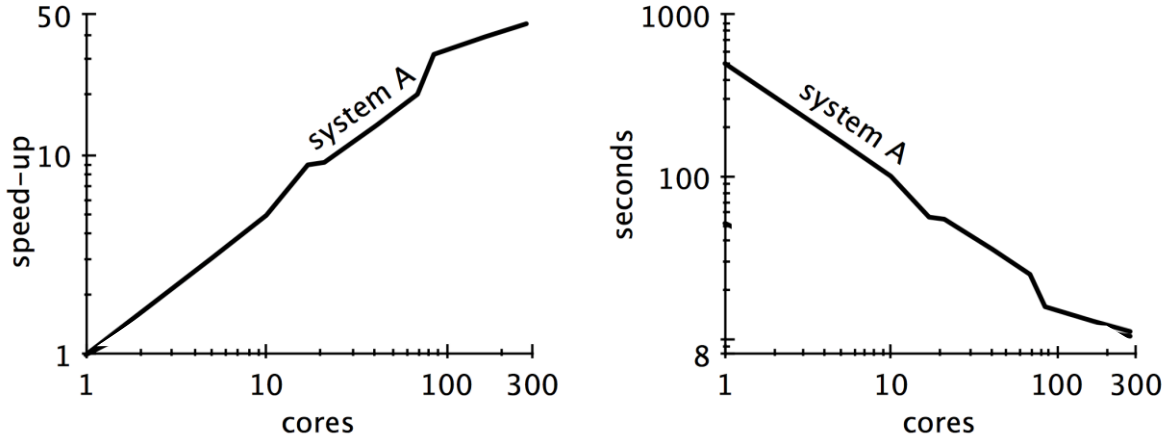
Scalability! But at what COST?

Frank McSherry, Michael Isard, Derek G. Murray

Alex Gubbay

What's Wrong With Distributed Systems Reporting?

- Scalability often touted as the most important feature
- Fail to evaluate absolute performance
- Direct distributed system design towards salability from better systems



NAIAD computation before (system A) and after (system B) optimisation [1]

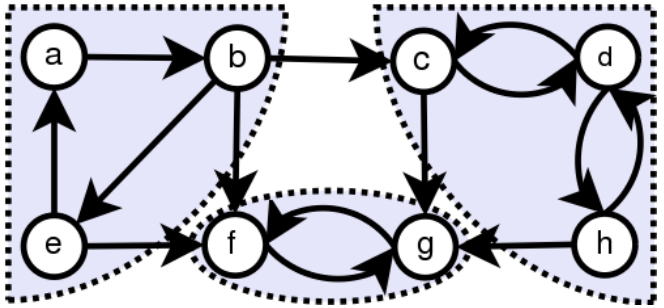
COST – **C**onfiguration that **O**utperforms a **S**ingle **T**hread

- A distributed hardware configuration that outperforms a single threaded implementation.
- Investigate published performance of distributed systems and compare a reasonable implementation on a single core
- Consider total run time

- Some systems have unbounded COST!

Comparisons Against Existing Systems

- PageRank
- Connected Components – Label Propagation
- Implemented in C# on high end 2014 laptop



