MPI-GIS: High Performance Computing and I/O for Spatial Overlay and Join
Satish Puri, Marquette University
Danial Aghajarian and Sushil Prasad, Georgia State University

I. Introduction
Spatial Overlay and Join are complex operations in Geographic Information Systems (GIS). In GIS, a typical polygon tends to be large in size often consisting of thousands of vertices. Sequential algorithms for this problem are in abundance in literature and most of the parallel algorithms concentrate on parallelizing edge intersection phase only. Moreover, spatial data files tend to be large in size (in GBs) and the underlying overlay computation is highly irregular and compute intensive. Motivated by GPGPU and Parallel IO, we propose to develop and implement scalable distributed algorithms to solve large-scale overlay and join processing.

II. Applications
1. Geographic Information System.
2. Spatial Databases.
3. Computational Geometry.
5. VLSI CAD.

III. Problem description
Polygon overlay combines the input polygons from different maps into a single new map. The input to binary map overlay are two map layers $L_1 = \{p_1, p_2, ..., p_n\}$ and $L_2 = \{q_1, q_2, ..., q_m\}$ where $p$ and $q$ are polygons represented as $(x, y)$ co-ordinates of vertices. The output of the overlay operation is a third layer $L_3 = L_1 + L_2 = \{p_1, q_1, ..., q_m\}$ represented by $k$ output polygons and this output depends on the overlay operator denoted as $\oplus$. (Union, Intersection, XOR, Difference) determines how map layers are combined.

IV. Technical Contributions
1. (Oi(o) n) time polygon clipping algorithm using plane sweep method using Ginkgo-k processors in PRAM model.
2. Design and implementation of a GPU based end-to-end spatial join system.
3. MPI-GIS yields absolute speedup of 44x using R-tree indexing and grid partitioning using MPI compared to ArcGIS 10.1.
4. Scalable Parallel I/O and spatial indexing for large spatial datasets on HPC platform.

V. Different Architectures for Distributed Overlay and Join System

VI. Experimental Results

Fig. 1 Intersection of two polygonal maps

Fig. 2 Overlap Graph for polygons from two map layers

Fig. 3 and 4 shows load distribution for two sets of polygonal data

Fig. 5 MPI-GIS is a user-space parallel library that is built on top of MPI-I/O.

Fig. 6 MPI-GIS with Pthread and CUDA based overlay

Fig. 7 GPU Spatial Join with Filter and Refine sub-systems. MRBs are used for filtering out undesirable polygon pairs and polygon vertices are used for refinement phase.

Fig. 8 Sequential and Parallel I/O time comparison using three datasets.

Fig. 9 Scalability of parallel I/O for Lakes dataset.

Fig. 10 MPI-GIS using 160 MPI processes on 8 nodes.

Fig. 11 MPI-GIS using 320 MPI processes on 16 nodes.

Fig. 12 GPU Spatial Join: breakdown of execution time.

VII. Future Work
1. GPU implementation of Segment tree and TPR tree.
3. Load-balanced overlay processing using MapReduce based CPU-GPU cluster.

VIII. References