Parallel computation of community structures in graphs

Department of Computer Science
Independent Study by Bobby Srisan
Advisor: Dr. Martin Burtscher
May 1, 2020

TEXAS STATE UNIVERSITY

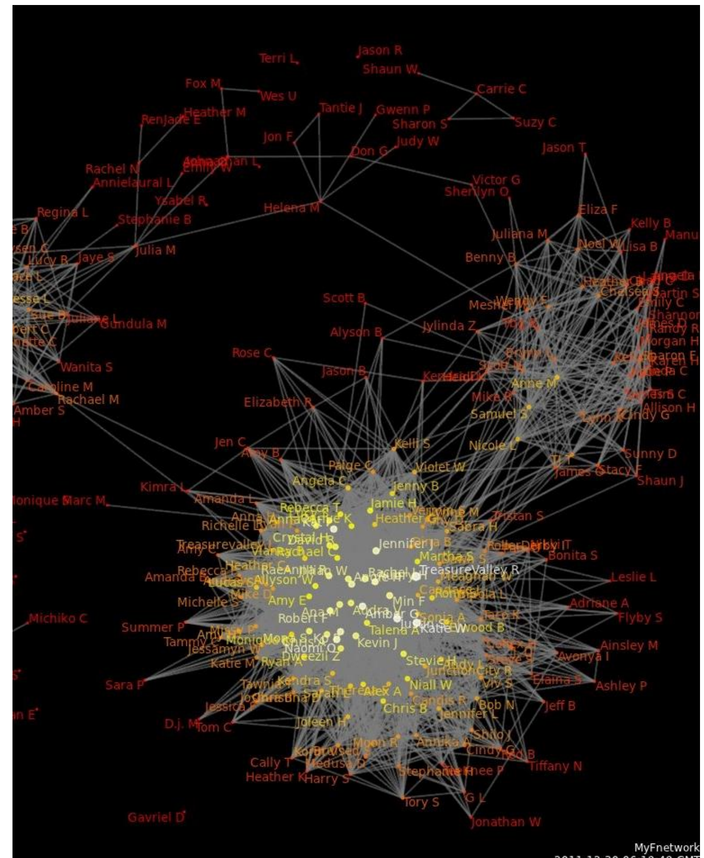
The rising STAR *of Texas*

Overview

- Background of communities in network analysis
- Parallelization project
- Implementation
- Results
- Analysis
- Summary

Network Analysis

- Information systems and complex networks are represented as graphs
- Nodes/vertices connected by links/edges
- Use in social networks, mobile phone networks, web
- Organization in the randomness



Data visualization of Facebook relationships by the third-party app MyFnetwork
https://commons.wikimedia.org/wiki/User:Kencf0618

Communities in Networks

Communities: subunits in a network

Identifying communities helps uncover a-priori unknown ordering

Protein to protein interactions	Communities with shared language dialects	Different group identities discussing a Twitter trend

TEXAS STATE UNIVERSITY
The rising STAR of Texas

Community Detection

- General idea: Optimized network partitioning
  - Communities of densely connected nodes and sparse interconnections between communities
- Precise formulations are known to be computationally intractable
- "Goodness" of partition measured by modularity value Q
- Objective: maximize Q modularity

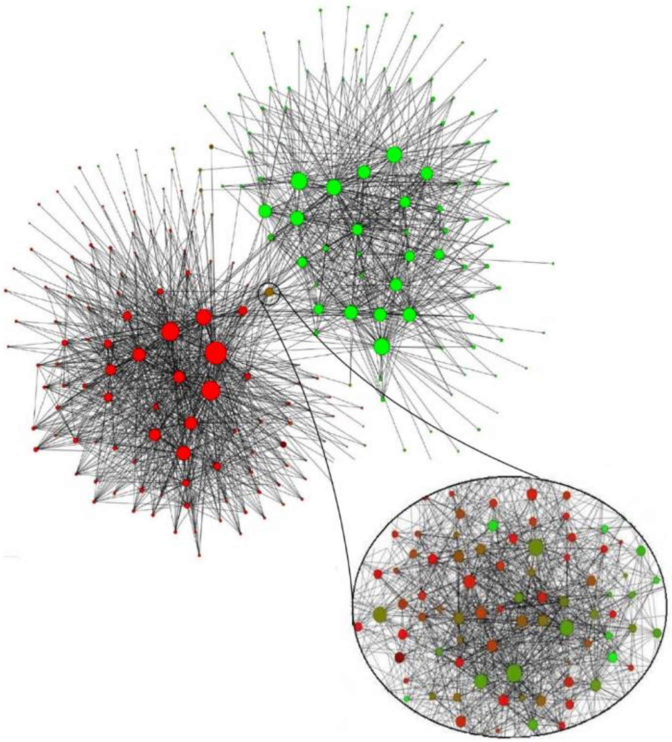
$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j)$$

