Comparing benchmarking experiment results from Release 2.5 and Release 2.6

Gyuri Barabás

Introduction

This report analyzes and compares the results of the latest two benchmarking experiments. The first was run in June 2025, just before Release 2.6 went out. We will call this the "Release 2.5" experiment. The other one was run in July 2025 just after (the "Release 2.6" experiment). The comparison of the Release 2.5 and Release 2.6 experiments ought to reveal whether any of the download and upload speeds were affected by the new release, and how.

We begin by a recap of the experimental design. The purpose is to measure up- and download speeds on Swarm for various file sizes, erasure settings, and retrieval strategies. We vary all parameters in a fully factorial way, to get at all their possible combinations. The factors are as follows:

- `size`: The size of the uploaded random file. We have 6 distinct factor levels: 1 KB, 10 KB, 100 KB, 1 MB, 10 MB, and 50 MB. Every file has a size that matches one of these values exactly. Importantly, every single upload is a unique random file, even if the file sizes are otherwise equal—this removes the confounding effects of caching.
- `erasure`: The strength of erasure coding. We have five factor levels: `0` (= `NONE`), `1` (= `MEDIUM`), `2` (= `STRONG`), `3` (= `INSANE`), and `4` (= `PARANOID`).
- `strategy`: The retrieval strategy used to download the file. Its value is necessarily `NONE` in the absence of erasure coding—i.e., when `erasure = 0`. Otherwise, it is either `DATA` or `RACE`.
- `server`: The identity of the server initiating the downloads might influence download speeds. For a fair comparison, servers should be identical within an experiment, but it makes sense to perform the whole experimental suite over multiple different servers, to control for server-specific effects. This means that we have an extra experimental factor, with as many distinct levels as the number of distinct servers used. Here we use three distinct servers: `Server 1`, `Server 2`, and `Server 3`.
- `replicate`: To gain sufficient sample sizes for proper statistical analysis, every single combination of the above factors is replicated 30 times. For example, given the unique combination of 1MB files uploaded without erasure coding on Server 1, we actually up- and downloaded at least 30 such files (each being a unique random file).

The above design has (6 file sizes) × (5 erasure code levels) × (3 retrieval strategies) × (3 servers) × (30 replicates). However, the `NONE` retrieval strategy is only ever used when `erasure` is `NONE`, and the `DATA` and `RACE` strategies only when `erasure` is not `NONE`. So the total number of unique download experiments is (30 replicates) × (6 file sizes) × (3 servers) × (1 strategy & erasure level + 2 strategies × 4 erasure levels), or 4860.

Additionally, here are some further notes about the experimental design outlined above:

- All uploads are direct, as opposed to deferred.
- We need to make sure that no download starts after the system has properly stored the data. Since our files are relatively small, uploading should be done in a few minutes at most (as we will see later, the longest upload in our data took below 3 minutes). So we opted for a crude but reliable way of eliminating any syncing issues: we waited exactly 2 hours after every upload, and began downloading only then.
- All downloads are done using nodes with an active chequebook.
- Every download is re-attempted in case of a failure. In total, 15 attempts are made before giving up and declaring that the file cannot be retrieved.

Preliminary data check

All files uploaded without problems, in both experiments. The same holds for downloads, except for 6 failed attempts in the Release 2.6 experiment. These failed download attempts were all for 50 MB files with `STRONG` erasure coding and the `DATA` retrieval strategy, distributed evenly across the three servers (i.e., two fails per each). These six entries were removed from the data before the analysis. This leaves 9714 downloads in our dataset: 4860 per each of the two experiments, minus the 6 failed ones.

Download times

We can visually compare download times across the 2.5 and 2.6 benchmarking runs. We do this both for the `DATA` (Figure 1) and `RACE` (Figure 2) retrieval strategies. For smaller file sizes (especially in the absence of erasure coding), the new release appears to be faster, but for large files it looks clearly slower. To facilitate the comparison, it helps to bring the download times, which vary over several orders of magnitude depending on file size, to the same scale. One can do this by z-transforming download times: for data points within each combination of file size category, server identity, erasure level, and retrieval strategy (this will leave 60 data points in every factor combination category: 30-30 for the Release 2.5 and Release 2.6 runs), we subtract the mean from the download times, and divide this difference by their standard deviation:

$$z_i = \frac{t_i - \bar{t}}{\mathrm{sd}(t)}. \tag{1}$$

Here t_i is the ith measured download time (in seconds), $\bar{t}$ is their mean, and $\mathrm{sd}(t)$ their standard deviation. The resulting z-scores, z_i, are unitless, and have by definition mean zero and variance one. This makes them much easier to compare visually across various experimental factor combinations (Figure 3, Figure 4).

The z-scores do not change the interpretation of the original figures much, but do highlight that the speed gains for small file sizes, while probably real, are not particularly interesting or impressive. To rigorously compare the Release 2.5 and Release 2.6 results within each file size, server, erasure level, and retrieval strategy category, we can perform Wilcoxon rank sum tests.[1] Since we are performing a large number of these tests, p-values should be corrected for multiple comparisons. Here we do this using the false discovery rate method.

[1] It does not matter whether the tests are performed on the z-scores or the original download times, because the Wilcoxon rank sum test only cares about the ranks of values (i.e., whether a value is the largest, second largest, etc.) and not the values themselves. The transformation of Equation 1 preserves the order of ranks. Below we opt to do the test with the original data, to make the resulting estimates correspond to the actual download times as opposed to the z-transformed times.

