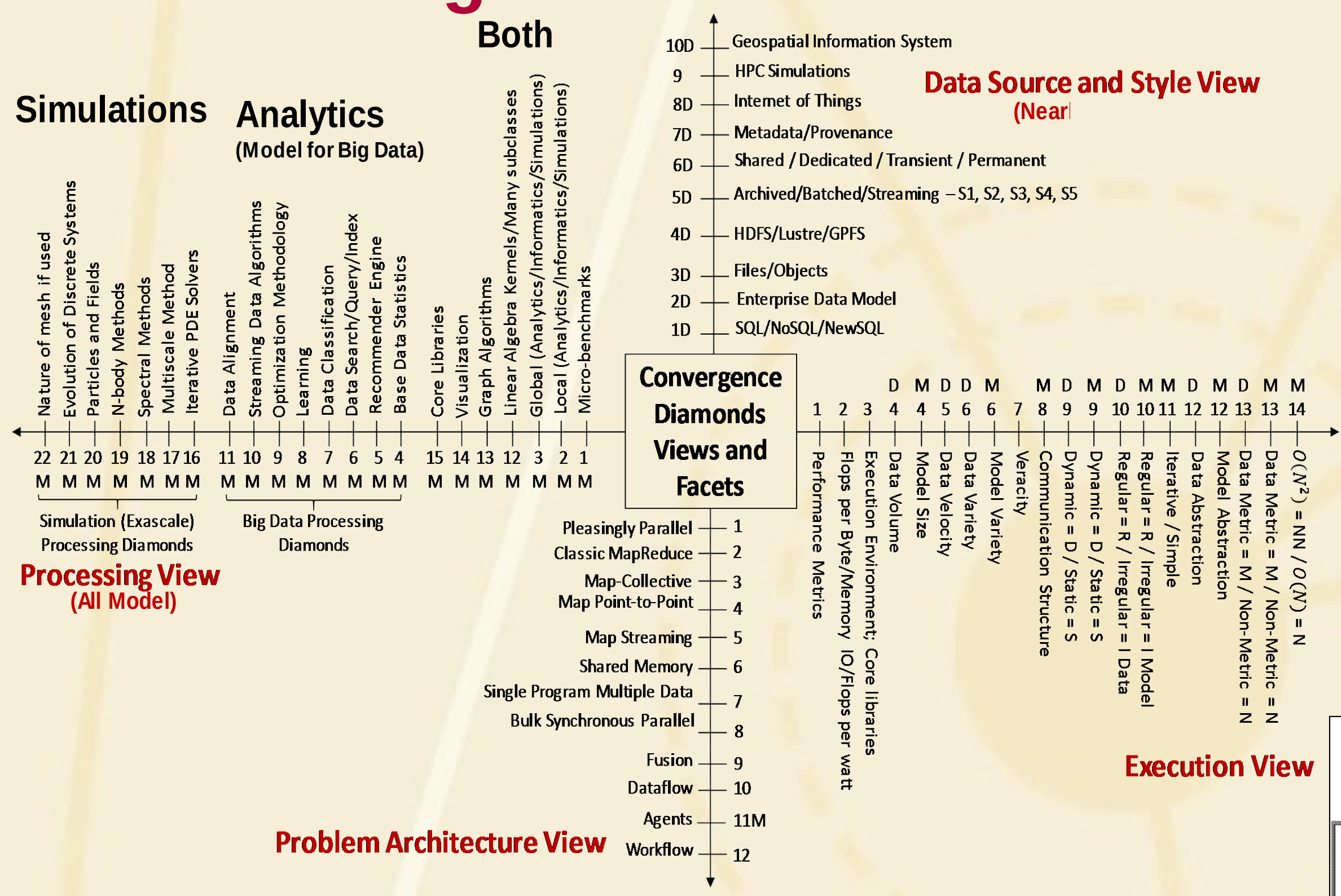
NSF 1443054: CIF21 DIBBs: Middleware and High Performance Analytics Libraries for Scalable Data Science PI: Geoffrey C. Fox

Components of NSF 144305

- Design and build **SPIDAL** – a Scalable Parallel Interoperable Data Analytics Library
  - Domain specific libraries – mainly from project
  - Core Machine Learning Libraries
- High Performance for Java, Libraries and MIDAS
- NIST **Big Data Application Analysis**: Features of data intensive Applications deriving 50 Ogres and 64 Convergence Diamonds
- HPC-ABDS: Cloud-HPC Interoperable software with performance of HPC and rich functionality of commodity Apache Stack
- Implementations: HPC and Clouds with DevOps
- Applications: Biomolecular Simulations, Network Science, Epidemiology, Computer Vision, Spatial Geographical Information Systems, Remote Sensing for Polar Science and Pathology Informatics.

Building Blocks of Proposal

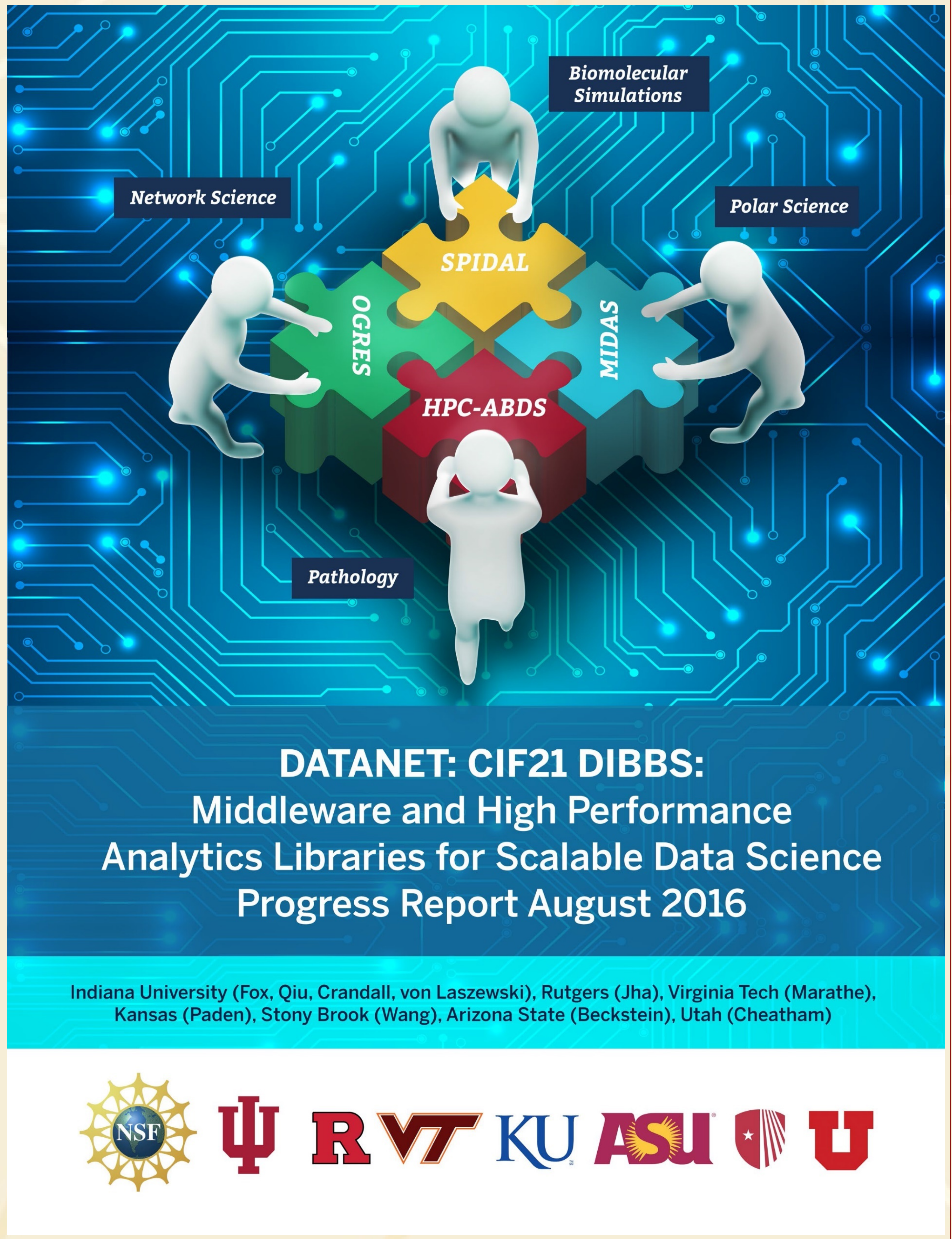
64 Features in 4 views for HPC-Big Data Converged Classification



