On-the-fly generated HATS products

F.-X. Pineau[1], P. Fernique[1], G. Landais[1], M. Marchand[1]
and the LSDB/HATS team[2]

francois-xavier.pineau@astro.unistra.fr

[1] CDS ObAS, Strasbourg

[2] University of Washington, Seattle; Carnegie Mellon University, Pittsburgh

Abstract

We present the test developments conducted by CDS for the on-the-fly generation of catalogs in HATS format.

HATS is a framework described in the recently published IVOA note ”HATS: A Standard for the Hierarchical Adaptive Tiling Scheme in the Virtual Observatory.” Developed as part of the LINCC project for Vera C. Rubin data, and partly inspired by the HIPS format, it basically resorts to HEALPix tessellated Parquet datasets to distribute and manipulate possibly very large tabular data.

We present a prototype CDS set up to distribute VizieR large catalogues as on-the-fly generated HATS products, and discuss the advantages and drawbacks of such an approach. A similar solution can be adopted for all HEALPix indexed catalogues, benefiting from HATS aware tools without having to duplicate the data.

CDS large tables and QATSS

- At CDS, each large table is stored as a single *HEALPix sorted and indexed* binary file. Examples and sizes of such tables :
  - US Naval Observatory UCAC5: 8.9 GB
  - ICCUB/AIP StarHorse 2021: 62 GB
  - ESO VHS DR5: 708 GB
  - ESA Gaia DR3 main table: 1.5 TB
- Queries on those tables are performed using the *qat2s* tool presented at ADASS 2023 (*QATSS* stands for *Query Astronomical Table Serialization System*) :
  - HEALPix clustering and indexation allows for fast positional queries;
  - flat files plus a single executable are easier to use and to mirror than using a DBMS;
  - *qat2s* supports multi-threading and multiple input and output file formats.

ADASS 2023

The HATS format

HATS is basically a way of storing an astronomical table as a Parquet dataset, with a structure that follows an HEALPix adaptive grid. It is:

- being developed in the LINCC framework;
- described in a recent IVOA note;
- supported by the scalable large catalogs analysis python tool LSDB.

HATS Repo

LSDB Repo

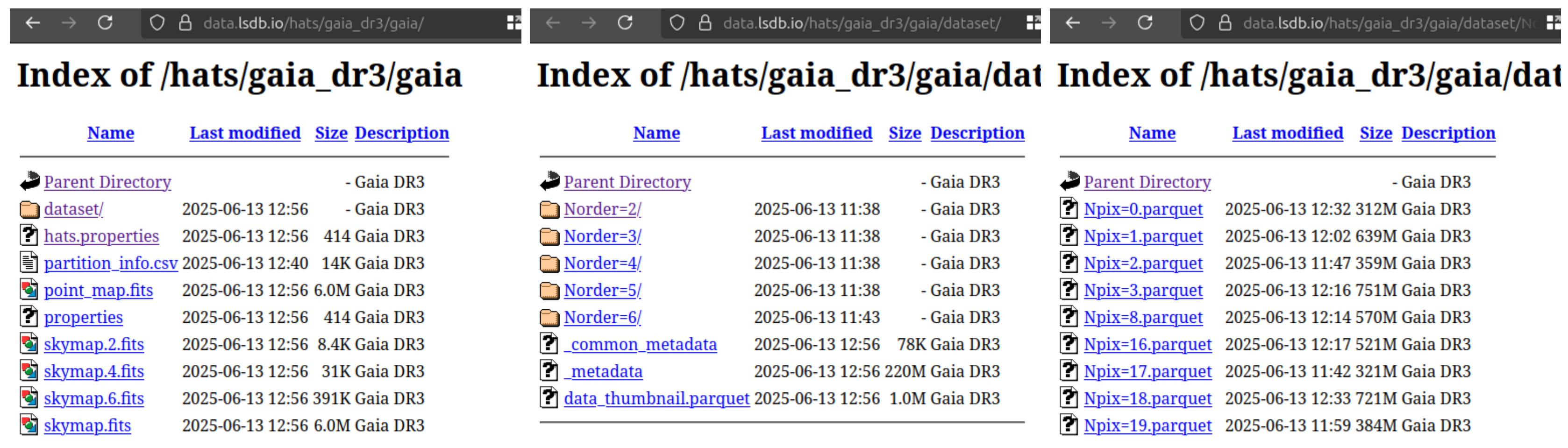
IVOA Note

International Virtual Observatory Alliance

HATS: A Standard for the Hierarchical Adaptive Tiling Scheme in the Virtual Observatory