where
A_{ij}: weight of link between i and j
k_i: sum of link weights attached to node i
c_i: community of node i
$\delta(c_i, c_j)$: 1 if $c_i == c_j$, else 0
m = sum of link weights in graph

TEXAS STATE UNIVERSITY
The rising STAR of Texas

Louvain Method of Community Detection

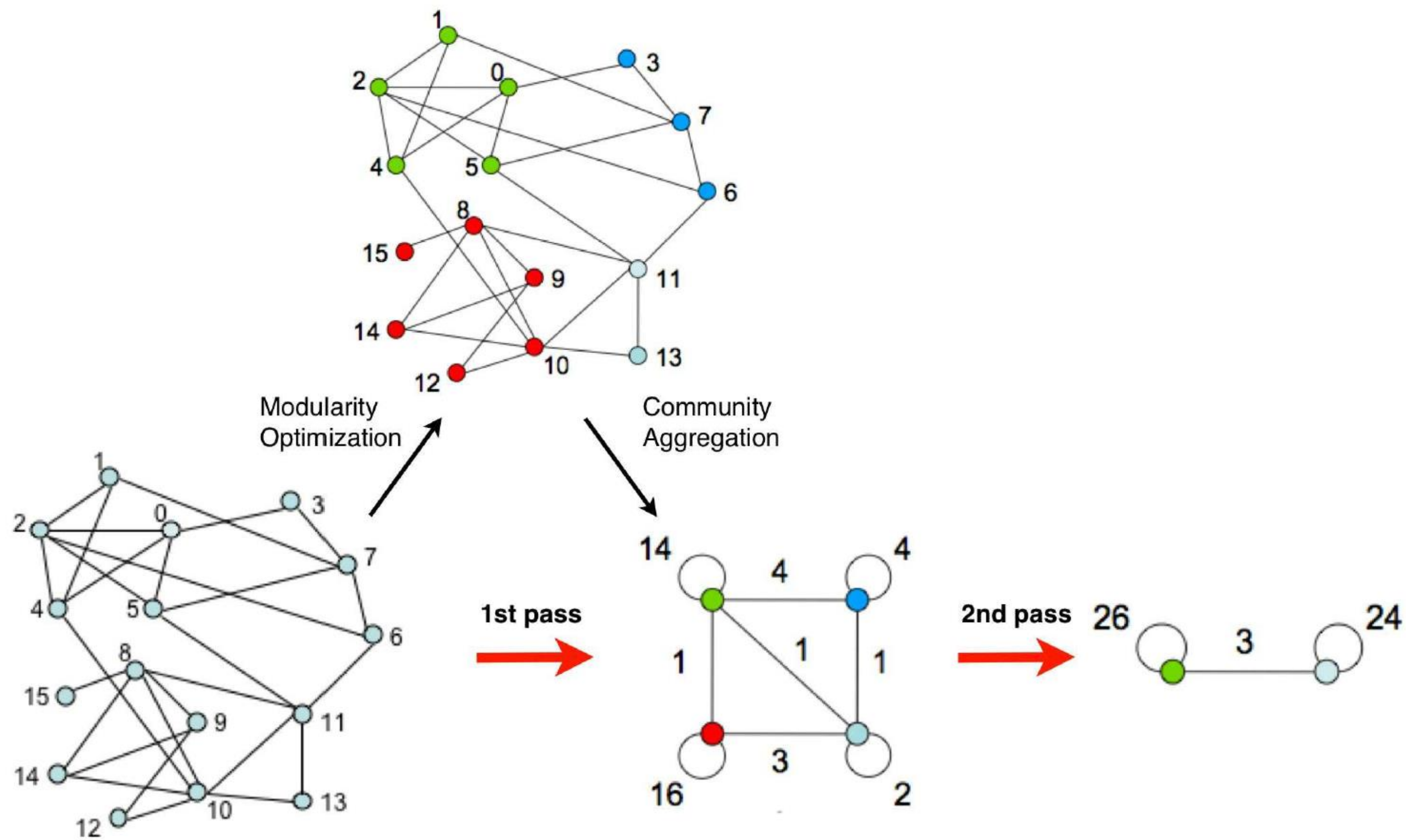
- Rapidly finds high modularity partitions of large networks; *Blondel et al*.
- Unfolds hierarchical community structure for the network
- Unsupervised clustering



