On-the-fly HATS using QATSS

François-Xavier Pineau[1]

[1]Centre de Données astronomiques de Strasbourg

IVOA, College Park, 03 June 2025

Introduction

- CDS historically developed its own large catalogues format, indices, and query codes. Today:
  - HEALPix sorted and indexed file format
  - QATSS query code, including query by HPX/MOC
- HATS structure is HEALPix based
- What about on-the-fly HATS view on CDS large cat. files?
  - independent implementation
    - provide feedbacks, collaboration with HATS team
  - benefit from HATS tools, avoiding a copy of our data?
  - easy way to create regular HATS from our data if needed?
  - proof-of-concept
    - assess performances
    - assess tools interoperability

ATSS and QATSS

- ATSS: *Astronomical Table Serialisation System*
  - see Adass 2023 P909 poster
  - **abstractions to access/serialize/deserialize tabular data**
  - inspired by serde.rs + schema (≈ TOPCAT StarTable)

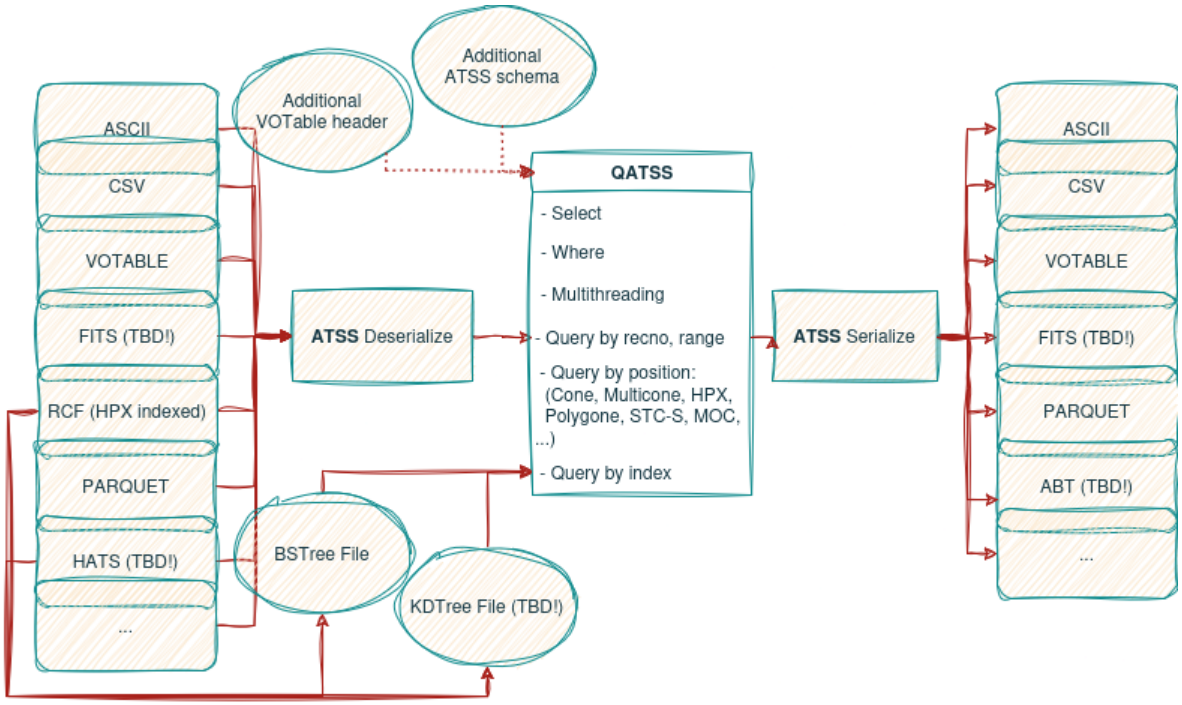


Figure 1: QATSS: **Q**uery tool for **ATSS**

QATSS

- 3 tools sharing the same code and args structures
  - `qat2s`: CLI to query local files (local)
  - `qat2s.cgi`: CGI for remote queries (server side)
  - `qat2s_cli`: CLI to query `qat2s.cgi` (client side)
    - uses reqwest to handle HTTP queries
- Both clap and serde decorate same arg structs
  - clap to transforms command lines into structs
  - serde for generic serialisation/deserialisation
  - serde_qs to create/parse GET query strings from/to structs
  - serde_json to encode/decode POST JSON from/to structs

QATSS

- Example of command line to HTTP query

```
# Get rows 10 to 15 of Gaia DR3 in VOTable
> qat2s_cli --print-url -f vot -s 'DR3Name,/.*_ICRS/' \
            gaia_dr3 rows range 10 5

BASE_URL/qat2s.cgi?file=gaia_dr3
 &output=VOTableTableData
 &columns[cols][0]=DR3Name
 &columns[cols][1]=%2F.*_ICRS%2F
 &mode[Rows][Range][from]=10
 &mode[Rows][Range][n]=5
```

(Not advertised, but easy way to download a full catalogue by regular chunks; similar to ORDER BY oid + OFFSET + LIMIT in DBMS)

