Data Vortex® Technologies

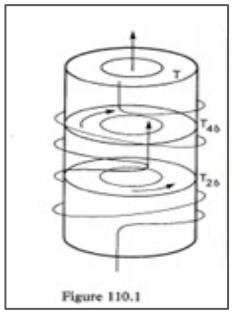
High Level Description and POC Results Summary

November 2023

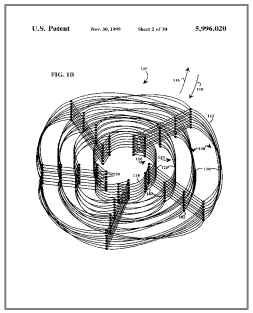
www.datavortex.com
john.labry@datavortex.com

For Those Not Already Familiar

The Scottish Café
Lwow, Poland
c. 1935



Fundamenta Mathematica
Reed and Kuperberg
1981



"A Multi-Level Minimum-Logic Network"
Coke S. Reed, PhD
US Patent No. 5996021, 1999

- The Data Vortex® is a highly efficient network fabric topology derived from the solution to a very complex problem in particle flow dynamics.
- It has been around for over two decades in various instantiations and limited deployment.
- For many years, the only interface protocol available for connecting host servers with the Data Vortex® was PCIe, which presented complexity and latency not optimum for taking best advantage of the network's capabilities.
- Even then, the performance per core was the best on the Graph 500 BFS problem

Fast Forward to Today

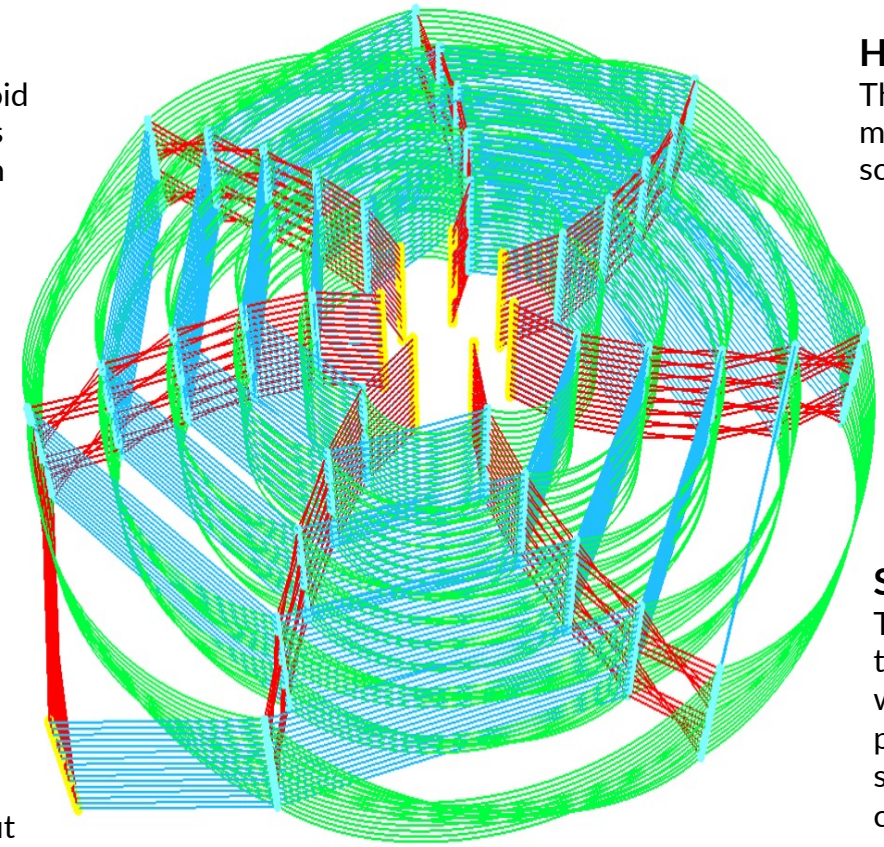
- Computing heterogeneity with application accelerators
- Accelerators:
  - Allow for less complex and lower latency I/O protocols
  - Data Vortex as an accelerator interconnect
  - Enables dramatically improved scaling of accelerator performance

Advantages of the Data Vortex® Network for Large-Scale Problems with High Network Traffic

Inherent Arbitration
Packets self-route to avoid collisions. Data is always flowing within the switch core. Packets are never dropped.

