Collaborative Climate Community Data and Processing Grid (C3-Grid)

Data Preparation (Preprocessing for Model Data)

Work Package:	WP4 – Data Preparation and Preprocessing
Author:	M. Stockhause
Version:	P01/2008
Publishing Date:	04/2008
Pages:	22
WP Coordination:	M&D / MPI-M
Partners:	MPI-M, IFM-GEOMAR, M&D/MPI-M, FUB, Uni Köln, DLR, AWI, PIK, DWD
Contact:	M. Stockhause
Email:	martina.stockhauseATzmaw.de

GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung

D-GRID

Content

1 Introduction

1.1 Overview

C3Grid defined a general data request functionality with the technical implementation of a webservice (Data Request Webservice) at the data providers. The different principal steps for this so called preprocessing functionality are shown in **Figure 1** [1] [2]. The C3Grid preprocessing functionality includes

- the selection of content parameters (Climate and Forecast, CF standard names) and
- the selection of a part of the total extent (temporal and spatial).

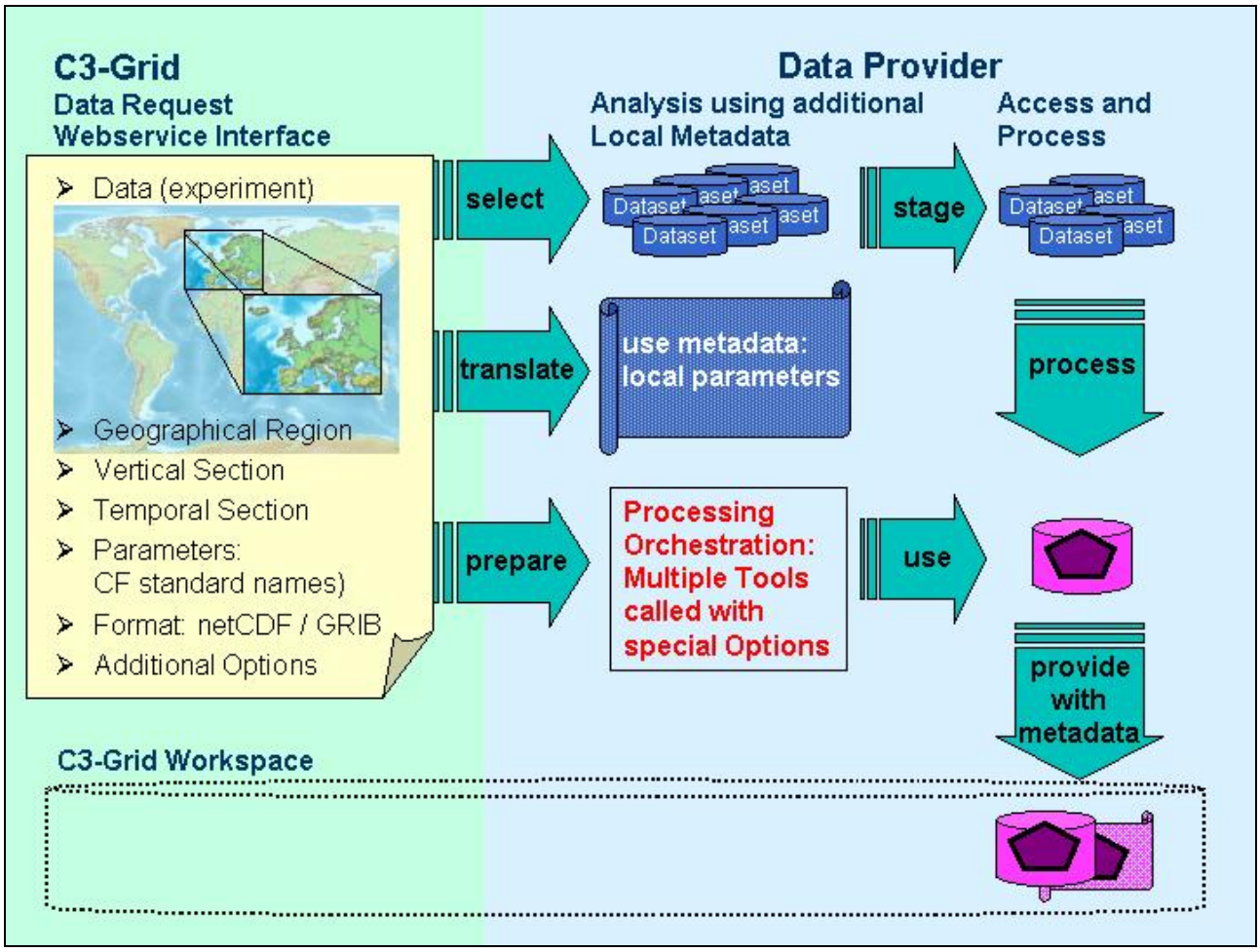

Figure 1: Functionality of data preprocessing.

The functionality for atmospheric data is restructured: preprocessing covers only the functionality of the C3Grid data request (spatial, temporal and parameter selection from an experiment, e.g. model run). Further processing options like spatial or temporal means or other statistical operations are moved to a later (workflow) processing step. Besides, the preprocessing code was re-written in python to ease a later shift of functionality into the web service layer.

Along with the selected data metadata in the C3Grid profile of ISO 19115 / 19139 format is prepared [3].

