PostGIS for Managers

pramsey@carto.com

CARTO

Thanks for coming. This is PostGIS for Managers.
FOSS4G does tend to be a pretty technical conference.
The ratio of nerds to suits is very high, so I have to admit I was originally tempted to title this talk

---

PostGIS for Managers!?!?!?!

pramsey@carto.com

CARTO

PostGIS for Managers?!?!?!?
But in fact there's a real constituency for this kind of thing, I think, amongst decision makers who are considering getting their feet wet with open source, or who have started down the road, but are still uncertain about experimenting with something as mission critical as the enterprise database.

CARTO

and I'll let you in on a little secret, at least one of my points will involve recommending engaging companies I've worked for in the past, and talk about how we're using PostGIS in CARTO right now.
so I have decided that,
in order to avoid any appearance of conflict of interest
I will precede any blatantly self-serving advice

All glory to the Hypnotoad!

with an invocation of the hypnotoad.
All glory to the Hypnotoad.

(who is) Paul Ramsey

- PostGIS project co-founder
- PostGIS steering committee chair
- PostGIS developer
- Geography
- Pointcloud
- Linear referencing
- Curvilinear geometry

And since I've just shown the hypnotoad, I might as well cover some self-serving biographical details
...<x>...<x>...<x>
and I'm also an engineer at Carto.
I'm here all week, try the veal.

(who are) You?

- **"Managers"**
- What is this thing?
- Can it do what I need?
- Who else is using it?
- What is the plan of action?

So, as an audience, here for "postgis for managers", what kind of folks are you.
How many of you would describe yourselves as managers?
What kinds of question of questions are important to managers?
Probably not language of implementation,
or SQL analysis tricks, or library dependencies.
More practical things, from an organizational strategy point of view, like...
...<x>...<x>...<x>...<x>
So, to start off, what is this thing?

Or as Gary Coleman might put it,
whatchoo talking bout, Ramsey?
What is this "postgis" thing, anyhow?

PostGIS is a *spatial* database

In IT whiteboard terms, it's the great big cylinder.
It's the place you store your data,
and it's the thing that answers questions about your data, like
"what's in this polygon" or
"how far from the store?"
But perhaps it's better to reason by analogy,

You know, like...

ORACLE is to PostgreSQL

as

ORACLE SPATIAL is to PostGIS

As Oracle is to PostgreSQL,
<x> Oracle Spatial is to PostGIS.
A spatial database is just a database at the heart, but with some extra goodness added for dealing with spatial data and queries.

PostGIS

adds...

- **Types**
  - geometry, geography, raster
- **Functions**
  - ST_Area(), ST_Union(), ST_Buffer()
- **Indexes**
  - R-Tree, GeoHash

So PostGIS takes plain vanilla PostgreSQL and adds,
<x> types, like...
<x> functions, like...
<x> indexes, like...

And once you have these extra pieces in your ordinary database,
these types, functions and indexes,
your database becomes less ordinary, it can do new and wonderful things

---

PostGIS

allows...

- **"GIS in SQL"**
  - answer spatial questions in the database
- **Shared Editing**
  - transactional and data integrity guarantees
- **Performance and Scale**
  - Large datasets, large workloads

like do GIS queries directly in the database
: generate a list of neighbours to notify, calculate the average daily drive distance of the truck, summarize parcel area by zoning code, you name it
<x> unlike GIS files, databases are built to maintain data integrity, under write load from multiple sources, so no passing files around, and more integration with other real-time systems
<x> also unlike files, databases expect to be asked to handle large datasets and large workloads with aplomb

---

so do others...

