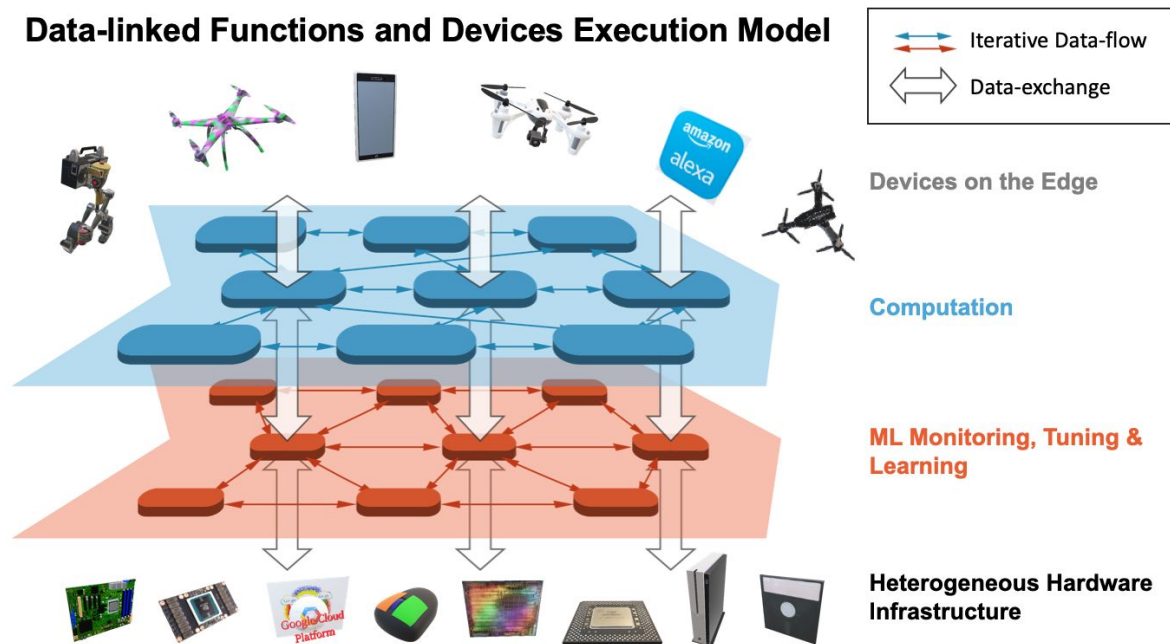


## Twister2 Demonstrations for BDEC2

Geoffrey Fox, Vibhatha Abeykoon, Selahattin Akkas, Kannan Govindarajan, Gurhan Gunduz, Supun Kamburugamuve, Niranda Perera, Ahmet Uyar, Pulasthi Wickramasinghe, Chathura Widanage, *Digital Science Center, Indiana University Bloomington; gcf@indiana.edu*

We can demonstrate a novel middleware Twister2 [1], [2] that supports programming in components linked together by high performance messaging. This model captures recent cloud (native) ideas such as Function as a Service and microservices; it also supports geographically distributed computing as one gets when linking edge device, fog, and data-center components. One also sees a need for linked components in many familiar HPC scenarios such as those linking different simulations for different parts of a complete system. Recent studies of the use of Machine Learning (ML) to enhance HPC suggest this architecture sketched in the figure below.



We can describe all these scenarios as horizontally and vertically Data-Linked Functions and Devices. Here the computation (data analytics or simulation) and Machine learning are executing in tandem with the machine learning monitoring, tuning and learning results of the computation. The ML-computation interaction can be at the fine grain (function) component level. The breakup into functions leads to horizontal data flow in the figure within each of the two segments with vertical linkage to represent the ML-computation or the edge-fog-cloud linkage. As one builds more domain-specific hardware we can also see a vertical linkage between software and hardware units. Data flow is familiar for many successful workflow environments but the programming model just described requires fine-grain dataflow with greater performance challenges. Twister2 addresses this with Twister2:Net [3] high-performance communication subsystem that outperforms those in systems like Spark and Flink. It has four distinct message

systems: as well as dataflow DFW, it supports native MPI, a custom map-collective model Harp and the standard streaming publish-subscribe based approach. Each of these is suitable in different circumstances.

Twister2 assumes that it is integrated with separate high-performance simulation or data analytics such as that in Tensorflow. It is available for some demonstrations already as it offers the full Apache Storm or Heron streaming capability with built-in high performance for edge-fog-cloud use cases. It has two major additional capabilities that will be operational in 6 months; high-performance Python binding and connection to Apache Beam for the SQL query capability and orchestration offered by Beam. With these added, Twister2 will support a rich set of demonstrations and has a tutorial available [4].

