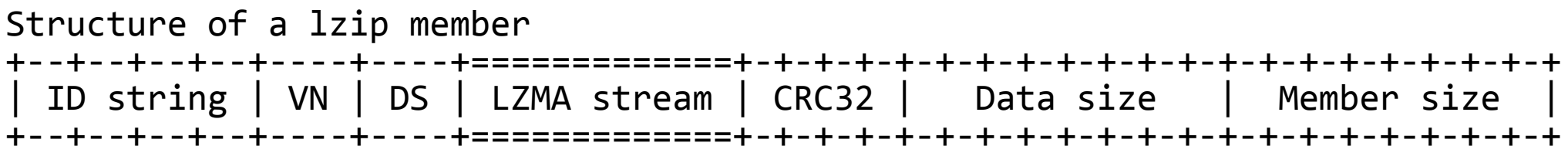
LZMA variants used by lzip

- Fast (used by option '-0')
- Normal (used by all other compression levels)

Lzip format

Features

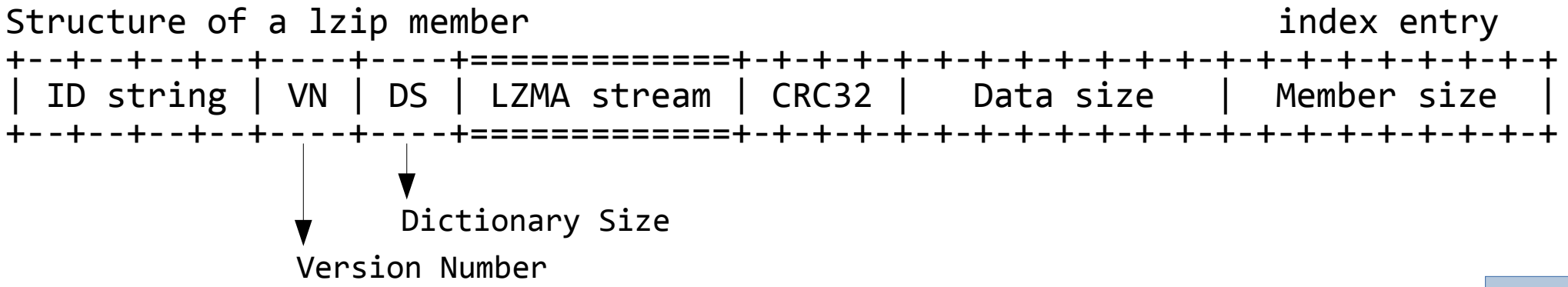
- Fast multi-threaded decompression (index)
- Friendly to 32-bit architectures
- Data recovery friendly
- Very safe 3 factor integrity checking
- Simple but complete
- Reproducible
- Prepared for the future
  - → 64-bit uncompressed and compressed sizes
  - → Unlimited file size of multimember files