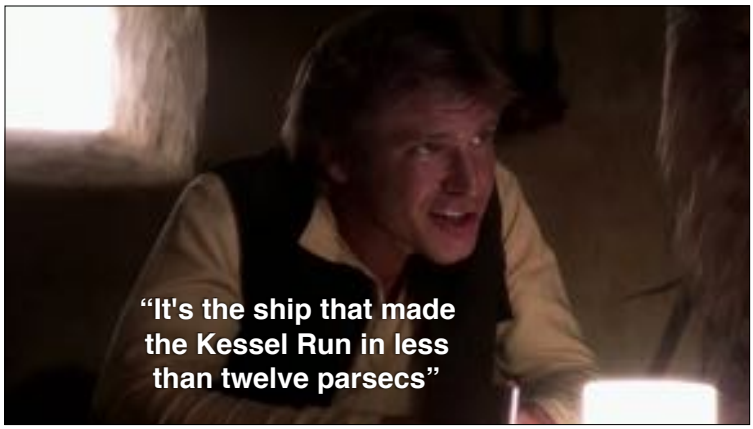
ORACLE is to PostgreSQL

as

ORACLE SPATIAL is to PostGIS

and really, all the awesome things about PostGIS
are also awesome things about other spatial databases too,
they all let you answer GIS questions in the database,
they all break away from the tyranny of files and file-oriented tools, and
they all handle big data sets efficiently.

But is PostGIS in the same league as those other databases,
as the SQL Servers and the Oracles,



Let's talk technology. Let's talk raw capability, let's get our nerdy on.

Rather than just talk about what PostGIS can do, which is a lot,
people find it more convincing when I put it next to the "leading commercial brand",
which is, let's face it, Oracle.

So, here's a quick, but thorough feature comparison,

Core Geometry - 1

ORACLE SPATIAL

- OGC SFSQL 1.2
  point, line, poly, collections
- Curved features
  circular arcs and also **other kinds of curves**

PostGIS

- OGC SFSQL 1.2
  point, line, poly, collections
- Curved features
  circular arcs

Both databases support basic geometry types (points, lines and polygons)
as defined by the Open Geospatial Consortium,
and both support arcs as defined by the ISO,
but Oracle supports some fancy kinds of arcs that CAD folks like,
like splines and clothoids,
while PostGIS stops with basic circular arcs.

Core Geometry - 2

ORACLE SPATIAL

- volumetric support
  "solid"
- "tin" type
- volumetric distance,
  predicates, intersections

PostGIS

- volumetric support
  "polyhedral surface"
- "tin" type
- volumetric distance,
  predicates, intersection

Both databases have similar support for storing and retrieving 3d "volumes".
The databases seem to have come at the problem for different reasons,
the PostGIS support seems to have been motivated a lot by folks with building models and a desire for 3d visualization of those, and
the Oracle support seems to have been motivated by oil and gas and subsurface use cases.
As with the curves, the CAD use case seems to have motivated Oracle development a lot more.

Core Geometry - 3

ORACLE SPATIAL

- X/Y/Z/M on all geometries
- up to 4d r-tree index
- LRS functions for M dimension
- **annotation type**

PostGIS

- X/Y/Z/M on all geometries
- up to 4d r-tree index
- LRS functions for M dimension

Both databases store geometries with more than 2 dimensions,
including a Z dimension and a “measure” or “M” dimension,
and both can index geometries over all four dimensions.
Oracle has a quirky “annotation” type, which again is for CAD program support.

Coordinate Systems

ORACLE SPATIAL

- coordinate transformation in database
- geodetic handled within geometry

PostGIS

- coordinate transformation in database
- geodetic as “geography” type

Both databases can handle geodetic coordinates,
latitudes and longitudes, natively,
though the implementation details are slightly different.
Oracle wrapped their geodetic support into the geometry type.
PostGIS followed the example of SQL Server
and went for a separate geography type, so that planar and spherical calculation would be obviously demarcated by the type in use. It’s not obvious which approach is better…

Coordinate Systems

ORACLE SPATIAL

- geodetic index and calculations
- **geodetic support for “all” functions**

PostGIS

- geodetic index and calculations
- geodetic support limited to: area, length, distance, intersects, dwithin, contains

Both have indexes and functions that support geodetic coordinates,
though Oracle has support for a few more geodetic functions than PostGIS.

Indexing

ORACLE SPATIAL

- 4d r-tree index
- functional spatial indexes

PostGIS

- 4d r-tree index
- **2d r-tree index**
- functional spatial indexes
- **multi-key indexes including spatial term**

Both have spatial indexing, though PostGIS has an extra 2d-only index for higher performance in common cases, and can do multi-key indexes against non-spatial columns, which is a result of the spatial index functionality being more generically integrated into the database.
Multi-key indexes can be handy for use cases where there some strong partitioning in the data, like vehicle tracks, for example.