1.2 Improvements

The preprocessing was improved in the following aspects related to the last Version 0.2:

- The input data for processing is now given by the experiment uid (model run level) and no longer as explicit list of input data sets generated by the portal.
- The preprocessing has become much more generic than the previous version: All additionally needed metadata information for translation and interpretation of the user's request is derived from the metadata in the CERA DB.
- The preprocessing is divided into two separated steps (modularized): staging and preprocessing, which can be performed separately or together driven by the option –p (c.f. section 0). Therefore in a later generation of C3Grid the preprocessing could be scheduled and if needed be performed elsewhere. That could be of advantage e.g. for the IPCC-AR5 data, where a sudden download peak is expected after publication.
- Atmospheric data can be delivered on the original model levels or interpolated on standard pressure levels.
- The file size estimation was improved and adapted to the enlarged preprocessing functionality (data and different vertical scales for atmospheric data).

1.3 Usage of preprocessing

usage: python cera_dkrz.py -i <inifile> -l <logfile> -m <metafile> -p <preproc> -d <dir infiles> -o <dir outfiles> -f <outfile>

-i	name of initialization file in javaproperties format in <dir infiles> for –p proc : tar of initialization file and restart files
-l	name of logfile in <dir outfiles>
-m	name of metadata file describing input data in <dir infiles>
-p	processing option
	stage only staging of files performed
	stproc staging with preprocessing performed
	proc only processing of staged files performed
-d	path to directory of input files
-o	path to directory of output files (if not given in <inifile>, for –p proc)
-f	name of output file (if not given in <inifile>, for –p proc)

Examples:

- **staging and preprocessing:**
  cera_dkrz.py -i *.txt -m *.xml -l *.log -p stproc -d mypath/in -o mypath/out
- **staging:**
  cera_dkrz.py -i *.txt -m *.xml -l *.log -p stage -d mypath/in –o mypath/out
- **preprocessing:**
  cera_dkrz.py -i *.tar -l *.log -m *.xml -p proc -d mypath2/in –o mypath2/out –f *.grb

The script cera_dkrz.py accepts the request, translates the discovery information into use information, stages the needed datasets (gsi-ftp or jblob) and does the processing to provide the user requested data with metadata. The scripts use the agreed environment variables for the workspace [4].

They are accessible through NQSII (Generation 0-Implementation) as well as the SGE.

1.4 Available Data with implemented preprocessing functionality

At the moment the following data types can be handled and preprocessed:

- raw result data of the atmospheric model ECHAM in versions 4 and 5 stored in the DKRZ archive (data of MPI-M and GKSS),
- derived time series of results of ECHAM (2D slices on standard pressure levels for a single variable) stored in CERA DB (WDCC),
- derived multi-level time series of results of MPI-OM (single variable) stored in CERA DB,
- derived time series of HOAPS III satellite data (single variable) stored in CERA, and
- raw result data of the ocean model OPA stored in the DKRZ archive.

Table 1 gives an overview over the processing options for the different data types and the used tools for preprocessing.

Data Type	vertical units	output data formats	staging tool	processing tools	metadata tool
ECHAM raw data	model levels or standard pressure levels	GRIB, netCDF	GridFTP	afterburner, cdo	xmlstarlet
ECHAM time series (CERA)	standard pressure levels	GRIB, netCDF	jblob	cdo	xmlstarlet
MPI-OM time series (CERA)	meter	netCDF	jblob	cdo, nco	xmlstarlet
HOAPS III time series (CERA)	meter	netCDF	jblob	cdo, nco	xmlstarlet
OPA raw data	no vertical selection yet	netCDF	GridFTP	crush	xmlstarlet

Table 1: Data available at DKRZ (CERA DB and archive data of providers MPI-M, IFM-GEOMAR, GKSS) and the possible vertical units of the requested data as well as the tools used for processing.

2 Functionality of the ECHAM data preprocessing

2.1 Functionality

2.1.1 Explicit Functionality:

- Cut parts out of datasets belonging to an experiment (model run) according to the user's request: longitude, latitude, altitude, time and CFs
- Merging to form a single output file (name as defined by the user's request)
- Conversion in netCDF if requested (original format GRIB)
- **Changed:** Additional processing options like temporal or spatial means (former option 'monthly_mean'), other statistical indices or transformations are shifted to a following processing (outside preprocessing and therefore scheduled by the C3Grid middleware).

2.1.2 Implicit Functionality:

Changes for raw data: For CERA DB time series the data is already gridded and interpolated on standard pressure levels and therefore available only defined on this grid. For raw data the user can now alternatively select the data

1. on the coordinate system of the model run (vertical on hybrid model levels) and as ungridded spectral variables (therefore then no regional selection is possible) or
2. interpolated on the longitude-latitude-standard pressure level grid.

- **Changed:** Conversion of spectral variables on Gaussian Grid for regional selection in standard pressure level case; for hybrid model level case no conversion of spectral variables and therefore no regional selection.
- **Changed:** Depending on user's selection the data is prepared on the original hybrid model levels or on standard pressure levels in the selected altitude interval.
- **Changed:** No Diagnosis of further CF variables by the afterburner (ECHAM postprocessing tool) in this generic form without additional service metadata possible at the moment! These codes are:

code	CF standard name
34	
35	
36	
131	eastward_wind
132	northward_wind
135	lagrangian_tendency_of_air_pressure
148	

149	
151	air_pressure_at_sea_level
156	geopotential_height
191	
192	
259	
189	
190	
260	precipitation_flux
261	
262	

2.1.3 Further Functionality Restrictions:

- Output file smaller than 100 GB (ds7: 30 GB due to file system restrictions)
- At the moment no unit conversions are performed from local units to the units recommended by the CF group. **New:** Documentation of both units in log file.