Fine Grained
Optimized for small packets (8 bytes up to cache line length).

Very Low and Uniform Latency Regardless of Scale
Packets move at a very low and consistent latency regardless of network size and without congestion.



The Data Vortex® Topology by Dr. Coke S. Reed

High Bandwidth
Throughput per node is maintained regardless of scale.

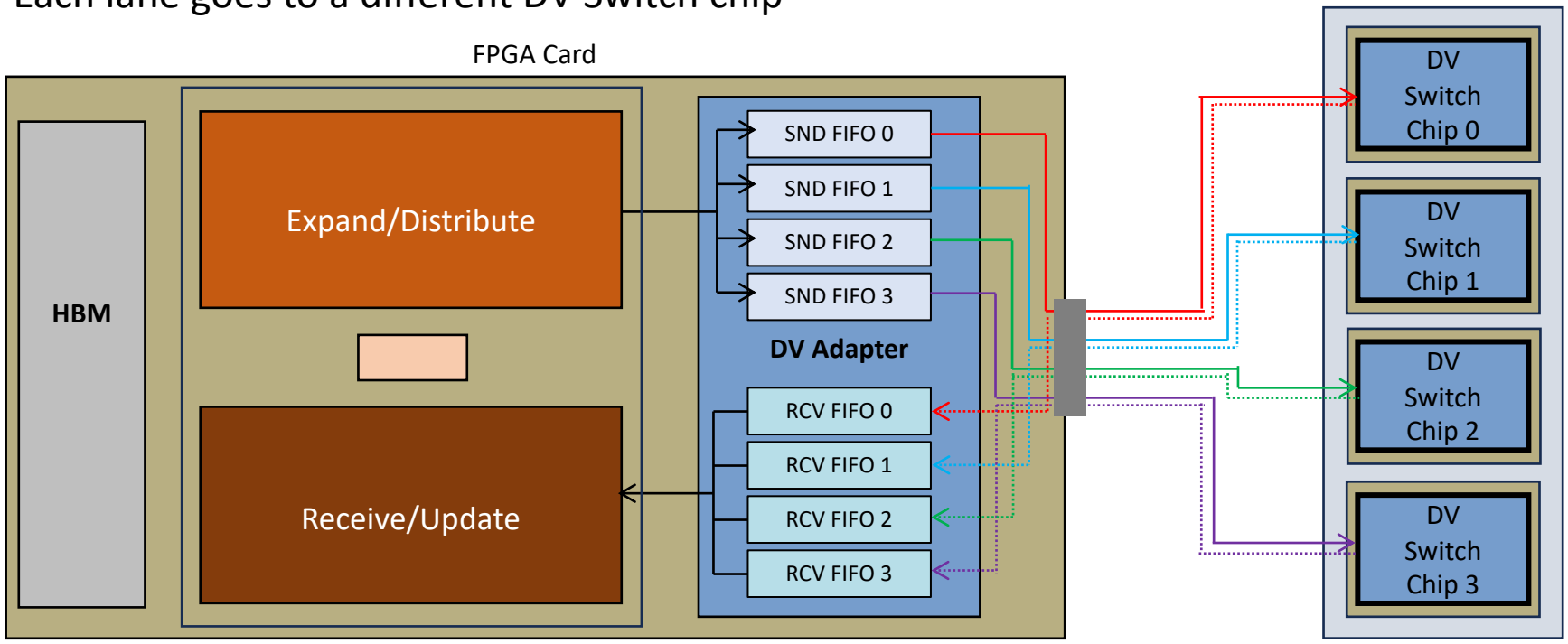
High Radix
The network can be instantiated in hardware at very large size, reducing the number of hops.