The current Twister2 release supports resource provisioning in standalone mode, Kubernetes, Mesos, Slurm, and Nomad with several task schedulers and a web GUI for monitoring. The task graph module allows the creation of dataflow graphs for streaming and batch analysis including iterative computations with data caching. Twister2:TSet [5] supports distributed data with similar functionality to Spark RDD, Flink DataSet and Heron Streamlet. There are programming API's for streaming and batch applications with Apache Storm compatibility, as well as API's for operators, task graphs, TSets, and data and message level communication in different modes discussed above. Local file systems and distributed storages as in HDFS or shared NFS are supported while support for NoSQL and RDMS databases are planned for upcoming releases. Think of Twister2 as Apache Spark or Flink with high performance built in from scratch and currently able to support streaming computations, data operations in batch mode, dataflow and iterative computations.

The extensive messaging seen in the picture suggests the importance of hardware that supports high speed messaging and disk access. Natural hardware would be Omni-Path or Infiniband networking, Optane style memory and NVMe disks. The CPU's should support both machine learning and computation with both running close to each other and not separated in distinct clusters.

#### Questions Answered

1. *What innovative capabilities/functionalities will the proposed candidate platform demonstrate?*  
We are not aware of high-performance software environment offering the range of capabilities supported by Twister2
2. *What applications/communities would/could be addressed?*  
Initially we will focus on edge-fog-cloud applications as well as the "learning everywhere" MLforHPC applications
3. *What is the "platform vision," i.e. what kind of shared cyberinfrastructure (CI) for science would the further research/design/development of this platform lead to?*  
This is described above as the integration of diverse functions or microservices across

ML and computation. It should enable integration of multiple application models with good fine grain parallel performance

4. *How available/ready/complete is the set of software components to be used to build the demonstrator?*

It is available for experimentation (0.2.0 release [1]) now with broad capabilities supported 6 months from now.

5. *As far as one can tell at this early date, to what extent can this be done with existing and/or otherwise available hardware/software/human resources?*

We listed the preferred hardware above but standard HPC clusters can be used especially if they have local SSD or NVMe disk storage

6. *What is the potential international footprint of the demonstrator?*

We don't see any problems with international use. All software is available from Github with an Apache 2 license.

## References

- [1] "Twister2 Release," *Twister2 high performance data analytics hosting environment that can work in both cloud and HPC environments*. [Online]. Available: <https://github.com/DSC-SPIDAL/twister2/releases>. [Accessed: 26-Oct-2018]
- [2] Kamburugamuve, Supun and Govindarajan, Kannan and Wickramasinghe, Pulasthi and Abeykoon, Vibhatha and Fox, Geoffrey, "Twister2: Design of a Big Data Toolkit," *Concurr. Comput.*, vol. EXAMPI 2017 workshop at SC17 conference, 2019 [Online]. Available: [http://dsc.soic.indiana.edu/publications/twister2\\_design\\_big\\_data\\_toolkit.pdf](http://dsc.soic.indiana.edu/publications/twister2_design_big_data_toolkit.pdf)
- [3] Supun Kamburugamuve, Pulasthi Wickramasinghe, Kannan Govindarajan, Ahmet Uyar, Gurhan Gunduz, Vibhatha Abeykoon, Geoffrey Fox, "Twister:Net - Communication Library for Big Data Processing in HPC and Cloud Environments," in *Proceedings of Cloud 2018 Conference*, San Francisco [Online]. Available: [http://dsc.soic.indiana.edu/publications/twister\\_net.pdf](http://dsc.soic.indiana.edu/publications/twister_net.pdf)
- [4] G. Fox, "Twister2 Tutorial at BigDat2019 Cambridge UK January 8-11 2019." [Online]. Available: <https://twister2.gitbook.io/twister2/tutorial>. [Accessed: 30-Jan-2019]
- [5] Pulasthi Wickramasinghe, Supun Kamburugamuve, Kannan Govindarajan, Vibhatha Abeykoon, Chathura Widanage, Niranda Perera, Ahmet Uyar, Gurhan Gunduz, Selahattin Akkas, Geoffrey Fox, "Twister2:TSet High-Performance Iterative Dataflow," Apr. 2019 [Online]. Available: [http://dsc.soic.indiana.edu/publications/dataflow\\_twister2\\_new.pdf](http://dsc.soic.indiana.edu/publications/dataflow_twister2_new.pdf)