ForeGraph: Exploring Large-scale Graph Processing on Multi-FPGA Architecture

Guohao Dai[1], Tianhao Huang[1], Yuze Chi[2], Ningyi Xu[3], Yu Wang[1], Huazhong Yang[1]

[1]Tsinghua University, [2]UCLA, [3]MSRA

dgh14@mails.tsinghua.edu.cn

2/25/17

Content

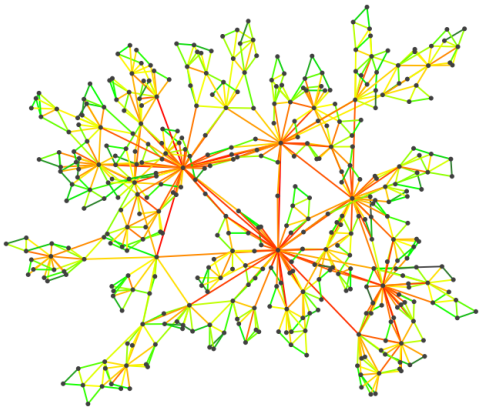
- **Background**
- **Motivation**
- **Related Work**
- **Architecture and Detailed Implementation**
- **Experiment Results**
- **Conclusion and Future Work**

Content

- **Background**
- Motivation
- Related Work
- Architecture and Detailed Implementation
- Experiment Results
- Conclusion and Future Work

Large-scale graphs are widely used!

- Large-scale graphs are widely used in different domains
- Involved with **billions** of edges and **Gbytes ~ Tbytes** storage
  - WeChat: 0.65 billions active users (2015)
  - Facebook: 1.55 billions active users (2015Q3)
  - Twitter2010: 1.5 billions edges, 13GB
  - Yahoo-web: 6.6 billions edges, 51GB
- Different graph algorithms
  - Generality requirement



Social network analysis

User behavior analysis

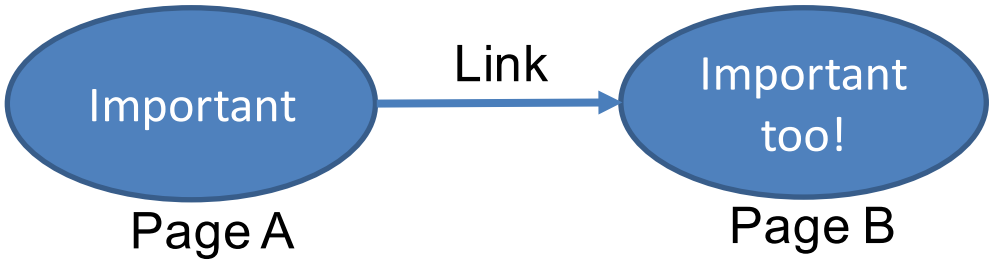
Bio-sequence analysis

User preference recommendation

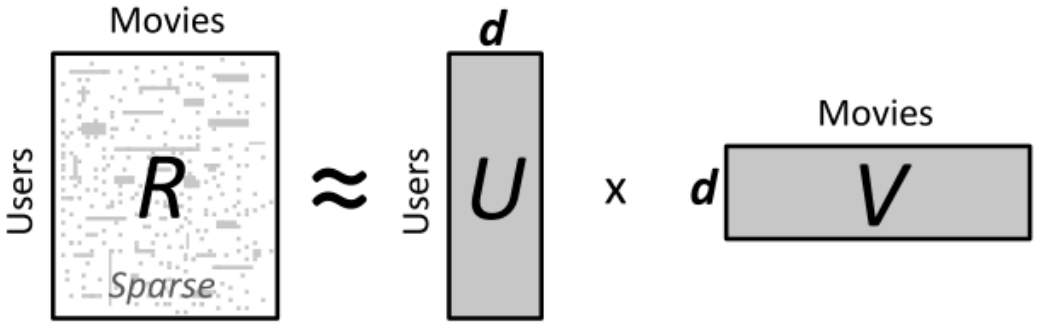
G. Dror, N. Koenigstein, Y. Koren, and M. Weimer. The yahoo! music dataset and kdd-cup'11
H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media?

Different graph algorithms

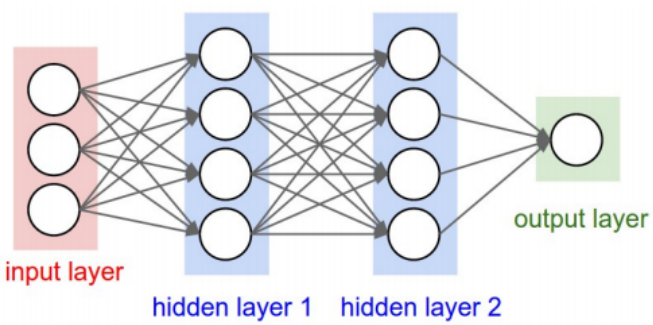
- PageRank
  - The rank of a page depends on ranks of pages which **link** to it

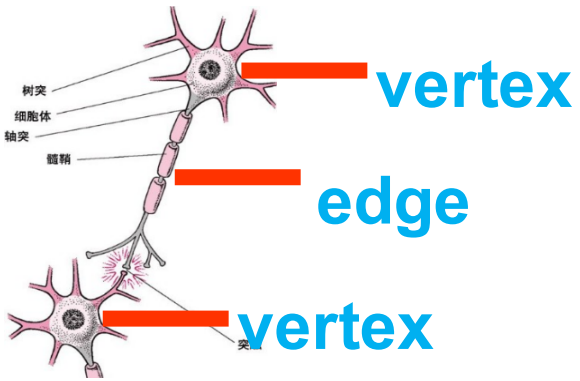


- User Recommendation
  - **Matrix → Graph**



- Deep Learning
  - **Network → Graph**

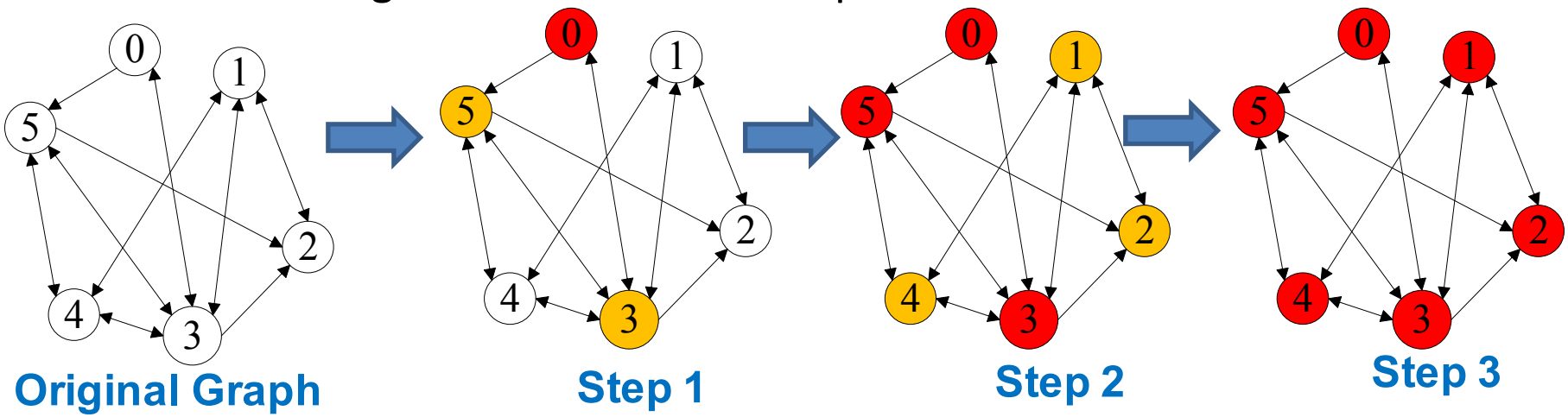




Page, Lawrence, et al. *The PageRank citation ranking: Bringing order to the web*. Stanford InfoLab, 1999.
Low, Yucheng, et al. "Distributed GraphLab: a framework for machine learning and data mining in the cloud." *Proceedings of the VLDB Endowment* 5.8 (2012): 716-727
Qiu, Jiantao, et al. "Going deeper with embedded fpga platform for convolutional neural network." *Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*. ACM, 2016.

