Modelling and Optimizing a Graph-based Settlement Delineation and Analysis Workflow

Lorenz Gruber, Nikolas Herbst, Samuel Kounev
Julius-Maximilians-University, Würzburg, Germany

Thomas Esch
DLR, Köln, Germany

Thanh Nguyen
University Kassel, Kassel, Germany

Abstract

State-of-the-art settlement pattern analysis relies on satellite imagery, mapping the outlines of residential areas. The high resolution of these images yields multiple disjoint settlements for a single urban area, distorting the analysis of settlement patterns. Therefore, a workflow was developed to aggregate and analyze urban areas modelled as a settlement graph. Clustered settlements are merged according to the edge contraction procedure, and their importance is evaluated with selected centrality measures. Through workload distribution and parallel execution of *Common Workflow Language* (CWL) tasks with a proprietary scheduler, the workflow can process even the most densely populated regions while respecting workflow stage interdependencies modelled in a *DAG*. The procedure is ultimately deployed and executed on the *HPC* platform "terrabyte". The workflow's configuration, especially implicit parameters like the settlement density in the region of interest, significantly impacts the time required to complete the procedure. Concurrent processing is restricted on the level of connected components, leading to an unbalanced workload distribution when processing large urban areas. To combat inefficient resource utilization on the deployment platform, endeavours are made towards a more flexible *CWL* modelling and elastic resource allocation.

1 Context

In the research field of settlement pattern analysis, scientists are interested in the extent and distribution of settlements and their spatial relationships. Current research often depends on remote sensing data, collected using satellites and leveraged through image processing techniques; however, due to natural barriers or socioeconomic effects, a city is often split into multiple polygon outlines, each representing a cohesive area of human development. Therefore, a workflow was developed that aggregates clusters of settlements into single urban areas, enabling a more representative analysis of settlement patterns. The implementation is based on the *Fishnet*[1] framework and uses a settlement graph to determine mergeable settlement clusters and to compute selected centrality measures. The workflow aims to exemplify the applicability of the framework for scientific computing. Moreover, extensive execution time measurements of the workflow stages provide insights into the impact of parameters and configurations on the time to result and the overall resource utilization.

2 Workflow Stages and Orchestration

The implemented workflow takes a *GIS* raster image from the *World Settlement Footprint* (WSF) [2] dataset as input. After that, the settlement outlines are polygonized and stored in a *Shapefile*.

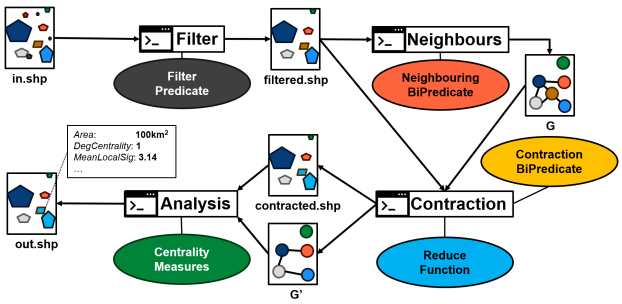

Figure 1: Illustration of the workflow stages, generating, contracting and analysing the settlement graph.

The workflow solely operates on *Shapefiles* and initially filters the settlement polygons according to a filter criterion. After discarding irrelevant settlements from the data, the adjacencies between settlements are computed with a neighbouring predicate indicating whether two settlements are adjacent in the graph. After that, the settlement graph gets contracted. An edge between two settlement nodes is eligible for contraction whenever the binary contraction predicate is fulfilled for the incident settlement vertices. The nodes involved get merged into a single vertex using a reduce function, which may alter the attributes and geometric representation of the settlements, e.g., by

[1] https://gitlab2.informatik.uni-wuerzburg.de/log66jr/fishnet (14.10.2024)

combining their polygons and accumulating their attributes. In the final step of the workflow, the settlement graph is analyzed using selected centrality measures. The possibly merged settlement polygons and their attributes are stored in the workflow's output.

