DIVISION OF RESEARCH
TEXAS A&M UNIVERSITY

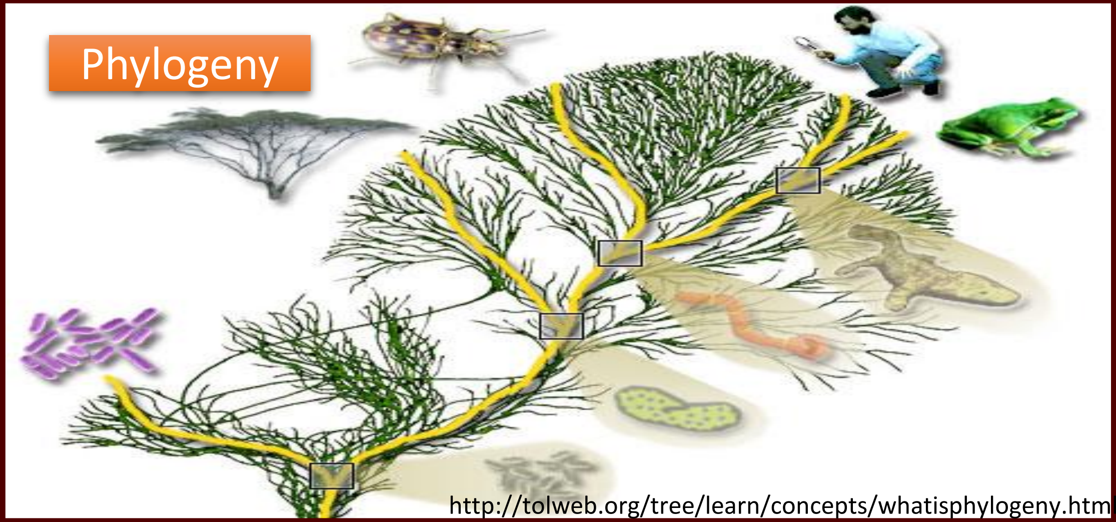
Finding Vertex Cover: Acceleration Via CUDA

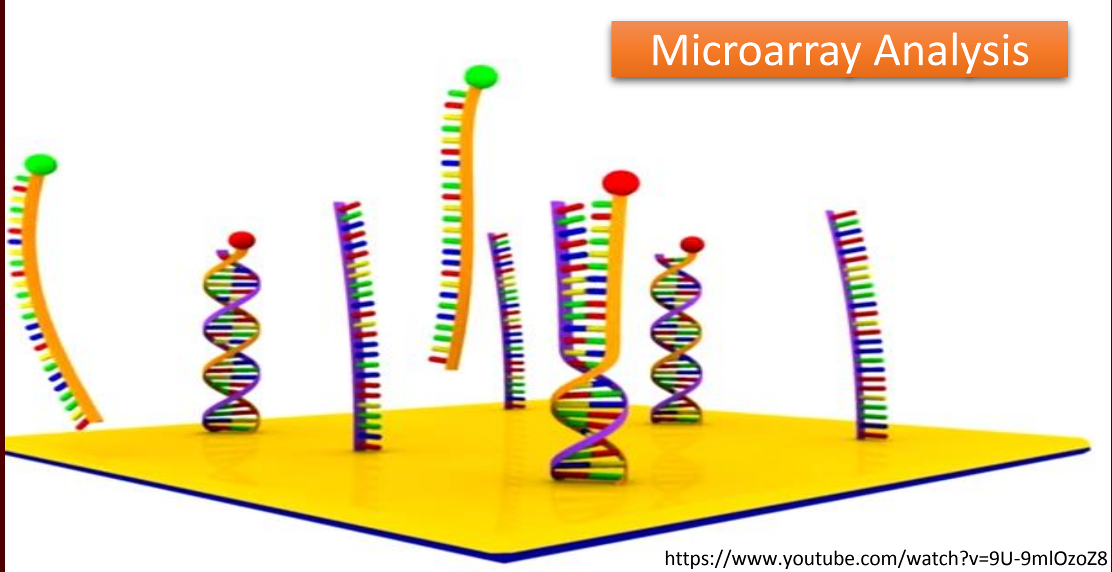
Yang Liu, High Performance Research Computing, Texas A&M University

Jinbin Ju, Electrical Engineering, Texas A&M University

Derek Rodriguez, Computer Engineering, Texas A&M University

Motivation

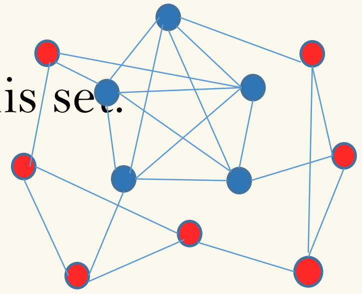

Phylogeny
http://tolweb.org/tree/learn/concepts/whatisphylogeny.html


Microarray Analysis
https://www.youtube.com/watch?v=9U-9mlOzoZ8


Motif Finding
Active Site
Groove
www.ncbi.nlm.nih.gov/pmc/articles/PMC2286532/figure/f2

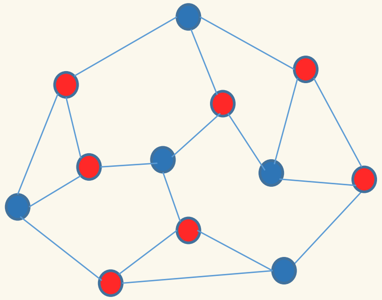
Related Problems: Maximum Clique and Maximum Independent Set

- Clique
  - A set of vertices such that there is an edge between any pair of vertices in this set.



