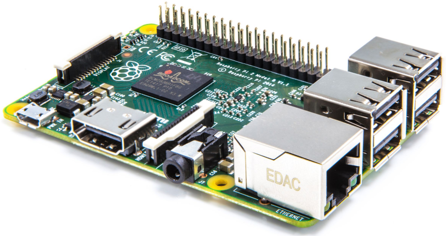
Massive Graph Processing on Nanocomputers

Bryan Rainey and David F. Gleich
Department of Computer Science, Purdue University

Nanocomputers

Raspberry Pi 2	Scaleway Server	Kangaroo
$35	2.99 euro/month	$99.99
4-core ARMv7	4-core ARMv7	4-core Intel Atom
1GB RAM	2GB RAM	2GB RAM
2TB HDD ($89.99)	50GB SSD	32GB SSD



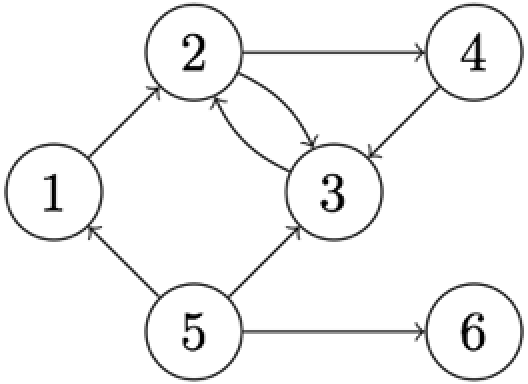
Focus of Today’s Talk

We can use nanocomputers to compute PageRank and connected components reasonably quickly on graphs with a few billion edges.

What are the bottlenecks?

Graph Processing

Algorithms: PageRank and Connected Components.



Maintain a piece of data for each node in a vector.

For each iteration,

    For each edge,

        Update each node's entry in the vector
        by sending information along the edge.

Implementation: https://github.com/brainey421/badjgraph

Datasets

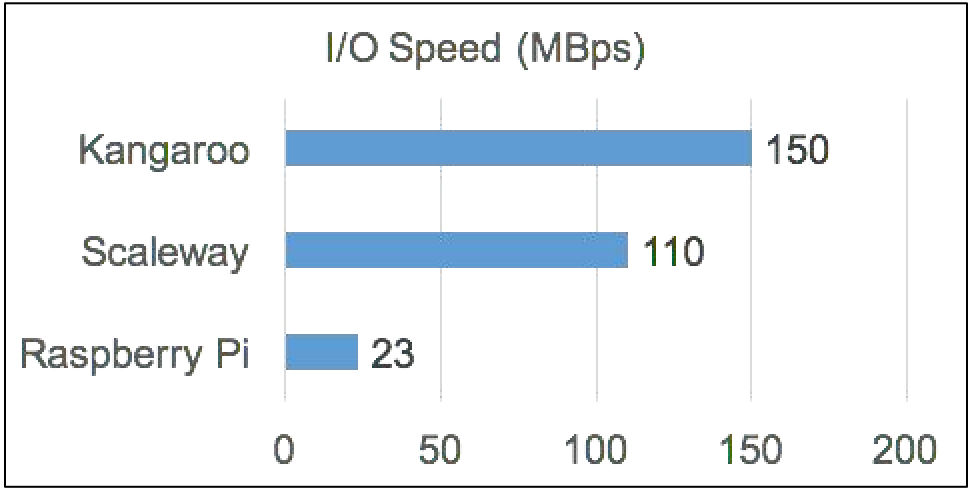
From the Laboratory for Web Algorithmics at the University of Milan.

Converted from the compressed WebGraph format into a binary adjacency list format.