HATS

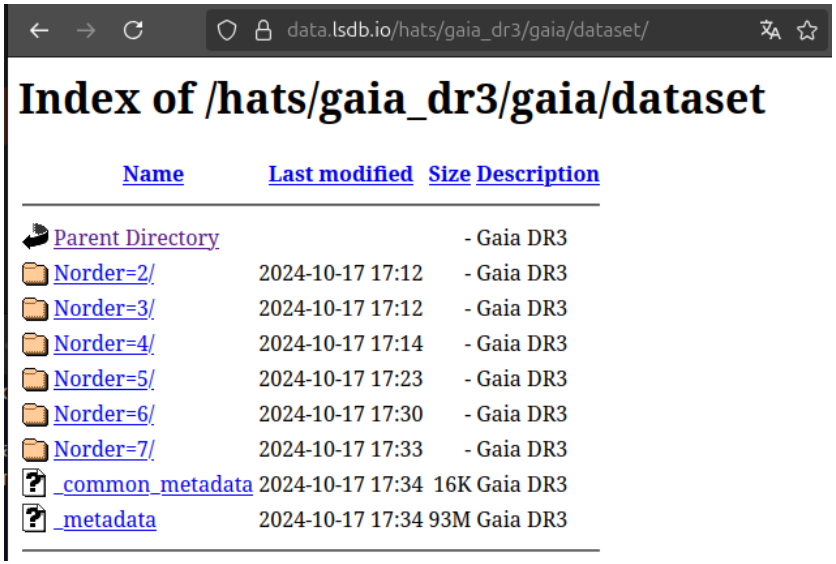
- Catalogues as *Parquet datasets*, i.e. column oriented
  - optimal for compression (storage $): depends on algo
  - ideal for full sky analysis involving a few columns
- *Dataset* based on an *HEALPix adaptative grid* (~MOM)
  - HEALPIx very common (interoperability)
  - balanced (not *well* balanced: sizes ratio x4, possibly x10)

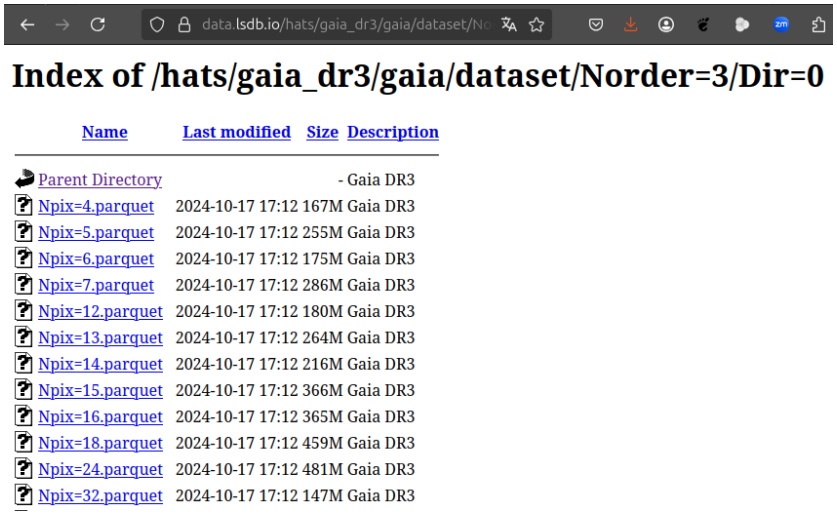
```
dataset/
|-- . . .
|-- Norder=6/
|    |-- Dir=0/
|    |    |-- Npix=0.parquet
|    |    |-- . . .
|    |    +-- Npix=9999.parquet
|    +-- Dir=10000/
|         +-- Npix=10000.parquet
|              +-- . . .
|-- Norder=7/
+-- . . .
```

Figure 2: From the IVOA note in prep. by N. Caplar et al.

HATS

LSDB (Large Survey DataBase) Gaia DR3 HATS product pages examples (simple Apache directories).

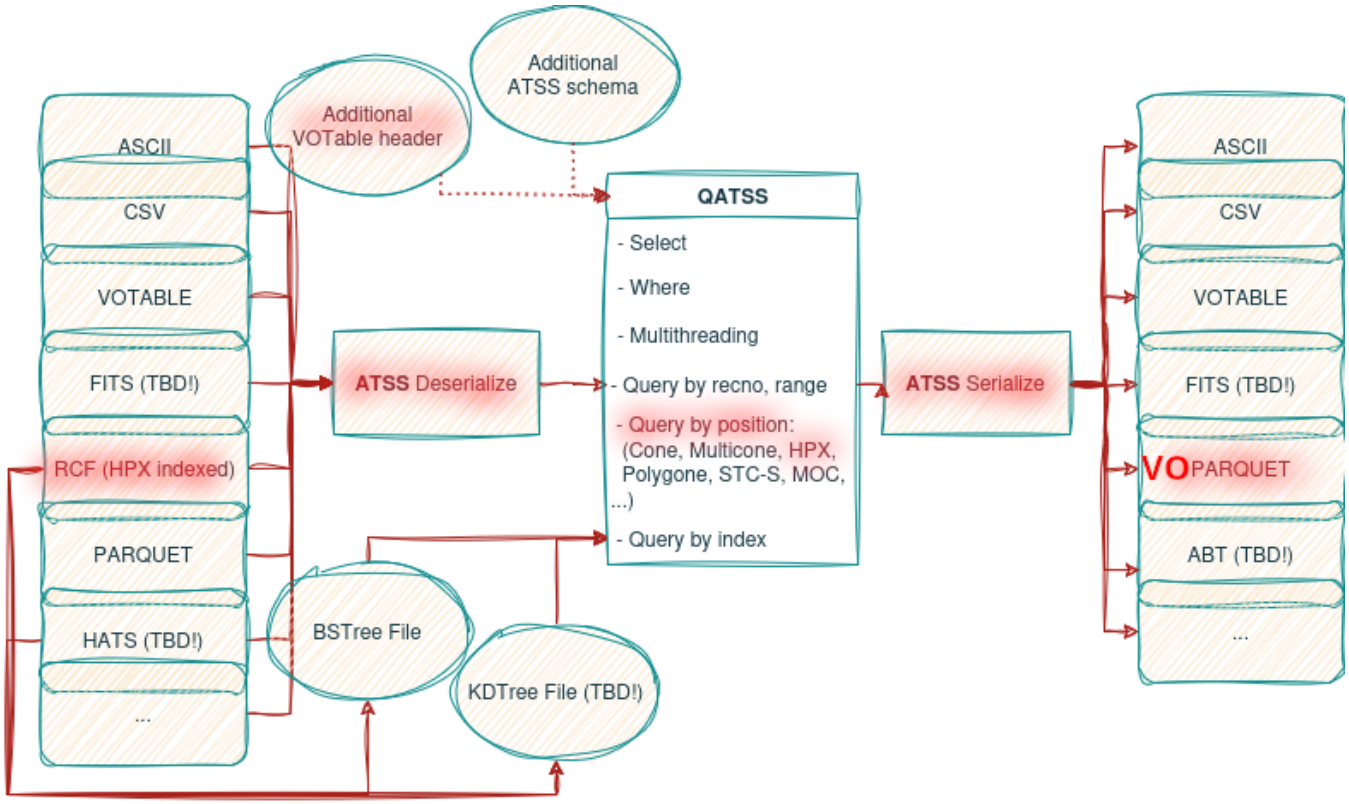




HATS from QATSS

To create on-the-fly HATS views, we need:

