Part II: Vectorisation

QUEEN'S UNIVERSITY BELFAST

MOTIVATION

Graph processing - Irregular, data-dependent memory accesses

Vectorization a.k.a. Single Instruction Multiple Data (SIMD) parallelism

Push vs pull – open question which one to use
Require vectorized push and pull style traversal; no generic solution for push in literature

1. CleanCut: conflict-free and load-balanced partitions
2. VectorFast: compact graph representation to ease memory pressure
   Fast-forwarding for frontiers or convergence optimization
3. Graptor Domain-Specific Language (DSL) and Compiler
   Ease of use (e.g., auto-vectorization) and efficiency (compiler optimizations)

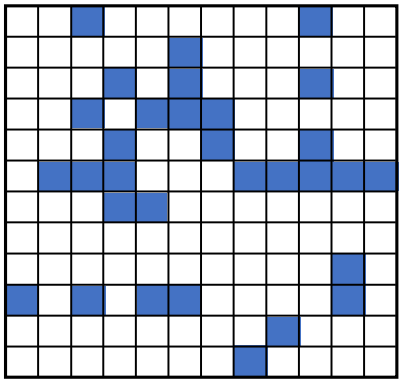
H. Vandierendonck, “Graptor: Efficient Pull and Push Style Vectorized Graph Processing”, ICS’20.

BACKGROUND: GRAPH DATA STRUCTURES (PULL)

Colors show distinct vector lanes
Assuming L=4 lanes

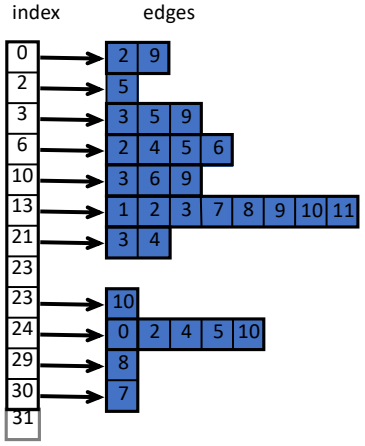
Adjacency matrix

12 vertices -> 12x12 matrix
31 edges -> 31 non-zero elements



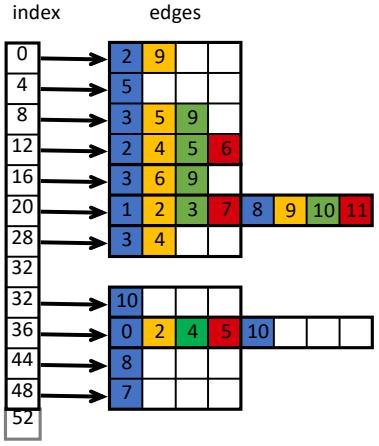
Compressed Sparse (scalar)

Ghost cell in index array shows number of edges



Padded Compressed Sparse

Padding ensures alignment
L-to-1 reductions required

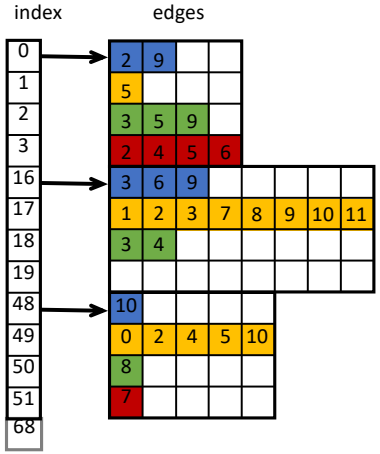


BACKGROUND: GRAPH DATA STRUCTURES (PULL)

Data-Parallel Vectorization

Lanes process distinct neighbor lists

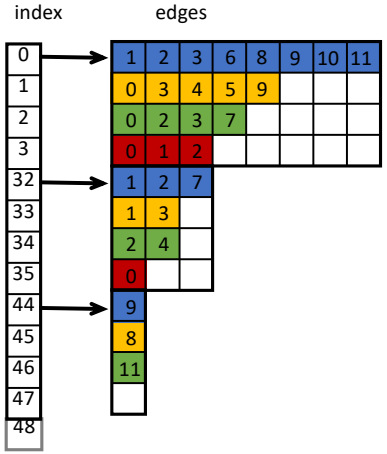
Many disabled lanes



SELL-C-sigma

Apply degree sorting [Kreutzer SIAM'14]

