# Summary Perspectives of the SPIDAL Project NSF #1443054 from 2014-2020

CIF21 DIBBs: Middleware and High Performance Analytics Libraries for Scalable Data Science
PI: G. Fox, Co-PIs: Madhav Marathe, Shantenu Jha, Judy Qiu, Fusheng Wang
Institutions: Arizona State, Indiana (lead), Kansas, Rutgers, Stony Brook, Virginia, and Utah.
Award #: 1443054

## MLforHPC and HPCforML

- Project mainly addresses **HPCforML** enabling high performance big data
- **MLaroundHPC** broadly applicable but current **use is nonuniform** across domains
  - we need to improve Cyberinfrastructure to make MLforHPC more effective for **more users**
- Use of modest DL network to **map material/potential drug structure to properties** (generalized QSAR) with simulation and observation: Advanced Progress
- Learn **surrogates** for **large scale simulations**: initial small scale results
- Use of MLforHPC in **agent-based systems** (learn agents): Very promising but few results
- **Macroscopic** Structure as in learn complex multi-particle **potentials** scaling to $N^2$: many great successes
- **Learn Collective coordinates and guide ensemble** computations: dramatic progress with speedups up to $10^8$
- **Microscale**; learn dynamics of small scale such as clouds, turbulence: Interesting results but much more to do
- Use of **Recurrent NN**'s to represent dynamics (**learn numerical differential operators**): Promising but only studied in small problems


## Community Driven High-Performance Big Data

**for bio-physical applications based on HPC, distributed systems, network science, GIS and machine/deep learning**

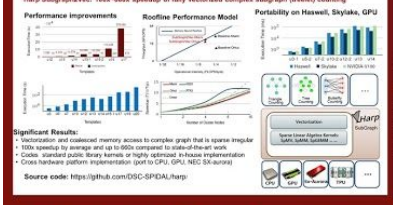
### Completed Activities

- MIDAS Pilot Jobs HPC Task Management
- High Performance for Java/C++ for Machine Learning
- NIST Big Data Application Analysis: Features of data intensive Applications deriving 64 Convergence Diamonds
- HPC-ABDS: Cloud-HPC Interoperable software with performance of HPC and rich functionality of commodity Apache Stack
- Implementation of HPC and Clouds with DevOps
- Image Processing and Machine Learning Toolkit http://hpcanalytics.org

## HPC Big Data Cloud Convergence

### Harp-DAAL: A HPC-Cloud convergence Data Analytics framework

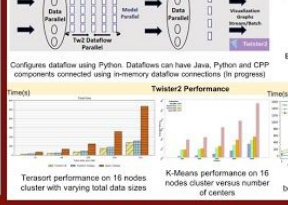
Harp Subgraph2Vec: 100x–660x speedup of fully vectorized complex subgraph (treelet) counting




**Significant Results:**
- Vectorization and coalesced memory access to complex graph that is sparse irregular
- 100x speedup by average and up to 660x compared to state-of-the-art work
- Codes: standard public library kernels or highly optimized in-house implementation
- Cross hardware platform implementation (port to CPU, GPU, NEC SX-aurora)

**Source code:** https://github.com/DSC-SPIDAL/harp

### High Performance Deep Learning: Data, Model, Pipelined Parallel




Configures dataflow using Python. Dataflows can have Java, Python and CPP components connected using in-memory dataflow connections (In progress)

**Twister2** replaces Spark and Hadoop with native high performance

**Twister2 Apache Beam Runner**

Apache Beam API

Twister2 Runtime

Enables high performance Beam pipelines

**Twister2 Performance**



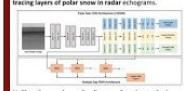

## Polar Science Community

### Structure of polar ice sheets from radar informatics

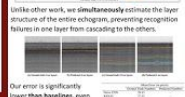
Ongoing Work

Deep learning has been successful for segmenting photos, but parsing echograms involves reasoning about weak, noisy evidence and prior information (e.g., physical constraints).

We propose a novel Convolutional Neural Network for tracing layers of polar snow in radar echograms.



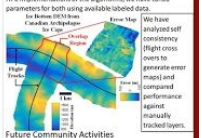
Unlike other work, we **simultaneously** estimate the layer structure of the entire echogram, preventing recognition failures in one layer from cascading to the others.



Our error is significantly **lower than baselines**, even without knowing the number of layers *a priori*.

Community Contributions

Our Viterbi-based layer tracker is now used routinely to process 2D radar data collected by CReSIS (50-100 TB/year), and our tree-reweighted sequential (TRW-S) surface tracker is used routinely by three groups for 3D reconstructions. Using HPC implementations of the algorithms, we have tuned parameters for both using available labeled data.