- 1 - to compute the HATS directory structure
- 2 - to mimick apache directories web pages
- 3 - to transform Parquet file URLs into queries



HATS from QATSS

- 1 - to compute the HATS directory structure
  - one input parameter: the max number of rows per file
  - RCF files are HEALPix indexed (and physically sorted)
    - HEALPix implicit cumulative count maps
    - order 0 to 11 available
    - $\Rightarrow$ recursive top-down approach on memory mapped blocks

```
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);
}
```

HATS from QATSS

- 2 - to mimick apache directories web pages
  - Single exec (Rust): both a CLI and a CGI
    - no framework needed
  - Apache rewrite rules

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

```
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 the `properties` file
    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
```

HATS from QATSS

- Example: build the *partition_info.csv* file
  - recursive top-down creation of the HATS structure
    - (4x faster than down-top approach here)
  - includes checking a list of authorised names

```
> time ./rcf2mrc.cgi -n 1000000 partition gaia_dr3 | wc -l
Url args: "n=1000000&action[partition][tabname]=gaia_dr3"
3937

real    0m0,066s
user    0m0,053s
sys     0m0,021s
```

HATS from QATSS

```
fxpineau@vizcat:/var/www/cgi-bin$ time ./rcf2mrc.cgi -n 1000000 dataset gaia_dr3 2> /dev/null
Accept-Ranges: None
Content-Type: text/html;charset=ISO-8859-1

<html>
  <head>
    <title>gaia_dr3 HATS</title>
  </head>
  <body>
    <h1>Index of the HATS product: gaia_dr3.</h1>
    <p>
      The current structure of each product is based on
      <a href="https://data.lsdb.io/hats/gaia_dr3/gaia/">the LSDB Gaia example</a>
      plus discussion with the <a href="https://github.com/astronomy-commons/hats">HATS team repo</a>.
      Please tell us if you encounter discrepencies.
    </p>
    <p>
      This page is generated on-the-fly from HEALPix indexed data.
    </p>
    <table>

<tr><th valign="top"><img src="/icons/blank.gif" alt="[ICO]"></th><th>Name</th><th>Last modified</th><th>Size</th><th>Description</th></tr>
<tr><th colspan="5"><hr></th></tr>
<tr><td valign="top"><img src="/icons/back.gif" alt="[PARENTDIR]"></td><td><a href="/hats:n=1000000/gaia_dr3/">Parent Directory</a></td><td>
<tr><td valign="top"><img src="/icons/folder.gif" alt="[DIR]"></td><td><a href="Norder=2/">Norder=2/</a></td><td align="right">2022-05-31 00
<tr><td valign="top"><img src="/icons/folder.gif" alt="[DIR]"></td><td><a href="Norder=3/">Norder=3/</a></td><td align="right">2022-05-31 00
<tr><td valign="top"><img src="/icons/folder.gif" alt="[DIR]"></td><td><a href="Norder=4/">Norder=4/</a></td><td align="right">2022-05-31 00
<tr><td valign="top"><img src="/icons/folder.gif" alt="[DIR]"></td><td><a href="Norder=5/">Norder=5/</a></td><td align="right">2022-05-31 00
<tr><td valign="top"><img src="/icons/folder.gif" alt="[DIR]"></td><td><a href="Norder=6/">Norder=6/</a></td><td align="right">2022-05-31 00
<tr><td valign="top"><img src="/icons/folder.gif" alt="[DIR]"></td><td><a href="Norder=7/">Norder=7/</a></td><td align="right">2022-05-31 00
<tr><td valign="top"><img src="/icons/unknown.gif" alt="[   ]"></td><td><a href="_common_metadata">_common_metadata</a></td><td align="right
<tr><td valign="top"><img src="/icons/unknown.gif" alt="[   ]"></td><td><a href="_metadata">_metadata</a> (read text)</td><td align="right">
<tr><th colspan="5"><hr></th></tr>

    </table>
    <p>
      WARNING: This is a proof-of-concept in an <b>pre-alpha version: do not use for serious buisness!</b><br />
      WARNING: <b>Do not use as a way to harvest larges catalogues</b> without the consent of the CDS (and possibly of original providers)!
    </p>
  </body>
</html>


real    0m0,059s
user    0m0,056s
sys     0m0,004s
```

HATS from QATSS

Index of the HATS product: gaia_dr3.

The current structure of each product is based on the LSDB Gaia example plus discussion with the HATS team repo. Please tell us if you encounter discrepencies.

This page is generated on-the-fly from HEALPix indexed data.

Name	Last modified	Size	Description
Parent Directory		-	gaia_dr3
Norder=2/	2022-05-31 00:00	-	gaia_dr3
Norder=3/	2022-05-31 00:00	-	gaia_dr3
Norder=4/	2022-05-31 00:00	-	gaia_dr3
Norder=5/	2022-05-31 00:00	-	gaia_dr3
Norder=6/	2022-05-31 00:00	-	gaia_dr3
Norder=7/	2022-05-31 00:00	-	gaia_dr3
_common_metadata	2022-05-31 00:00	-	gaia_dr3
_metadata (read text)	2022-05-31 00:00	-	gaia_dr3

WARNING: This is a proof-of-concept in an **pre-alpha version: do not use for serious buisness!**
WARNING: **Do not use as a way to harvest larges catalogues** without the consent of the CDS (and possibly of original providers)!