Load imbalance for power-law graphs [Besta IPDPS'17]

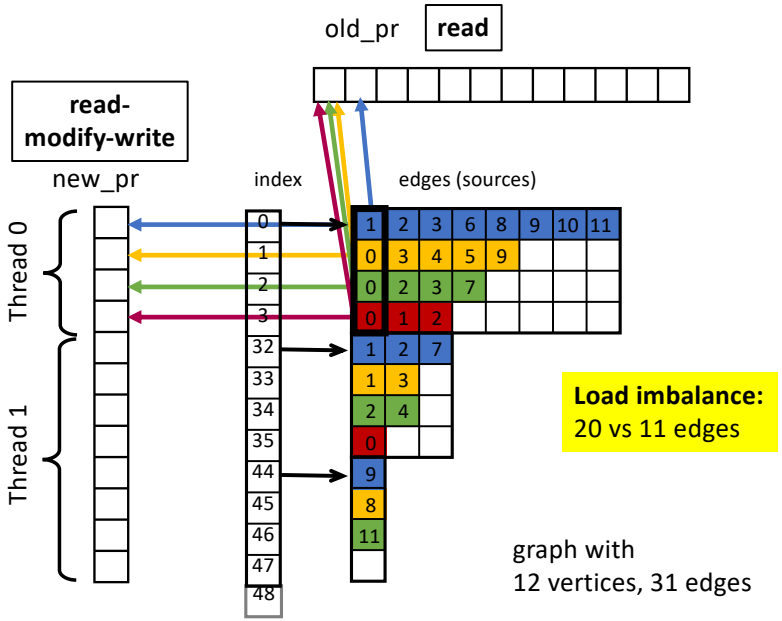


BACKGROUND: GRAPH DATA STRUCTURES (PULL)

SELL-C-Sigma [Kreutzer SIAM'14]

Load imbalance for power-law graphs [Besta IPDPS'17]

For each edge u → v:
new_pr[v] += old_pr[u]/deg[u];



QUEEN'S UNIVERSITY BELFAST

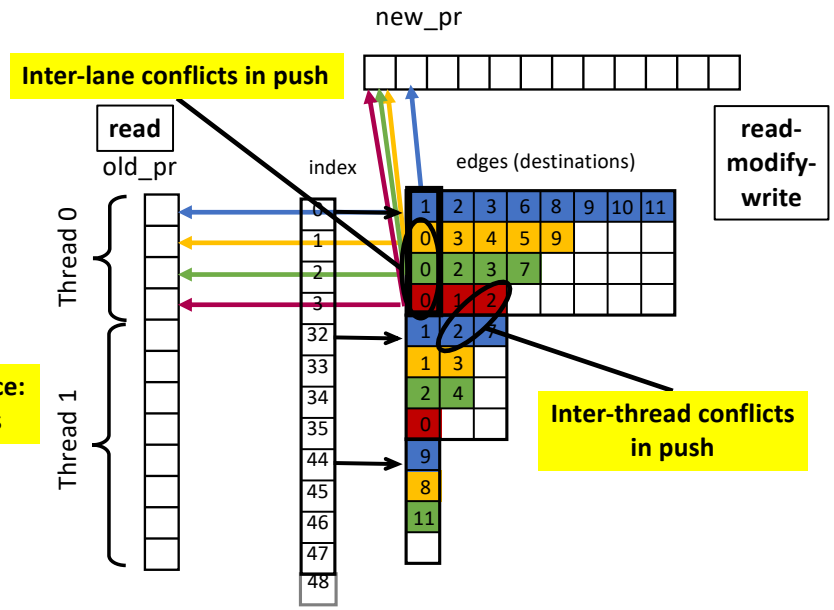
CHALLENGES OF PUSH STYLE VECTORIZATION

For each edge u → v:
new_pr[v] += old_pr[u]/deg[u];