We have analyzed self consistency (Right cross overs to generate error maps) and compared performance against manually tracked layers.

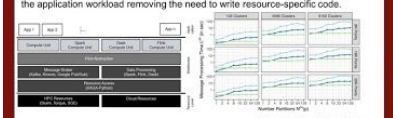
Future Community Activities

- Establish polar radar center to maintain polar radar software "toolbox" and develop ingest interfaces for more radar systems to increase access to SPIDAL tools.
- Investigate new problems: cross-track slope estimation, model order estimation, signal processing integration.
- Integrate physical model constraints into algorithms (e.g. weather models, ice flow laws, etc.)

## Streaming Data Systems

### Pilot-Streaming (deployed)

Pilot-Streaming provides a unified abstraction for resource management for HPC, cloud, and serverless, and allocates resource containers independent of the application workload removing the need to write resource-specific code.

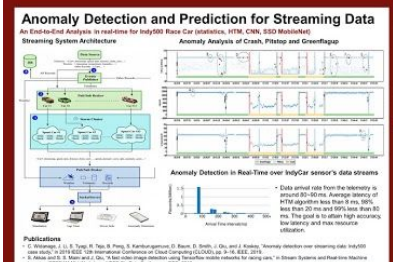



**Pilot-Streaming on Serverless High-Level Architecture and Interactions:** Pilot-Streaming enables the unified management of batch and streaming compute tasks on serverless infrastructure. It orchestrates the setup of Kinesis and Lambda and manages the streaming workload.

**Message Processing Time $L^{PS}$ for K-Means on AWS Lambda and HPC executed via Pilot-Streaming:** Broken down by # partitions, message size, and workload complexity. The processing times increase with the message size and a higher # of clusters. While for Lambda the processing times remain constant with increasing parallelism, thy increase for Dask/Kafka on HPC due to the use of shared filesystem and network resources.

### Anomaly Detection and Prediction for Streaming Data

An End-to-End Analysis in real-time for IndyCar (statistics, HTM, CNN, SSD MobileNet)




- Data arrival rate from the telemetry is around 80–90 ms. Average latency of HTM algorithm less than 8 ms, 98% less than 20 ms and 99% less than 80 ms. The goal is to attain high accuracy, low latency and max resource utilization.

**Publications**

C. Wildrangel, J. Li, S. Tyagi, R. Teja, B. Peng, S. Kambhampamoot, D. Baum, D. Smith, J. Qiu, and J. Kosiey, "Anomaly detection over streaming data: IndyS00 case study," in 2019 IEEE 12th International Conference on Cloud Computing (CLOUD), pp. 9–16, IEEE, 2019.
S. Akbar and S. S. Mann and J. Qiu, "A fast video image detection using TensorFlow mobile networks for racing cars," in Stream Systems and Real-time Machine Learning (STREAM ML) Workshop of IEEE Big Data Conference, IEEE, 2019.

Source code: https://github.com/DSC-SPIDAL/IndyCar

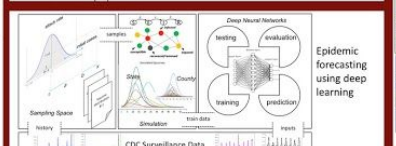
## Network Science Community

### Network Analytics

- **Substantial Community Contributions of Project:** New advances in sequential and parallel algorithms for a broad class of network problems
  - Subgraph detection and counting, anomaly detection, dense subgraph and community detection
  - Parallel network generation: generating very large instances from many random graph models and by parallel edge switching, with different kinds of constraints
  - Applications: public health, social network analysis, scaling studies of network algorithms, sensitivity analysis, synthetic population generation
- **Ongoing Project work:** Algorithms for dense subgraphs, communities in dynamic and temporal networks, and parallel network generation
- **Future Community Activities:** Integration of algorithms within CINES cyberinfrastructure

### Epidemic analysis and forecasting

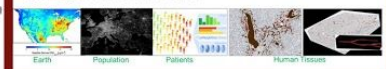
- **Overview.** We have developed a theory/simulation guided machine learning method (TDEFSI) to forecast influenza dynamics. Our methods provide a fundamentally new way to train and use DNNs when measured data is sparse.
  - TDEFSI produces accurate weekly high-resolution ILI forecasts from flat resolution observations. We focus on state level ILI surveillance and state (flat-resolution) or county level (high-resolution) ILI forecasts.
  - TDEFSI uses a two-branch neural network model for ILI forecasting. It combines within-season observations (observed data points of the previous weeks that characterize the ongoing epidemic) and between-season historical observations (observed data points from similar weeks of the past seasons that characterize general trends around the current week)
- **Community Impact:** Variants of TDEFSI have been used to support our ongoing participation in various forecasting challenges.
- **Current year:**
  - Extend TDEFSI and integrate in production pipelines for forecasting for Accuweather with the following extensions: (i) use it to understand the role of interventions and improved forecasts during such times, (ii) use additional high resolution data that is starting to become available to improve the forecasts and the method
  - Integrate within CINES
- Supporting the **Coronavirus outbreak** mitigation by studying questions such as: (i) risk of importation to US, (ii) possible ways it will spread in US if epidemic takes off here, (iii) possible interventions and preparedness