[1] Blondel, Vincent D.; Guillaume, Jean-Loup; Lambiotte, Renaud; Lefebvre, Etienne (2008). "Fast unfolding of communities in large networks". Journal of Statistical Mechanics: Theory and Experiment. 2008

TEXAS STATE UNIVERSITY
The rising STAR of Texas

Louvain Algorithm



[1] Blondel, Vincent D.; Guillaume, Jean-Loup; Lambiotte, Renaud; Lefebvre, Etienne (2008). "Fast unfolding of communities in large networks". Journal of Statistical Mechanics: Theory and Experiment. 2008

TEXAS STATE UNIVERSITY
The rising STAR of Texas

Room for Improvement?

- While Louvain outperforms other methods, graphs of many millions of nodes/links will take minutes
  - Example: "Web uk-2005" has 39M/783M nodes/links took 738s to achieve 0.979 modularity
- Explore parallel processing using GPUs to further reduce overall computation time in partitioning nodes into communities

Modularity Optimization/Computation Comparison

Graph name	Karate	Arxiv	Internet	Web nd.edu	Phone	Web uk-2005	Web WebBase 2001
Nodes/links	34 / 77	9k / 24k	70k / 351k	325k / 1M	2.6M / 6.3M	39M / 783M	118M / 1B
Clauset, Newman, and Moore method	.38 / 0s	.772 / 3.6s	.692 / 799s	.927 / 5034s	-/-	-/-	-/-
Pons and Latapy method	.42 / 0s	.757 / 3.3s	.729 / 575s	.895 / 6666s	-/-	-/-	-/-
Wakita and Tsurumi method	.42 / 0s	.761 / 0.7s	.667 / 62s	.898 / 248s	.56 / 464s	-/-	-/-
Louvain Method	**.42 / 0s**	**.813 / 0s**	**.781 / 1s**	**.935 / 3s**	**.769 / 134s**	**.979 / 738s**	**.984 / 152min**

[1] Blondel, Vincent D.; Guillaume, Jean-Loup; Lambiotte, Renaud; Lefebvre, Etienne (2008). "Fast unfolding of communities in large networks". Journal of Statistical Mechanics: Theory and Experiment. 2008

* modularity value/run time

** -/- means the method took over 24hrs to run.

TEXAS STATE UNIVERSITY

The rising STAR of Texas

Parallelizing the Louvain Method

TEXAS STATE UNIVERSITY

The rising STAR of Texas

Porting Strategy

- Understand general description of algorithm
- Top down read of serial code: Start at general operation and drill into specific functions.
- Understand flow and data structures
- Identify what is calculated, modified, and accounted
- Profile serial code and identify functions where time is spent
- Identify parallelizable section(s) where most time can be gained back

TEXAS STATE UNIVERSITY
The rising STAR of Texas

Serial Timing

- Serial code: Most time spent computing “one level”