Generality requirement

- High-level abstraction model
  - Read-based/Queue-based Model for BFS/APSP [Stanford, PACT'10] ×
  - Vertex-Centric Model (VCM) [Google, SIGMOD'10] √
- In VCM
  - A vertex updated → Neighbor vertices to be updated
  - Different graph algorithms → Different updating functions
  - Traverse edges in VCM for each step



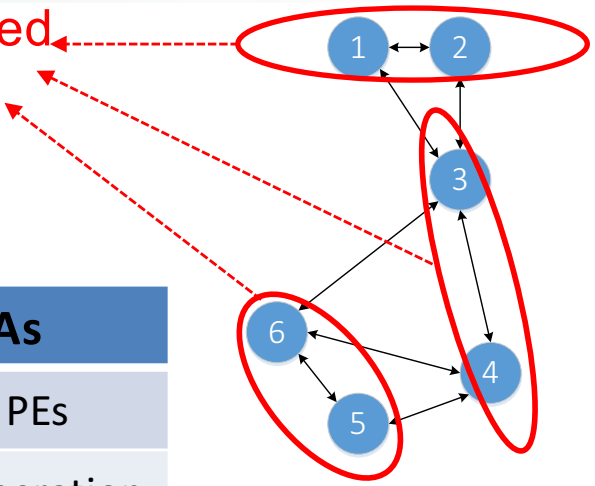
Malewicz, Grzegorz, et al. "Pregel: a system for large-scale graph processing." *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. ACM, 2010.
Hong, Sungpack, Tayo Oguntebi, and Kunle Olukotun. "Efficient parallel graph exploration on multi-core CPU and GPU." *Parallel Architectures and Compilation Techniques (PACT), 2011 International Conference on*. IEEE, 2011.

Content

- Background
- **Motivation**
- Related Work
- Architecture and Detailed Implementation
- Experiment Results
- Conclusion and Future Work

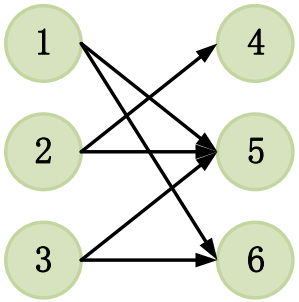
Why FPGA?

- High potential parallelism
- Relatively simple operations
  - e.g. Breadth-First Search: comparison



	CPUs	GPUs	FPGAs
Parallelism	10~100 threads	>1000 threads	>1000 PEs
Architecture	Complex	Simple	Bit-level operation
Suitable for graphs?	☹	☺	☺

- Bandwidth is essential
  - Suffer from random access
  - Suitable memory
    - Disk, DRAM, cache? ×
    - SRAM? √



Src: 1,2,3	Dst: 4,5,6
Dst: 5,6, 4,5,5,6	Src: 2,1, 2,3,1,3

FPGA： Xilinx xvcu190	GPU： NVIDIA Tesla P100
Block RAM	Shared Memory
16.61MB ☺	2.7MB ☹

Why Multi-FPGA?

- Using more FPGAs means…
  - Larger on-chip storage
  - Higher degree of parallelism
  - Higher bandwidth of data access
- Scalability
  - Size of BRAMs on a chip ~ MB
  - Size of large-scale graphs ~ GB to TB

10^3 ~ 10^6 gap!

  - Using **multi-FPGA** based on **scalable interconnection** schemes can be a solution to large-scale graph processing problems **in future**
    - Full connection? ×
    - Mesh/Torus √

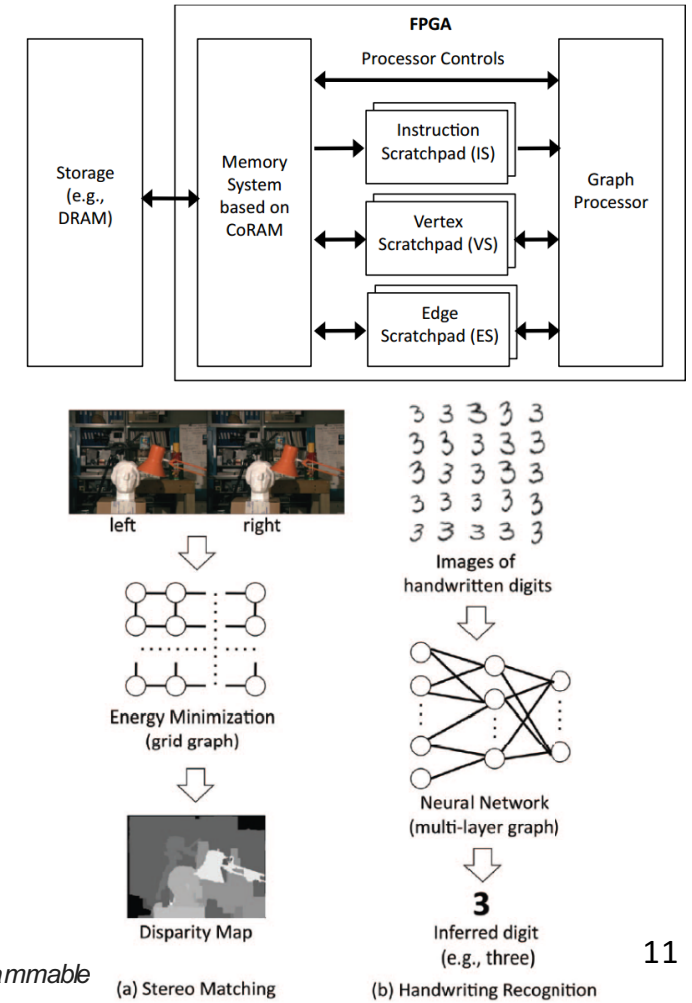
Content

- Background
- Motivation
- **Related Work**
- Architecture and Detailed Implementation
- Experiment Results
- Conclusion and Future Work

GraphGen [CMU, FCCM'14]

- First vertex-centric system on FPGA
  - Storing graphs on off-chip DRAMs using CoRAMs
  - ML support
- **However…**
  - **Do not support large-scale graphs**

