

# IMPROVING GRAPHCHI[6] FOR LARGE GRAPH PROCESSING: FAST RADIX SORT IN PRE-PROCESSING[8]

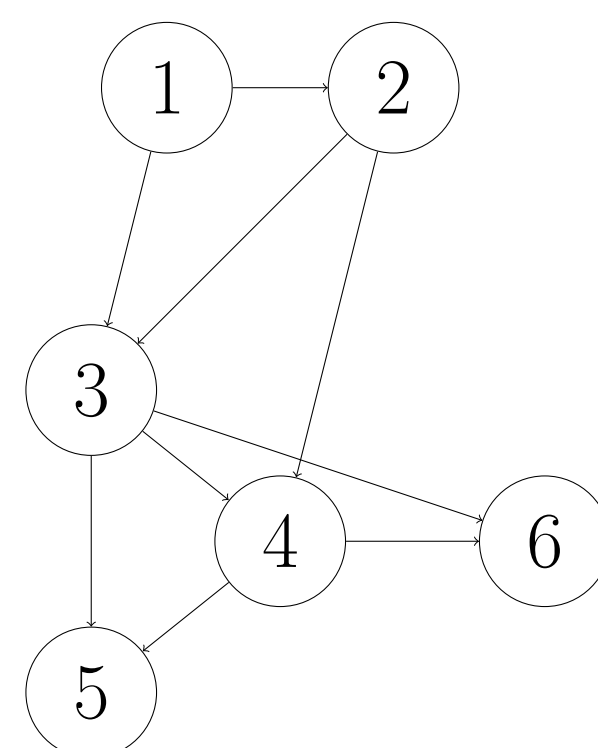
Stuart Thiel, Dr. Greg Butler, Larry Thiel  
Concordia University

## Contributions

- Fast Radix: plug-in replacement
- Improved GraphChi Performance

- Large graph processing
- Examining pre-processing
- An improved LSD Radix Sort
- Experimental Results

## Sample Graph



## Data

- Social media/Web graphs
- 1M–150M vertices, 5M–5.5B edges
- google[7]: web graph
- twitter-2010[5]: social media graph
- live-journal[1]: social media graph
- friendster[9]: social media graph
- hollywood[4, 2]: related actors
- uk-union[3]: web graph

- GraphChi uses Problem Based Benchmark Suite (PBBS) Radix Sort
- PBBS is a Least Significant Digit Radix Sort
  - Scan right-most digit
  - Deal into buckets
  - Consider next digit and repeat

## Background: GraphChi

- Graph processing: sorting is frequent
- Working with large graphs: 10s of billions of edges
- ~20% of server time spent sorting
- Operations such as Page Rank
- Better sorting can save time and money
- Operations such as Page Rank
- Offers competitive performance, one PC
- Improvements can make it even better

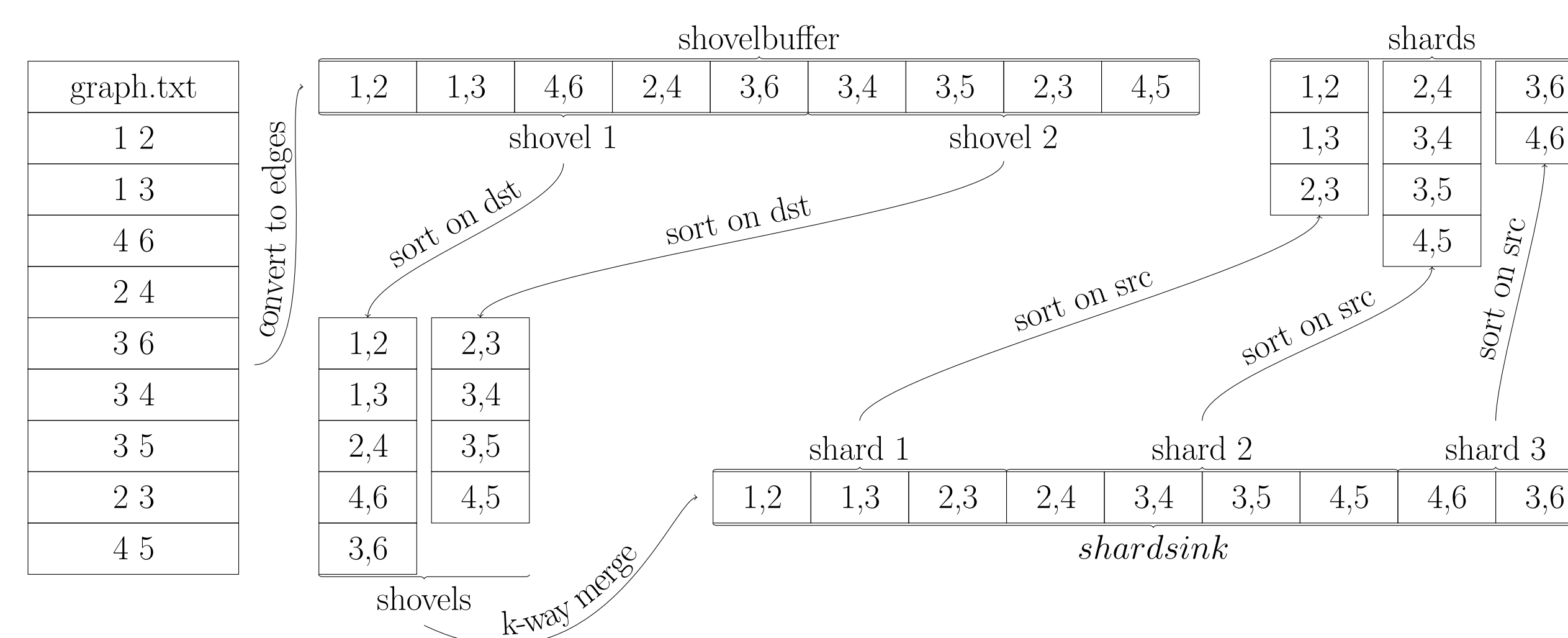
- GraphChi Pre-processing 20–80% of total processing time
- Pre-process once per graph, apply many graph processing operations after
- Sorting happens in pre-processing

