Parallel Brain Network Analysis Platform

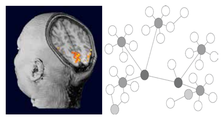
Haixiao Du, Xiaoming Chen, Yu Wang
Department of Electronic Engineering, Tsinghua University, Beijing,China

Source Code Released On
http://parabna.weebly.com

Motivation

Human Connectome
- The structural and functional connectivity patterns of the human brain

A Popular Method for Mapping Human Connectome
- A combination of non-invasive neuroimaging techniques and graph theoretical approaches.
- Under this framework, the brain is modeled as a complex network containing a large quantity of nodes and connections.
  - Reveal the properties of the brain (small-world, modular…)
  - Understand the mechanism of brain diseases (Alzheimer, Schizophrenia…)



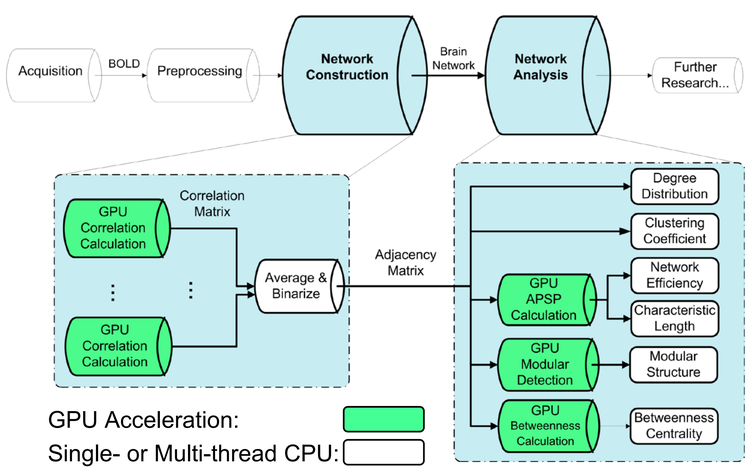
Challenges
- Large-scale network derived from high-resolution imaging data
- Large datasets of multi/many Subjects
- Time consuming

Solutions
- Down sampling to a lower resolution using anatomical atlas
  - Biased by anatomical structure
  - Loss of large information and intra-region connections
- Parallel acceleration
  - Multi-core CPU for coarse-grained and non-regular data parallelism
  - NVIDIA GPU for fine-grained and regular data parallelism

Platform Overview

Environment
- CPU: Intel Xeon E5405 & Xeon X5680
- GPU: NVIDIA GTX 580 & Tesla K20C
- Programming Language: CUDA & C++

Platform Overview



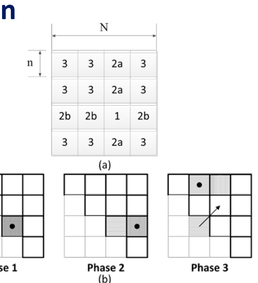
✓The input data format of our platform is 4D nifti which is generally used in the area. No problem of compatibility.

Implementation & Speedup

Why and How We Use GPU
- Some matrix operations very efficient on GPU
  - GEMM (General Matrix Multiplication)
  - SpMV (Sparse Matrix-Vector multiplication)
- Some Graph analytical algorithms can be expressed as basic matrix operations
  - Transform traditional graph computations into regular matrix operations:
    Correlation Matrix, Floyd Warshall APSP -> GEMM
    Spectrum Partition -> SpMV
  - Blocked design to assure scalability for large-scale network
- Other efficient applications on GPU which has SIMD structure and continuous access pattern
  - Calculation of betweenness centrality
  - Probabilistic fiber tractography

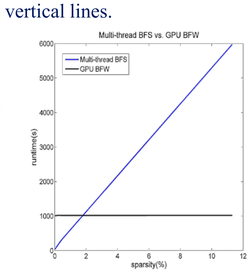
Some Details

All-pairs Shortest Paths (APSP) Calculation
- Blocked Floyd-Warshall algorithm
  - The cost matrix was divided into r blocks
  - r rounds, 3 phases in each round
  - Process in a round shown in the figure.
    phase 1: update primary block;
    phase 2: update blocks (same row or the same column with the primary blocks;
    phase 3: update all of the other blocks.