2.2 Applied Tools

cdo climate data operators	http://www.mpimet.mpg.de/~cdo	Central Postprocessing Operators, esp. for GRIB	Diagnosis
jblob	http://cera-www.dkrz.de/CERA/jblob	CERA-DB-Access (Java Application)	Data Access
afterburner	http://www.mpimet.mpg.de/~afterburner	Basic ECHAM-Rawdata Postprocessing Program	Diagnosis

2.3 Call

Called by main python script cera_dkrz.py.

2.4 Input

2.4.1. Webservice Interface to DMS (Interface D):

The input parameters of the user's request are read from the initialization file in javaproperties format [1]. An example can be found in Appendices A.1 ECHAM Archive Data Example and A.2 ECHAM CERA DB Example.
Changed: It is no longer possible to choose all codes, even those for which no CF standard name is known, by writing as single CFItem in the CFList the name "ALL" (all in capital letters). On the one hand for most of the codes appropriate CF standard names have been found or registered as new CF standard name and on the other hand is this more likely a feature of the portal ('select all' option) then deep down in the preprocessing.

2.4.2 Interface to Portal (Interface G):

The in the portal selected parameters for this workflow can be found in the examples in the

appendix. Later there will be a definition in a workflow information system (WFIS).
In this service parameters out of the logical ISO-Metadata for the Model Run and the to be defined workflow specification *(Preconditions)* will be used to form the user's request for the scheduling in WSL, which delivers the values of interface D at the data provider site.

Changed: The more generic preprocessing requires less data from the portal as well as the change from a delivered dataset list to the experiment uid (model run identifier). Thus searched and used for the composition of the user's request are only information from the parent level (model run level) and no longer from the child level (dataset level).

I. Preconditions:
a) Format Input/Output File (**Portal/User Selection**):
Input File: *GRIB* (ISO: <resourceFormat>)
Output File: *GRIB or netCDF* (interface D: TargetFileFormat with possible values "nc" or "grb")
b) **Data (ISO)**:
ISO: <fileIdentifier> to Interface D as <ObjectList>

II. Cut/Selection:
a) CF Standard names (**ISO + Portal/User Selection**):
all existing in the metadata (ISO: <contentInfo> <attributeDescription> for all data for the search the ISO (model run level) is used **(later:** on dataset or child level only filled if differing from model run level, i.e. for raw data not filled**)**.
b) Temporal (**ISO + Portal/User Selection**):
Temporal part of the whole temporal extent given in the metadata (ISO: <extent> <temporalElement> in Interface D as minTime maxTime in ISO-Date Format: YYYY-MM-DDThh:mm)
c) Geographical (**ISO + Portal/User Selection**):
Lon/Lat-Parts in the whole geographical extent given in the metadata (ISO: <extent> <geographicElement>)
d) Vertical (**ISO + Portal/User Selection**):
Alt-Parts in the whole vertical extent given in the metadata with unit hPa OR '' [Level] (ISO: <extent> <verticalElement> with unit under <CRS> to Interface D: Unit:hPa and MinName:surface or Min and MaxName:surface or Max OR Unit:'' and Min and Max)
The results are prepared on standard pressure levels or hybrid model levels in the vertical interval selected by the user.

III. Output Name (**Portal/User Selection**):
User selection (to Interface D as TargetBaseDataFile.grb or *.nc and TargetMetaDataFile.xml)

2.4.3 Metadata:

- For the interpretation and translation of the user's request into useable information for the preprocessing additional metadata from the CERA DB is accessed by servlets.
- For the preparation of the appropriate ISO metadata for the prepared dataset the ISO metadata file for the parent (model run) level is used as reference file. Different xslt scripts called by xmlstarlet do the preprocessing dependent alternations and additions.

2.5 Output

The output files are prepared in the workspace of the DKRZ, i.e. under $C3GRID_WORKSPACE. At the moment the following files are written:

- **Data:** as <TargetBaseDataFile> in the specified <TargetFileFormat> in the user data request, e.g. MyData.grb oder MyData.nc
- **Metadata:** as <TargetMetaDataFile> in ISO-Format, e.g. MyData.xml
- **Logfile:** documentation of the script run for the user's information
- **Error:** as <TargetBaseDataFile>.err (since G0 written in **Logfile**), e.g. MyData.err
- **Warning:** as <TargetBaseDataFile>.wrn (since G0 written in **Logfile**), e.g. MyData.wrn

2.6 Examples

Example datasets are stored under cross.dkrz.de/$C3GRID_ROOT/outdata/examples and interface D examples for data requests can be found in A.1 ECHAM Archive Data Example and A.2 ECHAM CERA DB Example.
The used scripts in the current version are stored under $C3GRID_ROOT/scripts/gen1+.

2.7 Problems / ToDo

- Definition of a workflow template as first step for a WFIS.
- Solution of the problem of huge file sizes for preprocessed files (DKRZ Filesystem-User)
- Communication problems for metadata and data download from CERA DB by servlets and jblob and file transfer by gridFTP out of tape archive to be solved by DKRZ.

3 Functionality CERA data preprocessing

3.1 Functionality

3.1.1 Explicit Functionality:

- Cut parts out of datasets belonging to an experiment (MPI-OM model run) or a temporal resolution of HOAPS satellite data according to the user's request: longitude, latitude, depth (for MPI-OM), time and CFs
- Merging to form a single output file in netCDF format (name as defined by the user's request)

3.1.2 Implicit Functionality:

None

3.1.3 Further Functionality Restrictions:

- Output file smaller than 100 GB (ds7: 30 GB due to file system restrictions)
- At the moment no unit conversions are performed from local units to the units recommended by the CF group. **New:** Documentation of both units in log file.

