Accelerating access to data archives with the new version of pgSphere

Markus Nullmeier

Zentrum für Astronomie der Universität Heidelberg
Astronomisches Rechen-Institut

mnullmei@ari.uni.heidelberg.de

Accelerating access to data archives with the new version of pgSphere

Markus Nullmeier

mnullmei@ari.uni.heidelberg.de

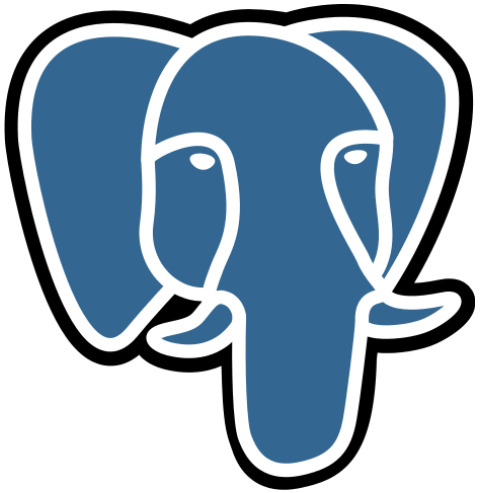
- About pgSphere
- New pgSphere features since 2014
- Extending pgSphere with sky coverage data types

About pgSphere

- pgSphere?

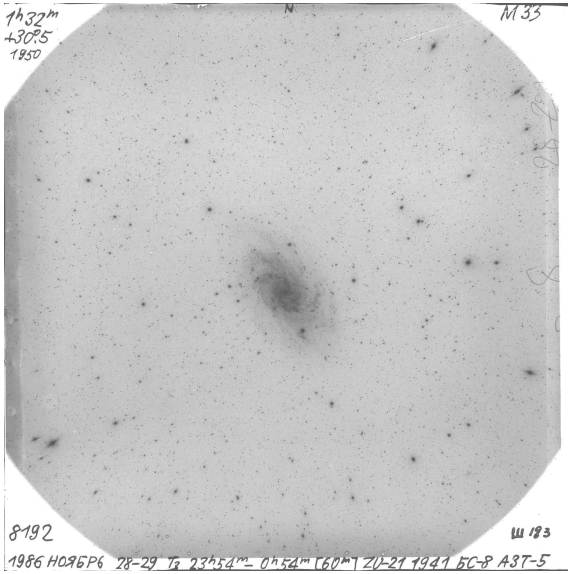
About Pgsphere

- **PostgreSQL extension: new SQL data types, functions, indexes**
- **PostgreSQL: "The world's most advanced open source database"**