HATS from QATSS

- 3 - to transform Parquet file URLs into queries

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

Gen/Download speed

- 15s to generate a **382 MB** tile (+ 1.6s to download it locally)
  - mean of **25 MB/ in memory** (single thread, **SNAPPY**)
    - multi-thread? (does not seem trivial)
  - 15s = **delay before download** starts!
    - because of row-to-col conversion with a single *RowGroup*
- **Total time doubled** (30s) with a 25 MB/s network connexion
- Same tile in CSV: 15s, 970 MB (**394 MB gzipped**)
  - no row-to-col conversion: streaming, no delay
  - would be 39s at 25 MB/s (without compression)
- ZSTD l3: CSV = 411 MB; Parquet = 339 MB (-18%)

```
fxpineau@vizcat:~/tmp$ time wget 'https://vizcat.cds.unistra.fr/hats:n=1000000/gaia_dr3/dataset/Norder=2/Dir=0/Npix=0.parquet'
--2025-05-29 11:04:01--  https://vizcat.cds.unistra.fr/hats:n=1000000/gaia_dr3/dataset/Norder=2/Dir=0/Npix=0.parquet
Resolving vizcat.cds.unistra.fr (vizcat.cds.unistra.fr)... 130.79.128.6
Connecting to vizcat.cds.unistra.fr (vizcat.cds.unistra.fr)|130.79.128.6|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/vnd.apache.parquet]
Saving to: ‘Npix=0.parquet.1’

Npix=0.parquet.1                    [         <=>                                    ] 381,62M   234MB/s    in 1,6s

2025-05-29 11:04:18 (234 MB/s) - ‘Npix=0.parquet.1’ saved [400154779]


real    0m16,722s
user    0m0,542s
sys     0m1,105s
fxpineau@vizcat:~/tmp$ ls -lh *.parquet
-rw-rw-r-- 1 fxpineau fxpineau 382M mai   29 11:02 'Npix=0.parquet'
```

Gen/Download speed

- Downloading the equivalent tile from LSBD:
  - 411 MB, 24s, **18 MB/s** (=network speed, I guess)
- On-the-fly vs normal: **factor less than x2** (in this case)
  - **generation time faster than network speed** between University of Washington and University of Strasbourg
  - **factor would be x1** with *RowGroup* of e.g. 10 000 rows
    - but not great for compression/big data tools

```
fxpineau@vizcat:~/tmp$ time wget 'https://data.lsdb.io/hats/gaia_dr3/gaia/dataset/Norder=2/Dir=0/Npix=0.parquet'
--2025-05-30 10:25:34--  https://data.lsdb.io/hats/gaia_dr3/gaia/dataset/Norder=2/Dir=0/Npix=0.parquet
Resolving data.lsdb.io (data.lsdb.io)... 128.95.0.72
Connecting to data.lsdb.io (data.lsdb.io)|128.95.0.72|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 431023122 (411M)
Saving to: ‘Npix=0.parquet’

Npix=0.parquet              100%[=========================================>] 411,06M  19,1MB/s    in 23s

2025-05-30 10:25:58 (18,1 MB/s) - ‘Npix=0.parquet’ saved [431023122/431023122]


real    0m23,992s
user    0m0,445s
sys     0m1,615s
```

Gen/Download speed

Conclusion on generation/download speed and compression:

- compressed CSV vs compressed Parquet: $\approx 20\%$ "only" for Gaia DR3
  - (Parquet high compression rate reached for de-normalized tables, categories, ...)
- generation speed faster than network between Washington and Strasbourg
- main limit: a delay almost doubling download time
  - due to row-oriented to col-oriented conversion
  - due to a single *RowGroup*
  - almost no delay if small *RowGroups* were ok

The Main Limitation!

As for today, **LSDB tools do not read on-the-fly HATS**!

- **Parquet meant for storage**, accessed by **smart readers** to prune blocks to be read
  - **smart readers** need random access, even through the network with **HTTPRange**
  - e.g. metadata in the footer, to be read first
  - (Not the Unix Philosophy: the **new CSV?**)
- On-the-fly: single pass write, streamed
  - **no seek**, **unknown size**
  - **no *_metadata*** file

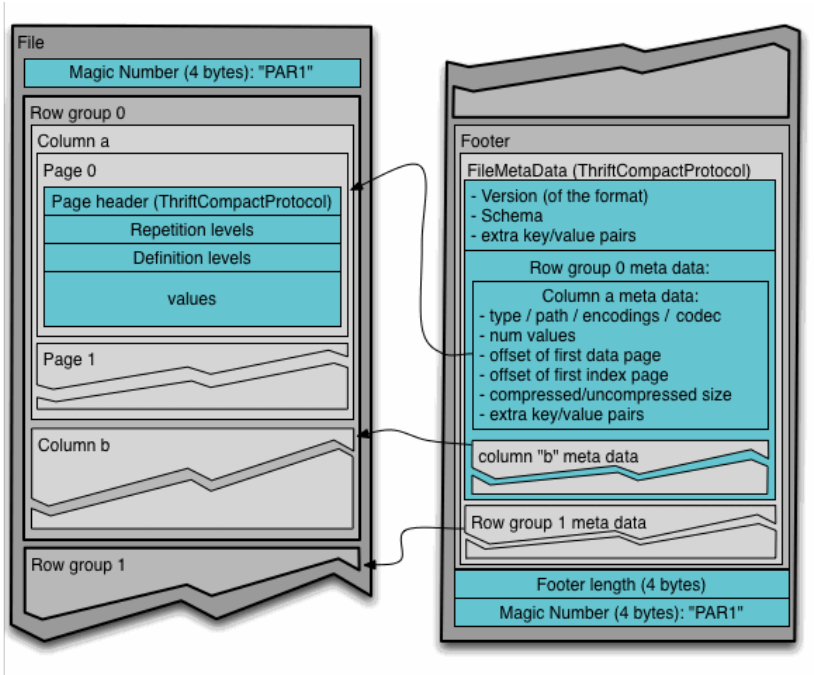


Figure 3: https://parquet.apache.org/docs/file-format/

Possible solutions

- Hold the full file in memory (or copy on disk) on server side
  - see G. Schwarz, cited by K. Malanchev, work here
    - done for server side pruning
  - I am reluctant to do so
    - hold memory/disk on a server with possibly a lot of queries
    - no more "fire and forget": protocol needed?
- Cache file on client side, see e.g. this discussion
  - transfert even data to be pruned
    - except if pruning constraints are provided to the server
  - too much work, need to modify external libraries (if I understood correctly M. DeLucchi)

$\Rightarrow$ we seem to be stuck!