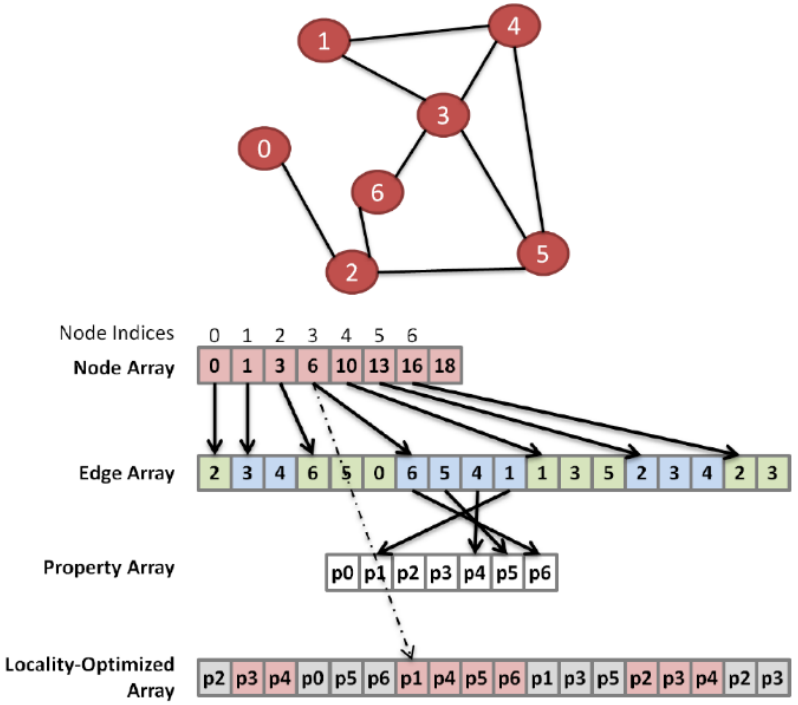
Applications	Stereo Matching	Handwriting Recognition
Algorithm	Tree-Reweighted Message Passing (TRW-S)	Convolutional Neural Network (CNN)
CPU software base	Middlebury [11]	CodeProject [9]
Dataset	Tsukuba (384x288 images)	MNIST database (29x29 images)
Graph size	110,592 vertices, 221,184 edges	5,589 vertices, 341,224 edges
Graph shape	Grid, regular	Multi-layer, irregular
Graph traversal	Diagonal	First to last layer
Graph partitioning	Manual (partition to tiles)	Automatic
Custom instruction implementation	Adapted an existing hardware IP from [2]	Manually made
Subgraph size	Tile of 12x64 vertices	Up to 2K vertices and 16K edges
Vertex/edge size	16 x 32-bit	32-bit

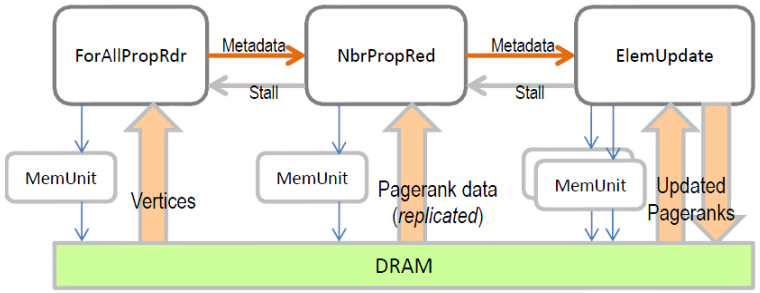


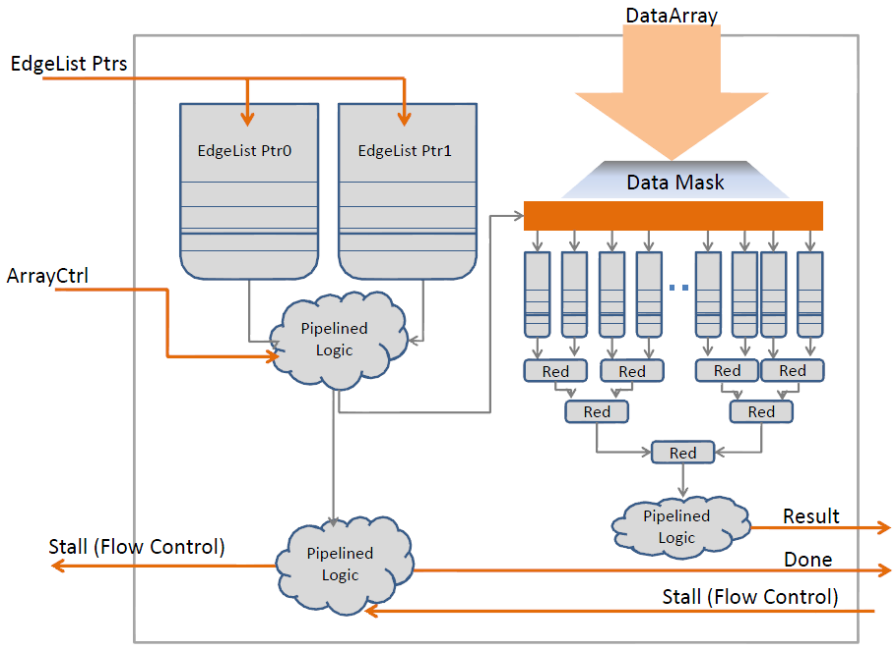
Nurvitadhi, Eriko, et al. "GraphGen: An FPGA framework for vertex-centric graph computation." *Field-Programmable Custom Computing Machines (FCCM), 2014 IEEE 22nd Annual International Symposium on*. IEEE, 2014.

GraphOps [Stanford, FPGA'16]

- Graph processing library on FPGA
  - APIs for different operations in graphs
- **However…**
  - **Preprocessing overhead**
  - **Scalability to multi-FPGAs**



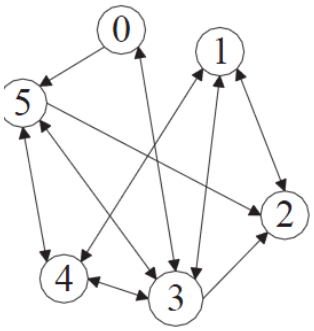




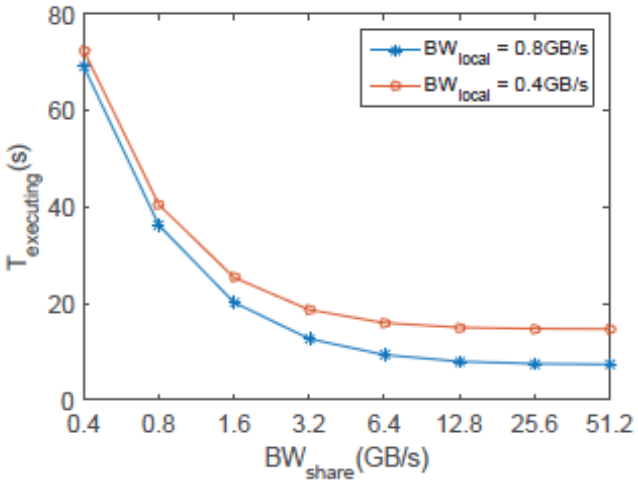
Oguntebi, Tayo, and Kunle Olukotun. "Graphops: A dataflow library for graph analytics acceleration." *Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*. ACM, 2016.

