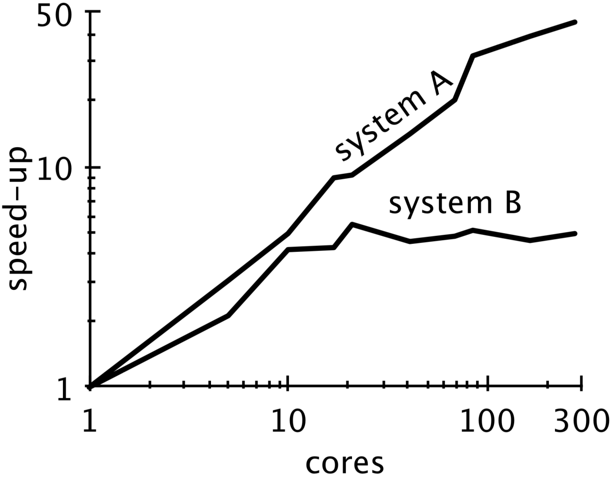
Scalability! but at what COST?

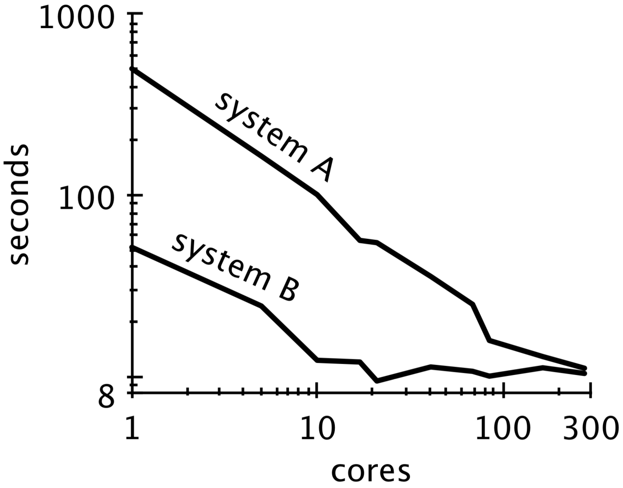
Frank McSherry et al., HotOS 2015

Presentation by:
SB Ramalingam Santhanakrishnan
K Kleeberger
Z Sun
C Zhu
(Group 5)

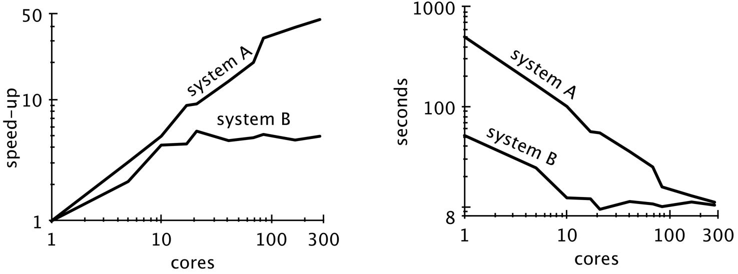
Which system is better, A or B?



What about now, A or B?

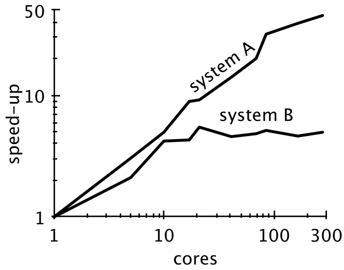


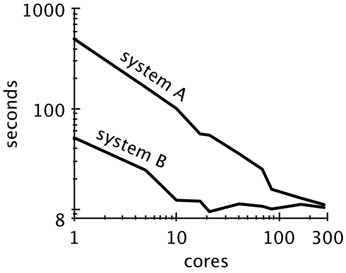
Question in hand



- Scalability is often touted as an essential attribute.
- Absolute performance is not related to scalability.

Question in hand





- Scalability is often touted as an essential attribute.
- Absolute performance is not related to scalability.

To what degree are scalable systems truly improving performance, as opposed to parallelizing overheads introduced?

How can we measure?

"What you can't measure, you can't improve"

How can we measure?

"What you can't measure, you can't improve"

COST - **C**onfiguration that **O**utperforms a **S**ingle **T**hread

Why measure against a single thread?

- Distributed systems can have huge overheads.
- Most systems have *unbounded* COST!
- More optimizations can be applied

A case study - *Graph* Big Data Systems

Why choose Graph?