## Health Science Community

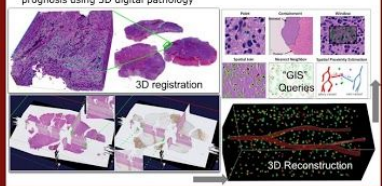
### Spatial Big Data Query Systems

- Managing and querying big spatial data is challenged by explosion of data, multi-dimensions and the complexity of geometric computation
- Developed scalable and efficient spatial big data querying systems running on big data platforms, for both 2D and 3D
  - Hadoop-GIS, SparkGIS and iSPEED (3D)
- Integrative CPU-GPU based high performance 3D spatial queries
- The platforms can be used to support geospatial applications, big data driven public health studies, and digital pathology
- **Future:** Spatial analysis for understanding resource availability for opioid epidemic prevention and intervention



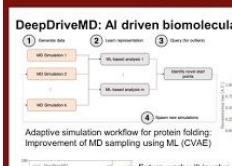

### Digital pathology image analysis

- Created a suite of deep-learning based image analysis tools for level-set based image segmentation, 3D registration, reconstruction, and spatial analysis
- Future: Data analysis for understanding breast cancer treatment and prognosis using 3D digital pathology

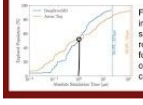



## Biomolecular Science Simulation Community

### DeepDriveMD: AI driven biomolecular simulations on HPC




Adaptive simulation workflow for protein folding: Improvement of MD sampling using ML (CVAE)



Future work will involve intermediate state sampling and reinforcement learning for docking scores in order to manage computational costs

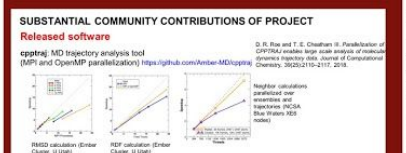
Two folding simulations (completed) show the increase in the data gathered from the number of concurrent simulations across GPUs

### SUBSTANTIAL COMMUNITY CONTRIBUTIONS OF PROJECT

**Released software**

**cpptraj**: MD trajectory analysis tool (MPI and OpenMP parallelization) https://github.com/Amber-MD/cpptraj

D. R. Roe and T. E. Cheatham III. *Parallelization of CPPTRAJ enables large scale analysis of molecular dynamics trajectory data.* Journal of Computational Chemistry, 39(25):2110–2117, 2018.




**PMDA**: parallel molecular dynamics analysis library (task-parallelization) https://www.mdanalysis.org/pmda/

Shujia Fan, Max Linke, Ioannis Paraskevakos, Richard J. Gowers, Michael Gecht, and Oliver Beckstein. *PMDA - Parallel Molecular Dynamics Analysis.* In Proceedings of the 18th Python in Science Conference, pages 134 – 142, Austin, TX, 2019. SciPy. DOI: 10.25080/Majora-7ddc1dd1-013

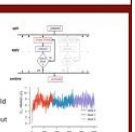
**alchemlyb**: general library for processing and analyzing free energy data from MD and MC simulations https://github.com/alchemistry/alchemlyb

D. Dotson, O. Beckstein, D. Wille, I. Kenney, ahuaii, irjp0703, H. Lee, V. Lim, B. Allen, and M. S. Barhaghi. alchemistry/alchemlyb: 0.3.1. Jan. 2020. DOI: 10.5281/zenodo.3610364

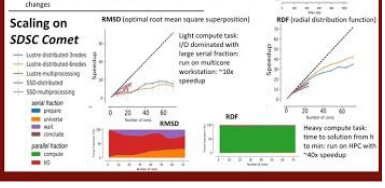
### Parallel Molecular Dynamics Analysis (PMDA)

https://www.mdanalysis.org/pmda/

- Python package (release 0.3.0, GPL v2 license)
- Parallelizes widely used **MDAnalysis** Python package for MD trajectory analysis
- *split-apply-combine* trajectory analysis with *task-based parallelization*: build and executed directed acyclic graph with *Dask* (https://dask.org)
- Flexible: runs on multicore laptops/workstations to supercomputers without changes




**Scaling on *SDSC Comet***



# 1 Introduction

The SPIDAL (Scalable Parallel Interoperable Data Analytics Library) project was begun in Fall 2014 and has reached a technical completion in Fall 2020 with outreach activities continuing in 2021. The February Poster summarizes the 2020 status and activity very well [1] with previous work through September 2018 summarized in a book chapter [2] with extensive references. This builds on our 21-month report [3] which has much material not repeated in the later reports and paper. Institutions and key people involved were Arizona State (Beckstein), Indiana (Fox, Qiu, von Laszewski), Kansas (Paden), Rutgers (Jha), Stony Brook (Wang), Virginia (Marathe, Vullikanti), and Utah (Cheatham).