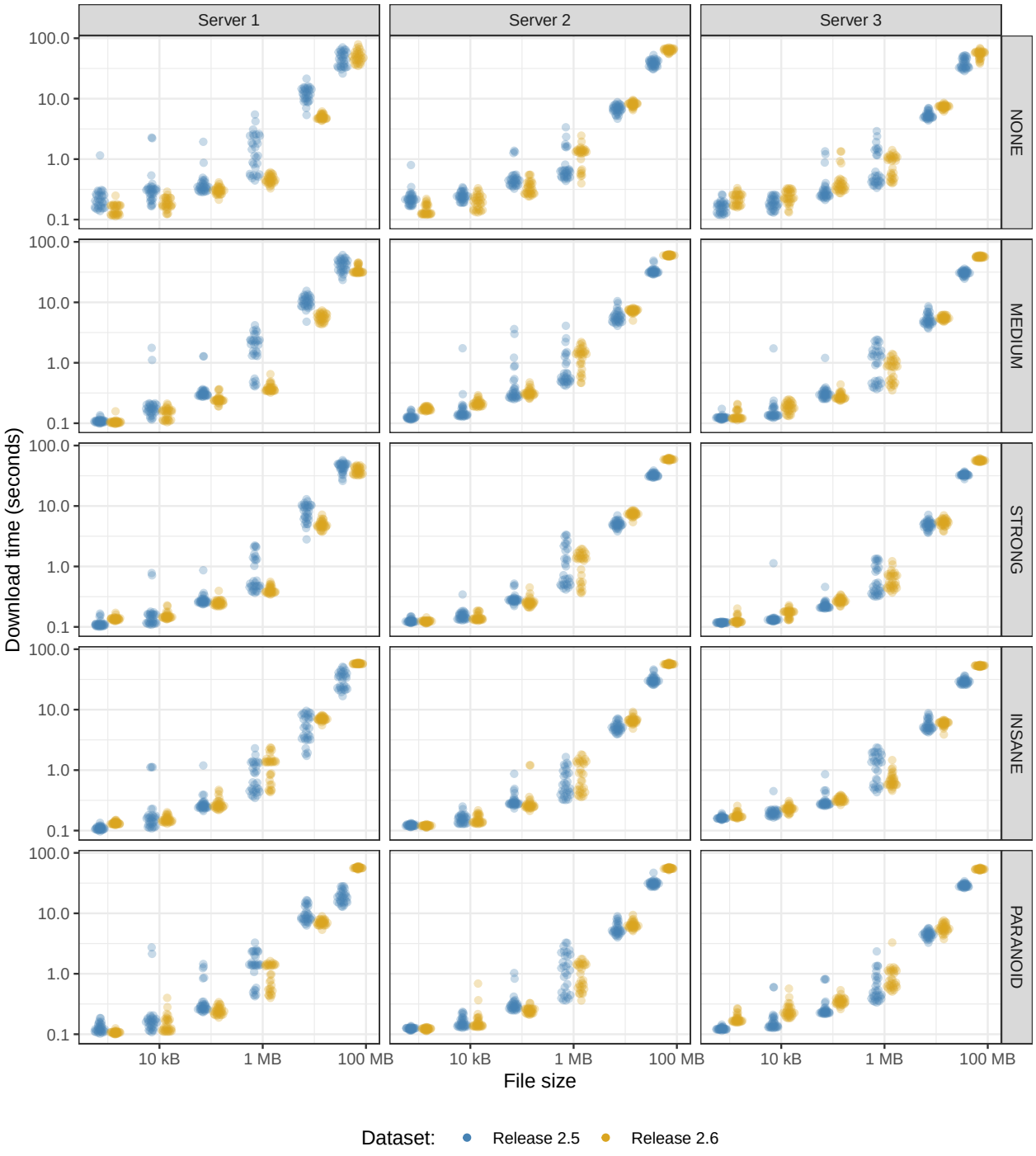

Figure 1: Comparison of the 2.5 (blue) and 2.6 (yellow) benchmarking experiments, for the `DATA` retrieval strategy. Panel rows are erasure levels, panel columns are the three servers, the x-axis is file size, and the y-axis is download time in seconds (both are on the log scale). Each point is one download. Points are arranged to reflect their underlying distribution.