Version 1.0

IVOA Note 2025-08-22



data.lsdb.io/hats/gaia_dr3/gaia/

Index of /hats/gaia_dr3/gaia

Name	Last modified	Size	Description
Parent Directory		-	Gaia DR3
dataset/	2025-06-13 12:56	-	Gaia DR3
hats.properties	2025-06-13 12:56	414	Gaia DR3
partition_info.csv	2025-06-13 12:40	14K	Gaia DR3
point_map.fits	2025-06-13 12:56	6.0M	Gaia DR3
properties	2025-06-13 12:56	414	Gaia DR3
skymap.2.fits	2025-06-13 12:56	8.4K	Gaia DR3
skymap.4.fits	2025-06-13 12:56	31K	Gaia DR3
skymap.6.fits	2025-06-13 12:56	391K	Gaia DR3
skymap.fits	2025-06-13 12:56	6.0M	Gaia DR3

data.lsdb.io/hats/gaia_dr3/gaia/dataset/

Index of /hats/gaia_dr3/gaia/dat

Name	Last modified	Size	Description
Parent Directory		-	Gaia DR3
Norder=2/	2025-06-13 11:38	-	Gaia DR3
Norder=3/	2025-06-13 11:38	-	Gaia DR3
Norder=4/	2025-06-13 11:38	-	Gaia DR3
Norder=5/	2025-06-13 11:38	-	Gaia DR3
Norder=6/	2025-06-13 11:43	-	Gaia DR3
_common_metadata	2025-06-13 12:56	78K	Gaia DR3
_metadata	2025-06-13 12:56	220M	Gaia DR3
data_thumbnail.parquet	2025-06-13 12:56	1.0M	Gaia DR3

data.lsdb.io/hats/gaia_dr3/gaia/dataset/N

Index of /hats/gaia_dr3/gaia/dat

Name	Last modified	Size	Description
Parent Directory		-	Gaia DR3
Npix=0.parquet	2025-06-13 12:32	312M	Gaia DR3
Npix=1.parquet	2025-06-13 12:02	639M	Gaia DR3
Npix=2.parquet	2025-06-13 11:47	359M	Gaia DR3
Npix=3.parquet	2025-06-13 12:16	751M	Gaia DR3
Npix=8.parquet	2025-06-13 12:14	570M	Gaia DR3
Npix=16.parquet	2025-06-13 12:17	521M	Gaia DR3
Npix=17.parquet	2025-06-13 11:42	321M	Gaia DR3
Npix=18.parquet	2025-06-13 12:33	721M	Gaia DR3
Npix=19.parquet	2025-06-13 11:59	384M	Gaia DR3

Figure 1: Gaia DR3 HATS directories on the LSDB HTTP server.

On-the-fly HATS with QATSS

The purpose of on-the-fly, streamed, HATS products is to provide HATS views of CDS tables, without having to duplicate the data.
So far, all VizieR large tables are potentially available in the HATS format. We are working on possibly adding all other tables from indexed FITS or VOTable files.
For the current list of CDS HATS available products, see:
`https://vizcat.cds.unistra.fr/hats`

CDS HATS Products

How?

1 - Compute the HATS directory structure

- HATS constraint: a tile (i.e. a Parquet file) contains maximum *n* rows.
- CDS large tables are indexed by HEALPix indices from order 0 to 11 (≈ density maps).
- Thus, the HATS structure is computed by a recursive top-down algo on a set of HEALPix density maps.
- Typical performance around 0.06 s (Gaia DR3 structure).

```
let n_srcs = get_index(order).get_cell(ipix).n;
if order == order_max || n_srcs < n_rows_max {
  res.push((order, ipix, n_srcs));
} else {
  split(order + 1, (ipix << 2) | 0, res);
  split(order + 1, (ipix << 2) | 1, res);
  split(order + 1, (ipix << 2) | 2, res);
  split(order + 1, (ipix << 2) | 3, res);
}
```

Figure 2: Recursive HATS structure generation.

