

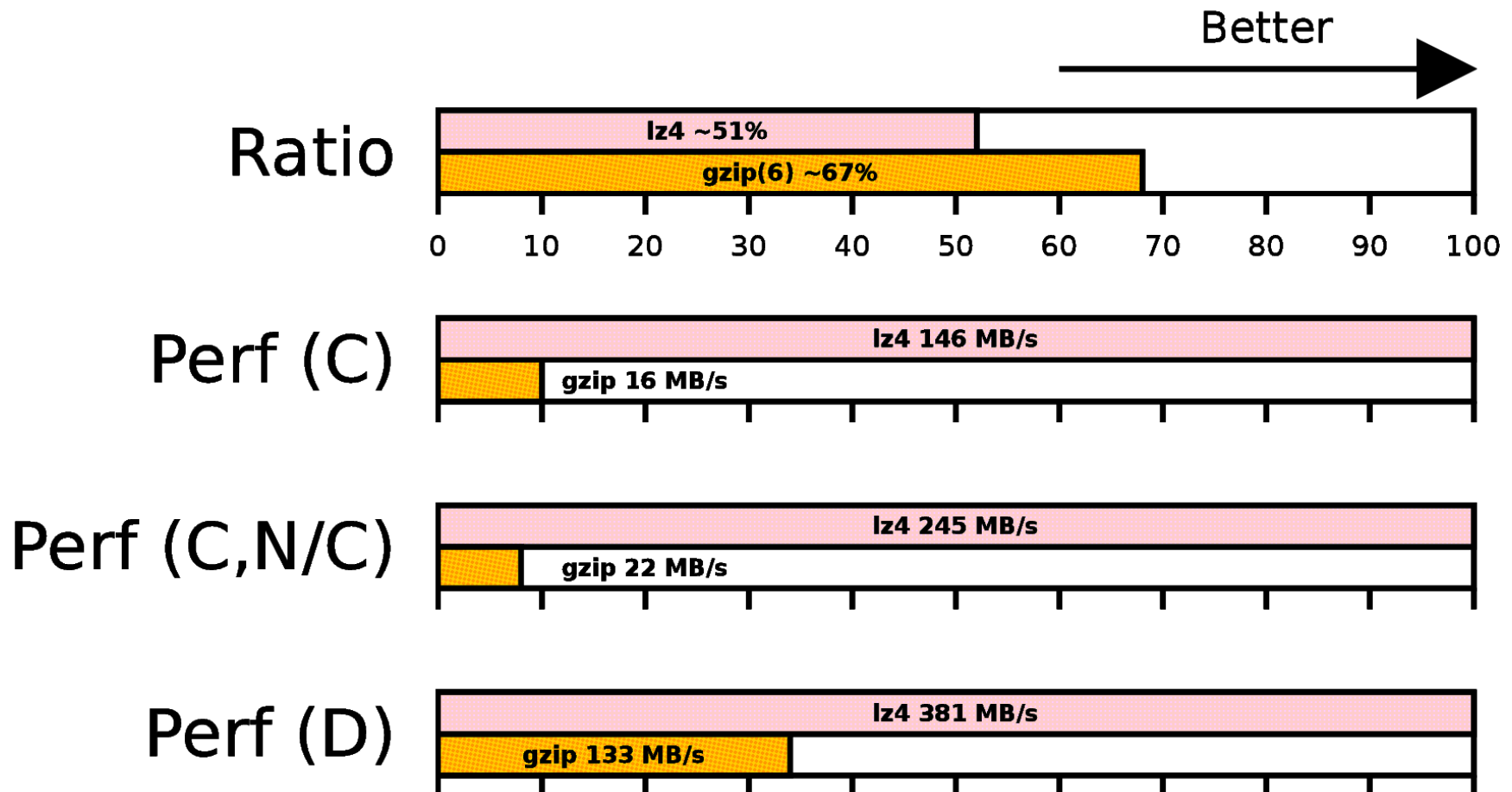
# ZFS (Smart?) Compression

Musings about pointless work  
and how to do less of it

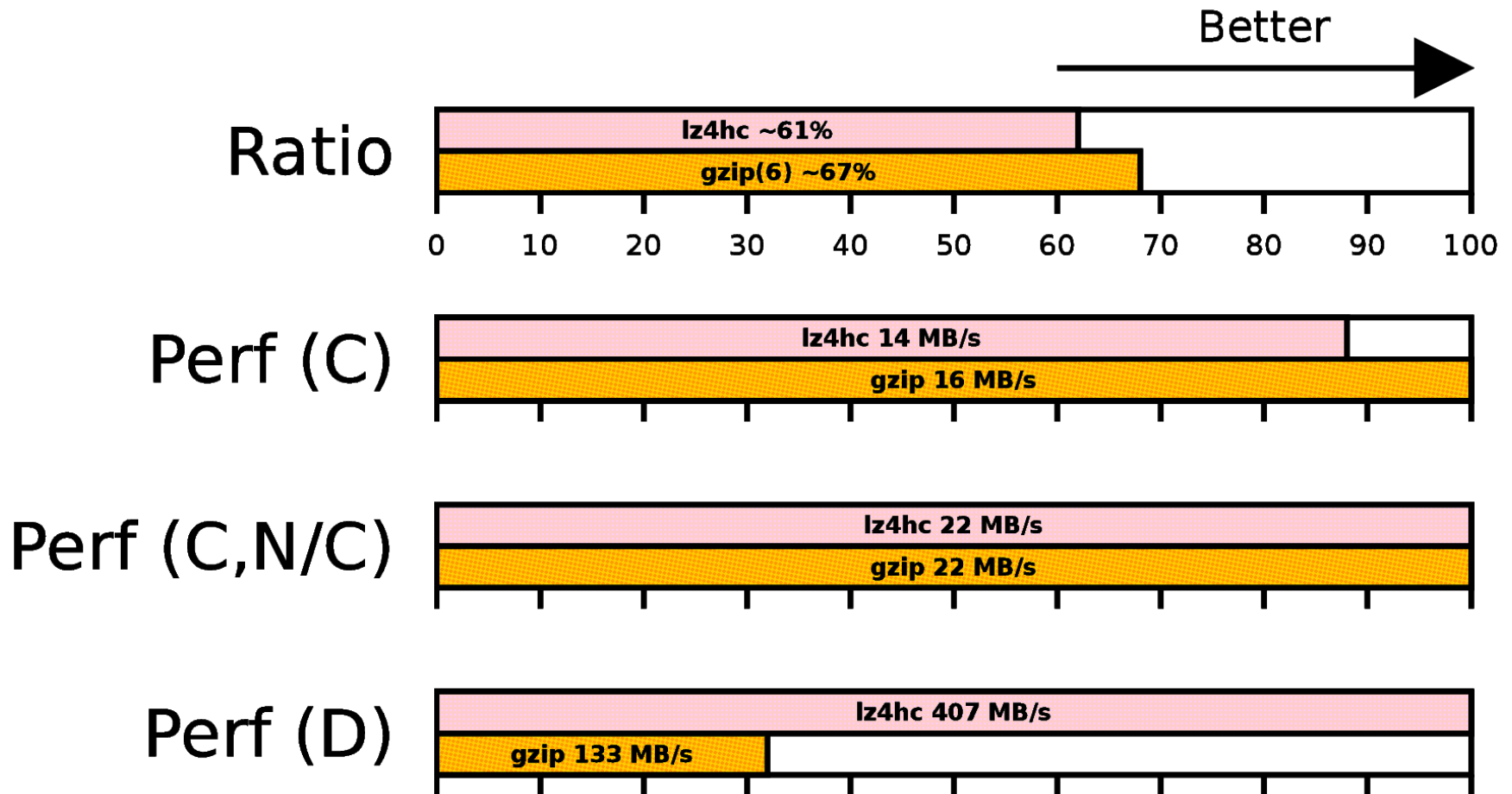
# Types of Compression Algos

- Archiving algorithms:
  - Good at compression ratio, but highly CPU intensive (read: slow)
  - gzip, bzip2, LZMA
- Real-time algorithms:
  - Trade compression ratio for speed
  - lzjb, lz4, lzf