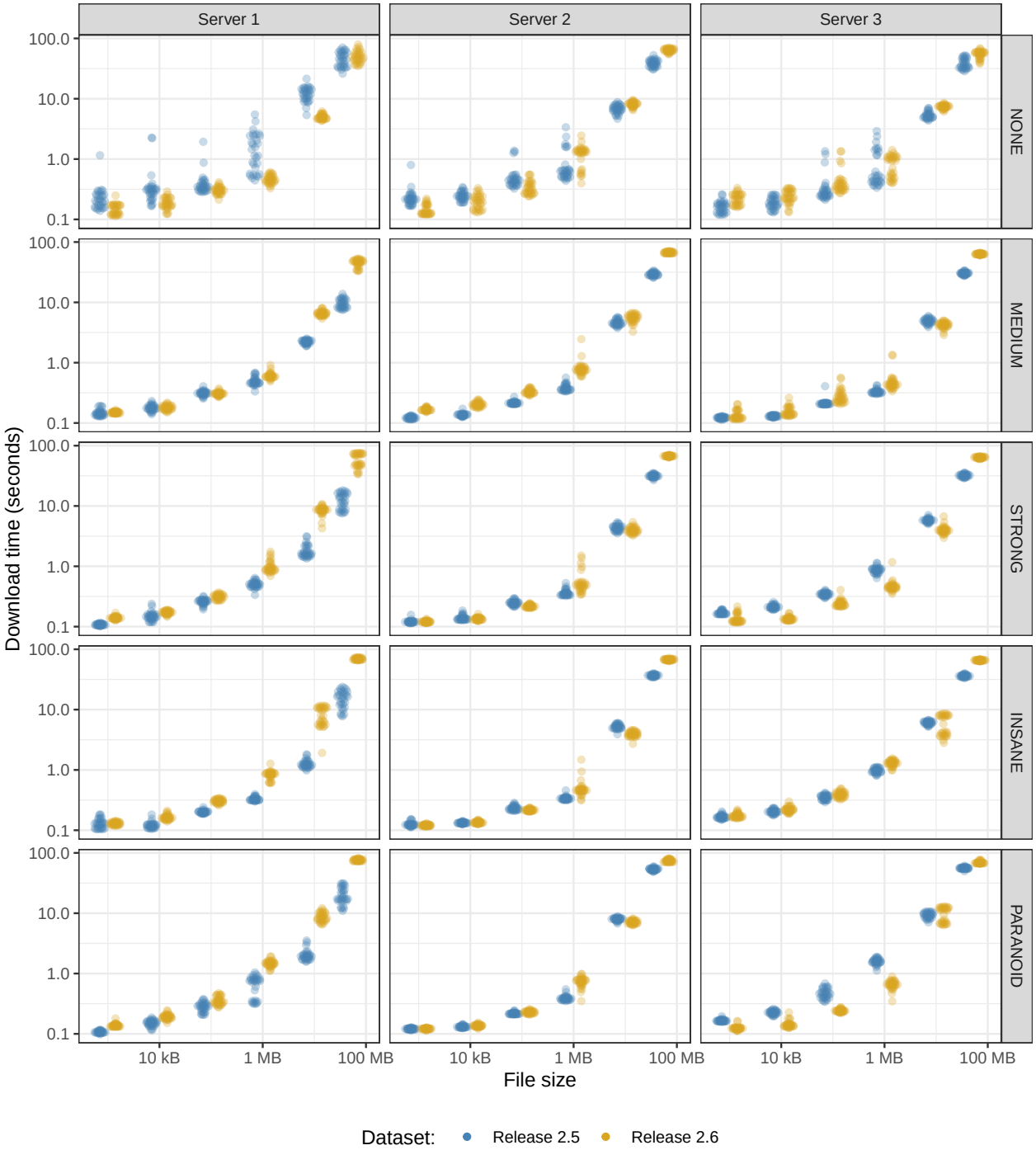

Figure 2: As Figure 1, but for the `RACE` retrieval strategy.

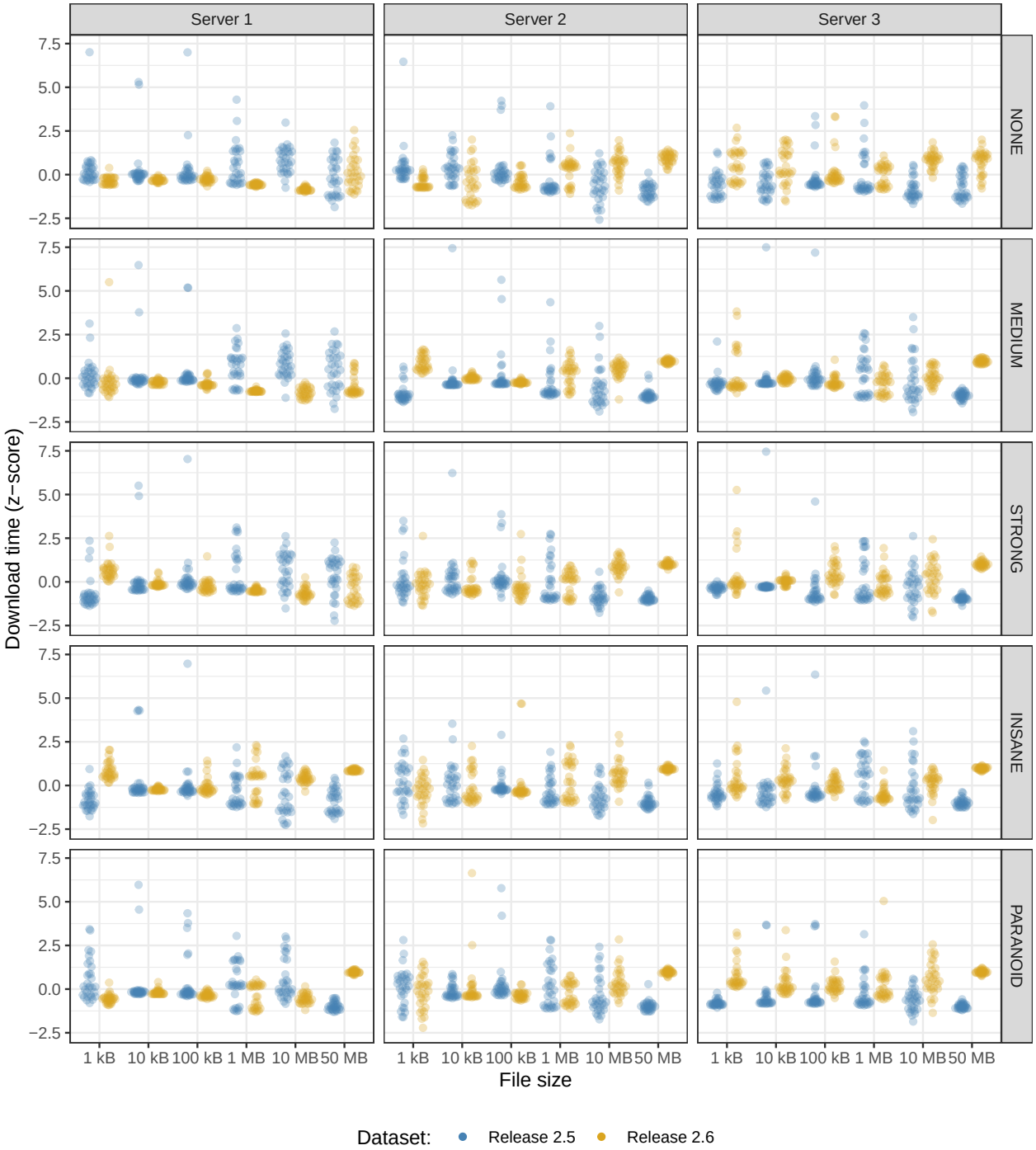

Figure 3: As Figure 1, with z-scores along the y-axis.