Example of Clique (Blue Vertices)

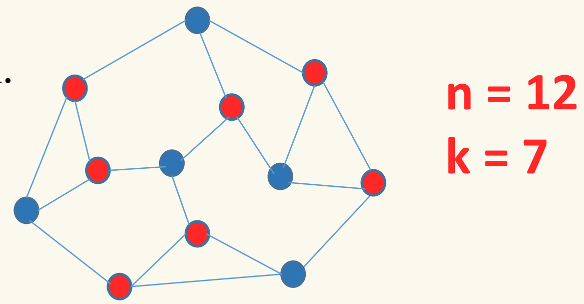
- Independent Set
  - A set of vertices such that there is no edge between any pair of vertices in this set.



Example of Independent Set (Blue Vertices)

Vertex Cover

- Vertex Cover:
  - A subset of vertices such that every edge is incident to at least one vertex in the subset
- Minimum Vertex Cover:
  - Find a vertex cover of minimum size.
  - One of Richard Karp's 21 NP-Complete Problems
- Parameterized Vertex Cover
  - Given a parameter k, find a vertex cover of size at most k.



Example of Vertex Covers (Red Vertices)

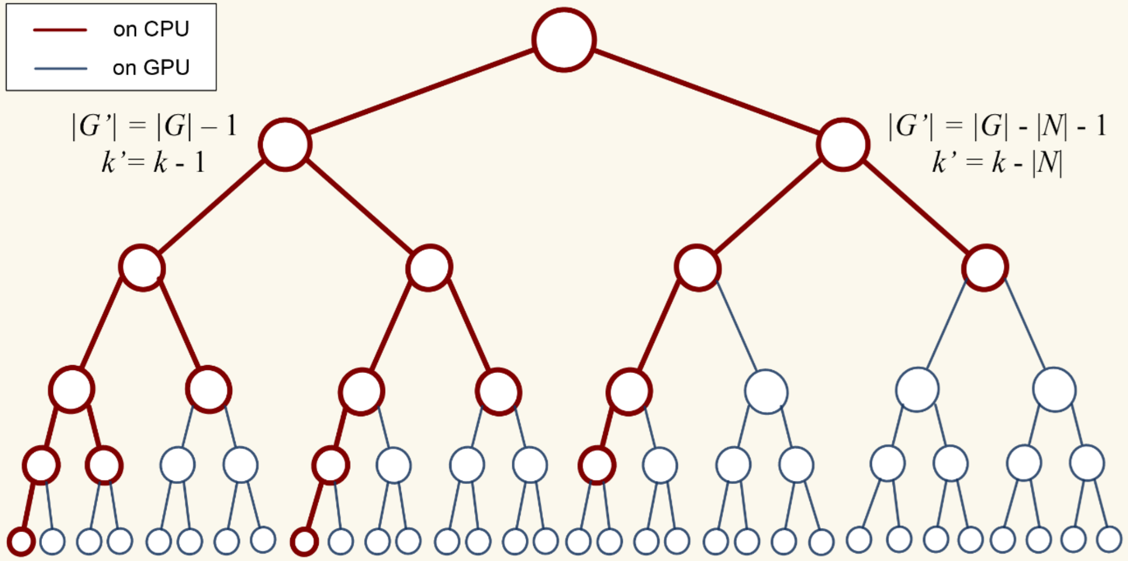
Related Work

- These problems have been extensively studied (exact algorithms, parameterized algorithms, approximation algorithms, and heuristics).
- Parameterized Vertex Cover
  - Best parameterize algorithm: $O^*(1.2738^k)$ (by Chen, et. al. 2010)
  - Parallel implementation scales up to 2400 CPUs (by Weerapurage et. al. 2011).
- Independent Set
  - Best exact algorithm: $O^*(1.1888^n)$ (by Robson, 2001).
  - W[1]-hard, i.e., unlikely to have algorithms of complexity $O(f(k)p(n))$ where $f(k)$ is independent of n.
- Clique
  - A problem for the second DIMACS (Discrete Mathematics & Theoretical Computer Science) implementation Challenge: 1992-1993.