Advantages

On-the-fly advantages:

- Agile solution: perform changes without re-writing
  - update VOParquet metadata in all files instantaneously
  - update Parquet version without re-writing files
  - change HATS version (new file names, new layout)
  - change compression (i.e. SNAPPY -> ZSTD)
- Possibility to dynamically adapt tile sizes
  - e.g. dynamic max number of rows depending on requested number of columns

Thus despite previous slide drawbacks:

- Still useful to process a full catalogue tile by tile?
  - on a resource limited computer
  - using specific tools downloading each file
  - not looking at best performances

Conclusion

- **Parquet (hence HATS) is great**:
  - **for storage** (Gaia DR3: $\approx -20\%$ wrt CSV, using ZSTD)
    - assuming compatible implementations
  - for **batch analysis** involving a few columns and a lot of rows
    - (see also ColFITS by M. Taylor)
- **Parquet is not a transfert format**:
  - full file on disk or in memory, HTTPRange needed
  - stream reading not possible
- **HiPS** made for remote progressive visualisation with limited data transferts;
- **HATS** made for batch processing with fast network between workers and storage?
- **On-the-fly, streamed, HATS only useful to make copies on a distributed storage** (on the same network as executor machines) **?**
  - if so, raises political questions

Appendix

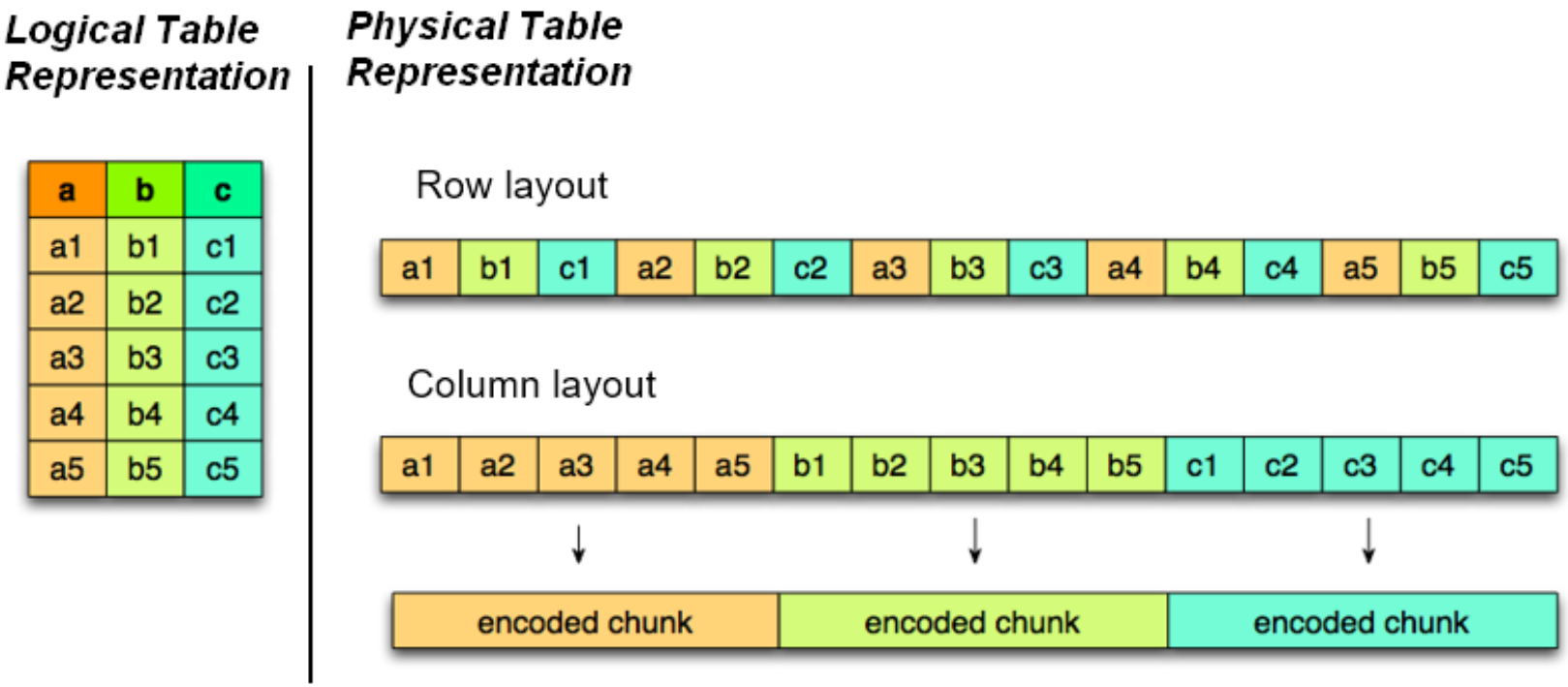
Backup slides

Going further

- What about column selection and row filtering on server side?
  - G. Schwarz experiment, LSDB team discussions
  - orthogonal to Parquet purpose?
  - defining interfaces (URLs, Query Params, transfert format) instead of the implementation (both Parquet format + readers) isn't better for interoperability?
- On-the-fly HATS from RCF: what about other formats?
  - any HEALPix indexed (and clustered!!) DBMS table is OK
    - better if col-oriented format/database
  - see hpx-cli and vot-cli for HPX external sort and HPX indexation of large CSV and VOTable files
    - build HATS from a large (a few TB) CSV file on a regular computer? See K. Malanchev issue

Row-oriented to col-oriented

- Row-oriented to col-oriented conversion is not efficient
  - hold a full *RowGroup* data in memory (~400 MB)
  - not a CPU cache friendly operation
- One row-oriented + one col-oriented version of the data, depending on the use case?