# 2 Summary of Accomplishments

## 2.1 Architecture

The project was built around community-driven High Performance Big Data biophysical applications based on HPC, distributed systems, network science, GIS, and machine/deep learning. It involved cyberinfrastructure, algorithms, and applications with seven participating organizations. We were inspired by the beneficial impact that scientific libraries such as PETSc, MPI, and ScaLAPACK have had for supercomputer simulations and hope that our building blocks MIDAS and SPIDAL will have a similar impact on data analytics. The project has an overall architecture built around the twin concepts of HPC-ABDS (High-Performance Computing Enhanced Apache Big Data Stack) software [4] and classification of Big data applications – the Ogres – that defined the key qualities exhibited by applications and required to be supported in software. These underpinning ideas are described in section 2 of the 21-month report [3] and briefly summarized in our 2018 paper [2], which includes a sophisticated discussion of Big Data – Big Simulation and HPC-Cloud convergence [5]. The original big data Ogres work was a collaboration between Indiana University and the NIST Public Big Data Working Group [6] that collected 54 use cases – each with 26 properties. The Ogres were a set of 50 features that categorized applications and allowed one to identify common classes such as Global GML and Local LML Machine Learning. GML is highly suitable for HPC systems while the very common LML and MapReduce categories also perform well on more commodity systems. As another example, “Streaming” was a feature seen in 80% of the NIST applications [7], [8].

## 2.2 Cyberinfrastructure

Our approach to data-intensive applications relies on Apache Big Data stack ABDS for the core software building blocks where we added an interface layer MIDAS – the Middleware for Data-Intensive Analytics and Science, that will enable scalable

applications with the performance of HPC (High-Performance Computing) and the rich functionality of the commodity ABDS (Apache Big Data Stack). Here we developed major HPC enhancements to the ABDS software including Harp based on Hadoop and Cylon/Twister2 based on Heron, Spark, and Flink for both batch and streaming scenarios. Pilot jobs from Rutgers were very successful in resource management and scheduling for high throughput parallel computing on NSF and DoE systems. We contributed with new techniques to get high performance across C++, Java and Python coded systems. MIDAS will allow our libraries to be scalable and interoperable across a range of computing systems including clouds, clusters, and supercomputers. We also recognized two important broad categories HPCforML (CIforAI) or MLforHPC (AIforCI), where our early contributions were in HPCforML but recently we also contributed in the second area [9].

## 2.3 Community Applications and Algorithms

Another major project product was a cross-cutting high-performance data-analysis library – SPIDAL (Scalable Parallel Interoperable Data Analytics Library) [10]. The library has 4 components: a) a core library covering well-established functionality such as optimization and clustering; b) parallel graph and network algorithms; c) analysis of biomolecular simulations (high-performance versions of existing libraries from Utah and Arizona State) and d) image processing in both Polar Science and Pathology.

Community application highlights The project has also led to significant algorithmic advances in machine learning methods for networks, including motif detection, anomaly detection, explainability of clustering, deep learning for epidemic forecasting (TDEFSI in MLforHPC category), and the foundations of dynamical systems on networks. We supported the mitigation of the Coronavirus outbreak with the simulation of different spreading scenarios and possible interventions. For Polar Science, we developed operational ML/DL to locate ice sheet boundaries and snow layers from radar data. In Public Health GIS, we researched and implemented spatial big data query for opioid epidemic prevention and intervention while for pathology, we developed DL based image analysis tools for image segmentation, 3D registration, reconstruction, and spatial analysis. For the major Biomolecular Simulation community, SPIDAL developed PMDA which parallelizes the widely used MDAnalysis Python package for MD (Molecular Dynamics) trajectory analysis. In this area, recent MLforHPC research by us has shown surrogates that improve molecular dynamics simulation performance by very large factors for both short times (using recurrent neural nets) and long time scales (with fully connected networks). We gave a roadmap for other applications [9], [11].