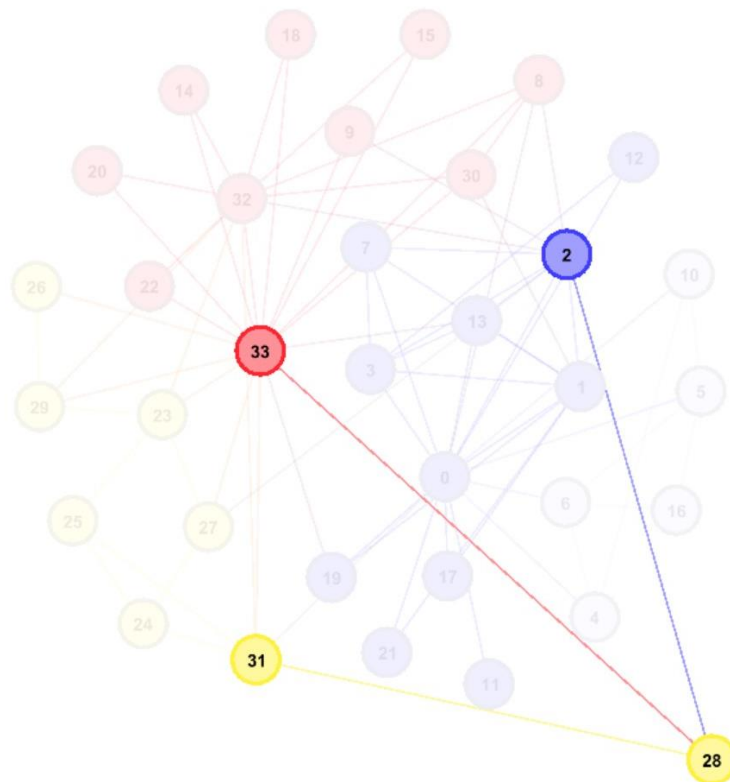
```
      1741849860 function calls (1740434396 primitive calls) in 2173.463 seconds

Ordered by: cumulative time

ncalls  tottime  percall  cumtime  percall filename:lineno(function)
     1    0.004    0.004 2173.463 2173.463 {built-in method builtins.exec}
     1    0.093    0.093 2173.458 2173.458 <string>:1(<module>)
     1    0.175    0.175 2173.365 2173.365 community_louvain.py:161(best_partition)
     1    2.714    2.714 2171.409 2171.409 community_louvain.py:253(generate_dendrogram)
     6  433.219   72.203 2059.197  343.200 community_louvain.py:463(__one_level)
```

TEXAS STATE UNIVERSITY

The rising STAR of Texas

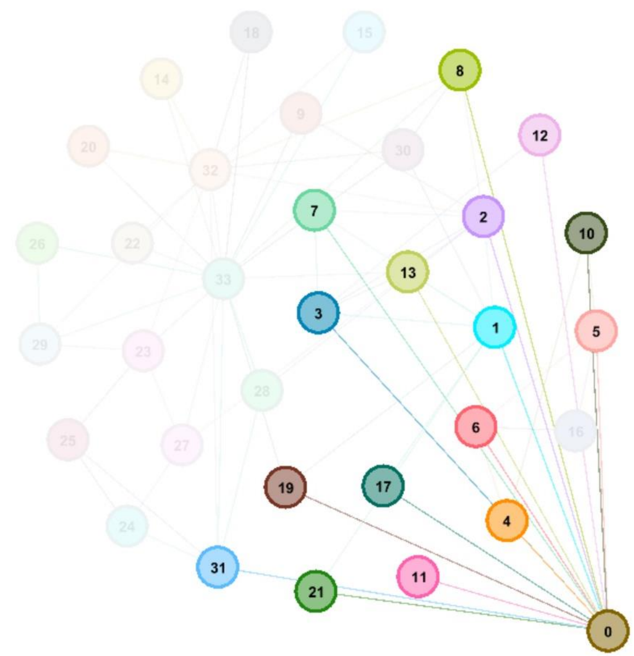
Computing One Level

- For each node, consider all neighbors and their communities
- Compute hypothetical modularity if node were reassigned to a neighboring community
- Keep track of change in modularity
- Reassign node to community that gave highest positive change



