Big Data Benchmarking: Applications and Systems

Geoffrey Fox, December 10, 2018

2018 International Symposium on Benchmarking, Measuring and Optimizing (Bench' 18) at IEEE Big Data 2018
Dec 10 - Dec 11, 2018 @ Seattle, WA, USA

Digital Science Center
Indiana University
gcf@indiana.edu, http://www.dsc.soic.indiana.edu/, http://spidal.org/

Big Data and Extreme-scale Computing http://www.exascale.org/bdec/

- **BDEC Pathways to Convergence Report** http://www.exascale.org/bdec/sites/www.exascale.org.bdec/files/whitepapers/bdec2017pathways.pdf
- New series BDEC2 "Common Digital Continuum Platform for Big Data and Extreme Scale Computing" with first meeting November 28-30, 2018 Bloomington Indiana USA (focus on applications).
  - Working groups on platform (technology), applications, community building
  - BigDataBench presented a white paper
- Next meetings: February 19-21 Kobe, Japan (focus on platform) followed by two in Europe, one in USA and one in China.

Benchmarks should mimic Use Cases?

Need to collect use cases?

Can classify use cases and benchmarks along several different dimensions

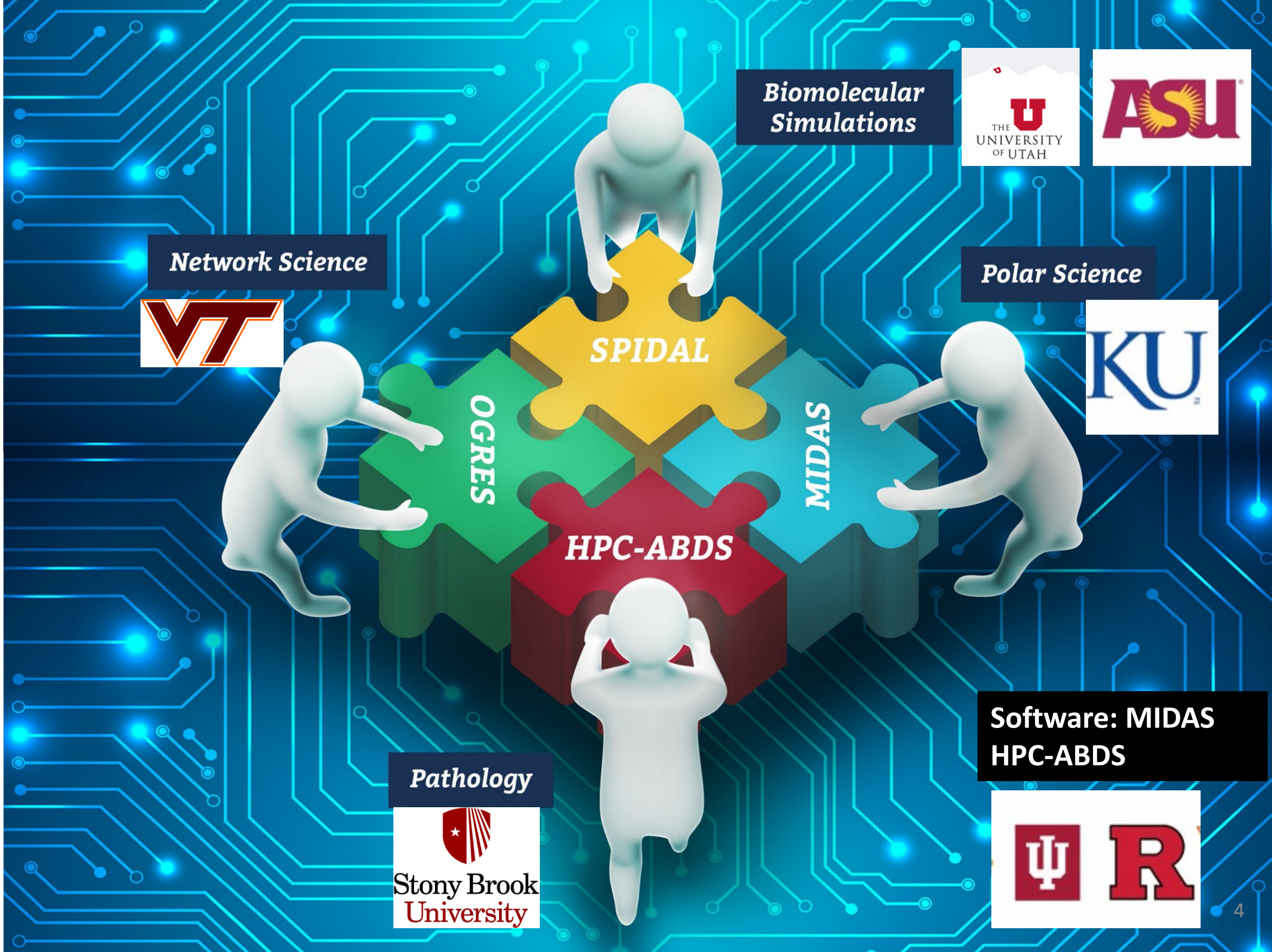
NSF 1443054: CIF21 DIBBs: Middleware and High Performance Analytics Libraries for Scalable Data Science

Ogres Application Analysis

HPC-ABDS and HPC-FaaS Software

Harp and Twister2 Building Blocks

SPIDAL Data Analytics Library



My view of System GAIMSC

Systems Challenges for GAIMSC

- Microsoft noted we are collectively building the Global AI Supercomputer.
- Generalize by adding **modeling**
- Architecture of the **Global AI and Modeling Supercomputer GAIMSC** must reflect
  - **Global** captures the need to mashup services from many different sources;
  - **AI** captures the incredible progress in machine learning (ML);
  - **Modeling** captures both traditional large-scale simulations and the models and digital twins needed for data interpretation;
  - **Supercomputer** captures that everything is huge and needs to be done quickly and often in real time for streaming applications.
- The GAIMSC includes an intelligent HPC cloud linked via an intelligent HPC Fog to an intelligent HPC edge. We consider this distributed environment as a set of computational and data-intensive nuggets swimming in an intelligent aether.
  - We will use a dataflow graph to define a mesh in the aether

Global AI and Modeling Supercomputer GAIMSC

- There is only a cloud at the logical center but it's physically distributed and owned by a few major players
- There is a very distributed set of devices surrounded by local Fog computing; this forms the logically and physically distribute edge
- The edge is structured and largely data
    - These are two differences from the Grid of the past
    - e.g. self driving car will have its own fog and will not share fog with truck that it is about to collide with
- The cloud and edge will both be very heterogeneous with varying accelerators, memory size and disk structure.
- GAIMSC requires parallel computing to achieve high performance on large ML and simulation nuggets and distributed system technology to build the aether and support the distributed but connected nuggets.
- In the latter respect, the intelligent aether mimics a grid but it is a data grid where there are computations but typically those associated with data (often from edge devices).
    - So unlike the distributed simulation supercomputer that was often studied in previous grids, GAIMSC is a supercomputer aimed at very different data intensive AI-enriched problems.

GAIMSC Global AI & Modeling Supercomputer Questions

- What do gain from the concept? e.g. Ability to work with Big Data community
- What do we lose from the concept? e.g. everything runs as slow as Spark
- Is GAIMSC useful for BDEC2 initiative? For NSF? For DoE?
  For Universities? For Industry? For users?
- Does adding modeling to concept add value?
- What are the research issues for GAIMSC? e.g. how to program?
- What can we do with GAIMSC that we couldn’t do with classic Big Data technologies?
- What can we do with GAIMSC that we couldn’t do with classic HPC technologies?
- Are there deep or important issues associated with the “Global” in GAIMSC?
- Is the concept of an auto-tuned Global AI and Modeling Supercomputer scary?

Integration of Data and Model functions with ML wrappers in GAIMSC

- There is a rapid increase in the integration of ML and simulations.
- ML can analyze results, guide the execution and set up initial configurations (auto-tuning). This is equally true for AI itself -- the GAIMSC will use itself to optimize its execution for both analytics and simulations.
- In principle every transfer of control (job or function invocation, a link from device to the fog/cloud) should pass through an AI wrapper that learns from each call and can decide both if call needs to be executed (maybe we have learned the answer already and need not compute it) and how to optimize the call if it really needs to be executed.
- The digital continuum proposed by BDEC2 is an intelligent aether learning from and informing the interconnected computational actions that are embedded in the aether.
  - Implementing the intelligent aether embracing and extending the edge, fog, and cloud is a major research challenge where bold new ideas are needed!
  - We need to understand how to make it easy to automatically wrap every nugget with ML.

Implementing the GAIMSC

- The new MIDAS middleware designed in SPIDAL has been engineered to support high-performance technologies and yet preserve the key features of the Apache Big Data Software.
  - MIDAS seems well suited to build the prototype intelligent high-performance aether.
- Note this will mix many relatively small nuggets with AI wrappers generating parallelism from the number of nuggets and not internally to the nugget and its wrapper.
- However, there will be also large global jobs requiring internal parallelism for individual large-scale machine learning or simulation tasks.
- Thus parallel computing and distributed systems (grids) must be linked in a deep fashion although the key parallel computing ideas needed for ML are closely related to those already developed for simulations.