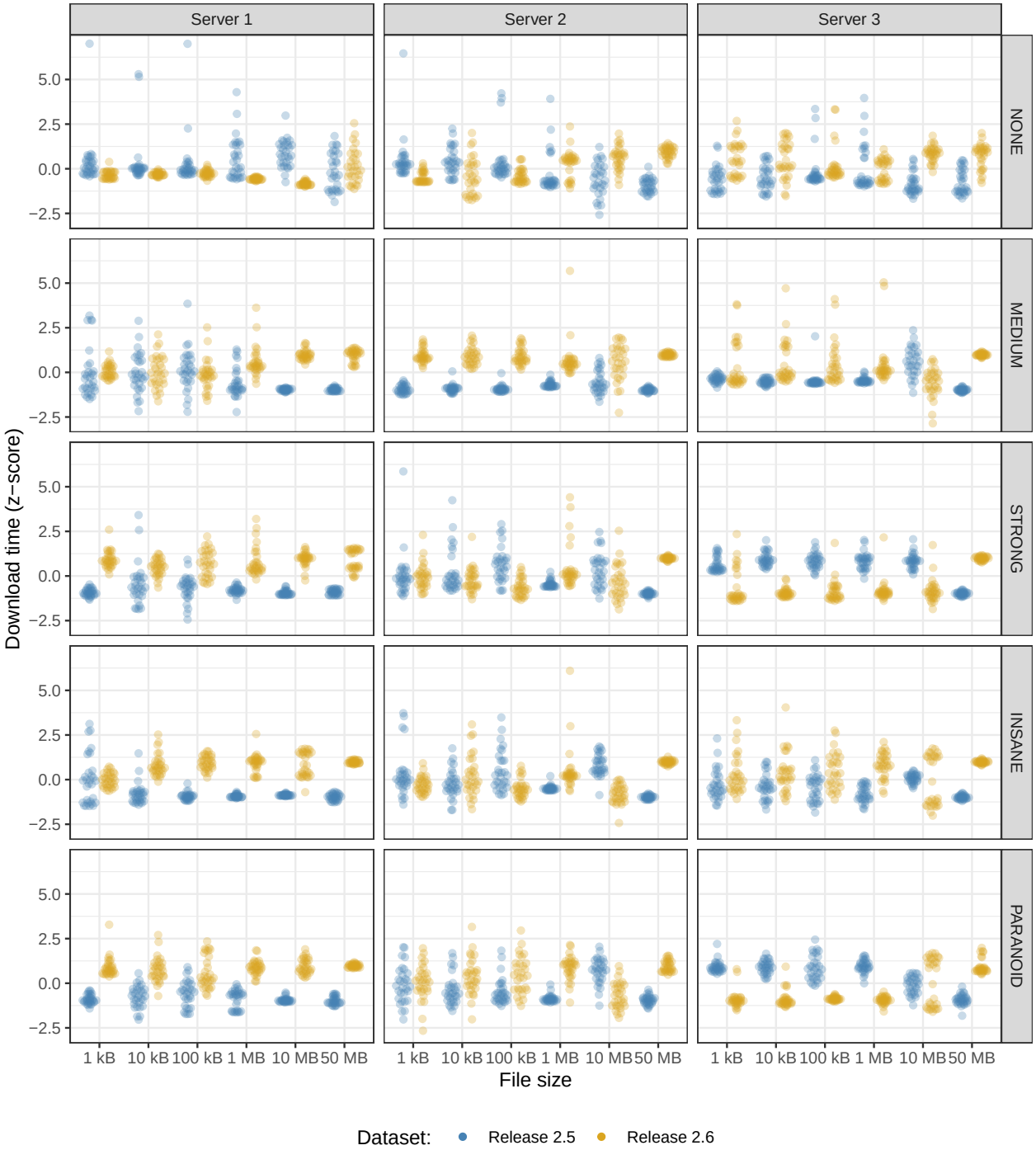

Figure 4: As Figure 3, but for the `RACE` retrieval strategy.

The results of these statistical tests are in Figure 5. We see that our original intuitions are confirmed. Despite statistical significance, for file sizes below 50 MB the observed differences are unlikely to make a practical difference to user experience. However, the slowing down for 50 MB files is more substantial: Release 2.6 is between 10 and 30 seconds slower than Release 2.5. Since downloading 50 MB files typically takes between half a minute to a minute,[2] this is a more serious loss of speed.