3.2 Applied Tools

nco	netCDF operators	http://nco.sourceforge.net/	Operators for netCDF files	Diagnosis
cdo	climate data operators	http://www.mpimet.mpg.de/~cdo	Central Postprocessing Operators	Diagnosis
jblob		http://cera-www.dkrz.de/CERA/jblob	CERA-DB-Access (Java Application)	Data Access

3.3 Call

Called by main python script cera_dkrz.py.

3.4 Input

3.4.1. Webservice Interface to DMS (Interface D):

The input parameters of the user's request are read from the initialization file in javaproperties format [1]. Examples can be found in Appendix A.3 MPI-OM CERA DB Example and A.4 HOAPS III CERA DB Example.

3.4.2 Interface to Portal (Interface G):

The in the portal selected parameters for this workflow can be found in the examples in the appendix. Later there will be a definition in a workflow information system (WFIS). In this service parameters out of the logical ISO-Metadata for the Model Run and the to be defined workflow specification *(Preconditions)* will be used to form the user's request for the scheduling in WSL, which delivers the values of interface D at the data provider site.
The search and the composition of the user's request in the portal are performed with the parent or model run level of the ISO metadata.

I. Preconditions:
a) Format Input/Output File (**Portal/User Selection**):
Input File: *netCDF* (ISO: <resourceFormat>)
Output File: *netCDF* (interface D: TargetFileFormat with possible values "nc")
b) **Data (ISO)**:
ISO: <fileIdentifier> to Interface D as <ObjectList>

II. Cut/Selection:
a) CF Standard names (**ISO + Portal/User Selection**):
all existing in the metadata (ISO: <contentInfo> <attributeDescription>)
b) Temporal (**ISO + Portal/User Selection**):

Temporal part of the whole temporal extent given in the metadata (ISO: <extent> <temporalElement> in Interface D as minTime maxTime in ISO-Date Format: YYYY-MM-DDThh:mm)
c) Geographical (**ISO + Portal/User Selection**):
Lon/Lat-Parts in the whole geographical extent given in the metadata (ISO: <extent> <geographicElement>)
d) Vertical (**ISO + Portal/User Selection**):
Alt-Parts in the whole vertical extent given in the metadata with unit m (ISO: <extent> <verticalElement> with unit under <CRS> to Interface D: Unit:m and Min and Max)
(note: HOAPS III data is 2D)

III. Output Name (**Portal/User Selection**):
User selection (to Interface D as TargetBaseDataFile.nc and TargetMetaDataFile.xml)

3.4.3 Metadata:

- For the interpretation and translation of the user's request into useable information for the preprocessing additional metadata from the CERA DB is accessed by servlets.
- For the preparation of the appropriate ISO metadata for the prepared dataset the ISO metadata file for the parent (model run) level is used as reference file. Different xslt scripts called by xmlstarlet do the preprocessing dependent alternations and additions.

3.5 Output

The output files are prepared in the workspace of the DKRZ, i.e. under $C3GRID_WORKSPACE. At the moment the following files are written:

- **Data:** as <TargetBaseDataFile> in the specified <TargetFileFormat> in the user data request, e.g. MyData.grb oder MyData.nc
- **Metadata:** as <TargetMetaDataFile> in ISO-Format, e.g. MyData.xml
- **Logfile:** documentation of the script run for the user's information
- **Error:** as <TargetBaseDataFile>.err (since G0 written in **Logfile**), e.g. MyData.err
- **Warning:** as <TargetBaseDataFile>.wrn (since G0 written in **Logfile**), e.g. MyData.wrn

3.6 Examples

Example datasets are stored under cross.dkrz.de/$C3GRID_ROOT/outdata/examples and interface D examples for data requests can be found in A.3 MPI-OM CERA DB Example and A.4 HOAPS III CERA DB Example.
The used scripts in the current version are stored under $C3GRID_ROOT/scripts/gen1+.

3.7 Problems / ToDo

- Definition of a workflow template as first step for a WFIS.
- Solution of the problem of huge file sizes for preprocessed files (DKRZ Filesystem-User)

- Communication problems for metadata and data download from CERA DB by servlets and jblob to be solved by DKRZ.

4 Functionality of the OPA data preprocessing

4.1 Functionality

The principle procedure after the analysis of the user's request consists of an initialization and an analysis call of the program "crush". In the initialization a so called namelist is build, which contolls the following analysis. The output data is prepared in four different output files, one for each of the ARACAWA C grids. Finally, these output files are tar-ed before they are prepared in the workspace.

4.1.1 Explicit Functionality:

- Cut parts out of datasets belonging to an experiment (model run) according to the user's request: longitude, latitude, time and CFs (no vertical selection, yet)
- Create a tar file for the one up to four output files (ARACAWA C-grids): The name of tar file is given by the user's request.
- Momentarily, only processing option –timeseries out of the crush program package is implemented, calculating time series out of raw data.

4.1.2 Implicit Functionality:

None.

4.1.3 Further Functionality Restrictions:

- Output file smaller than 100 GB (ds7: 30 GB due to file system restrictions)
- At the moment no unit conversions are performed from local units to the units recommended by the CF group, e.g. for the temperature with its local unit °C into the CF unit K. **New:** Documentation of both units in log file.
- Momentarily, no vertical selection possible.
- Only nc output format available: No format conversion from nc to grb implemented, because this format is not used (very uncommon) in the OPA community. Additionally, there is an information loss in this format conversion.

