Parallel GIS Processing
Thesis Proposal

Nathan Kerr

May 2009

1 Introduction

Geospatial (GIS) simulation, analysis, and simulation are important processes to understanding and improving our environment, both urban and natural. Geospatial data is made up of a georefrenced geometry such as a point or a polygon at a certain logitude, latitude, and altitude with related descriptive information such as a land use type. GIS data can be used to represent a large range of real-world objects such as road or power networks, building locations, or natural features such as lakes and rivers. Utilizing GIS data is one method toward processing, analyzing, and simulating real-world systems. One consumer application of GIS are the GPS based car navigation systems common today.

Larger scale GIS applications also exist in areas such as city planning. City planners use GIS to study road networks, zoning issues, and to simulate population growth. As cities and metropolises grow, the amount of data required to represent these areas also increases. As the data increases, so does the processing power required to complete the geospatial analysis, processing, and simulation.

Desktop GIS packages such as ArcGIS[2], QuantumGIS[3], and GRASS GIS[4] are commonly used for GIS processing and analysis. While these programs provide graphical interfaces to their GIS capabilities, their capabilites are limited by the computers they run on. Datasets can be too large for their memories and computations can take too long to be practical. The popular simulation package, UrbanSim[5] faces these same constraints.

An alternative approach to using these desktop programs is to employ a geospatial databases like PostGIS[6]. Geospatial databases allow centralized access to, and processing of, geospatial data through query languages such as SQL. As the data is stored by and managed by the database software, advanced database features such as indexes can be utilized to speedup data access and processing.

PostGIS is utilized as the core component of the Urban Systems Framework[**?**] (USF) designed by the Digital Phoenix[12] project group at Arizona State University. Digital Phoenix tries to integrate 3D visualization technology with simulated and gathered GIS data to better understands impacts of planning decisions.

GIS data has become easier to gather in recent years with the proliferation on low cost GPS devices. The availability of these devices significantly reduces the cost of data collection, moving it from a government funded service to the capabilities of private companies and individuals even to the point of open source style maps such as OpenStreetMap[?]. With the reduced cost of data collection, the amount of data has grown significantly. As datasets grow and the associated processing becomes more sophisticated, the abilities of programs that only work on a single computer are felt through long processing times and inability to work with the required data. Thus a method of utilizing multiple computers to complete the required processing is needed to increase the size of the data that can be processed and reduce the time required to complete the processing.

In 1994 David DeWitt and Jim Gray stated that parallel databases were the future for high performance databases [?]. Parallel databases should be able to spread both data storage and query execution across multiple computers without complicating the SQL writing process. This ability would give the SQL query writer access to execution on more computers at a time, thus decreasing time to solution, while not increasing the complexity of the researcher's query.

While commercial parallel databases such as TeraData [7] and Oracle [8] do exist today, they come with a hefty price tag that seems to overwhelm what an individual researcher or small research group can afford. Adding to the price of the database software and the associated support fees, these systems often need specialized knowledge to get, and keep, running.

Many universities and research institutions already have investment in compute clusters. These clusters are groups of computer linked together with high speed network interconnects with the hardware and software already being managed by competent staffs. Of the capabilities of these high performance computing centers is available for the cost of compute cycles.

If sufficient software was available to enable GIS processing on these machines, then researchers utilizing GIS would be able to make use of high performance computing to speedup processing and computation time while enabling larger datasets at minimal cost.

2 Thesis Work

I intend to demonstrate two different approaches to executing GIS queries on high performance computing clusters. These approaches will be compared with the database driven SQL query approach in terms of ease of programming and execution time.

The first approach will utilize the standard means of programming for high performance computing clusters, MPI [**?**]. The second will use the open source version of Google's MapReduce [**?**], Hadoop [9].

A set of sample queries will be taken from work encountered with the Digital Phoenix project and with other GIS research as available.

3 Related Work

The Open Geospatial Consortium has created several standards for capabilities of GIS systems including interoperability. Among these standards are the set of Simple Features standards which define data types and operations for GIS data. This research makes use of several implementations of libraries supporting these standards. This research does not intend to create this functionality, but instead consume it. The libraries used are the Java Topology Suite (JTS) [10] and the GEOS library [11]. GEOS is a port of the JTS and was developed as part of the PostGIS extensions for PostgreSQL.

The collaboration capabilities of GRASS GIS have been used to distribute sub-queries among computers. The very existence of this paper shows the lack of penetration of parallel GIS capable databases. Because the databases aren't known about or are too expensive to be utilized, other solutions are being sought. The method described in this paper utilizes multiple instances of GRASS in a master-slave configuration where all participants access a shared data repository or filesystem. The geometries are portioned between the various nodes. Operations are done on the subsets, and the results are merged to produce the final result.[13] This methodology is very much like Map/Reduce.

4 Value and Benefits of Research

While any sort of programming or query model moves away from the seeming simplicity of graphical, clickable, user interfaces, any form of script is easy to save for later replication and thus traceability of the research done. In addition, complicated tasks are more easily repeatable.

The benefits of a programming or query model come with the definite tradeoff of a seemingly increased learning curve for the researchers. With the right model, these approaches will be no harder than writing simple SQL statements, while proving reduced time to solution on inexpensive and more available resources than the commercial parallel databases offer.

5 Research Plan

The research will progress on two interrelated paths. The first path is determining a sufficient set of GIS operations to validate a processing environment on real world operations. The second path will be developing the MPI and Hadoop environments to execute the set of operations determined. All operations will also be executed using PostGIS to compare for query development and execution times.

An approximate timeline for these activities follows:

1. Fall 2008: Definition of GIS queries

2. Fall 2008 - Early Spring 2009: Development and execution of queries

in PostGIS

3. February - April 2009: Development of queries in MPI

4. April 2009 - May 2009: Development of queries in Hadoop

5. June - July 2009: Analysis and Evaluation

6 Completion Criteria

1. A set of GIS operations sufficient to validate an environment against real world operation

2. Working implementations for each operation in each environment.

3. Comparisons of the ease of programability and execution times for each environment

References

[1] http://postgresql.org/

[2] http://www.esri.com/software/arcgis/

[3] http://www.qgis.org/

[4] http://grass.itc.it/

[5] http://urbansim.org/

[6] http://postgis.refractions.net/

[7] http://teradata.com/

[8] http://www.oracle.com/

[9] http://hadoop.apache.org/

[10] http://www.vividsolutions.com/jts/jtshome.htm

[11] http://trac.osgeo.org/geos/

[12] http://digitalphoenix-asu.net/

[13] Fang Huang; Dingsheng Liu; Peng Liu; Shaogang Wang; Yi Zeng; Guoqing Li; Wenyang Yu; Jian Wang; Lingjun Zhao; Lv Pang, ”Research On Cluster-Based Parallel GIS with the Example of Parallelization on GRASS GIS,” Grid and Cooperative Computing, 2007. GCC 2007. Sixth International Conference on , vol., no., pp.642-649, 16-18 Aug. 2007

[14] Jeffrey Dean; Sanjay Ghemawat, ”MapReduce: Simplified Data Processing on Large Clusters,” OSDI 2004.

[15] http://www.mpi-forum.org/

[16] DeWitt, D. and Gray, J. 1992. Parallel database systems: the future of high performance database systems. Commun. ACM 35, 6 (Jun. 1992), 85-98. DOI= http://doi.acm.org.ezproxy1.lib.asu.edu/10.1145/129888.129894

[17] Robert Pahle and Nathan Kerr. 2009. A data centric framework for research in planning. UPE8.

[18] http://www.openstreetmap.org