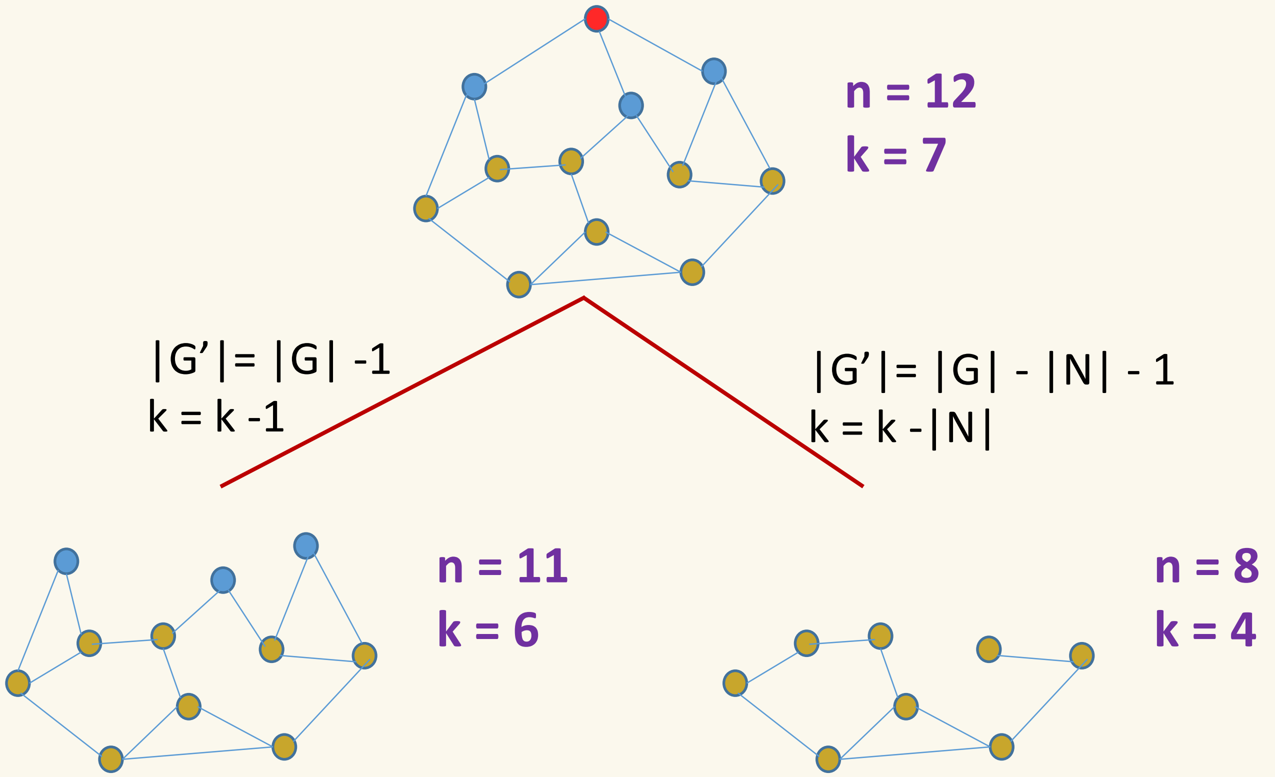
Branching Process

Branching Process

- Find max degree vertex v
- Two branch sets: v is in vertex cover or v's neighbors are in vertex cover