2 - Mimick Apache directories web pages

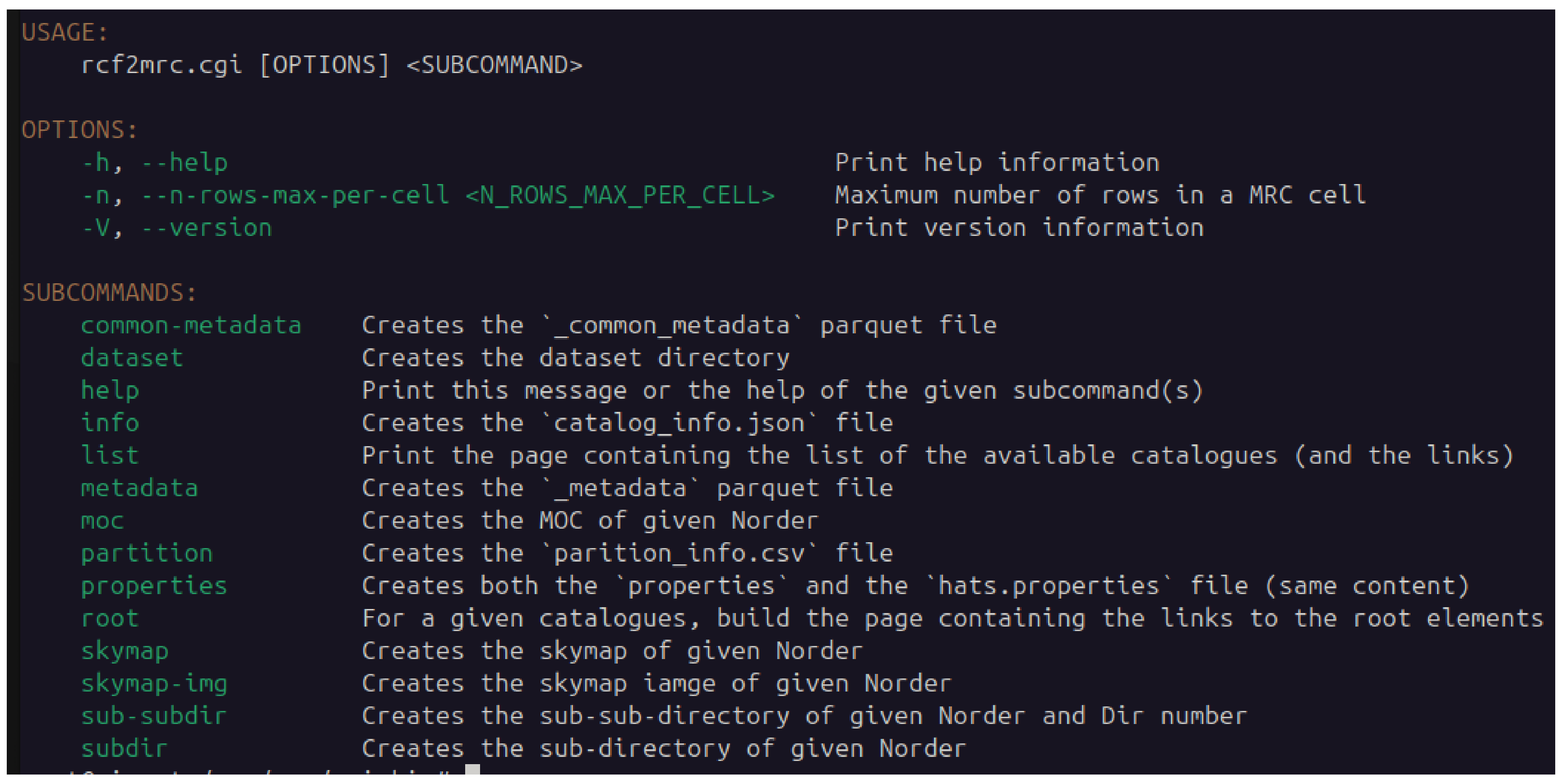
- We developed a command line tool (CLI), usable as a CGI, with commands mapping the various HATS directory traversal actions.
  - print the web page of a HATS product root
  - generate the *properties* file
  - generate skymaps
  - ...
- The mapping between HATS URLs and CGI commands is made using Apache rewrite rules

```
RewriteRule ^/hats:n=(.*)/(.*)/
  /cgi-bin/rcf2mrc.cgi?n=$1
    &action[root][tabname]=$2

RewriteRule ^/hats:n=(.*)/(.*)/dataset/
 /cgi-bin/rcf2mrc.cgi?n=$1
   &action[dataset][tabname]=$2

RewriteRule ^/hats:n=(.*)/(.*)/hats.properties
 /cgi-bin/rcf2mrc.cgi?n=$1
   &action[properties][tabname]=$2
```