HPC-ABDS Apache Big Data Stack

Kaleidoscope of (Apache) Big Data Stack (ABDS) and HPC Technologies

Cross-Cutting Functions	17) **Workflow-Orchestration:** ODE, ActiveBPEL, Airavata, Pegasus, Kepler, Swift, Taverna, Triana, Trident, BioKepler, Galaxy, IPython, Dryad, Naiad, Oozie, Tez, Google FlumeJava, Crunch, Cascading, Scalding, e-Science Central, Azure Data Factory, Google Cloud Dataflow, NiFi (NSA), Jitterbit, Talend, Pentaho, Apatar, Docker Compose, KeystoneML
1) **Message and Data Protocols:** Avro, Thrift, Protobuf	16) **Application and Analytics:** Mahout , MLlib , MLbase, DataFu, R, pbdR, Bioconductor, ImageJ, OpenCV, Scalapack, PetSc, PLASMA MAGMA, Azure Machine Learning, Google Prediction API & Translation API, mlpy, scikit-learn, PyBrain, CompLearn, DAAL(Intel), Caffe, Torch, Theano, DL4j, H2O, IBM Watson, Oracle PGX, GraphLab, GraphX, IBM System G, GraphBuilder(Intel), TinkerPop, Parasol, Dream:Lab, Google Fusion Tables, CINET, NWB, Elasticsearch, Kibana, Logstash, Graylog, Splunk, Tableau, D3.js, three.js, Potree, DC.js, TensorFlow, CNTK
2) **Distributed Coordination**: Google Chubby, Zookeeper, Giraffe, JGroups	15B) **Application Hosting Frameworks:** Google App Engine, AppScale, Red Hat OpenShift, Heroku, Aerobatic, AWS Elastic Beanstalk, Azure, Cloud Foundry, Pivotal, IBM BlueMix, Ninefold, Jelastic, Stackato, appfog, CloudBees, Engine Yard, CloudControl, dotCloud, Dokku, OSGi, HUBzero, OODT, Agave, Atmosphere 15A) **High level Programming:** Kite, Hive, HCatalog, Tajo, Shark, Phoenix, Impala, MRQL, SAP HANA, HadoopDB, PolyBase, Pivotal HD/Hawq, Presto, Google Dremel, Google BigQuery, Amazon Redshift, Drill, Kyoto Cabinet, Pig, Sawzall, Google Cloud DataFlow, Summingbird
	14B) **Streams:** Storm, S4, Samza, Granules, Neptune, Google MillWheel, Amazon Kinesis, LinkedIn, Twitter Heron, Databus, Facebook Puma/Ptail/Scribe/ODS, Azure Stream Analytics, Floe, Spark Streaming, Flink Streaming, DataTurbine 14A) **Basic Programming model and runtime**, **SPMD, MapReduce:** Hadoop, Spark, Twister, MR-MPI, Stratosphere (Apache Flink), Reef, Disco, Hama, Giraph, Pregel, Pegasus, Ligra, GraphChi, Galois, Medusa-GPU, MapGraph, Totem
3) **Security & Privacy:** InCommon, Eduroam OpenStack Keystone, LDAP, Sentry, Sqrrl, OpenID, SAML OAuth	13) **Inter process communication Collectives, point-to-point, publish-subscribe:** MPI, HPX-5, Argo BEAST HPX-5 BEAST PULSAR, Harp, Netty, ZeroMQ, ActiveMQ, RabbitMQ, NaradaBrokering, QPid, Kafka, Kestrel, JMS, AMQP, Stomp, MQTT, Marionette Collective, **Public Cloud:** Amazon SNS, Lambda, Google Pub Sub, Azure Queues, Event Hubs
	12) **In-memory databases/caches:** Gora (general object from NoSQL), Memcached, Redis, LMDB (key value), Hazelcast, Ehcache, Infinispan, VoltDB, H-Store
	12) **Object-relational mapping:** Hibernate, OpenJPA, EclipseLink, DataNucleus, ODBC/JDBC
	12) **Extraction Tools:** UIMA, Tika
	11C) **SQL(NewSQL):** Oracle, DB2, SQL Server, SQLite, MySQL, PostgreSQL, CUBRID, Galera Cluster, SciDB, Rasdaman, Apache Derby, Pivotal Greenplum, Google Cloud SQL, Azure SQL, Amazon RDS, Google F1, IBM dashDB, N1QL, BlinkDB, Spark SQL
4) **Monitoring:** Ambari, Ganglia, Nagios, Inca	11B) **NoSQL:** Lucene, Solr, Solandra, Voldemort, Riak, ZHT, Berkeley DB, Kyoto/Tokyo Cabinet, Tycoon, Tyrant, MongoDB, Espresso, CouchDB, Couchbase, IBM Cloudant, Pivotal Gemfire, HBase, Google Bigtable, LevelDB, Megastore and Spanner, Accumulo, Cassandra, RYA, Sqrrl, Neo4J, graphdb, Yarcdata, AllegroGraph, Blazegraph, Facebook Tao, Titan:db, Jena, Sesame **Public Cloud:** Azure Table, Amazon Dynamo, Google DataStore
	11A) **File management:** iRODS, NetCDF, CDF, HDF, OPeNDAP, FITS, RCFile, ORC, Parquet
21 layers Over 350 Software Packages **January 29 2016**	10) **Data Transport:** BitTorrent, HTTP, FTP, SSH, Globus Online (GridFTP), Flume, Sqoop, Pivotal GPLOAD/GPFDIST
	9) **Cluster Resource Management**: Mesos, Yarn, Helix, Llama, Google Omega, Facebook Corona, Celery, HTCondor, SGE, OpenPBS, Moab, Slurm, Torque, Globus Tools, Pilot Jobs
	8) **File systems:** HDFS, Swift, Haystack, f4, Cinder, Ceph, FUSE, Gluster, Lustre, GPFS, GFFS **Public Cloud:** Amazon S3, Azure Blob, Google Cloud Storage
	7) **Interoperability:** Libvirt, Libcloud, JClouds, TOSCA, OCCI, CDMI, Whirr, Saga, Genesis
	6) **DevOps:** Docker (Machine, Swarm), Puppet, Chef, Ansible, SaltStack, Boto, Cobbler, Xcat, Razor, CloudMesh, Juju, Foreman, OpenStack Heat, Sahara, Rocks, Cisco Intelligent Automation for Cloud, Ubuntu MaaS, Facebook Tupperware, AWS OpsWorks, OpenStack Ironic, Google Kubernetes, Buildstep, Gitreceive, OpenTOSCA, Winery, CloudML, Blueprints, Terraform, DevOpSlang, Any2Api
	5) **IaaS Management from HPC to hypervisors:** Xen, KVM, QEMU, Hyper-V, VirtualBox, OpenVZ, LXC, Linux-Vserver, OpenStack, OpenNebula, Eucalyptus, Nimbus, CloudStack, CoreOS, rkt, VMware ESXi, vSphere and vCloud, Amazon, Azure, Google and other public Clouds **Networking:** Google Cloud DNS, Amazon Route 53



