

INTRODUCTION

On February 7th, 2019, PCIC released a new version (2.6.0) of our PCIC Data Portal (the “pdp”), that replaced our original set of statistically downscaled climate scenarios (BCCAQ) with an updated methodology (BCCAQv2). Our pdp code has been running in production for 5-6 years and has seen relatively few problems, so we expected this dataset to work well, and our testing indicated that this was the case.

Unfortunately, shortly after the release and deployment, we began receiving reports of slow downloads, failed downloads, server errors (504 Gateway Timeouts), and corrupted data for the BCCAQv2 downloads.

Progress on identifying the problems was, unfortunately, very slow. Analysis was frustrated by the fact that there ended up being multiple separate issues, none of which had ever been triggered by the many datasets that we have served in the past.

THE BUGS

The first and most severe bug was a mistake in which if the dataportal attempted to serve data from a NetCDF file which had *no* “record variable” (a variable with an unknown and possibly infinite length of record), the code would drop into an infinite loop without emitting any data. This would basically hold the client connection open without returning any data, and also tie up a server worker until the process was restarted. If enough of these requests were made to the server, all workers would be hung on this bug and subsequent requests would return a “504 Gateway Timeout” error until the application was restarted. PCIC fixed this bug and deployed [a release](#) on March 11, 2019.

Two other bugs were identified. Both of these only affected datasets for which a variable in the NetCDF file was [packed](#) into a data type represented by less than 4 bytes.

The first bug miscalculated the padding for packed record variables, thus returning an incorrect Content-length header at the beginning of the server response. The result of this was that the amount of data returned would typically be less (usually only 2 bytes less) than what the server said it should be. Many clients would get to the last two bytes and would simply sit there waiting to receive the last two bytes that never came (making the download appear “slow” or appear to hang). Some clients would close the connection, compare the content-length header with the bytes received and would throw an error, even though the data had actually been received and was in-tact.

The second packing bug was that the data loop provided padding in the wrong place for packed, non-record variables. This made it possible for the data stream to be offset incorrectly in the response and for the data to appear completely unintelligible.

Both of these packing bug would only occur in requests for a single grid cell with an odd number of time steps, so they were particularly rare and very challenging to identify and diagnose.

PCIC released a fix for these bugs on March 15, 2019 and deployed it on March 18, 2019.

Until February 7, 2019, PCIC had not served datasets with packed variables, or datasets with no record variable, hence the fact that we haven’t identified these bugs in the last five years.

Finally, on March 21, 2019 PCIC released and deployed a fix to a problem that affected downloads of time slices that are later in the timeseries. This problem would show up as downloads starting quickly with a few kilobytes of

data and then appear to stall for minutes or hours. Eventually it was possible for the download to timeout intermediate proxies, making the download fail. This problem has been completely fixed, such that this class of downloads should proceed normally.

We sincerely apologize to all of our data portal users for these issues, and hope that it has not been too much of an inconvenience in the weeks that it has taken us to find and correct them. We appreciate your patience, and hope that in the future, the pdp will continue to reliably serve high-quality, credible climate information for many years to come.

THE DETAILS

The [infinite loop bug](#) and the [fix](#).

The [record-length bugs](#) and the [fix](#).

The [late time slice stalling problem](#) and the [fix](#).