



Image credit: © Duncan Davidson Photography

Remember the database? Now look at the ROI

John Powell of emapsite examines the technical and investment merits of open source database MySQL as a mechanism for truly extending location intelligence across the business enterprise.

Although it is by no means the first database to do so, MySQL has finally implemented the Open Geospatial Consortium's (OGC) spatial standards.

This means that serious "heavy lifting" type analysis often considered core to conventional GIS can now be done at the level of the database.

While MySQL has long supported the OGC-defined datatypes of point, line, polygon, multi-point, multi-line, multi-polygon, and geometry collection, the spatial relation functions such as 'contains' and 'intersects' only worked with minimum bounding rectangles (MBRs).

Now these functions also work with polygonal areas of interest. In addition to functions to test spatial relations, set-based functions such as 'intersection', 'difference' and 'union' also now work with polygonal data.

Other useful GIS functions include 'buffer', 'centroid', 'area', 'perimeter', 'distance' and 'length'.

For example, Figure 1 shows houses in a particular postcode that are deemed to be in a zone of environmental risk. The dashed black lines show CodePoint polygons, the shaded areas different percentage risk probabilities and the red circles the location of the addressable buildings that are within the line delineating amalgamated environmental risk. Such a query involves a spatial intersection of three national datasets, but just one query, and it runs in tests in around 0.03 seconds. Results will of course vary with hardware.

Figures 2 and 3 indicate the relationship between that risk and contours, but plotted against OS MasterMap. The contours are derived using an open source contouring library and Ordnance Survey's Profile height grid. The Profile table contains almost 2.9 billion rows and OS MasterMap 429 million. While combined data and index sizes approach a terabyte of storage, such an image can be generated on the fly and returned to a client, such as a web browser, in a couple of seconds.

Begging the question?

These numbers illustrate two other features of MySQL that will excite developers within and more importantly beyond the GI community: that queries based on spatial indexing are very fast and that MySQL is built for large geospatial datasets. In fact, as illustrated at the recent MySQL conference in California, these numbers are really quite small compared to those of some of the high visibility, high resilience, high volume websites that use MySQL such as Google, Facebook and YouTube.

Established GIS vendors rely on expensive, proprietary database backends and the ongoing maintenance and support revenues therefrom. With many common geospatial queries available on the database and with MySQL designed with a pluggable architecture, developers and integrators now have a number of choices for storage engines depending on the requirements of their applications.

Under the bonnet – something for the developers

The two main engines, InnoDB and MyISAM, both have different strengths and limitations. You can choose which engine suits depending on the requirements of your application. If you have a high insert need and data integrity is vital, you would probably choose InnoDB. If you have a high select (read) requirement, MyISAM may well be the best choice. For data warehouse purposes there is even an archive engine. It is even possible to mix engines in a replication environment, so that the master uses one engine (InnoDB) which has strong support for transactions and data safety, where you do all the inserts and then use another engine (MyISAM) on the slave for fast selects. There may be a slight lag so that the data on the slave are a few seconds behind the master, depending on the volume of inserts on the master, but in



Figure 1

most cases this will not be important. New engines are constantly being added – version 6 will see the Maria engine, which is going to combine the strengths of MyISAM and InnoDB, which means that high-insert spatial tables will get added transactional safety and foreign key support, for example.

And the fourth dimension?

This very value is increasingly seen in exploring and tracking change. Core GIS vendors have traditionally offered quite poor support for temporal data with their proprietary architectures. This does not bode well for the public sector, academics and researchers, the third sector, charities or commercial analysts when considering benchmark applications of our time such as climate change modelling, crime analysis, epidemiology or location based services.

With the recent release of version 5.0, MySQL have introduced partitioning, enabling users to “break” very large data tables into smaller physical chunks while still behaving as one logical table. In other words, queries are written exactly the same as if it were one table. Time can be used as the function to partition the table on, so it is possible to have a different partition for anything from a decade to an hour’s worth of data. So any temporal query need only have to access the relevant partition leading to order of magnitude query speed improvements. It is impossible to give any concrete metrics as it depends on too many factors, but for multi-gigabyte multi-billion row tables, which are increasingly common, the speed increase of selects is likely to be at least doubled.

Should the enterprise be interested?

Well MySQL is open source and totally “free”, though it is of course possible to take out service level agreements and appropriate consulting contracts where required from third parties. Lifecycle costs are lower, investment costs are all but eliminated; in conjunction with high scalability, support for all the main operating systems and the choice of storage engines to suit requirements, spatial databases now make an excellent choice for corporate data centre deployment. With big moves towards cloud computing and virtualisation, it is self evident that MySQL’s capability in this space (especially in comparison to the late arriving SQL Server and behemoth ORACLE) will bring more integrators and developers into the GI domain. I think we can expect MySQL to begin to take an increasing market share. The MySQL toolset represents an exciting proposition at a time of accelerating interest in leveraging the social value of location-centric public and other sector information through powerful mashups that may or may not include map-centric visualisation.

Many companies have tens, hundreds or even thousands of separate GIS (as well as CAD and modelling tools that frequently have geospatial query capability at their heart) deployed as well as disparate data spread around different offices. Most of this data will contain explicit or implicit location related



Figure 2



Figure 3

information be it in coordinates, addresses or some more prosaic form. Beyond “maps” and “drawings” most enterprises are unaware and unable to extract value from such data in part due to its incompatibility with the established GIS-type architectures for handling “spatial”. Corporate data, be it client specific, enterprise specific, third party or subject to supplier licensing, can be “centralised” in a single (potentially virtual distributed) datacentre and replicated to a second instance in a geographically discrete datacentre. MySQL has emerged as a datacentre deployment option for the location-centric enterprise (and that is just about any enterprise) with a considerably lower total cost of ownership for such corporate data. MySQL provides tools and capabilities that bear comparison with that offered by the more costly, niche, less extensible toolsets known today as GIS.

Much GIS analysis could be moved directly to the database, in the form of server applications such as Java servlets and MySQL stored procedures. The results could then be viewed using a thin client such as a web browser or a web delivered application, for example a Java Swing application invoked using Java web start or, if scale distortions are not a concern, Google Earth/maps or Virtual Earth. The advantage with web-delivered applications is that code/application enhancements can be rolled out to all users of a company in distinct geographic locations and new users can be added at essentially no extra cost. The CEO really can be brought into the loop without a visit to the bank, the system administrator or the corporate trainer!

Of course, specialised desktop GIS will always be required for some users, and the approach suggested here will involve hiring database administrators and developers, but it is highly likely that any company with several offices, disparate GIS installations and diverse data holdings will already have many employees looking after the data and GIS. Overall, then, moving to a data storage model that supports analytical code designed for the specific needs of that company delivered via the web is likely to lead to considerable cost savings and performance and productivity enhancements.

John Powell is a GI developer at emapsite. This article reflects the themes of a presentation he gave at the April 2009 MySQL Conference and Expo in Santa Clara, California. For more information, visit www.emapsite.com.