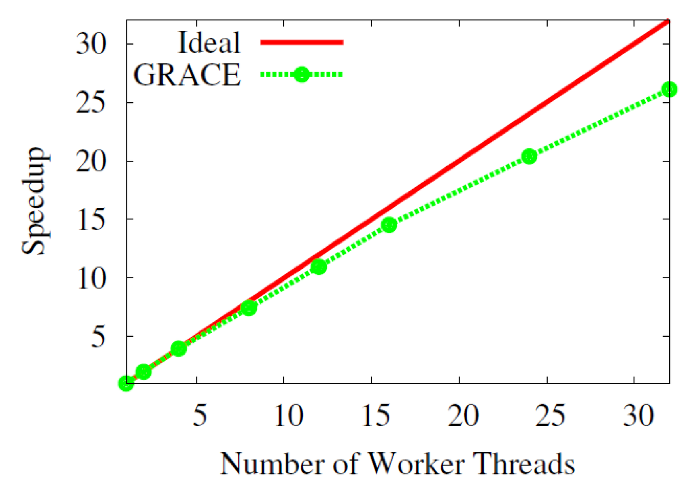
Fast Iterative Graph Computation with Block Updates

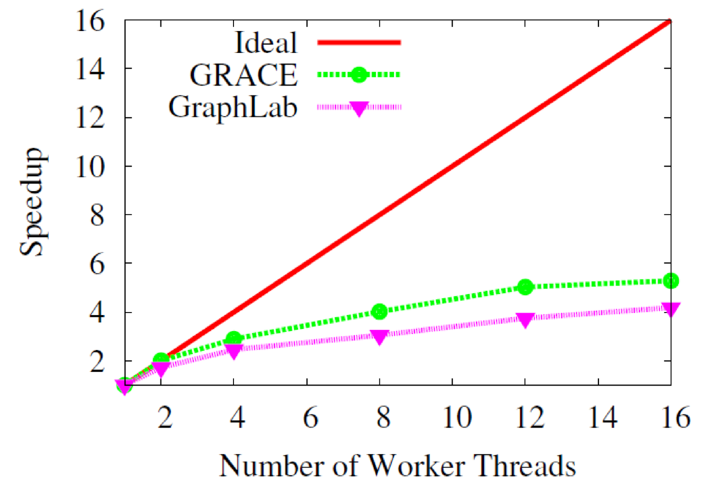
Wenlei Xie*, Guozhang Wang+, David Bindel*, Alan Demers* and Johannes Gehrke*