- Single-machine graph processing
- Edge-wise graph processing
- Sorts to grid
- Processes graph using parallel sliding windows approach

## GraphChi: Pre-Processing

- parallel sliding windows needs organized data
- Sorting during pre-processing creates a grid to support later processing

## Shard Creation



- Converting graph input into edge format
- Filling a shovel buffer with edges
- Sorting edges by destination into shovels
- Performing a k-way merge of shovels into a shard buffer
- sorting edges in shard buffer into shards

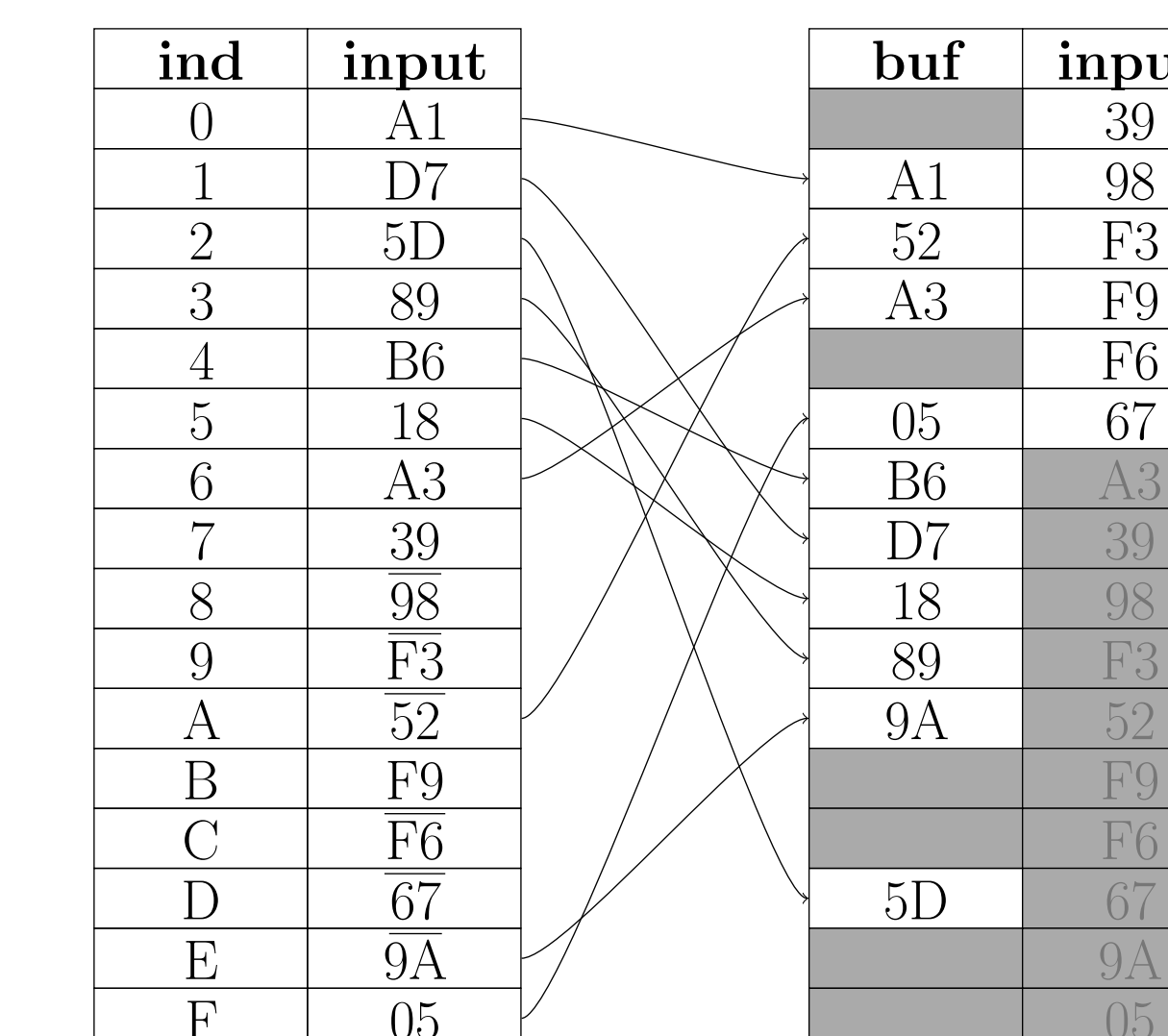
## Fast Radix

- We introduce Fast Radix as an alternative

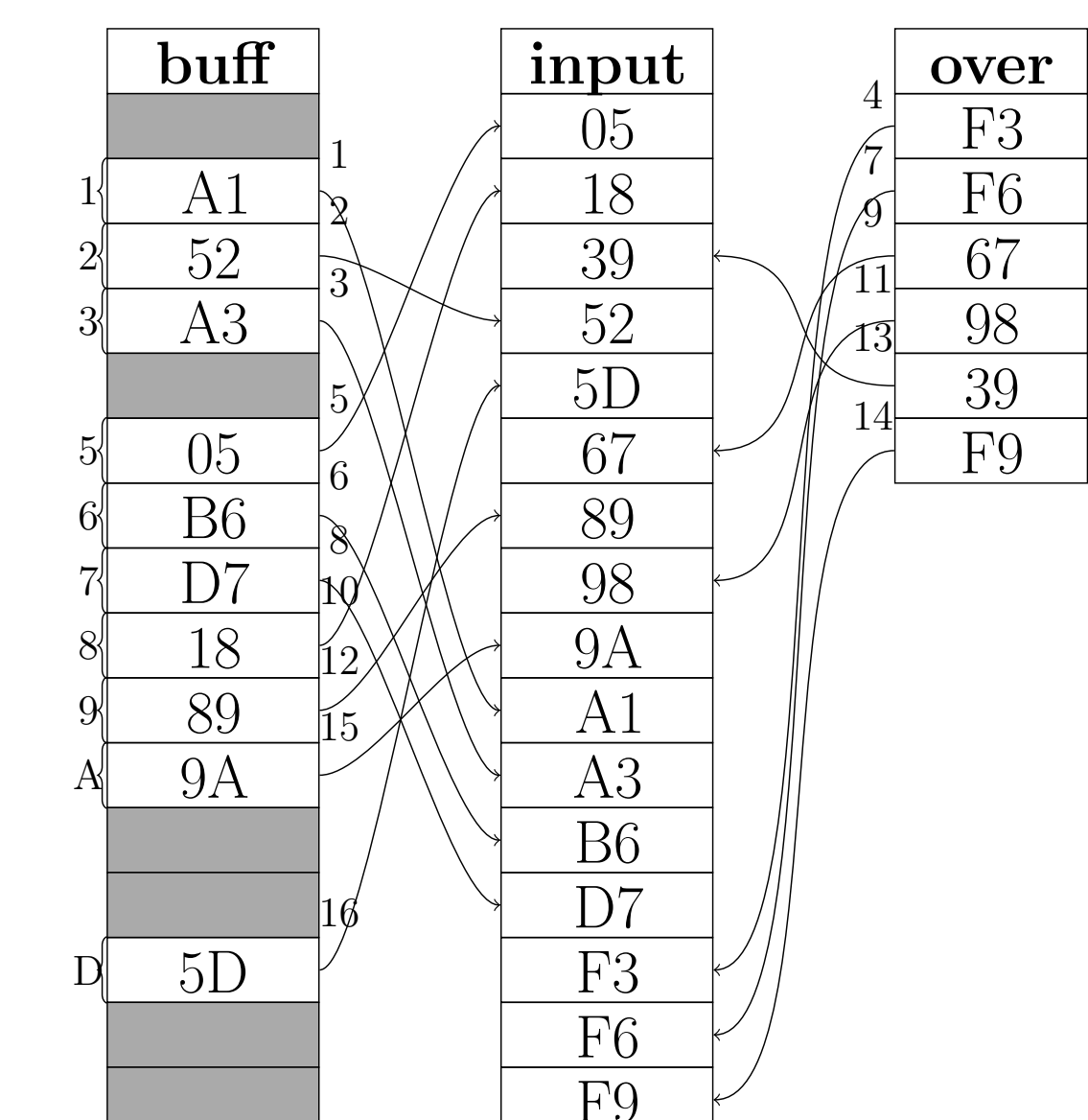
- Estimate bucket size
  - Skips initial counting pass, Manages overflow