Computing One Level

- Larger size => more nodes and neighbors to iterate through
- So, assign GPU threads to work on one node and its neighbors
- All nodes simultaneously search for their best community
- Perform a global consensus on which one has the most to gain



TEXAS STATE UNIVERSITY
The rising STAR of Texas

Computing Modularity Change

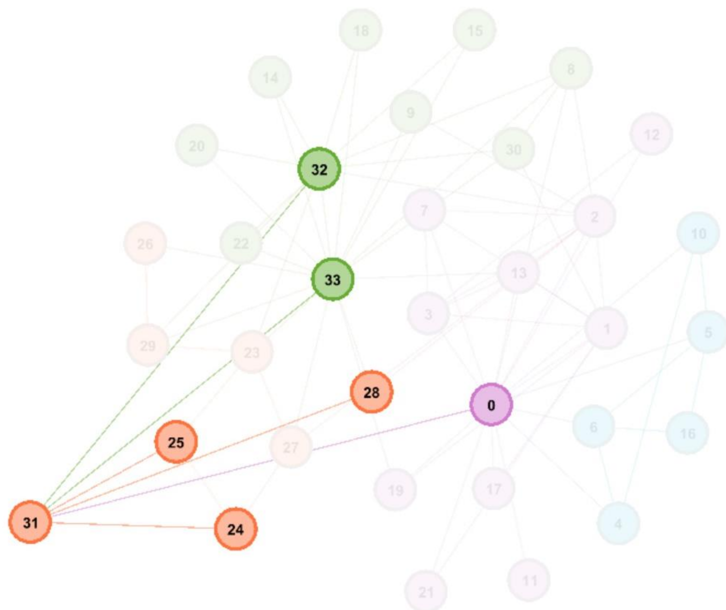
$$\Delta Q = \left[\frac{\sum_{in} + k_{i,in}}{2m} - \left(\frac{\sum_{tot} + k_i}{2m}\right)^2\right] - \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m}\right)^2 - \left(\frac{k_i}{2m}\right)^2\right]$$

Given node i and neighboring community C,

- Σ_{in} sum of links strictly inside community
- $k_{i,in}$ sum of links from i to nodes in C
- Σ_{tot} sum of links strictly inside community
- k_i sum of links to node i
- m sum of all links in entire graph

TEXAS STATE UNIVERSITY
The rising STAR of Texas

Computing One Level



Problem: Data dependency of shared community arrays; Q is calculated from these

Solution: Keep arrays as read only; do some personal accounting if community under consideration is current community