- Non-trivial to parallelize
- Data-driven
- No structure
- More time to pass information

A case study - *Graph* Big Data Systems

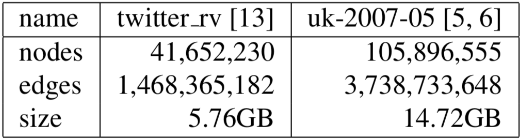
Why choose Graph?

- Non-trivial to parallelize
- Data-driven
- No structure
- More time to pass information

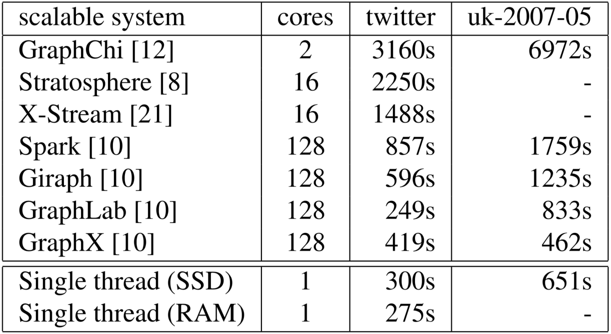
Vertex Centric

- Program from a vertex perspective
- Only messages from other vertices as input
- Useful for PageRank and other graph algos
- "Think Like A Vertex", Pregel, etc.

PageRank (20 Iterations)



name	twitter_rv [13]	uk-2007-05 [5, 6]
nodes	41,652,230	105,896,555
edges	1,468,365,182	3,738,733,648
size	5.76GB	14.72GB



scalable system	cores	twitter	uk-2007-05
GraphChi [12]	2	3160s	6972s
Stratosphere [8]	16	2250s	-
X-Stream [21]	16	1488s	-
Spark [10]	128	857s	1759s
Giraph [10]	128	596s	1235s
GraphLab [10]	128	249s	833s
GraphX [10]	128	419s	462s
Single thread (SSD)	1	300s	651s
Single thread (RAM)	1	275s	-

Label Propagation (Connected Components)

A common machine learning technique

scalable system	cores	twitter	uk-2007-05
Stratosphere [8]	16	950s	-
X-Stream [21]	16	1159s	-
Spark [10]	128	1784s	$\geq$ 8000s
Giraph [10]	128	200s	$\geq$ 8000s
GraphLab [10]	128	242s	714s
GraphX [10]	128	251s	800s
Single thread (SSD)	1	153s	417s

More Optimization - Data Layout

- The order in which edges are presented affects performance.
- Hilbert order vs Vertex order.

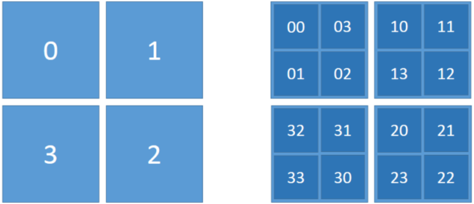
[1] More at https://bigdataatsvc.wordpress.com/2013/07/02/graph-analysis-and-hilbert-space-filling-curves/

More Optimization - Data Layout

- The order in which edges are presented affects performance.
- Hilbert order vs Vertex order.

Hilbert Curves - Cleverly ordering the edges[1]

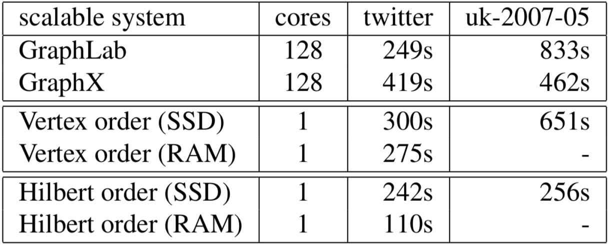
- Assume that edges are stored in an adjacency matrix
- Recursively partitions the matrix
- Excellent for memory locality + parallelizing



[1] More at https://bigdataatsvc.wordpress.com/2013/07/02/graph-analysis-and-hilbert-space-filling-curves/

More Optimization - Data Layout

- The order in which edges are presented affects performance.
- Hilbert order vs Vertex order.