Underlying HPC Big Data Convergence Issues

Data and Model in Big Data and Simulations I

- Need to discuss **Data** and **Model** as problems have both intermingled, but we can get insight by separating which allows better understanding of **Big Data - Big Simulation “convergence” (or differences!)**
- The **Model** is a user construction and it has a “**concept**”, **parameters** and gives **results** determined by the computation. We use term “model” in a general fashion to cover all of these.
- **Big Data** problems can be broken up into **Data** and **Model**
  - For **clustering,** the model parameters are cluster centers while the data is set of points to be clustered
  - For **queries**, the model is structure of database and results of this query while the data is whole database queried and SQL query
  - For **deep learning** with ImageNet, the model is chosen network with model parameters as the network link weights. The data is set of images used for training or classification

Data and Model in Big Data and Simulations II

- **Simulations** can also be considered as **Data** plus **Model**
  - **Model** can be formulation with particle dynamics or partial differential equations defined by parameters such as particle positions and discretized velocity, pressure, density values
  - **Data** could be small when just boundary conditions
  - **Data** large with data assimilation (weather forecasting) or when data visualizations are produced by simulation
- **Big Data** implies Data is large but Model varies in size
  - e.g. **LDA** (Latent Dirichlet Allocation) with many topics or **deep learning** has a large model
  - **Clustering** or **Dimension reduction** can be quite small in model size
- **Data** often static between iterations (unless streaming); **Model parameters** vary between iterations
- **Data** and **Model Parameters** are often **confused** in papers as term data used to describe the parameters of models.
- **Models** in **Big Data** and **Simulations** have many similarities and allow **convergence**

Local and Global Machine Learning

- **Many applications** use **LML or Local machine Learning** where machine learning (often from R or Python or Matlab) is run separately on every data item such as on every image
- **But others** are **GML** Global Machine Learning where machine learning is a basic algorithm run over all data items (over all nodes in computer)
  - **maximum likelihood or χ^2** with a sum over the N data items – documents, sequences, items to be sold, images etc. and often links (point-pairs).
  - **GML includes Graph analytics, clustering**/community detection, mixture models, topic determination, Multidimensional scaling, (**Deep**) **Learning Networks**
- Note Facebook may need lots of small graphs (one per person and ~LML) rather than one giant graph of connected people (GML)

Convergence/Divergence Points for HPC-Cloud-Edge-Big Data-Simulation

- **Applications** – Divide use cases into **Data** and **Model** and compare characteristics separately in these two components with 64 Convergence Diamonds (features).
  - Identify importance of streaming data, pleasingly parallel, global/local machine-learning
- **Software** – Single model of **High Performance Computing (HPC) Enhanced Big Data Stack HPC-ABDS**. 21 Layers adding high performance runtime to Apache systems **HPC-FaaS** Programming Model
  - **Serverless** Infrastructure as a Service IaaS
- **Hardware** system designed for functionality and performance of application type e.g. disks, interconnect, memory, CPU acceleration different for machine learning, pleasingly parallel, data management, streaming, simulations
  - Use DevOps to automate deployment of event-driven software defined systems on hardware: **HPCCloud 2.0**
- **Total System Solutions (wisdom) as a Service: HPCCloud 3.0**

Twister 2

Iterative MapReduce

http://www.iterativemapreduce.org/

Application Structure

Structure of Applications

- **Real-time** (**streaming**) data is increasingly common in scientific and engineering research, and it is ubiquitous in commercial Big Data (e.g., social network analysis, recommender systems and consumer behavior classification)
  - So far little use of commercial and Apache technology in analysis of scientific streaming data
- **Pleasingly parallel** applications important in science (long tail) and data communities
- **Commercial-Science application differences:** Search and recommender engines have different structure to deep learning, clustering, topic models, graph analyses such as subgraph mining
  - Latter very sensitive to communication and can be hard to parallelize
  - Search typically not as important in Science as in commercial use as search volume scales by number of users
- Should discuss **data** and **model** separately
  - Term data often used rather sloppily and often refers to model

Distinctive Features of Applications

- Ratio of data to model sizes: vertical axis on next slide
- Importance of Synchronization – ratio of inter-node communication to node computing: horizontal axis on next slide
- Sparsity of Data or Model; impacts value of GPU's or vector computing
- Irregularity of Data or Model
- Geographic distribution of Data as in edge computing; use of streaming (dynamic data) versus batch paradigms
- Dynamic model structure as in some iterative algorithms

Big Data and Simulation Difficulty in Parallelism

Size of Synchronization constraints

Loosely Coupled

Tightly Coupled

Commodity Clouds

HPC Clouds: Accelerators
High Performance Interconnect

HPC Clouds/Supercomputers
Memory access also critical

Size of Disk I/O

MapReduce as in scalable databases

Global Machine Learning
e.g. parallel clustering

Deep Learning

LDA

Graph Analytics e.g. subgraph mining

Pleasingly Parallel
Often independent events

Unstructured Adaptive Sparse

Current major Big Data category

Linear Algebra at core (often not sparse)

Structured Adaptive Sparse

Parameter sweep simulations

Largest scale simulations

Just two problem characteristics
There is also data/compute distribution seen in grid/edge computing

Exascale Supercomputers

Application Nexus of HPC, Big Data, Simulation Convergence

Use-case Data and Model

NIST Collection

Big Data Ogres

Convergence Diamonds

https://bigdatawg.nist.gov/

NBD(NIST Big Data) Requirements WG Use Case Template Aug 11 2013

Use Case Title		
Vertical (area)		
Author/Company/Email		
Actors/Stakeholders and their roles and responsibilities		
Goals		
Use Case Description		
Current Solutions	Compute(System)	
	Storage	
	Networking	
	Software	
Big Data Characteristics	Data Source (distributed/centralized)	
	Volume (size)	
	Velocity (e.g. real time)	
	Variety (multiple datasets, mashup)	
	Variability (rate of change)	
Big Data Science (collection, curation, analysis, action)	Veracity (Robustness Issues, semantics)	
	Visualization	
	Data Quality (syntax)	
	Data Types	
	Data Analytics	
Big Data Specific Challenges (Gaps)		
Big Data Specific Challenges in Mobility		
Security & Privacy Requirements		
Highlight issues for generalizing this use case (e.g. for ref. architecture)		
More Information (URLs)		
Note: <additional comments>		

Note: No proprietary or confidential information should be included
ADD picture of operation or data architecture of application below table

Original Use Case Template

- 26 fields completed for 51 areas
- **Government Operation: 4**
- **Commercial: 8**
- **Defense: 3**
- **Healthcare and Life Sciences: 10**
- **Deep Learning and Social Media: 6**
- **The Ecosystem for Research: 4**
- **Astronomy and Physics: 5**
- **Earth, Environmental and Polar Science: 10**
- **Energy: 1**

- **Security & Privacy Enhanced version 2**
- **BDEC HPC enhanced version**

51 Detailed Use Cases: Contributed July-September 2013
Covers goals, data features such as 3 V's, software, hardware

- **Government Operation(4):** National Archives and Records Administration, Census Bureau
- **Commercial(8):** Finance in Cloud, Cloud Backup, Mendeley (Citations), Netflix, Web Search, Digital Materials, Cargo shipping (as in UPS)
- **Defense(3):** Sensors, Image surveillance, Situation Assessment
- **Healthcare and Life Sciences(10):** Medical records, Graph and Probabilistic analysis, Pathology, Bioimaging, Genomics, Epidemiology, People Activity models, Biodiversity
- **Deep Learning and Social Media(6):** Driving Car, Geolocate images/cameras, Twitter, Crowd Sourcing, Network Science, NIST benchmark datasets
- **The Ecosystem for Research(4):** Metadata, Collaboration, Language Translation, Light source experiments
- **Astronomy and Physics(5):** Sky Surveys including comparison to simulation, Large Hadron Collider at CERN, Belle Accelerator II in Japan
- **Earth, Environmental and Polar Science(10):** Radar Scattering in Atmosphere, Earthquake, Ocean, Earth Observation, Ice sheet Radar scattering, Earth radar mapping, Climate simulation datasets, Atmospheric turbulence identification, Subsurface Biogeochemistry (microbes to watersheds), AmeriFlux and FLUXNET gas sensors
- **Energy(1):** Smart grid
- Published by NIST as version 2 https://bigdatawg.nist.gov/_uploadfiles/NIST.SP.1500-3r1.pdf with common set of 26 features recorded for each use-case

26 Features for each use case Biased to science

Part of Property Summary Table

No.	Use Case	Volume	Velocity	Variety	Software	Analytics
23	**M0172 World Population Scale Epidemiological Study**	100TB	Data feeding into the simulation is small but real time data generated by simulation is massive.	Can be rich with various population activities, geographical, socio-economic, cultural variations	Charm++, MPI	Simulations on a Synthetic population
24	**M0173 Social Contagion Modeling for Planning**	10s of TB per year	During social unrest events, human interactions and mobility leads to rapid changes in data; e.g., who follows whom in Twitter.	Data fusion a big issue. How to combine data from different sources and how to deal with missing or incomplete data?	Specialized simulators, open source software, and proprietary modeling environments. Databases.	Models of behavior of humans and hard infrastructures, and their interactions. Visualization of results
25	**M0141 Biodiversity and LifeWatch**	N/A	Real time processing and analysis in case of the natural or industrial disaster	Rich variety and number of involved databases and observation data	RDMS	Requires advanced and rich visualization
26	**M0136 Large-scale Deep Learning**	Current datasets typically 1 to 10 TB. Training a self-driving car could take 100 million images.	Much faster than real-time processing is required. For autonomous driving need to process 1000's high-resolution (6 megapixels or more) images per second.	Neural Net very heterogeneous as it learns many different features	In-house GPU kernels and MPI-based communication developed by Stanford. C++/Python source.	Small degree of batch statistical pre-processing; all other data analysis is performed by the learning algorithm itself.
27	**M0171 Organizing large-scale image collections**	500+ billion photos on Facebook, 5+ billion photos on Flickr.	over 500M images uploaded to Facebook each day	Images and metadata including EXIF tags (focal distance, camera type, etc),	Hadoop Map-reduce, simple hand-written multithreaded tools (ssh and sockets for communication)	Robust non-linear least squares optimization problem. Support Vector Machine
28	**M0160 Truthy**	30TB/year compressed data	Near real-time data storage, querying & analysis	Schema provided by social media data source. Currently using Twitter only. We plan to expand	Hadoop IndexedHBase & HDFS. Hadoop, Hive, Redis for data management. Python:	Anomaly detection, stream clustering, signal classification and online-learning; Information diffusion,