Scalable & Flat
The network allows scaling to thousands of nodes while maintaining uniform performance. Near linear scaling for unstructured communication patterns.

"The Data Vortex® as the central element of an innovative, radial network topology enabling a highly efficient accelerator message passing architecture or a unified memory space"

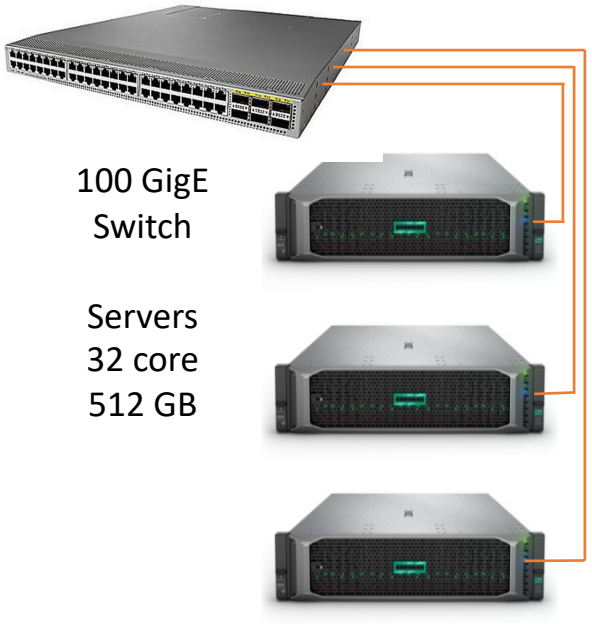
An Example Accelerator Configuration

- QSFP28 (Quad Small Form-factor Pluggable) ports are used to connect accelerator cards to independent DV switch chips
- Each QSFP28 port has **four bi-directional lanes**
- There is a **Send FIFO** and a **Receive FIFO for each lane**
- Each lane goes to a different DV Switch chip

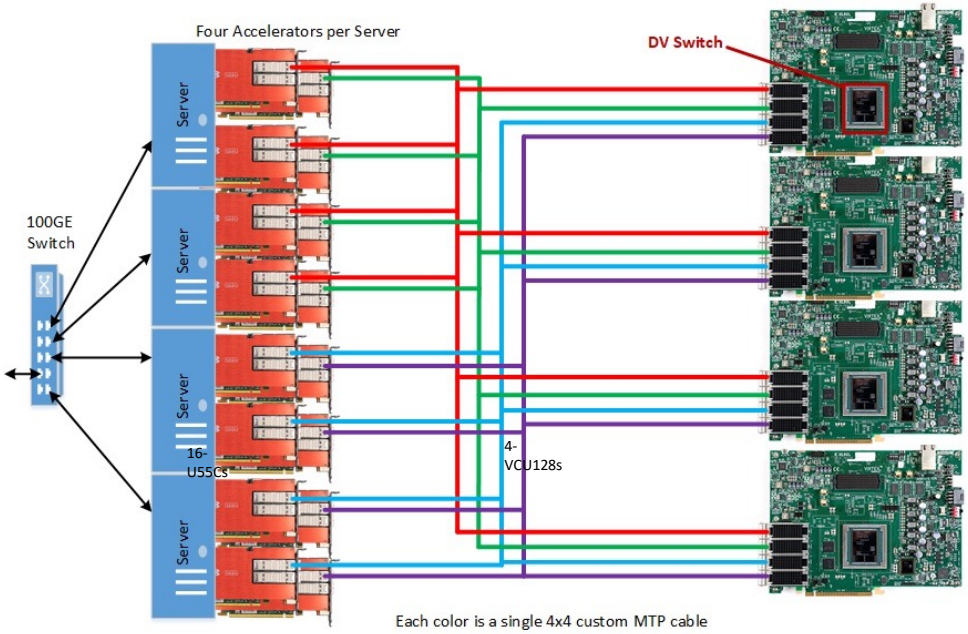


The DV switch fabric is implemented on multiple FPGA chips and the QSFP lanes are not aggregated and can be utilized separately.