Functions - 1

ORACLE SPATIAL

- **Controllable tolerance model**
- SFSQL predicates
- Partial relate patterns
- Union aggregate

PostGIS

- Double precision tolerance model
- SFSQL predicates
- **Full relate patterns**
- Union aggregate
- **Array aggregate**

Both databases have a large collection of functions that operate on spatial data.
Oracle's functions have a controllable tolerance model, which is nice.
PostGIS has finer detail on spatial relationships, and some handy array and aggregation support Oracle lacks.
In general PostGIS just has a lot more functions than Oracle, and that advantage in functionality has only grown over the years.

Functions - 2

ORACLE SPATIAL

- Nearest neighbor search
- Minimal shape deconstruction functions

PostGIS

- Nearest neighbor search
- **Deconstruct sub-geoms, rings, point arrays**
- **Clustering functions**
- **Geometry analysis**

Both support high performance nearest-neighbour searches,
PostGIS has better utility functions around geometry manipulation.
PostGIS has also added more and more analytical functions,
for things like spatial clustering and for geometry analysis.

Advanced Features

ORACLE SPATIAL

- Topology
- Point Clouds
- Raster
- GeoCoding
- Routing

PostGIS

- Topology
- Point Clouds
- Raster
- GeoCoding
- Routing

And both have a big collection of features that are related to, or built on top of, or often used with, the core spatial functionality, like topology handling vector coverages like point clouds for lidar data like raster support (for raw storage in Oracle's case and for analytics in PostGIS's case) like geocoding and routing for location based services use cases.

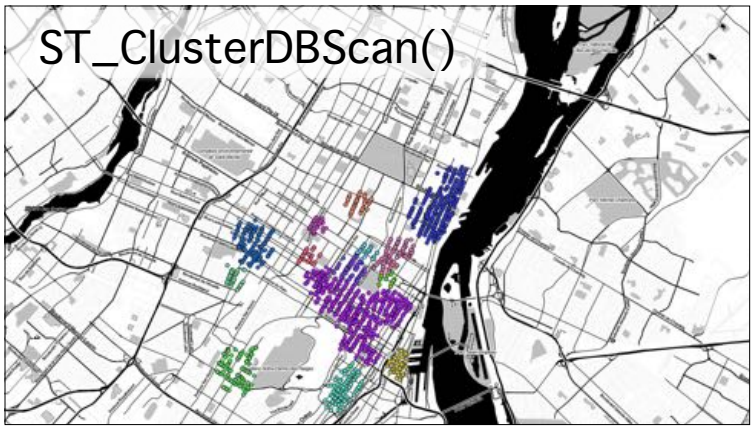
Both databases do an awful lot, and when you add it all up, the bit where Oracle does a little more and the bits where PostGIS does a little more, you have to conclude that,

---

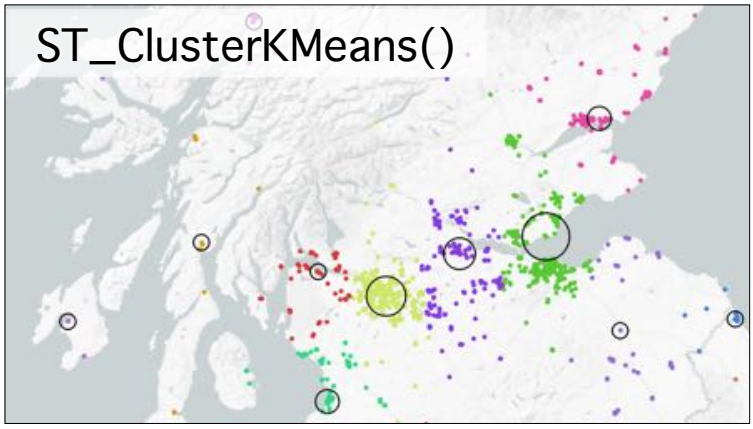
ORACLE SPATIAL ≈ PostGIS

They really aren't all that different. They have much the same functionality, but for some small differences. OK, so PostGIS can do what you need, next question...

OK, yeah, back when I first gave this talk, which was four years ago now, maybe I was willing to be circumspect and say, OK, sure, PostGIS and Oracle, basically similar value propositions. Not really anymore.
Two reasons:
First, PostGIS actually is quite full of wonderful esoteric but useful features now, that Oracle just doesn't have.
Second, integrating PostGIS into the "new GIS" is just way way easier and there's huge value in that.

