Parallelizing a Cellular Automaton Model of Urban Growth Prediction with OpenCL

Jason B. Smith
Department of Computer Science, Hood College, Frederick, Maryland 21701

Abstract—SLEUTH is a well-known simulation model for forecasting urban growth using cellular automata techniques. Although recent work has been done to produce parallel implementations of SLEUTH, most of these efforts have been directed towards clustered MPI solutions. Recent advances in both the computing power of graphics cards and the development of general purpose programming interfaces for these devices provide a new arena for parallelization. We have developed OpenCL-based algorithms for the SLEUTH implementation which provide performance improvements through parallelization on throughput-optimized graphic computing devices.

I. INTRODUCTION

Within the last century planned urban development has seen an increase in interest and research as urbanization has become more common and mathematical and computational techniques have been developed and matured [1]. Models of urban growth have progressed from “comprehensive plans” of an ideal future state to include dynamic models emphasizing the path to achieve goals while including and anticipating short-term difficulties. Harris notes that in both the static and dynamic planning approaches that the mathematical problem of urban simulation is both very large and non-linear with many local optima.

The rapid growth of computer power has allowed an increase in the size and complexity of urban growth simulations, yet the power of current devices still impose limits on the size and expressiveness of existing models and simulations. Modification of existing computational simulation approaches to make use of emerging parallel technology and devices holds promise as a technique to allow us to perform more comprehensive, complex, and larger simulations.

II. BACKGROUND

One of the more recent urban simulation models based on the theory of cellular automata is called SLEUTH [2]. Continuing work on extending this model has been undertaken by a collaboration between the US Geological Survey and the Department of Geography, UC Santa Barbara and is a keystone of “Project Gigalopolis” [3]. Project Gigalopolis predicts that “Urban settlements and their connectivity will be the dominant driver of global change during the twenty-first century.” The goal of this collaboration is develop the models and tools necessary to forecast urban growth on a “regional, continental, and eventually global scale.”

A. Cellular Automata

Cellular automata are models of computation usually based on simple rules but which can exhibit complex behavior [4]. One of the most recognizable cellular automata models is Conway’s Game of Life, in which very simple rules for cell birth and death can result in a variety of structures and behaviors. In general a cellular automata model consists of a grid of cells with with discrete size, and each cell can be a finite number of states. At each time step, the state of the cell can change based on a list of transition rules. These transitions rules consider the current state of the cell as well as the state of some of its neighbors.

B. The SLEUTH Model

The SLEUTH simulation package is named as an acronym for its required image inputs: slope, land cover, exclusion, urbanization, transportation, and hill shade. These inputs are provided as grayscale GIF-format files, and the SLEUTH model parameters are specified in a configuration script. This script contains both the coefficients of variables affecting model operation, but also mechanics of how and where to read input and output. Figure 1 copied from Project Gigalopolis [3] shows a high-level view of the prediction process.

The SLEUTH simulation model actually incorporates two different models: the Clarke Urban Growth Model (UGM) [5] and the Deltatron Land Use/Land Cover Model (DLM) [6]. The UGM is a cellular automaton which considers urban regions along with transportation networks and slopes in order to generate predictions of urban growth in an area over time. The DLM model uses land use information along with urbanization information from the UGM to model how land use might transition in the presence of urbanization or other land use change.

The SLEUTH software itself can handle three distinct modes of operations: testing, calibration, and prediction [3]. Testing mode is used to create a set of historical data without having the hard constraints on model parameters that calibration does. This test mode allows verificaiton of coefficients generated by hand or by calibration processes.

The calibration process is a more complex procedure used to fine-tune the model parameters by running multiple scenarios with randomized parameter values. The calibration process performs a Monte Carlo simulation of simultaneous growth simulations and emits statistical output to allow hand selection of the most desirable or most accurate input parameters by the investigator.
The prediction process of SLEUTH is also a Monte Carlo simulation of multiple simulations, but differs from the calibration mode by having the initial state and parameters identical for each chain of simulations. The randomization within the model allows each simulation to evolve differently.

1) **Clarke Urban Growth Model:** The UGM is a cell-based model in which each time-step or cycle represents one year. The behavior of the model itself is controlled by user supplied coefficients which are applied and possibly modified during the course of a simulation. During each cycle the region is examined on a cell-by-cell basis and growth rules are applied to determine how the state transitions of urbanization occur. The behavior of the UGM is parameterized by five coefficients which affect the behavior of the system [5]. These coefficients are:

1) diffusion factor - determines the overall dispersion of an urban distribution
2) breed coefficient - a probability measure controlling how likely a newly generated urban cell will be to start its own growth
3) spread coefficient - determines how much growth from existing urban regions occurs
4) slope resistance - influences the likelihood of urbanization of cells on steeper slopes
5) road gravity - a measure which controls how much roads attract new settlements toward themselves

When beginning to model a new region much of the early work consists of tuning these coefficients to accurately reflect historical growth of the area to be modeled. SLEUTH itself provides some capabilities for attempting to calibrate these parameters through brute-force methods.