A Current Collaboration Proof of Concept



VS



The DataVortex Switch capability has been demonstrated connecting 16 AMD/XILINX U55c FPGA cards
Near-perfect scaling from 5 cards to 16 cards was achieved for an example nHop graph problem

POC Results Demonstrate Near Linear Scaling of Accelerator Performance with the Data Vortex Fabric and 3500 Times the Performance of the Problem on a Server Cluster

Example Problem

Modified Stochastic Kroneker Graph
(Directed Graph)
Scale 27, Edge Factor 16
Probability: A=0.45, B=0.20
10,000 Source/Destination Pairs
123M Vertices, 2.1B Edges
Avg. Out Degree: Approx. 17
Largest Out Degree: Approx. 19,000

This is a synthetic graph
Parameters were chosen to
closely approximate a
real-world problem

The problem was run multiple times, on 5 thru 16 cards in order to get an accurate representation of scaling of the accelerator performance on the 4-Hop algorithm across the Data Vortex fabric

Server Cluster Baseline 4Hop Runtime	Accelerators with Data Vortex Backplane		
	Number of U55C Cards	4Hop Algorithm Runtime	4Hop Runtime incl. Loading of FPGA Kernels
240200 sec (2.8 days)	5	148.7 sec	166.6 sec
	6	124.7 sec	141.4 sec
	7	107.5 sec	124.3 sec
	8	95.2 sec	111.8 sec
	9	85.9 sec	102.2 sec
	10	78.1 sec	94.0 sec
	11	72.4 sec	88.3 sec
	12	67.1 sec	82.6 sec
	13	63.5 sec	79.2 sec
	14	57.5 sec	73.7 sec
	15	55.1 sec	70.8 sec
	16	52.7 sec	68.5 sec

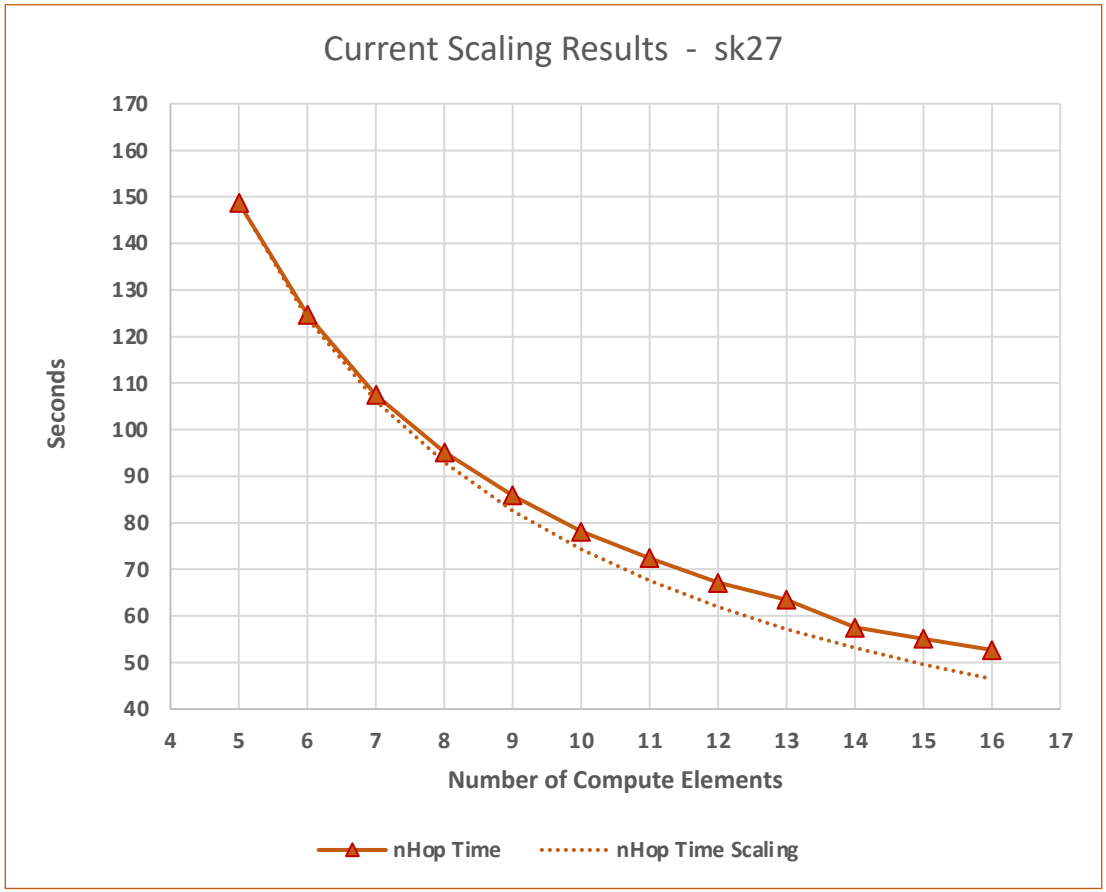
At 16 accelerators, total runtime was 3500x the server cluster baseline