51 Use Cases: What is Parallelism Over?

- **People**: either the users (but see below) or subjects of application and often both
- **Decision makers** like researchers or doctors (users of application)
- **Items** such as Images, EMR, Sequences below; observations or contents of online store
  - **Images** or “Electronic Information nuggets”
  - **EMR**: Electronic Medical Records (often similar to people parallelism)
  - Protein or Gene **Sequences**;
  - **Material** properties, **Manufactured Object** specifications, etc., in custom dataset
  - **Modelled entities** like vehicles and people
- **Sensors** – Internet of Things
- **Events** such as detected anomalies in telescope or credit card data or atmosphere
- **(Complex) Nodes** in RDF Graph
- **Simple nodes** as in a learning network
- **Tweets**, **Blogs**, **Documents**, **Web Pages,** etc.
  - And characters/words in them
- **Files** or data to be backed up, moved or assigned metadata
- **Particles**/**cells**/**mesh points** as in parallel simulations

Sample Features of 51 Use Cases I

- **PP (26)** **“All”** Pleasingly Parallel or Map Only
- **MR (18)** Classic MapReduce MR (add MRStat below for full count)
- **MRStat (7**) Simple version of MR where key computations are simple reduction as found in statistical averages such as histograms and averages
- **MRIter (23**) Iterative MapReduce or MPI (Flink, Spark, Twister)
- **Graph (9)** Complex graph data structure needed in analysis
- **Fusion (11)** Integrate diverse data to aid discovery/decision making; could involve sophisticated algorithms or could just be a portal
- **Streaming (41)** Some data comes in incrementally and is processed this way
- **Classify** **(30)** Classification: divide data into categories
- **S/Q (12)** Index, Search and Query

Sample Features of 51 Use Cases II

- **CF (4)** Collaborative Filtering for recommender engines
- **LML (36)** Local Machine Learning (Independent for each parallel entity) – application could have GML as well
- **GML (23)** Global Machine Learning: Deep Learning, Clustering, LDA, PLSI, MDS,
  - Large Scale Optimizations as in Variational Bayes, MCMC, Lifted Belief Propagation, Stochastic Gradient Descent, L-BFGS, Levenberg-Marquardt . Can call EGO or Exascale Global Optimization with scalable parallel algorithm
- **Workflow (51)** Universal
- **GIS (16)** Geotagged data and often displayed in ESRI, Microsoft Virtual Earth, Google Earth, GeoServer etc.
- **HPC (5)** Classic large-scale simulation of cosmos, materials, etc. generating (visualization) data
- **Agent (2)** Simulations of models of data-defined macroscopic entities represented as agents

BDEC2 and NIST Use Cases

53 NIST Use Cases for Research Space I

- **GOVERNMENT OPERATION**
  - 1: Census 2010 and 2000—Title 13 Big Data
  - 2: NARA Accession, Search, Retrieve, Preservation
  - 3: Statistical Survey Response Improvement
  - 4: Non-Traditional Data in Statistical Survey Response Improvement (Adaptive Design)

  1-4 are related to social science survey problems and are "Classic Data+ML" with interesting algorithms (recommender engines) plus databases and important privacy issues which are present in research cases

- **COMMERCIAL**
  - 5: Cloud Eco-System for Financial Industries NO
  - 6: Mendeley—An International Network of Research
  - 7: Netflix Movie Service NO
  - 8: Web Search NO
  - 9: Big Data Business Continuity and Disaster Recovery Within a Cloud Eco-System NO
  - 10: Cargo Shipping Edge Computing NO
  - 11: Materials Data for Manufacturing
  - 12: Simulation-Driven Materials Genomics

  6 is "Classic Data+ML" with Text Analysis (citation identification, topic models etc.)
  10 is DHL/Fedex/UPS and has no direct scientific analog. However, it is a good example of Edge computing system of a similar nature to the scientific research case.
  11 and 12 are material science covered in BDEC2 meeting

53 NIST Use Cases for Research Space II

- **DEFENSE**
  - 13: Cloud Large-Scale Geospatial Analysis and Visualization
  - 14: Object Identification and Tracking from Wide-Area Large Format Imagery or Full Motion Video—Persistent Surveillance
  - 15: Intelligence Data Processing and Analysis

  13-15 are very similar to disaster response problems. They involve extensive "Classic Data+ML" for sensor collections and/or image processing. GIS and spatial analysis are important as in BDEC2 Pathology and Spatial Imagery talk. The geospatial aspect of applications means they are similar to earth science examples.

- Color coding of use cases
- NO means not similar to research application
- Red means not relevant to BDEC2
- Orange means related to BDEC2 Bloomington presentations
- Black are unique use cases of relevance to BDEC2 but not presented at Bloomington
- Purple are comments

Use Cases III: HEALTH CARE AND LIFE SCIENCES

- 16: Electronic Medical Record
- 17: Pathology Imaging/Digital Pathology
- 18: Computational Bioimaging
- 19: Genomic Measurements
- 20: Comparative Analysis for Metagenomes and Genome
- 21: Individualized Diabetes Management
- 22: Statistical Relational Artificial Intelligence for Health Care
- 23: World Population-Scale Epidemiological Study
- 24: Social Contagion Modeling for Planning, Public Health, and Disaster Management
- 25: Biodiversity and LifeWatch

Comments on Use Cases III

- 16 and 22 are "classic data + ML + database" based use cases using an important technique, understood by the community but not presented at BDEC2
- 17 came originally from Saltz's group and was updated in his BDEC2 talk
- 18 describes biology image processing from many instruments microscopes, MRI and light sources. The latter was directly discussed at BDEC2 and the other instruments were implicit.
- 19 and 20 are well recognized as a distributed Big Data problems with significant computing. They were represented by Chandrasekaran's presentation at BDEC2 which inevitably only covered part (gene assembly) of problem.
- 21 relies on "classic data + graph analytics" which was not discussed in BDEC2 meeting but is certainly actively pursued.
- 23 and 24 originally came from Marathe and were updated in his BDEC2 presentation on massive bio-social systems
- 25 generalizes BDEC2 talks by Taufer and Rahnemoonfar on ocean and land monitoring and sensor array analysis.

Use Cases IV: DEEP LEARNING AND SOCIAL MEDIA

- 26: Large-Scale Deep Learning
- 27: Organizing Large-Scale, Unstructured Collections of Consumer Photos NO
- 28: Truthy—Information Diffusion Research from Twitter Data
- 29: Crowd Sourcing in the Humanities as Source for Big and Dynamic Data
- 30: CINET—Cyberinfrastructure for Network (Graph) Science and Analytics
- 31: NIST Information Access Division—Analytic Technology Performance Measurements, Evaluations, and Standards
- 26 on deep learning was covered in great depth at the BDEC2 meeting
- 27 describes an interesting image processing challenge of geolocating multiple photographs which is not so far directly related to scientific data analysis although related image processing algorithms are certainly important
- 28-30 are "classic data + ML" use cases with a focus on graph and text mining algorithms not covered in BDEC2 but certainly relevant to the process
- 31 on benchmarking and standard datasets is related to BigDataBench talk at end of BDEC2 meeting and Fosters talk on a model database

53 NIST Use Cases for Research Space VI

- **THE ECOSYSTEM FOR RESEARCH**
- 32: DataNet Federation Consortium
- 33: The Discinnet Process NO
- 34: Semantic Graph Search on Scientific Chemical and Text-Based Data
- 35: Light Source Beamlines

32 covers data management with iRODS which is well regarded by the community but not discussed in BDEC2.
33 is a Teamwork approach that doesn't seem relevant to BDEC2
34 is a "classic data+ML" use case with a similar comments to 28-30
35 was covered with more advanced deep learning algorithms in Yager and Foster's BDEC2 talks

ASTRONOMY AND PHYSICS

- 36: Catalina Real-Time Transient Survey: A Digital, Panoramic, Synoptic Sky Survey
- 37: DOE Extreme Data from Cosmological Sky Survey and Simulations
- 38: Large Survey Data for Cosmology
- 39: Particle Physics—Analysis of Large Hadron Collider Data: Discovery of Higgs Particle
- 40: Belle II High Energy Physics Experiment

36 to 38 are "Classic Data+ML" astronomy use cases related to BDEC2 SKA presentation and covering both archival and event detection cases. Use case 37 covers the integration of simulation data and observational data FOR ASTRONOMY; A TOPIC COVERED IN OTHER CASES AT BDEC2.
39 and 40 are "Classic Data+ML" use cases for accelerator data analysis. This was not covered in BDEC2 but is currently the largest volume scientific data analysis problem whose importance and relevance is well understood.