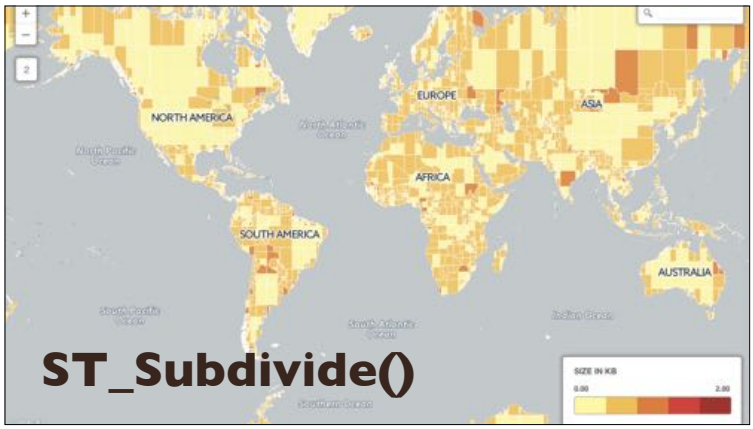


This is hardly a comprehensive review of the crazy things PostGIS does that Oracle Spatial does not, but, how about clustering?
This is a DBSCAN cluster, which doesn't require an initial guess about the number of clusters, but is a little sensitive to tolerance distances.

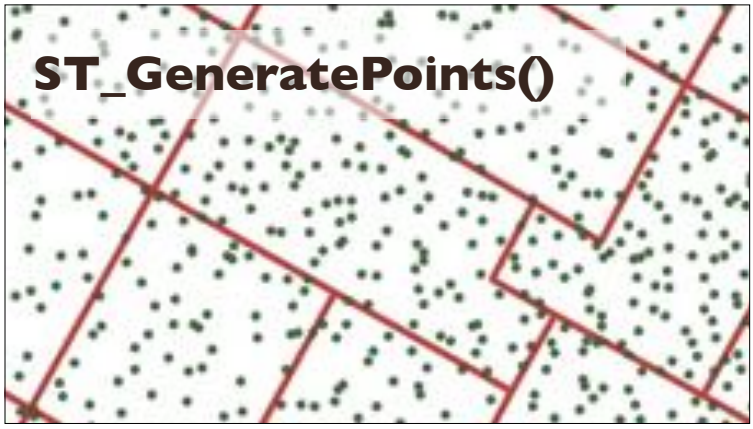


And this is the classic distance-based k-means clustering algorithm. Both the clustering algorithms are predictably a little computationally intensive.

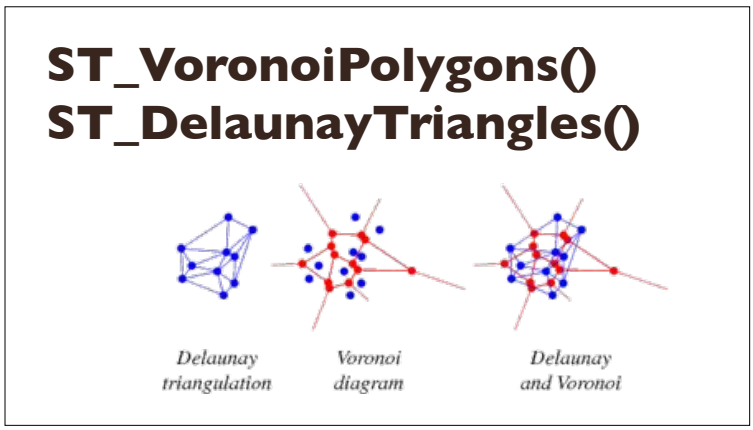
Or how about polygon skeletonization? That's pretty crazy, right? Build the medial axis on the fly?



Or geometry subdivision? Seems kind of useless, But, subdividing your input polygons can make a point-on-poly overlay query 30x faster.
So, actually, that seems kind of **very very useful**!