FPGA Accelerator Performance Scaling when Connected with the Data Vortex

FPGA accelerators perform better on the problem type than server CPUs

With the Data Vortex® the performance of the system increases nearly linearly as additional accelerator cards are applied to the problem

4-Hop Algorithm Scaling



Dotted line represents perfect linear scaling
Solid line represents measured results

Data Vortex Switch Latency Test - 16 Cards
Very Low Latency Persists at Scale with a Heavily Loaded Fabric

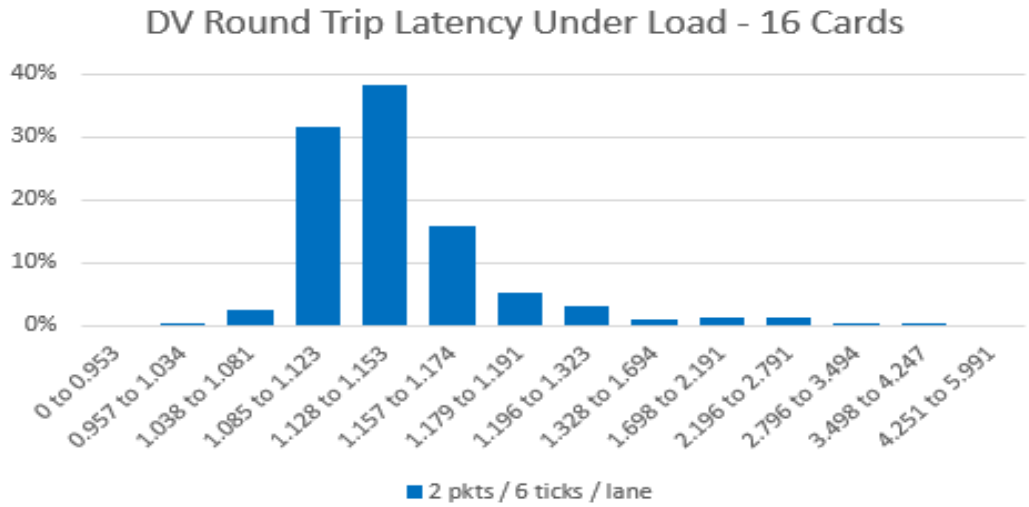
Latency is measured from an **application kernel on card *i***, through the DV network to **an application kernel on card *j***, and then back to the **application kernel on card *i***

