Processing Mass Data for a Nationwide Web 3D Service

Martin Over, Arne Schilling, Steffen Neubauer, Sandra Lanig, Alexander Zipf
Research Group Cartography, Department of Geography, University of Bonn
Meckenheimer Allee 172, 53115 Bonn, Germany
http://wwww.geographie.uni-bonn.de/karto/

1 INTRODUCTION

In this paper the lessons learned when dealing with high performance computing systems for processing high volume spatial data are presented. The data were used to develop the first nationwide Web 3D Service (W3DS – OGC, 2005) for Germany based on Open Street Map (OSM) data and height data of the Shuttle Radar Topography Mission (SRTM). The resulting service is available at www.osm-3d.org.

There are two ways to display the OSM Data in 3D. It can be done by mapping raster images on a Digital Elevation Mode (DEM) or by combining OSM vector data and the SRTM height data in an integrated Triangulated Irregular Network (TIN) (figure 1).

Figure 1: Visualization of the integrated DEM by our web client.

This paper deals with the problem how to realize the very resource intensive integration of the 2D vector data in the DEM triangulation. For these purpose two different high performance clusters (HPC) were tested (located at the University of Bonn (HPC Bonn) and the University of Aachen (HPC Aachen)). The total processing time on the clusters was the main object of investigation.

2 DEM PREPROCESSING

For yielding a better performance when streaming the DEM from server to client the integrated TIN had to be computed at different Levels of Detail (LODs). The complex integration of the vector data in the triangulation of the DEM data is described in Schilling et al. (2007, 2008). The following four approaches were used to reduce the amount of information (for values see table 1.).

1. *Layer selection:* Manual selection of the required 2D layers for each LOD.
2. *Mesh simplification algorithm:* Edge contraction algorithm (Garland and Heckbert, 1997).
3. *Egde factor*: Large penalty factor in order to preserve the boundaries of the 2D layers (Garland and Heckbert, 1997).
4. *Layer Generalisation*: The Douglas-Peucker (1973) algorithm was used.

LOD tile size in meter	2D layers	Simplification error	Edge factor	Generalization threshold	Size in Database in MB
500	105	1	4	-	14000
1000	69	4	4	-	5977
2000	63	16	4	-	3196
4000	57	64	4	-	1552
8000	45	256	4	-	829
16000	10	16384	100	50	250
32000	10	65536	90	150	143
64000	6	5636096	2500	350	32

Table 1: Overview of the LODs, data reduction factors and the total size of the LODs in the database.

3 HIGH PERFORMANCE COMPUTING SYSTEMS

Table 2 presents the basic parameters of the two HPC systems that are used for the preprocessing.

Name:	**HPC Aachen**	**HPC Bonn**
System:	Windows HPC Server 2008	LINUX Torque 2.2.1 - pbs_shed
Prozessor Type:	Xeon E5450	Opteron 2220
Cores per node:	8	4
CPU-speed:	3.000MHz	2.800MHz

Table 2: Basic parameters of the HPC Aachen and the HPC Bonn.

4 RESULTS

A comparison of the processing results for the two HPC systems is shown in table 3 below.

	500	**1000**	**2000**	**4000**	**8000**	**16000**	**32000**	**64000**
HPC Bonn -Tile 19	**592**	144	**122**	**37**	33	251	1880	1990
HPC Bonn -Tile 20	**465**	202	73	**50**	**69**	**129**	360	1140
HPC Bonn -Tile 21	**294**	90	47	47	**52**	60	212	1808
HPC Aachen -Tile 19	1895	**115**	928	202	**15**	**21**	**58**	**117**
HPC Aachen -Tile 20	760	**114**	**70**	94	84	647	**62**	**112**
HPC Aachen -Tile 21	551	**64**	**24**	**11**	117	**11**	**33**	**30**

Table 3: Time period of the processing done with the HPC Aachen and the HPC Bonn in minutes for the LODs 500 – 64000. The shorter processing time periods are highlighted in bold numbers.

Contrary to the prospects the processing at the HPC Aachen takes at average about two times longer than HPC Bonn for the LOD 500. Reason for this is the huge number of files written on the HPC Aachen file server.

The file server problem is not only resulting in a longer processing time, but also in several cases it was not possible to write some files on the file server. The lesson learned indicates the importance to implement extensive error handling functions regarding file writing.

For the processing of LODs 16000 - 64000 also a Douglas-Peucker (1973) generalization was applied to reduce the amount of data and the computation got more complex. In this case the HPC Aachen was faster because of the higher computing power and less operational demands on the file server.

5 CONCLUSION

In general both HPC Systems are not designed for massive file writing. When dealing with a huge amount of files be aware of implementing functions in your programs to avoid the problem of unreachable file servers.

The question how long it takes to process the data cannot fully be answered. The preprocessing period depends on the traffic on the file server and therefore not controllable by ourselves. Multiple users are the usual case when using clusters and a controlled environment is not given.

The adaptation of the jobs to the skills of all components of the two high performance clusters will result in an at average approximately 2,5 times faster processing .

BIBLIOGRAPHY

Douglas, D. and Peucker T. (1973) Algorithms for the reduction of the number of points required to represent a digitised line or its caricature, in: The Canadian Cartographer, Vol 10, pp. 112-122.

Garland, M. and Heckbert, P. (1997) Surface simplification using quadric error metrics, in: Computer Graphics (SIGGRAPH '97) Proceedings, pp. 209-216.

Open Geospatial Consortium (OGC) (2005) Web 3D Service, Discussion paper, Ref No. OGC 05-019.

Schilling, A., Basanow, J. and Zipf, A. (2007) VECTOR BASED MAPPING OF POLYGONS ON IRREGULAR TERRAIN MESHES FOR WEB 3D MAP SERVICES, 3rd International Conference on Web Information Systems and Technologies (WEBIST), Barcelona, Spain.

Schilling, A., Lanig, S., Neis, P. and Zipf, A. (2008) Integrating Terrain Surface and Street Network for 3D Routing, 3D Geoinfo 08, 3rd International Workshop on 3D Geo-Information, Seoul, South Korea.