

VOTable Metadata for Parquet

Mark Taylor (University of Bristol)

Applications WG
Telecon

5 November 2024

`$Id: votparquet.tex,v 1.12 2024/11/05 12:59:46 mbt Exp $`

History

This builds on previous work

- Gregory Dubois-Felsmann [presentation Tucson 2023](#)
- Brigitta Sipőcz [Astropy PR #16375](#)
- François-Xavier Pineau [CDS HATS](#) service implementation
- Probably others (LINCC/LSDB?)

Basic Idea

Parquet

- 😊 Efficient (column-oriented, transparent compression)
- 😊 Industry standard, lots of tool support
- 😊 Good for storage/distribution/processing of very large tables, science-platform friendly
- 😞 Very limited standard metadata (columns just have names + types)

VOTable

- 😊 Rich metadata
- 😞 Standard within VO, but not used in industry
- 😞 Not good for huge tables

⇒ Combine the two

- Parquet data storage + VOTable metadata
- If you can already work with Parquet and VOTable, it is not particularly hard to implement (depending on the details...)

Parquet/VOTable Combination Options

VOTable-in-Parquet

- Prepare a **DATA**-less VOTable describing the table metadata
- Store VOTable XML in Parquet file-level Key-Value metadata
 - ▷ Implementations: [Astropy PR #16375](#), [CDS HATS](#), [TOPCAT prototype](#), [Firefly PR #1633](#)
 - ▷ File is valid parquet: non-VOTable-aware consumers just ignore the extra metadata
 - ▷ Resembles “FITS-plus” convention long used by TOPCAT

VOTable-in-Avro-in-Parquet

- Prepare a **DATA**-less VOTable describing the table metadata
- Store VOTable XML in Avro schema, which is encoded and stored in Parquet file-level Key-Value metadata (**ARROW:schema**)
 - ▷ Implementations: [Astropy PR #16375](#), [CDS HATS](#), [LSDB](#)
 - ▷ File is valid parquet: non-VOTable-aware/non-Avro-aware consumers just ignore the extra metadata
 - ▷ Makes assumptions about Parquet parsing software (Avro-based)

Parquet-in-VOTable

- Write a VOTable document with normal metadata
- But **DATA** element has a **PARQUET** child referencing external file (not **TABLEDATA**, **BINARY** or **FITS**)
 - ▷ Implementations: [Astropy 6.0](#)
 - ▷ Resembles existing VOTable-FITS wrapping
 - ▷ File not valid VOTable; requires major update to VOTable standard (VOTable parsers would have to parse Parquet)

Parquet/VOTable Combination Options

VOTable-in-Parquet

- Prepare a **DATA**-less VOTable describing the table metadata
- Store VOTable XML in Parquet file-level Key-Value metadata
 - ▷ Implementations: [Astropy PR #16375](#), [CDS HATS](#), [TOPCAT prototype](#), [Firefly PR #1633](#)
 - ▷ File is valid parquet: non-VOTable-aware consumers just ignore the extra metadata
 - ▷ Resembles “FITS-plus” convention long used by TOPCAT

← I like this one

VOTable-in-Avro-in-Parquet

- Prepare a **DATA**-less VOTable describing the table metadata
- Store VOTable XML in Avro schema, which is encoded and stored in Parquet file-level Key-Value metadata (**ARROW:schema**)
 - ▷ Implementations: [Astropy PR #16375](#), [CDS HATS](#), [LSDB](#)
 - ▷ File is valid parquet: non-VOTable-aware/non-Avro-aware consumers just ignore the extra metadata
 - ▷ Makes assumptions about Parquet parsing software (Avro-based)

Parquet-in-VOTable

- Write a VOTable document with normal metadata
- But **DATA** element has a **PARQUET** child referencing external file (not **TABLEDATA**, **BINARY** or **FITS**)
 - ▷ Implementations: [Astropy 6.0](#)
 - ▷ Resembles existing VOTable-FITS wrapping
 - ▷ File not valid VOTable; requires major update to VOTable standard (VOTable parsers would have to parse Parquet)