Use Cases VII: EARTH, ENVIRONMENTAL, AND POLAR SCIENCE

- 41: European Incoherent Scatter Scientific Association 3D Incoherent Scatter Radar System Big Radar instrument monitoring atmosphere.
- 42: Common Operations of Environmental Research Infrastructure
- 43: Radar Data Analysis for the Center for Remote Sensing of Ice Sheets
- 44: Unmanned Air Vehicle Synthetic Aperture Radar (UAVSAR) Data Processing, Data Product Delivery, and Data Services
- 45: NASA Langley Research Center/ Goddard Space Flight Center iRODS Federation Test Bed
- 46: MERRA Analytic Services (MERRA/AS) Instrument
- 47: Atmospheric Turbulence – Event Discovery and Predictive Analytics Imaging
- 48: Climate Studies Using the Community Earth System Model at the U.S. Department of Energy (DOE) NERSC Center
- 49: DOE Biological and Environmental Research (BER) Subsurface Biogeochemistry Scientific Focus Area
- 50: DOE BER AmeriFlux and FLUXNET Networks Sensor Networks
- 2-1: NASA Earth Observing System Data and Information System (EOSDIS) Instrument
- 2-2: Web-Enabled Landsat Data (WELD) Processing Instrument

Comments on Use Cases VII

- 41 43 44 are "Classic Data+ML" use cases involving radar data from different instruments-- specialized ground, vehicle/plane, satellite - not directly covered in BDEC2
- 2-1 and 2-2 are use cases similar to 41 43 and 44 but applied to EOSDIS and LANDSAT earth observations from satellites in multiple modalities.
- 42 49 and 50 are "Classic Data+ML" environmental sensor arrays that extend the scope of talks of Taufer and Rahnemoonfar at BDEC2. See also use case 25 above
- 45 to 47 describe datasets from instruments and computations relevant to climate and weather. It relates to BDEC2 talk by Denvil and Miyoshi. 47 discusses the correlation of aircraft turbulent reports with simulation datasets
- 48 is data analytics and management associated with climate studies as covered in BDEC2 talk by Denvil

Use Cases VIII: ENERGY

- 51: Consumption Forecasting in Smart Grids

51 is a different subproblem but in the same area as Pothen and Azad's talk on the electric power grid at BDEC2. This is a challenging edge computing problem as a large number of distributed but correlated sensors

- SC-18 BOF Application/Industry Perspective by David Keyes, King Abdullah University of Science and Technology (KAUST)
- https://www.exascale.org/bdec/sites/www.exascale.org.bdec/files/SC18_BDEC2_BoF-Keyes.pdf

This is a presentation by David Keyes on seismic imaging for oil discovery and exploitation. It is "Classic Data+ML" for an array of sonic sensors

BDEC2 Use Cases I: Classic Observational Data plus ML

- BDEC2-1: M. Deegan, Big Data and Extreme Scale Computing, 2nd Series (BDEC2) - Statement of Interest from the Square Kilometre Array Organisation (SKAO)
- **Environmental Science**
- BDEC2-2: M. Rahnemoonfar, Semantic Segmentation of Underwater Sonar Imagery based on Deep Learning
- BDEC2-3: M. Taufer, Cyberinfrastructure Tools for Precision Agriculture in the 21st Century
- **Healthcare and Life sciences**
- BDEC2-4: J. Saltz, Multiscale Spatial Data and Deep Learning
- BDEC2-5: R. Stevens, Exascale Deep Learning for Cancer
- BDEC2-6: S. Chandrasekaran, Development of a parallel algorithm for whole genome alignment for rapid delivery of personalized genomics
- BDEC2-7: M. Marathe, Pervasive, Personalized and Precision (P3) analytics for massive bio-social systems

Instruments include Satellites, UAV's, Sensors (see edge examples), Light sources (X-ray MRI Microscope etc.), Telescopes, Accelerators, Tokomaks (Fusion), Computers (as in Control, Simulation, Data, ML Integration)

BDEC2 Use Cases II: Control, Simulation, Data, ML Integration

- BDEC2-8: W. Tang, New Models for Integrated Inquiry: Fusion Energy Exemplar
- BDEC2-9: O. Beckstein, Convergence of data generation and analysis in the biomolecular simulation community
- BDEC2-10: S. Denvil, From the production to the analysis phase: new approaches needed in climate modeling
- BDEC2-11: T. Miyoshi, Prediction Science: The 5th Paradigm Fusing the Computational Science and Data Science (weather forecasting)

See also Marathe and Stevens talks
See also instruments under Classic Observational Data plus ML

- **Material Science**
- BDEC2-12: K. Yager, Autonomous Experimentation as a Paradigm for Materials Discovery
- BDEC2-13: L. Ward, Deep Learning, HPC, and Data for Materials Design
- BDEC2-14: J. Ahrens, A vision for a validated distributed knowledge base of material behavior at extreme conditions using the Advanced Cyberinfrastructure Platform
- BDEC2-15: T. Deutsch, Digital transition of Material Nano-Characterization.

Comments on Control, Simulation, Data, ML Integration

- Simulations often involve outside Data but always inside Data (from simulation itself). Fields covered include Materials (nano), Climate, Weather, Biomolecular, Virtual tissues (no use case written up)
- We can see ML wrapping simulations to achieve many goals. ML replaces functions and/or ML guides functions
  - Initial Conditions
  - Boundary Conditions
  - Data assimilation
  - Configuration -- blocking, use of cache etc.
  - Steering and Control
  - Support multi-scale
  - ML learns from previous simulations and so can predict function calls

- Digital Twins are a commercial link between simulation and systems
- There are fundamental simulations covered by laws of physics and growingly Complex System simulations with Bio (tissue) or social entities.

BDEC2 Use Cases III: Edge Computing

- **Smart City and Related Edge Applications**
- BDEC2-16: P. Beckman, Edge to HPC Cloud
- BDEC2-17: G. Ricart, Smart Community CyberInfrastructure at the Speed of Life
- BDEC2-18: T. El-Ghazawi, Convergence of AI, Big Data, Computing and IOT (ABCI)-Smart City as an Application Driver and Virtual Intelligence Management (VIM)
- BDEC2-19: M. Kondo, The Challenges and opportunities of BDEC systems for Smart Cities

- **Other Edge Applications**
- BDEC2-20: A Pothen, High-End Data Science and HPC for the Electrical Power Grid
- BDEC2-21: J. Qiu, Real-Time Anomaly Detection from Edge to HPC-Cloud

There are correlated edge devices such as power grid and nearby vehicles (racing, road). Also largely independent edge devices interacting via databases such as surveillance cameras

BDEC Use Cases IV

- **BDEC Ecosystem**
- BDEC2-22: I Foster, Learning Systems for Deep Science
- BDEC2-23: W. Gao, BigDataBench: A Scalable and Unified Big Data and AI Benchmark Suite

- ***Image-based Applications***
- One cross-cutting theme is understanding **Generalized (light, sound, other sensors such as temperature, chemistry, moisture) Images with** 2D, 3D spatial and time dependence
- Modalities include Radar, MRI, Microscopes, Surveillance and other cameras, X-ray scattering, UAV hosted, and related non-optical sensor networks as in agriculture, wildfires, disaster monitoring and Oil exploration. GIS and geospatial properties are often relevant

NIST Generic Data Processing Use Cases

10 Generic Data Processing Use Cases

1) Multiple users performing interactive queries and updates on a database with basic availability and eventual consistency (BASE = (Basically Available, Soft state, Eventual consistency) as opposed to ACID = (Atomicity, Consistency, Isolation, Durability))
2) Perform real time analytics on data source streams and notify users when specified events occur
3) Move data from external data sources into a highly horizontally scalable data store, transform it using highly horizontally scalable processing (e.g. Map-Reduce), and return it to the horizontally scalable data store (ELT Extract Load Transform)
4) Perform batch analytics on the data in a highly horizontally scalable data store using highly horizontally scalable processing (e.g MapReduce) with a user-friendly interface (e.g. SQL like)
5) Perform interactive analytics on data in analytics-optimized database with 5A) Science
6) Visualize data extracted from horizontally scalable Big Data store
7) Move data from a highly horizontally scalable data store into a traditional Enterprise Data Warehouse (EDW)
8) Extract, process, and move data from data stores to archives
9) Combine data from Cloud databases and on premise data stores for analytics, data mining, and/or machine learning
10) Orchestrate multiple sequential and parallel data transformations and/or analytic processing using a workflow manager

Access Pattern 5A. Perform interactive analytics on observational scientific data

Science Analysis Code, Mahout, R

Grid or Many Task Software, Hadoop, Spark, Giraph, Pig …

Data Storage: HDFS, Hbase, File Collection

Direct Transfer

Transport batch of data to primary analysis data system

Streaming Twitter data for Social Networking

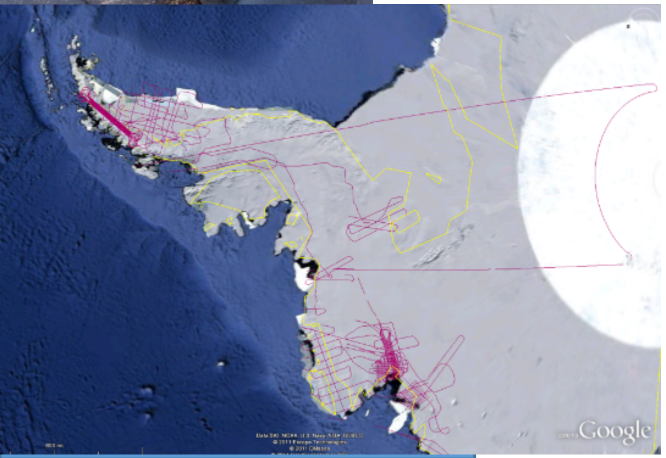
Record Scientific Data in "field"