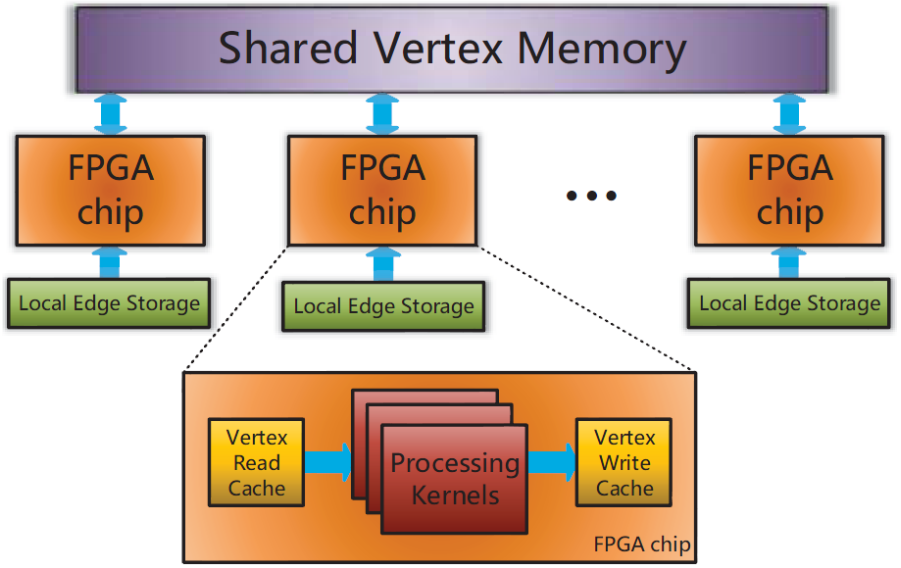
FPGP [ours, FPGA'16]

- Multi-FPGA support
- One FPGA chip – One graph partition
  - Independent edge storage
  - Optimized data allocation
- **However**
  - **All FPGAs linked to one SVM**
  - **Lack of scalability**



I_1	I_2	I_3
0, 1	2, 3	4, 5
S_1	S_2	S_3
$SS_{1,1}$	$SS_{1,2}$	$SS_{1,3}$
	1→2 0,1→3	1→4 0→5
$SS_{2,1}$	$SS_{2,2}$	$SS_{2,3}$
3→0 2,3→1	3→2	3→4 3→5
$SS_{3,1}$	$SS_{3,2}$	$SS_{3,3}$
4→1	5→2 4,5→3	5→4 4→5

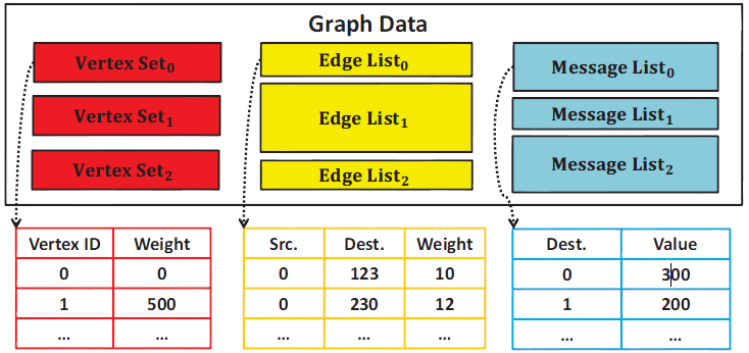




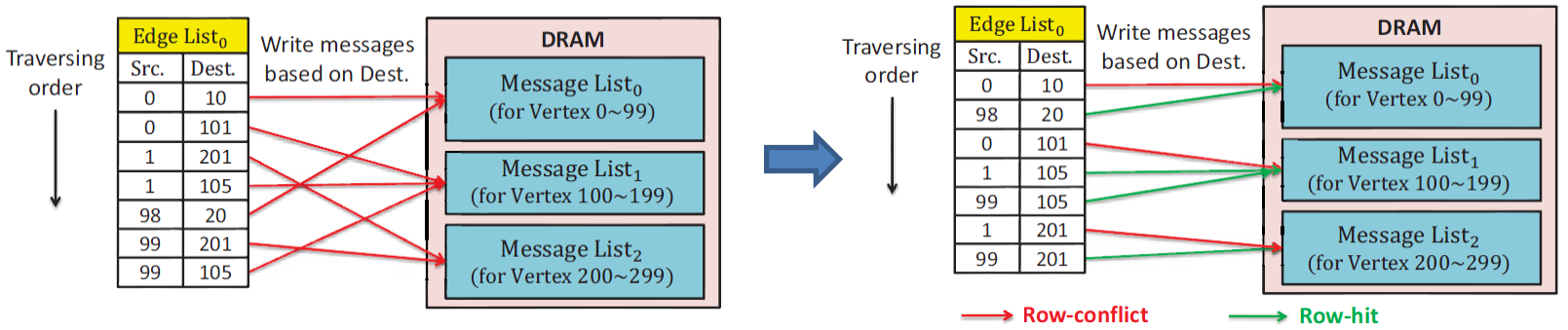
Dai, Guohao, et al. "FPGP: Graph Processing Framework on FPGA A Case Study of Breadth-First Search." *Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*. ACM, 2016.

Zhou's work [USC, FCCM'16]

- Using edges to store value of vertices
  - One edge – One message (src to dst)
  - Edges stored in DRAMs



- Improve off-chip DRAM hit ratio

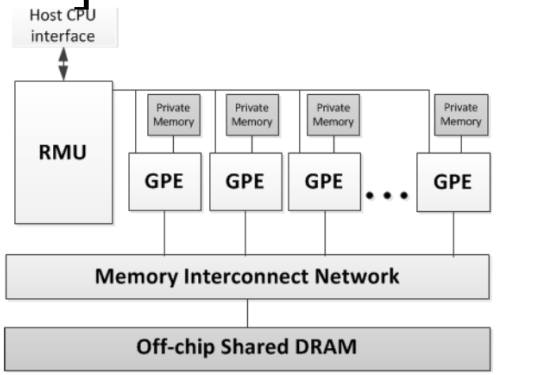


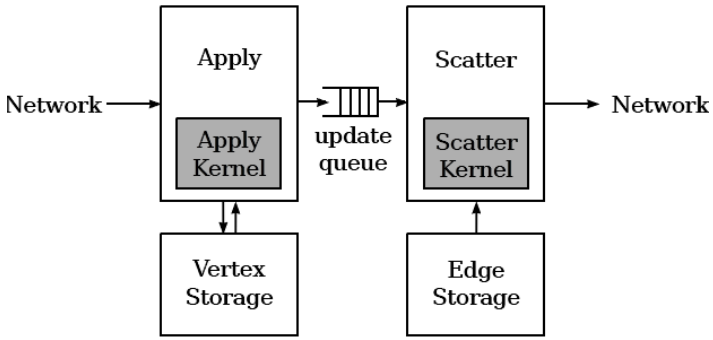
- **However…**
  - **The largest graph in its experiment: ~65M edges**
  - **Cannot scale to multi-FPGAs**

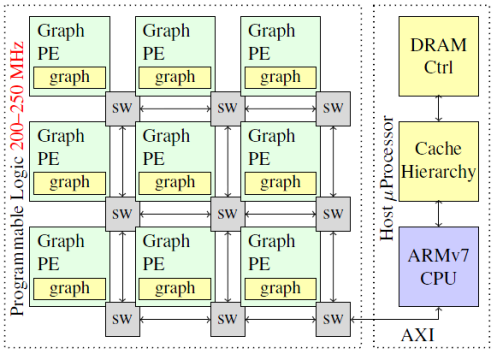
Zhou, Shijie, Charalampos Chelmis, and Viktor K. Prasanna. "High-throughput and Energy-efficient Graph Processing on FPGA." *Field-Programmable Custom Computing Machines (FCCM), 2016 IEEE 24th Annual International Symposium on*. IEEE, 2016.

