MESHED - A Memory-Efficient OpenMP Parallel Algorithm for Delineating a Large Number of Watersheds from High-Resolution Continental-Scale Digital Elevation Models

H51L-0851. Friday, December 13, 2024. 08:30 - 12:20 EST

Huidae Cho <hcho@nmsu.edu>

Department of Civil Engineering, New Mexico State University, Las Cruces, NM 88003

CLAWRIM The Computing Lab for Advanced Water Resources Informatics and Modeling

BE BOLD. Shape the Future.® College of Engineering — NM STATE

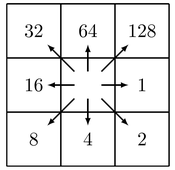
I. Introduction

I set out to delineate watersheds for 91,611 dams across the Contiguous United States (CONUS) in the National Inventory of Dams (NID) database. The task involved processing the USGS 1-arcsecond Digital Elevation Model (DEM), which contains 15 billion cells. The flow direction grid, stored as 1-byte (unsigned 8-bit integer) values, required 14GB of memory. Storing watershed IDs as 4-byte (32-bit integer) values added an additional 56GB, totaling 70GB—well beyond the 64GB memory capacity of my computer. This excessive memory requirement posed a significant challenge. Could I tackle a watershed delineation problem at this continental scale using **existing algorithms** with my hardware limitations? **NO!**

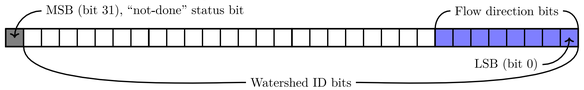
II. Methods and Data

1. Input: Flow directions in 1 byte using GRASS GIS r.watershed and r.mapcalc
2. Input: Outlets
3. Output: Watershed IDs in 4 bytes
4. Input + Output > Memory ➔ Cannot keep both input and output in memory
5. Output > Input ➔ Can store input values in the output matrix if input and output have no interruptions
6. Start with a 4-byte output matrix
7. Read 1-byte input data into this 4-byte matrix
8. For each outlet, perform a **Node-Skipping Depth-First Search** (NSDFS) to trace upstream flow directions
9. Overwrite flow direction values in the traced cells with a watershed ID
10. Embarrassingly parallelize steps 7-8 for all outlets

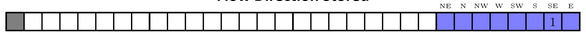
1-Byte Flow Direction Encoding



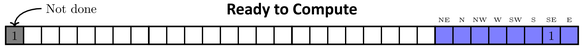
Input/Output Matrix Data Structure



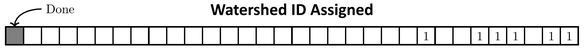
Flow Direction Stored



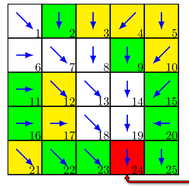
Ready to Compute



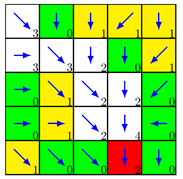
Watershed ID Assigned



Cell IDs and Flow Directions



Number of Input Drainage Paths (NIDP)

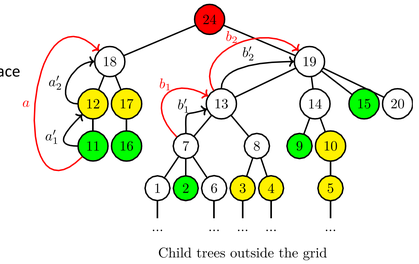


NIDP Statistics for the 30m CONUS DEM

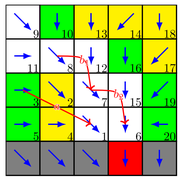
NIDP	0	1	2	3	4	5	6	7
Percent	29.87	49.43	13.99	4.74	1.49	0.48	0.00	0.00

Nearly half of the cells have only one upstream cell, forming a single path that does not require revisiting.