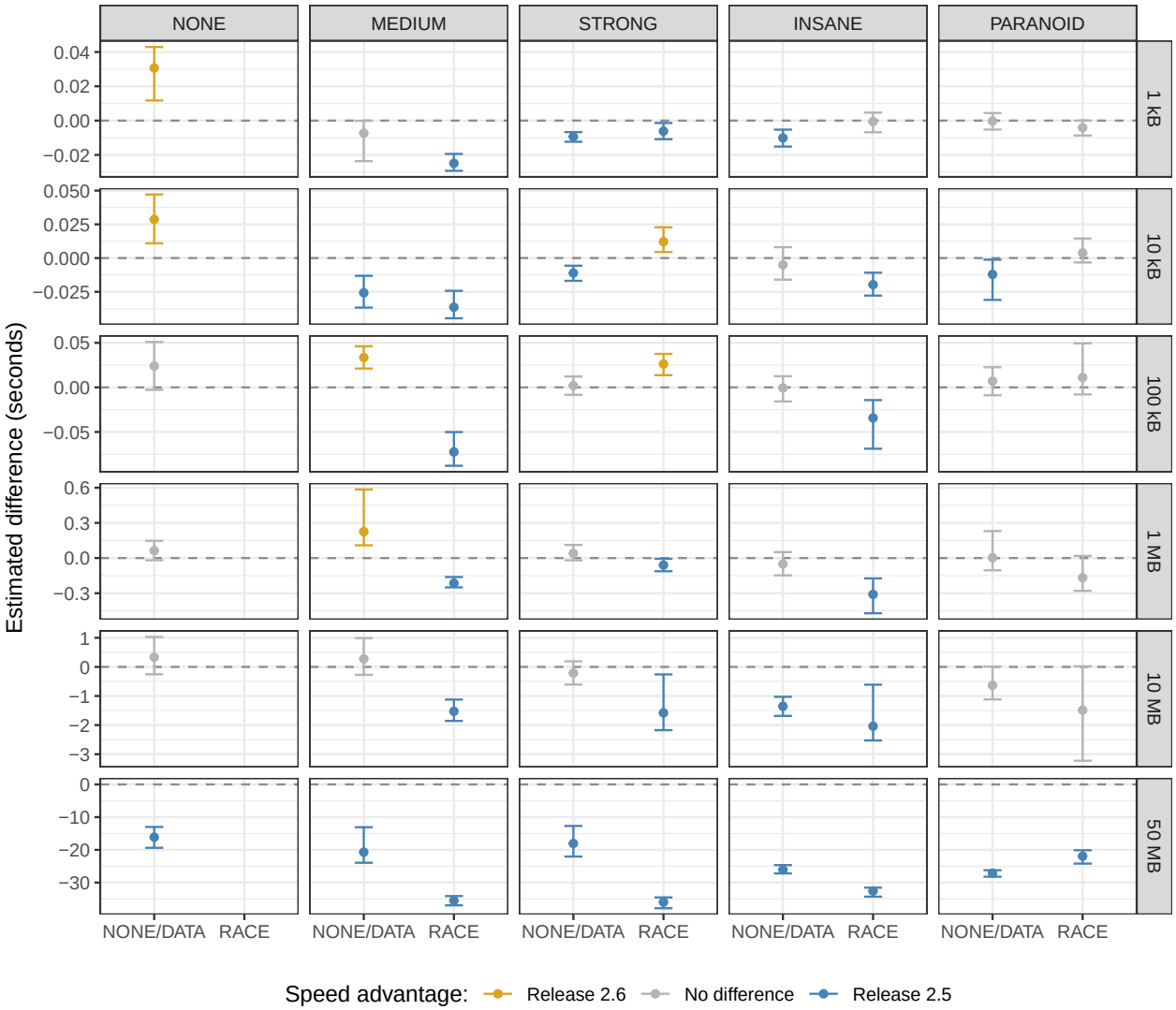

Figure 5: Group-by-group comparison of results from the Release 2.5 and Release 2.6 download time data. The y-axis shows the estimated difference (point) plus/minus 95% confidence intervals (error bars) from a Wilcoxon rank sum test applied to each distinct file size / retrieval strategy / erasure coding combination. Results favoring the new 2.6 release with $p < 0.05$ (after false discovery rate correction) are in yellow, those favoring the old 2.5 release are in blue, and those with $p > 0.05$ (non-significant results) are in grey.

[2]This corresponds to the interquartile range of 50 MB file downloads in our data.

The relative influence of retrieval strategy

We can also check how much advantage the `RACE` strategy offers over `DATA`, for each file size category. The results are qualitatively similar for the 2.5 release (Figure 6) as for the 2.6 release (Figure 7) runs, except for 50 MB files. Here the 2.5 release sometimes gave a speed advantage to the `RACE` strategy. In the 2.6 release, `RACE` appears to be inferior for 50 MB files in general.