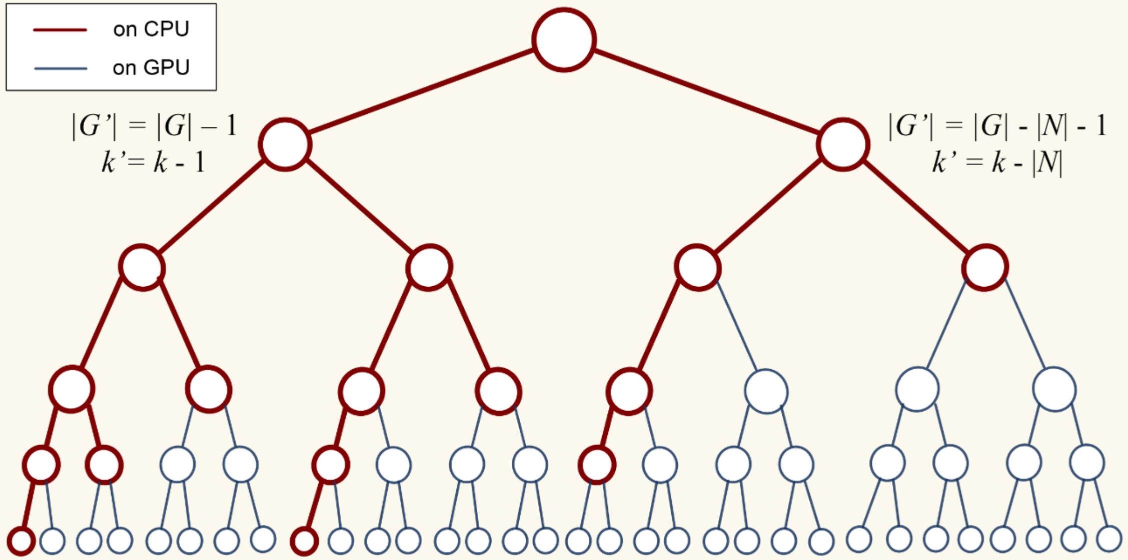
Branch Searching Process



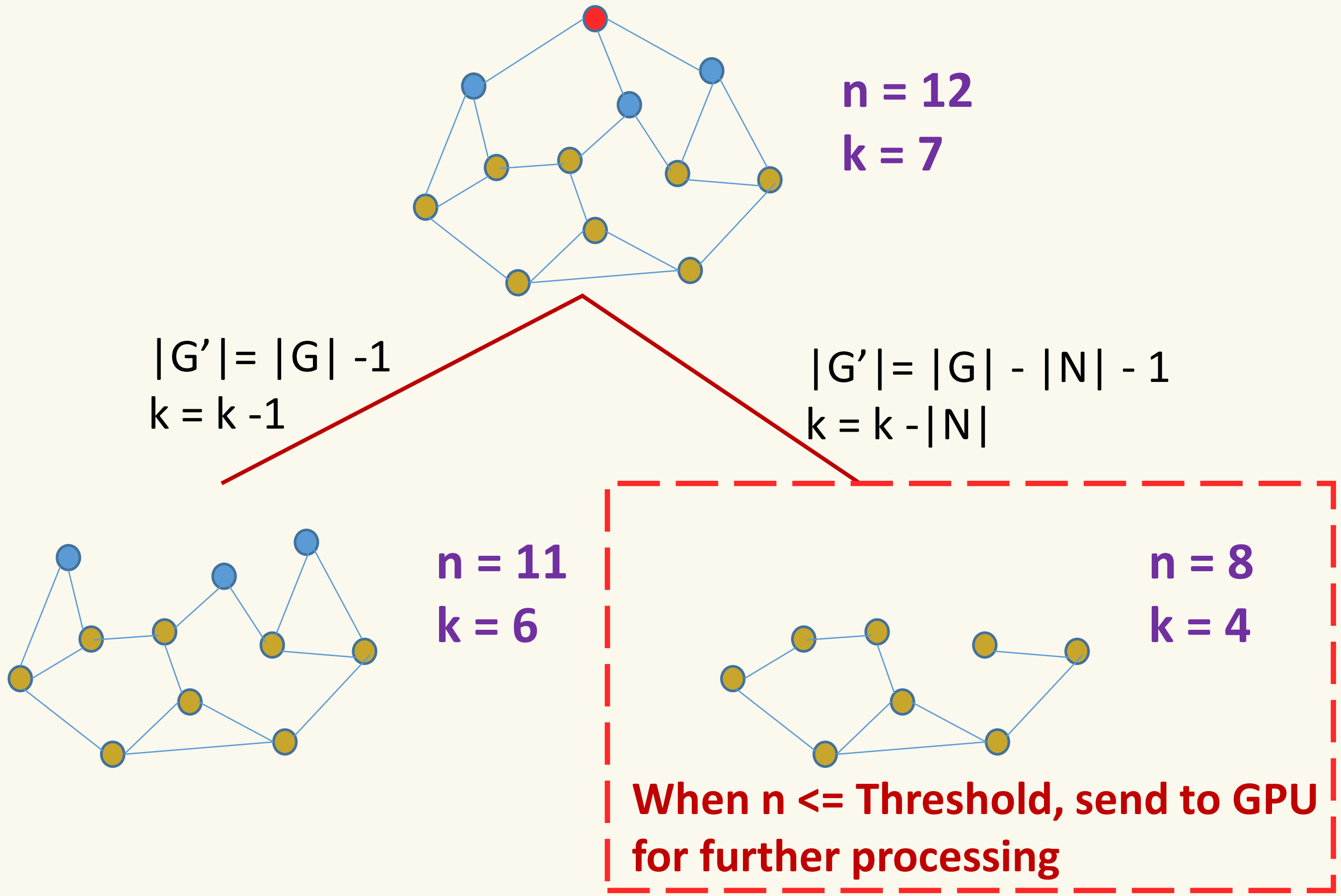
Branching Process—Threshold for GPU Processing

Branching Process

- Find max degree vertex v
- Two branch sets: v is in vertex cover or v's neighbors are in vertex cover