4.2 Applied Tool

crush	Internal Wiki IFM-GEOMAR; availability on request from A.Biastoch.	Special program package for NEMO (OPA+icemod) model used at IFM-GEOMAR	Diagnosis

4.3 Call

Called by main python script cera_dkrz.py.

4.4 Input

4.4.1. Webinterface to DMS (Interface D):

The input parameters of the user's request are read from the initialization file in javaproperties format [1]. An example can be found in Appendix A.5 Nemo-Crush Example.

4.4.2 Interface to Portal (Interface G):

The in the portal selected parameters for this workflow can be found in the examples in the appendix. Later there will be a definition in a workflow information system (WFIS).
In this service parameters out of the logical ISO-Metadata for the Model Run and the to be defined workflow specification *(Preconditions)* will be used to form the user request for the scheduling in WSL, which delivers the values of interface D at the data provider site.

I. Preconditions:
a) Format Input/Output File (**Portal/User Selection**):
Input File: *netCDF*
Output File: *netCDF*
b) **Data (ISO)**:
ISO: <fileIdentifier> to Interface D as <ObjectList>

II. Cut / Selection:
a) CF Standard names (**ISO + Portal/User Selection**):
all that are defined in the metadata
b) Temporal (**ISO + Portal/User Selection**):
Temporal part of the whole temporal extent defined in the metadata
c) Geographical (**ISO + Portal/User Selection**):
Lon/Lat- parts of the whole geographical extent defined in the metadata
d) Vertical (**ISO + Portal/User Selection**):
Momentarily, **not** selectable; later with altitude unit AltUnit = m.

III. Output Name (**Portal/User Selection**):
User defined, but tar specification in <TargetFileFormat>tar

4.4.3. Metadata:

- For the interpretation and translation of the user's request into useable information for the preprocessing additional metadata from the CERA DB is accessed by servlets.
- For the preparation of the appropriate ISO metadata for the prepared dataset the ISO metadata file for the parent (model run) level is used as reference file. Different xslt scripts called by xmlstarlet do the preprocessing dependent alternations and additions.

4.5 Output

The output files are prepared in the workspace of the DKRZ, i.e. under $C3GRID_WORKSPACE. At the moment the following files are written:

- **Data:** as <TargetBaseDataFile> tar file with tared datasets in netCDF format, e.g. MyData.tar.
- **Metadata:** as <TargetMetaDataFile> in ISO-Format, e.g. MyData.xml
- **Logfile:** $LOGFILE of the script run for the user information
- **Error:** as <TargetBaseDataFile>.err (since G0 written in **Logfile**), e.g. MyData.err
- **Warning:** as <TargetBaseDataFile>.wrn (since G0 written in **Logfile**), e.g. MyData.wrn

4.6 Examples

Example datasets are stored under cross.dkrz.de/$C3GRID_ROOT/outdata/examples and interface D examples for data requests can be found in A.5 Nemo-Crush Example.
The used scripts in the current version are stored under $C3GRID_ROOT/scripts/gen1+.

4.7 Problems / ToDo

- Definition of a workflow template as first step for a WFIS.
- Solution of the problem of huge file sizes for preprocessed files (DKRZ Filesystem-User)
- Communication problems for metadata and data download from CERA DB by servlets and gridFTP transfers from tape archive to be solved by DKRZ.
- Enlargement of the functionality:
    - Additional implementation of diagnostic crush routines
    - Vertical part selection (on model levels?)

5 File Size

For the scheduling estimations of the preprocessing time and the size of the preprocessed file that has to be transferred are important in order to plan and optimize the execution of the workflows (for the estimation of preprocessing time cf. Chapter 0.

Because of the complexity of the data preprocessing (different formats and conversion are performed), the derivation of the file size isn't trivial. A coarse guess like adding the file sizes of the needed (staged datasets) isn't of any help, because it commonly overestimates the actual file size by orders. Therefore an analytic approach is used to calculate the file size in GRIB format. This depends on the model resolution, the number and type (spherical harmonics or gridded; stored in 2 or 3 byte precision) of the requested content parameters, and the user selected spatial-temporal extent of the data. For netCDF chosen as output file format an additional factor of two is included. **Changed:** There is no longer a dependency on an advanced processing option to be considered (e.g. temporal mean).
Thus, the file size is dependent on the source data and the user's data request.

File size on basis of GRIB Files:

FileSize = f_{format} * n_t * ($\sum_{ct}$(n_c*($Size_{Grid}$, n_L, prec) * $Size_{Grid}$ * n_L * prec + header))

It is the sum of the sizes of different 2- and 3-dimensional arrays of code types (ct) times the number of time steps n_t and a format factor for netCDF (f_{format}(grb)=1, f_{format}(netCDF)=2).

These code type arrays for a time step exist in different combinations of:

- **Size_grid:** horizontal grid (spectral/Gaussian) with Size_grid calcutated from the model resolution (e.g. T63N48L31 refering to the ECMWF definition) and the geographical cutting options (minLon, maxLon, minLat, maxLat):
  global: $Size_{Grid}$(spectral) = (T+1)*(T+2)
  global: $Size_{Grid}$(gaussian) = 2*4*N^2
  regional: $Size_{Grid}$(gaussian) = ((maxLon-minLon)*(max_Lat-minLat)/(180*360)*2*4*N^2

- **n_l:** number of vertical levels, where some codes are 2 (n_L=1) and other 3 dimensional (e.g. n_L=31). The user selection is of relevance, because his/her vertical selection defines the number of model levels or pressure levels / depth levels in m in this height interval. For the provision of atmospheric raw data on standard pressure levels an additional postprocessing is done, including a coordinate transformation from the hybrid model levels to the standard pressure levels with n_L=17 for the 3 dimensional codes.

