Columnar Data Products

Feb 2020 DMLT

Session charge:

- “We should be clear on our overall strategy for Parquet data products, including:
  - Are we committed to support Parquet (or more generally a columnar data format) as a user facing format for LSST catalog data products.
  - if so, how do we slice/tile the data within the files?
  - How do we make these available? Bulk download? By sky region?
  - What is the strategy on using catalog data in Parquet files for backup or disaster recovery.
  - Who controls the schema for Parquet data products?
  - Who validates the generated data against the schema?
- We should also decide which documents, and how, need to be updated to reflect the decisions taken above.”

Are we committed to support Parquet (or more generally a columnar data format) as a user facing format for LSST catalog data products?

- “On the path” to making that commitment (RFC-662), but not formal yet.
- The data would be useful and we are already using columnar formats (caveats follow in next slides).

- Disk space is one reason why we would not want to do this; but if we can afford the space then we should provide it.

Next questions are closely linked:

- If [we provide parquet], how do we slice/tile the data within the files?
- How do we make these available? Bulk download? By sky region?
- What is the strategy on using catalog data in Parquet files for backup or disaster recovery.

Route #1 - spatially organized like Qserv

- Pipelines produces Science Data Model products in parquet format, organized natively to Pipeline's processing in tracts and patches.
- DAX vacuums these up with the Qserv partitioning tool, which reorganizes them into "chunks" and chunk-overlap regions.
- The chunks are then stored in new parquet files, and in qserv.
- In a node-lost scenario, the Qserv replication service identifies the lost chunks, grabs the matching parquet chunk files, and re-loads them into a new database server.

Route #2 - spatially organized like Pipelines

- Pipelines produces Science Data Model products in parquet format, organized natively to Pipeline's processing in tracts and patches.
- DAX vacuums these up with the Qserv partitioning tool, which reorganizes them into "chunks" and chunk-overlap regions.
- The chunks (partitioned according to Qserv's scheme) are loaded into the database workers, but the original Pipelines parquet files remain and are served to users.
- In a node-lost scenario, the Qserv replication service identifies the lost chunks, does some sort of search for the tracts and patches containing the data that belong in that chunk, finds the (several) corresponding parquet files, and reconstitutes the missing chunks/chunk overlaps on the DB workers. Potentially complicated.

- Question boils down to: do we provide parquet with Pipelines-like organization or DB-like organization.
- Ideally, the user doesn’t need to know much about tracts, and only in rare cases needs to know about chunks. Neither seem *obviously* preferred from that perspective.
- For butler-based access to images and catalogs, having correspondence between coadd organization and SDM parquet files makes sense.
- We have not evaluated how hard the single-node recovery process would be from non-chunk-partitioned files. Ingest work so far is in the other direction (I have these input files, put their data in the right chunks)

Other organization questions

- Other non-spatial vertical or horizontal partitions: does one have to access Object in its entirety, or are there useful “convenience” subsets of columns/rows that could stand on their own? (Like SDSS “PhotoObjAll” vs “Star” and “Galaxy” views)
- Technical details to be sorted out on internal parquet block sizes (DAX has experience and will provide recommendations). There are also more public-facing aspects like NULL-handling, data types, sentinel values/columns.

- Who controls the schema for Parquet data products?
  - The DM Subsystem Scientist (until otherwise specified, one should assume that the LDM-153 schema applies, so DM-CCB, etc.).
  - The goal should be to keep the Parquet files as similar as possible to the database contents; how well we can accomplish that in practice is TBD.
  - Re: prior slide, will need to standardize translations.

- Who validates the generated data against the schema?
  - I'd argue this depends on how we answer the "Qserv-like" or "Pipelines-like" question. Whoever partitions the files does the validation.

“How do we make these available?”

- This question is pertinent to all “bulk” products. How does someone retrieve a bunch of (non-cutout) images? Or catalog FITS files? Requirements for parquet are not particularly unique.

- Files-on-GPFS would be moderately acceptable, but limits remote access.
- IMO a better overall solution would be serving files over HTTP (s3-compatible, Minio?)

- Also opinion: we always get questions about bulk download; we should address what users want as “bulk download” and let “MOU-based transfer” be a separate problem.