Push style traversal interchanges read/write property arrays

- *Edges* array shows array indices being **upated**
- Sequential reads of old_pr
- Data-dependent, irregular access to new_pr

Load imbalance: 20 vs 11 edges

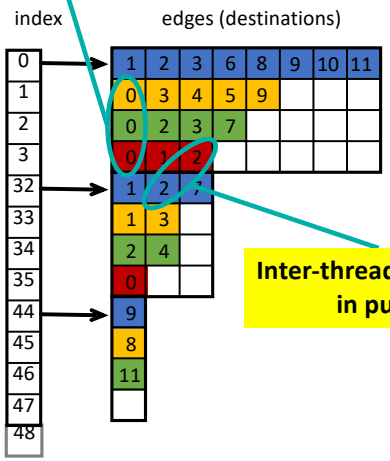


QUEEN'S UNIVERSITY BELFAST

CLEANCUT: RACE-FREE GRAPH PARTITIONS: PUSH

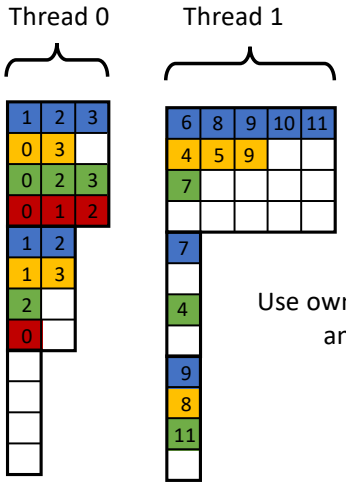
Approach:
Partition destinations
No inter-thread races
Load balanced (number of edges)

Inter-lane conflicts in push



Load imbalance:
20 vs 11 edges

Inter-thread conflicts in push



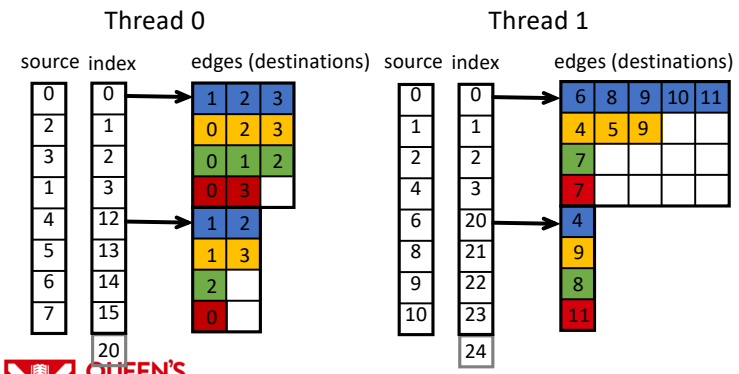
Use owner-computes and VEBO

No longer degree-sorted
Padding overhead

QUEEN'S UNIVERSITY BELFAST

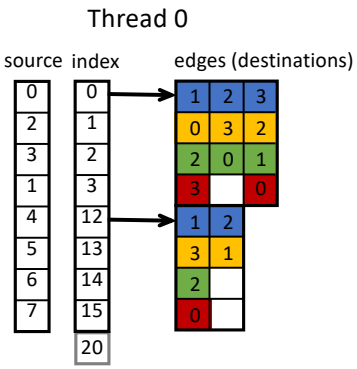
CLEANCUT: RACE-FREE GRAPH PARTITIONS: PUSH

Approach:
Compact and sort differently in each partition
Additional array to indicate sort order (source vertex IDs)



Load balanced and no inter-thread conflicts
Still has inter-lane conflicts

Approach:
Swap destinations or add padding to avoid inter-lane race conditions



May increase padding

EXPERIMENTAL EVALUATION

Hybrid processing with sparse (push)/dense (push or pull) phases based on frontier density