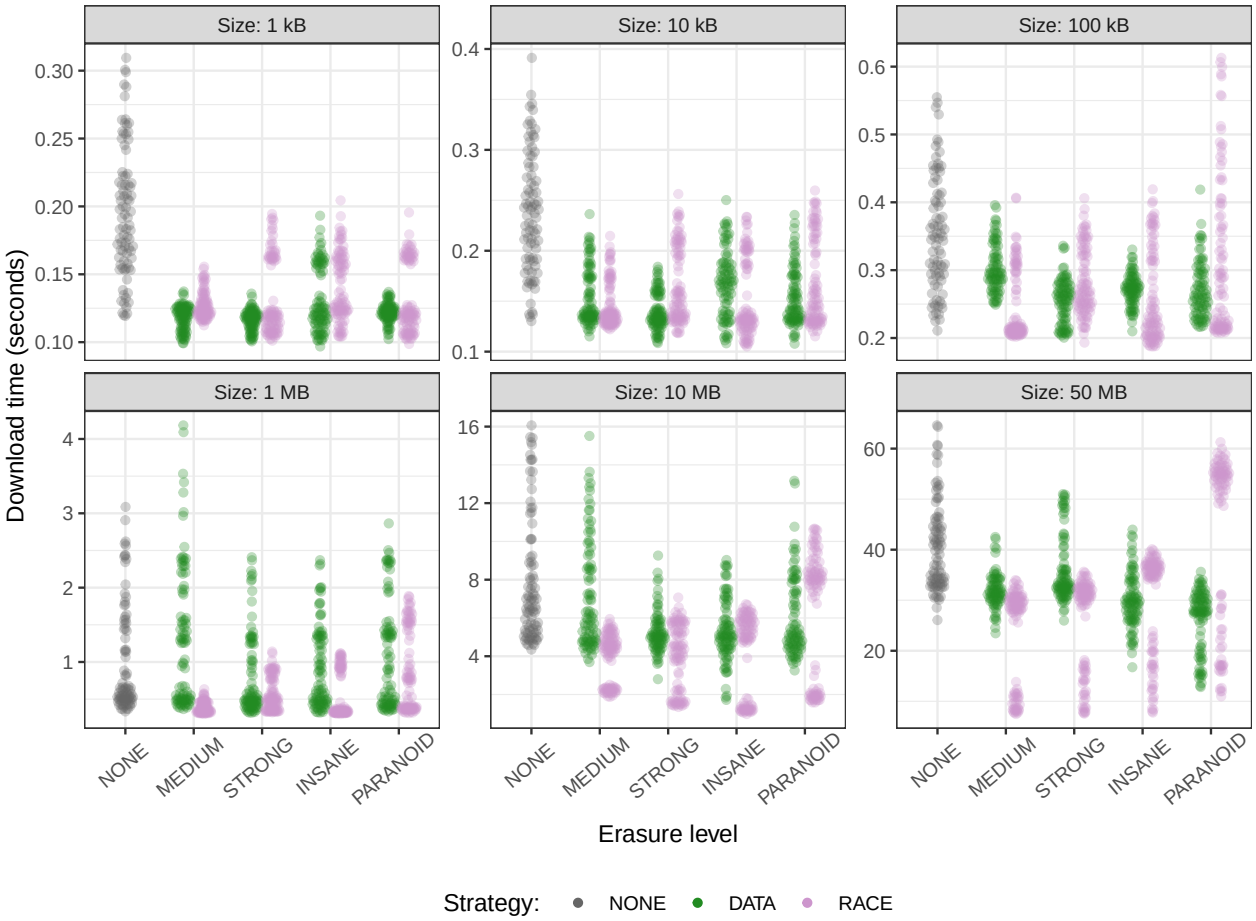

Figure 6: Download times (y-axis) as a function of erasure level (x-axis), file size (panels), and retrieval strategy (colors). Data are only for the Release 2.5 experiment. The y-axis is individually scaled for each panel. High outliers (points more than 1.5 times the interquartile range outside the top quartile in the upper direction) have been removed, because they otherwise distort the plots and make the corresponding results difficult to compare visually.

Upload times

A comparison of the raw upload times from Releases 2.5 and 2.6 are shown in Figure 8. Unlike for download times, it appears clear that Release 2.6 has faster upload times in most file size and erasure coding categories. This impression is further reinforced by looking at the z-scores instead of the raw upload times (Figure 9).

Like in the case of download times, we can make a category-by-category comparison of corresponding upload times across the two experiments, using non-parametric Wilcoxon rank sum tests (and correcting p-values for multiple comparisons using the false discovery rate method). The results are