- Estimate bucket sizes
- Deal into buckets based on LSD
- Deal from buckets and overflow based on next digit
- Repeat until all significant digits dealt, everything is sorted

## Deal and Count



## Deal



## Results and References

### Pre-processing Results

Graph Name	Vertices	Edges	GraphChi: Pre-processing	Fast Radix: Pre-processing	Δ Time
google [7]	900K	5M	1.7s	1.5s	14%
live-journal [1]	4.8M	69M	32s	32s	0%
hollywood [4, 2]	1.1M	113M	45s	38s	18%
twitter-2010 [5]	42M	1.5B	759s	693s	9.4%
friendster [9]	66M	1.8B	1022s	921s	11%
uk-union [3]	134M	5.5B	2549	2135	19%

### Sorting Results

Graph Name	Vertices	Edges	GraphChi: Sort	Fast Radix: Sort	Δ Sort	overflow
google [7]	900K	5M	0.15s	0.077s	99%	0.65%
live-journal [1]	4.8M	69M	5.4s	5.4s	0%	0.72%
hollywood [4, 2]	1.1M	113M	7.2s	4.9s	47%	0.50%
twitter-2010 [5]	42M	1.5B	128s	106s	21%	1.54%
friendster [9]	66M	1.8B	165s	133s	24%	0.12%
uk-union [3]	134M	5.5B	469s	329s	43%	0.49%

### References

- [1] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan. Group formation in large social networks: membership, growth, and evolution. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge discovery and data mining*, pages 44–54. ACM, 2006.
- [2] P. Boldi, M. Rosa, M. Santini, and S. Vigna. Layered Label Propagation: A Multi-Resolution Coordinate-Free Ordering for Compressing Social Networks. In S. Srinivasan, K. Ramamritham, A. Kumar, M. P. Ravindra, E. Bertino, and R. Kumar, editors, *Proceedings of the 20th International Conference on World Wide Web*, pages 587–596. ACM Press, 2011.
- [3] P. Boldi, M. Santini, and S. Vigna. A Large Time-Aware Graph. *SIGIR Forum*, 42(2):33–38, 2008.
- [4] P. Boldi and S. Vigna. The WebGraph framework I: Compression techniques. In *Proc. of the Thirteenth International World Wide Web Conference (WWW 2004)*, pages 595–601, Manhattan, USA, 2004. ACM Press.
- [5] H. Kwak, C. Lee, H. Park, and S. Moon. What is Twitter, a social network or a news media? In *WWW '10: Proceedings of the 19th International Conference on World Wide Web*, pages 591–600, New York, NY, USA, 2010. ACM.
- [6] A. Kyrola, G. E. Blelloch, and C. Guetstrin. Graphchi: Large-scale graph computation on just a pc. *USENIX*, 2012.
- [7] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney. Community Structure in Large Networks: Natural Cluster Sizes and the Absence of Large Well-Defined Clusters. *CoRR*, abs/0810.1355, 2008.
- [8] S. Thiel, G. Butler, and L. Thiel. Improving graphchi for large graph processing: Fast radix sort in pre-processing. In *Proceedings of the 20th International Database Engineering & Applications Symposium, IDEAS '16*, pages 135–141, New York, NY, USA, 2016. ACM.
- [9] J. Yang and J. Leskovec. Defining and Evaluating Network Communities based on Ground-truth. *CoRR*, abs/1205.0233, 2012.