DATANET: CIF21 DIBBS:
Middleware and High Performance Analytics Libraries for Scalable Data Science
Progress Report August 2016

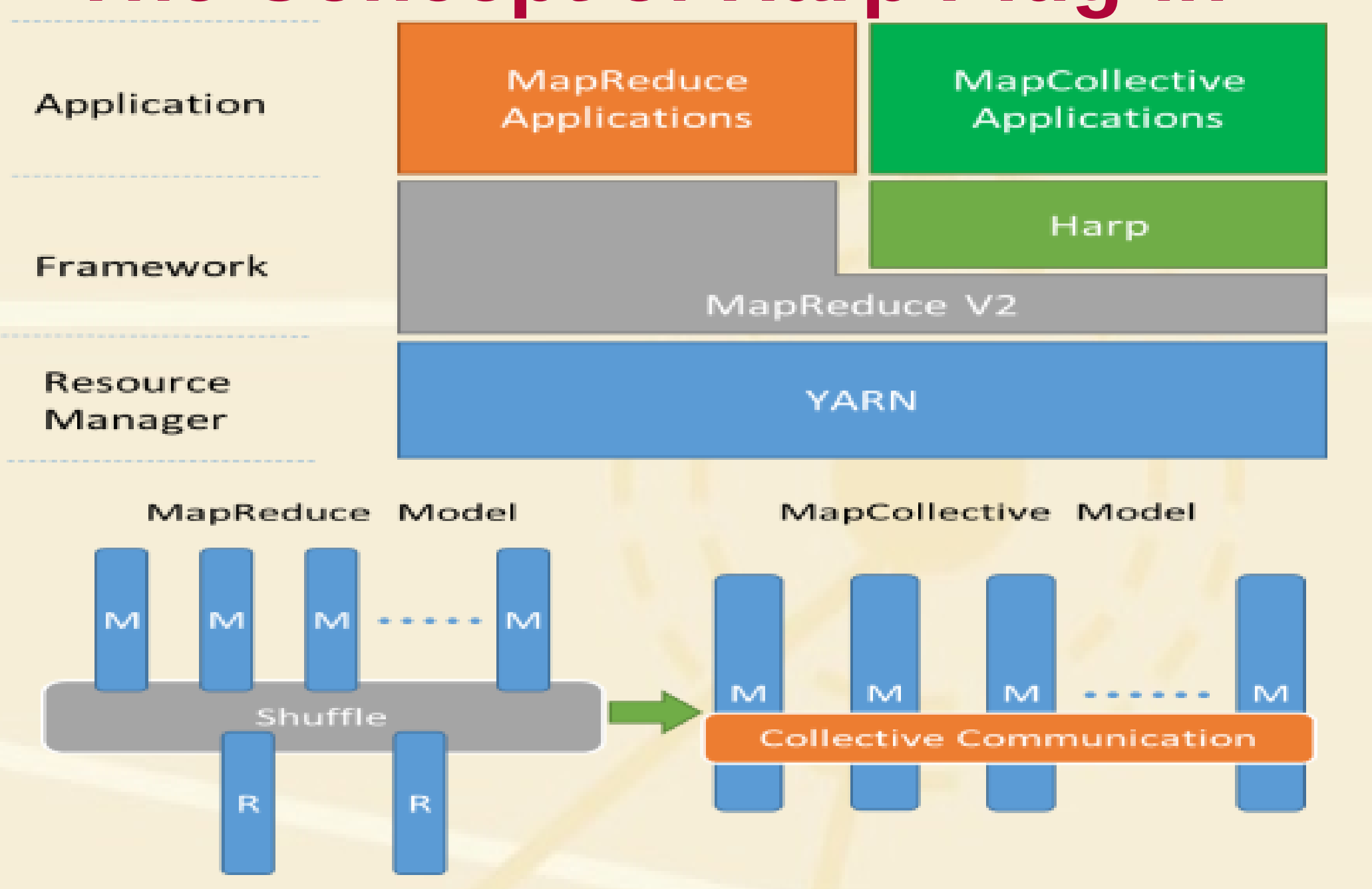
Indiana University (Fox, Qiu, Crandall, von Laszewski), Rutgers (Jha), Virginia Tech (Marathe), Kansas (Paden), Stony Brook (Wang), Arizona State (Beckstein), Utah (Cheatham)

MIDAS and SPIDAL Java High Performance Middleware and Language

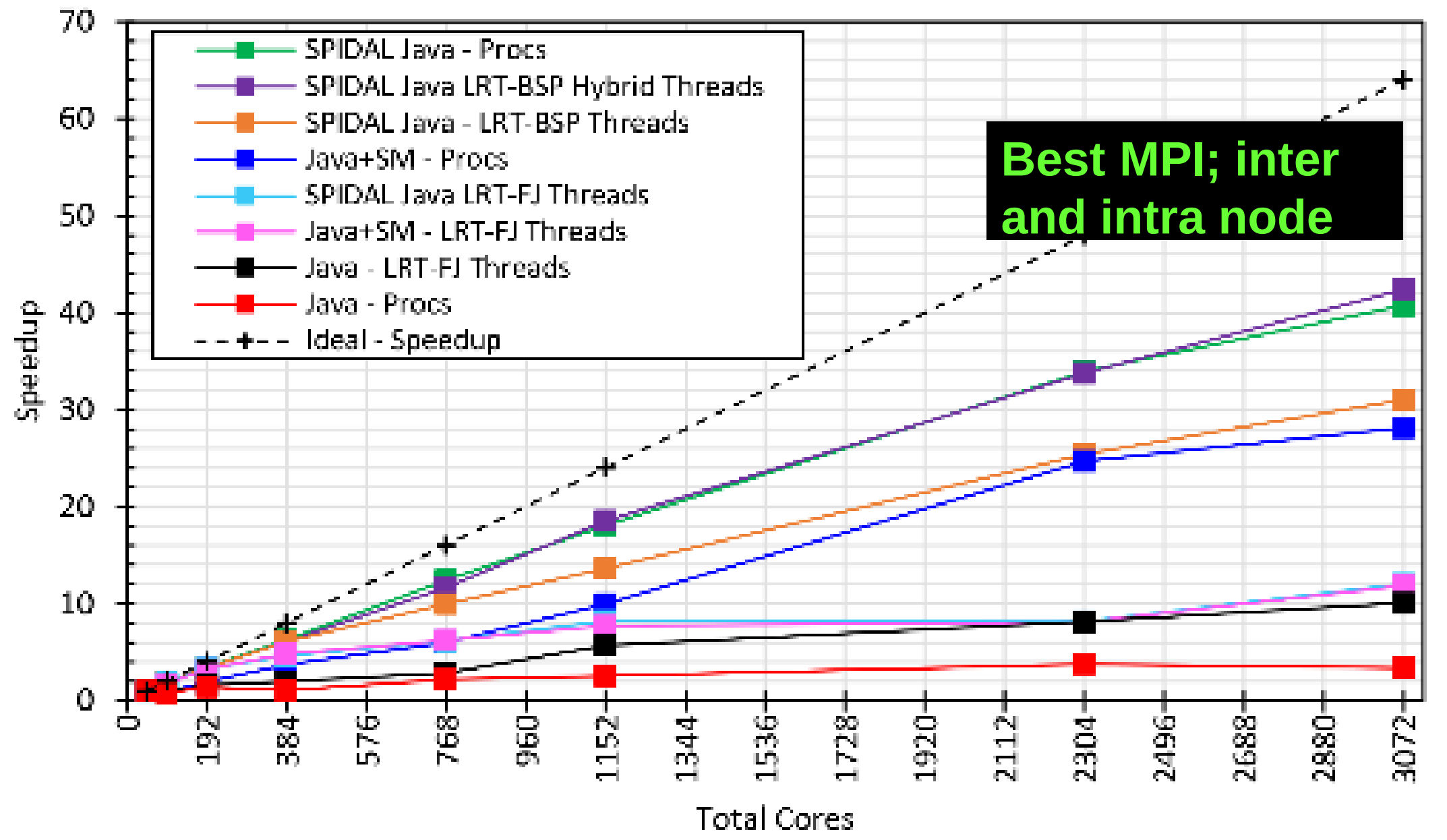
http://dsc.soic.indiana.edu/publications/SPIDAL-DIBBSreport_July2016.pdf