- **prec:** precision of the content parameter representation (2 oder 3 byte). The difference in the derived file size estimation isn't that large. Moreover this precision information isn't available in CERA metadata. So, this factor is neglected and a 2 byte precision assumed for all codes.

Note: The soil temperature, which is defined on five depth levels, is neglected in G0 and G1 as well, because it is not selectable in the portal by the vertical unit options 'hPa' and '' [levels] for the parent or model level.

Further dependencies:

- **n_t:** number of output times, derivable out of temporal resolution of source data (e.g. dt=6h or 1 month) and total modeled time period or user's selection, respectively: n_t = (maxTime-minTime)/dt. **Changed:** There is no longer a processing dependency existent for the temporal mean option, i.e. for monthly_mean is dt=1 month.

- **f_{format}:** In the currently used GRIB version 1 (grb) and netCDF version 2 (nc) at WDC Climate and DKRZ archives, the conversion from GRIB into netCDF adds a factor of roughly 2. Therefore it is set: f_{format}(nc)=2 with f_{format}(grb)=1.

Constant:

- **header:** header of a 2D section in GRIB file, approximately 88 byte for gridded and 344 for spherical harmonics layers.

Limitations:

- Estimations for **raw data (ocean)** are imprecise because of irregular grid structures; quite large errors for region selection might occur: **Under-** or **over**estimation in size dependent on selected region.
- **3 byte accuracy** ignored and therefore these codes are **under**estimated in size
- An ordinary calendar is used for file size estimation, therefore for **synthetic calendars** (e.g. 360 days-a-year) the number of time points and the file size are slightly **over**estimated.
- If the user selects **parts of days as start or end time**, a slight **over**estimation is caused (estimation is based on whole days)

6 Processing Times

There are several dependencies including the sizes of the staged datasets, the processing complexity (user's data request) and on the working load of the local system (tape robot usage, disk and processor memory utilization). There are no ready-to-use approaches for such an estimation, therefore the derived estimation will be imprecise at first and has to be improved step-by-step using logging information about the estimated time and the needed time as well as the contributions to them.

For generation 1 a first estimation for the needed staging time was developed. The calculation involves empirical coefficients because of a lacking connection between tape waiting times to jobs.

Additive contributions to the staging time have:

1. tape waiting time (if files are on tape): basic time for loading and system depending access time
2. internal transfer time via gridFTP to C3Grid scratch directory
3. preprocessing time

After checking, if the requested files are cached (all files stored in CERA are assumed as on tape), for non-cached files the mean tape waiting time for files on tape over the last two hours is analysed and assumed for the current waiting time. It is guessed that the files belonging to the same model run are stored on the same tape, so that the tape waiting time is added only once. The time for tape load is set to one minute.

In the second step the time for internal file transfers is added. At the moment the maximum data transfer rate at Dkrz is 100 MBit/s for all transfer. Taking into account the number of actual file transfer and the size of files to be staged for the request, a transfer time is derived.

The third part, the time for preprocessing the staged files is neglected in generation 1.

Later it could be estimated using the size of the staged files and perhaps one or two additional parameters.

Note that for HOAPS data currently no staging time estimation is available due to incomplete metadata information in CERA DB.

7 Interaction with the grid

7.1 Status of discussion for the Data Request Interface

- Only discovery metadata needed from portal in user's request: All translations of the user's request into local parameters are now done using additional metadata from CERA.
- Altitude: The portal supports two vertical scales: m and hPa, not yet the original model levels for a data request. For the surface value of the hPa scale the value "99999" is used.
- Precision of parameter values in data request: two decimals; times as full ISO DateTime format.

7.2 Open Issues / ToDos

- Improvement of staging time estimation
- Use of cached files instead of files in CERA DB or in DKRZ archive (probably on tape)
- Precision of conventions for tool and workflow installations on sites (with modules package)
- Possibility of scheduler job changes by data providers: submission of preprocessing tasks to be discussed.
- WFIS: 1. Development of a C3Grid profile of a service metadata schema to describe workflow components (ISO 19119 ?), 2. Development and set-up of a workflow information service (WFIS) to connect data with workflows
- User specific workflows and data: How can they be integrated in the grid: metadata generation (tool)? Workflow description / quality?

8 References

[1] T. Langhammer, F. Schintke:
T5.1 (Supplement A):Interface Specification for C3-Grid Archives
http://www.c3grid.de/documents/interface_archive.pdf

[2] S. Kindermann, M. Stockhause, and K. Ronneberger:
Intelligent Data Networking for the Earth System Science Community;
German e-science conference 02.-04.05.2007 in Baden-Baden;
http://edoc.mpg.de/316512

[3] S. Kindermann:
C3Grid ISO Metadata Profile
http://www.c3grid.de/documents/c3grid_metadata.pdf
http://www.c3grid.de/documents/C3Datenkatalog-V_1_0.pdf
http://www.c3grid.de/documents/xml-schema_G0-C3GRID.xml

[4] T. Langhammer, V. Achter, and V. Winkelmann:
Richtlinien für die Konfiguration des Workspaces [German]
http://www.c3grid.de/documents/Konfiguration_Workspace_deliverable.pdf

