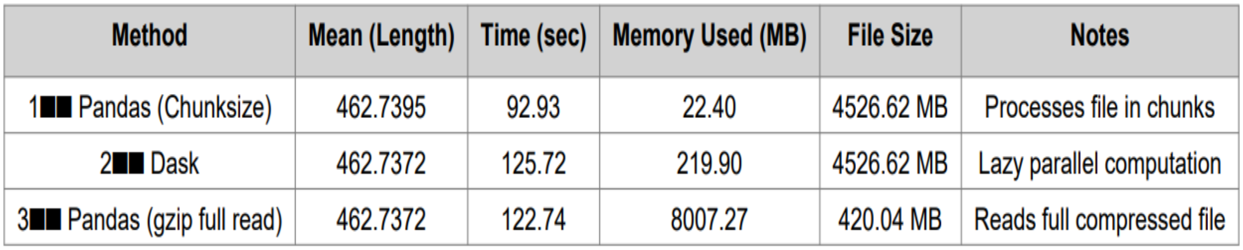
# Big Data Loading Methods Report



| Method | Mean (Length) | Time (sec) | Memory Used (MB) | File Size | Notes |
|---|---|---|---|---|---|
| 1■■ Pandas (Chunksize) | 462.7395 | 92.93 | 22.40 | 4526.62 MB | Processes file in chunks |
| 2■■ Dask | 462.7372 | 125.72 | 219.90 | 4526.62 MB | Lazy parallel computation |
| 3■■ Pandas (gzip full read) | 462.7372 | 122.74 | 8007.27 | 420.04 MB | Reads full compressed file |

## Method 1 — Chunksize

**Advantages**:

- Efficient for very large files.
- Low and controlled memory usage.
- No extra libraries needed.

**Disadvantages**:

- Slightly slower due to chunk processing.
- Some operations require manual aggregation.

**When to Use:**

- Use for very large files that cannot fit into RAM.

## Method 2 — Dask

**Advantages**:

- Handles very large datasets efficiently.
- Parallel and distributed computation.
- Moderate memory usage (≈220 MB) and good scalability.

**Disadvantages**:

- Slightly slower than Pandas (due to scheduling overhead).
- Some pandas functions not fully supported.

**When to Use:**

- Best for large datasets and multi-core processing, especially when memory is limited but parallelism is available.

## Method 3 — Compressed CSV (gzip, full read)

**Advantages**:

- Saves disk space drastically.
- Simple single-step loading.

**Disadvantages**:

- High memory usage (≈8 GB).
- Slowest method when decompression is included.

**When to Use:**

- Suitable for small-to-medium datasets that fit in memory.