# Why we bother with real-time algos



# Sort-of middle ground solutions



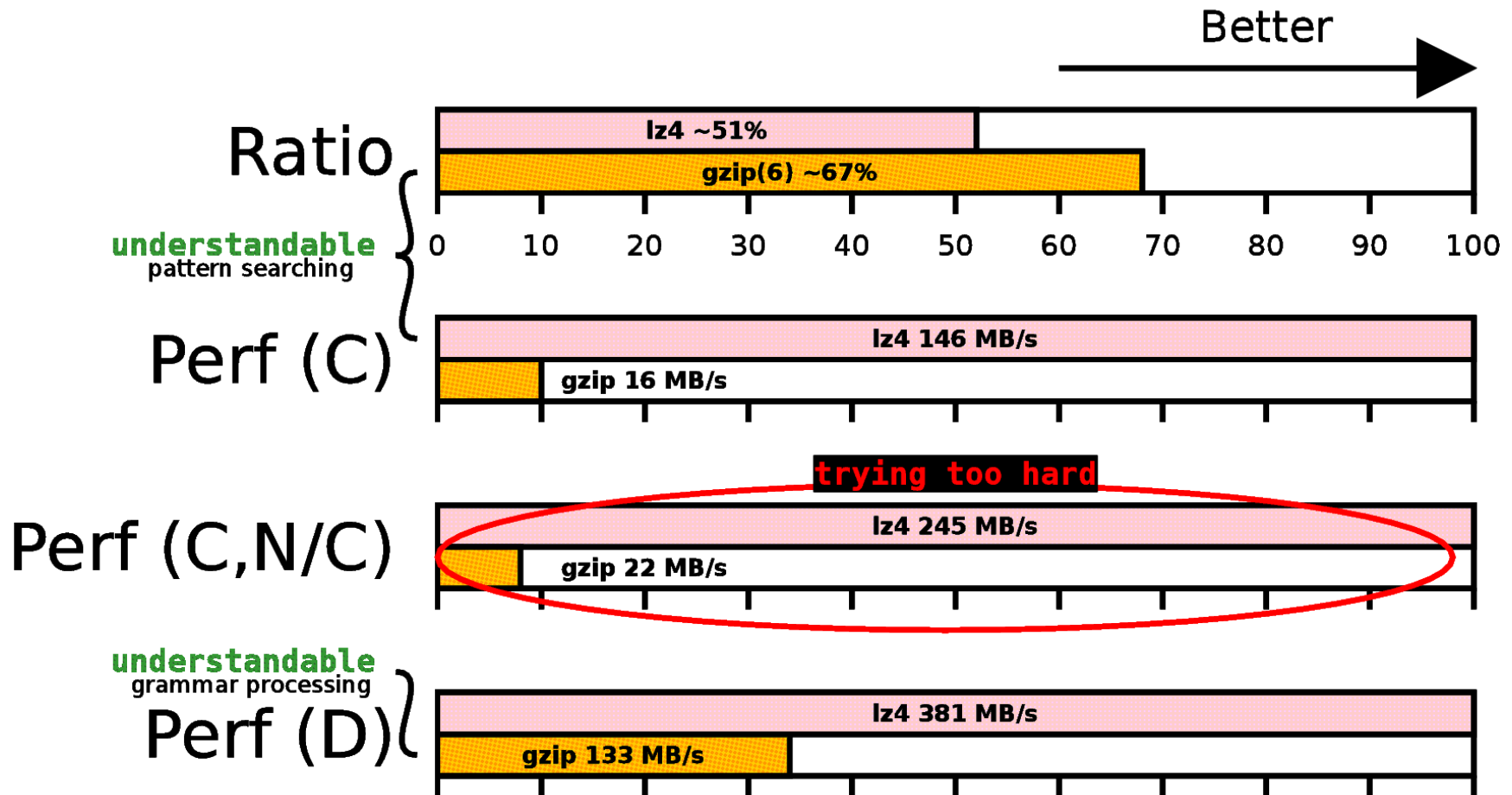
# So why archivers?

- Sometimes space savings are worth it:
  - Write once, read many workloads (initial CPU cost diluted over many reuses of the savings achieved)
  - High bandwidth costs
  - Certain workloads compress really well (up to or over 90%):
    - text files
    - HTML documents
  - Certain workloads, however, don't compress at all:
    - Pre-compressed stuff (multimedia, file archives, etc.)

# What drives **compression** settings

- Am I prepared to pay the CPU cost?
- Is my workload compressible?
  - **This shouldn't really be a question, but it is**
- We're a filesystem and individual files typically are either compressible or incompressible:
  - Compressible: .txt, .html, .doc, .wav
  - Incompressible: .mpg, .mp3, .gz
- But our **compression** setting is per FS!