# 3 Details of SPIDAL Middleware Research

## 3.1 MIDAS Pilot Jobs HPC Task Management.

The Rutgers/RADICAL team in collaboration with Fox began the SPIDAL project investigating resource management abstractions and software systems defining the two “ecosystems” and seeking to bridge two hitherto distinct paradigms. We investigated the Pilot concept, first as a way of bridging, then as a way of unifying resource management across HPC and data-intensive systems. Even as we made progress in managing the software divergence and complexity, both platforms started to shift: during the course of SPIDAL, HPC platforms became significantly more heterogeneous in their architecture, where simple multicore nodes were replaced by complex GPU-CPU and manycore architectures, including some transitory architectures (e.g., KNL family). On the other hand, the data-intensive platforms morphed from high-memory and localized compute-data affinity systems (e.g., TACC’ Wrangler) to platforms geared towards deep learning.

## 3.2 Harp Big Data HPC Convergence Middleware

Harp is an HPC-ABDS (High Performance Computing Enhanced Apache Big Data Stack) framework [12] from Qiu that aims to support distributed machine learning and other data-intensive applications. To improve the expressiveness and performance in big data processing, the Harp library is introduced, which provides data abstractions and related communication abstractions and transforms map-reduce programming models into map-collective models. The word “harp” symbolizes the effort to make parallel processes cooperate together through collective communication for efficient data processing, just as strings in a harp can make a concordant sound. Harp can integrate with Hadoop and supports data abstraction types such as arrays, key-values, and graphs with related collective communication operations on top of each type. Several applications are developed based on the Harp framework, including K-means clustering, multidimensional scaling, and PageRank. Being based on Hadoop, Harp has better sustainability and fault-tolerance properties than Twister or Twister4Azure that inspired it.

## 3.3 Cylon and Twister2 Big Data Processing Systems

Cylon [13] and Twister2 [14] are two open-source data analytics projects developed at Indiana University Digital Science centers enabling the processing of large data sets and integrating AI tools with data engineering. Twister2 is a dataflow engine for processing large data sets. It is a flexible, high-performance data processing engine that is part of the MIDAS software environment. The project is open-source and available under the Apache License Version 2.0. Recently Twister2 became compatible with Apache Beam API. Apache Beam is a project originated from Google for large scale

data processing which has roots to original map-reduce papers. Twister2 is one of few other engines such as Apache Spark, Apache Flink, and Samza that has the same capability to work with Apache Beam.
Cylon addresses the need to integrate Python (Jupyter) notebooks and data engineering with data analytics including tools such as Pytorch and Tensorflow. Cylon has a fast and scalable distributed kernel for analyzing structured data and integrates with Python natively to provide access to the rich python ecosystem with high performance. The Cylon project was started in January 2020 and now is in its second release. It has the core distributed operations implemented in C++ and Python APIs are provided for these. Cylon had three papers this year that illustrate its role in data engineering and the Python APIs of Cylon for High-performance computing.

# 4 SPIDAL CommunityApplication Research

## 4.1 Biomolecular Simulations BMS

In the biomolecular simulation field covered by Arizona State and Utah, analysis (in the sense of the original SPIDAL big data ideas) remains important. But a clear growth area compared to the start of the SPIDAL project is the extraction of descriptors for ML approaches, both offline and online/streaming. Being able to use HPC resources efficiently for these related tasks remains an important area.
Many users in BMS are not expert programmers so two key takeaways were:
1) Easily accessible programming languages are important, even if that does not always result in the highest performance out of the box. In data science and most of the physical sciences, this means overwhelmingly Python.
2) Documentation and tutorials are more important than originally anticipated in order for the new software to gain any traction. Initiatives such as Google Season of Docs https://developers.google.com/season-of-docs/ recognized this as a problem for the wider open source community. Academic projects also have a hard time to budget enough time and resources for this crucial task and also don’t necessarily have the expertise to generate appropriate documentation efficiently.

## 4.2 Polar Science

SPIDAL in a Kansas-Indiana collaboration focused on machine learning research on radioglaciology, where only a couple of ML publications existed at the start of the award. Machine learning research is now ongoing in nearly all aspects of radioglaciology (signal processing to image analysis to ice models) and SPIDAL developed algorithms are readily available to the radioglaciology community in an open-source community-built software toolbox hosted on GitHub.  Recent work is [15]

### 4.3 Epidemiology and Network Science

The UVA team in collaboration with Qiu, made several contributions to network science, graph dynamical systems, and public health, both from a theoretical and applied perspective. Our work in network science included developing highly scalable network generators and subgraph detection methods. A number of random graph models have been proposed in network science. However, they don't scale easily, making it difficult to do detailed analyses with such random graph models. We developed some of the most scalable methods for several models and developed a general framework that can handle multiple models. For subgraph detection, we developed the first parallelization of fixed-parameter tractable algorithms, which scale very well, but also give rigorous guarantees, unlike prior methods. We adapted these methods for problems of scan statistics, which are a very commonly used approach for anomaly detection. Our methods were the first to scale to large networked data while providing rigorous guarantees at the same time. In the area of graph dynamical systems (GDS), we developed the foundations of the learnability of GDS by obtaining tight lower and upper bounds on the sample complexity. We also show that our methods work well on both synthetic and real-world networks. Finally, we developed novel deep learning methods for epidemic forecasting [16], which use a theory-guided approach, and provide more robust performance in practice.