A Appendix: Examples

A.1 ECHAM Archive Data Example

```
#
# Selection of vertical heights on hybrid model levels
#
c3grid.StageFileRequest.TargetBaseDataFile=gsiftp\://cross.dkrz.de/prj/bb0300/work/test_ms/out/dkrz_v06_DT_L.grb
c3grid.StageFileRequest.TimeConstr.MaxTime=2003-02-28T23\:00\:41.184Z
c3grid.StageFileRequest.SpaceConstr.Latitude.Min=-30.0
c3grid.StageFileRequest.SpaceConstr.Latitude.Max=30.0
c3grid.StageFileRequest.ObjectList.Item.Count=0
c3grid.StageFileRequest.ConstraintList=
c3grid.StageFileRequest.TimeConstr.MinTime=2002-11-01T00\:00\:41.184Z
c3grid.StageFileRequest.TargetFileFormat=grb
c3grid.StageFileRequest.CFList.CFItem=eastward_wind-at10m northward_wind-at10m air_temperature geopotential_height
c3grid.StageFileRequest.SpaceConstr.Longitude.Max=360.0
c3grid.StageFileRequest.SpaceConstr.Altitude.Max=20
c3grid.StageFileRequest.JustAsk=false
c3grid.StageFileRequest.SpaceConstr.Altitude.Min=8
c3grid.StageFileRequest.ObjectList.Item.0=de.dkrz.wdcc.iso2093450
c3grid.StageFileRequest.TargetMetaDataFile=gsiftp\://cross.dkrz.de/prj/bb0300/work/test_ms/out/dkrz_v06_DT_L.xml
c3grid.StageFileRequest.SpaceConstr.Longitude.Min=0.0
c3grid.StageFileRequest.SpaceConstr.Altitude.Unit=
c3grid.StageFileRequest.ConstraintList.ExpId=de.dkrz.wdcc.iso2093450
```

A.2 ECHAM CERA DB Example

```
#
#
c3grid.StageFileRequest.TimeConstr.MaxTime=2100-12-21T08\:00\:41.184Z
c3grid.StageFileRequest.TargetBaseDataFile=gsiftp\://cross.dkrz.de/prj/bb0300/work/test_ms/out/cera_v06_atmbot2.grb
c3grid.StageFileRequest.TargetFileFormat=grb
c3grid.StageFileRequest.CFList.CFItem=eastward_wind northward_wind air_temperature geopotential_height
relative_humidity specific_humidity atmosphere_cloud_ice_content air_pressure_at_sea_level surface_temperature
surface_downward_eastward_stress atmosphere_cloud_condensed_water_content atmosphere_water_vapor_content
cloud_area_fraction
c3grid.StageFileRequest.SpaceConstr.Latitude.Min=-30.0
c3grid.StageFileRequest.SpaceConstr.Altitude.MaxName=surface
c3grid.StageFileRequest.SpaceConstr.Latitude.Max=30.0
c3grid.StageFileRequest.ObjectList.Item.Count=0
c3grid.StageFileRequest.TimeConstr.MinTime=2098-01-10T15\:00\:41.184Z
c3grid.StageFileRequest.SpaceConstr.Altitude.Unit=hPa
c3grid.StageFileRequest.JustAsk=false
c3grid.StageFileRequest.ObjectList.Item.0=de.dkrz.wdcc.iso2035543
c3grid.StageFileRequest.TargetMetaDataFile=gsiftp\://cross.dkrz.de/prj/bb0300/work/test_ms/out/cera_v06_atmbot2.xml
c3grid.StageFileRequest.ConstraintList=
c3grid.StageFileRequest.SpaceConstr.Altitude.MaxName=surface
c3grid.StageFileRequest.SpaceConstr.Altitude.Min=10.0
c3grid.StageFileRequest.SpaceConstr.Longitude.Min=0.0
```

A.3 MPI-OM CERA DB Example

```
#
#
c3grid.StageFileRequest.TargetMetaDataFile=gsiftp\://cross.dkrz.de/prj/bb0300/work/test_ms/out/cera_v06_ozeanV3.xml
c3grid.StageFileRequest.TimeConstr.MaxTime=2010-12-31T23\:59\:41.184Z
c3grid.StageFileRequest.SpaceConstr.Latitude.Min=-90.0
c3grid.StageFileRequest.SpaceConstr.Latitude.Max=90.0
c3grid.StageFileRequest.ObjectList.Item.Count=0
c3grid.StageFileRequest.ConstraintList=
c3grid.StageFileRequest.TimeConstr.MinTime=2010-01-01T00\:00\:41.184Z
c3grid.StageFileRequest.TargetFileFormat=nc
c3grid.StageFileRequest.CFList.CFItem=eastward_sea_ice_velocity eastward_sea_water_velocity
northward_sea_ice_velocity northward_sea_water_velocity sea_ice_area_fraction sea_ice_thickness
sea_surface_height_above_geoid sea_surface_temperature sea_water_potential_temperature sea_water_salinity
upward_sea_water_velocity water_flux_into_ocean
c3grid.StageFileRequest.SpaceConstr.Longitude.Max=360.0
c3grid.StageFileRequest.SpaceConstr.Altitude.Max=-200.0
c3grid.StageFileRequest.JustAsk=false
c3grid.StageFileRequest.SpaceConstr.Altitude.Min=-3000.0
c3grid.StageFileRequest.ObjectList.Item.0=de.dkrz.wdcc.iso2084050
c3grid.StageFileRequest.TargetBaseDataFile=gsiftp\://cross.dkrz.de/prj/bb0300/work/test_ms/out/cera_v06_ozeanV3.nc
c3grid.StageFileRequest.SpaceConstr.Longitude.Min=0.0
```