Two implementations

1. Basic
2. Optimised

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

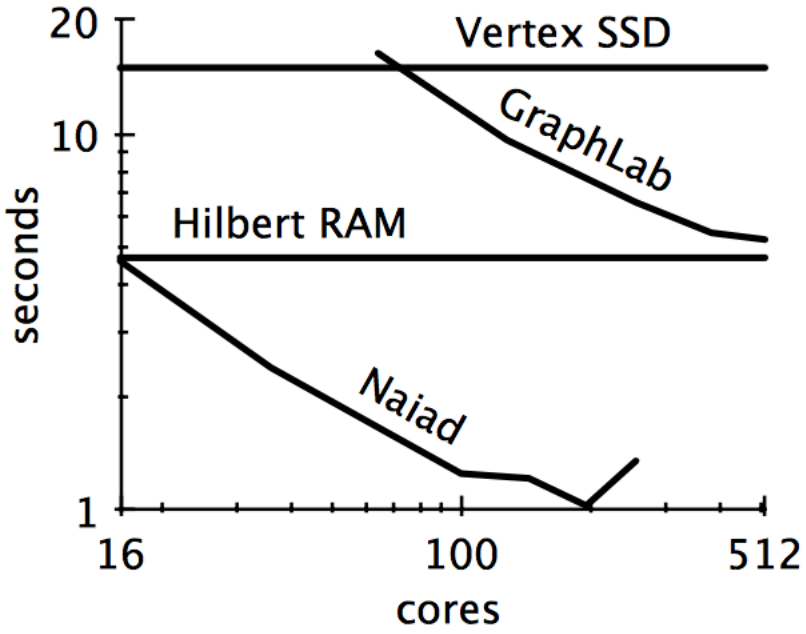
Optimisations of the Baseline

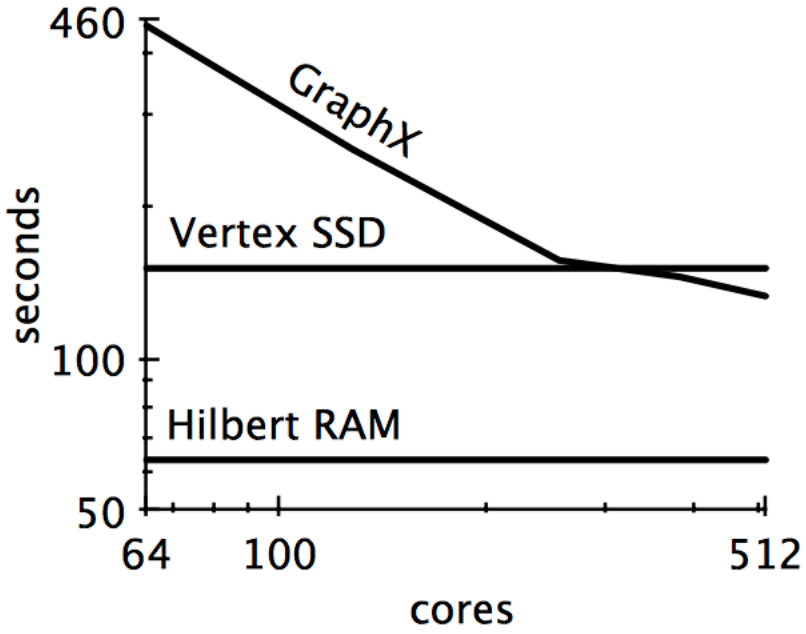
- Better Graph Layout
  - Naïve implementation processes in vertex order
  - GraphLab and GraphX partition to reduce communication between workers [3,4]
  - Ordering on the single thread impacts cache performance
    - Edge ordering described by a Hilbert curve
- Better Algorithm
  - Label Propagation is not an optimal algorithm [5]
  - Union Find runs in $O(m \log n)$

Results and COST Evaluation - PageRank [1,2,3,4]

Scalable System	Cores	Twitter (Secs)	UK Internet 2007 (Secs)
GraphChi	2	3160	6972
Stratosphere	16	2250	-
X-Stream	16	1488	-
Spark	128	857	1759
Giraph	128	596	1235
GraphLab	128	**249**	833
GraphX	128	419	**462**
Single Thread (SSD)	1	300	651
Single Thread (RAM)	1	275	-
Hilbert Order (SSD)	1	242	**256**
Hilbert Order (RAM)	1	**110**	-

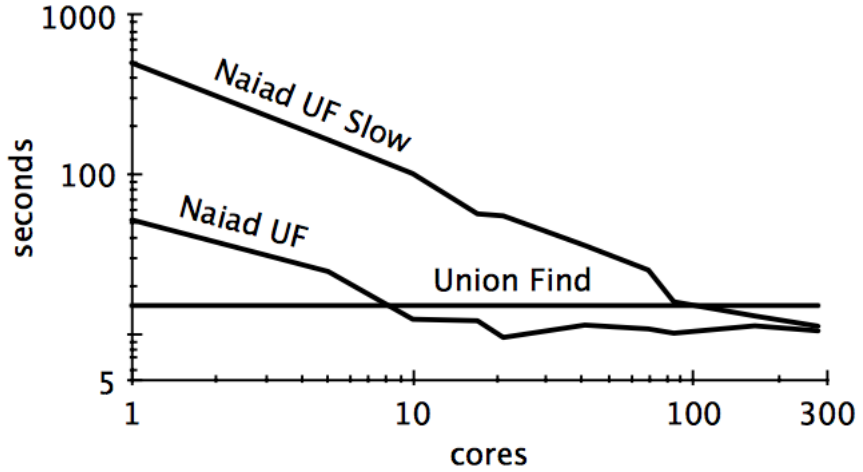
Results and COST Evaluation - PageRank [1,2,3,4]





Results and COST Evaluation – Connected Components

Scalable System	Cores	Twitter (Secs)	UK Internet 2007 (Secs)
GraphLab	128	242	**714**
GraphX	128	**251**	800
Single Thread (SSD)	1	153	417
Hilbert Order (SSD)	1	**15**	**30**



Two NAIAD Implementations for Connected Components

Conclusions

- Clearly need to consider absolute performance
  - Distributed systems have a surprisingly high overhead
  - “Important to distinguish scalability from efficient use of resources” [1]

But

- More to consider than computation time
  - Hardware environment – cluster hardware vs laptop
  - Systems described are prototypes
- Qualitative advantages of distributed system
  - High availability, security, ecosystem integration

Questions?

References

1. *F. McSherry, M. Isard and D. Murray: Scalability! But at what COST? , HOTOS, 2015*
2. Derek G. Murray, Frank McSherry, Rebecca Isaacs, Michael Is- ard, Paul Barham, and Marťın Abadi. *Naiad: A Timely Dataflow System*. SOSP 2013.
3. Joseph E. Gonzalez, Yucheng Low, Haijie Gu, Danny Bickson, Carlos Guestrin. *PowerGraph: Distributed Graph-Parallel Computation on Natural Graphs*. OSDI 2012.
4. Joseph E. Gonzalez, Reynold S. Xin, Ankur Dave, Daniel Crankshaw, and Michael J. Franklin, and Ion Stoica. *GraphX: Graph Processing in a Distributed Dataflow Framework*. OSDI 2014.
5. U Kang, Charalampos E. Tsourakakis, and Christos Faloutsos. *PEGASUS: Mining Peta-Scale Graphs*. ICDM 2009.