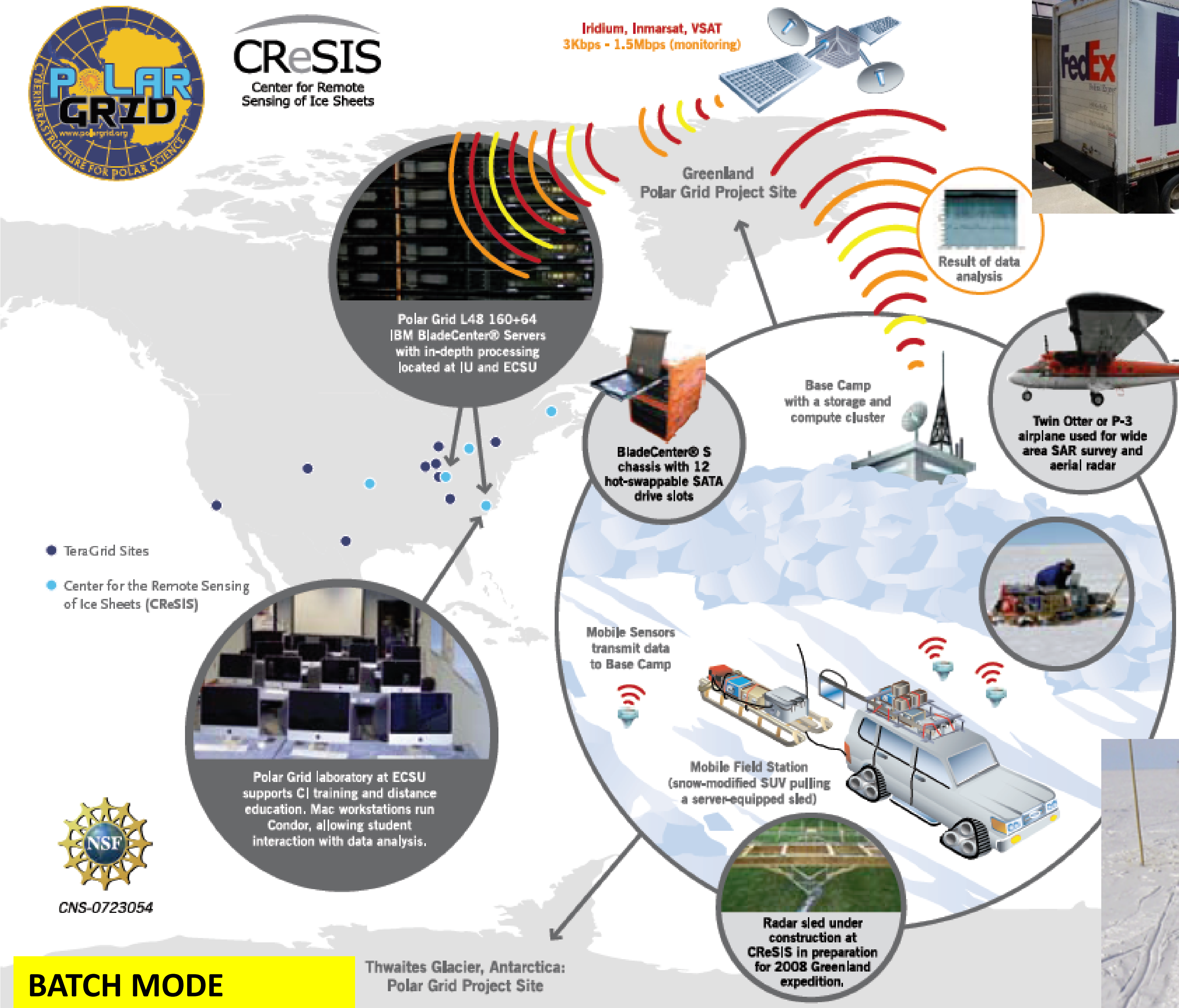
Local Accumulate and initial computing

Examples include LHC, Remote Sensing, Astronomy and Bioinformatics

Polar Grid

Lightweight Cyberinfrastructure to support mobile Data gathering expeditions plus classic central resources (as a cloud)





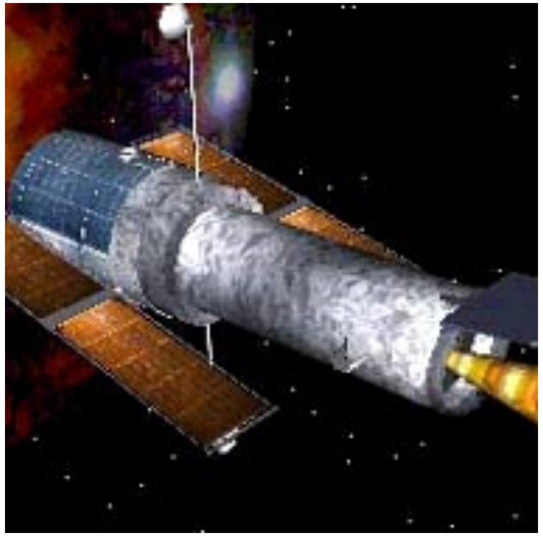
CNS-0723054

BATCH MODE

Tracking the Heavens

"The Universe is now being explored systematically, in a panchromatic way, over a range of spatial and temporal scales that lead to a more complete, and less biased understanding of its constituents, their evolution, their origins, and the physical processes governing them."

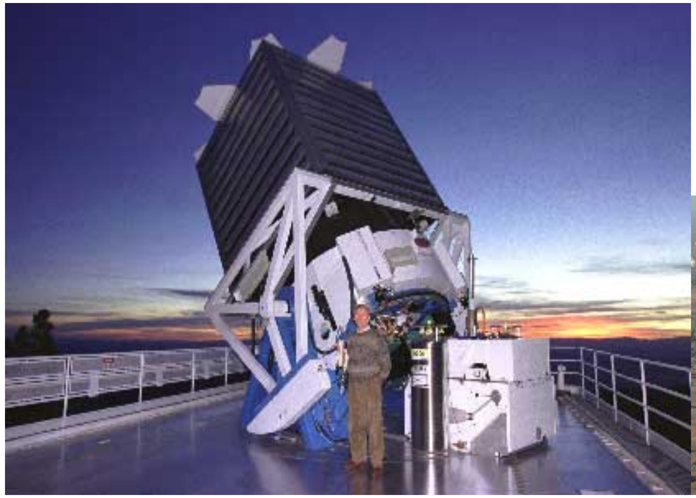
Towards a National Virtual Observatory



Hubble Telescope

Palomar Telescope

Sloan Telescope



SKA
SQUARE KILOMETRE ARRAY

DISTRIBUTED LARGE INSTRUMENT MODE

Other Use-case Collections

7 Computational Giants of NRC Massive Data Analysis Report

http://www.nap.edu/catalog.php?record_id=18374 **Big Data Models?**

1) **G1:** Basic Statistics e.g. MRStat
2) **G2:** Generalized N-Body Problems
3) **G3:** Graph-Theoretic Computations
4) **G4:** Linear Algebraic Computations
5) **G5:** Optimizations e.g. Linear Programming
6) **G6:** Integration e.g. LDA and other GML
7) **G7:** Alignment Problems e.g. BLAST

HPC (Simulation) Benchmark Classics

- **Linpack** or HPL: Parallel LU factorization for solution of linear equations; **HPCG**
- **NPB** version 1: Mainly classic HPC solver kernels
  - MG: Multigrid
  - CG: Conjugate Gradient
  - FT: Fast Fourier Transform
  - IS: Integer sort
  - EP: Embarrassingly Parallel
  - BT: Block Tridiagonal
  - SP: Scalar Pentadiagonal
  - LU: Lower-Upper symmetric Gauss Seidel

Simulation Models

13 Berkeley Dwarfs

1) Dense Linear Algebra
2) Sparse Linear Algebra
3) Spectral Methods
4) N-Body Methods
5) Structured Grids
6) Unstructured Grids
7) MapReduce
8) Combinational Logic
9) Graph Traversal
10) Dynamic Programming
11) Backtrack and Branch-and-Bound
12) Graphical Models
13) Finite State Machines

First 6 of these correspond to Colella's original. (Classic simulations)
Monte Carlo dropped.
N-body methods are a subset of Particle in Colella.

Note a little inconsistent in that MapReduce is a programming model and spectral method is a numerical method.
Need multiple facets to classify use cases!