A.4 HOAPS III CERA DB Example

```
#
#
c3grid.StageFileRequest.TargetMetaDataFile=gsiftp\://cross.dkrz.de/prj/bb0300/work/test_ms/out/cera_v06_hoapsD.xml
c3grid.StageFileRequest.TimeConstr.MaxTime=2003-12-31T23\:59\:41.184Z
c3grid.StageFileRequest.SpaceConstr.Latitude.Min=-80.0
c3grid.StageFileRequest.SpaceConstr.Latitude.Max=80.0
c3grid.StageFileRequest.ObjectList.Item.Count=0
c3grid.StageFileRequest.ConstraintList=
c3grid.StageFileRequest.TimeConstr.MinTime=2002-06-15T00\:00\:41.184Z
c3grid.StageFileRequest.TargetFileFormat=nc
c3grid.StageFileRequest.CFList.CFItem=lwe_water_evaporation_rate sea_surface_temperature surface_upward_water_flux
surface_net_upward_longwave_flux atmosphere_cloud_liquid_water_content
c3grid.StageFileRequest.SpaceConstr.Longitude.Max=10.0
c3grid.StageFileRequest.SpaceConstr.Altitude.Max=0.0
c3grid.StageFileRequest.JustAsk=false
c3grid.StageFileRequest.SpaceConstr.Altitude.Min=0.0
c3grid.StageFileRequest.ObjectList.Item.0=de.dkrz.wdcc.iso2143469
c3grid.StageFileRequest.TargetBaseDataFile=gsiftp\://cross.dkrz.de/prj/bb0300/work/test_ms/out/cera_v06_hoapsD.tar
c3grid.StageFileRequest.SpaceConstr.Longitude.Min=-180.0
c3grid.StageFileRequest.SpaceConstr.Altitude.Unit=m
```

A.5 Nemo-Crush Example

```
#
#
c3grid.StageFileRequest.TimeConstr.MaxTime=2000-12-31T23\:00\:41.184Z
c3grid.StageFileRequest.TargetBaseDataFile=gsiftp\://cross.dkrz.de/prj/bb0300/work/test_ms/out/nemo_v02_t2.tar
c3grid.StageFileRequest.TargetFileFormat=nc
c3grid.StageFileRequest.CFList.CFItem=sea_water_potential_temperature sea_surface_temperature
sea_surface_height_above_sea_level runoff_flux downward_heat_flux_in_air ocean_mixed_layer_thickness
sea_water_x_velocity surface_downward_x_stress sea_water_y_velocity surface_downward_y_stress
upward_sea_water_velocity
c3grid.StageFileRequest.SpaceConstr.Latitude.Min=-30.0
c3grid.StageFileRequest.SpaceConstr.Latitude.Max=30.0
c3grid.StageFileRequest.ObjectList.Item.Count=0
c3grid.StageFileRequest.TimeConstr.MinTime=1998-01-01T00\:00\:41.184Z
c3grid.StageFileRequest.SpaceConstr.Altitude.Unit=m
c3grid.StageFileRequest.JustAsk=false
c3grid.StageFileRequest.ObjectList.Item.0=de.dkrz.wdcc.iso2101677
c3grid.StageFileRequest.TargetMetaDataFile=gsiftp\://cross.dkrz.de/prj/bb0300/work/test_ms/out/nemo_v02_t2.xml
c3grid.StageFileRequest.ConstraintList=
c3grid.StageFileRequest.SpaceConstr.Longitude.Max=70.0
c3grid.StageFileRequest.SpaceConstr.Altitude.Max=6000.0
c3grid.StageFileRequest.SpaceConstr.Altitude.Min=0.0
c3grid.StageFileRequest.SpaceConstr.Longitude.Min=-10.0
```

With:

- ObjectList.Item: uid of parent or model run level
- CFList.CFItem: List of CF standard names
- TimeConstr.MinTime: min_Time (YYYY-MM-DDTHH:MM:SS.SSSZ)
- TimeConstr.MaxTime: max_Time (YYYY-MM-DDTHH:MM:SS.SSSZ)
- SpaceConstr.Longitude.Min and …Max: min_Lon max_Lon (Longitudes, defined from 0 to 360 for ECHAM and MPI-OM, from -180 to 180 for OPA)
- SpaceConstr.Latitude.Min and …Max: min_Lat max_Lat (Latitudes, defined from -90 to +90)
- SpaceConstr.Altitude.Min and …Max: min_Alt max_Alt (pressure levels in hPa or dimensionless hybrid model levels or depth in m depending on SpaceConstr.Altitude.Unit)
- SpaceConstr.Altitude.MinName and …MaxName: min_Alt max_Alt (surface value as string "surface" for SpaceConstr.Altitude.Unit=hPa)
- SpaceConstr.Altitude.Unit: Unit of vertical scale (possible values: ‘’ for hybrid model levels – ECHAM raw data, ‘hPa’ for standard pressure levels – ECHAM raw and DB data, or ‘m’ for depth – ocean models raw data)
- TargetMetaDataFile: path to Metadata File (*.xml)
- TargetBaseDataFile: path to Output File (*.grb or *.nc or *.tar)
- TargetFileFormat: Data Format selected by the user (“nc” and for atmospheric data additionally “grb”)