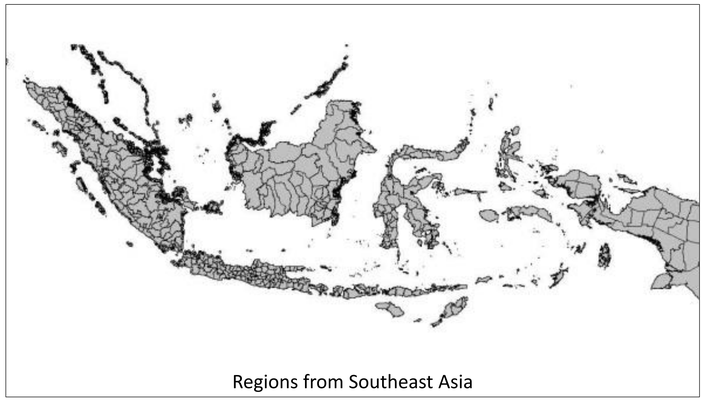
From days, to hours, to minutes, to seconds: an exploration in parallel processing and GIS

Alan Young[1], Robbie Stancil[2], and Arthur Lembo[2]

Introduction

This project focused on solving the classic point-in-polygon GIS problem to locate and summarize 37 million mangrove locations with 400 provincial boundaries in Southeast Asia.



Regions from Southeast Asia

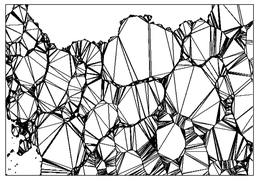
From Days…

The initial results using leading commercial and open source GIS software could not complete the point-in-polygon tests, as the number of vertices in the province boundary layer, combined with the 37 million mangrove locations were simply too extreme. Also, the configuration and extent of the polygonal boundaries created such large minimum bounding rectangles (MBR) that even the incorporation of spatial indexes were of no value to the process.

Complexity of the polygons made processing the images in a reasonable time impossible, as seen in the images above .

To Hours…

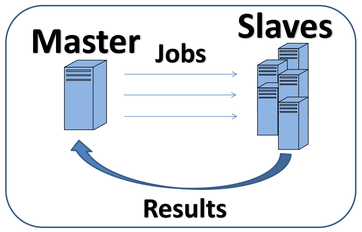
To overcome the large MBR problem, the 400 provincial boundaries were decomposed into smaller, convex parts, creating a 700,000 polygon layer. Somewhat non-intuitively, the point-in-polygon problem was able to complete in just over an hour using the commercial GIS software, the reason being that the 700,000 polygons now had smaller MBRs, and the spatial index was able to eliminate a substantial number of points.

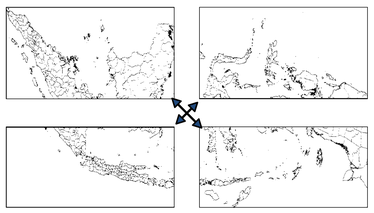


Decompositions of the complex regions data that allowed the task to be completed in hours on GIS software, and minutes using Hadoop. (Left) Convex part decomposition. (Right) Quad-tree decomposition before and after. (Far right) A chart showing the time to complete the spatial join of regions and points when compared to the max complexity of those regions, and how long (in seconds) it took to decompose the original data to that level of simplicity.

Processing with Hadoop

Hadoop is a MapReduce framework that allows users to write code in java using a map-reduce style that can then be issued as jobs from a single master node to as many slave nodes as are set up on a cluster