http://spidal.org

The Concept of Harp Plug-in



DA-MDS speedup for 200K with different optimization techniques

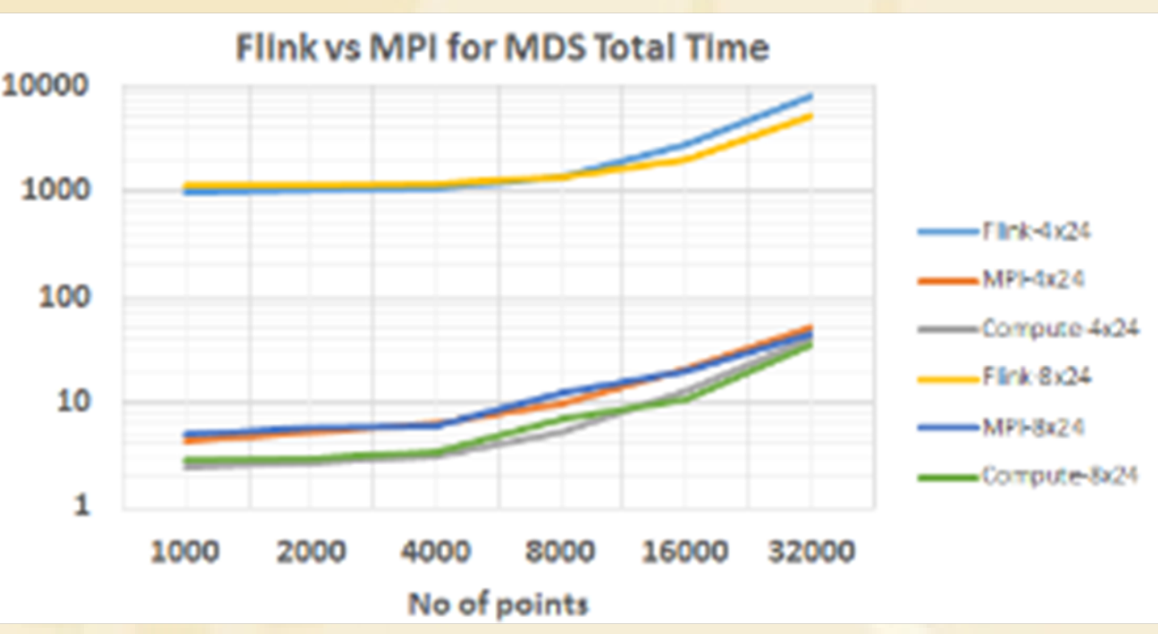


Multidimensional Scaling with Flink

- Projects NxN distance matrix to NxD matrix where N is the number of points and D is the target dimension.
- Row wise partitioning of NxN matrix for parallelization
- Our algorithm uses deterministic annealing and has three nested loops called Temperature, Stress and CG (Conjugate Gradient) in that order.

Flink vs MPI DA-MDS Performance

Total time of MPI Java and Flink MDS implementations for 96 and 192 parallel tasks.



Test with 5 Temperature Loops, 2 Stress Loops and 16 CG Loops.

Applications in action

MIDAS and Biomolecular Simulations

- **Previous Work (Middleware):** We have created a hybrid execution framework by coupling the Pilot-abstraction with Apache Hadoop and Spark. This enhances the ability of applications to utilizing HPC resources in conjunction with Hadoop frameworks concurrently or sequentially as needed.
- **Current Work: (Applications)** [i] We run Molecular Dynamics trajectory analysis using the hybrid execution framework, on a set of heterogenous of resources using task level parallel implementation and MapReduce programming through Spark. [ii] We are investigating the characteristics of these type of applications and the performance trade-off. Initial results show that these types of applications can greatly benefit from such an approach.
- **Future Work (Streaming):** Large volumes of streaming data that are real time or close to realtime need to be processed or saved for later analysis and thus need high performance processing capabilities. We are integrating Kafka to handle messages, Spark for processing the data to RADICAL-Pilot to handle the deployment and the job submissions on HPC systems (Pilot-Streaming) and will employ these for HPC simulations and analysis.

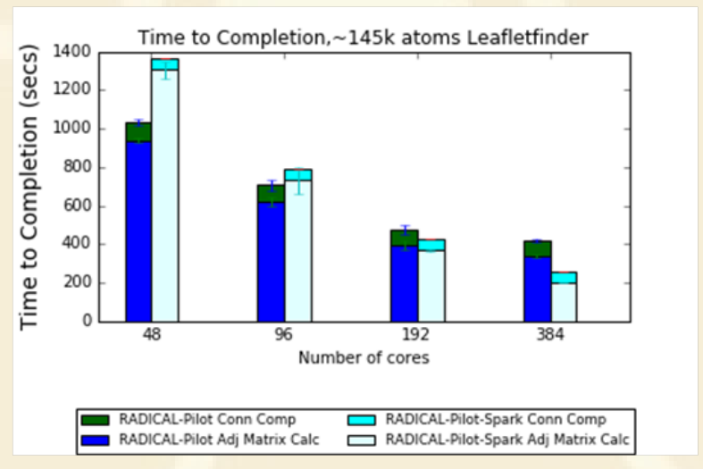
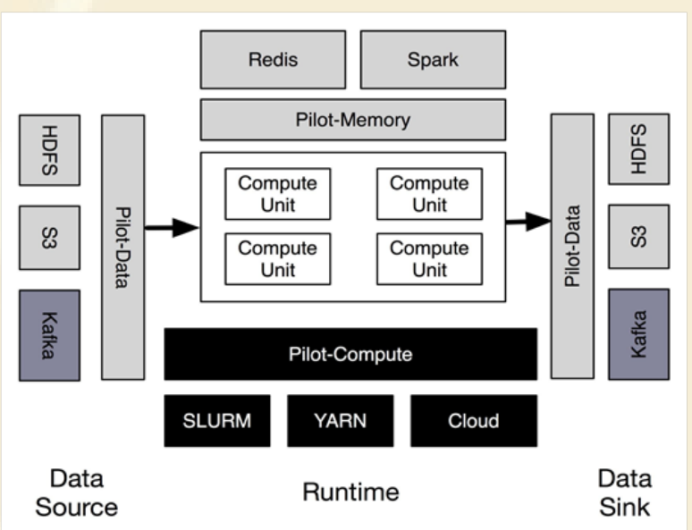