Other systems

- Brahim's work [ICT, FPT'11, FPL'12, ASAP'12]
  - Using multi-FPGA system
  - **Designed for dedicated algorithms**
    - BFS/ASAP
    - Graphlet counting
- GraVF [HKU, FPL'16]
  - Scatter value from src to dst
  - **Lack of optimization for data access**
- GraphSoC [NTU, ASAP'15]
  - Using soft cores on FPGAs
  - **Lack of optimization for data access**







Betkaoui, Brahim, et al. "A framework for FPGA acceleration of large graph problems: Graphlet counting case study." *Field-Programmable Technology (FPT), 2011 International Conference on*. IEEE, 2011.

Betkaoui, Brahim, et al. "A reconfigurable computing approach for efficient and scalable parallel graph exploration." *Application-Specific Systems, Architectures and Processors (ASAP), 2012 IEEE 23rd International Conference on*. IEEE, 2012.

Betkaoui, Brahim, et al. "Parallel FPGA-based all pairs shortest paths for sparse networks: A human brain connectome case study." *Field Programmable Logic and Applications (FPL), 2012 22nd International Conference on*. IEEE, 2012.

Engelhardt, Nina, and Hayden Kwok-Hay So. "GraVF: A vertex-centric distributed graph processing framework on FPGAs." *Field Programmable Logic and Applications (FPL), 2016 26th International Conference on*. IEEE, 2016.

Kapre, Nachiket. "Custom FPGA-based soft-processors for sparse graph acceleration." *Application-specific Systems, Architectures and Processors (ASAP), 2015 IEEE 26th International Conference on*. IEEE, 2015.

Related work - Conclusion

	Year & Conference	Support different algorithms	Size of graphs（#edges）	Scalability to Multi-FPGAs
GraphGen	FCCM'14	Support	221 k	
GraphOps	FPGA'16	Support	30 m	
FPGP	FPGA'16	Support	1.4 b	
Zhou's work	FCCM'16	Support	65.8 m	
Brahim's work	11~12	Not support	80 m	
GraVF	FPL'16	Support	512 k	
GraphSoc	ASAP'15	Support	12 k	

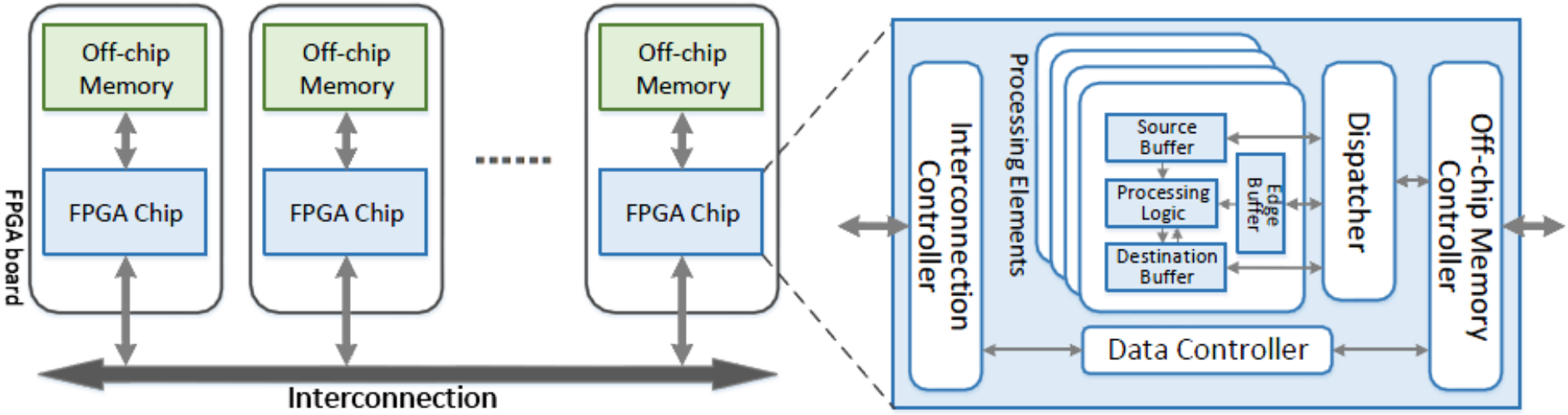
- A general purposed large-scale graph processing system using multi-FPGAs is required
  - **Generality**: Support different algorithms
  - **Velocity**: Process large-scale graphs (>1 billion edges) fast
  - **Scalability**: Multi-FPGAs with scalable connections

Content

- Background
- Motivation
- Related Work
- **Architecture and Detailed Implementation**
- Experiment Results
- Conclusion and Future Work

Overall Architecture

- Overall architecture

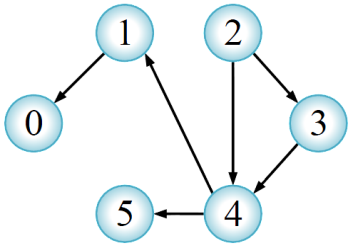


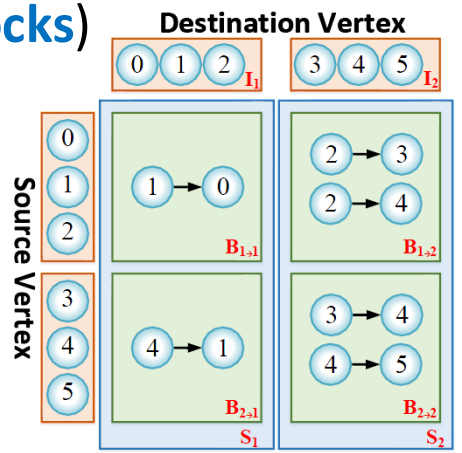
- Multi processing units: Multi-FPGA + Multi-PE
  - One FPGA board = one FPGA chip + exclusive DRAM
  - One FPGA chip include several PEs to perform graph updating
- **We need to avoid conflict among units**
  - **Well-designed data allocation is required**

