Efficient Visual Analytics on Large Graphs

Xintian Yang, Advisor: Srinivasan Parthasarathy
[yangxin.srini@cse.ohio-state.edu]
Department of Computer Science and Engineering, Ohio State University

Motivation

- Large graph represents data from various domains
  - Nodes represent entities of interest
  - Edges represent interactions among them
- Mining interaction networks is important to gain insight into structure, properties and behavior of these networks
- Visual Exploration: key to understand structure and behavior
- Scalability: efficient back-end algorithm improves the interactivity of front-end visualization.
- GPU: highly parallel, high performance.
- Power-law property: leading to inefficient execution of mining kernels on GPU.

Proposed Visual-Analytic Framework

- Knowledge
- Selection
- Analysis
- Graph Mining Algorithms
- PageRank
- Random Walk with Restart
- HITS
- Zooming and filtering

Graph View

- Hierarchical Coarsening
- Community View
- Node View
- Power-law Graphs
- Tiling

CUDA Architecture

- Programming Model
- Hardware

Power-law Graphs

- Three kinds of memory accesses.
  - Access to A. Access to x and writes to y
  - Previous methods have optimized accesses to A
- Observation 1: Each row accesses random elements in vector x
- Observation 2: The accesses are non-coalesced – poor locality
- Solution 1: Tile A by columns and store x on the texture cache
- Observation 3: Column lengths are power-law distribution
- Many short columns, little re-use of x in these columns
- No benefit from tiling
- Solution 2: Reorder by column length and partially tile A.

Composite Storage

- Observation 4: Rows (non-zeros) in each tile follow power law
- Observation 5: Within each tile performance is limited by
  - load imbalance
  - conditional thread divergence
- Solution 3: Composite tile storage scheme
- Basic observation from benchmarking study
  - CSR-vector kernel performs well on long rows if they are stored in row major.
  - ELL kernel performs well on short rows if they are stored in column major.
- Reorder the rows in each tile from long to short.
- Rows are partitioned into workloads with similar size.
- A thread warp is assigned to compute one workload.
- A workload is a rectangle area of non-zeros with width w and height h.
- If w > h, row major storage, CSR-vector kernel
  - Else, column major storage, ELL kernel

Experimental Results

- Parallel implementation of SpMV kernel on power-law matrices
- Performance (GFLOPS)

Conclusions

- Presented an efficient visual analytic framework for large power-law graphs
  - Toolkit designed to capture and track evolutionary behavior of nodes and communities
  - Use of coarsening to visualize large graphs
  - Presentation of multiple views with visual cues for interactive and exploratory visual analysis
  - Architecture conscious optimizations for SpMV
  - Architecture features of GPU
  - Characteristics of graph mining applications
  - Significant performance improvement on power-law graph datasets.

Acknowledgement

- This work is supported by the following grants:
  - DOE Early Career Principal Investigator Award No. DE- FG02-04ER25611
  - NSF CAREER Grant IIS-0347662
  - NSF SGER Grant IIS-0742999

- This is joint work with Dr. Sharan Asur and Professor P. Sadasivan.

References