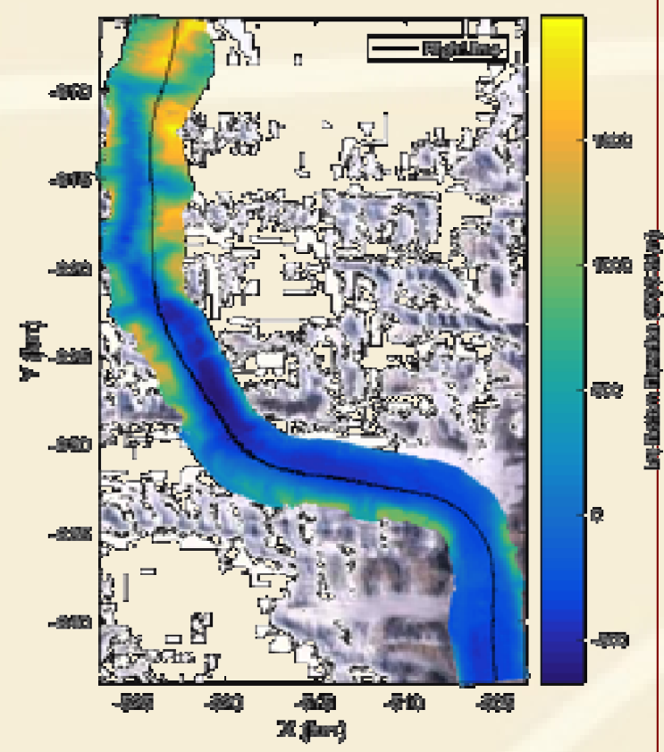
Figure Caption: [U] Data from MD Trajectory Analysis using MIDAS on SPARK for long trajectory for large system (>100000 atoms) on XSEDE. [L] Extending MIDAS to integrate Streaming into HPC simulation and analysis workflows.



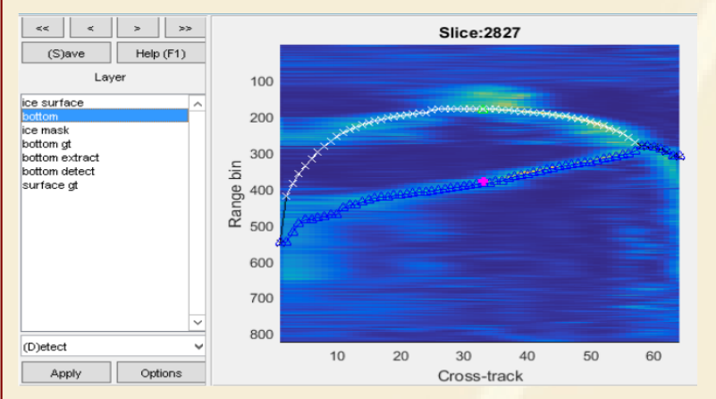
Polar Remote Sensing Algorithms

Many problems can be posed as mathematical optimization tasks, where the goal is to find the values for a set of variables that minimize a given objective or energy function of those variables. In applications, these tasks often arise in the context of fitting a model to data. For instance, nearly all machine learning algorithms are simply optimizing a set of model parameters by minimizing an objective function that measures the error of the model on a set of labeled training examples. Meanwhile data mining algorithms like K-means and LDA are similarly fitting model parameters to data to minimize some residual error. Many of the imaging applications in Section 5, for example, are simply finding a "simple" model to explain complicated image data, e.g. a model that compactly describes a noisy radar echogram in terms of ice layer structure, or a segmentation that compactly describes a high-resolution CT scan. This section describes our approach to many classes of optimization.

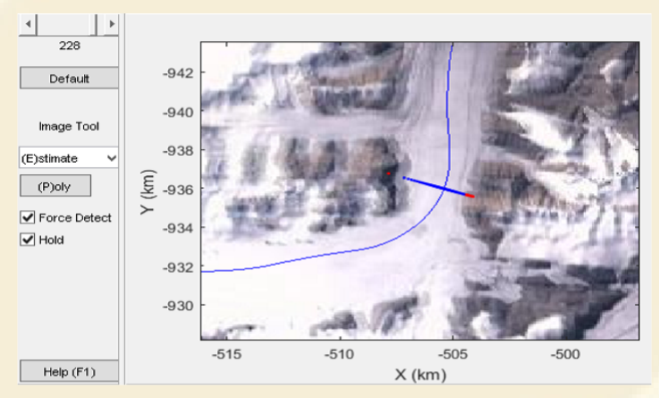
We have successfully applied SPIDAL optimization algorithms to 3D radar data collected by NASA Operation IceBridge over the Canadian Arctic Archipelago ice caps. The basal topography underlying these ice caps is unknown for many of the glaciers that drain them. The high volume of data produced by 3D imaging renders manual tracking of the ice bottom impractical. To solve this problem, we used SPIDAL algorithms to extract ice bottom surfaces using a probabilistic graphical model. We first estimate layer boundaries to generate a seed surface, and then incorporate additional evidence, such as ice masks, surface elevation models, and feedback from users, to refine the surface in a discrete energy minimization formulation. The sequence of figures below shows the user interface for examining the optimization algorithm's inputs and outputs. A sample result of an ice bottom digital elevation model is shown to the right, overlaid on a Landsat-7 image.



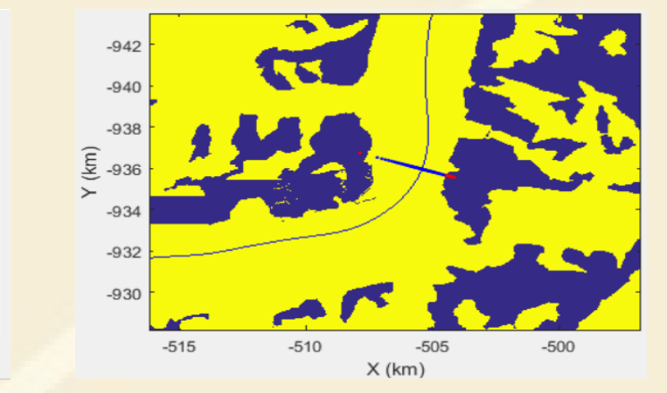
Ice bottom digital elevation model extracted using optimization algorithm on 3D image.



GUI to browse ice surface and bottom layer results, enter ground truth, and rerun optimization.



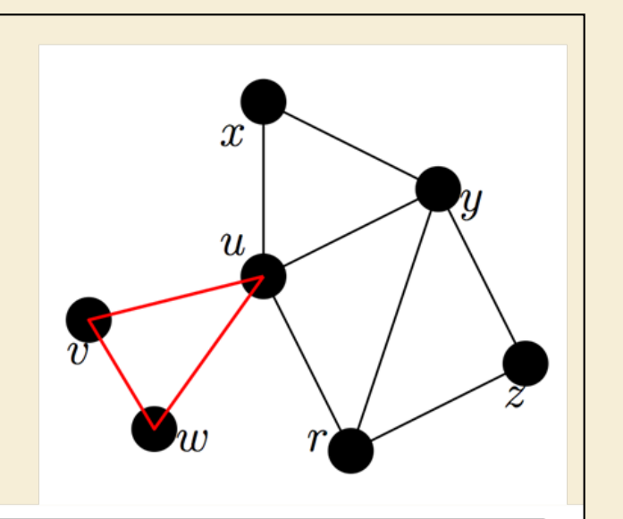
Map view showing projection of 3D image slice on a geographic map.