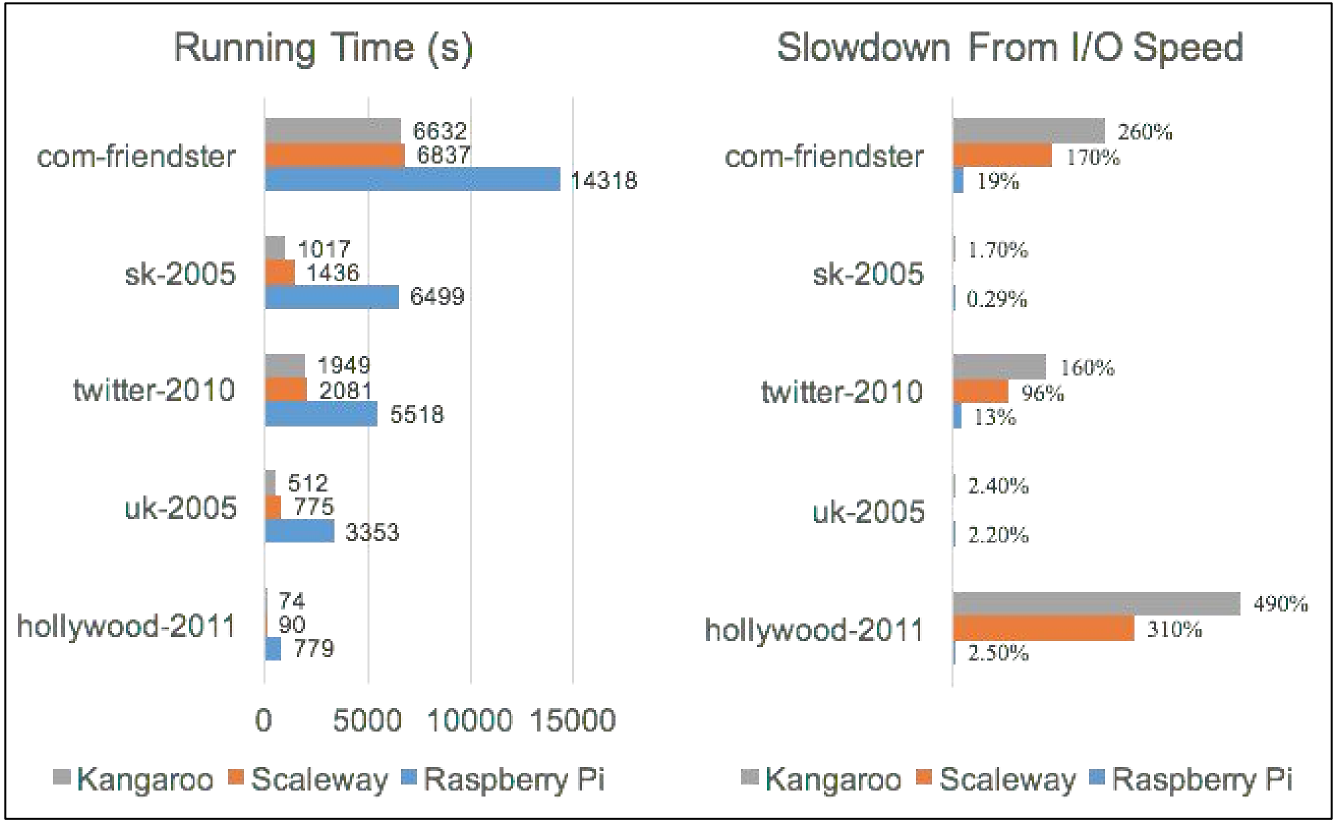
Graph	Nodes	Edges	Size
hollywood-2011	2,180,759	228,985,632	0.8612GB
uk-2005	39,459,925	936,364,282	3.635GB
twitter-2010	41,652,230	1,468,365,182	5.625GB
sk-2005	50,636,154	1,949,412,601	7.451GB
com-friendster	65,608,366	3,612,134,270	13.70GB

Streaming Graphs

Baseline measurement of performance.



20 Iterations of PageRank



What are the potential bottlenecks?

1. **I/O Bandwidth:** 23MBps (RPi), 110MBps (Scaleway), 150MBps (Kangaroo).
2. **Memory Bandwidth*:** 810MBps, 400MBps, 2100MBps.
3. **Memory Latency:** 250 ns, 130 ns, 150 ns.

*Cache line size: 32B, 32B, 64B.

Graph	Size	% Remote Accesses
hollywood-2011	0.8612GB	8.7%
uk-2005	3.635GB	3.7%
twitter-2010	5.625GB	35% and 34%
sk-2005	7.451GB	7.9%
com-friendster	13.70GB	85%

Performance Model

$$T = \max\left\{\frac{S}{\beta_1}, \frac{mRC}{\beta_2}, mRL\right\}$$

Running time per iteration = max {

(size of graph) / (**I/O bandwidth**),

(number of remote accesses) * (cache line size) / (**memory bandwidth**),

(number of remote accesses) * (**memory latency**)

}

- Raspberry Pi is usually bottlenecked on I/O bandwidth.
- Scaleway and Kangaroo are usually bottlenecked on memory latency.
- The two web crawls are always bottlenecked on I/O bandwidth.
- Friendster is always bottlenecked on memory latency.

Performance Model Evaluation

I/O bandwidth and **memory latency** generally predict PageRank's running time on nanocomputers within a factor of 2.

Graph	Raspberry Pi Predicted/Actual	Scaleway Predicted/Actual	Kangaroo Predicted/Actual
hollywood-2011	98%	57%	120%
uk-2005	97%	96%	99%
twitter-2010	91% and 94%	67% and 73%	110% and 120%
sk-2005	100%	97%	100%
com-friendster	110%	120%	230%

Final Thoughts

Nanocomputers sometimes fall short in **scalability** and **flexibility**.