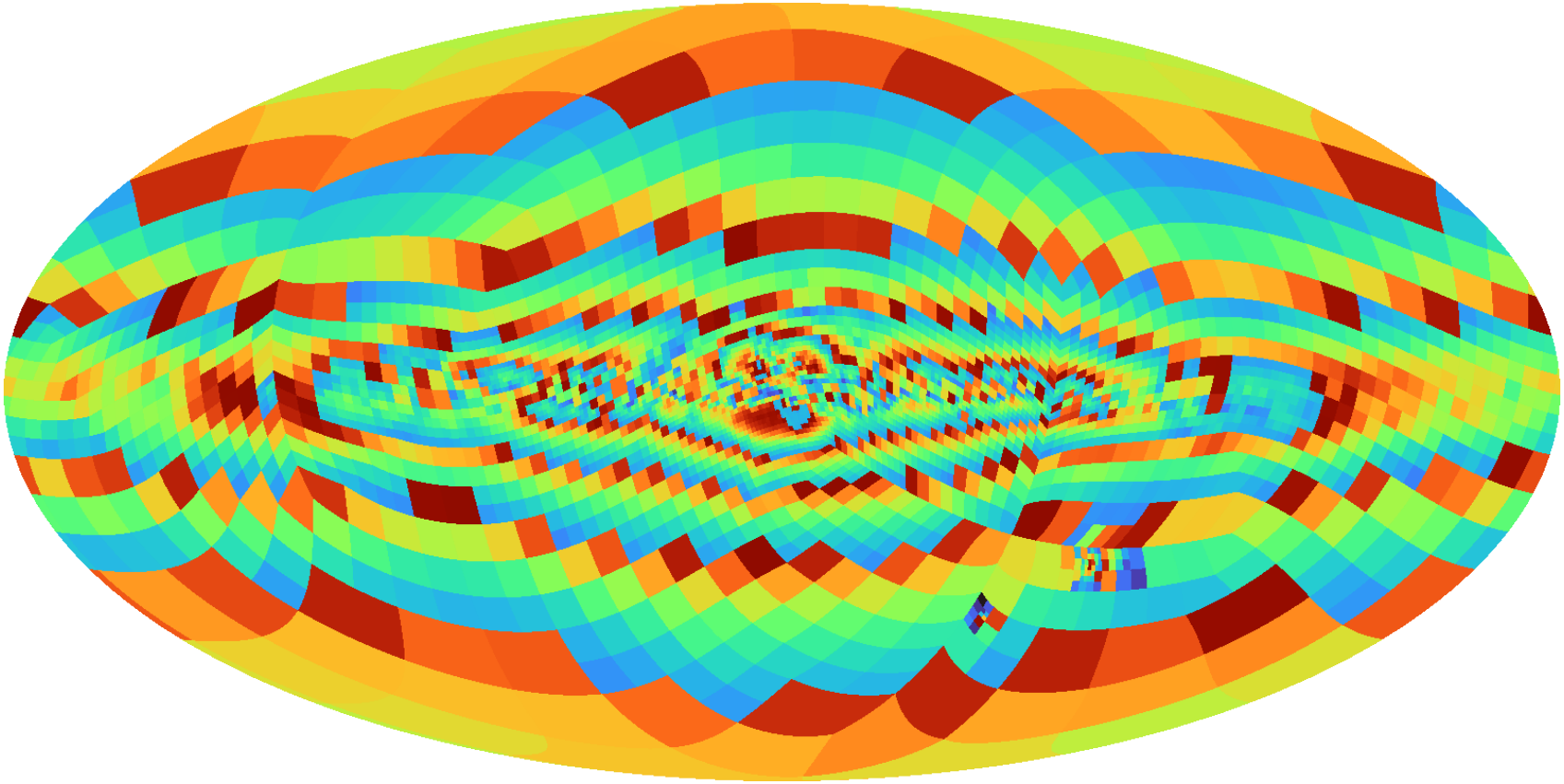
Row-oriented vs col-oriented

- What about a lot of parallel queries on various tables, with a lot of columns and a few rows? (E.g. VizieR queries)
  - Parquet page size = a few MB
  - to get 1 row and 30 columns: read and uncompress 30 pages, i.e. 30 seeks, a few 100's of MB
- Row- vs col-oriented depends on problems to be solved

Gaia DR3 HATS struct

Computing and exploring the Gaia DR3 HATS structure.