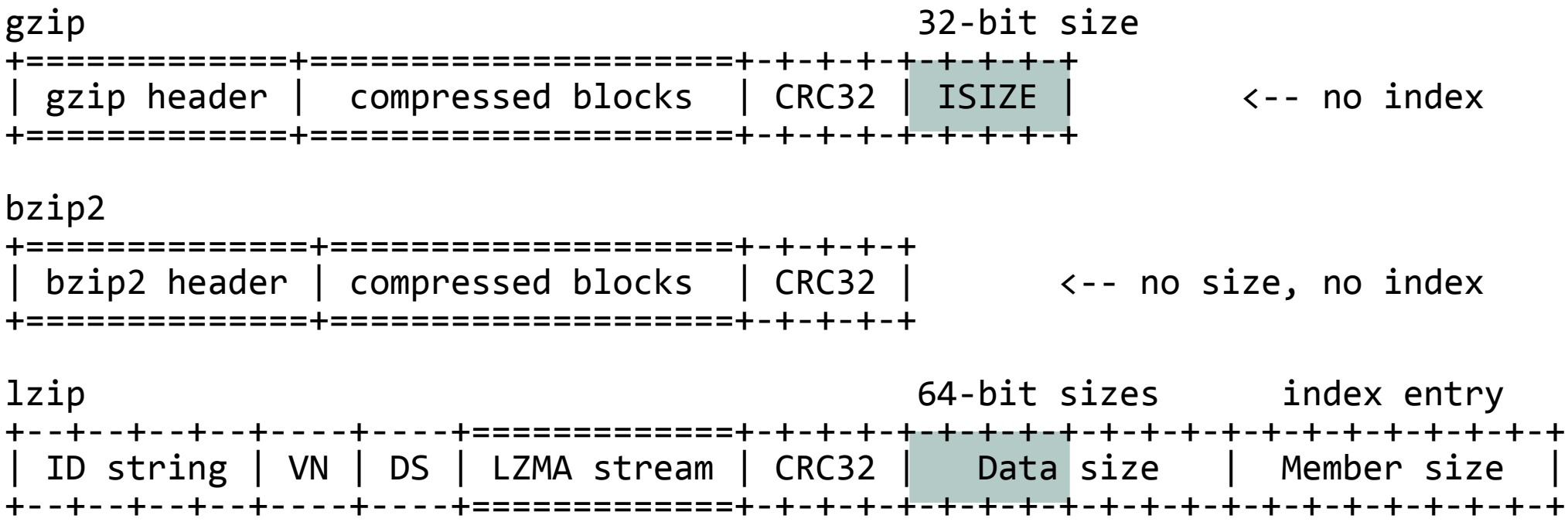
Distributed index

What is the index used for?

- Essential for efficient multi-threaded decompression
- Improves verification of stream integrity
- Helps lziprecover in the recovery of damaged files
- Allows the listing of correct file sizes even for multimember files



Distributed index

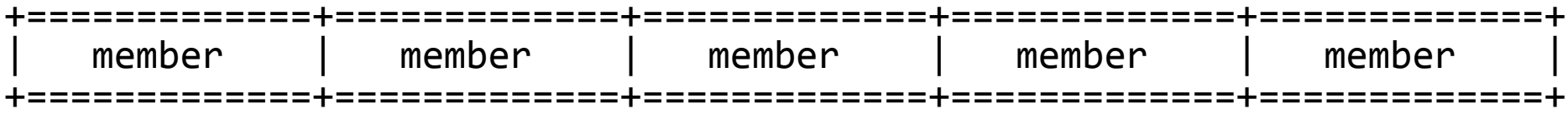


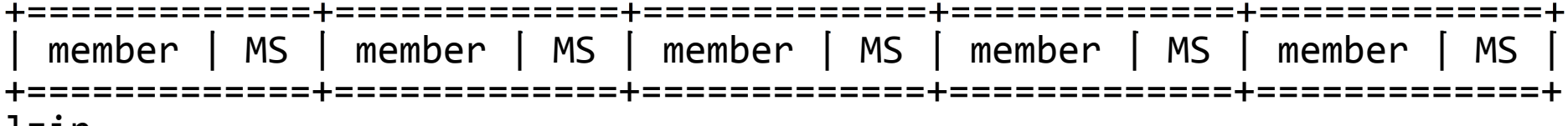
The diagrams for gzip and bzip2 formats are simplified.
The bzip2 trailer is not byte-aligned, but this is not important here.

Parallel compression and decompression

- Parallel compression is easy; split, compress, write in order
- Efficient parallel decompression requires an indexed format
  - ➔ The index tells the threads where to start decompressing
  - ➔ The index makes plzip fast and scalable
- Plzip also decompresses data from non-seekable streams in parallel
  - ➔ 'Member size' (MS) validates identification string in lzip header

gzip





lzip

Implementations

The lzip family:

Lzip	The reference implementation (C++).
Clzip	A C implementation of lzip for systems lacking a C++ compiler.
Plzip	A multi-threaded compressor using the lzip file format.
Lzlib	A compression library for the lzip file format, written in C.
Lziprecover	A data recovery tool and decompressor for the lzip format.
Lunzip	A decompressor for lzip files, written in C.
Pdlzip	A limited, "public domain" C implementation of lzip.
Lzd	An educational decompressor for the lzip format.
Tarlz	An archiver with multi-threaded lzip compression.

Other people maintains tools and libraries with support for the lzip format.

Quality of implementation

Three independent compressor implementations:

lzip, clzip, lzlib.