Largely Models for Data or Simulation

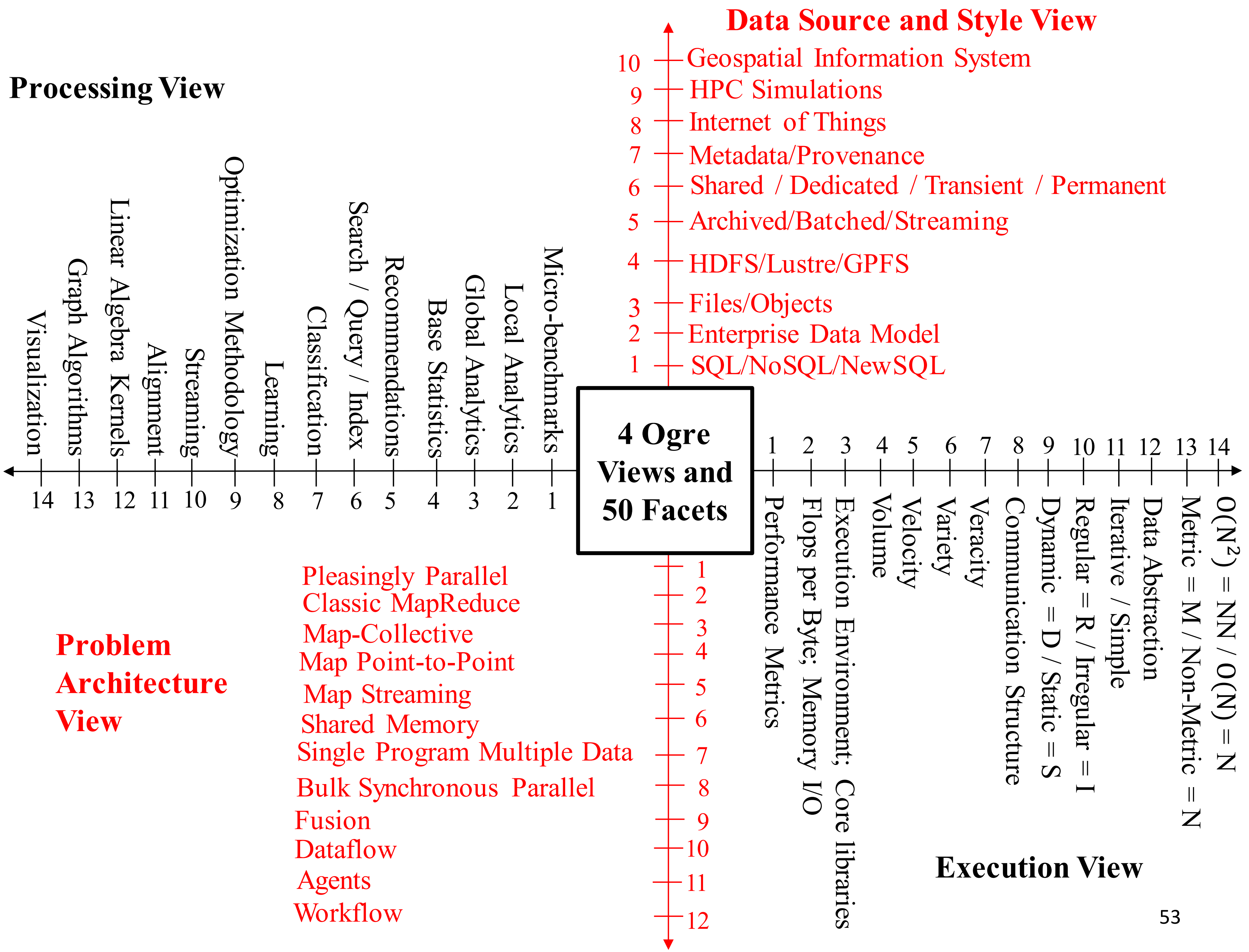
Classifying Use cases

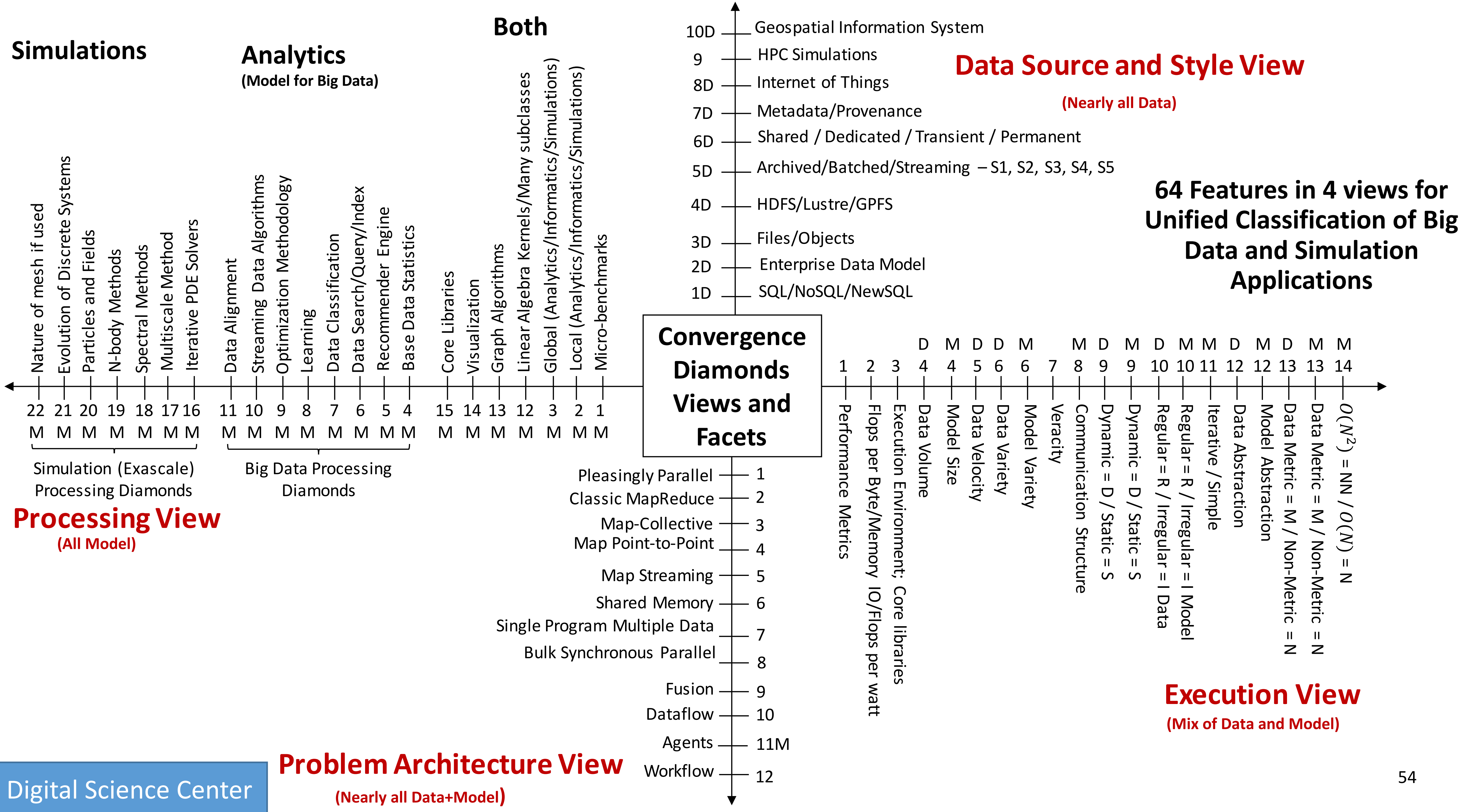
Classifying Use Cases

- The **Big Data Ogres** built on a collection of 51 big data uses gathered by the NIST Public Working Group where 26 properties were gathered for each application.
- This information was combined with other studies including the **Berkeley dwarfs**, the **NAS parallel benchmarks** and the **Computational Giants of the NRC Massive Data Analysis Report**.
- The Ogre analysis led to a set of **50 features** divided into four views that could be used to categorize and distinguish between applications.
- The four views are **Problem Architecture** (Macro pattern); **Execution Features** (Micro patterns); **Data Source and Style**; and finally the **Processing View** or runtime features.
- We generalized this approach to integrate Big Data and Simulation applications into a single classification looking separately at **Data** and **Model** with the total facets growing to 64 in number, called **convergence diamonds**, and split between the same 4 views.
- A mapping of facets into work of the SPIDAL project has been given.

The original 50 Ogres in 4 views

- Processing
- Data Source and Style
- Problem Architecture (metapatterns)
- Execution (micropatterns)





Simulations
Analytics
(Model for Big Data)
Both
Nature of mesh if used 22 M
Evolution of Discrete Systems 21 M
Particles and Fields 20 M
N-body Methods 19 M
Spectral Methods 18 M
Multiscale Method 17 M
Iterative PDE Solvers 16 M
Simulation (Exascale) Processing Diamonds
Data Alignment 11 M
Streaming Data Algorithms 10 M
Optimization Methodology 9 M
Learning 8 M
Data Classification 7 M
Data Search/Query/Index 6 M
Recommender Engine 5 M
Base Data Statistics 4 M
Big Data Processing Diamonds
Core Libraries 15 M
Visualization 14 M
Graph Algorithms 13 M
Linear Algebra Kernels/Many subclasses 12 M
Global (Analytics/Informatics/Simulations) 3 M
Local (Analytics/Informatics/Simulations) 2 M
Micro-benchmarks 1 M
Processing View
(All Model)
Convergence Diamonds Views and Facets
10D Geospatial Information System
9 HPC Simulations
8D Internet of Things
7D Metadata/Provenance
6D Shared / Dedicated / Transient / Permanent
5D Archived/Batched/Streaming – S1, S2, S3, S4, S5
4D HDFS/Lustre/GPFS
3D Files/Objects
2D Enterprise Data Model
1D SQL/NoSQL/NewSQL
Data Source and Style View
(Nearly all Data)
64 Features in 4 views for Unified Classification of Big Data and Simulation Applications
1 Performance Metrics
2 Flops per Byte/Memory IO/Flops per watt
3 Execution Environment; Core libraries
D 4 Data Volume
M 4 Model Size
D 5 Data Velocity
D 6 Data Variety
M 6 Model Variety
7 Veracity
M 8 Communication Structure
D 9 Dynamic = D / Static = S
M 9 Dynamic = D / Static = S
D 10 Regular = R / Irregular = I Data
M 10 Regular = R / Irregular = I Model
M 11 Iterative / Simple
D 12 Data Abstraction
M 12 Model Abstraction
D 13 Data Metric = M / Non-Metric = N
M 13 Data Metric = M / Non-Metric = N
M 14 $O(N^2)$ = NN / $O(N)$ = N
Execution View
(Mix of Data and Model)
Pleasingly Parallel 1
Classic MapReduce 2
Map-Collective 3
Map Point-to-Point 4
Map Streaming 5
Shared Memory 6
Single Program Multiple Data 7
Bulk Synchronous Parallel 8
Fusion 9
Dataflow 10
Agents 11M
Workflow 12
Problem Architecture View
(Nearly all Data+Model)