Figure 3: Examples of Apache rewrite rules.



```
USAGE:
    rcf2mrc.cgi [OPTIONS] <SUBCOMMAND>

OPTIONS:
    -h, --help                                       Print help information
    -n, --n-rows-max-per-cell <N_ROWS_MAX_PER_CELL>  Maximum number of rows in a MRC cell
    -V, --version                                    Print version information

SUBCOMMANDS:
    common-metadata    Creates the `_common_metadata` parquet file
    dataset            Creates the dataset directory
    help               Print this message or the help of the given subcommand(s)
    info               Creates the `catalog_info.json` file
    list               Print the page containing the list of the available catalogues (and the links)
    metadata           Creates the `_metadata` parquet file
    moc                Creates the MOC of given Norder
    partition          Creates the `parition_info.csv` file
    properties         Creates both the `properties` and the `hats.properties` file (same content)
    root               For a given catalogues, build the page containing the links to the root elements
    skymap             Creates the skymap of given Norder
    skymap-img         Creates the skymap iamge of given Norder
    sub-subdir         Creates the sub-sub-directory of given Norder and Dir number
    subdir             Creates the sub-directory of given Norder
```

Figure 4: Pure Rust CLI / CGI mimicking a static HATS product.

3 - Transform Parquet file URLs into QATSS queries

The URLs of a Parquet file is transformed, thanks to Apache rewrite rules, in a positional query on the *qat2s* CGI. Additional information are extracted from VizieR so that the output Parquet file includes VOTable metadata in the header, as described in the IVOA *VOParquet note*.

VOParquet Note

```
RewriteRule ^/hats:n=(.*)/(.*)/dataset/Norder=(.*)/Dir=.*/Npix=(.*).parquet
  /cgi-bin/qat2s.cgi?file=$2&output=Parquet
    &columns[cols][0]=hpx(29,DEFAULT,DEFAULT)+as+i64+as+_healpix_29
    &columns[cols][1]=*
    &mode[Positional][lon]=DEFAULT
    &mode[Positional][lat]=DEFAULT
    &mode[Positional][geom][Hpx][depth]=$3
    &mode[Positional][geom][Hpx][hash]=$4
```

Figure 5: Simplified rewrite rule transforming a Parquet file URL into a QATSS query.