Vectorize dense phases only

Knights Landing (KNL): Xeon Phi 7210, 1.3 GHz

64 threads, 16-way SIMD with AVX512F

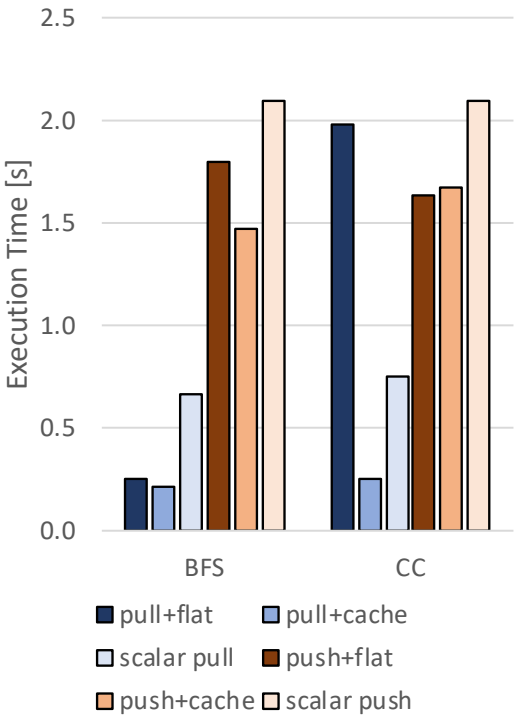
gcc 7.2.0

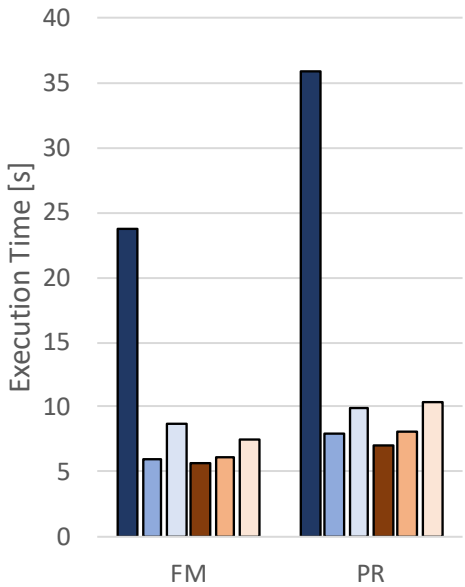
Intel Cilkplus parallel runtime

QUEEN'S UNIVERSITY BELFAST

Graph	Vertices	Edges
orkut	3.07M	234M
twitter	41.7M	2.41G
friendster	125M	5.17G
RMAT27	134M	2.21G
USAroad	23.9M	57.7M

	Algorithm
BFS	Breadth-first-search
CC	Connected components
LVL	BFS variant
FM	Radius estimation
MIS	Maximal Independent Set
PR	PageRank
APR	Accelerated PageRank





Twitter graph
Xeon Phi 7210, 1.3 GHz
64 threads, 16-way SIMD

Some algorithms prefer push, others prefer pull

Good vectorized configurations are faster than scalar

Caching generally helps in pull, not always in push

QUEEN'S UNIVERSITY BELFAST

Part III: Reduced Precision

QUEEN'S UNIVERSITY BELFAST

REDUCED PRECISION COMPUTATION

Motivation: reduce memory pressure related to ”random” memory accesses

Fewer bits/vertex, so more vertex data in cache

Half-precision floating point – also used for compact neural networks

Need software emulation!

Format	Origin	Size	Exponent	Mantissa	Application
FP16	IEEE	16	5	10	General
BFloat	Intel	16	8	7	Deep Learning
DLFloat	IBM	16	6	9	Deep Learning
CPMS	KIT	16	11	4	PageRank
6+10	**QUB**	**16**	**6**	**10**	**PageRank a.o.**