BUT they could save **money** and **energy** (7.0 W Raspberry Pi, 9.3 W Kangaroo).

Thank You: David F. Gleich and the Network/Matrix Group at Purdue University.

Related Work:

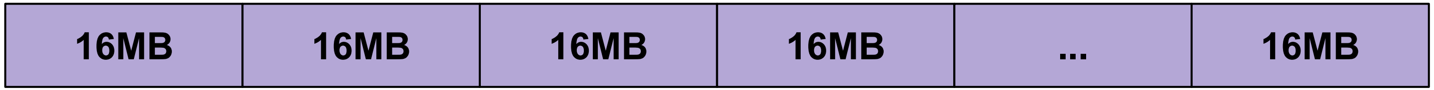
Kyrola et al., “GraphChi: Large-scale graph computation on just a PC,” 2012.
McSherry et al., “Scalability! But at what COST?” 2015.

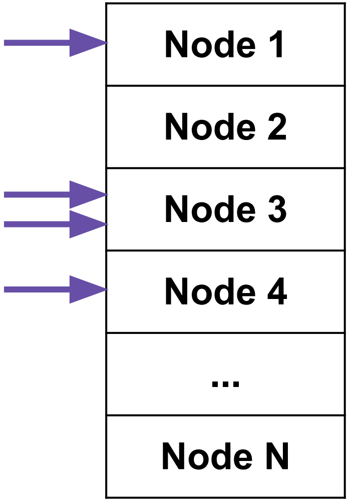
System Comparisons for the Twitter Graph

System	Cores	Memory	PageRank	Components
Raspberry Pi	4	1GB	5516 s	1841 s
Scaleway	4	2GB	2008 s	490 s
Kangaroo	4	2GB	1391 s	298 s
GraphChi	2	8GB	3160 s	–
Stratosphere	16	192GB	2250 s	950 s
X-Stream	16	64GB	1488 s	1159 s
Vertica	48	192GB	1287 s	378 s
Giraph	128	1088GB	596 s	200 s
GraphX	128	1088GB	419 s	251 s
GraphLab	128	1088GB	249 s	242 s
Ringo	80	1024GB	121 s	–
Galois	48	1024GB	90 s	–
EmptyHeaded	48	1024GB	77 s	–

Parallelization

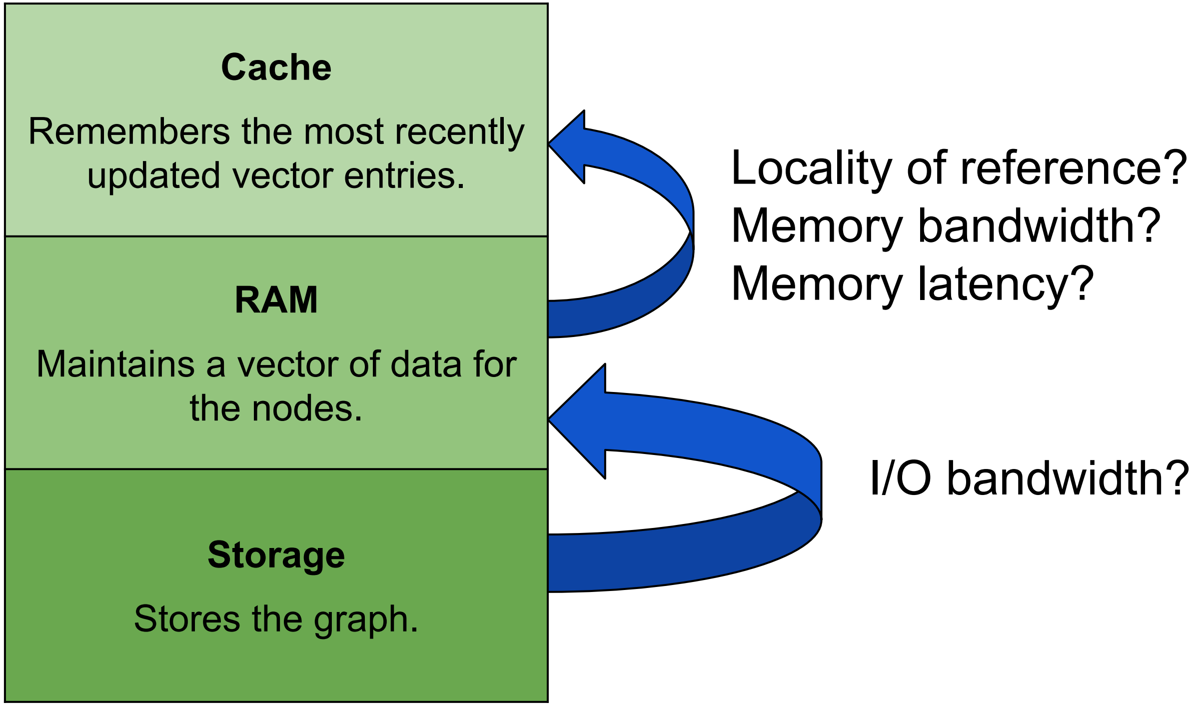
Split the graph file into blocks to divide among the threads.