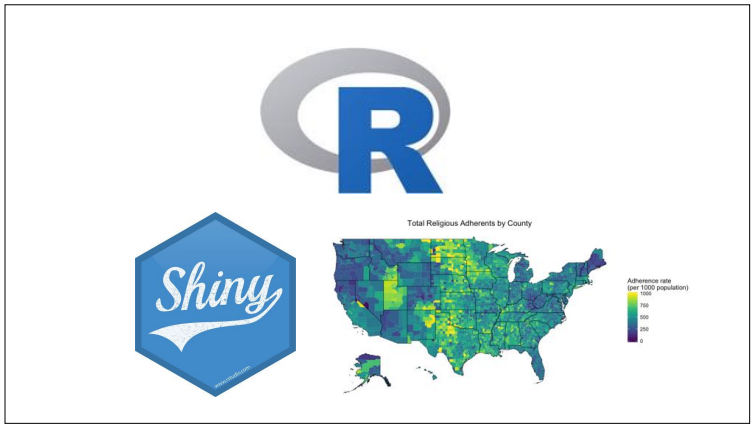
Or converting polygons to point fields, which is handy for generating synthetic populations with census data
or for other data you want to distribute onto another polygon coverage.



Or how about functions for creating a Voronoi diagram?
Or the triangulation instead?
This is all esoteric, but really useful stuff when you're doing true spatial analysis in your database,
which is what you **should be doing,** once you've committed to a spatial database strategy.

Almost **more** important than the discrete extra features PostGIS has,
is the fact that PostgreSQL and PostGIS are about 1M times easier to hook up to the new hotness of geospatial,
and by new hotness, what I mean, is the spatial and statistical analytical tools of Python and R.

So, the new spatial hotness: hopefully you’re seeing more of it at this conference.
Go out, managers, and ask folks:
do you use python?
yes?
what do you use it for? oh, machine learning,
natural language processing, general data cleaning and processing,
hooking up esoteric libraries into our data management chain,
it’s the glue that binds together a million scientific and AI libraries,
so, you know, nothing important.
have you ever connected your python to Postgres? yes?
have you ever connected your python to



And then ask the same questions about R.
Same value proposition, a million statistics routines, amazing plotting and visualization, access to a whole machine learning community around R, spatial statistics in R, and on an on.
Have you connected your R to Postgres?
How about to Oracle?

It’s not that the concepts of connecting R and Python aren’t exactly the same for Oracle as they are from Postgres, it’s just that the practicalities, of getting the right versions of client libraries and putting them in the right place, and typing the right connection info, all that practical stuff, is 1M times harder with

OK. SO.
Who else is using PostGIS?
It's ordinary human nature to feel more comfortable if a decision has been validated by other folks, so let's take a look at other organizations using PostGIS.

CAIXA MasterCard

PostgreSQL Operating at Scale

First, Caixa bank!
With 190 Million customers, they say they "use Postgresql in mission-critical financial environments because it has the quality to support our operations".
Similarly, MasterCard has taken up PostgreSQL, moving from initially using it for web-side work, to now co-processing in multiple data centres.
It's pretty hard to beat a bank or a credit card company, in terms of both scale requirements and reliability requirements, certainly in terms of "enterprise" things like security and reliability.
But, you can beat them in scale, by pointing to one of the start-up users of PostgreSQL, and there have been several.

Instagram

Instagram

PostgreSQL Operating at Scale

Instagram built out their service using PostgreSQL. They currently have around 800M monthly users, which probably means quite a lot of concurrency. It's unlikely there are many organizations with larger scale requirements than Instagram.

Google

- Huge catalogs of images in PostGIS
- Where / what / when queries
- Images stored separately, database holds metadata

Google is a big organization that uses PostGIS, to manage their collection of imagery. Not the images themselves, but the image metadata, which allows them to control image preparation automation, find the “best” images for any particular area, find historical data, and so on.

CARTO

- Large numbers of clients
- SaaS uptime requirements
- Performance to support cartography **and** analysis

I cannot **not** mention Carto, since we’re such a significant deployment of PostGIS now (all hail the hypnotoad).
We have hundreds of instances, running on clouds at Amazon, Google and Azure and also in the data centres of our on-premise customers. Every one of those instances supports multiple users, and workloads with serious performance requirements.

