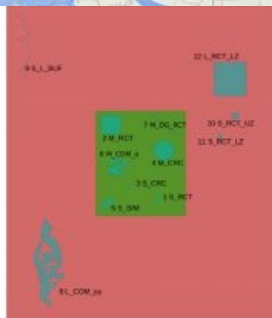


Managing Massive Point Cloud Data: Performance of DBMS and File-based Solutions



Today, Lidar and photogrammetry enable the collection of massive point clouds. Faced with hundreds of billions or even trillions of points, the traditional solutions for handling point clouds usually underperform. To obtain insight into the features affecting performance, tests have been carried out on various systems and the pros and cons have been identified.

(By Oscar Martinez-Rubi, Peter van

Oosterom and Theo Tijssen, The Netherlands)

Point clouds have traditionally been processed into grids, vector objects or other types of data to support further processing in a GIS environment. Today, point clouds are also directly used for estimating volumes of complex objects, visibility analysis, roof solar potential analysis, 3D visualisations and other applications. In archaeology, for example, point clouds are crucial for 3D documentation and analysis of sites. In addition to using data management solutions to manage grids, vectors or TINs, users are increasingly demanding that they can handle massive point clouds. The performances of the various current systems for managing point cloud data were investigated in the 'Massive Point

Clouds for eSciences' project, a collaboration between Rijkswaterstaat, Fugro, Oracle, Netherlands eScience Center and TU Delft.

Systems

Since there is a continuous debate about whether database management systems (DBMSs) are suitable for managing point cloud data, the project considered both DBMS and file-based solutions. In the latter, points are stored in files in a certain format and accessed and processed by solution-specific software. In DBMSs, two storage models can be distinguished:

- Blocks model: nearby points are grouped in blocks which are stored in a database table, one row per block
- Flat table model: points are directly stored in a database table, one row per point, resulting in tables with many rows.

All file-based solutions use a type of blocks model. It was decided to test the widely used LAStools by Rapidlasso with both LAS and compressed LAZ files. The blocks model DBMSs tested were Oracle and PostgreSQL. Flat table model DBMSs in the tests were Oracle, PostgreSQL and MonetDB, which organises data per column instead of using the classic row storage architecture.

Benchmark

Initially the wishes of users in government, industry and academia were inventoried using structured interviews. The highest-ranked features were investigated using datasets varying from a few million points to several hundred billion points. The point clouds were subsets of AHN2, the second National Height Model of the Netherlands, which consists of 640 billion points (Figure 1). All systems run on the same platform, a HP DL380p Gen8 server with 128GB RAM and 2 x 8 Intel Xeon processors E5-2690 at 2.9GHz, RHEL 6 as operative system and different disks directly attached including 400GB SSD, 5TB SAS 15,000rpm in RAID 5 configuration (internal), and 2 x 41TB SATA 7,200rpm in RAID 5 configuration (in Yotta disk cabinet).

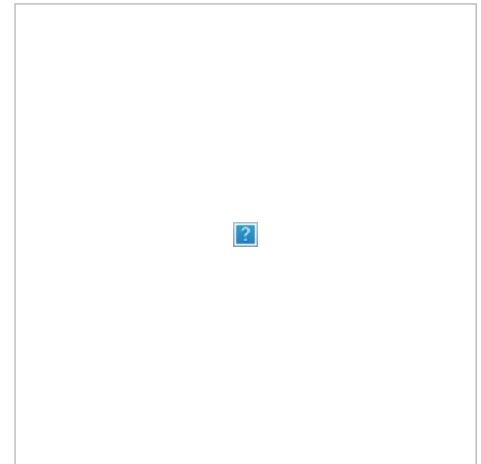
Storage, Preparation and Loading

Compared to flat table systems, the blocks model DBMSs are faster and compress the data better during preparation and loading. Flat table systems enable modifications of the table definition or the data values as in any database table. This is more complicated in the blocks model. For both, the integration with other types of data is straightforward and all the key features of DBMSs are present, i.e. data interface through the SQL language, multi-user access, transaction processing, remote access and advanced security. LAStools prepares data faster than any DBMS since no loading is needed, only resorting and indexing. The storage requirements of the compressed LAZ

format are lower than those of the DBMSs, but with its fixed file format the data model loses flexibility as one is restricted to the specified format. For example, the standard LAS format allows only one byte for user data.

Retrieval

Data retrieval was tested by selecting points within rectangles, circular areas and simple and complex polygons (Figure 2). Also tested were nearest neighbours queries and simple operations such as the computation of minimum, maximum and average elevation in an area. Blocks model DBMSs performed well on larger areas or complex polygons, independent of the point cloud size. However, the blocks model added an overhead which affects simple queries most. The flat table model DBMSs performed well for simple queries on small point clouds, but for large point clouds the native indexing methods became inefficient. Alternative flat table models based on space-filling curves provided nearly constant response times, independent of the stored point cloud size. The file-based solution using LAStools performed best for simple queries. The queries to LAZ data were slower than to LAS data because of the need to uncompress the data. In addition, massive point clouds required an external DBMS to maintain good performance.