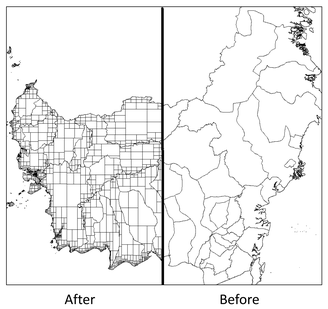




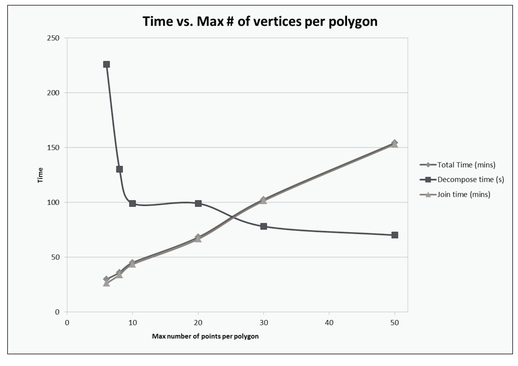
Spatial Hadoop splits up the data to work over multiple machines simultaneously to complete the task much faster than traditional GIS software on bigger data.

To Minutes…

Further reduction in the time was achieved by implementing the point-in-polygon problem using SpatialHadoop, which is an extension to Apache Hadoop designed specially to work with spatial data. The initial configuration utilized two PCs, one operating as a master, and one as a slave, each with 4 CPU cores. Also, further decomposition of the polygonal boundaries were performed to ensure that no polygon had more than 6 vertices, and nearly rectangular.



After Before

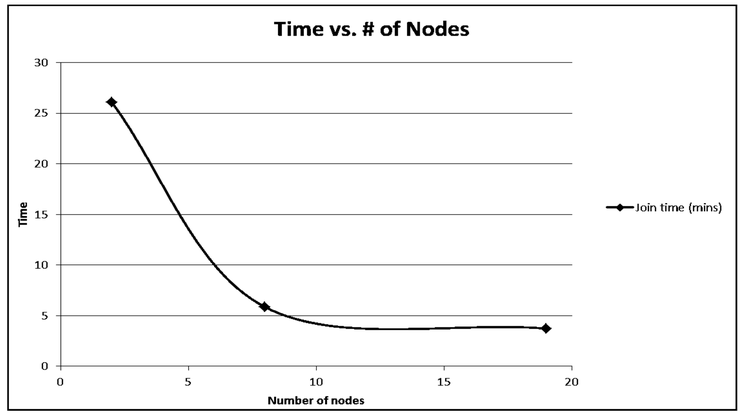


To Seconds

It was believed that our configuration had two limiting factors: a small number of CPUs, and a slow connection speed among the master and slave computers. Therefore, we rented time on Amazon's EC2 m3.large instances for $0.133 per hour of runtime. The results of the point-in-polygon test finished in 210 seconds.

Cluster Limitations

While experimenting with Spatial Hadoop's runtimes, we noticed that our results weren't what we were expecting. After running some tests we found that our cluster was only writing data at a speed of around 1 mb/sec due to the way our network was set up. On top of this we weren't using SSDs which would speed up our times as well. We fixed both of these issues by switching over to Amazon's EC2 service.



This graph depicts the runtime of the spatial join on a Spatial Hadoop cluster of n computers working together. X values are 2, 8, and 19.

Conclusion

Severe processing issues occur when using traditional GIS with very large data. And while some modifications are useful for reducing the processing time, one can achieve greater results when parallelizing the overall problem using large numbers of CPUs. Our results showed that spatial data can be processed effectively at a large scale, but sometimes breaking down the problem is the best first step.

Bibliography

[1] Ahmed Eldawy and Mohamed F. Mokbel. "The Ecosystem of SpatialHadoop". The SIGSPATIAL Special, 6(3), 2014 paper
[2] Ahmed Eldawy and Mohamed F. Mokbel. "SpatialHadoop: A MapReduce Framework for Spatial Data". In Proceedings of the IEEE International Conference on Data Engineering, ICDE 2015, Seoul, South Korea, April, 2015. paper

Acknowledgements
Ahmed Eldawy, for answering any questions we had about SpatialHadoop
This project funded by the NSF grant CCF-1460900

[1] Berry College [2] Salisbury University