

# Tradeoffs Between Synchronous and Asynchronous Execution in PowerGraph

Joshua Send  
Trinity Hall  
28 November, 2017

# PowerGraph <sup>[1]</sup>

- Recall: GraphLab => PowerGraph
- Motivation: large natural graphs
  - Follow power law distribution  $P(d) \propto d^{-\alpha}$
- PowerGraph contributions
  - Generalized vertex programs
  - Vertex Cuts
  - Parallel locking

# PowerGraph

- Recall: Huge array of system parameters
  - Edge distribution
    - Random
    - Heuristic – oblivious (estimate from local state only)
    - Heuristic – coordinated (distributed table of vertex replication)
  - Execution Strategies
    - Synchronous supersteps
    - Full Asynchronous
    - Asynchronous + serializable

# 2015: PowerSwitch [2]

- Extends PowerGraph with a new switching mode
- Choose execution mode (sync/async) based on current problem
- Async
  - Favors CPU-heavy workload
  - High communication costs (no barrier = no batching)
  - Heavy contention for shared resources
    - Favors problems with few active vertices at a time
  - Some problems (graph coloring) only converge in Async
- Sync
  - Many active vertices and scales well with graph size
  - Favors lightweight computation & heavy IO

# PowerSwitch

- Instrument system to measure throughput
- Also estimate/sample convergence rates
- Use Neural network or online sampling to measure throughput of mode not currently in
- Switch according to some heuristics and the throughput & convergence rates

# Project

- Check results from the PowerSwitch paper – source was found online
- Modify heuristics/add new parameter to manually bias execution toward one paradigm or the other
- Their experiments were run with relatively large clusters – 48 machines. Attempt running with smaller quantities, compare results
  - Expect Synchronous to be used most of the time

# Current Status

- GraphLab/GraphChi => Turi => Apple
- graphlab.org no longer a valid domain... dependencies used to be hosted here
- Have to manually modify CmakeLists to resolve these issues...

```
CMake Error at eigen-stamp/download-eigen.cmake:27 (message):
  error: downloading 'http://graphlab.org/deps/eigen_3.1.2.tar.bz2' failed

  status_code: 6
  status_string: "Couldn't resolve host name"
  log: Curl_ipv4_resolve_r failed for graphlab.org

Couldn't resolve host 'graphlab.org'

Closing connection 0
```

# References

- 1) Gonzalez, Joseph E., et al. "PowerGraph: Distributed Graph-Parallel Computation on Natural Graphs." OSDI. Vol. 12. No. 1. 2012.
- 2) Xie, Chenning, et al. "Sync or async: Time to fuse for distributed graph-parallel computation." ACM SIGPLAN Notices 50.8 (2015): 194-204.