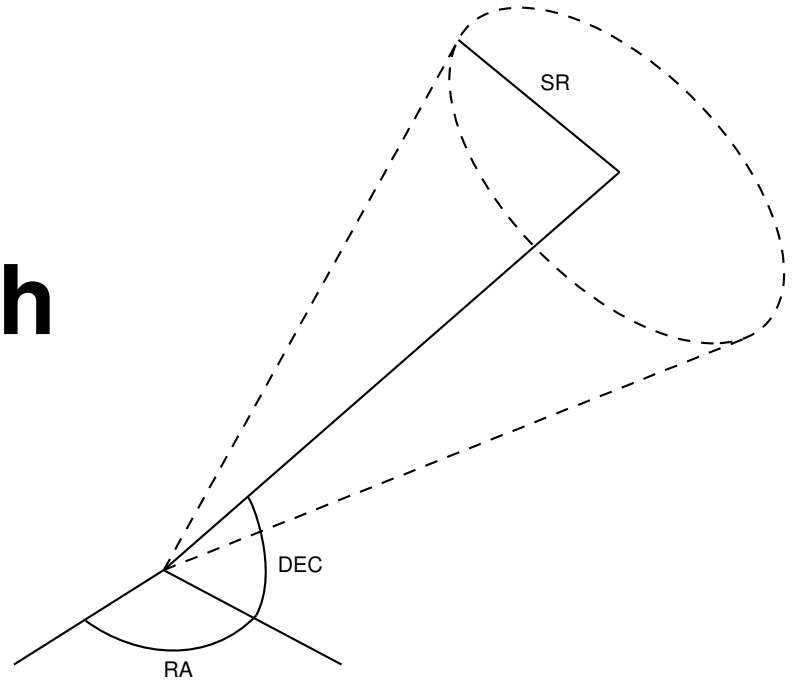


- **SQL data types: spherical points (RA, DEC),**
  spherical lines, polygons, ellipses, paths,
  spherical transformations (rotations)