### 4.4 Biomedical Research

Biomedical research is increasingly driven by computer science due to the explosion of data, in particular multidimensional data, including spatial,spatial-temporal and imaging data, in both 2D and 3D. The Stony Brook team has made major contributions on spatial big data systems, computational digital pathology, and spatial big data driven opioid epidemic research.

1. We have developed scalable and efficient spatial big data management and querying systems for both 2D and 3D data, which achieves high scalability and efficiency through on-demand querying engines, novel indexing methods, progressive compression, progressive queries, and in-memory processing [17].
2. We have developed a suite of scalable pathology image processing libraries on registration of serial sections, and segmentation of nuclei, blood vessels, and liver steatosis, with deep learning oriented segmentation methods [18].
3. We applied the spatial computing methods to opioid epidemic research, which leads to major public health findings.

## 5 Education / Outreach and Training

Summer REU programs were emphasized throughout the program with a total of about 5 each year. Arizona State found REUs were successful in bringing domain scientists into the data analytics arena. With some statistics out of 5 REU students: 1 continued

for a Ph.D. in the area at ASU, 2 continued undergrad projects in the research group at ASU, 1 took up a position at a national lab. IU's REU program was very successful with a focus on students from Tribal Colleges and HBCU's. In summer 2020 we perhaps had our most successful experience with as fully virtual REU with 4 students from Tribal Colleges. Another highlight was Stony Brook which involved multiple REU students conducting summer research, and two research manuscripts were produced through the REU work.

SPIDAL research gave rise to many new collaborations with new DoE links developed at Rutgers and Indiana. Work on big data benchmarking led to major links with MLPerf (an active organization with over 80 mainly industry members) and internationally with the SciML group at the UK's Rutherford Laboratory.

SPIDAL research was included in many courses developed by partners as exemplified by HPC-Cyberinfrastructure/Machine courses at IU [19].

## 6 Perspectives on SPIDAL Research

### 6.1 What worked (as well as or better than planned)

**Cyberinfrastructure Convergence** was an unanticipated important area at Rutgers and IU were both made good progress reported above. Rutgers found platform changes that reflected shifts in the application landscape. The platforms went from loosely coupled HPC and data-analytics (e.g, MD simulations, and trajectory analysis), to tightly coupled HPC - ML workflows [11], [20]. The RADICAL team managed these paradigmatic transitions by focusing on the needs of SPIDAL's driving applications, viz., biomolecular simulations, and high-resolution imagery. We developed conceptual abstractions and software systems to support diverse applications and brought them to the production scale. In the final 6 months of the project, the team developed new implementations to support the COVID campaign consisting of diverse but integrated workflows.

**Cyberinfrastructure Concepts and abstractions:** here we made progress in many areas including Ogres for classifying Big Data problems, Computation models in Harp, Big data system architectures in Cylon and Twister2 and Pilot Jobs and Data from Rutgers.

**Application Lessons:** At ASU, domain scientists working with CS specialists lead to improved performance in existing code and new parallel algorithms.

At Stony Brook, our work has been successfully applied to support two main research themes: big spatial data-driven public health research on the opioid epidemic, and computational digital pathology. This led to an NCI award on 3D digital pathology for

cancer research, and a working-in-progress proposal using machine learning and big spatial data for early prediction of opioid use disorder of young adults.
**At Kansas and IU, r**adioglaciology ML research benefited from the convergence of AI experts in areas of computer vision and HPC tools; SPIDAL provided an effective framework to pursue solutions in these spaces.

### 6.2 What didn't work (as well as expected)

**In Cyberinfrastructure,** we underestimated the Software Engineering challenges as well as documentation needs. In the latter area, we stopped using traditional websites but focussed on GitHub based web resources covering both software and documentation. More technically, we initially underestimated deep learning, and Python Notebooks but these are now a major focus. Also, the complexity of diverse platforms and the need to integrate with diverse ecosystems was not anticipated. SPIDAL unexpectedly needed multiple and diverse languages, starting with Java, but that early focus has been overshadowed by the growth and importance of Python
**Application lessons** come from Stony Brook. There is a major hurdle for biomedical researchers to use the software tools running on a distributed/cloud-based environment. GUI or Web-based portal is highly desired, and easy visualization and navigation of results are preferred. This has driven us to develop a Web portal to integrate big data computing in the backend. Further, accessing healthcare data is restricted due to privacy constraints. We have spent a major effort on getting institutional IRB approval and state-level approval to use electronic health records with location information. While now we can access a large statewide dataset, accessing data from our own hospital is still pending.