*Cornell University, +LinkedIn

Multicore Speedup for Graph Applications

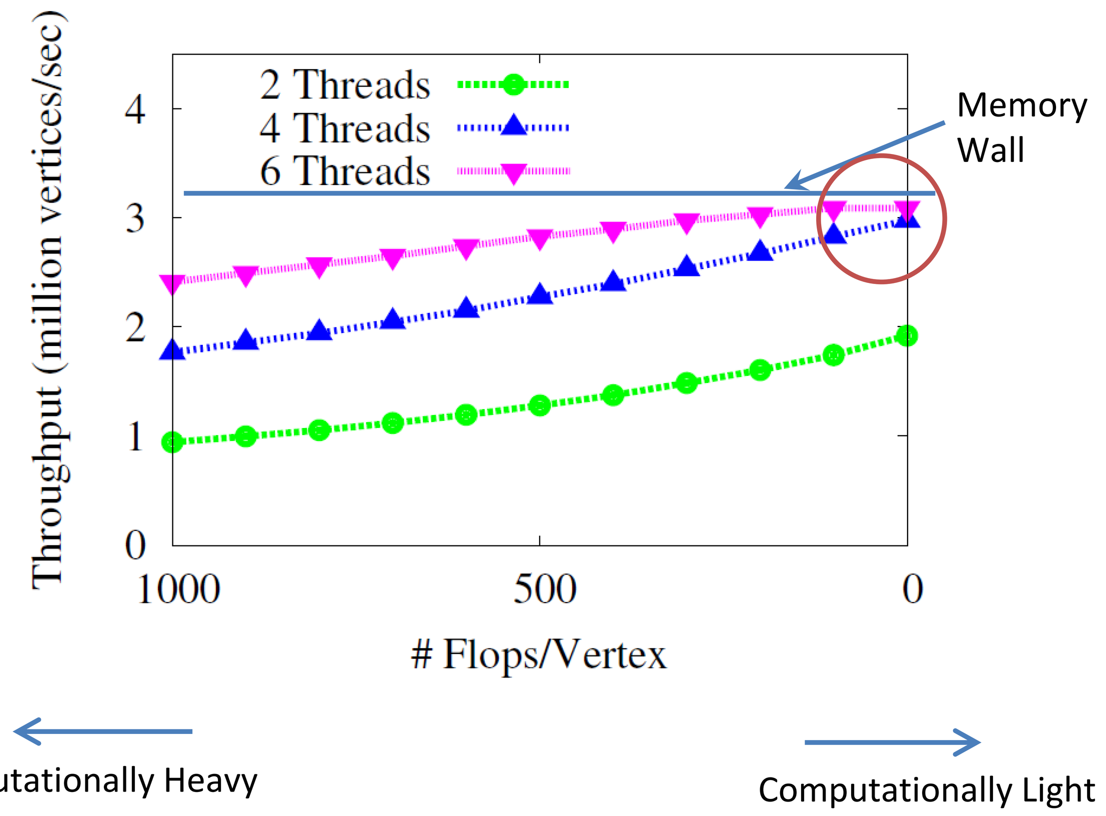


Belief Propagation

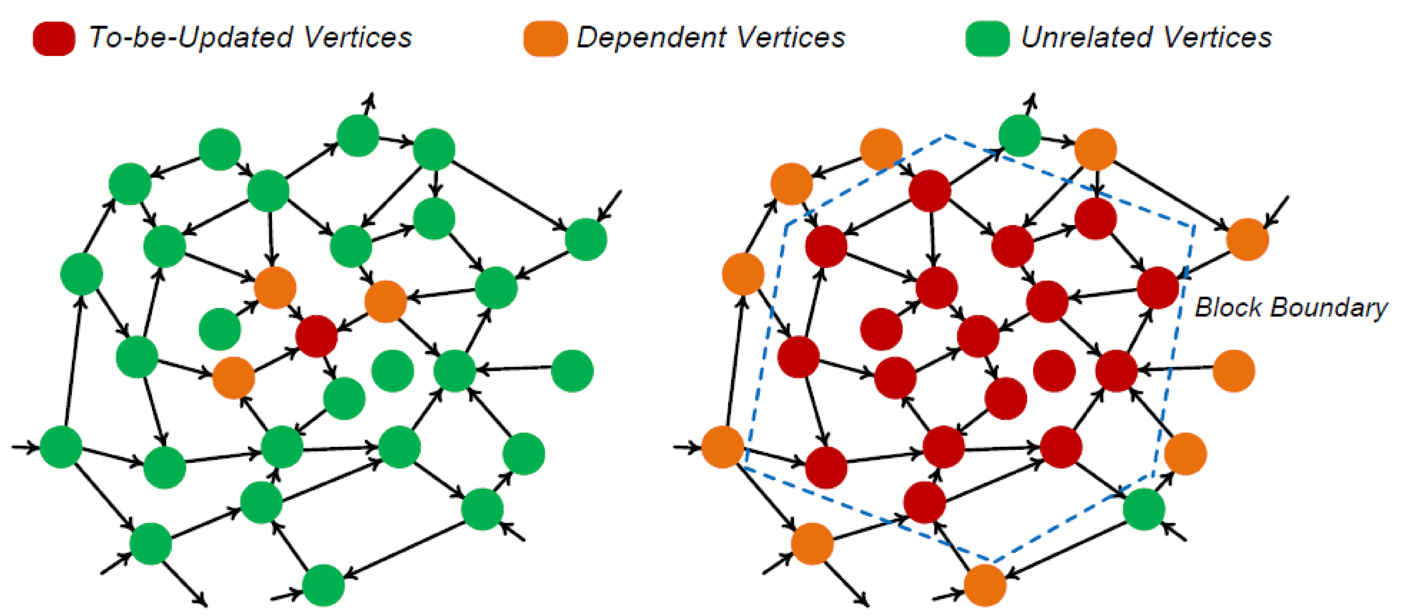


PageRank

Computation Load: Heavy vs. Light



Vertex vs. Block Update



(a) Vertex-Oriented Computation (b) Block-Oriented Computation

Block-Oriented Computation

- Block Formulation
  - –Block: Closely connected subgraph
  - –Graph is pre-partitioned into disjoint blocks
  - –Efficient software: (e.g. METIS)
- Block Update Function