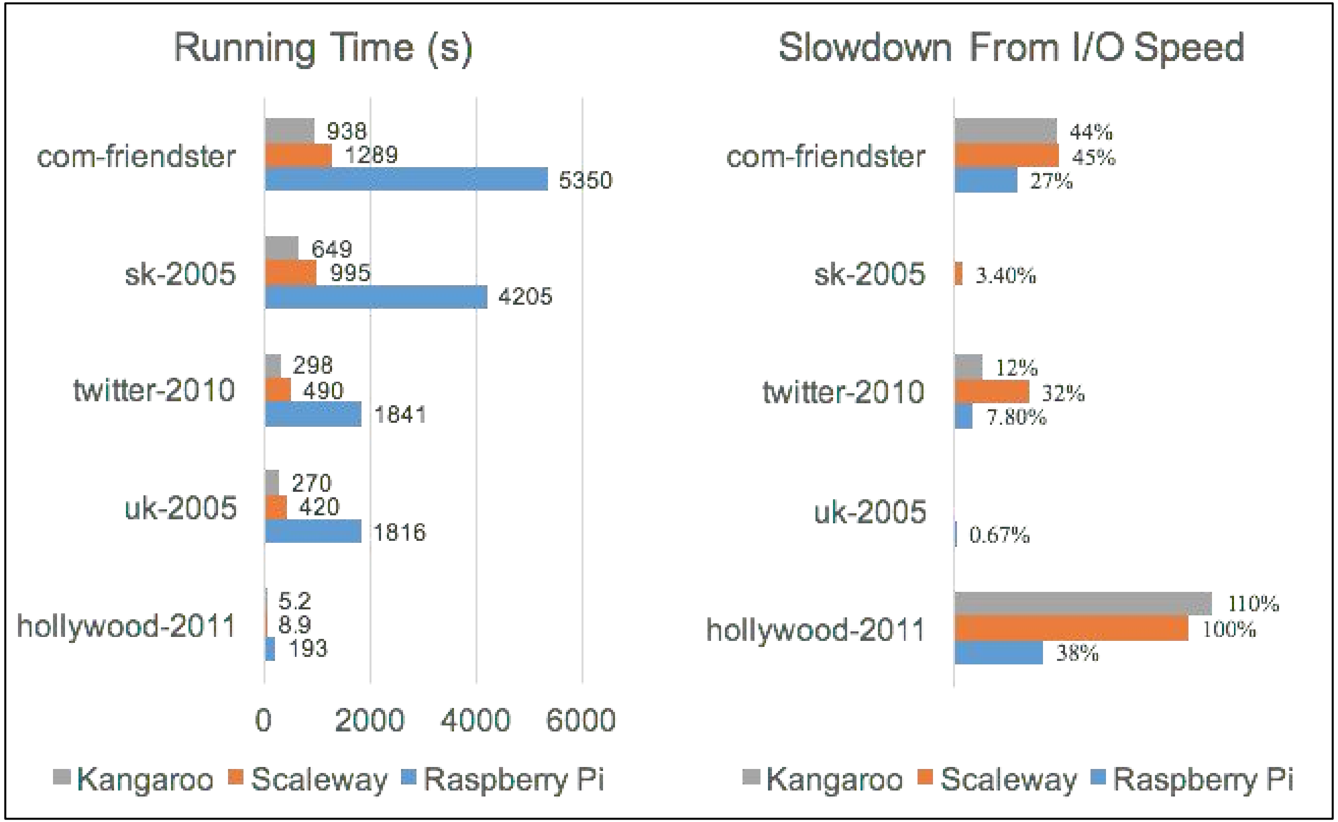


Deal with a potential race condition while updating the vector.

Memory Hierarchy



Connected Components



Massive Graph Processing on Nanocomputers

Idea: Write custom implementations of iterative graph algorithms for tiny, cheap computers like the Raspberry Pi.

Main limitation: Tiny computers don’t have enough RAM to store large graphs, so we stream graphs from storage.

Brief result: The Kangaroo ($100, 4-core Intel Atom) can compute PageRank on the Twitter graph as quickly as some graph-processing systems running on 12-48 cores.

Bottlenecks? Depends on the machine and the size/structure of the graph. It’s a matter of moving data around.