# Why we ask the compressibility question



# Bad Solutions

- Teach ZFS to recognize file extensions:
  - There's just too damn many of them...
  - Renaming a file changes our behavior?
    - **Security:** wanna escalate costs for your storage provider as much as possible? Rename everything .gz and fill with zeros.
- Let admin control “compression” per file:
  - GL trying to keep track of this with millions of files
  - Often admin != user (shared storage environment)

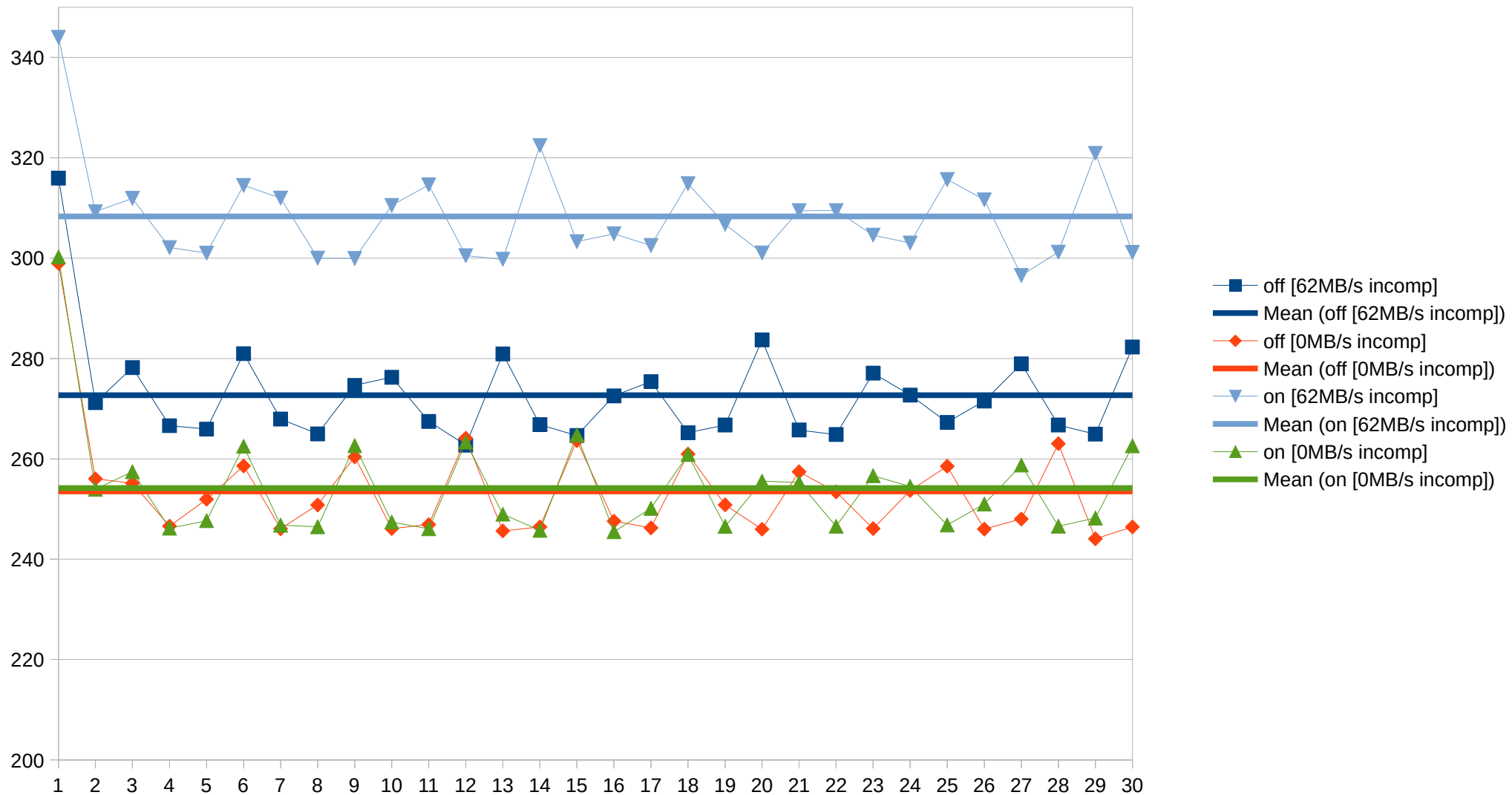


# ZFS Smart Compression

- We dynamically track per-file compression performance
- No change to on-disk format (in-core state only)
- Based on previous compression results we progressively back off retrying compression, or, if compression succeeds often, are more reluctant to skip it
- Works even on composite data files (VMDKs)