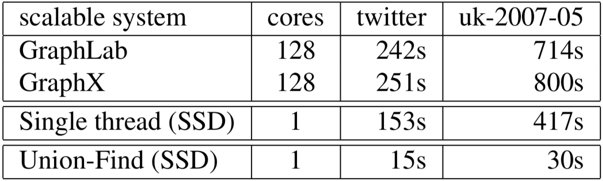
scalable system	cores	twitter	uk-2007-05
GraphLab	128	249s	833s
GraphX	128	419s	462s
Vertex order (SSD)	1	300s	651s
Vertex order (RAM)	1	275s	-
Hilbert order (SSD)	1	242s	256s
Hilbert order (RAM)	1	110s	-

Even More Optimization! - Programming Model

- We are not restricted to "Think like a Vertex" programming model.
- Label propagation is sub-optimal, typically $O(n^3 + mn^2)$
- Use Weighted Union-Find, $O(m \log n)$

Even More Optimization! - Programming Model

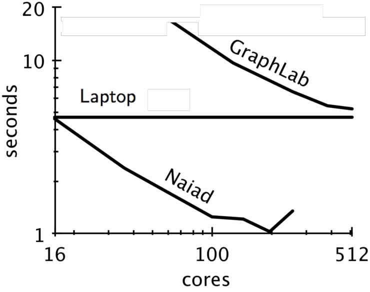
- We are not restricted to "Think like a Vertex" programming model.
- Label propagation is sub-optimal, typically $O(n^3 + mn^2)$
- Use Weighted Union-Find, $O(m \log n)$

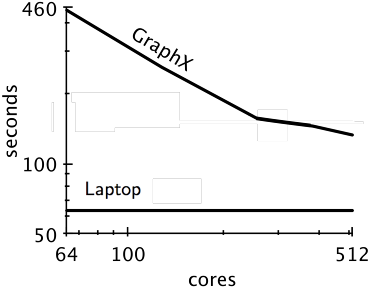


scalable system	cores	twitter	uk-2007-05
GraphLab	128	242s	714s
GraphX	128	251s	800s
Single thread (SSD)	1	153s	417s
Union-Find (SSD)	1	15s	30s

Applying COST

COST is the point of intersection[2]

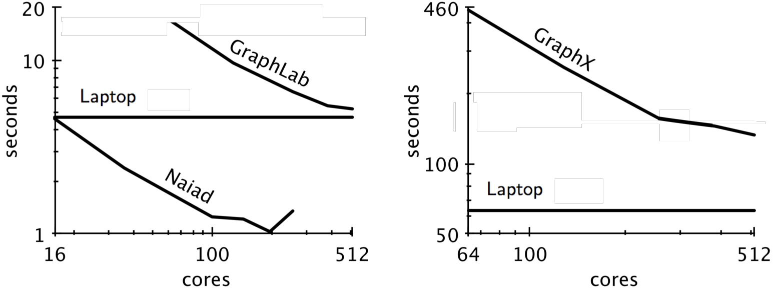




[2] Plots simplified for illustration purposes

Applying COST

COST is the point of intersection[2]



- Naiad has a COST of 16 cores for PageRank
- GraphX has an unbounded COST (does not intersect)

[2] Plots simplified for illustration purposes

Lessons

Lessons

Scalability != Performance

"*Can it scale well?*" - not the right question!

Lessons

Scalability != Performance

"Can it scale well?" - not the right question!

Before you build a big data system,

- Beware of misleading marketing. "One tool for all screws"
- Self-investigation is necessary.
- Use appropriate algorithms.
- Choose to solve the problem locally,
  don't distribute unless absolutely necessary.

Interesting stuff

Further reading:

- Boruvkas algorithm
- Galois and Ligra systems
- Naiad - timely dataflow

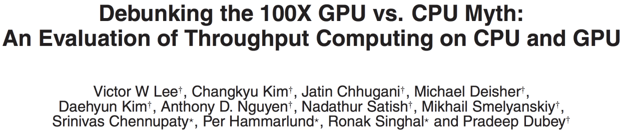
People/Things to follow:

Frank McSherry - https://github.com/frankmcsherry/

Kyle Kingsbury - https://aphyr.com/

Jepsen - https://github.com/jepsen-io/jepsen

Interesting stuff



Debunking the 100X GPU vs. CPU Myth: An Evaluation of Throughput Computing on CPU and GPU

Victor W Lee[†], Changkyu Kim[†], Jatin Chhugani[†], Michael Deisher[†],
Daehyun Kim[†], Anthony D. Nguyen[†], Nadathur Satish[†], Mikhail Smelyanskiy[†],
Srinivas Chennupaty[*], Per Hammarlund[*], Ronak Singhal[*] and Pradeep Dubey[†]

Thank you!

Questions