Each card sends **two billion echo-request packets** (1B per lane) to random destination cards

Each card, on receiving an echo-request packet, sends an echo reply packet back to the source of the echo-request packet

DV round-trip latency on a 16 card run (measured throughput of 60.2 Gbits/sec per card)

ranges between .95 and 5.98 microseconds, average 1.17 microseconds
ranges between .95 and 1.19 microseconds for 93.6% of packets
ranges between .95 and 1.32 microseconds for 96.7% of packets



Non-Linear Scale
In Microseconds

The longer tails seen in the histogram result from the Xilinx Aurora protocol retraining connections periodically.

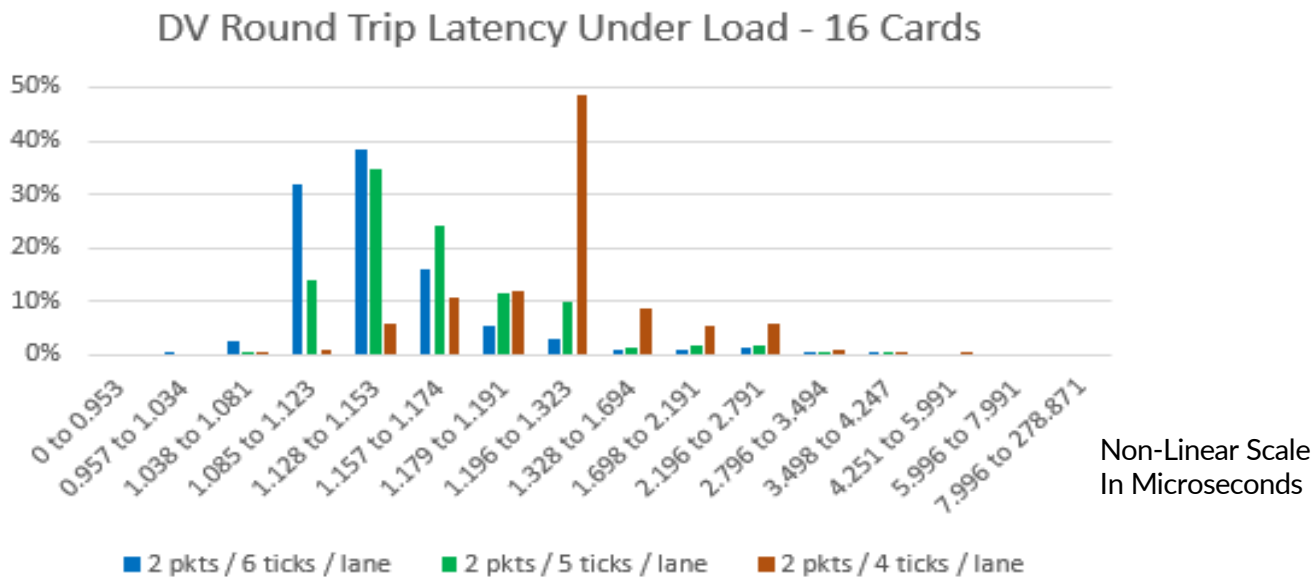
DV Latency - 16 Cards

Latency is measured from an **application kernel on card *i***, through the DV network to **an application kernel on card *j***, and then back to the **application kernel on card *i***

Each card sends **two billion echo-request packets** (1B per lane) to random destination cards

Each card, on receiving an echo-request packet, send an echo-reply packet back to the source of the echo-request packet