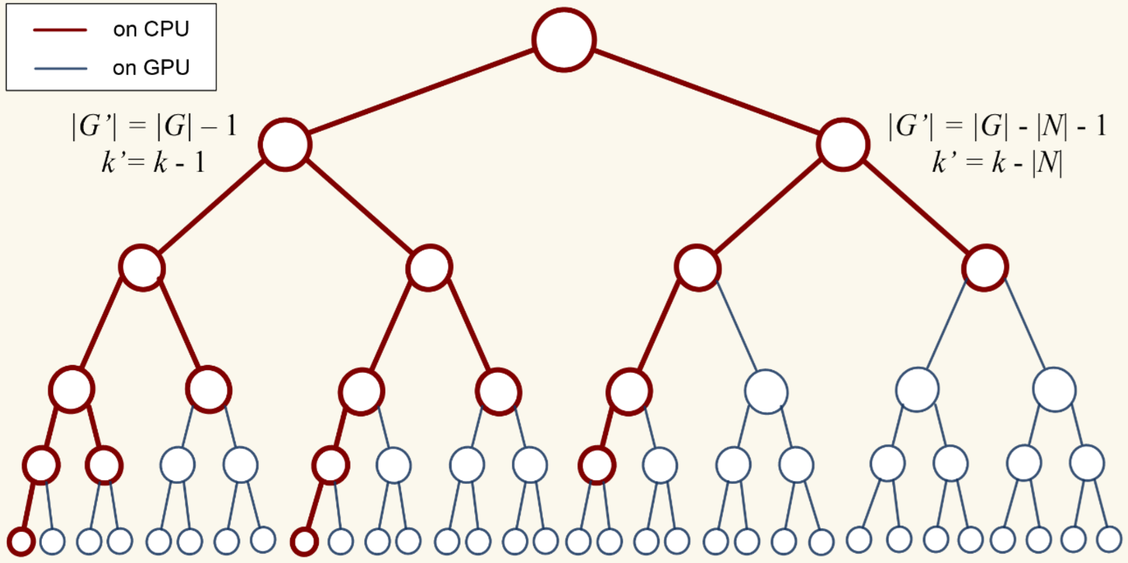
Branch Searching Process



Synchronization between CPU and GPU

Branching Process

- Find max degree vertex v
- Two branch sets: v is in vertex cover or v's neighbors are in vertex cover



Branch Searching Process

CPU:

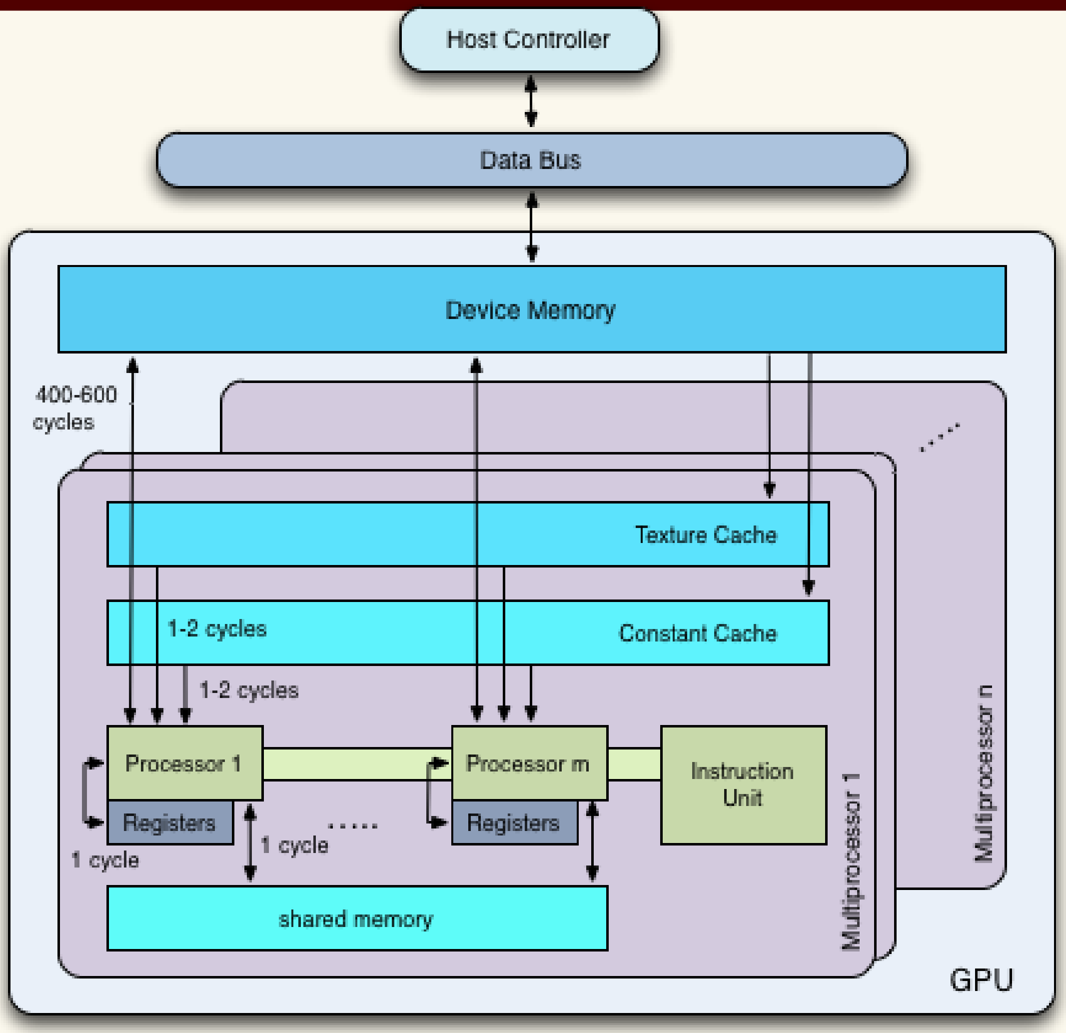
```
While (there is a graph to branch){
   branch and create a small graph    // n <= THRESHOLD
   if (a vertex cover is found by kernel)
      return;
   if  (SMALL-GRAPH-COUNT small graphs are created){
       start kernel to process SMALL-GRAPH-COUNT small graphs
                              //can overlap with creation of small graphs
   }
}
```

Synchronization between CPU and GPU:

Kernel and CPU communicate on solution state (vertex cover is found or not) via **memory copy from GPU to CPU.**

(We tried mapped memory, but somehow our program is unstable, and very difficult to debug)