Each UGM growth cycle consists of four growth steps called Spontaneous Growth, New Spreading Centers, Edge Growth, and Road-Influenced Growth [2]. One cycle of these steps results in the prediction for a year worth of growth. The growth rules considered for each cell during a single cycle are:

1) Spontaneous Growth - this rule provides for spontaneous stochastic urbanization of a cell considering only the attributes found within that cell.
2) New Spreading Centers - this rule determines whether any of the newly urbanized cells from Spontaneous Growth will become new urban centers capable of having an increased chance of affecting the transitions of their adjacent neighbors.
3) Edge Growth - this rule represents the spread of urban areas from urban centers. When an urban area has two or more urban neighbors then the non-urban neighboring cells have a chance of becoming urbanized.
4) Road Growth - this rule models how roads are not only attractors of urban development, but also influence the direction of urbanized growth. When a new road growth is triggered, a random walk is performed along the road within some distance to trigger an urban transition.

2) **Deltatron Land Use/Land Change Model:** Initially described in [6], the DLM models how land cover changes in response to urbanization transition events. Clarke observed that land use rarely converges into a long-term stable equilibrium, rather it is marked by periods of short-term stability followed by periods of transition. The DLM works by tracking the current state or value of for a particular cell, its neighbors, and how recently a state transition has occurred in a cell. This aging layer is called the “deltatron layer”. The DLM has three growth rules and is driven by urbanization events triggered from the UGM. These growth rules are:

1) Initiate Change - when a new urbanization event occurs, that is when a cell is transitioned to be in an urban state in the UGM, then a new potential for land use change is handled. This probability of change is weighted by the average slopes of each class, historical land cover changes forming a transition probability matrix, and the slope of the current cells. Taken together these factors define a Random Markov Field for the cell with probabilities for transitioning from the current state of the cell to another.
2) Create Change Cluster - when a new deltatron is created as a result of a land use change, the probability of the change affecting a larger number of neighboring cells is handled by this rule. The maximum growth of this potential cluster is configurable, and the state to transition to is the same as the initiating land use state.
3) Propagate Change - this rule can be considered to perform the same function as the Edge Growth rules in the UGM model. It defines the probability that adjacent neighbors to a cell will be converted to the same type as a neighbor’s land type.

At the end of the application of these rules, each existing deltatron is aged by one year. After a specified amount of time has elapsed for a single cell, the deltatron is removed. The period when a deltatron exists for a particular cell represents the time that it has achieved a state of local equilibrium. The aging process which removes deltatrons frees them for participation in future change events.

C. **OpenCL and GPGPU Computation**

We have designed and implemented a modification to SLEUTH to allow integration with recent graphics devices
capable of general computational tasks. Specifically, we have created a SLEUTH implementation modified to make use of OpenCL [11] to address those areas of the SLEUTH model cycle that are amenable to parallel processing.

OpenCL is a software framework for creating software capable of executing in heterogenous computing environments. We specifically use the portions of OpenCL addressing parallel computation tasks intended for execution on GPUs. One of the compelling reasons to use OpenCL is its status as a common API supported by multiple device vendors. It is hoped that applications making use of OpenCL can enjoy more widespread adoption and usage since the required hardware would ideally be vendor-neutral.

III. RELATED WORK

The SLEUTH software is still under active development addressing model improvements and research regarding additional transition rules to better model real world urbanization. In addition, work has been performed to address scalability concerns with larger datasets. These larger datasets allow urban growth forecasting to extend beyond local regions to address larger scales such as regional and eventually global simulations.

A native SLEUTH installation is capable of being run on a cluster using MPI. Recent work by Xian resulted in modification for the model to run on a Beowulf cluster with a reportedly significant decrease in computational time required [8]. More extensive work on parallelization of SLEUTH has been performed by Guan with the end result of a general purpose parallel raster library called pRPL [9] targeted at the geospatial domain along with a parallel implementation of SLEUTH called pSLEUTH which makes use of the pRPL library [10].

Although these techniques have been successful in reducing the overall computational time through speedups due to parallelism, both of them require MPI based infrastructures to distribute SLEUTH workloads.

