Pacific Northwest
NATIONAL LABORATORY
Proudly Operated by Battelle *Since 1965*

System Software for Data-Vortex-based Environments

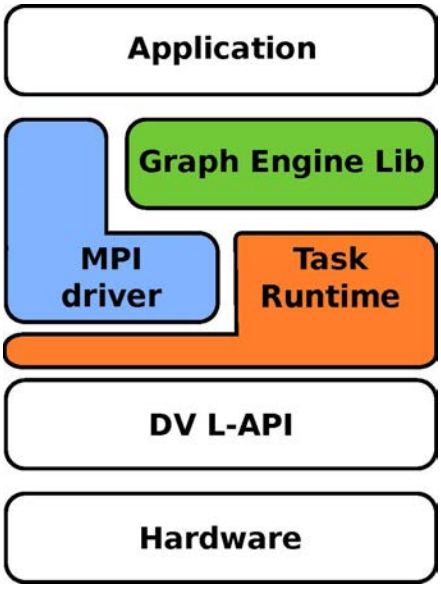
CHALLENGE

Data Vortex networks offer a unique environment to deploy high-performance data analytics applications on clusters. However, the current system software stack is relatively low-level and requires programmers to understand architectural and networking protocol details. Therefore, porting existing code or algorithms to a Data Vortex system is cumbersome. We are developing a high-level system software stack that abstracts most of the low-level details and provides functionalities, such as memory management or message passing, commonly available in other computing environments.

Develop a high-level system software stack for Data Vortex systems to improve performance and programmability of both data analytics and traditional high-performance computing applications.

CURRENT PRACTICE

The current application programming interface (API) for Data Vortex systems sits at a very low level in the system software stack. As such, the programmer is required to understand a large number of architectural and networking communication protocol details before porting even simple kernels. As a result, porting applications to Data Vortex systems requires implementing even basic communication primitives. The programmer must be familiar with low-level architectural details and system components, such as how to use a direct memory access (DMA) engine, which is not typical for most domain scientists. Moreover, the current parallel programming model requires users to manually manage the static random-access memory (SRAM), understand the physical layout of their data structures in the Data Vortex SRAM, and implement their own synchronization primitives.



System software stack.

U.S. DEPARTMENT OF ENERGY

TECHNICAL APPROACH

Distributed Task Parallel Programming Model and Runtime

The base of our designed system software stack consists of a programming model and programming model runtimes for Data Vortex systems that abstract most of the low-level architectural details. The system software being developed consists of the following components:

- **Memory management.** A lightweight dynamic system that provides basic memory management functionalities, such as allocating/releasing memory regions and caching of memory management objects, as well as support for virtual global address space.
- **High-level messaging primitives.** Our system software stack will provide other basic communication primitives, such as "broadcast," "all_to_all," and reduction operations that are not currently implemented in the latest Data Vortex APIs.
- **Active message support.** In addition to classical message operations, such as "send" and "receive," high-performance data analytics applications might benefit from active message support. This includes operations started by a source node and completed at a destination node.
- **Task support.** We will evaluate how task-based programming models can efficiently leverage Data Vortex capabilities when migrating tasks from one node to another while fully exploiting symmetric multiprocessor and non-uniform memory access (NUMA) capabilities within a single node.

Graph Engine Library

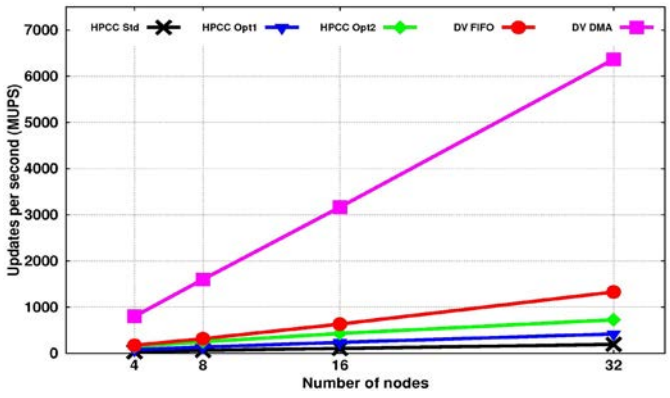
We are also designing a graph engine library (GELib) for Data Vortex systems that includes a collection of graph algorithms, from breadth-first search (BFS), to Dijkstra's Shortest Paths and connected components. In addition, we are designing APIs so existing applications based on the Boost Graph Library (BGL) or Graph Engine for Multithreaded Systems (GEMS) APIs will seamlessly integrate with GELib.

Message Passing Interface Driver

Most parallel applications designed for distributed systems rely on the standard message passing interface (MPI) library to exchange data among computing nodes. Unless the programmer completely rewrites the communication layer, these applications currently cannot be executed on Data Vortex systems. We will provide an MPI compatibility layer that will enable programmers to run MPI applications natively on the Data Vortex network and use a combination of Data Vortex and InfiniBand networks simultaneously.

IMPACT

Our system software stack will enable a larger set of programmers to port their code to Data Vortex systems, as well as improve performance of both data analytics and traditional high-performance computing applications. So the community can benefit from this work, we plan to release the Data Vortex system software stack as an open-source product.



Comparison of varying versions of giga updates per second (GUPS) benchmark for MPI/ InfiniBand (HPCC Std, Opt1, Opt2) and Data Vortex (DV FIFO, DMA) at scale. The y-axis reported the aggregated number of memory updates per second.

Contacts

Roberto Gioiosa
Principal Investigator
(509) 375-3605
Roberto.Gioiosa@pnnl.gov

John R. Johnson
Program Director
(509) 375-2651
John.Johnson@pnnl.gov

Pacific Northwest NATIONAL LABORATORY
Proudly Operated by Battelle Since 1965

PNNL-SA-121613