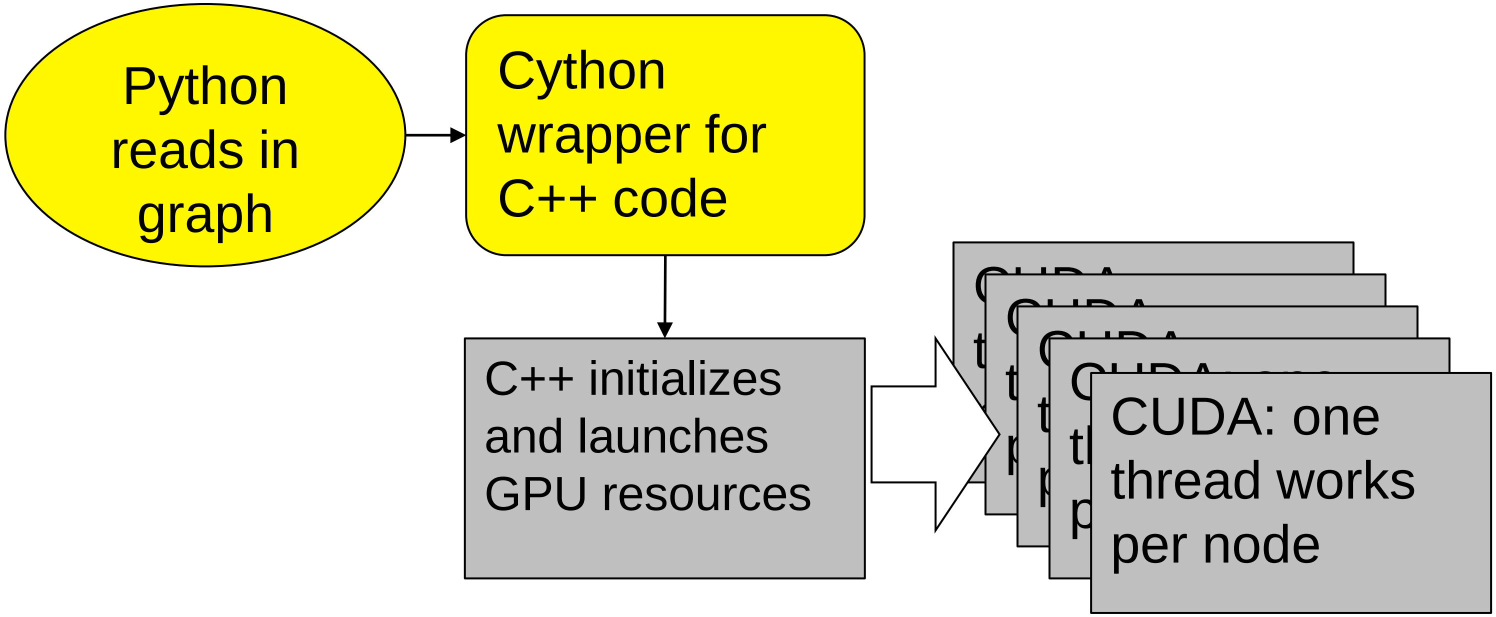
Implementation

Python, Cython, CUDA

- Python: High-level, general-purpose, “batteries included”
  - very useful for data analysis when we aren’t too concerned with writing new functions from scratch
- Cython: Lets us go “under the hood” to optimize performance
- CUDA C/C++ API: Interface with host (CPU) to GPU device and call operations to take advantage of parallelism
- During analysis, import the Cython wrapper module to run and get results

Data Passing

- Read in graph and allocate arrays in CPU and GPU memory
- CPU initiates the GPU compute kernel
- GPUs cores run computation for each node in parallel



Hardware

- CPU: Two Intel Xeon X5690 @ 3.47 GHz
  - Number of Cores: 12 (2x 6-core UMA)
  - Number of Threads: 12
  - Main Memory: 24 GB
- GPU: NVIDIA Tesla K20 GPU
  - 13 multiprocessors
  - 3.52 TFLOPS single, 1.17 TFLOPS double
  - Threads per multiprocessor: 2048

TEXAS STATE UNIVERSITY
The rising STAR of Texas

Data

- Karate social network
- Nodes: 34
- Links: 78
- Average degree: 5
- Max degree: 16

Results

Graph	Louvain run time	GPU run time	Louvain modularity	GPU modularity
karate	0.0044	0.0033	0.417	0.394

Average times are in seconds

Serial runs repeated 5 times, randomized nodes

TEXAS STATE UNIVERSITY

The rising STAR of Texas

Observations

- GPU produces a result that original Louvain method can obtain
- Max of max can easily result in lower modularity at the end
- GPU initialization overhead

TEXAS STATE UNIVERSITY

The rising STAR of Texas

More Parallelism?

- Even with shared arrays, some information can be read/written without interfering other thread's calculations.
- Two active nodes are independent if neither writes to same community
  - Process independent nodes in parallel
- Cautious operations
  - Node A and node B are joining/leaving same community
  - Node A leaving a community that node B is joining, or vice versa

TEXAS STATE UNIVERSITY
The rising STAR of Texas

Summary

- Improve Louvain implementation through speedup
    - Parallelization of one level
- Simple global greedy search for highest increase in modularity does not ensure maximum modularity
- Simultaneously assigning non-adjacent communities
    - Requires thread safety mechanisms

TEXAS STATE UNIVERSITY
The rising STAR of Texas

TEXAS STATE UNIVERSITY

The rising STAR of Texas

Thank you!