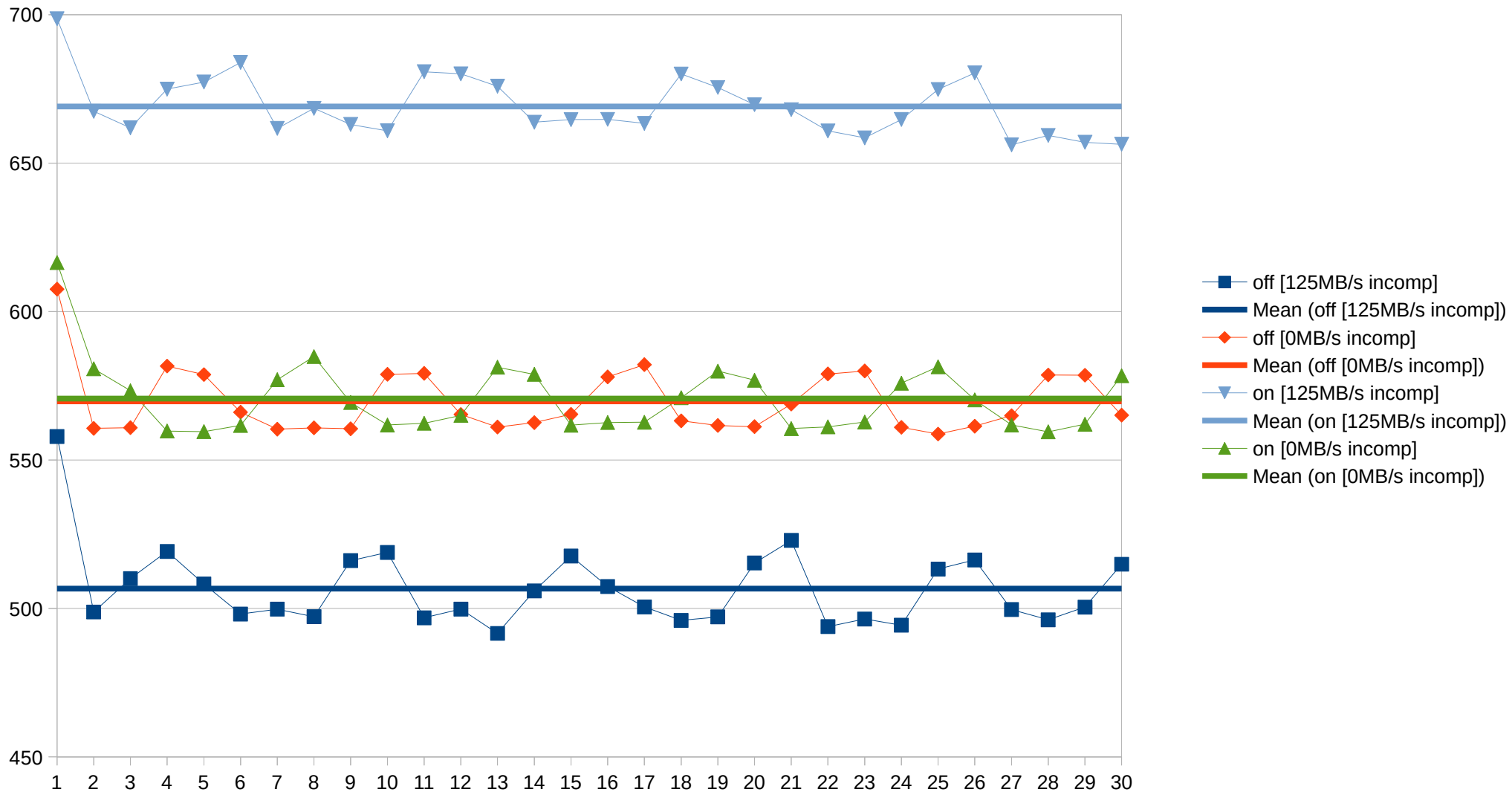
# Smart Compression by the numbers

gzip-6



# Smart Compression by the numbers

gzip-1



# Thanks!

Questions? Discussion? Rotten tomatoes?