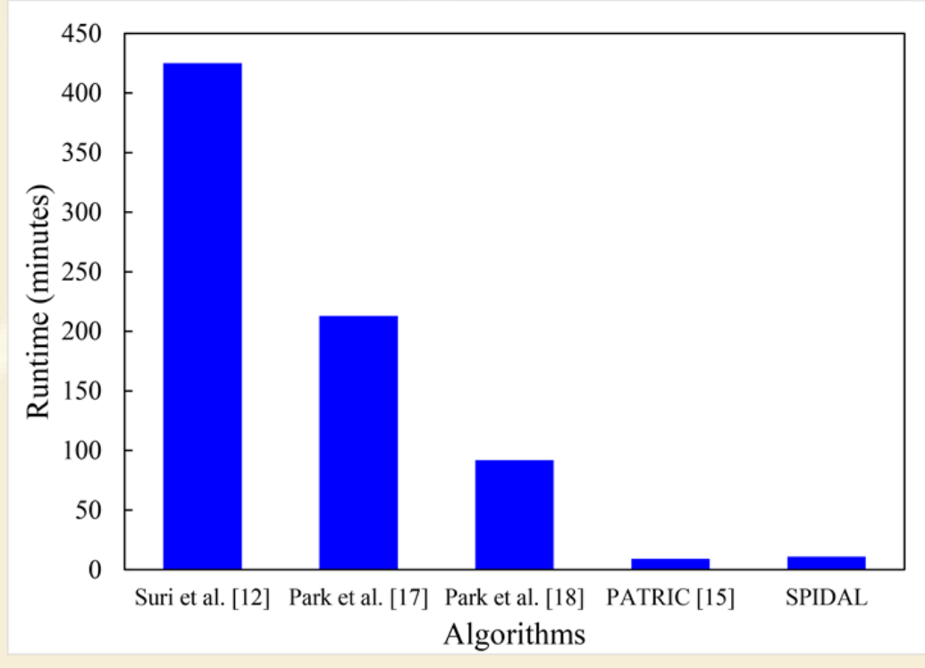


Ice mask (rock/ vs ice) that informs algorithm where layers should merge.

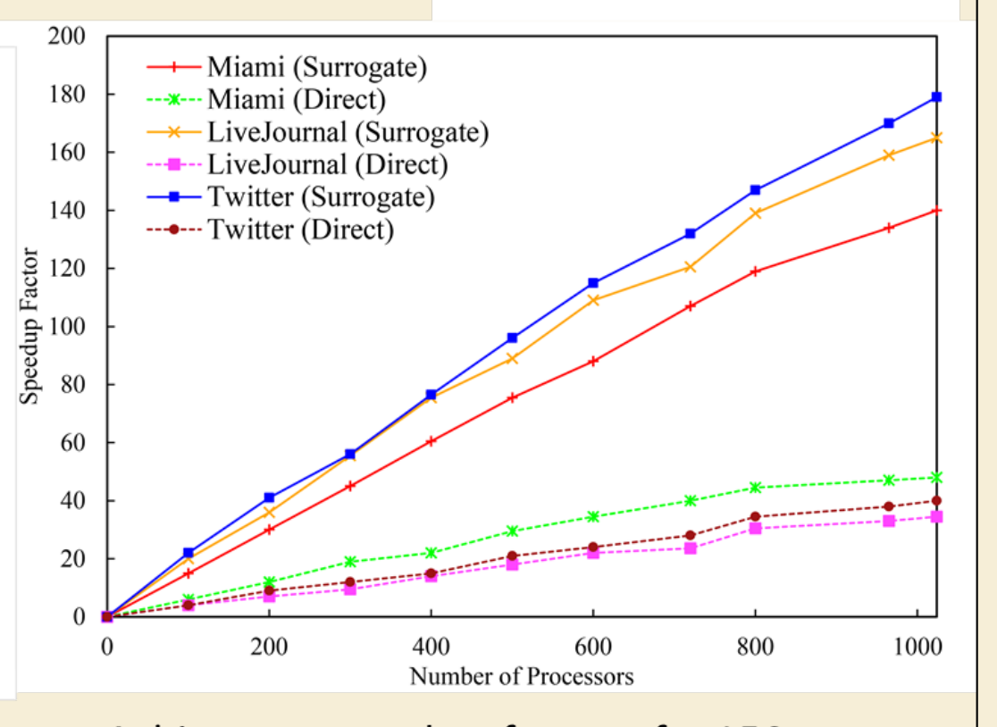
Counting Triangles in Massive Networks

Many applications in data mining, network analysis, social science, and database systems





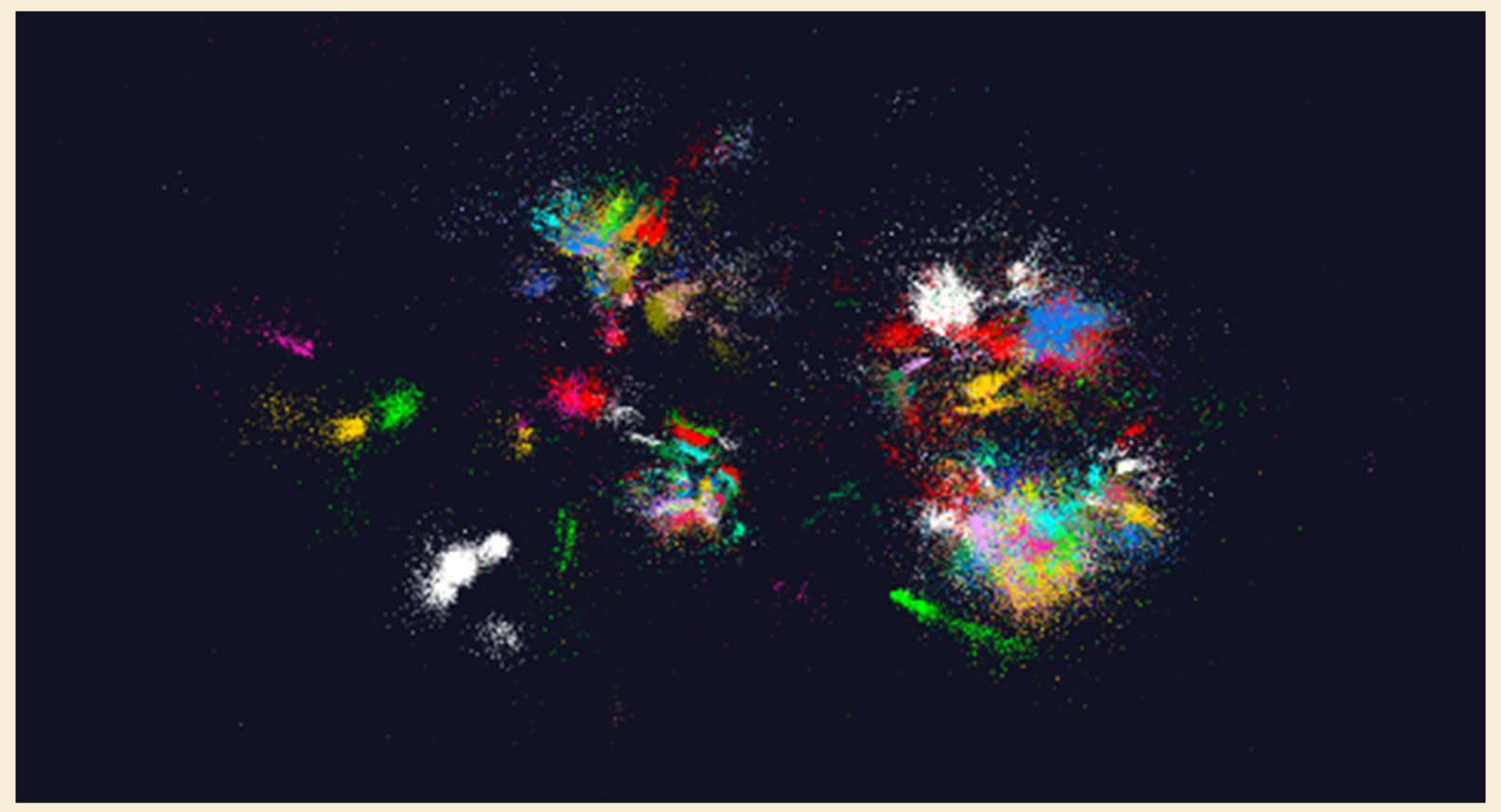
Run time comparison with other state-of-the-art algorithms



