Fast File Compressor (FFC) application example

- Problem: compress all files in a directory tree in parallel. The same of the following command:

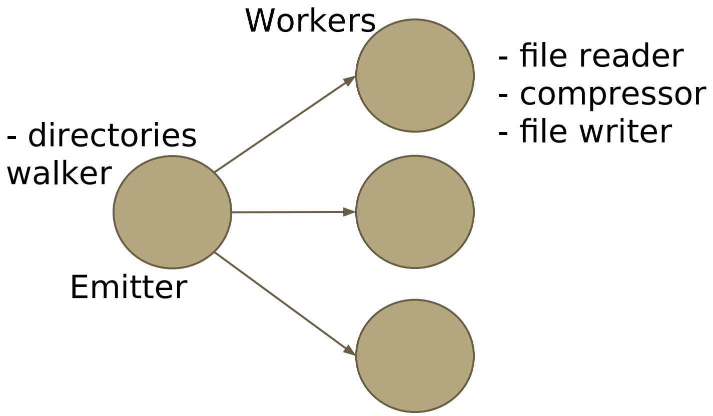
```
find -type f -print0 | xargs -0 -P N -n 1 gzip
```

  Here ‘N’ il the number of gzip processes to spawn

- Idea: compress small file in parallel with different Worker threads, files are compressed by using miniz
- **Miniz** is a single file all-in-one compressor/decompressor
  - Miniz project: https://github.com/richgel999/miniz

FFC initial version

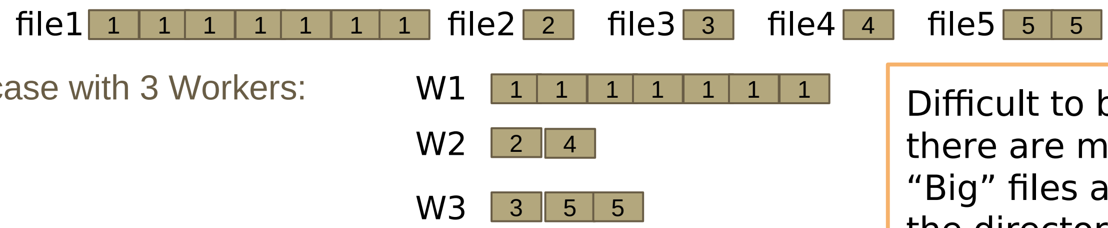
- Initial version



This version works well only if there are many files all having almost the same size

“Big files” problem

- The farm-based (master-worker) configuration does not help when we need to compress a few “Big files”
- Example:



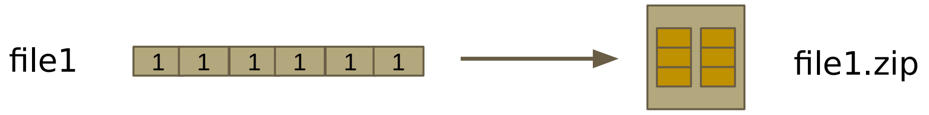
Best case with 3 Workers:

Difficult to balance the workload unless there are many files, and “Small” and “Big” files are distributed uniformly in the directory tree

- What if we split “BIG files” in multiple blocks (▭)?

Splitting Big files

- We have to use the low-level API of Miniz
- Contiguous blocks of data are not independent (i.e., cannot be compressed in a simple way)
- … but we can split the “BIG files” in multiple files and compress them independently. Then we have to merge all compressed parts in a single (**non standard**) zip file, for example by using ***tar***.

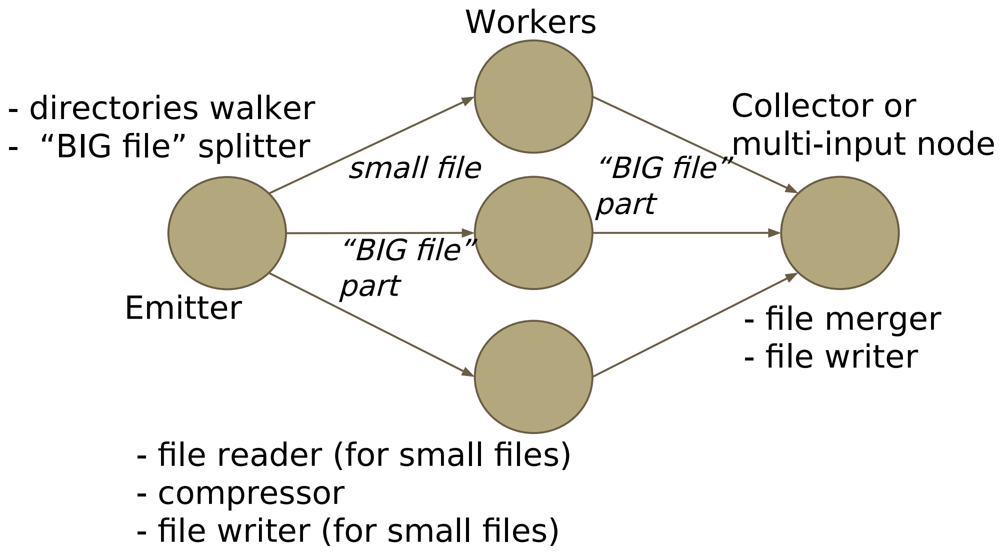


- This means that we have to build our decompressor for such compressed “BIG files”.
- However, this approach is easier than working with Miniz streams (or gzip streams), it does not loose too much in terms of compressed size, and allows us to speed-up the compression of “BIG files”.

FFC proof-of-concept solution

- We modified the base farm version so to schedule both small and “BIG files”. “BIG files” are split on the basis of a user-defined BIGFILE_LOW_THRESHOLD, compress those parts with Miniz and “merge” them in a single file
  - Naive approach for merging multiple parts (filename.part1.zip, filename.part2.zip,..., filename.partK.zip):

    tar cf filename.zip filename.part*.zip



Compression of one “BIG file” 1.1GB (binary data

Versions	Time	Compr. size
Miniz sequential	43s	261M
gzip sequential	57s	256M
ffc sequential	43s	261M
ffc parallel (**)	2.1s	261M

(*) dual socket Xeon E5-2695 @2.4GHz server
(**) 48 Ws, no mapping, blocking mode