QUEEN'S UNIVERSITY BELFAST

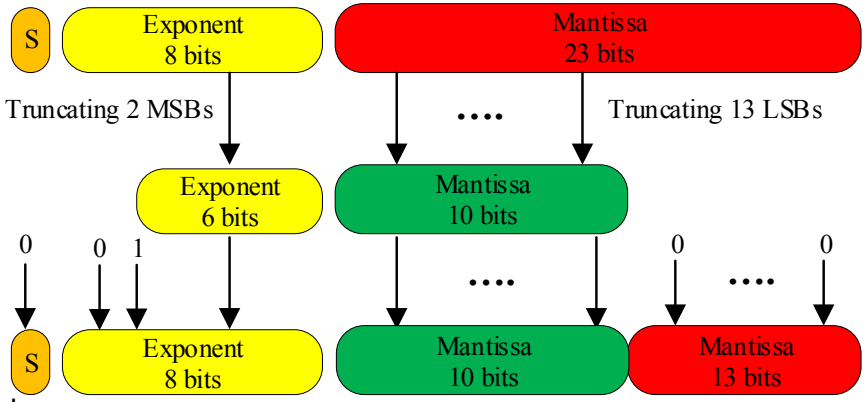
Amir Sabbagh Mollahosseini, Hans Vandierendonck, “Half-Precision Floating-Point Formats for PageRank: Opportunities and Challenges”, IEEE HPEC, 2020.

A CUSTOM FORMAT FOR PAGERANK

- All PageRank values are **non-zero** and **less than 1**
- Values are strictly positive – no sign bit needed
- The smallest PageRank value is (1-d)/|V|
  - d is damping factor, typically 0.85
- Exponents take between $-\log |V| - 1$ bits
- 6 exponent bits cover graphs with ~2^{64} vertices
- 5 exponent bits cover graphs with ~2^{32} vertices

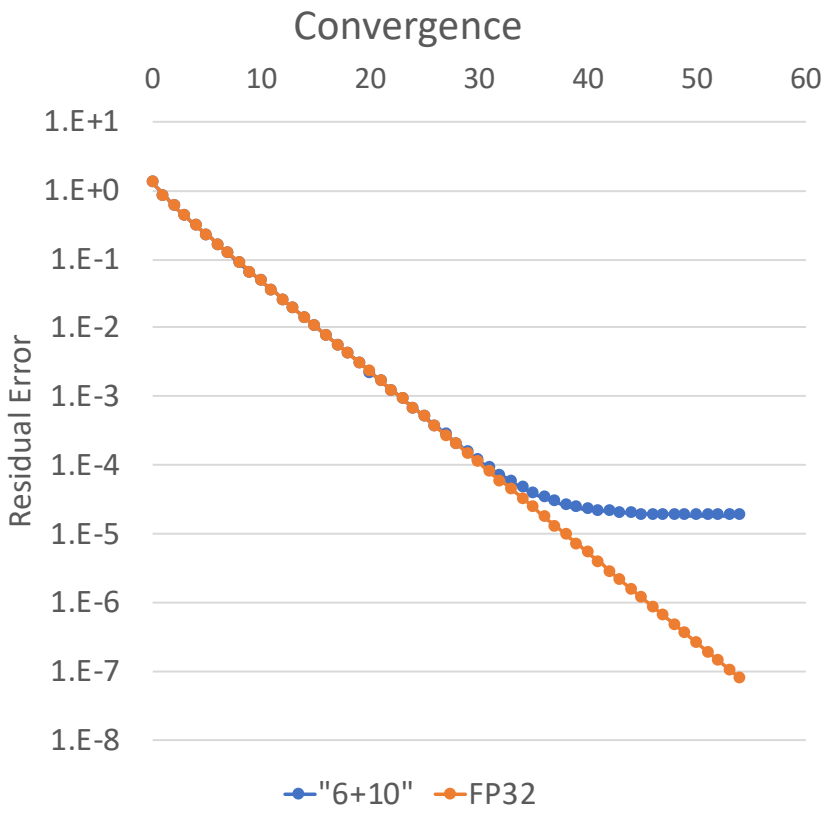
QUEEN'S UNIVERSITY BELFAST

CONVERSION



- Trick: store compact numbers in memory, convert to hardware-supported format when computing
- No range checking, rounding.
- No zero, NaN, Inf, denormalized numbers
- Reinterpret integer values as floating-point values
  - Note: scalar execution on x86-64 adds 3-cycle overhead for move between register files
- Two 1-cycle assembly operations to convert “6+10” format to IEEE-754 FP32 or FP64
  - Performance very sensitive to details of code!