- Achieves a speedup factor of ≈ 150 with 1024 processors.
- Almost linear speedup up to a large number of processors

WebPlotViz – Browser Visualization of High Dimensional Data

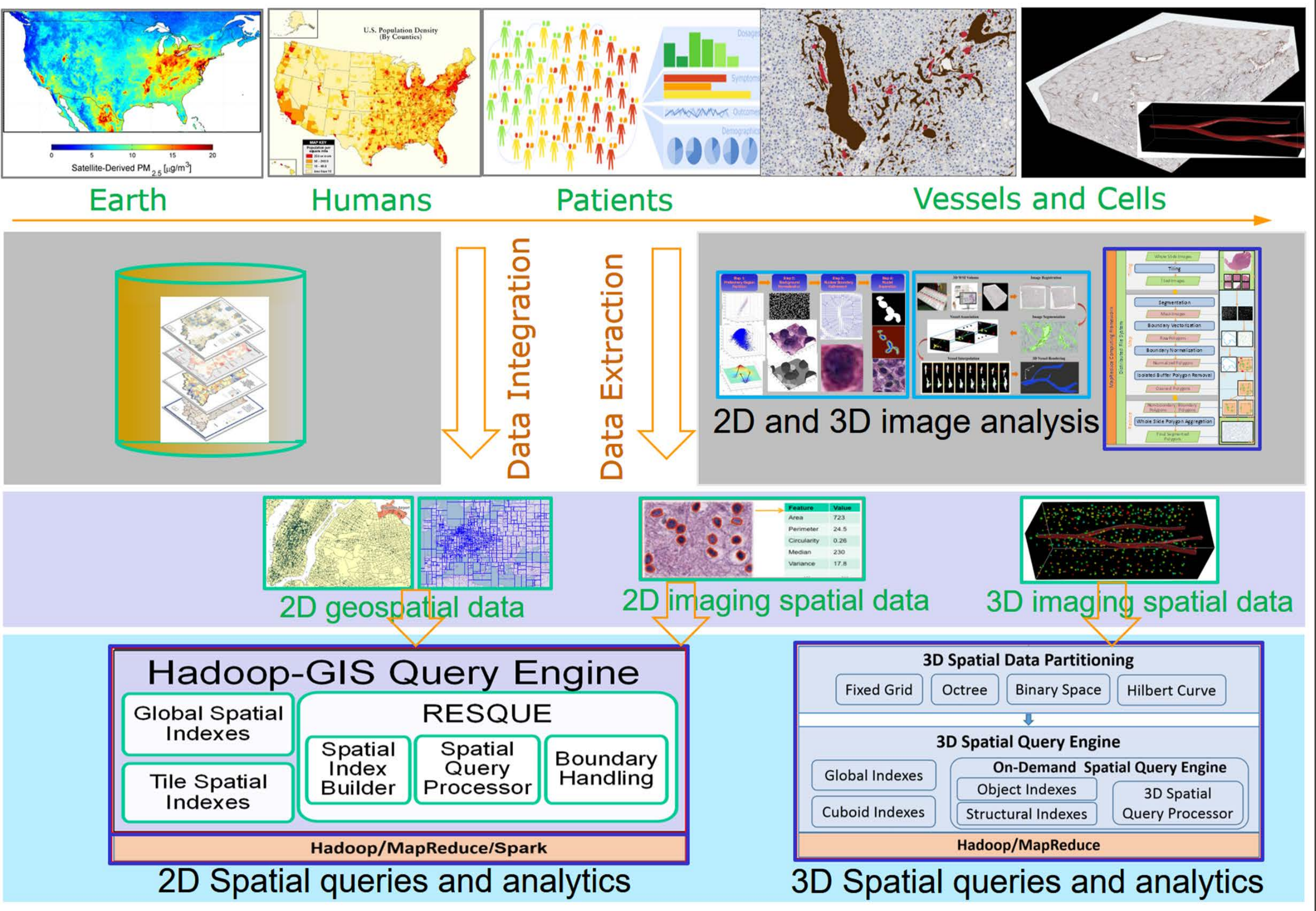
WebPlotViz is a 2D/3D data point browser that can visualize very large volumes of 2D or 3D data, as points in a virtual space and explore them interactively. WebPlotViz also includes support for Time Series Data plots



446K sequences and ~100 clusters visualized in WebPlotViz

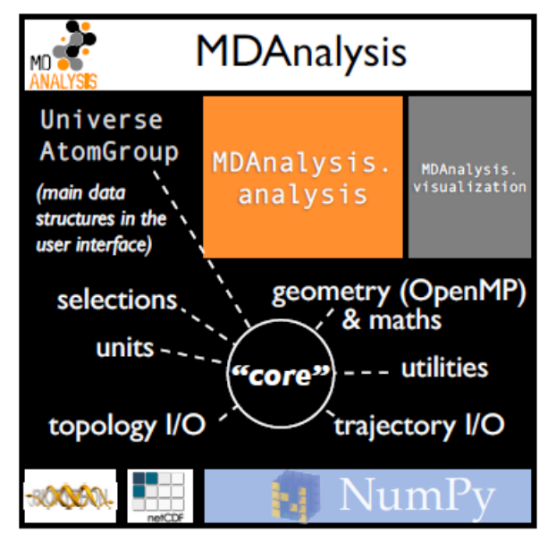
Applications in action with MIDAS/SPIDAL

Multi-Scale Imaging and Spatial Data Analytics



Biomolecular Simulations: Parallel analysis in the *MDAnalysis* Library: Benchmark of Trajectory File Formats

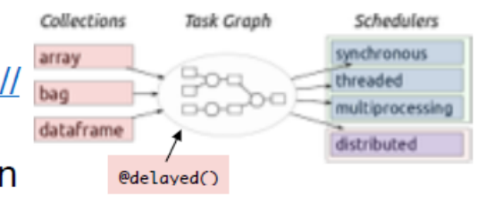
Mahzad Khoshlessan and Oliver Beckstein (Arizona State University)