- Tested to verify that they produce identical output
- Tested with unzcrash, valgrind and 'american fuzzy lop'
- No data-losing bugs in lzip since 2009
- Adequate for both new (64-bit) and old (32-bit) hardware
- Adequate for embedded devices

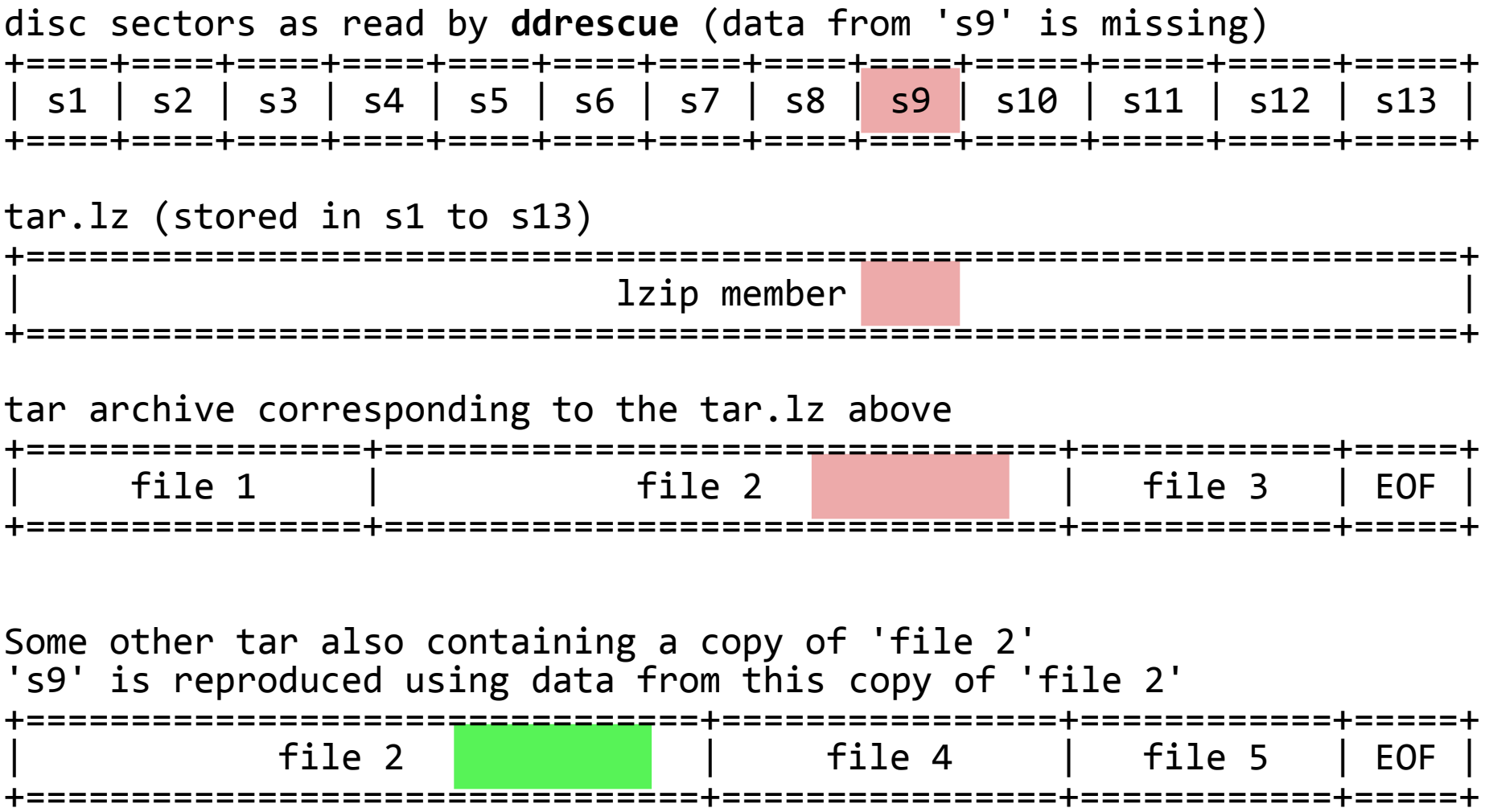
Earth friendly : -)