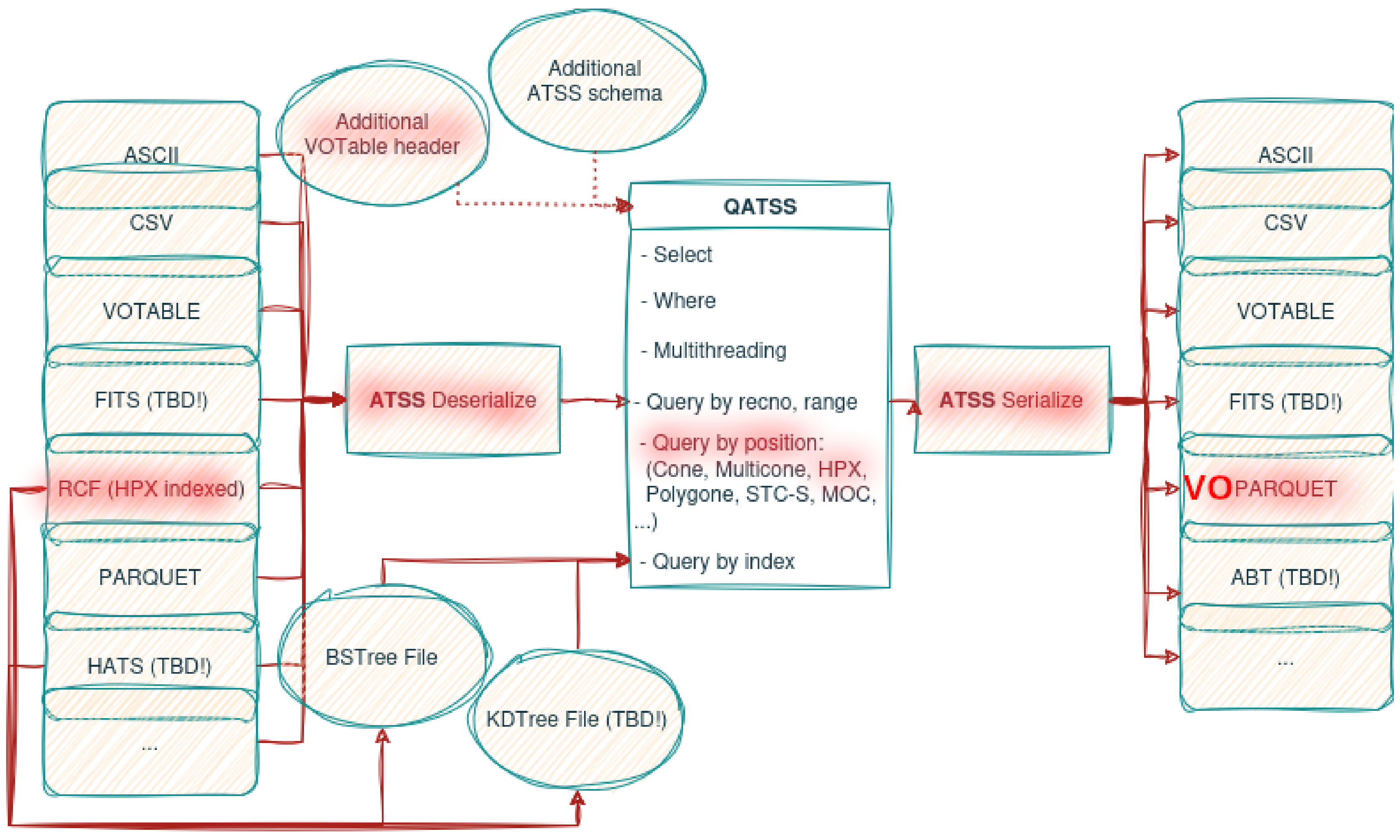


Figure 6: Components around the *qat2s* CGI used for on-the-fly Parquet files generation.

Discussion

On the local network, using ZSTD 13 compression and row groups of 10 000 rows, generating and download the 404 MB of a Gaia DR3 Parquet file takes 20s, i.e. a download rate of 20 MB/s. This could be improved by resorting to multi-threading or even by preparing a row group while the previous one is being transferred.

One of the advantages of Parquet is the ability, for a reader using HTTP range capabilities, to read only a selection of columns without having to download the full file. This is possible by first reading the Parquet footer containing the file structure. This functionality is not possible with on-the-fly, streamed, HATS products. However, the LSDB/HATS steam supports a *columns* query parameter in Parquet file URLs so that the column selection can be performed on server side. Converting from a row oriented (RCF or FITS) to a column oriented (Parquet) format, all columns still have to be read, but the amount of data to be transferred through the network is possibly reduced. Similarly, the team put in place a *filters* query parameter to select rows on server side.

Future work

- Improve download rate by generating a row group while the previous one is being transferred.
- Publish a stand-alone tool generating HATS products from a HEALPIx sorted and indexed FITS file?
  - Methods sorting and indexing FITS-plus files are already implemented in *fitstable-cli*, see the URL `https://github.com/cds-astro/cds-fitstable-rust/tree/main/crates/cli`

François-Xavier Pineau (francois-xavier.pineau@astro.unistra.fr)
Université de Strasbourg, CNRS, Observatoire astronomique de Strasbourg, UMR 7550, F–67000 Strasbourg, France

ADASS XXXV, Astronomical Data Analysis Software & Systems
9 – 13 November 2025, Görlizt, Germany