GPU Memory Hierarchy

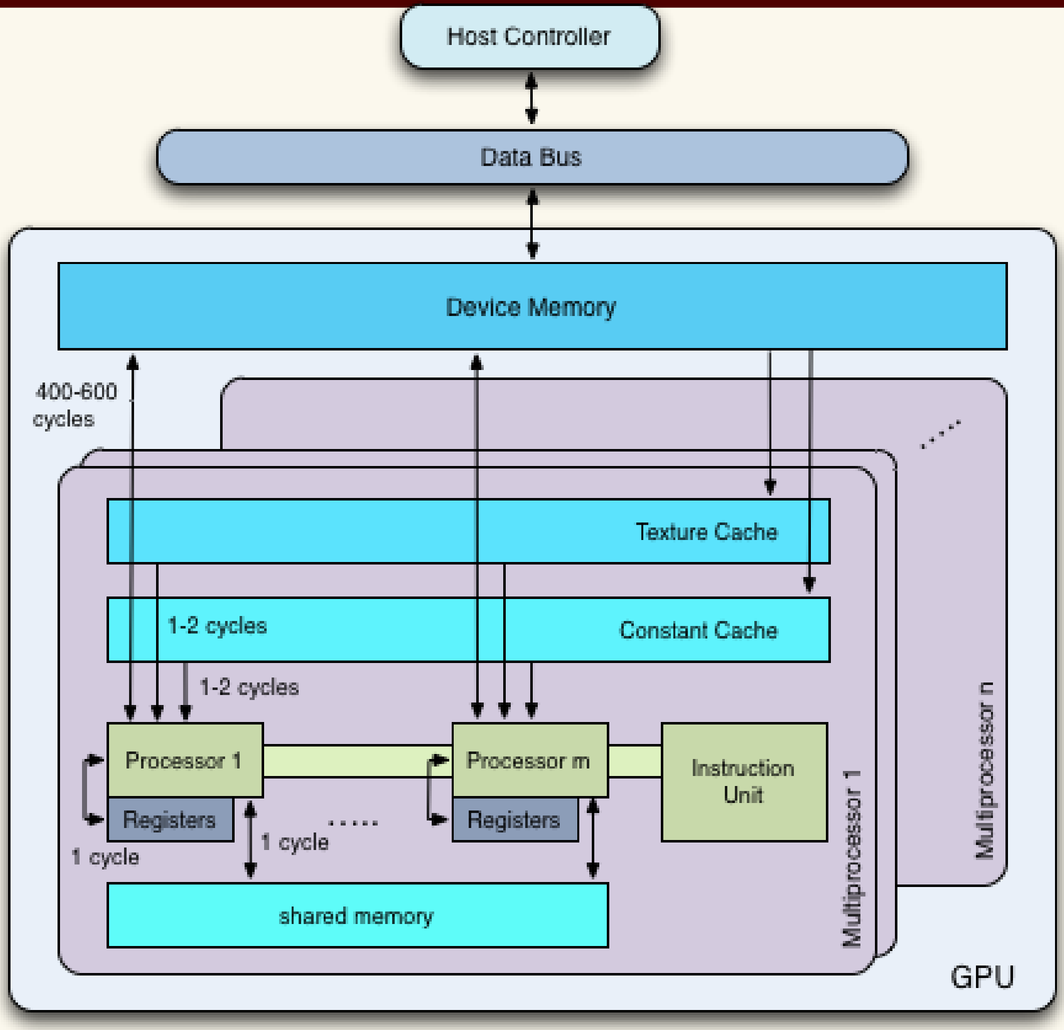


https://www.quantalea.net/media/_doc/2/7/manual/index.html?GPUHardwareImplementation.html

Memory	Bandwidth
Register Memory	~8,000 GB/s
Shared Memory	~1,600 GB/s
Global Memory	~177 GB/s
Mapped Memory	~8 GB/s