Convergence Diamonds and their 4 Views I

- One **view** is the overall **problem architecture or macropatterns** which is naturally related to the machine architecture needed to support application.
  - **Unchanged** from Ogres and describes properties of problem such as "Pleasing Parallel" or "Uses Collective Communication"
- The **execution (computational) features or micropatterns view**, describes issues such as I/O versus compute rates, iterative nature and regularity of computation and the classic V's of Big Data: defining problem size, rate of change, etc.
  - **Significant changes** from ogres to separate **Data** and **Model** and add characteristics of Simulation models. e.g. both model and data have "V's"; Data Volume, Model Size
  - *e.g. $O(N^2)$ Algorithm* relevant to big data or big simulation model

Convergence Diamonds and their 4 Views II

- The **data source & style view** includes facets specifying how the data is collected, stored and accessed. Has classic database characteristics
  - Simulations can have facets here to describe input or output data
  - Examples: Streaming, files versus objects, HDFS v. Lustre
- **Processing view** has model (not data) facets which describe types of processing steps including nature of algorithms and kernels by model e.g. Linear Programming, Learning, Maximum Likelihood, Spectral methods, Mesh type,
  - mix of Big Data Processing View and Big Simulation Processing View and includes some facets like "uses linear algebra" needed in both: has specifics of key simulation kernels and in particular includes facets seen in NAS Parallel Benchmarks and Berkeley Dwarfs
- **Instances of Diamonds** are particular problems and a set of Diamond instances that cover enough of the facets could form a comprehensive **benchmark/mini-app** set
- Diamonds and their instances can be **atomic** or **composite**

Twister 2

Iterative MapReduce

http://www.iterativemapreduce.org/

Programming Environment for Global AI and Modeling Supercomputer GAIMSC

Ways of adding High Performance to Global AI (and Modeling) Supercomputer

- Fix performance issues in Spark, Heron, Hadoop, Flink etc.
  - Messy as some features of these big data systems intrinsically slow in some (not all) cases
  - All these systems are “monolithic” and difficult to deal with individual components
- Execute HPBDC from classic big data system with custom communication environment – approach of Harp for the relatively simple Hadoop environment
- Provide a native Mesos/Yarn/Kubernetes/HDFS high performance execution environment with all capabilities of Spark, Hadoop and Heron – goal of Twister2
- Execute with MPI in classic (Slurm, Lustre) HPC environment
- Add modules to existing frameworks like Scikit-Learn or Tensorflow either as new capability or as a higher performance version of existing module.

GAIMSC Programming Environment Components I

Area	Component	Implementation	Comments: User API
Architecture Specification	Coordination Points	State and Configuration Management; Program, Data and Message Level	Change execution mode; save and reset state
	Execution Semantics	Mapping of Resources to Bolts/Maps in Containers, Processes, Threads	Different systems make different choices - why?
	Parallel Computing	Spark Flink Hadoop Pregel MPI modes	Owner Computes Rule
Job Submission	(Dynamic/Static) Resource Allocation	Plugins for Slurm, Yarn, Mesos, Marathon, Aurora	Client API (e.g. Python) for Job Management
Task System	Task migration	Monitoring of tasks and migrating tasks for better resource utilization	Task-based programming with Dynamic or Static Graph API; FaaS API; Support accelerators (CUDA,FPGA, KNL)
	Elasticity	OpenWhisk	
	Streaming and FaaS Events	Heron, OpenWhisk, Kafka/RabbitMQ	
	Task Execution	Process, Threads, Queues	
	Task Scheduling	Dynamic Scheduling, Static Scheduling, Pluggable Scheduling Algorithms	
	Task Graph	Static Graph, Dynamic Graph Generation	

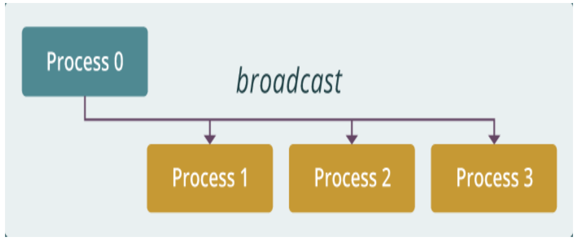
GAIMSC Programming Environment Components II

Area	Component	Implementation	Comments
Communication API	Messages	Heron	This is user level and could map to multiple communication systems
	Dataflow Communication	Fine-Grain Twister2 Dataflow communications: MPI,TCP and RMA Coarse grain Dataflow from NiFi, Kepler?	Streaming, ETL data pipelines; Define new Dataflow communication API and library
	BSP Communication Map-Collective	Conventional MPI, Harp	MPI Point to Point and Collective API
Data Access	Static (Batch) Data	File Systems, NoSQL, SQL	Data API
	Streaming Data	Message Brokers, Spouts	
Data Management	Distributed Data Set	Relaxed Distributed Shared Memory(immutable data), Mutable Distributed Data	Data Transformation API; Spark RDD, Heron Streamlet
Fault Tolerance	Check Pointing	Upstream (streaming) backup; Lightweight; Coordination Points; Spark/Flink, MPI and Heron models	Streaming and batch cases distinct; Crosses all components
Security	Storage, Messaging, execution	Research needed	Crosses all Components

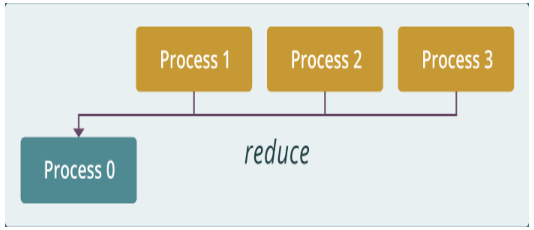
Integrating HPC and Apache Programming Environments

- **Harp-DAAL** with a kernel Machine Learning library exploiting the Intel node library DAAL and HPC communication collectives within the Hadoop ecosystem.
  - Harp-DAAL supports all 5 classes of data-intensive AI first computation, from pleasingly parallel to machine learning and simulations.
- **Twister2** is a toolkit of components that can be packaged in different ways
  - Integrated batch or streaming data capabilities familiar from Apache Hadoop, Spark, Heron and Flink but with high performance.
  - Separate bulk synchronous and data flow communication;
  - Task management as in Mesos, Yarn and Kubernetes
  - Dataflow graph execution models
  - Launching of the Harp-DAAL library with native Mesos/Kubernetes/HDFS environment
  - Streaming and repository data access interfaces,
  - In-memory databases and fault tolerance at dataflow nodes. (use RDD to do classic checkpoint-restart)