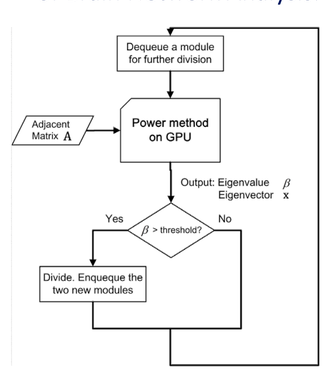


(a) Illustration of which phase each block belongs to. (b) Updating the dotted block requires two source blocks denoted with horizontal lines and vertical lines.

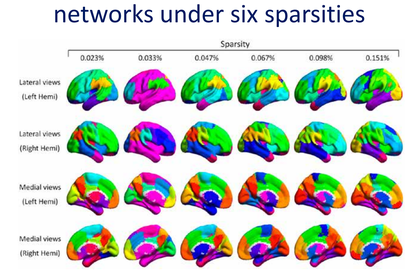
- its performance compared with multi-thread Breadth first Search.
- We released source code for both algorithms on the website. Users can choose from them depending on the sparsity of the network.



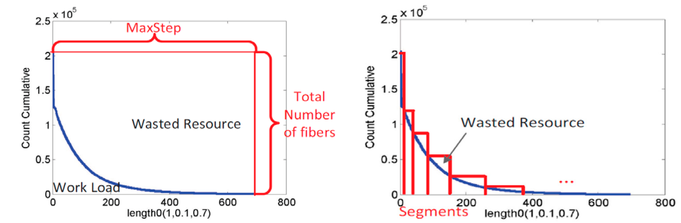
Modular Detection
- Divide one module into 2 subdivisions using Spectrum partition methods in each round
- Power method: repetition of SpMV
- GPU implementation for sparse matrix vector multiplication present in our previous work (Wu D, Wu T, Shan Y, Wang Y, He Y, et al. (2010) Making Human Connectome Faster: GPU Acceleration of Brain Network Analysis. ICPADS)



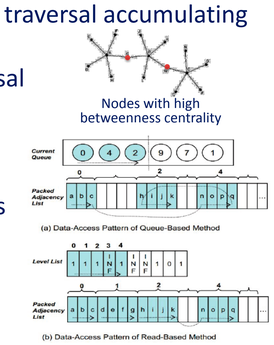
- The modular structure of brain networks under six sparsities



Probabilistic Fiber Tractography
- Input: Diffusion Tensor Magnetic Resonance Imaging(DT-MRI)
- 2 Steps:
  MCMC (Markov chain Monte Carlo) and Probabilistic Streamlining
- Parallel: streamlining starting from multiple seed voxels.
  - Problem: unbalanced workload
  - Solution: Task-Segmentation reduce wasted resource



Betweenness Centrality
- 2 Steps: 1) Breadth first Search; 2) Reverse traversal accumulating
- Parallel:
  - Thread-level parallelism for edge traversal
  - One block calculates the betweenness centrality of one node
- Optimization:
  - Read-based methods enable continuous memory access
  - A virtual warp accesses all neighbors of a node



Speedup and Performance

Algorithm	CPU time (s)	GPU time (s)	Speedup
Correlation Matrix	245.8	2.0	123
Modular Detection	568.29	218.67	2.6
Blocked FW	138830	326.3	425.5
Betweenness	1114	116	10
Probabilistic Streamlining	306.6	5.82	52.7

Data
- Network Size (from fMRI data)
  - ~40k nodes (voxels)
  - Sparsity: 0.13%
- DTI data for Probabilistic Streamlining
  - ~205k voxels
  - step length : 0.3 voxel

Here are results under only 1 sparsity of functional network and 1 setting of DTI data. Although running time of modular detection and betweenness calculation is sensitive to the network sparsity, the speedup results are relatively stable.

Why Do You Need ParaBNA

High Speed
- Can do fast mapping of high-resolution human brain connectome
- Faster than other widely used tools such as FSL and SPM
- Parallel calculation of betweenness is available

Scalable to a larger network size
- Blocked design for calculation of the correlation matrix and APSP which need to store a whole or half matrix.
- Use CSR format which require little memory to store sparse networks on GPU for other applications.

Low cost
- Only need a PC with an NVIDIA GPU (of course a good one is better)
- More proper for analysis of human brain network compared with PC cluster.