

A Column-based Data Explorer

Clive Page

IoA, Cambridge

2004 November 18

Growth in Data Volume

- Important new surveys (WFCAM, VISTA)
- Large-format instruments (OmegaCAM, MegaCAM)
- High-time resolution data (SWIFT, SuperWASP)
- Many new data sources: XMM-Newton, INTEGRAL, eMERLIN, GAIA...

Data mining depends mostly on tabular data, especially source catalogues.

At present largest source catalogues (around a billion rows long) mostly come from old technology - scans of photographic plates dating back to the 1950s.

Source Catalogues not only longer also wider

SDSS DR3 has about 1 Terabyte of tabular data.

Table	Number of Columns
USNO-B catalog	50
2MASS point sources	61
1XMM source catalogue	379
SDSS DR3 PhotObjAll	446

Can we keep up with Moore's Law?

- Astronomical data volume – doubling in ~2 years.
- Processor power – doubling every 2 years or less.
- Disc storage per unit cost – doubling every 2 years or less.

So we can process and store the data. But:

- I/O bandwidth and seek times are improving much more slowly (~10% per year)

Conclusion: avoid I/O if at all possible.

What have the RDBMS ever done for us?

- Can handle large datasets, well beyond the 2 GB boundary of 32-bit filing systems.
- Can use indices for fast searching (find any row in ~30 ms)
- Have SQL, which is a powerful query language.
- Can handle null (missing) values properly (3-way logic)
- Mature technology with reliable software products.
- Open Source products like MySQL and Postgres are generally “good enough” for data archives.
- **But: sequential scans are slow, because tables are stored row-wise.**

What should a data explorer package support?

- **Basic database operations:** select subsets, compute new columns, sort, group, equi-joins with other tables.
- **Statistical operations:** find means, medians and other quantiles, find outliers, compute regressions, etc.
- **Cross-matching:** spatial join - needs spatial index or equivalent.
- **Graphics and visualisation:** histograms, scatterplots, image overlays, density maps, etc. Need fully interactive access, e.g. to allow graphical selection of groups of points.

Queries fall into two types

- Indexed: fast (20 to 30 ms to get selected rows).
- Sequential scans: slow (30 to 60 minutes to scan large tables)

Conclusion: avoid sequential scans if at all possible.

Unfortunately it isn't always possible.

Queries which usually involve a sequential scan:

- CREATE INDEX
- Create/populate a new column, e.g.
 - UPDATE table SET column = expression
- SELECT * FROM table WHERE (bmag-vmag) > 0.5
 - Even if indices exist for both bmag and vmag
- SELECT AVG(col) FROM table
- Finding largest N or smallest N in some column
- Joining one table with another (must scan smaller table)
- SELECT * FROM table WHERE ABS(glat) < 5.0
 - Index selectivity needs to be >1000 to be useful.

Data exploration in practice

- Astronomical data explorations and data mining operations will almost always need to include some sequential scans.
- These inevitably take up nearly all the user's time.
- Aim: speed these up to convert them from **batch** to **interactive** operations.

Three ways to speed up sequential scans

- Column-oriented storage:
 - Most queries only involve a few columns out of many, so greatly reduces I/O.
- Data compression:
 - gain of factor of ~ 2 typically
 - Hard to implement while keeping indexed access.
- Parallel I/O e.g. using Beowulf cluster or similar:
 - Many clusters available to astronomers.
 - Large potential large gains in speed.
 - Not yet exploited in general-purpose data mining.

Column-oriented Tables

- Sybase-IQ best known commercial product from makers of Sybase-ASE relational database. Fast, but
 - No Linux version yet
 - No spatial indexing
 - Very expensive licence costs
- ESO/MIDAS table system
- STSDAS table system (part of IRAF)
- FITS? Tables normally row-orientated, but devised a way to get column-based storage. Prototype tested; snags:
 - cFITSIO library still has some 32-bit limits
 - Files were “standard” in theory but in practice not compatible with other FITS software.

Using HDF5

- **Hierarchical Data Format 5**
 - Comes from NCSA, well-supported library and tools
 - Compatible with Globus, GridFTP
 - Designed for efficient access to big files (no 2 GB limit)
 - Simple API (bindings to C, Fortran90, Java, Python)
 - Can attach unlimited metadata to tables and columns.
- Prototype constructed and benchmarked
 - Used query parser and evaluator originally written for Starlink's CURSA about 1992.
 - Around 10% sample of 2MASS catalogue put into HDF5 column-based format.

Performance Gain

```
mysql> select count(phi_opt),min(phi_opt),max(phi_opt),avg(phi_opt) from  
twomass;
```

```
+-----+-----+-----+-----+  
| count(phi_opt) | min(phi_opt) | max(phi_opt) | avg(phi_opt) |  
+-----+-----+-----+-----+  
|          33971487 |          0 |          360 |    195.9702 |  
+-----+-----+-----+-----+
```

```
1 row in set (4 min 30.73 sec)
```

```
hydra:~/hdf> hstats phi_opt
```

```
--Column--      points minimum      maximum      average  
    phi_opt  33971487  0.0000      360.00      195.97
```

```
8.37 seconds
```

32 times faster than MySQL, other tests show gains 13 to 400.

```
hydra:~/hdf> hstats phi_opt
```

```
--Column--      points minimum      maximum      average  
    phi_opt  33971487  0.0000      360.00      195.97
```

```
3.51 seconds
```

What is needed to build an HDF5-based data explorer?

- HDF5 library – reliable, efficient, easy to use.
- Query parser and evaluator – CURSA, or Java library
- Graphics package – great variety of them available
- Statistics routines – many available
- Indexing – many B-tree libraries available
- Spatial indexing – free R-tree code exists, or use pixel-code methods (based on HTM or HEALpix)
- Integration with VO for authentication, MySPACE, etc.
- Provide as stand-alone package, and as a Web Service.
- User interface – allowing iterative filtering, undo, etc.
 - Probably the most difficult part of the design.

Additional Advantages

- Makes distributed cross-match more feasible: only transfer positional columns from one site to another.
- Users with their own data mining code (clustering, classification, etc) can read HDF5 datasets directly.
- Data explorer can easily access astronomical tables in other formats such as FITS, VOTable, CSV, or (using JDBC) which reside in other DBMS.
- Can provide user with integrated graphics, statistics, etc. and a host of other desirable features missing from commercial RDBMS.

The Data Deluge Revisited

From the introduction to the AstroGrid-1 Grant Application:

A tidal wave of data is approaching astronomy, requiring **radical new approaches to database construction, management, and utilisation.**

Finally:

The HDF5-based data explorer is not intended to replace the existing DBSM-based archives but as a facility which can be run in parallel – all it costs is the extra disc space.