Gaia DR3 HATS struct



Gaia DR3 HATS struct

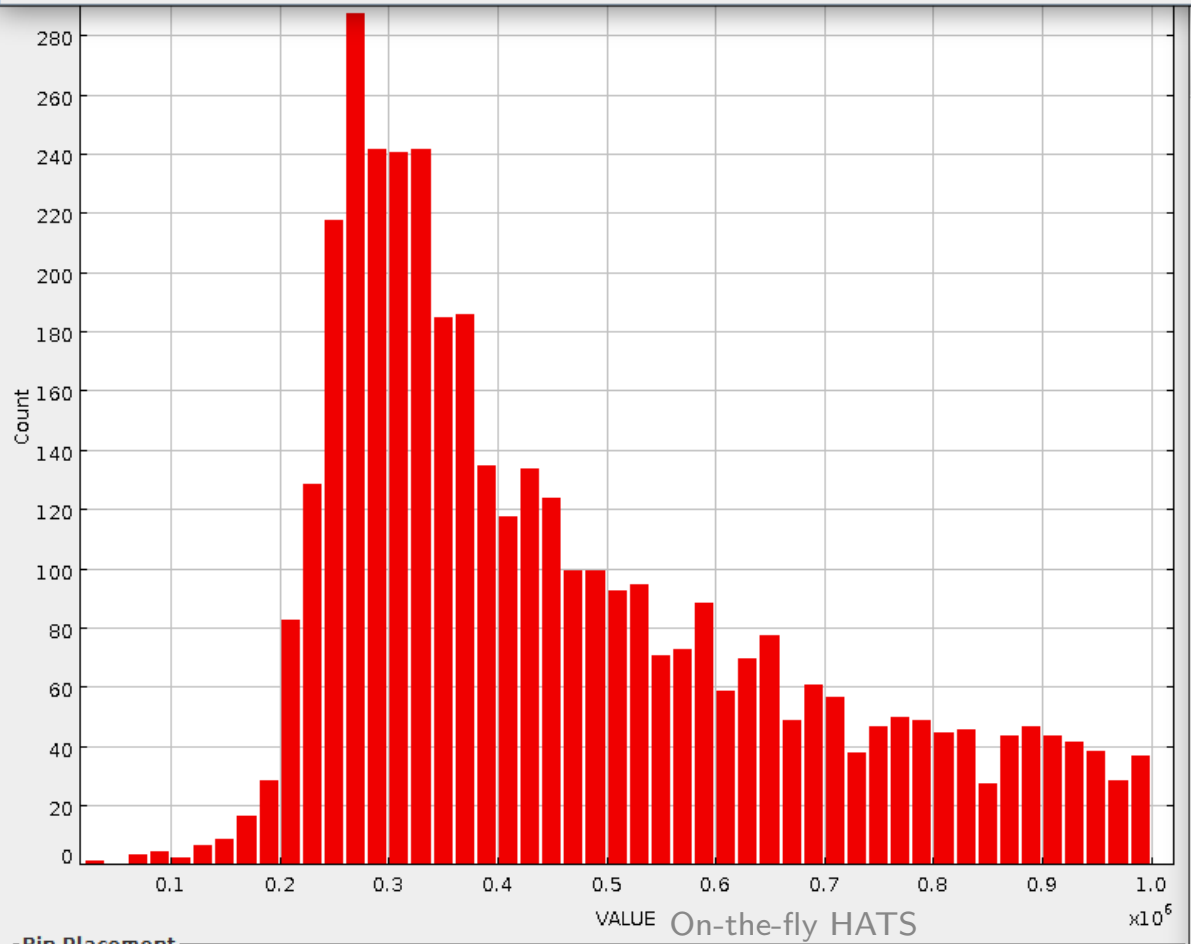
TOPCAT(1): Row Statistics

Window Export Statistics Display Help

Row Statistics for 1: gaia_dr3.hats.bintable.fits

Name	Sum	Median	Mean	SD	Minimum	Maximum	nGood
ZUNIQ	1,725996E22	4,311845E18	4,388497E18	1,63853E18	18014398509481984	6913025428013711360	3933
VALUE	1,811710E9	3,936600E5	4,606432E5	2,07970E5	34938	999836	3933

Subset for calculations: All



Window Rows Help

Table Browser for 1: gaia_dr3.hats.bintab

	ZUNIQ	VALUE
1	18014398509481984	726621
2	40532396646334464	271350
3	49539595901075456	435630
4	58546795155816448	285747
5	67553994410557440	492522
6	90071992547409920	837288
7	112589990684262400	294376
8	121597189939003392	448477
9	130604389193744384	361465
10	139611588448485376	641496
11	148618787703226368	640850
12	154248287237439488	267547
13	156500087051124736	400237
14	158751886864809984	249258
15	161003686678495232	343519
16	166633186212708352	821148
17	172262685746921472	278868
18	174514485560606720	312356
19	176766285374291968	289228
20	179018085187977216	339041
21	181269885001662464	527339
22	183521684815347712	572046
23	185773484629032960	526562
24	188025284442718208	672501
25	190277084256403456	554882
26	192528884070088704	478665
27	194780683883773952	625697
28	197032483697459200	420789
29	199284283511144448	551310
30	201536083324829696	779065
31	203787883138514944	627276
32	206039682952200192	700946

Gaia DR3 HATS struct

Instructions:

- download Skymap file (T. Boch) at order 10 ($3.4' x 3.4'$)
- create a MOM from the Skymap with hpx-cli
- convert the FITS MOM into FITS BINTABLE with hpx-cli
- view with hpx-cli
- load in TOPCAT to build the histogram

Gaia DR3 HATS struct

Command lines:

```
> wget -O gdr3.d10.skymap.fits "${URL}"
# gdr3.d10.skymap.fits is 97 MB large
> time hpx map convert gdr3.d10.skymap.fits gdr3.hats.fits
      count2mom 1000000

real    0m0,280s
user    0m0,155s
sys     0m0,115s
> hpx mom convert gdr3.hats.fits gdr3.hats.bintable.fits \
      bintable
> hpx mom view gdr3.hats.fits gdr3.hats.png \
      -c linear allsky 800
> topcat gdr3.hats.bintable.fits &
```

ATSS (or AT2S)

- ATSS: *Astronomical Table Serialisation System*
  - see Adass 2023 poster P909
  - **abstractions to serialize/deserialize tabular data**
  - specific implementation for each format
  - inspired by serde.rs (+ similar to TOPCAT StarTable?)
- Project started in 2020, during COVID
- Context: VizieR large catalogues, XMatch service
- Aims:
  - enhance the private/internal `B/RCF` (Block/Row Cat File) formats
  - better support for **various file formats**
    - both in input and output
    - neat distinction between **transport and display** formats
  - possible **compression** preserving **fixed row byte size**
  - same core code to **perform queries on many file formats**

QATSS STC-S example

- Data: **1.5 TB Gaia DR3** RCF file
- Simple query (first row) takes 100 ms:

```
> time qat2s --select 'DR3Name,RAdeg,DEdeg,Source' \
>            --print-header                        \
>            --limit 1                             \
>            gaia_dr3.rcf all
DR3Name,RAdeg,DEdeg,Source
Gaia DR3 34361129088,45.00432028915398,0.021047763781174733

real    0m0,101s
user    0m0,096s
sys     0m0,005s
```