Ordnance Survey

- 500 million feature database (MasterMap)
- Serving live web services (WMS)
- Production service with paying customers
- High availability, twinned servers
- Deployed on AWS

When people ask me what the biggest single dataset being stored in PostGIS is, I usually cite the UK Ordnance Survey, whose MasterMap database is over 500M records in size. There are certainly much larger spatial data sets now.
Things like LIDAR collections, or big feature-extracted data sets maybe, but as a genuine, classic GIS dataset, MasterMap is pretty good and very big.
UK Ordnance Survey uses PostGIS on their web tier, supporting WMS live rendering and deployed on Amazon Web Services, and there are lots of UK MasterMap customers who work with it directly in their own PostGIS instances.
Loading MasterMap into PostGIS is a bit of a UK cottage industry.

Federal Aviation Administration

- aim.faa.gov
- Converted airport management database from Oracle to PostGIS
- Digital NOTAM on PostGIS
- All new AIM projects on PostgreSQL

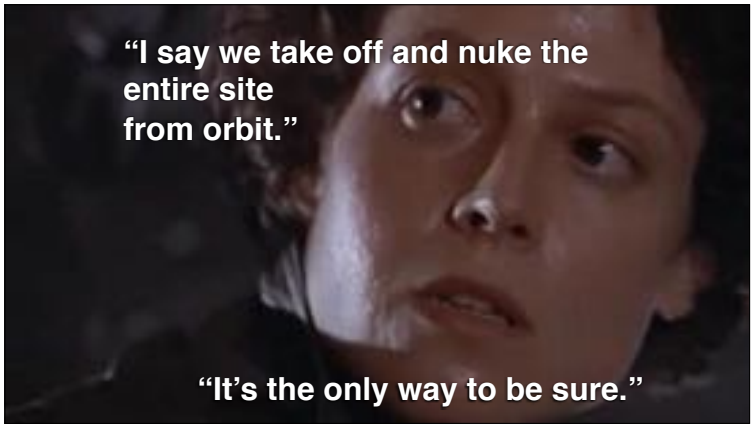
The FAA has moved their airport management data and the notice to aviators databases to PostGIS, and they found the results so good that they set up a policy that all **future** systems migrations are to move away from Oracle to PostgreSQL.

National Mapping Agencies

I already mentioned UK Ordnance Survey, but national mapping agencies in general have moved to PostGIS in a big way, particular in the web publishing space. Generally open source adoption seems to work in from the edges, starting with web publishing workloads and then taking over existing workloads like corporate databases once it has proven itself over a few years of operations.

So lots of organizations have been able to make a change, from proprietary infrastructures to open source infrastructures, from Oracle and SQL Server to PostGIS. What about an organization just getting started, how do

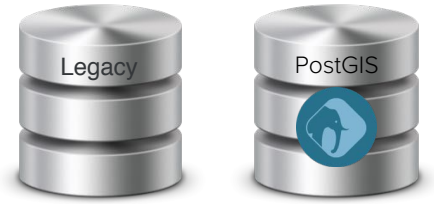
---



One way is to nuke the whole thing from orbit.
It's an emotionally satisfying plan, but it's not realistic.
An organization cannot cut over all its systems in one day.
Day to day operations have to go on, so most staff time will still be devoted to keeping the wheels turning, and only a fraction can be devoted to creating the systems of tomorrow.
What's the solution?

It takes two...

Old New



One thing to get comfortable with right away
is that during transition
there's **going to be two databases** running.
This situation could easily go on for a number of years.
It's not permanent, but it's not entirely temporary either: it's a phase.

Great place to...

- Sense of psychological reality
- Test out existing tools / interoperability
- Build staff expertise
- Build prototype applications
- Migrate first production applications

Standing up a working PostgreSQL/ PostGIS database for your organization is the first step.
It gives you an existence proof. This thing actually runs, under your roof.
It gives you a place to see how well your existing tools work with it.
It gives your staff the hands-on experience they are going to need as you migrate bigger and more complex applications.
It gives you a place to try test migrations of applications.
It gives you a place for migrated applications to live in production.

How to go from...

So, how do you go about turning an Oracle person into a PostgreSQL person.
Is it mission impossible?
Not at all.

Staff upgrading...

Chicago

12 Attendees

A couple years ago I got to go to the PostgresOpen conference in Chicago, and one of the things that struck me was all the folks I met who worked State Farm Insurance. They are based just down-state in Bloomington Illinois, and they were migrating from Oracle to PostgreSQL. So their migration team, Oracle specialists all, was at PostgresOpen, learning new things.