The workload of each stage, depicted in Figure 1, is distributed among several jobs per workflow step, processing only a subset of the input. Therefore, the initial input image is split into multiple *Shapefiles* before executing the workflow. Moreover, after the final analysis stage, the results are merged into a single output file. While the workload distribution is on the level of input files for the `Filter` and `Neighbours` stage, the subsequent stages `Contraction` and `Analysis` can only be parallelized according to the connected components of the settlement graph to ensure the correctness of the *contraction* procedure. Coherent with the *Open Geospatial Consortium* (OGC) best practices[2], every workflow stage is modelled in a single *CWL* [3] description file. Multiple jobs per workflow step trigger the execution of the respective *CWL* task with a partition of the input. The jobs are stored in a central database, and a proprietary scheduler executes the jobs in parallel while respecting the inter-stage dependencies of the jobs, which are modelled in a directed acyclic graph. Processing techniques on the implementation level, like a plane sweep [1] for finding adjacent settlements, further improve the workflow's performance. The workflow was ultimately deployed on the "terrabyte"[3] *HPC* platform and can customized through a configuration file to respect geographic peculiarities in the region of interest.

3 Visualization and Measurements

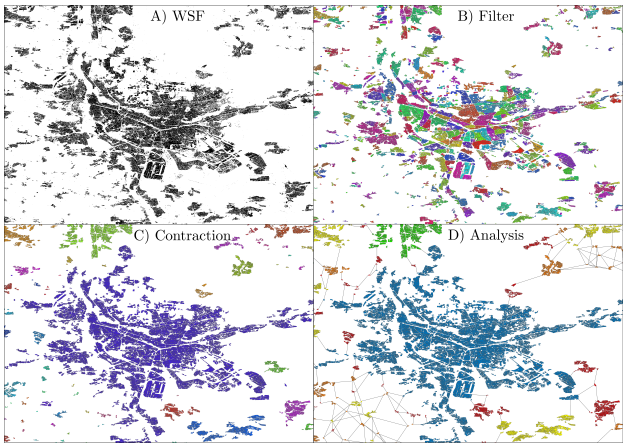

Figure 2: The city of Nuremberg after different workflow stages with the polygons coloured according to their settlement identifier.

[2]https://docs.ogc.org/bp/20-089r1.html (14.10.2024)
[3]https://docs.terrabyte.lrz.de/ (14.10.2024)

The visualizations in Figure 2 especially highlight the aggregation from clustered settlements to single urban areas. This benefits the analysis of settlement patterns, as the visualization of the settlement graph coloured according to *degree centrality* from blue to red exemplifies. In addition to the output *Shapefile*, intermediate results and *log* files are created, with the latter keeping track of every job's execution time. Furthermore, the execution on the "terrabyte" platform yields a report upon completion featuring the utilization of the allocated resources. Consequently, the performance characteristics of the workflow could be analyzed in relation to different input regions and configuration options.

The most impactful parameter is the selected input region, which has the implicit parameters of settlement count, distribution, and density, significantly influencing the properties of the settlement graph and, ultimately, the completion time of a workflow run. Figure 3 illustrates the impact of the input region for different *WSF* tiles, varying significantly in settlement count and density from remote regions to the largest urban areas on the Earth.

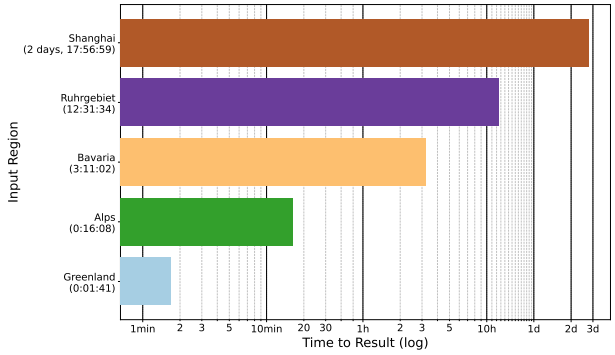

Figure 3: Time-to-result of workflow runs on different input regions, significantly varying in settlement complexity, plotted on a logarithmic scale.

Apart from the input region, the workflow's configuration options also impact the time to result. The *maximum edge distance* for settlement adjacencies emerged as the most impactful configuration option, significantly influencing the complexity of the graph and the execution time of the *polygon neighbours plane sweep*, which is part of the `Neighbours` stage. As Figure 4 shows, the `Neighbours` stage emerged to be the most resource-demanding on average, excluding the initial splitting and final merging, while the analysis of enormous urban areas resulted in the longest execution time of a single job. These single long-running jobs, processing a disproportionate share of the workload, ultimately hurt the resource efficiency of the workflow the most.

