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.

GIS processing and analysis is often accomplished through desktop applications such as ArcGIS [2], QuantumGIS [3], and GRASS GIS [4]. Simulation is often done with the popular UrbanSim [5] package. These programs provide graphical, clickable interfaces that allow for relatively easy processing of data.

An alternative approach to using these desktop programs is to employ a geospatial databases like PostGIS [6]. In addition to performing the required computation, databases allow collaboration and organization through multi-user access controls. PostGIS forms the core of the Urban Systems Framework [?] developed by the Digital Phoenix project [12].

As datasets grow and the associated processing becomes more sophisticated, these programs run into the limitations of being executed on a single computer.

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.[**?**] 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 and execution of queries in MPI
4. April 2009 - May 2009: Development and execution 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/