DIVISION OF RESEARCH
TEXAS A&M UNIVERSITY

Placement of Small Graphs in GPU

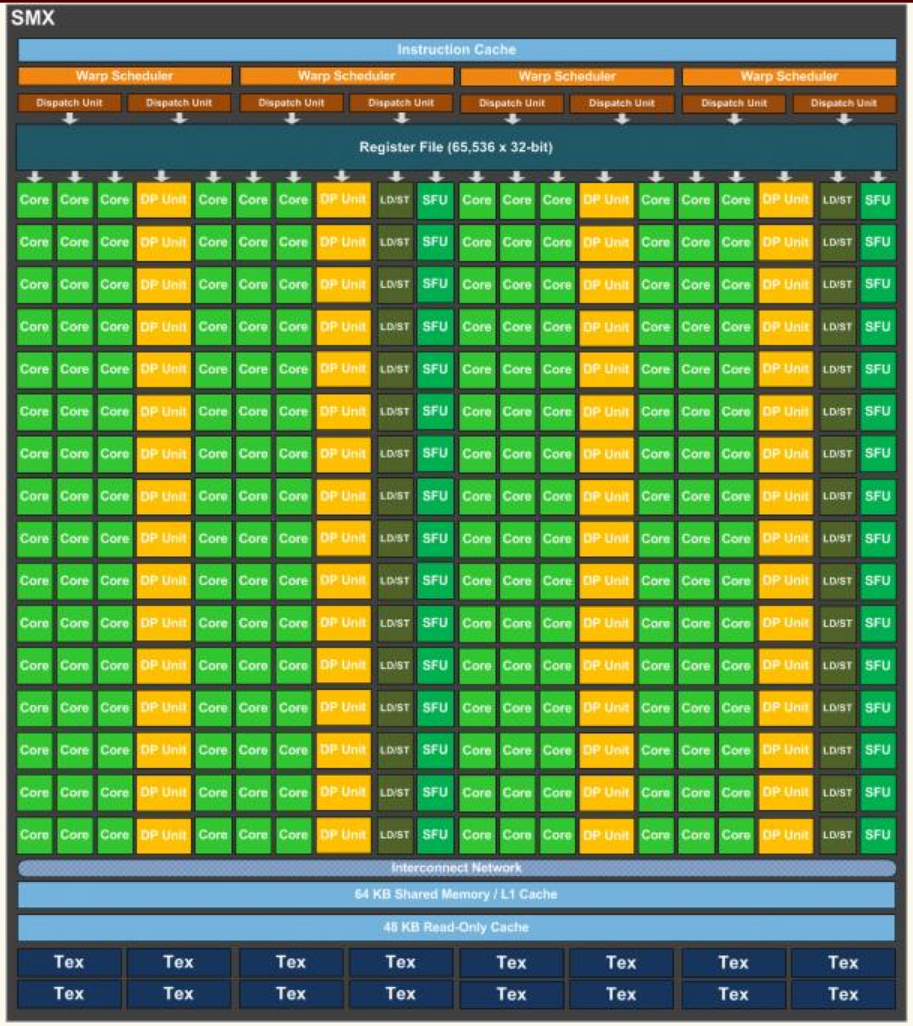


https://www.quantalea.net/media/_doc/2/7/manual/index.html?GPUHardwareImplementation.html

Memory	Bandwidth
Register Memory	~8,000 GB/s
Shared Memory	~1,600 GB/s
Global Memory	~177 GB/s
Mapped Memory	~8 GB/s

- A small sub-graph in shared memory for each block
- Only 48K bytes per SM (Streaming Multiprocessor) → This li the THRESHOLD for small sub-graphs
- Maximum concurrent blocks per SM is 16 (then each block ha only 3k shared memory) → this limits the THRESHOLD for s sub-graphs too.

Occupancy and Performance



- Tesla K20: 192 cores per SM, max active 64 warps (32 threads/warp), max resident 16 blocks
- More warps and/or blocks than 192 cores can actually execute → fast context switching to hide memory access latency.
- Occupancy: active warps/64 → higher occupancy is likely hide more memory latency, but not necessary implies better performance (see *Better Performance at Lower Occupancy*).
- Our program aims to have occupancy of 25%: 16 resident blocks so that 16 small graphs (<3K each) can be processed concurrently in each SM.

http://www.pcper.com/reviews/Graphics-Cards/NVIDIA-Reveals-GK110-GPU-Kepler-71B-Transistors-15-SMX-Units

Configurations for GPU Processing

- 2000 blocks
  - Allow maximum number of small graphs to be processed concurrently (Tesla K20 allows 13*16=208 blocks resident in shared memory)
  - reduce the overhead of memory copy
  - possibly reduce the impact of imbalanced computations among blocks.
- 32 threads per block
  - A warp for easy synchronization.
  - Number of threads per block determines the amount of shared memory needed by a block → affect the occupancy.
- Each block processes a small graph with vertices <= 80 (if available)
  - Number of vertices of a small graph determines the amount of shared memory needed by a block → affect the occupancy.

Occupancy and Performance of Our Program

- Current occupancy of our program is 25% (16 active warps/blocks per SM).