Lziprecover

Capabilities

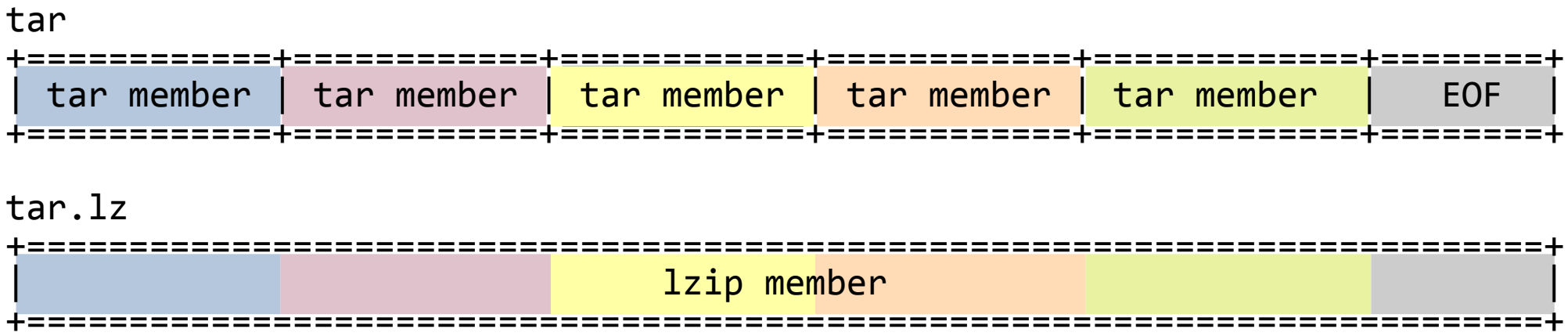
- Decompress
- Test integrity
- Repair slightly damaged files
- Merge the good parts of two or more damaged copies
- Reproduce a missing sector using a reference file
- Extract the recoverable data from damaged files
- Remove the damaged members from multimember files
- Provide random access to the data in multimember files
- Manage metadata stored as trailing data in lzip files

Lziprecover – reproduce mode



tar + lzip

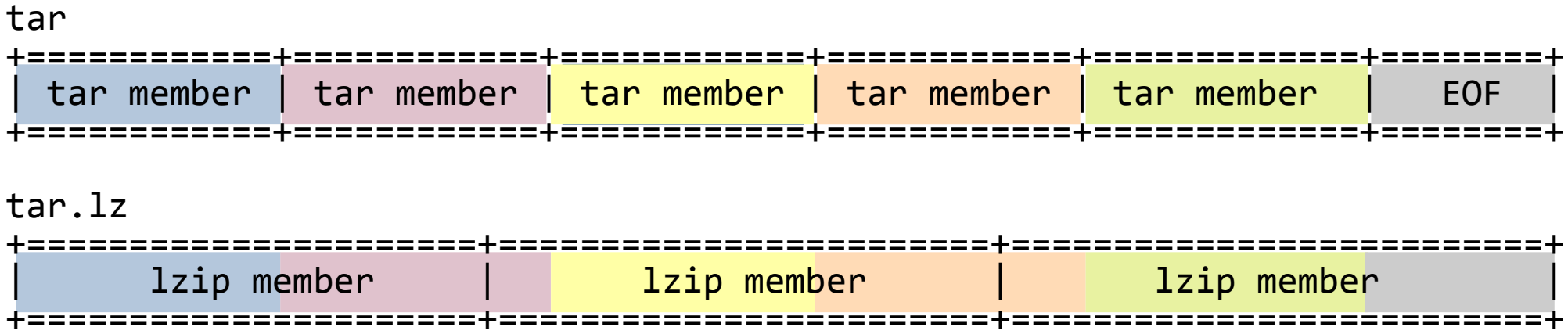
Diagram of single-threaded solid tar.lz compression



- Maximum compression ratio
- Can't be decompressed in parallel
- Can't be decoded (extracted) in parallel, not appendable
- Inefficient extraction of a single file
- Maximum data loss in case of corruption (half archive)

tar + plzip

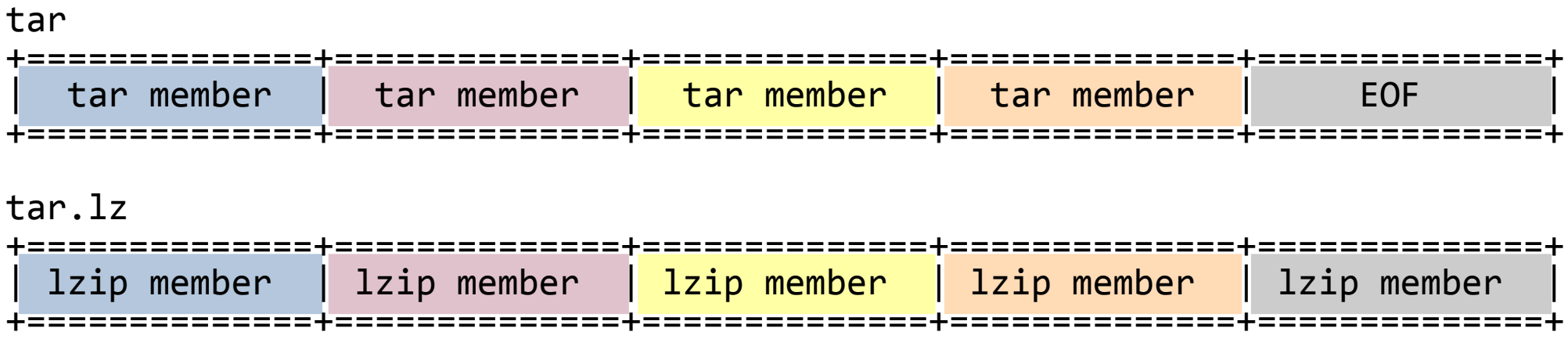
Diagram of plzip (unaligned) tar.lz compression