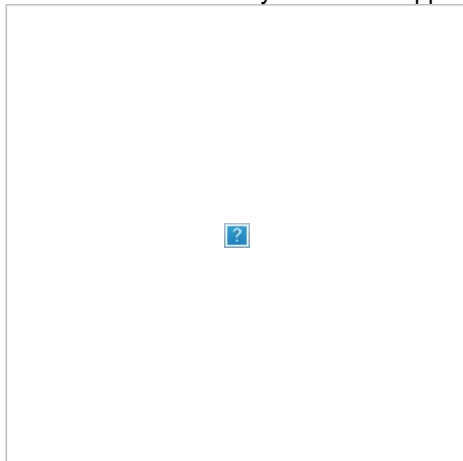


Oracle Exadata

An implementation of the flat table model in Oracle was also tested in Oracle Exadata X4-2 hardware, Oracle SUN hardware designed for the Oracle database with an advanced architecture including hardware hybrid columnar compression (HCC), massive parallel smart scans/predicate filtering and lesser data transfer. Storage requirements, speed of loading and data retrieval were comparable to LAStools but complex queries ran significantly better because of massive parallelisation.

Suggestions for Improvement

If a file-based solution fulfils the user requirements it is recommended to use that solution. However, if more flexibility, other types of (spatial) data and/or more advanced functionality are required, DBMSs are advisable. Point cloud support is steadily improved in most DBMSs and could be further improved by using the PDAL library which provides faster loading with more compressed data as well as faster data retrieval. Most systems miss two important features. Firstly, although data preparation and loading can be easily parallelised with additional tools only MonetDB supports native efficient parallel processing. The performance of DBMSs for which parallel algorithms for data retrieval were explored improved significantly. Oracle is currently adding parallel query support based on similar algorithms. Secondly, crucial for visualisation is support of level of detail, i.e. the ability to display points which are close to the viewer with higher density than those further away. Plas, Potree and other recent web-based frameworks have developed own data structures for visualising point clouds. Figure 3 shows a part of AHN2 visualised by Potree. These frameworks also run into difficulties with massive point clouds, and solutions are currently being sought. The authors are presently exploring alternatives for adding an efficient level of detail support in generic DBMSs. Standardisation of point cloud data at web-service level is the topic of ongoing debate.



Acknowledgements

Thanks are due to all members of the 'Massive Point Clouds for eSciences' project, which is supported in part by the Netherlands eScience Center under project code 027.012.101.

This article, written by Oscar Martinez-Rubi, Peter van Oostrom and Theo Tijssen, was published in the September 2015 issue of *GIM International*.

Figure 1, Areas covered by the four subsets and the number of million points in each area projected on Google Maps

Figure 2, Data retrieved from selected rectangles, circles and simple and complex polygons.

Figure 3, Visualisation of a small part of AHN2, representing the city of Delft using Potree; the colours represent elevation rather than strength of the reflected pulse, which is not present in AHN2.

Further Reading

Suijker, P., Alkemade, I., Kodde, M., Nonhebel, A. (2014) User requirements Massive Point Clouds for eSciences (WP1), Internal Report, 2014-04-25. <http://repository.tudelft.nl/view/ir/uuid%3A351e0d1e-f473-4651-bf15-8f9b29b7b800>.

Van Oosterom, P., Martinez-Rubi, O., Ivanova, M., Horhammer, M., Geringer, D., Ravada, S. Tijssen, T., Kodde, M., Gonçalves, R. (2015) Massive point cloud data management: Design, implementation and execution of a point cloud benchmark, *Computers & Graphics* 2015. <http://www.sciencedirect.com/science/article/pii/S0097849315000084>.

Martinez-Rubi, O., van Oosterom, P., Gonçalves, R., Tijssen, T., Ivanova, M., Kersten, M., Alvanaki, F. (2014) Benchmarking and Improving Point Cloud Data Management in MonetDB. *SIGSPATIAL Special* (ISSN 1946-7729) Volume 6, Number 2, July 2014. <http://www.sigspatial.org/sigspatial-special-issues/SigspatialSpecialJuly2014.pdf>.

Biographies of the Authors

Oscar Martinez-Rubi studied telecommunications and received an MSc in Astrophysics in 2009 from University of Barcelona (UB), Spain. After spending some time at UB and University of Groningen, he has been working at Netherlands eScience Center since 2013.

Peter van Oosterom holds an MSc in Computer Science from Delft University of Technology (TU Delft) and received a PhD from Leiden University in 1990. He was with TNO and the Dutch Cadastre before becoming professor of GIS Technology at TU Delft in 2000.

Theo Tijssen received an MSc in Human Geography from Utrecht University in 1985. He now works at TU Delft; having initially joined in the Department of Geodesy, he has been assistant professor of GIS Technology since 2003.

<https://www.gim-international.com/content/article/managing-massive-point-clouds>