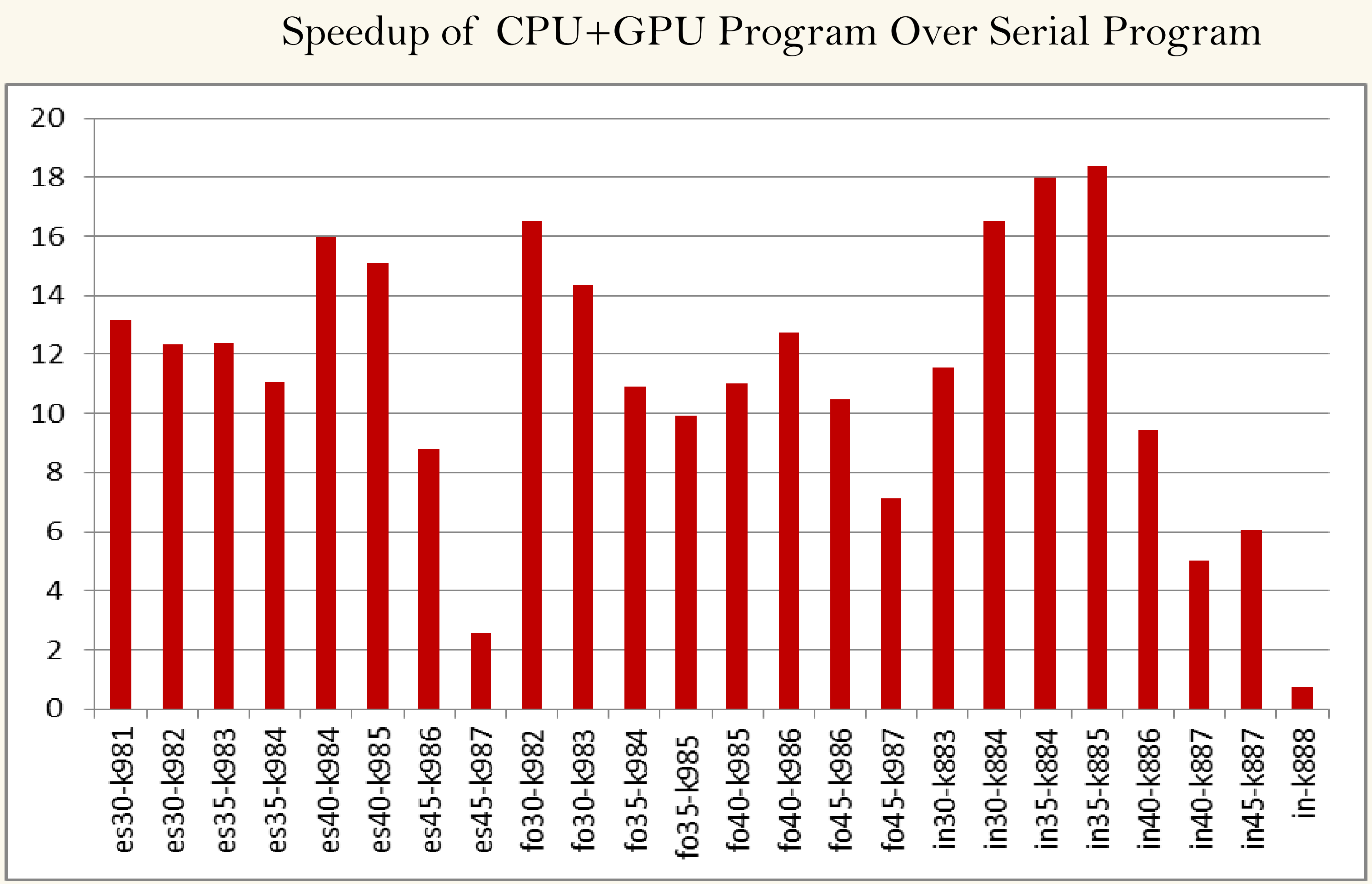
- Our tests show this configuration achieves the best performance in general
  - A particular graph may achieve better performance with different configurations.
  - Some graphs have bottleneck on cpu side while some graphs have bottleneck on gpu side.

- Number of registers per thread and shared memory per block → how many blocks can run concurrently
  - nvcc ---ptxas-options=-v provides such information
  - Our program: 69 registers per thread, and 2976 bytes shared memory per block

Test Data

- Provided by Michael Langston and Gary Rogers.
- Created from real biological data related to
  - Folic acid deficiency effect on colon cancer cells (fo30, fo35, fo40, fo45)
  - Low concentrations of 17 beta-estradiol effects on breast cancer cell line (es30, es35, es40, es45)
  - Interferon receptor deficient lymph node B cell response to influenza infection (in30, in35, in40, in45)
- Each test data is tested with two k values:
  - k = t when there is a vertex cover of at most t vertices
  - k = t-1 where there are no vertex covers of at most t-1 vertices.

Test Results



Graph-k	Serial (seconds)	CPU+GPU (seconds)
est30-k981	11679.7356	885.6192
est30-k982	3501.296	283.3502
est35-k983	3143.042	254.1331
est35-k984	327.1932	29.5715
est40-k984	849.564	53.2437
est40-k985	113.1256	7.5073
est45-k986	295.9202	33.608
est45-k987	6.523	2.5577
fo30-k982	31337.2464	1895.7333
fo30-k983	1782.8224	124.2507
fo30-k984	6.709	1.0532
fo35-k984	7110.7688	649.9995
fo35-k985	277.7758	27.9412
fo40-k985	2208.4914	200.178
fo40-k986	156.8424	12.2987
fo45-k986	574.8578	54.863
fo45-k987	44.4756	6.2211
inf30-k883	1574.9104	136.54
inf30-k884	366.9082	22.1952
inf35-k884	426.7548	23.6902
inf35-k885	404.494	21.9723
inf40-k886	156.3758	16.5719
inf40-k887	16.1454	3.2252
inf45-k887	75.0088	12.4427
inf45-k888	0.9464	1.3148

Test Environment

- Ada X86-64 Cluster at Texas A&M University (862 nodes)
    - Intel E5-2670 v2 (peak performance: 400 GFLOPS)
    - Nvidia K20 (peak performance: 3.52 TFLOPS)

- CUDA 6.5.14 and Intel Compiler 2013_sp.1.3.174

Acknowledgements

- Support from NSF grant 1442734.
- Support from High Performance Research Computing at Texas A&M University
- Test data provided by Michael Langston and Gary Rogers
- Discussions on CUDA with Robert Crovella

DIVISION OF RESEARCH
TEXAS A&M UNIVERSITY

Q & A

Thank You!