Data Allocation

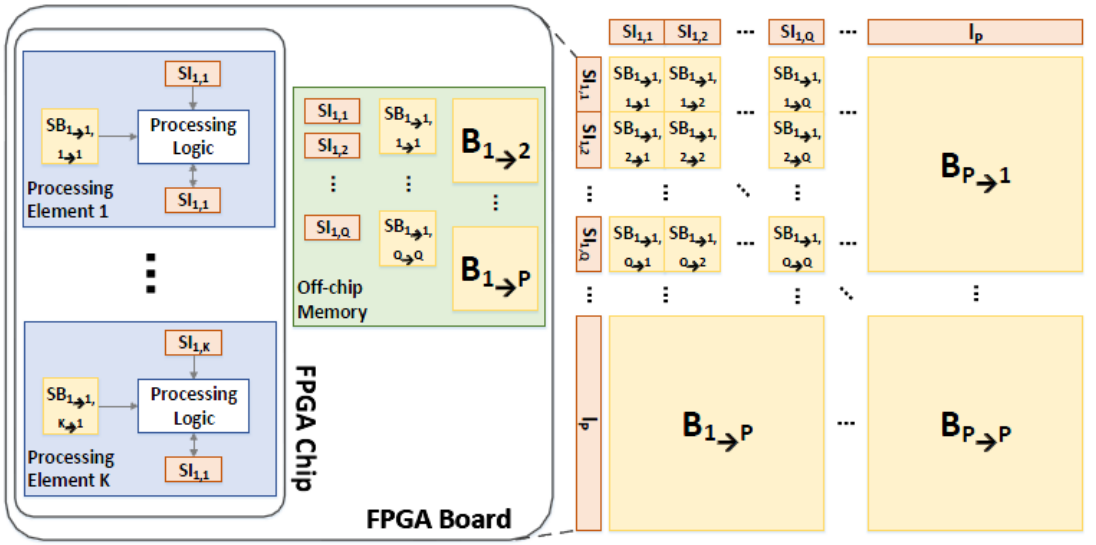
- Avoid data conflict among boards
  - Interval-block Model (**traverse edges → process all blocks**)
  - Vertices divided in to *P* **intervals**
  - Edges divided into P^2 **blocks**
  - **One FPGA board updates**
    - ***1* interval**
    - ***P* blocks**

No Conflict





- Only intervals are transferred among boards
- Further partitioning
  - *Q* **sub-intervals**
  - Q^2 **sub-blocks**
  - One PE on a chip
    - One src sub-interval
    - One dst sub-interval
    - One sub-block



Processing Flow

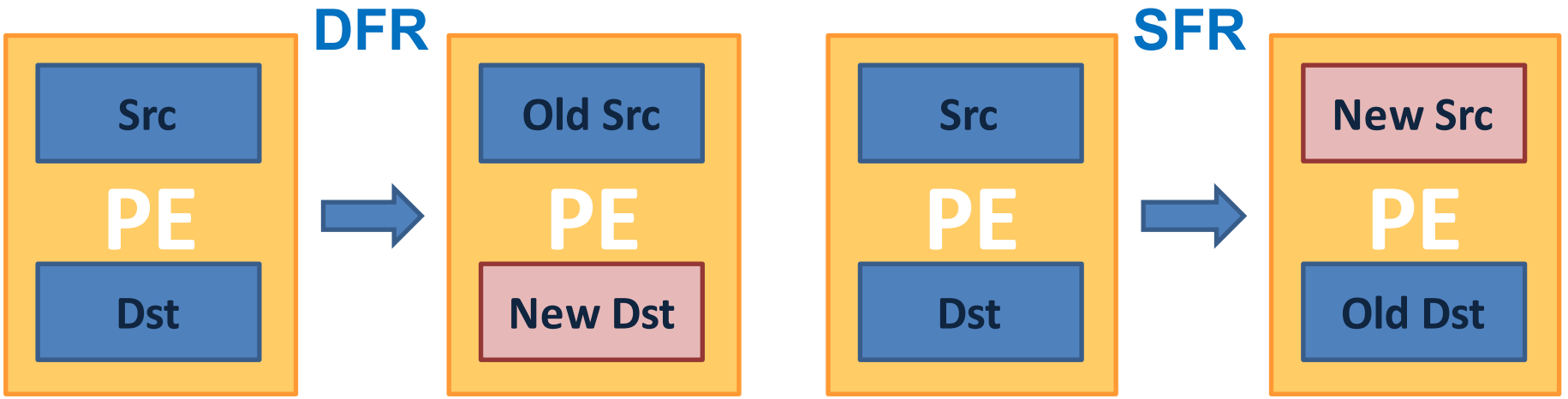
- *K* PEs on a chip
  - Processing ***K*** sub-blocks (One PE processes one sub-block)
  - ***P * Q²*** sub-blocks need to be processed

$$T = \underbrace{\alpha T_{loading\ a\ sub-interval}}_{\text{loading vertices}} + \underbrace{T_{loading\ all\ sub-blocks}}_{\text{loading edges}} + \underbrace{\frac{PQ^2 T_{procesing\ a\ sub-block}}{\beta}}_{\text{processing}}$$

- **Key points to accelerate processing**
  - Minimize α (Times of loading sub-intervals)
    - **Minimize substitutions of sub-intervals**
  - Maximize β (Number of PEs processing simultaneously)
    - **Avoid idle PEs during processing**
    - **Balance workloads of different PEs**

Opt. I: Minimized Substitutions

- When processing another sub-block
  - Substitute at least one sub-interval
  - **Less substitutions → less data transferred**
- Two different strategies

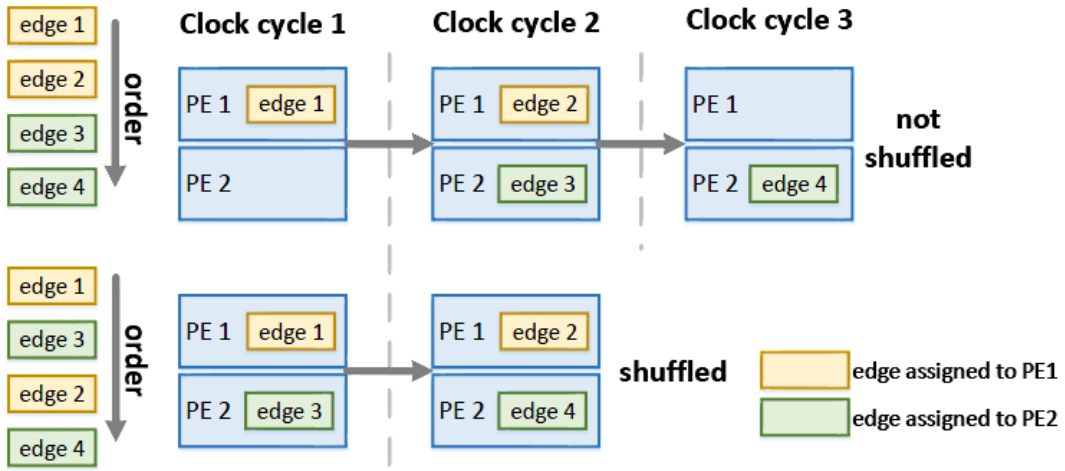


- **Minimize data transferred using DFR**

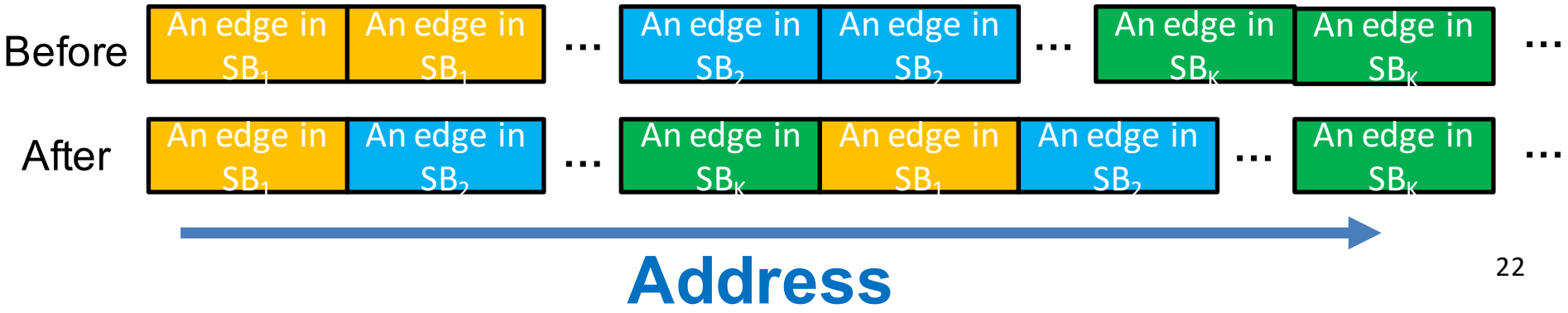
	#sub-intervals (read)	#sub-intervals (write)
DFR	$Q+Q^2/K$	Q^2/K
SFR	$Q+Q^2$	Q