## 7 Challenges/Futures at the close of SPIDAL

### 7.1 Cyberinfrastructure MIDAS

Given system heterogeneity, both coarse and fine-grained task-level parallelism has become more dominant for analytics. SPIDAL has had mixed / some success leveraging task-level parallelism but just recently technology like Cylon [13] has made major progress in HPC for Python, Our streaming interest has grown with work at several sites including IU's work on real-time analysis of data from care races [21]. We have already noted our increased focus on MLforHPC.
Perhaps the greatest change will come from the still increasing importance of deep learning and Jupyter notebooks.

### 7.2 Applications and Communities

Stony Brook notes that the NIH Human BioMolecular Atlas Program (HuBMAP) and the NCI Human Tumor Atlas Network (HTAN) are initiatives to generate extreme-scale biomedical data at the cellular or subcellular resolution to create 3D atlases of the

human body. This creates a huge opportunity in expanding our spatial big data research and poses new challenges as well due to the high complexity of data generated from human tissues.

Kansas notes that the SPIDAL work led to an NSF Convergence Accelerator proposal submission involving partnerships that formed as a consequence of SPIDAL; there is now community awareness that ML techniques should be applied on a grander scale to mine decades of underutilized radioglaciology data.

UVa explored different kinds of computing models for network algorithms, including Hadoop, Spark, MPI, and HARP [22]. In general, graph algorithms are challenging to parallelize due to their highly heterogeneous communication patterns, and the best model depends on the problem. We found HARP and its variants worked very well for subgraph counting problems. Further work is needed for problems such as network scan statistics, which are more difficult optimization problems.

## 8 Selected References


[1] Geoffrey Fox, Madhav Marathe, Shantenu Jha, Judy Qiu, Fusheng Wang, “CIF21 DIBBs: Middleware and High Performance Analytics Libraries for Scalable Data Science Status Poster, February 2020.” [Online]. Available: http://dsc.soic.indiana.edu/presentations/DibbsNSF1443054-CSSITemplatePoster.pptx,. [Accessed: 06-Oct-2020]

[2] O. Beckstein, G. Fox, J. Qiu, D. Crandall, G. von Laszewski, J. Paden, S. Jha, F. Wang, M. Marathe, A. Vullikanti, and T. Cheatham, “Contributions to High-Performance Big Data Computing,” in *Future Trends of HPC in a Disruptive Scenario*, vol. 34, Grandinetti, L., Joubert, G.R., Michielsen, K., Mirtaheri, S.L., Taufer, M., Yokota, R., Ed. IOS, 2019 [Online]. Available: http://dsc.soic.indiana.edu/publications/SPIDALPaperSept2018.pdf

[3] G. Fox, D. Crandall, J. Qiu, G. Von Laszewski, S. Jha, J. Paden, O. Beckstein, T. Cheatham, M. Marathe, and F. Wang, “Datanet: CIF21 DIBBs: Middleware and High Performance Analytics Libraries for Scalable Data Science NSF14-43054 Progress Report. A 21 month Project Report,” Sep. 2016 [Online]. Available: http://dsc.soic.indiana.edu/publications/SPIDAL-DIBBSreport_July2016.pdf

[4] “HPC-ABDS Kaleidoscope of over 350 Apache Big Data Stack and HPC Technologies.” [Online]. Available: http://hpc-abds.org/kaleidoscope/. [Accessed: 01-Dec-2018]

[5] Geoffrey Fox, Judy Qiu, Shantenu Jha, Saliya Ekanayake, and Supun Kamburugamuve, “Big Data, Simulations and HPC Convergence,” in *Springer Lecture Notes in Computer Science LNCS 10044*, New Delhi, India, 2016 [Online]. Available: http://dsc.soic.indiana.edu/publications/HPCBigDataConvergence.pdf

[6] Wo L. Chang, Geoffrey Fox, NBD-PWG NIST Big Data Public Working Group, “NIST Big Data Interoperability Framework: Volume 3, Big Data Use Cases and General Requirements [Version 2],” NIST, Jun. 2018 [Online]. Available: https://www.nist.gov/publications/nist-big-data-interoperability-framework-volume-3-big-data-use-cases-and-general

[7] Geoffrey Fox, Shantenu Jha, and Lavanya Ramakrishnan, *Streaming and Steering Applications: Requirements and Infrastructure STREAM2015*. 2015 [Online]. Available: http://streamingsystems.org/stream2015.html