- Slightly lower compression ratio
- Can be decompressed in parallel
- Can't be decoded (extracted) in parallel, not appendable
- Inefficient extraction of a single file
- Less data loss in case of corruption (half lzip member)

tarlz --no-solid

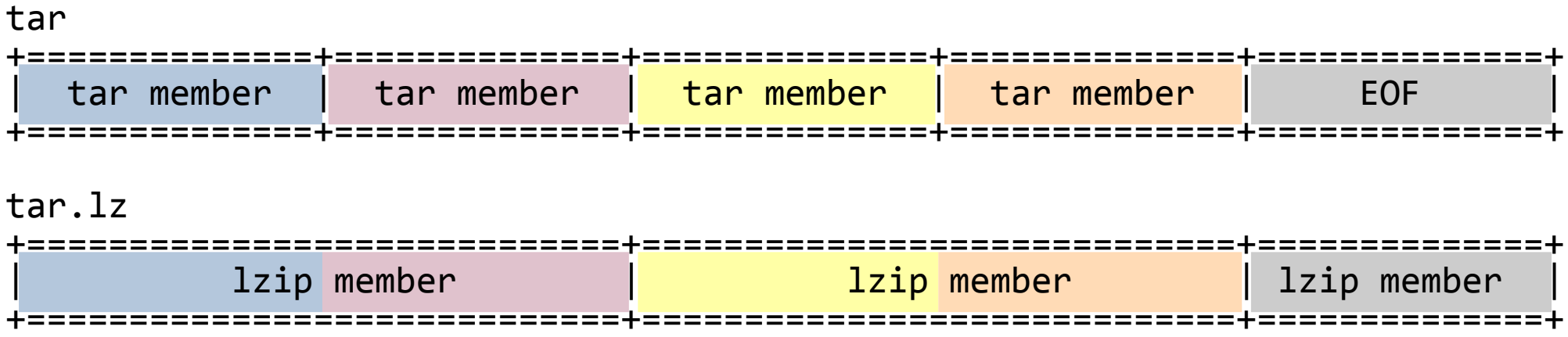
Diagram of tarlz aligned tar.lz compression



- Lower compression ratio (depending on lzip member size)
- Can be decompressed in parallel
- Can be decoded (extracted) in parallel, appendable
- Efficient extraction of a single file
- Minimum data loss in case of corruption

tarlz --bsolid

Diagram of tarlz grouped aligned tar.lz compression



- Good compression ratio (depending on lzip member size)
- Can be decompressed in parallel
- Can be decoded (extracted) in parallel, appendable
- Efficient extraction of a single file
- Less data loss in case of corruption (half lzip member)

Tarlz

What is tarlz?

- A multi-threaded combined implementation of tar and lzip

Why is a combined implementation needed?

- Because for multi-threaded tools, archive creation and archive compression are a single task, not two
- A single tool controlling both archiving and compression is required to guarantee the alignment between tar members and compressed members

Tarlz

Features

- Tarlz brings the compressed tar format to the multicore era
- Multi-threaded creation, extraction and listing
- Allows operations only available before with uncompressed tar
  - Appending
  - Concatenation
  - Efficient extraction or listing of one file
- Fully backward compatible with standard tar tools like GNU tar

Who is using the lzip format?

Support in:

- Automake
- GNU tar, bsdtar, star, RPM, KDE ark, GNOME archive manager...
- Guix lzlib module

Source and data distribution:

- Several GNU and nongnu packages
- Guix substitutes (packages)
- Dragora GNU/Linux
- IANA timezone database
- European Parliament

The end

Thank you for your attention!

Questions?

http://www.nongnu.org/lzip/lzip.html lzip-bug@nongnu.org