Opt. II: Avoid Idle PEs

- Rearrange edges can avoid idle PEs
  - Assuming 2 edges can be loaded from the DRAM per cycle



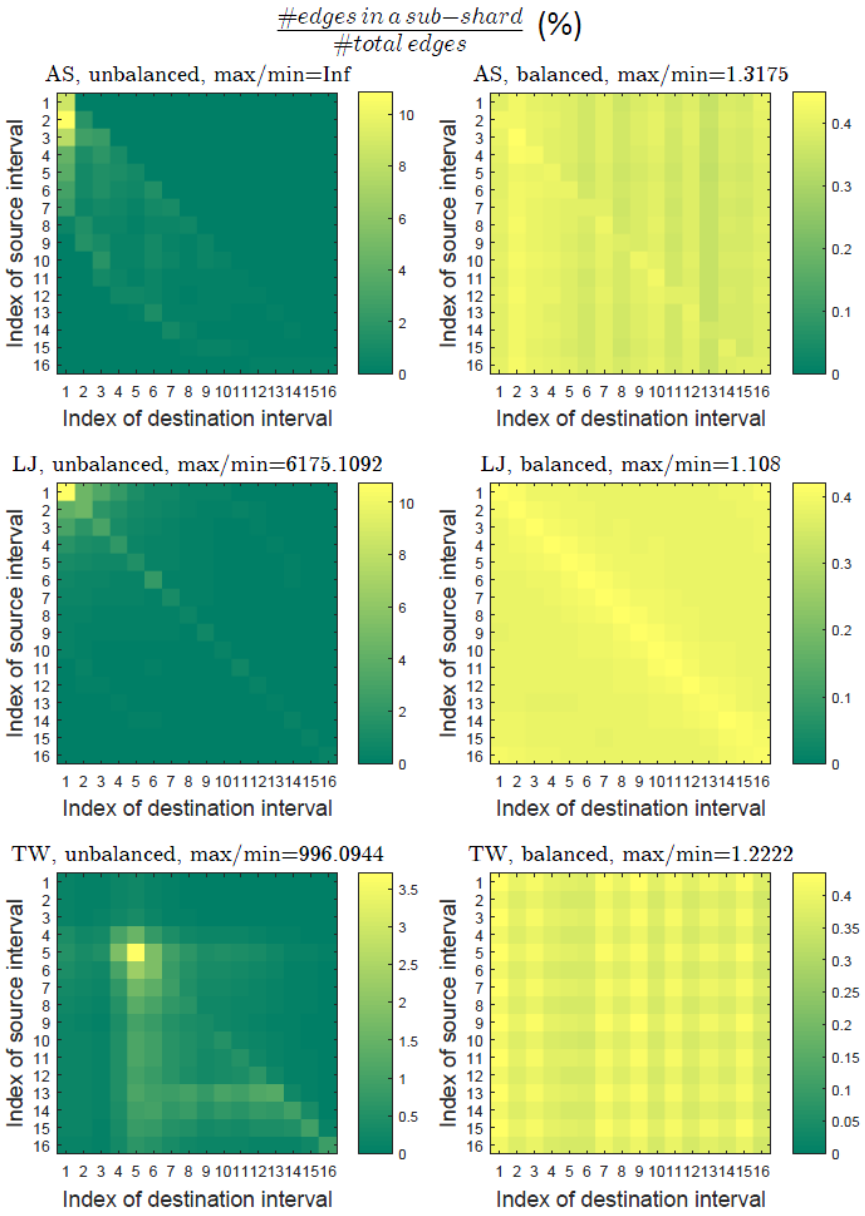
- K PEs on a chip
  - Edges in K consecutive sub-blocks are rearranged
  - **Avoid idle PEs using sub-block rearrangement**



Opt. III: Balanced Workloads

- K PEs need to be synchronized
  - Total execution time depends on the slowest PE
  - **Execution time of a PE $\propto$ #edges**
- Need to balance #edges in different sub-blocks
  - **Balance workloads of different PEs using hash function**
- Hash function

	Interval 1	Interval 2	Interval 3
division	v1, v2, v3	v4, v5, v6	v7, v8, v9
hash	v1, v4, v7	v2, v5, v8	v3, v6, v9



Content

- Background
- Motivation
- Related Work
- Architecture and Detailed Implementation
- **Experiment Results**
- Conclusion and Future Work

Experimental Setup

- Platform
  - Xilinx Virtex UltraScale VCU110 evaluation platform
  - Xilinx Vivado 2016.2
  - Post-place-and-route simulations
  - DRAM peak bandwidth: 19.2GB/s
- Datasets

	|V|	|E|
com-youtube (YT)	1.16 million	2.99 million
wiki-talk (WK)	2.39 million	5.02 million
live-journal (LJ)	4.85 million	69.0 million
twitter-2010 (TW)	41.7 million	1.47 billion
yahoo-web (YH)	1.41 billion	6.64 billion

Stanford large network dataset collection. http://snap.stanford.edu/data/index.html#web.
Yahoo! altavisata web page hyperlink connectivity graph, circa 2002. http://webscope.sandbox.yahoo.com/.
Kwak, Haewoon, et al. "What is Twitter, a social network or a news media?." *Proceedings of the 19th international conference on World wide web*. ACM, 2010.

Resource Utilization

- On-chip BRAM resources are key to large-scale graph processing on FPGAs!
  - **> 80% BRAM resources are used**

	BFS	PR	WCC
# PE per chip	96	24	24
LUTs	31.2%	33.4%	35.9%
Registers	17.3%	20.6%	19.7%
BRAMs	**89.4%**	**81.0%**	**81.0%**
Maximal clock frequency	205MHz	187MHz	173MHz
Simulation clock frequency	200MHz	150MHz	150MHz

Performance

Algorithm	Graph	Execution Time(s)	Throughput(MTEPS)
BFS	YT	0.010	897
	WK	0.027	929
	LJ	0.452	1069
	TW (4 chips)	15.12	1458 (364/chip)
PR	YT	0.030	997
	WK	0.052	965
	LJ	0.578	1193
	TW (4 chips)	7.921	1856 (464/chip)
WCC	YT	0.016	934
	WK	0.021	956
	LJ	0.307	1124
	TW (4 chips)	24.68	1727 (432/board)