That's some staff upgrading, and it doesn't take much of it.
- most staff recognize that learning new skills increases their value, it also makes work more interesting
- the transition from one database to another doesn't require much new

Staff can learn...

- **DBA skills** are highly transferrable
  - PostgreSQL simpler than Oracle
- **SQL skills** are readily acquired
  - Analysts happy to get "the power"
  - Developers happy to get spatial in familiar tools

So, staff transition is really not a problem.
<x> For the DBAs, the road is very smooth. Their core skills are the same, in many ways PostgreSQL is a much easier beast to manage than Oracle.
<x> For analysts and developers, moving more workflow to the database gives them access to the power of SQL, skills which are also readily acquired, with a few days of course work. It gives the analysts a new power tool, and it gives developers access to spatial functionality without the overhead of learning a new gestalt like GIS.

Take small bites...

So, how to transition? how to eat an elephant?
Take small bits.
- Stand up a parallel server
- Migrate some small applications
- Upgrade staff skills and exercise them on migrations

Keep on moving; once you get momentum, make shutting down the legacy server the goal to motivate continuing migrations.

But what about...

But what about support????

All glory to the Hypnotoad!

I thought you would never ask...

Commercial open source

- Companies exist to provide the services usually provided exclusively by proprietary vendors
- Software support (phone/web)
- Integration (tested builds/full stack)
- Training (administration/development)
- Professional Services (custom help)

Good news, there are actually companies that exist to provide all the things usually provided by proprietary vendors,
<x> the support functions
<x> testing and integration and building of the source code
<x> training services
<x> and professional services
Companies that provide the assurance that the software will be professionally maintained and built for you as a customer, and the insurance that if things DO go wrong, there will be someone to call who can help.

Commercial open source

EDB ENTERPRISEDB

CRUNCHY DATA SOLUTIONS, INC.

2ndQuadrant PostgreSQL

Boundless

I used to work for one of those companies, Boundless, which offers support for a full stack of open source geospatial software. In the PostgreSQL space, there's quite a few substantial support companies who employ core developers and offer support contracts.

This is hardly a unique model. Cloudera fills that role for Hadoop, and Red Hat for Linux, and EnterpriseDB, 2nd Quadrant and Crunch for plain vanilla PostgreSQL.

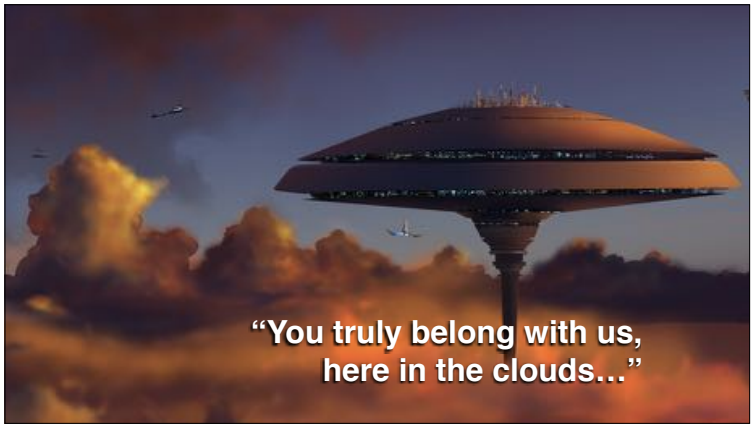
Commercial open source

- Don't pay for sales and marketing
- Don't pay for the software
- Pay for the support

Proprietary software

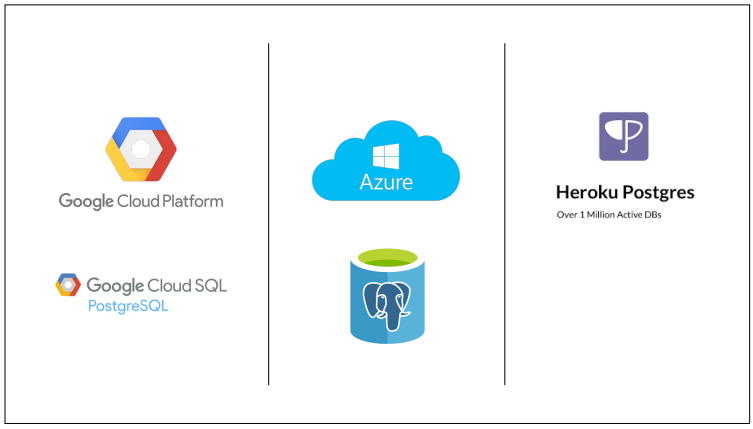
- Sales and marketing are "free"
- Pay (a lot) for the software
- Support is "free" (with your maintenance contract)