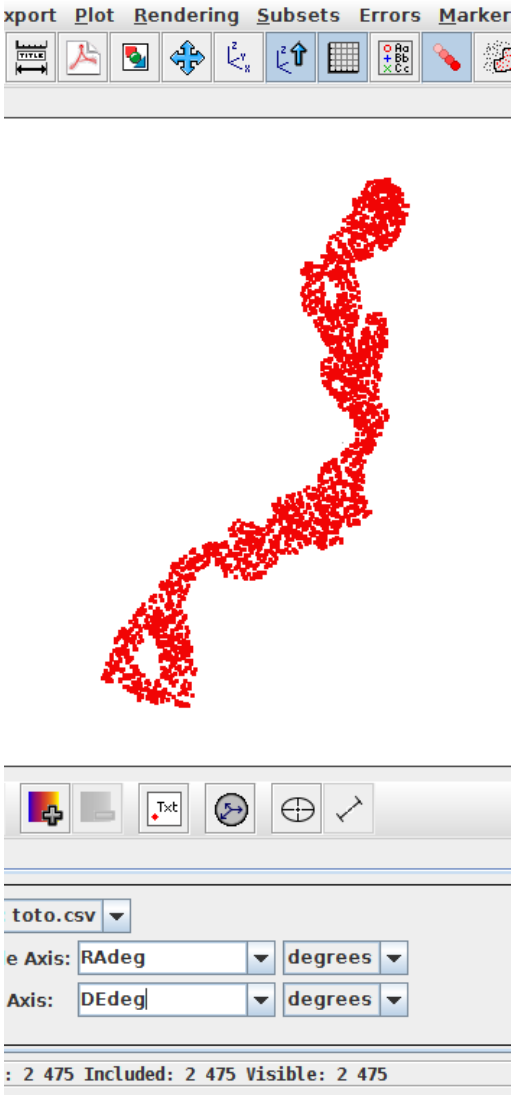
QATSS STC-S example

Complex (ALMA) STC-S query:

- takes 190 ms to
- (2475 rows returned)

```
time qat2s --print-header --select 'DR3Name,RAdeg,DEdeg,Source' gaia_dr3.rcf pos DEFAULT DEFAULT stcs \
  "Intersection ICRS (
    Polygon 272.536719 -19.461249 272.542612 -19.476380 272.537389 -19.491509 272.540192 -19.499823
            272.535455 -19.505218 272.528024 -19.505216 272.523437 -19.500298 272.514082 -19.503376
            272.502271 -19.500966 272.488647 -19.490390  272.481932 -19.490913 272.476737 -19.486589
            272.487633 -19.455645 272.500386 -19.444996 272.503003 -19.437557 272.512303 -19.432436
            272.514132 -19.423973 272.522103 -19.421523 272.524511 -19.413250 272.541021 -19.400024
            272.566264 -19.397500 272.564202 -19.389111 272.569055 -19.383210 272.588186 -19.386539
            272.593376 -19.381832 272.596327 -19.370541 272.624911 -19.358915 272.629256 -19.347842
            272.642277 -19.341020 272.651322 -19.330424 272.653174 -19.325079 272.648903 -19.313708
            272.639616 -19.311098 272.638128 -19.303083 272.632705 -19.299839 272.627971 -19.289408
            272.628226 -19.276293 272.633750 -19.270590 272.615109 -19.241810 272.614704 -19.221196
            272.618224 -19.215572 272.630809 -19.209945 272.633540 -19.198681 272.640711 -19.195292
            272.643028 -19.186751 272.651477 -19.182729 272.649821 -19.174859 272.656782 -19.169272
            272.658933 -19.161883 272.678012 -19.159481 272.689173 -19.176982 272.689395 -19.183512
            272.678006 -19.204016 272.671112 -19.206598 272.664854 -19.203523 272.662760 -19.211156
            272.654435 -19.214434 272.652969 -19.222085 272.656724 -19.242136 272.650071 -19.265092
            272.652868 -19.274296 272.660871 -19.249462 272.670041 -19.247807 272.675533 -19.254935
            272.673291 -19.273917 272.668710 -19.279245 272.671460 -19.287043 272.667507 -19.293933
            272.669261 -19.300601 272.663969 -19.307130 272.672626 -19.308954 272.675225 -19.316490
            272.657188 -19.349105 272.657638 -19.367455 272.662447 -19.372035 272.662232 -19.378566
            272.652479 -19.386871 272.645819 -19.387933 272.642279 -19.398277 272.629282 -19.402739
            272.621487 -19.398197 272.611782 -19.405716 272.603367 -19.404667 272.586162 -19.422703
            272.561792 -19.420008 272.555815 -19.413012 272.546500 -19.415611 272.537427 -19.424213
            272.533081 -19.441402
    Not (Polygon 272.511081 -19.487278 272.515300 -19.486595 272.517029 -19.471442
                 272.511714 -19.458837 272.506430 -19.459001 272.496401 -19.474322 272.504821 -19.484924)
    Not (Polygon 272.630446 -19.234210 272.637274 -19.248542 272.638942 -19.231476 272.630868 -19.226364)
)" | wc -l
2476

real    0m0,188s
user    0m0,179s
sys     0m0,011s
```



RCF File Format

- B/RCF (Block/Row Cat File) formats developped for the XMatch service (2010)
  - fixed size header (8 kB, easy to modify with *dd*)
    - name, type (B/CF), creation date, number of rows
    - blocks (ID, POSITIONS, ...): col names, datatype, units, ucds, format
  - header = sequence of "`KEY :`" + command line arguments
  - rows are HEALPix sorted
    - for any HEALPix cell, rows are consecutives!!
  - contains cumulative HEALPix count maps at orders 0 to 11
    - either implicit or explicit