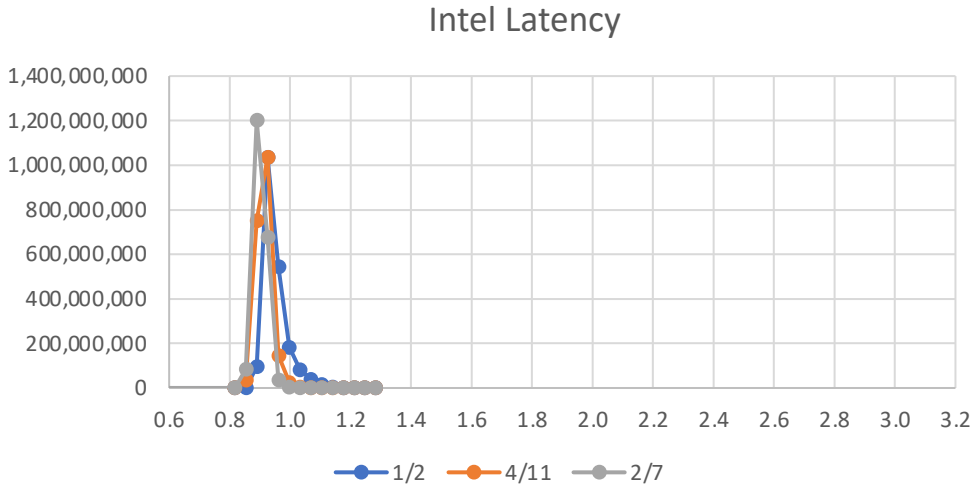
	throughput/card	max latency	avg latency	
2 pkts / 6 ticks / lane	60.2 Gbits/s	6.0 us	1.17 us	96.7% < 1.32 us
2 pkts / 5 ticks / lane	72.2 Gbits/s	6.6 us	1.20 us	96.4% < 1.70 us
2 pkts / 4 ticks / lane	90.2 Gbits/s	9.5 us	1.38 us	98.4% < 2.79 us



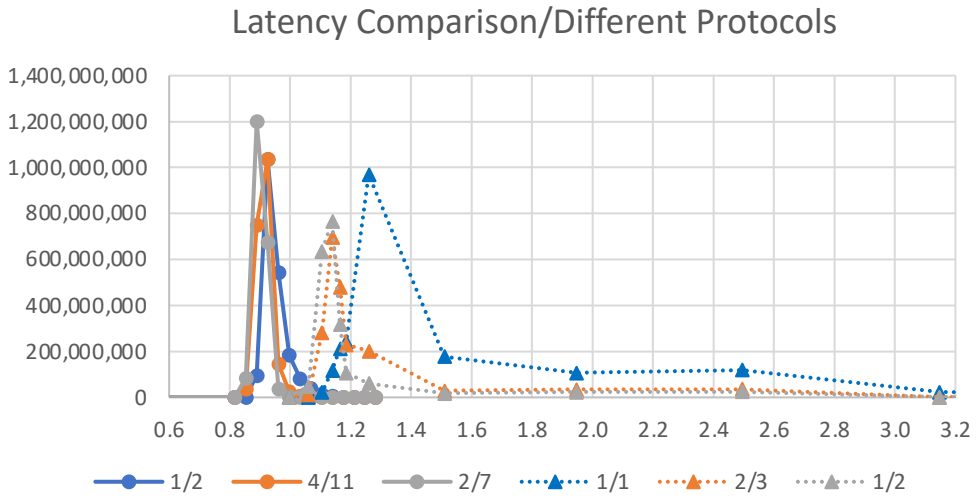
The longer tails seen in the histogram result from the Xilinx Aurora protocol retraining connections periodically.

Preliminary work with Intel AgileX cards

Using a DV slightly modified version of the Intel SuperLite protocol on the AgileX cards, (which took one person only a couple of weeks to port), round-trip, kernel-to-kernel latencies were exceptional and consistent. (Approximately 800 to 900 nanoseconds for 1-Billion packets per lane with no significant tails).



The bottom graph shows the comparison between the Intel and Xilinx protocols. Though the results are still quite good using the Xilinx protocol, it was originally designed for use with ethernet and did not have the benefit of DV modification, so has some additional overhead.