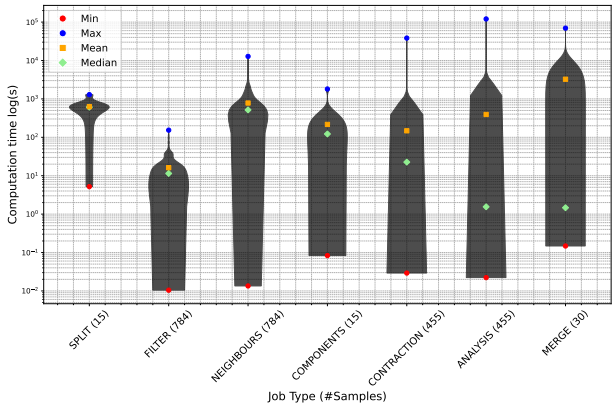

Figure 4: Execution time distribution of each workflow stage plotted on a logarithmic scale.

This effect significantly impacts the utilisation of the *CPU*, as Table 1 highlights. The poor average CPU and memory efficiency are due to the need to allocate computing resources beforehand, reserving according to the maximum expected demand, i.e. for the parallel construction of the settlement graph in the `Neighbours` stage. Consequently, subsequent stages, parallelized on the level of connected components, feature a potentially heavily unbalanced workload distribution, resulting in few jobs that utilize only a fraction of the allocated processing cores. Additionally, the memory footprint significantly depends on the complexity of the settlement network in the input region, with Greenland only requiring about 3GB and Shanghai utilizing almost all available memory of a single compute node.

Resource	Min	Max	Median
Time to Result	0:01:41	2-17:56:59	2:58:50
CPU Time	0:06:43	6-12:48:48	11:00:57
CPU Util. %	3.40%	15.78%	9.06%
Memory	2.99 GB	947.91 GB	16.44 GB
Memory Util. %	1.00%	94.79%	16.44%

Table 1: Minimum, maximum, and median value of selected computing resources across all 15 workflow runs with varying inputs and configurations.

Nevertheless, the workflow provides a performance improvement factor of about $500x$ compared to a sequential prototype implementation using brute-force realizations of the required procedures. When analyzing the *WSF* tile featuring Casablanca in Morocco, the time to result shrunk from about two weeks to just 41 minutes. This significant boost in performance highlights the impact of optimizing the algorithmic complexity of the necessary procedures and parallel processing of the *GIS* jobs.

4 Conclusion and Outlook

The implemented workflow enables geographic researchers to delineate and analyze settlements on a large scale. The performance measurements highlight the impact of configuration options and the input region on the workflow's completion time. Implicit parameters like the settlement distribution and density in the area of interest significantly influence the overall workload and its distribution among the jobs in the later stages. Although the construction of the settlement network is the most expensive stage on average, the potentially heavily unbalanced workload distribution in the analysis stage results in the most extended execution times of a single job. This ultimately results in poor resource utilization due to statically reserving computing resources beforehand according to the maximum expected demand. Consequently, future endeavours focus on a *serverless* [4] approach featuring elastic resource allocation for *GIS* workflows. Therefore, the *CWL* descriptions of workflow stages and the modelling as a *DAG* are generalized to cover typical *GIS* procedures, and the insights provided by the workflow execution time measurements are utilized to enable dynamic allocation of computing resources according to the current demand of the procedure. Ultimately, a *serverless* approach will significantly improve the resource utilization of *GIS* workflows.

Acknowledgement

In addition to the authors above, providing expertise in software engineering and settlement pattern analysis, respectively, *Peter Friedl* and *Julian Zeidler* assisted in deploying the workflow on the "terrabyte" *HPDA* computing platform. This work was partially funded by the Bavarian Research Institute for Digital Transformation (bidt).

References

[1] M. Berg et al. *Computational Geometry: Algorithms and Applications*. Third edition. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2008.

[2] M. Marconcini et al. "Understanding Current Trends in Global Urbanisation - The World Settlement Footprint Suite". In: *GI_Forum* 1 (2021), pp. 33–38. DOI: 10.1553/giscience2021_01_s33.

[3] M. R. Crusoe et al. "Methods Included: Standardizing Computational Reuse and Portability with the Common Workflow Language". In: *Communications of the ACM* 65.6 (2022), pp. 54–63. DOI: 10.1145/3486897.

[4] S. Kounev et al. "Serverless Computing: What It Is, and What It Is Not?" In: *Communications of the ACM* 66.9 (2023), pp. 80–92. DOI: 10.1145/3587249.