Node-Skipping Depth-First Search



Cell Discovery Orders



Tail Recursion

Prevent stack overflows by avoiding deep recursive calls. Tail recursion is memory-efficient and helps reduce space complexity. It can easily be rewritten as a while loop if needed.

Overwriting Flow Directions

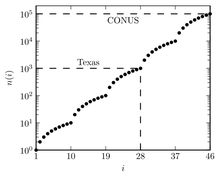
Once a watershed ID is determined for a cell, there is no need to revisit the cell. Just write the watershed ID to it, clearing its flow direction and not-done bit.

Embarrassingly Parallel

Since no watersheds share the same cells, all outlets can be processed in parallel without the need for special considerations. OpenMP is used for single-node parallelization.

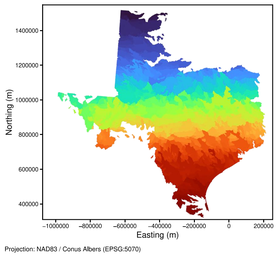
Data

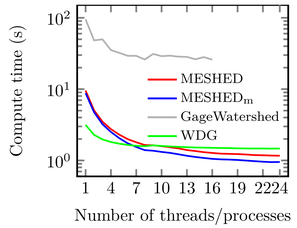
USGS 1-arcsecond (~30m) DEM for the entire state of Texas and the CONUS, with 28 and 46 sets of randomly selected outlets for Texas and the CONUS, respectively, for scaling tests.

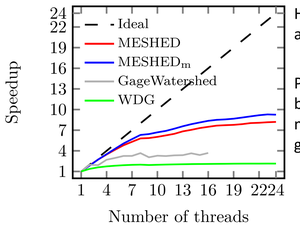


III. Results and Discussion

Texas Results



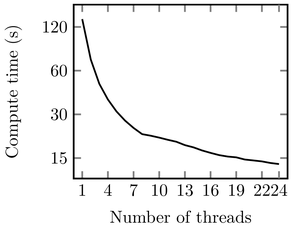


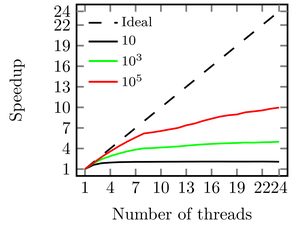


Highly scalable compared to its benchmark algorithms.

Performed 95% faster than the CPU benchmark algorithm using 33% less memory. Can solve 50% larger problems given the same memory.

CONUS Results





Memory-Efficient Watershed Delineation (MESHED)

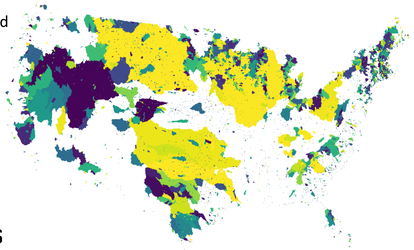
MESHED is an OpenMP parallel algorithm for delineating a large number of watersheds from high-resolution continental-scale DEMs. It is available at https://github.com/HuidaeCho/meshed under GPL3.

It leverages GDAL for data input and output, ensuring cross-platform compatibility.

MESHED is also integrated into GRASS GIS as a new addon, **r.watersheds** (https://grass.osgeo.org/grass-stable/manuals/addons/r.watersheds.html).

91,611 Dam Watersheds in the CONUS

- Dam points not snapped
- 24 threads on i9-12900
- 64GB RAM
- 17.3 seconds



IV. Conclusions

I developed the Memory-Efficient Watershed Delineation (MESHED) OpenMP algorithm to address continental-scale watershed delineation challenges. Its new GRASS GIS addon **r.watersheds** will be available soon.

Future work includes more Application Programming Interfaces (APIs) including Python and R.

V. Attribution

Most figures are reused from Cho (2025).

VI. References

Cho, H., 2025. *Avoid Backtracking and Burn Your Inputs: CONUS-Scale Watershed Delineation Using OpenMP.* Environmental Modelling & Software 183, 106244. doi:10.1016/j.envsoft.2024.106244.

VII. Disclaimer

Mention of trade names or commercial products does not constitute their endorsement either.