Throughput: ~ 1000 Millions Traversed Edges Per Second

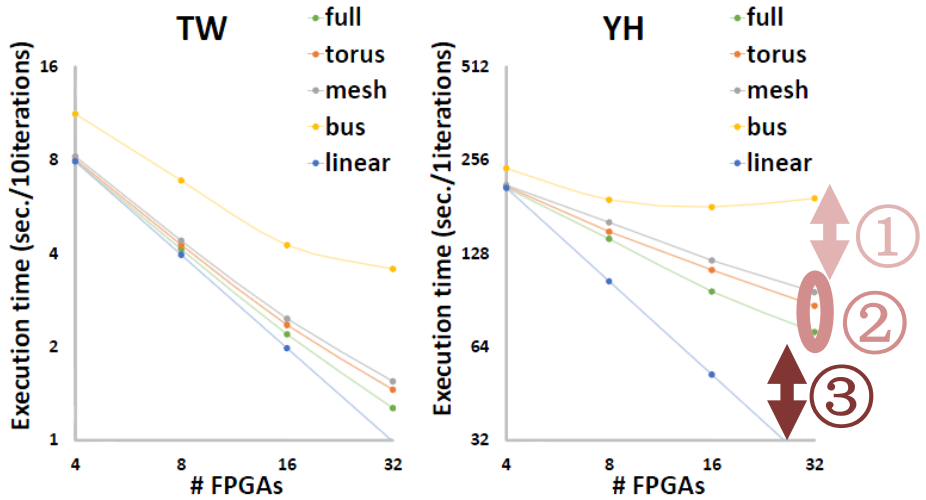
Performance

- Compared with state-of-the-art systems
  - **4.54x ~ 8.07x** speedup
  - **1.41x ~ 2.65x** throughput improvement

Alg.	Graph	Metric	ForeGraph		Comparison system			Improvement
			# FPGAs	Performance	System	Platform	Performance	
BFS	TW	time (s)	4	15.12	TurboGraph [SIGKDD13]	CPU	76.134	5.04x
BFS	TW	time (s)	4	15.12	FPGP [FPGA16]	1 FPGA	121.99	8.07x
PR	TW	time (s)	4	7.921	PowerGraph [OSDI12]	512 CPUs	36	4.54x
BFS	WK	MTEPS	1	1069	Zhou's work [FCCM16]	1 FPGA	657	1.41x
BFS	-	MTEPS	4	1458	CyGraph [IPDPSW16]	4 FPGAs	550	2.65x

Scalability

- Different interconnection schemes
  - 12.25 Gb/s bandwidth and 400ns latency



  - ① All FPGAs being connected to one bus
    - One bus line leads to heavy traffic
  - ② Similar performance, torus/mesh (ForeGraph) and full
    - **ForeGraph scales well to larger graphs by using more FPGA chips**
  - ③ Full connection scheme cannot achieve linear speedup
    - Due to characteristics of natural graphs (e.g. α-law)

Content

- Background
- Motivation
- Related Work
- Architecture and Detailed Implementation
- Experiment Results
- **Conclusion and Future Work**

Conclusion & Future Work

- Conclusion
  - ForeGraph can
    - **Generality**: Support different algorithms
    - **Velocity**: Process graphs with billions of edges with throughput at 1000 MTEPS
    - **Scalability**: Scale to larger graphs by using more FPGAs
  - Larger BRAMs → better performance
- Future work
  - Support for more applications
  - Open source or compatibility of big data framework

Reference

1. Page, Lawrence, et al. *The PageRank citation ranking: Bringing order to the web*. Stanford InfoLab, 1999.
2. Low, Yucheng, et al. "Distributed GraphLab: a framework for machine learning and data mining in the cloud." *Proceedings of the VLDB Endowment* 5.8 (2012): 716-727.
3. Qiu, Jiantao, et al. "Going deeper with embedded fpga platform for convolutional neural network." *Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*. ACM, 2016.
4. Malewicz, Grzegorz, et al. "Pregel: a system for large-scale graph processing." *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. ACM, 2010.
5. Hong, Sungpack, Tayo Oguntebi, and Kunle Olukotun. "Efficient parallel graph exploration on multi-core CPU and GPU." *Parallel Architectures and Compilation Techniques (PACT), 2011 International Conference on*. IEEE, 2011.
6. Nurvitadhi, Eriko, et al. "GraphGen: An FPGA framework for vertex-centric graph computation." *Field-Programmable Custom Computing Machines (FCCM), 2014 IEEE 22nd Annual International Symposium on*. IEEE, 2014.
7. Oguntebi, Tayo, and Kunle Olukotun. "Graphops: A dataflow library for graph analytics acceleration." *Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*. ACM, 2016.
8. Dai, Guohao, et al. "FPGP: Graph Processing Framework on FPGA A Case Study of Breadth-First Search." *Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*. ACM, 2016.

Reference

9. Zhou, Shijie, Charalampos Chelmis, and Viktor K. Prasanna. "High-throughput and Energy-efficient Graph Processing on FPGA." *Field-Programmable Custom Computing Machines (FCCM), 2016 IEEE 24th Annual International Symposium on*. IEEE, 2016.
10. Betkaoui, Brahim, et al. "A framework for FPGA acceleration of large graph problems: Graphlet counting case study." Field-Programmable Technology (FPT), 2011 International Conference on. IEEE, 2011.
11. Betkaoui, Brahim, et al. "A reconfigurable computing approach for efficient and scalable parallel graph exploration." Application-Specific Systems, Architectures and Processors (ASAP), 2012 IEEE 23rd International Conference on. IEEE, 2012.
12. Betkaoui, Brahim, et al. "Parallel FPGA-based all pairs shortest paths for sparse networks: A human brain connectome case study." Field Programmable Logic and Applications (FPL), 2012 22nd International Conference on. IEEE, 2012.
13. Engelhardt, Nina, and Hayden Kwok-Hay So. "GraVF: A vertex-centric distributed graph processing framework on FPGAs." Field Programmable Logic and Applications (FPL), 2016 26th International Conference on. IEEE, 2016.
14. Kapre, Nachiket. "Custom FPGA-based soft-processors for sparse graph acceleration." Application-specific Systems, Architectures and Processors (ASAP), 2015 IEEE 26th International Conference on. IEEE, 2015.
15. Han, Wook-Shin, et al. "TurboGraph: a fast parallel graph engine handling billion-scale graphs in a single PC." *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2013.

Reference

16. Gonzalez, Joseph E., et al. "PowerGraph: Distributed Graph-Parallel Computation on Natural Graphs." *OSDI*. Vol. 12. No. 1. 2012.
17. Kyrola, Aapo, Guy E. Blelloch, and Carlos Guestrin. "GraphChi: Large-Scale Graph Computation on Just a PC." *OSDI*. Vol. 12. 2012.
18. Attia, Osama G., et al. "Cygraph: A reconfigurable architecture for parallel breadth-first search." *Parallel & Distributed Processing Symposium Workshops (IPDPSW), 2014 IEEE International*. IEEE, 2014.
19. Stanford large network dataset collection. http://snap.stanford.edu/data/index.html#web.
20. Yahoo! altavisata web page hyperlink connectivity graph, circa 2002. http://webscope.sandbox.yahoo.com/.
21. Kwak, Haewoon, et al. "What is Twitter, a social network or a news media?." *Proceedings of the 19th international conference on World wide web*. ACM, 2010.

Thank you!

Q & A