Side note: Parquet per-column metadata

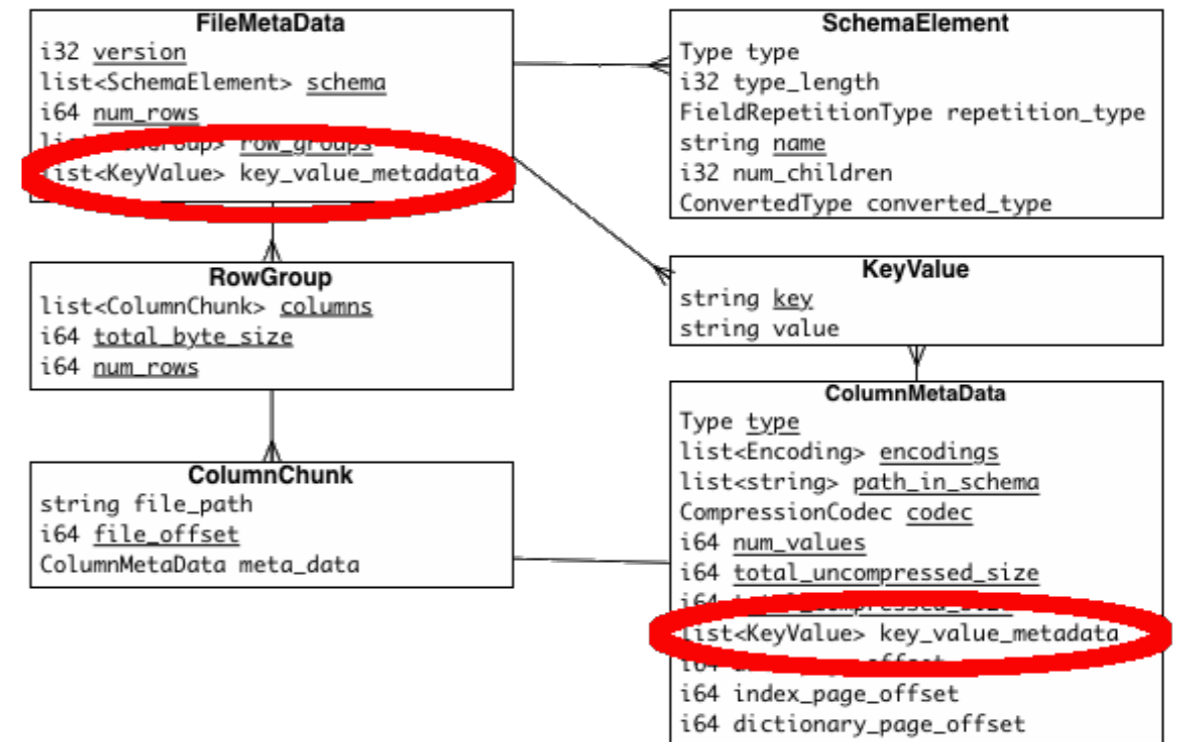
The Parquet format has per-column, as well as per-table, key-value metadata

- Could be used to store column units, descriptions, UCDs, ...

This might be a cleaner solution, but...

- This feature is not widely used in parquet files?
- Some parquet libraries do not provide access to it:
 - ✓ parquet-rs (rust)
 - ✗ parquet-mr (java)
 - ✗? fastparquet (python)
 - ✗? pyarrow (python)
- Doesn't cover per-file metadata items already standardised in VOTable
 - ▷ COOSYS, TIMESYS, truncation, ...

So ignore this possibility?



<https://parquet.apache.org/docs/file-format/metadata/>

VOTable-in-Parquet Details

From Gregory Dubois-Felsmann's [Tucson 2023 presentation](#) (p. 8):

The file-level metadata of the Parquet file MUST include these key-value pairs:

`IVOA.VOTable-Parquet.version = 1.0`

`IVOA.VOTable-Parquet.type = "Parquet-local-XML"`

`IVOA.VOTable-Parquet.encoding = (encoding for the XML content of the following)`

`IVOA.VOTable-Parquet.content = (character data in a format valid for an XML file,
comprising a single <VOTABLE> element and its hierarchical content)`

The `<VOTABLE>` must contain exactly one `<TABLE>` with its `<DATA>` content specified as follows:

`<DATA><PARQUET type="Parquet-local-XML" /></DATA>`

I have a few quibbles:

- VOTable should have no `<DATA>` element; suggested `<PARQUET>` content is invalid VOTable
 - ▷ Most prototypes omit `<PARQUET>` element: [Firefly PR #1633](#), [TOPCAT prototype](#), [CDS HATS](#), though [Astropy PR #16375](#) includes it
- Number of keys could be reduced (to only one: `content?`)
 - ▷ Fixing XML `encoding` to UTF-8 would reduce implementation burden
 - ▷ Not clear what readers should do if `version` or `type` items are different/missing