IV. METHODOLOGY

As seen in the SLEUTH prediction process in Figure 1, the simultaneous Monte Carlo method employed within the calibration and prediction processing operations of SLEUTH lends itself immediately to parallelization efforts. This type of parallelization can be considered “task-parallelization” [10]. The MPI based parallelization already included in SLEUTH exploits this fact by having different nodes handle different simulation instances of this step. However, this approach may not be suitable for an OpenCL approach for larger simulations since constraints on device memory may introduce an undesirable communication overhead with the host when swapping simulation models. In addition, this simulation-per-node parallelization implies that a researcher has access to a suitable cluster to obtain enough resources to benefit from this task decomposition.

Another possibility for parallelization which has been exploited by Guan [10] is data parallelization achieved through decomposition of the input data. In particular, Guan examines row-wise, column-wise, block-wise, and quad-tree decomposition techniques. One concern he raises, which quad-tree decomposition can help address through its uneven partitioning, is the problem with load balancing work units across available compute devices to maintain maximum throughput.

The data parallelization approach is the one which we have developed using OpenCL in order to make use of the large number of cores available in modern GPUs. Using this approach we will partition the input data sets in smaller pieces suitable for simultaneous computation within the GPU. The cellular automaton model of the SLEUTH urban growth model and its required inputs maps well with the requirements and processing capabilities of GPUs and OpenCL. The two-dimensional input images and coefficients form the input for the urban growth models, and each pixel of the input images is considered a cell in the cellular automaton. Each cell is handled by an individual GPU-based thread. Since the locality required for the cellular automaton implemented in UGM is relatively small, we minimize the need for inter-worker communication and exploit the data-parallel nature inherent in the input data.

Following is a sequential algorithm for general urban growth prediction where $G$ is the collection of input images and $C$ is a general structure containing the rule coefficients and model parameters. [TODO: Fill out with more specific information. Might need multiple algorithm listings - one for the main loop and others detailing the workings of individual model manipulations]

**Algorithm 1** PREDICT-URBAN-GROWTH($G, C, StopYear$)

**Input:** SLEUTH input images $G$

**Input:** SLEUTH rule coefficients and model parameters in a structure $C$

1: $MC = initializeMonteCarlo(C)$
2: $conditionWorkingModel(G)$
3: while $MC.currentYear < MC.stopYear$ do
   4: $MC.incrementYear()$
   5: $spreadUrbanGrowth(G, C)$
   6: $growModel(G, C)$
   7: $conditionworkingmodel(G, C)$
   8: $stats = updateStatistics(G, C)$
   9: $updateCoefficients(stats, C)$
10: $storeGrowthResults(G, C)$

The individual functions invoked within the main algorithm which work with the input images $G$ and model parameters and coefficients $C$ implement the SLEUTH rules described the background material. We have implemented a OpenCL-based functions for these discrete data manipulations and growth rules. Following the modular approach of the current SLEUTH implementation, we have provided alternative code modules for an OpenCL Growth package and associated OpenCL Utility package [TODO: Maybe insert figure of SLEUTH packages here?].
The Utility package defines the discrete OpenCL kernels which are applied to the input data. At a higher level, our version of the Growth package contains the infrastructure code required to initialize and prepare the data for use on GPU devices, call the appropriate Utility methods to perform the SLEUTH rules using OpenCL kernels, and finally retrieving the updated model data from the GPU device. [TODO: Maybe Peril-L algorithms for Utility kernels?]

V. RESULTS

The introduction of a local, highly-parallel computational environments to key areas of the low-level SLEUTH simulation flow result in significant speedups when compared to SLEUTH running on single-machine environments, even when those machines have multiple local cores available.

[TODO: Figure showing running times of normal vs. OpenCL sleuth with different input sizes]

[TODO: Figure showing speedup factor]

As seen in the the figures above, we obtain time speedups of ....

VI. CONCLUSION

We have implemented an OpenCL-based implementation of SLEUTH which allow modelers to make use of commodity GPU devices to reduce the time required to perform urban growth simulations. As shown in our results, this approach provides a speedup of [TODO: Insert Speedup] when using SLEUTH compared to the time required for the same simulation using traditional single-system simulations. This minimal introduction of GPU-based parallel approaches to the SLEUTH simulation model already provides substantial benefits for resource-limited researchers.

However there are still many opportunities for additional research and improvements to an OpenCL based SLEUTH implementation. Certain obstacles that will need to be overcome to make effective use of throughput-oriented computational devices running in a heterogeneous environment are still evident. Deficiencies in our existing models such as CTA and CUDA are mentioned in [12], in particular the problems with memory layout and communication in a heterogenous computing environment are still present and need to be addressed to ensure ease of use and scalability in different computing environments. Another area that must be addressed is how to partition larger datasets to be optimally handled and make effective use of both host computing cores along with graphics device environments. Finally, we anticipate that investigation intro hybrid parallelization techniques including both multi-core CPU and cluster-based task decomposition coupled with GPU operations providing data parallelization provide a tantalizing view of simulation modeling.

REFERENCES