I like to present the difference between proprietary software and commercial open source as one that's mostly about moving the point of monetization around. Proprietary software monetizes at the point of acquisition, and moves that money around to other points in the lifecycle.
Some of that's quite effective for them: the sales and marketing effort they can put behind their products is way better than what the open source companies and communities can afford.
On the other hand, actually making more copies of proprietary software is almost free, so it's strange to monetize on a "per copy" basis.
Paying for the thing that actually costs



This is probably the last time I'm going to give this kind of talk,
comparing PostGIS to a product like Oracle,
because Oracle in the old server-in-the-close sense is rapidly becoming irrelevant.
It's getting fairly rare for an organization to commission a new server and install database software on it, the choice of database is much more bound to whatever cloud infrastructure the organization has started migrating to.
So it's worth pointing out, again contra Oracle,
how much **choice** going with a PostgreSQL/PostGIS option provides.

It's really hard to overstate the extent to which the cloud database options, even though they are basically wrappers over existing software, take so much of the administrivia out of running a database. Amazon RDS provides automatic back-ups, automatic replicas, push-button re-provisioning on different servers, storage upgrades and so on, and all on a basically vanilla PostreSQL.
It supports PostGIS, is usually at the latest release, so there's nothing stopping anyone from plugging in a credit card and having a fully functional spatial SQL database in a couple minutes.
Aurora is all that, with some magic sauce underneath. Amazon basically left most



And there's no reason to stick with Amazon. At CARTO we run PostgreSQL/PostGIS infrastructure on all three major clouds.
All three offer fully managed PostgreSQL options, like Amazon RDS.
And the minor clouds, like Heroku and DatabaseLabs, also offer push-button, managed PostgreSQL.

What this says to me,
the fact that PostgreSQL is available by default on all these platforms, and even more, the fact that the one of the top questions in all the FAQ documents in these services, is "do you support PostGIS", and
the fact that the answer is "yes",
what this says to me, is that PostGIS as a spatial database has become the **industry standard** spatial database.
Scratch a new mapping product for database support, and you'll find that PostGIS support is there by default.
Read a performance paper, and the product they compare with is, inevitably, PostGIS. PostGIS is the default choice now, the **industry standard** for new

So, circling back around.
Managers have to make tough decisions, and sometimes make those decisions in a murky informational environment. It’s not easy.
And for something big, like considering new database options,
managers have a few good core questions,
which are important to answer,
and which PostGIS has good answers for.

What is this thing?

What is this thing?
PostGIS is a spatial database,
it allows you to store and query spatial data just like any other data,
using SQL as the query language.
It’s not new and scary, it’s an industry standard now.

Can it do what I need?

ORACLE SPATIAL ≈ PostGIS

Can it do what you need?
Well, It can do basically everything Oracle Spatial can do,
and a good deal more.
Not just data storage, but advanced analysis.
It’s also available on more platforms,
and with more and better cloud options.

Who is using it?

Who is using it?
Lots of big serious organizations,
both public and private,
are using PostGIS for real-world work.

What is the plan?

- Run a second database for a while (it's OK)
- Train up your staff, they already have core skills
- Migrate in increments to learn and gain confidence

What is the plan for transition?
Getting your organization into PostGIS should not a big lift:
- get a development environment up and running
- give your staff some training and learning time
- and migrate your applications a little bit at a time, to learn and grow your skills

What is the alternative?

First Law of Holes

But most importantly, looking into the future,
what is the realistic alternative?
What is the direction for the future?
Is it really a 25 year engagement with Oracle, or Microsoft.
Is **that** where the IT world is moving? No.
The alternative to starting with PostGIS runs afoul of the First Law of Holes, which is
"If you find yourself in a hole, the first thing to do is stop digging".
If you find yourself in a long-term relationship,
with a proprietary vendor,
the first thing to do is find some viable alternatives,
and PostGIS is one of those alternatives.

Questions?

Thanks.

PostGIS for Managers

pramsey@carto.com

CARTO

Thanks, any questions?
