Fig. 1

1. pgSphere News

Markus Demleitner
msdemlei@ari.uni-heidelberg.de

- There's now MOCs in pgSphere
- Long-term pgSphere maintenance

(cf. Fig. 1)

2. MOCs in pgSphere

Since June, there's a pgSphere out that has MOC arithmetic built in:

- Literal: `SELECT smoc '0/3 5/2 6/'` In particular, when parsing the notation of noting the maximal order with an empty pixel list is supported.
- Casting: `smoc(order, point)`, `smoc(order, circle)`, `smoc(order, poly)`. Drawback: currently based on the relatively bad approximations of HEALPix-C++
- Operators `smoc | smoc`, `smoc & smoc` – including aggregate functions `SUM` and `INTERSECT`.
- `smoc_degrade(new_order, smoc)`
- A simple GIN index on moc-valued columns

3. In ADQL

You can play around with these things in TAP on http://dc.g-vo.org/tap. See https://blog.g-vo.org/crazy-shapes-in-tap/ for examples.

We have support for:

- `MOC(geometry)`
- `AREA(MOC)`, `INTERSECTS`, `CONTAINS` between MOCs
- `gavo_mocunion(moc, moc)`, `gavo_mocintersect(moc, moc)` as ADQL UDFs
- The `SUM` aggregate function (no intersects yet)

4. Get it

The MOC-enabled pgSphere is at
https://github.com/credativ/pgsphere/

It will be in Debian starting with bullseye.

A buster backport is at GAVO's Debian repo[1].

It's not yet in github.com/pgsphere.

5. Maintaining pgSphere

The reason it's not in there is that the pgsphere organisation currently is dead.

We really need to change this. Items we ought to address:

- Fixing polygons
- Improving MOC indexing
- Using FX's MOC library as backend
- Tackling planner weaknesses with pgsphere joins
- Probably a lot more

6. A pgSphere user's committee?

What if data centers using pgsphere all nominated one person for the pgsphere organisation?

And promised to at least look at PRs?

We'd still need people to do the actual work (suitable funding programme anyone?)

But once someone does the work, we'll at least get it merged, and perhaps there'd be some sense of ownership.

Volunteers?

[1] `https://soft.g-vo.org/repo`