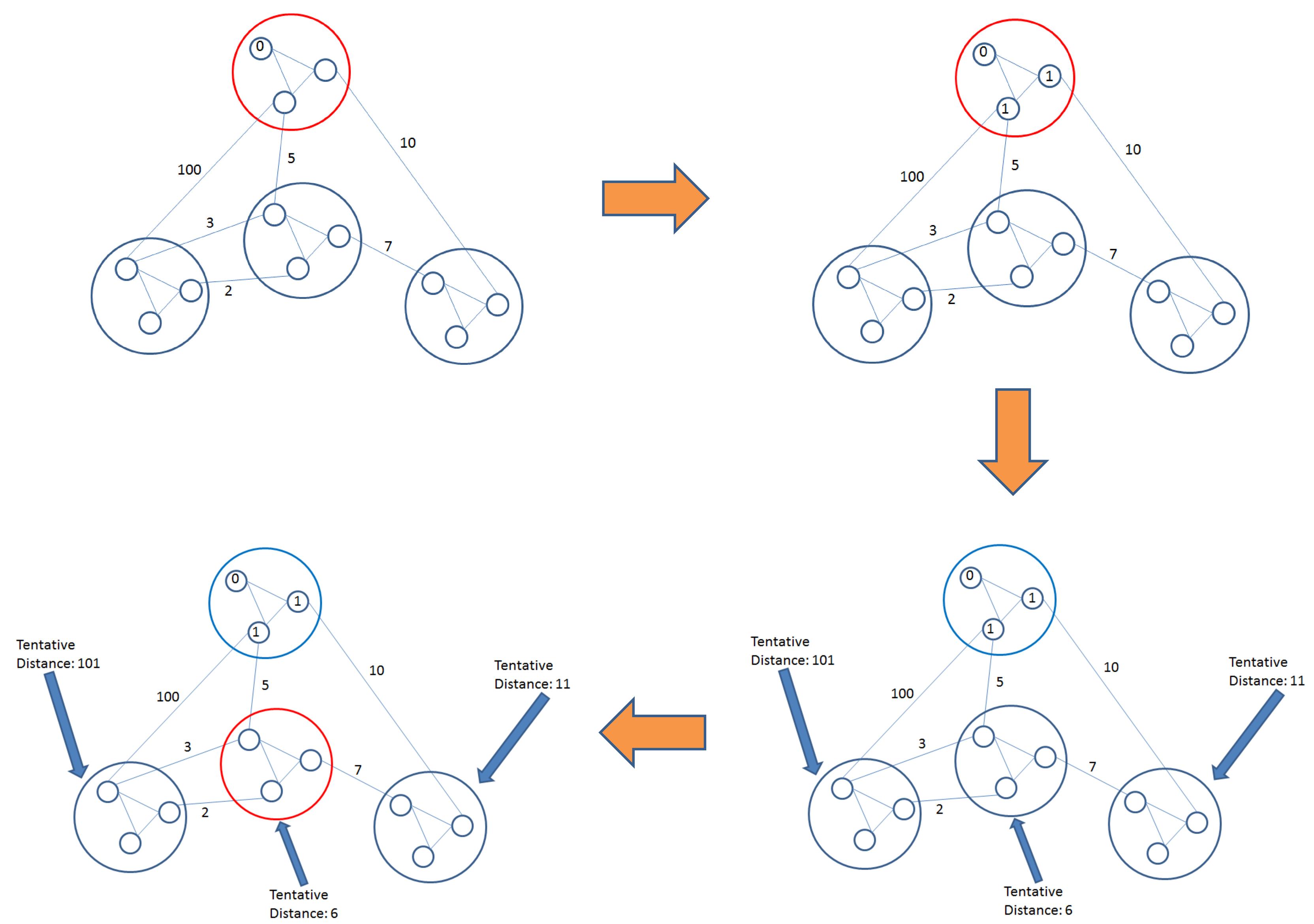
  $S_B^{\text{new}} = \text{BlockUpdate}(S_B^{\text{old}}, S_{NV(B)}, S_{NE(B)}),$

  - –Naturally extends the vertex update function

Two-Level Scheduling

- Define block update as iteratively applying vertex update
  - –BlockUpdate = VertexUpdate × InnerScheduler
  - –Block-Level Scheduler
- Benefits
  - –Better Cache Utilization
  - –Reduced Scheduling Overhead

Example: Shortest Path

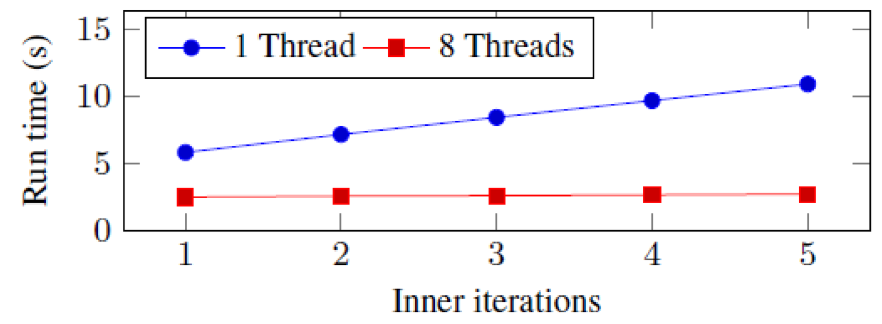


Datasets and Applications

Data Set	Vertices $\times 10^3$	Edges $\times 10^3$	Partition Time (s)	Application
DBLP	968	7,050	38	PPR
Web-Google	876	5,105	34	PPR
LiveJournal	4,848	68,994	659	SSSP
3D Grid	1,728	9,858	N/A	Etch Sim
UK02	18,520	298,114	1034	PPR

Microbenchmark: Cache Performance

Scheduler	Time	# Updates	# LLC Misses
Non Cache-Aware	9.52	34,152,807	197,500,000
Cache-Aware	5.15	34,152,807	37,500,000



Experimental Results