STIL Prototype Implementation

```
% java -jar topcat-extra_metaparq.jar -stilts tpipe in=:skysim:3 cmd='keepcols "ra dec"' ofmt='parquet(votmeta=true)' out=sky.parquet
% java -classpath /mbt/github/parquet-mr/parquet-cli/target/parquet-cli-1.13.1.jar:'echo /mbt/github/parquet-mr/parquet-cli/target/dependency/*.jar | tr ' ' :'
  org.apache.parquet.cli.Main meta sky.parquet
```

File path: sky.parquet

Created by: parquet-mr version 1.13.1 (build db4183109d5b734ec5930d870cdae161e408ddba)

Properties:

IVOA.VOTable-Parquet.content: <?xml version='1.0'?>

```
<VOTABLE version="1.4" xmlns="http://www.ivoa.net/xml/VOTable/v1.3">
```

```
<RESOURCE>
```

```
<TABLE name="SimulatedSky-3" nrows="3">
```

```
<DESCRIPTION>Simulated star catalogue, based on Gaia EDR3</DESCRIPTION>
```

```
<PARAM datatype="int" name="HealpixLevel" value="5">
```

```
<DESCRIPTION>Healpix level of aggregated statistics</DESCRIPTION>
```

```
</PARAM>
```

```
<FIELD datatype="float" name="ra" ucd="pos.eq.ra" unit="deg"><DESCRIPTION>ICRS Right Ascension</DESCRIPTION></FIELD>
```

```
<FIELD datatype="float" name="dec" ucd="pos.eq.dec" unit="deg"><DESCRIPTION>ICRS Declination</DESCRIPTION></FIELD>
```

```
<!-- Dummy VOTable - no DATA element -->
```

```
</TABLE>
```

```
</RESOURCE>
```

```
</VOTABLE>
```

name: SimulatedSky-3

writer.model.name: STIL

Schema:

```
message table {
  optional float ra;
  optional float dec;
}
```

Row group 0: count: 3 23.33 B records start: 4 total(compressed): 70 B total(uncompressed):70 B

```
-----
  type      encodings count    avg size  nulls  min / max
ra  FLOAT    - -         3      11.67 B   0      "104.37142" / "286.89078"
dec  FLOAT    - -         3      11.67 B   0      "-64.85177" / "29.448921"
```


Participation

Interested parties:

- ESA Science Data Centre
 - ▷ ESDC/DPAC will write Gaia DR4 as parquet files starting Spring 2025, very likely with VOTable metadata
 - ▷ They would like them to be interoperable (IVOA compliant)
 - ▷ See ESA document [Bulk_download_for_DR4+public.pdf](#) Sec 5.1
- TOPCAT/STIL
 - ▷ I would like to read/write interoperable rich metadata in parquet I/O handlers
- Astropy
- Rubin
- IPAC (Firefly, SPHEREx?)
- ... and probably others

Open Questions

- Assuming we do VOTable-in-Parquet:
 - Where does the VOTable go?
 - ▷ In parquet file-level key-value metadata?
 - ▷ Under ARROW:schema metadata?
 - Is `<DATA>` element included, if so what does it look like?
 - What metadata keys must/should be present?
 - ▷ Mandate keys for `version`, `type`, `encoding`, `content` as per GPDF suggestion?
 - ▷ Use `IVOA.VOTable-Parquet.*` naming scheme as per GPDF suggestion?
 - What to say about Parquet/VOTable data model mismatches?
 - ▷ Parquet can contain columns that VOTable cannot, e.g. structured data, unsigned integers, ...
 - Do we standardise Parquet-in-VOTable as well as VOTable-in-Parquet?
 - Others?

Suggested Next Steps

- Decide on (at least one) variant approach and detail
- Make sure we have interoperable implementations (we already do?)
- Write a document:
 - Recommendation track document within Apps WG?
 - Volunteer to write first draft: me?
 - Volunteer co-authors?
 - Content:
 - ▷ Basic prescription is not very complicated
 - ▷ Some consideration required of VOTable/Parquet metadata model mismatch
 - ▷ Possibility to make it more general:
 - Title:** VOTable metadata for metadata-poor formats
 - Section:** General approach
 - Section:** Application to Parquet
 - Section:** Application to FITS (=FITS-plus)
- Timeline:
 - ▷ Draft with essential details very quickly (end 2024?)
 - ▷ ESDC/others can then write files that *should* be compliant ASAP
 - ▷ Work on details/wording/standards process in following months

⇒ Present plans in Malta