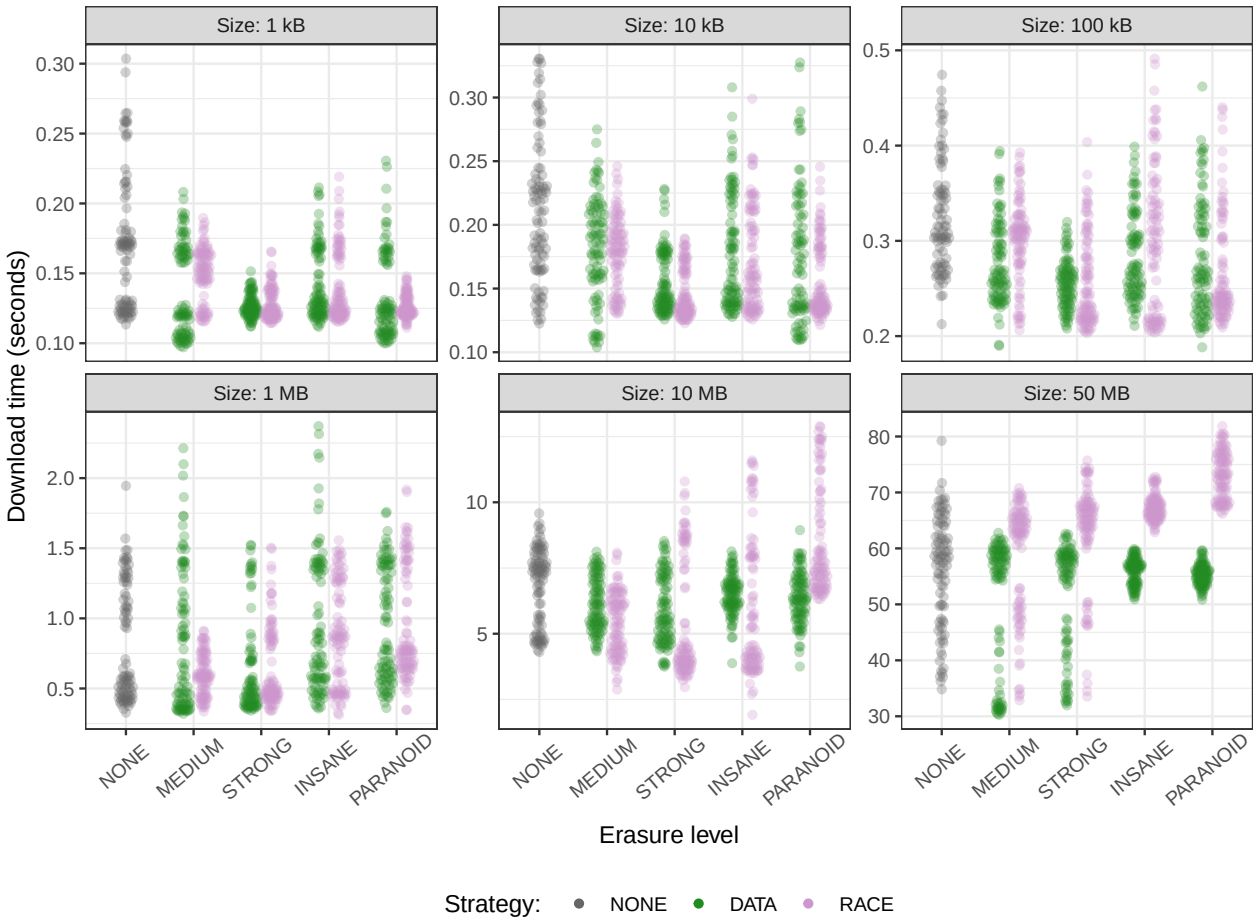

Figure 7: As Figure 6, but for the Release 2.6 experiment.

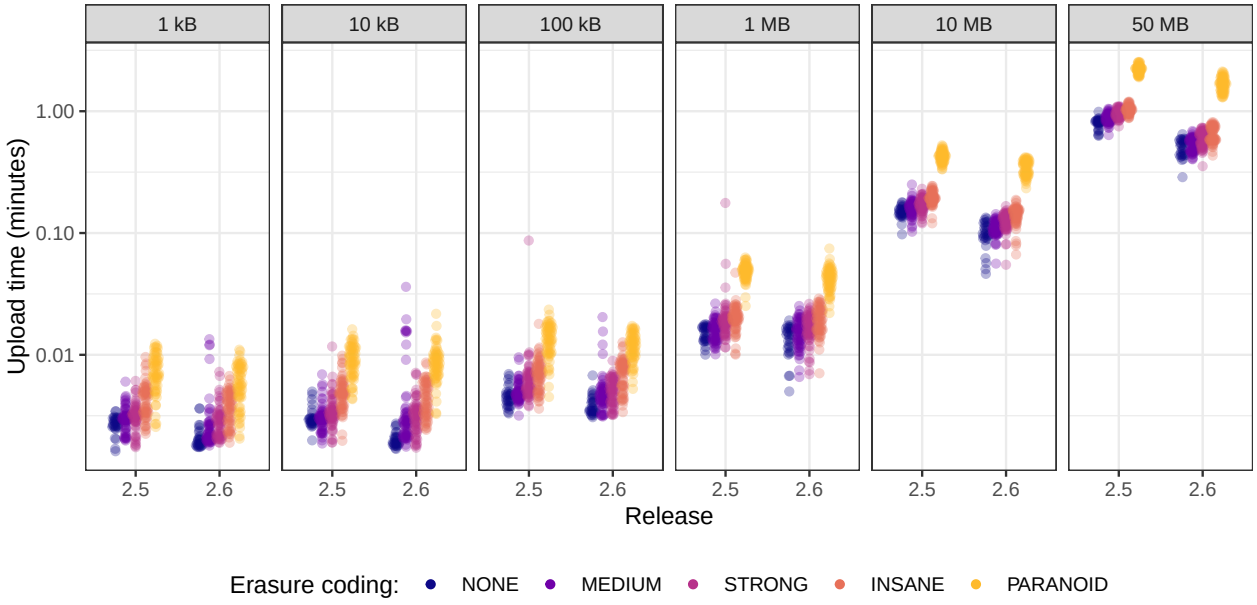

Figure 8: Upload times, in minutes (y-axis; log scale) for the release 2.5 and Release 2.6 benchmarking experiments (x-axis). Panels are different file size categories; colors are various erasure cosing levels.