QUEEN'S UNIVERSITY BELFAST



CONVERGENCE RATE PAGERANK

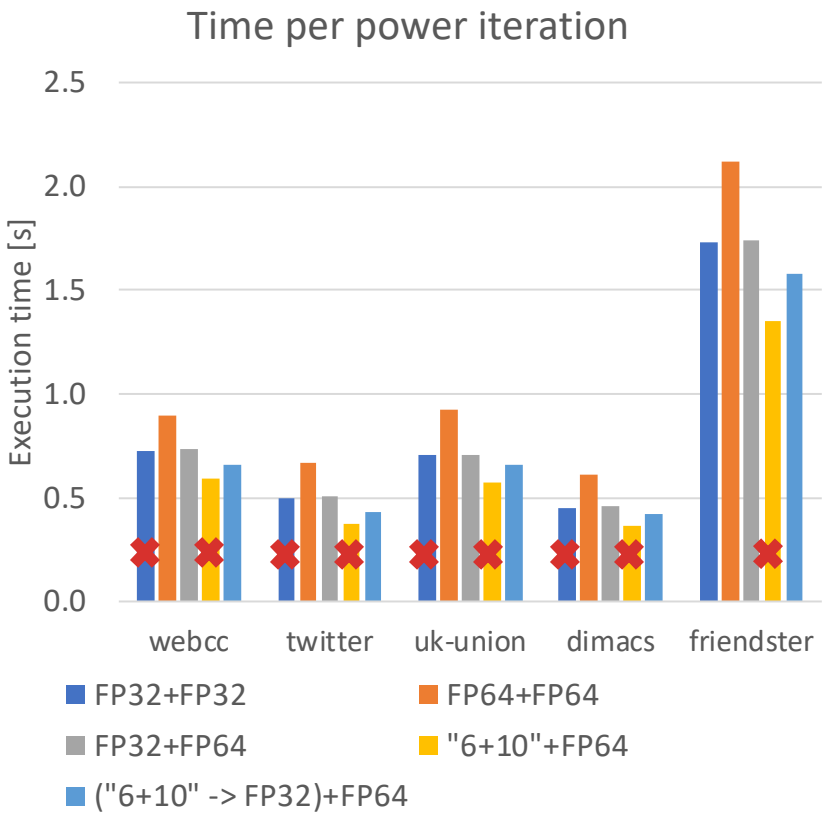
Twitter graph

"6+10" reduced precision format supports convergence to within 1e-4

Need to switch precision when convergences stalls:

- Residual error < 1e-4
- or improved by less than 10%

QUEEN'S UNIVERSITY BELFAST



PERFORMANCE RESULTS PAGERANK

Data formats for PageRank values in memory and accumulator in register

8-way parallelism

✖ does not converge

End-to-end speedup when switching ”6+10” to FP32 is 8.4% to 16.4%

Intel Xeon Gold 6130, 2 sockets, 16 cores per socket, AVX-512

QUEEN'S UNIVERSITY BELFAST

CONCLUSION

Graph analytics performance strongly impacted by memory issues

Data layout matters, need to differentiate by memory access patterns

Load-balanced partitions can be obtained by low-overhead vertex relabeling

Vectorisation requires handling of intra- and inter-thread conflicts

Memory pressure can be relieved by reduced-precision computation

Graptor provides high-level map abstractions and optimizes performance ”under the hood”

QUEEN'S UNIVERSITY BELFAST

WITH THANKS TO

- Jiawen Sun
- Amir Sabbagh Molahosseini
- Mohsen Koohi Esfahani
- Giorgis Georgakoudis
- Mahwish Arif
- Dimitrios S. Nikolopoulos

QUEEN'S UNIVERSITY BELFAST