ncagg Documentation

Author

Jul 06, 2023

Contents:

So… You want to aggregate NetCDF files? Ncagg is software that enables you to combine multiple input NetCDF files into a single output NetCDF file.

- For interactive use: *The CLI*.
- For programatic use: `ncagg.aggregator.aggregate()`.

Advanced users may want to familiarize themselves with *Templates*.

CHAPTER 1

`ncagg` features

- aggregate files along unlimited dimensions
- insert fill values in gaps
- deduplicate records
- sort a dimension by an indexing variable
- resolve global attributes according to a variety of strategies
- subset variables and dimensions
- create new dimensions variables in each input file variables with no existing unlimited dimension

This software was developed for the situation we faced with GOES-R series satellite data. We receive data in small "granules", where each granule is a NetCDF file containing several seconds or minutes of data. This format is very cumbersome for analysis, requiring thousands of file system operations just to read a single day of data. Using ncagg we can combine these small NetCDF files into a single dayfile for easier use. Often, we also create larger aggregations, a single file containing the entire year, or even mission.

Or, perhaps you have daily model result files over a month, and analysis is going to be easier if you could just operate on a single file.

1.1 The CLI

The Command Line Interface exposes ncagg as a command line program capable of aggregating a list of files to an output file.

The usage message is given below:

1.1.1 Notes

The basic usage is modeled after the `zip` program. The first argument is the output file, while the following specify the contents.

```
ncagg output.nc input1.nc input2.nc
```

If aggregating so many files that the system limits on command length are met, `ncagg` can also take the list of files to aggregate from stdin.

```
find . -type f -name "*.nc" | ncagg output.nc
```

1.1.2 Unlimited Dimension Config

An Unlimited Dimension Config specifies sorting capabilities: Given a dataset with an unlimited dimension called "report_number", and a variable time that is a function of report_number, ie. time(report_number), the following argument would configure `ncagg` to sort report_number according to time when it aggregates.

```
-u report_number:time
```

Furthermore, if one expects apriori for a report_number to occur every second, the frequency can be specified (in units of hz). This configuration will deduplicate overlapping record and insert records is a gap is detected:

```
-u report_number:time:1
```

1.1.3 Aggregation Boundaries

IF an Unlimited Dimension Config is given, boundaries can additionally be imposed in the form min:max:

```
-b 0:10
```

By default, these are expected to in the same units as the UDC ivar, however, when dealing with time, often these values have units inconvenient to humans such as seconds since j2k. For convenience, the bounds can be specified in Tstart:Tstop format where start and stop are YYYY[MM[DD[HH[MM]]]]. If only start is given, stop will be inferred to be one increment of the least significantly specified date component.

The following would create a dayfile for 2018 03 30

```
-b T20180330
```

A half day could be done like this:

```
-b T20180330:T2018033012
```

1.1.4 A note on templates

No template is required to perform aggregation. Template usage is considered advanced. See *Templates* for complete information.

Template syntax is verbose json. It is inconceivable that a human would create a template from scratch. Instead, one should modify a default template. The way to do this using ncagg would be:

```
ncagg --generate_template input1.nc > template.json
```

Then modify the template and perform aggregation usint it:

```
ncagg -t template.json output.nc input1.nc input2.nc
```

1.2 Templates

Coming soon. Content contained in README.md right now.

1.3 ncagg package

1.3.1 Submodules

1.3.2 ncagg.aggregator module

1.3.3 ncagg.aggrelist module

1.3.4 ncagg.attributes module

1.3.5 ncagg.cli module

1.3.6 ncagg.config module

CHAPTER 2

Indices and tables

- genindex
- modindex
- search