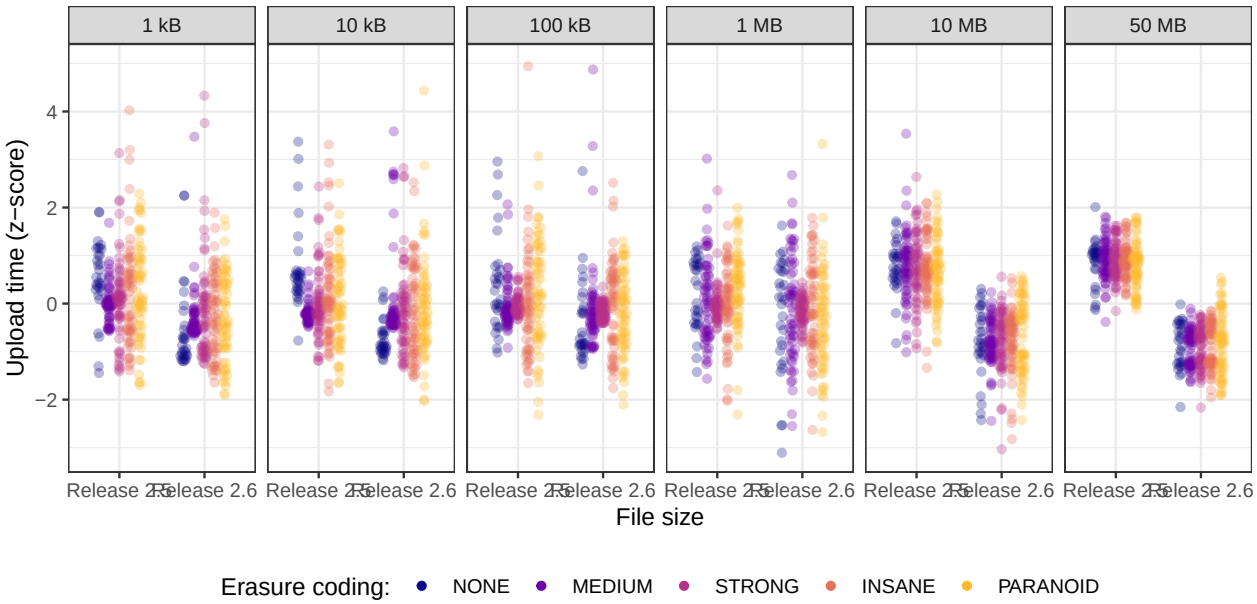

Figure 9: As Figure 8, but with the z-scores of the upload times along the y-axis. The y-axis is truncated at 5, leading to 9 large outliers not being shown (the largest z-score is 10.7). This is to avoid distorting the plot, making the rest easier to compare.

in Figure 10. Indeed, in most cases the new release has clearly faster uploads, and substantially so for larger (10 MB and 50 MB) files. The differences here are large enough to contribute positively to user experience.

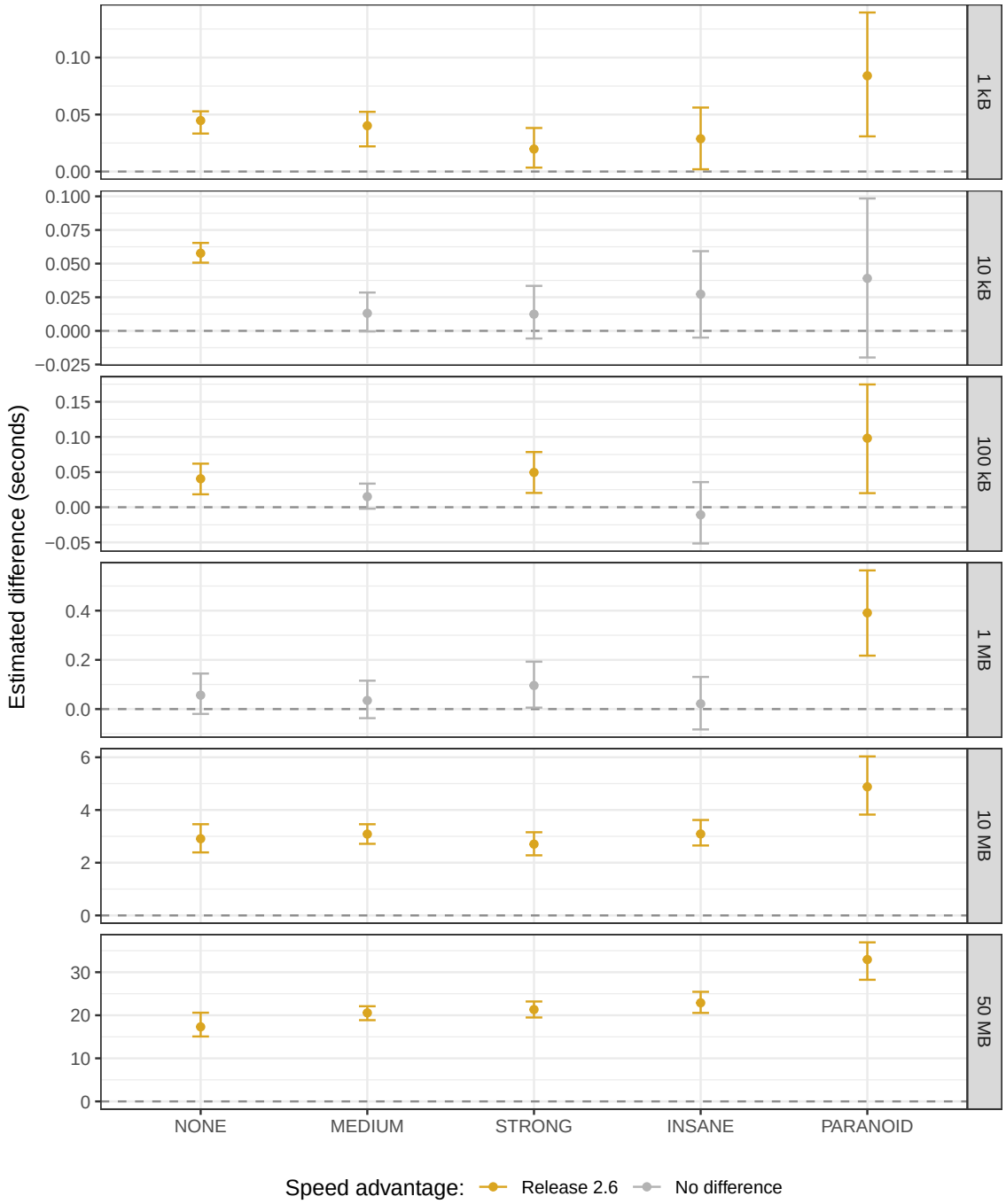

Figure 10: Group-by-group comparison of results from the Release 2.5 and Release 2.6 upload time data. The y-axis shows the estimated difference (point) plus/minus 95% confidence intervals (error bars) from a Wilcoxon rank sum test applied to each distinct file size / retrieval strategy / erasure coding combination. Results favoring the new 2.6 release with $p < 0.05$ (after false discovery rate correction) are in yellow, and those with $p > 0.05$ (non-significant results) are in grey. (Those favoring the old 2.5 release would be in blue, but there aren't any such results here.)