VO Usage of pgSphere



X-match



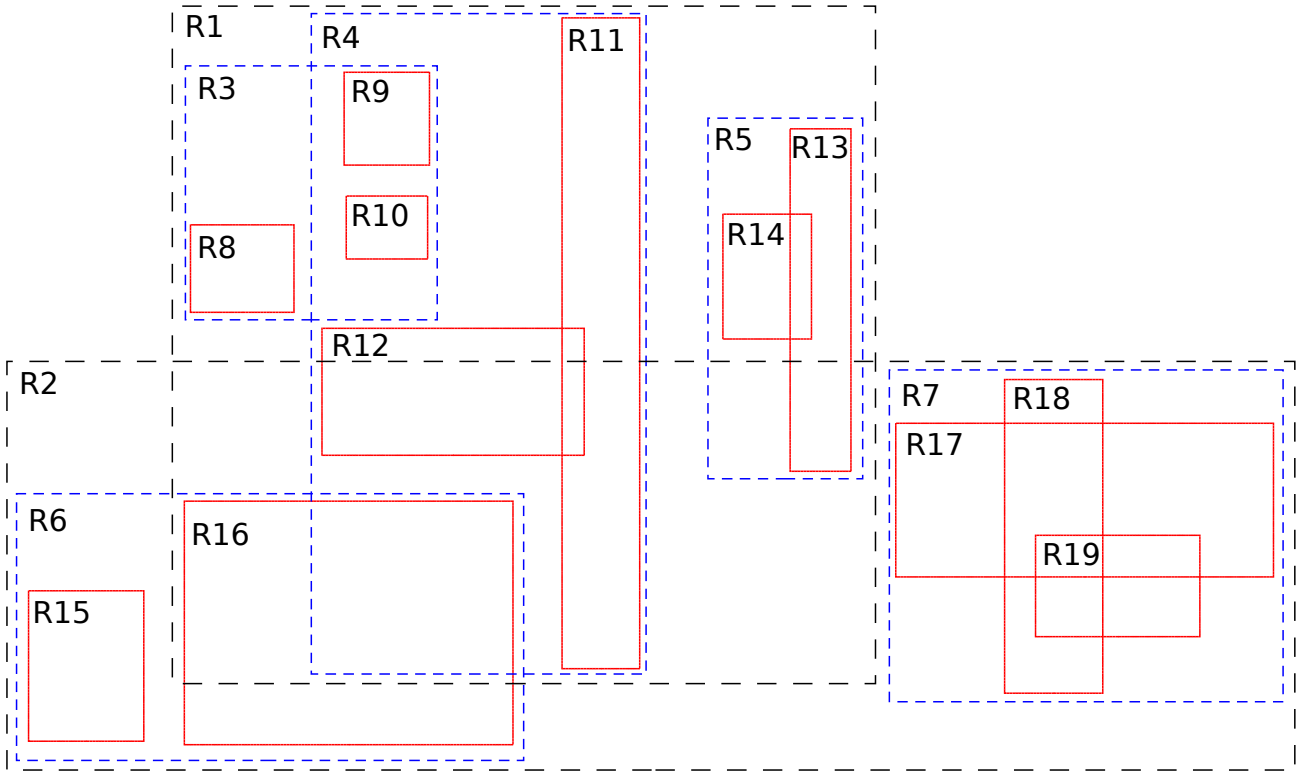
Pgsphere internals

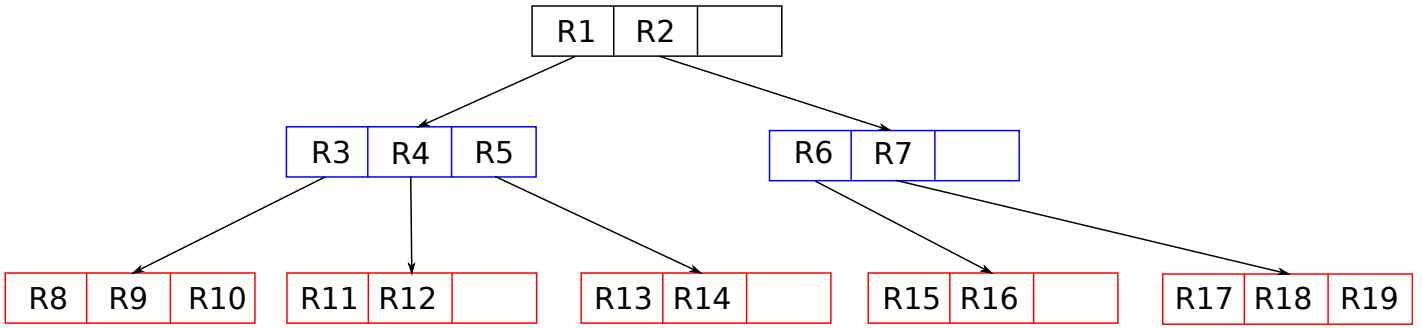
Database indexes of spherical coordinates for, e. g.:

- **Cone search**
- **Cross-match**
- **Images (e. g., digitised astronomical plates)**

Pgsphere internals

R-tree





Pgsphere development history

Janko Richter

Teodor Sigaev Oleg Bartunov

Igor Chilingarian

Pgsphere development nowadays

Dmitry Ivanov

Alexander Korotkov

Markus Nullmeier

contributors: Pat Dowler, Serge Monkewitz

New PgSphere features since 2014

- **Greatly improved R-tree indexing, 1..2 order of magnitude faster:**
  A. Korotkov "A new double sorting-based node splitting algorithm for R-tree", Programming and Computing Software **38(**3), 2012, DOI: 10.1134/S0361768812030024
- **All open / known open bugs fixed**
- **Addition of new-style SQL "contains" operators**
- **More numerical stability**
- **Custom PostgreSQL optimisation for spatial joins (= crossmatch)**

New R-tree indexing

[publication of benchmarks planned for ADASS XXVI, Trieste 2016]

Extending pgSphere with sky coverage data types

MOC = Multi-order coverage (HEALPix Multi-Order Coverage map)

- Concise mapping of a catalog's coverage of the sphere

→ go to the MOC tutorial tomorrow!

- Coverage made up from discrete elements
- ***Making MOC and sky maps a first-class SQL data type...***

WIP: sky coverage data types for pgSphere

MOC as indexable SQL data type

- **I/O to / from files**
- **Create one MOC from table column or query**
- **Specify your own MOC and search over all catalogs of a data center:**

  SELECT name FROM catalogs WHERE my_moc <@ catalogs.moc ;

Sky map data type: analogous to MOC

MOC: indexing

- **R-trees will not work for MOC representing catalogs**
- **PostgreSQL custom indexing will be in Release 9.6: https://github.com/postgrespro/rum**
- **Core of new index structure:**

RANGES OF NUMBERS OF HEALPIX ELEMENTS	SETS OF MOC IDs
range0	{ id7, id11 }
range1	{ id2, id108, id109 }
range2	{ id108, id732, id11030 }
...	...

Your involvement

- **Download, use, test, and join the community at the pgSphere home page:**
  `http://pgsphere.github.io`
- **Send in bug reports**
- **Send in test cases**
- **Send in patches**
- **Send in feature requests :-)**