Relevance

Data Sparsity

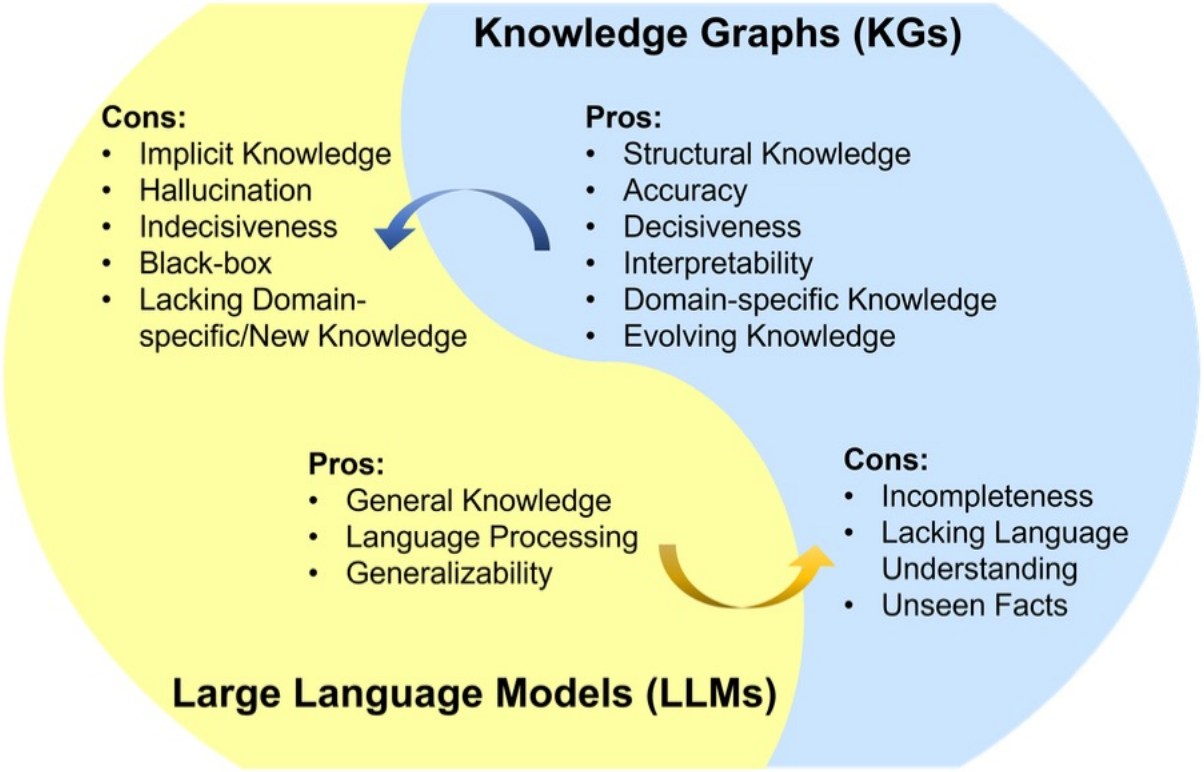
Small Message Efficiency

Big Graphs

Inferencing for Generative AI

Note: Most real-world problems involve sparsity

Progressing Toward the Combination of Graph Functionality with AI Models



Unifying Large Language Models and Knowledge Graphs: A Roadmap

Shirui Pan, *Senior Member, IEEE*, Linhao Luo,
Yufei Wang, Chen Chen, Jiapu Wang, Xindong Wu, *Fellow, IEEE*

JOURNAL OF LATEX CLASS FILES, VOL. 14, NO. 8, AUGUST 2021

Possibilities

1. The Data Vortex IP on die
   - Easily tiled, memory semantics, very low latency, non-blocking
   - Traversing the switch core is on average 10 ticks
2. On package
   - As a low latency chiplet interconnect
3. Memory semantics at scale
   - Allocation of large shared pools of memory and compute resources
4. As a shared memory fabric
   - Simpler programming models
5. NVMe over fabric