[8] G. Fox, S. Jha, and L. Ramakrishnan, *STREAM2016: Streaming Requirements, Experience, Applications and Middleware Workshop Workshop Final Report*. 2016 [Online]. Available: http://dx.doi.org/10.2172/1344785
[9] Geoffrey Fox, Shantenu Jha, “Learning Everywhere: A Taxonomy for the Integration of Machine Learning and Simulations,” in *IEEE eScience 2019 Conference*, San Diego, California [Online]. Available: https://arxiv.org/abs/1909.13340
[10] SPIDAL Project, “HPCAnalytics Big Data Resource.” [Online]. Available: https://hpcanalytics.org/. [Accessed: 01-Jan-2020]
[11] Geoffrey Fox, Shantenu Jha, “Understanding ML driven HPC: Applications and Infrastructure,” in *IEEE eScience 2019 Conference*, San Diego, California [Online]. Available: https://escience2019.sdsc.edu/
[12] “Harp-DAAL Framework.” [Online]. Available: https://dsc-spidal.github.io/harp/docs/harpdaal/harpdaal/. [Accessed: 24-Nov-2019]
[13] C. Widanage, N. Perera, V. Abeykoon, S. Kamburugamuve, T. A. Kanewala, H. Maithree, P. Wickramasinghe, A. Uyar, G. Gunduz, and G. Fox, “High Performance Data Engineering Everywhere,” *arXiv [cs.DC]*, 19-Jul-2020 [Online]. Available: http://arxiv.org/abs/2007.09589
[14] “Twister2 Tutorial and Project Home Page.” [Online]. Available: https://twister2.org/. [Accessed: 09-Oct-2020]
[15] Y. Wang, M. Xu, J. Paden, L. Koenig, G. Fox, and D. Crandall, “Deep Tiered Image Segmentation for detecting Internal Ice Layers in Radar Imagery,” *arXiv [cs.CV]*, 08-Oct-2020 [Online]. Available: http://arxiv.org/abs/2010.03712
[16] L. Wang, J. Chen, and M. Marathe, “TDEFSI: Theory-guided Deep Learning-based Epidemic Forecasting with Synthetic Information,” *ACM Trans. Spatial Algorithms Syst.*, vol. 6, no. 3, pp. 1–39, Apr. 2020 [Online]. Available: https://doi.org/10.1145/3380971
[17] Yanhui Liang, Hoang Vo, Jun Kong, Fusheng Wang, “iSPEED: a Scalable and Distributed In-Memory Based Spatial Query System for Large and Structurally Complex 3D Data. A Demo Paper,” in *Proceedings of the 44th International Conference on Very Large Data Bases (VLDB 2018) Volume 11 Issue 12,* Rio de Janeiro, Brazil., 2018, pp. 2078–2081 [Online]. Available: http://www.vldb.org/pvldb/vol11/p2078-vo.pdf
[18] M. Roy, F. Wang, H. Vo, D. Teng, G. Teodoro, A. B. Farris, E. Castillo-Leon, M. B. Vos, and J. Kong, “Deep-learning-based accurate hepatic steatosis quantification for histological assessment of liver biopsies,” *Lab. Invest.*, vol. 100, no. 10, pp. 1367–1383, Oct. 2020 [Online]. Available: http://dx.doi.org/10.1038/s41374-020-0463-y
[19] Digital Science Center, “Cybertraining training resource for Cyberinfrastructure-Machine Learning Interface.” [Online]. Available: https://cybertraining-dsc.github.io/about/. [Accessed: 1 October, 2020]
[20] H. Lee, H. Ma, M. Turilli, D. Bhowmik, S. Jha, and A. Ramanathan, “DeepDriveMD: Deep-Learning Driven Adaptive Molecular Simulations for Protein Folding,” in *Deep Learning (DL) on Supercomputers workshop (In cooperation with TCHPC and held in conjunction with SC19)*, 2019 [Online]. Available: http://arxiv.org/abs/1909.07817
[21] Bo Peng, Jiayu Li, Selahattin Akkas, Fugang Wang, Takuya Araki, Ohno Yoshiyuki, Judy Qiu, “Rank Position Forecasting in Car Racing,” Jul. 2020 [Online]. Available: http://dsc.soic.indiana.edu/publications/RankPrediction (1).pdf
[22] L. Chen, J. Li, C. Sahinalp, M. Marathe, A. Vullikanti, A. Nikolaev, E. Smirnov, R. Israfilov, and J. Qiu, “Subgraph2vec: Highly-vectorized tree-like subgraph counting,” in *2019 IEEE International Conference on Big Data,* Los Angeles [Online]. Available: http://dsc.soic.indiana.edu/publications/Bigdata_Subgraph2Vec.pdf