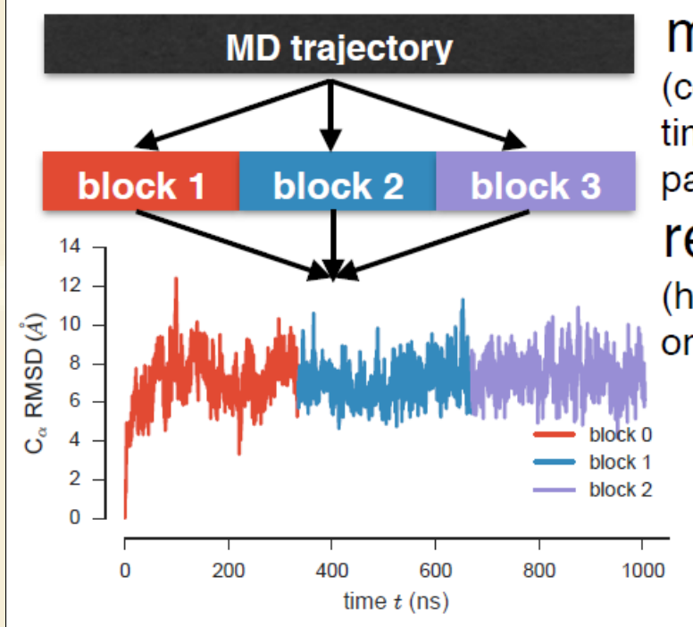


MDAnalysis http://mdanalysis.org is a widely used Python library to analyse molecular dynamics (MD) simulations of biomolecules, polymers, and materials (Gowers 2016; Michaud-Agrawal 2011).

Map-reduce parallelisation scheme

- implemented with *dask* (task-graph engine, http://dask.pydata.org/)
- test: calculation of protein RMSD after optimum structural superposition



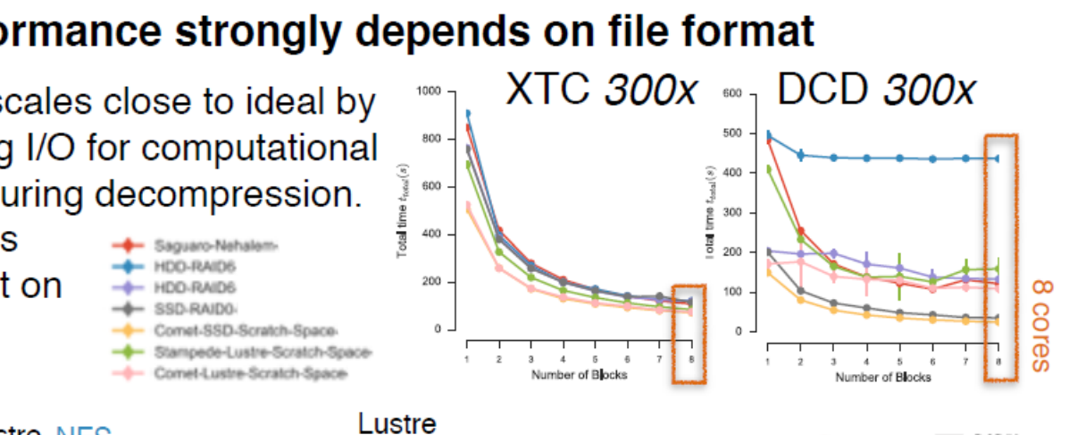


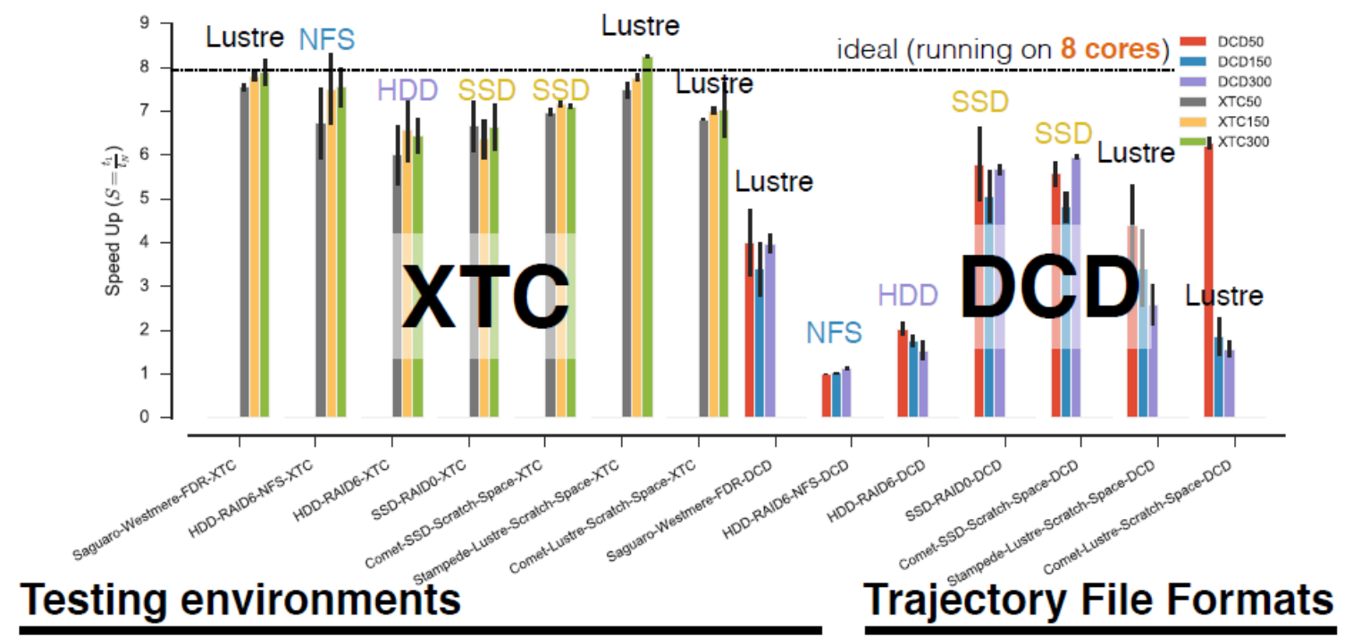
map (compute RMSD per timestep over blocks in parallel)

reduce (here: simple time-ordered join)

Performance strongly depends on file format

XTC scales close to ideal by trading I/O for computational cost during decompression. DCD is fastest on SSD.





Testing environments

resource	storage	interface	*file system*
ASU RC saguaro		FDR IB	Lustre
local	HDD	Gigabit	NFS
local	HDD	SATA3	xfs
local	SSD	SATA3	xfs
SDSC Comet		200 Gbps	Lustre
SDSC Comet	SSD		zfs
TACC Stampede		FDR IB	Lustre

Local and HPC resources with spinning disk HDD (locally attached/NFS/Lustre) or locally attached SSD.

Trajectory File Formats

trajectory	time steps	*DCD*	*XTC*
50x	209,350	7.9G	2.5G
150x	628,050	24G	7.5G
300x	1,256,100	47G	15G

Test trajectories with 3341 atoms per frame. **File formats**: *DCD:* binary float32/10^{-6} Å (CHARMM, NAMD). *XTC*: lossy compressed int16…int32/10^{-2} Å (Gromacs)

References

R. J. Gowers, M. Linke, J. Barnoud, T. J. E. Reddy, M. N. Melo, S. L. Seyler, D. L. Dotson, J. Domański, S. Buchoux, I. M. Kenney, and O. Beckstein. MDAnalysis: A Python package for the rapid analysis of molecular dynamics simulations. In S. Benthall and S. Rostrup, editors, *Proceedings of the 15th Python in Science Conference*, pages 102 – 109, Austin, TX, 2016. SciPy.

N. Michaud-Agrawal, E. J. Denning, T. B. Woolf, and O. Beckstein. MDAnalysis: A toolkit for the analysis of molecular dynamics simulations. *J Comp Chem*, 32:2319–2327, 2011.