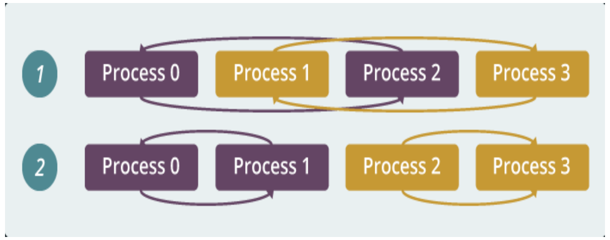
Run time software for Harp



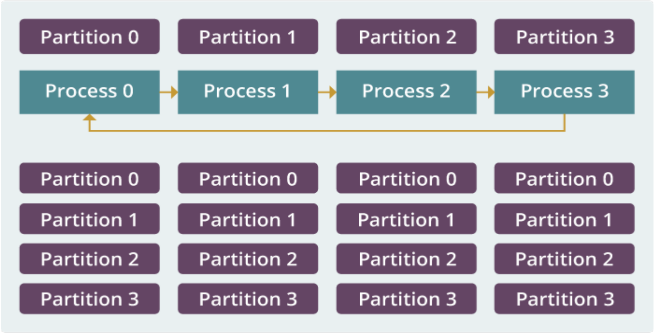
broadcast



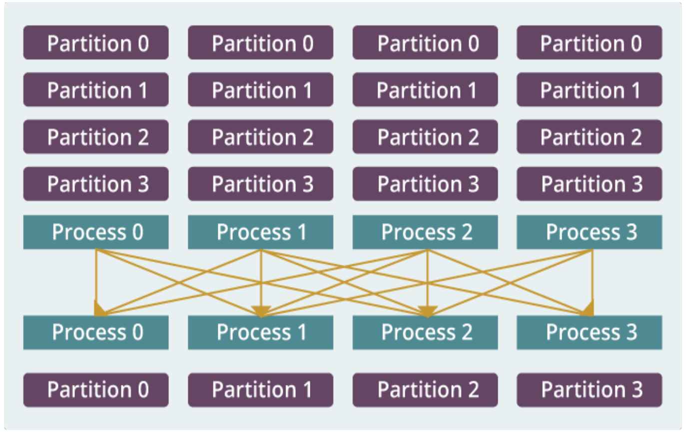
reduce



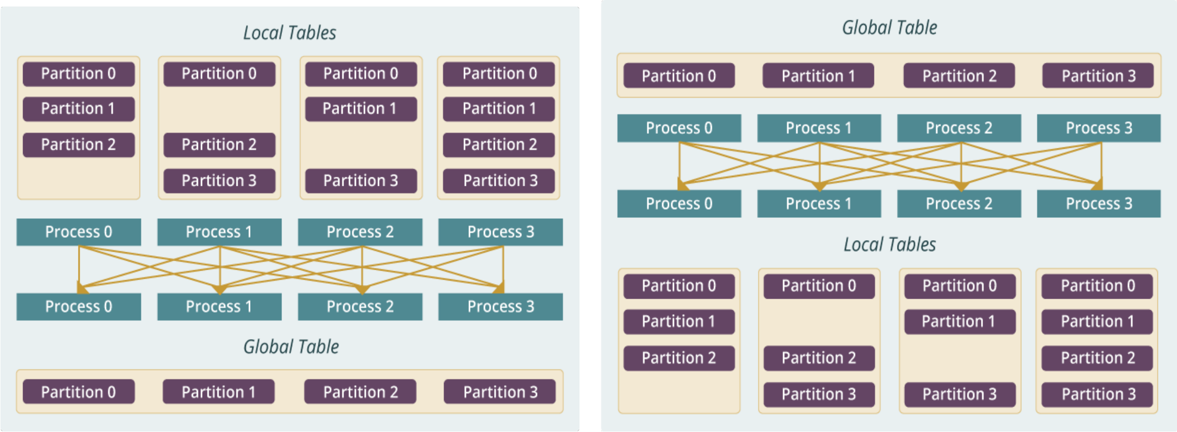
allreduce



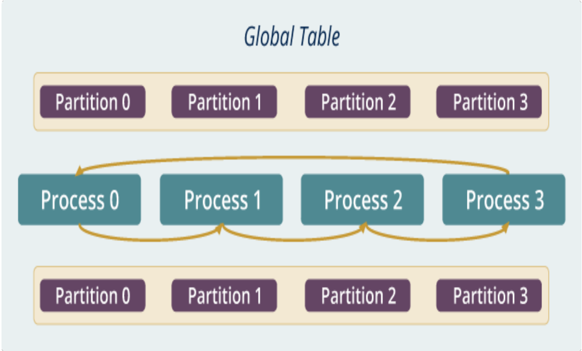
allgather



regroup



push & pull

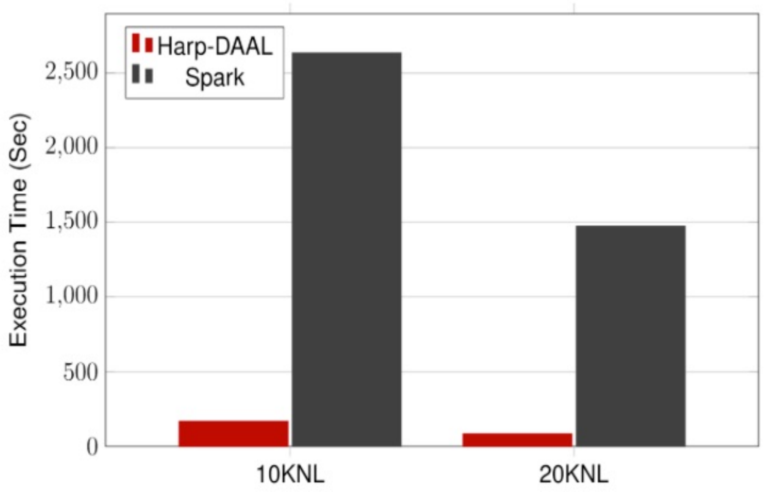


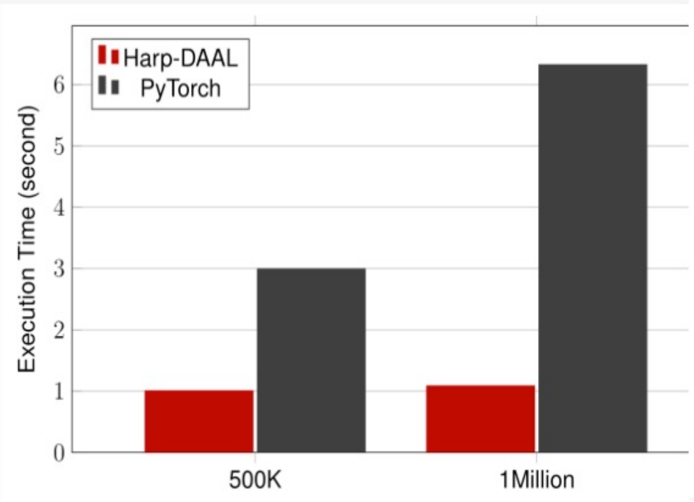
rotate

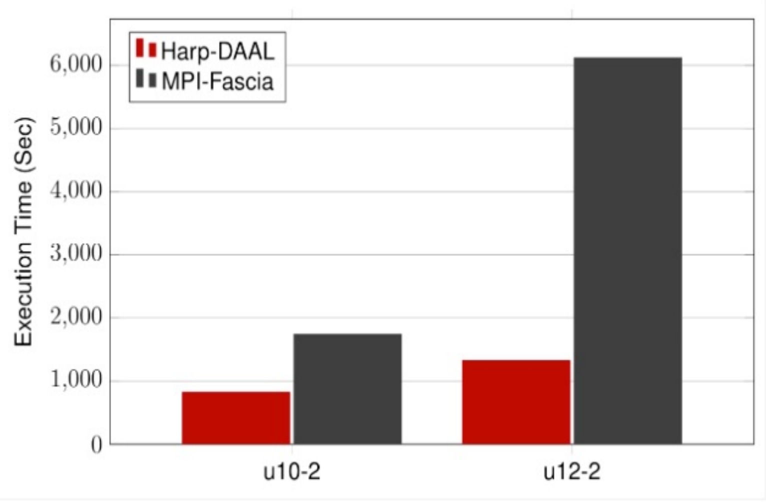
Map Collective Run time merges MapReduce and HPC

Harp v. Spark   Harp v. Torch   Harp v. MPI

Performance Comparison







K means

- Datasets: 5 million points, 10 thousand centroids, 10 feature dimensions
- 10 to 20 nodes of Intel KNL7250 processors
- Harp-DAAL has 15x speedups over Spark MLlib

PCA

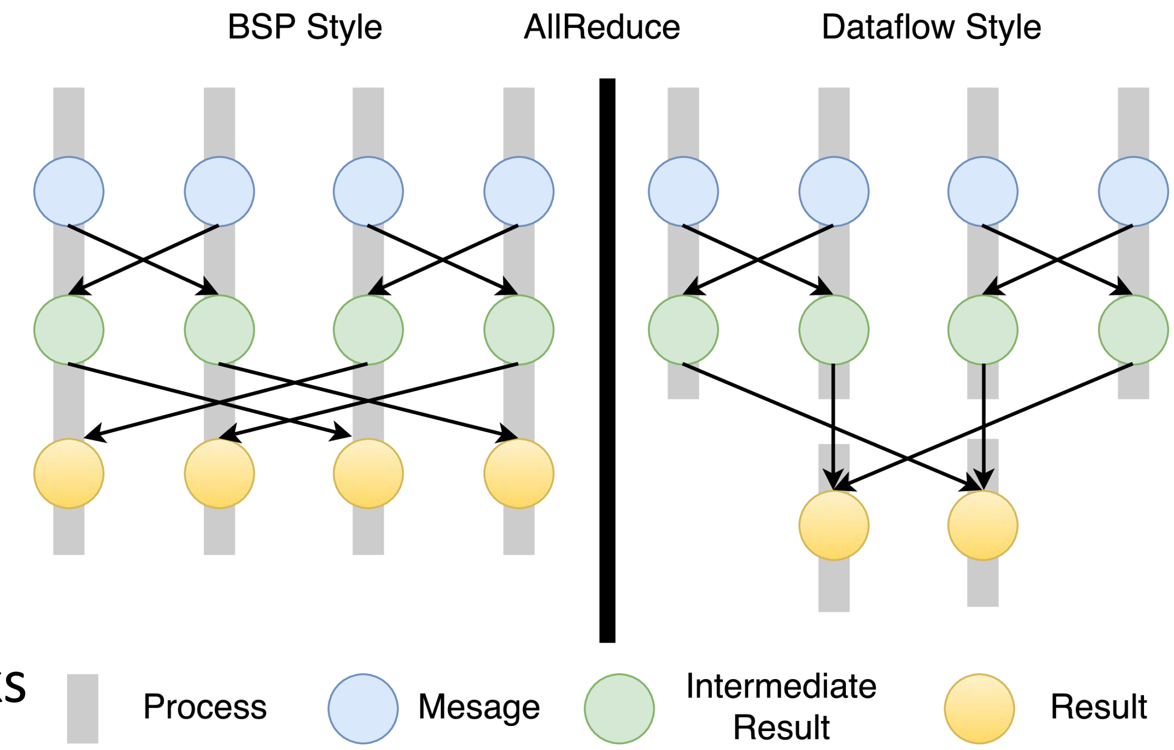
- Datasets: 500K or 1 million data points of feature dimension 300
- Running on single KNL 7250 (Harp-DAAL) vs. single K80 GPU (PyTorch)
- Harp-DAAL achieves 3x to 6x speedups

Subgraph

- Datasets: Twitter with 44 million vertices, 2 billion edges, subgraph templates of 10 to 12 vertices
- 25 nodes of Intel Xeon E5 2670
- Harp-DAAL has 2x to 5x speedups over state-of-the-art MPI-Fascia solution

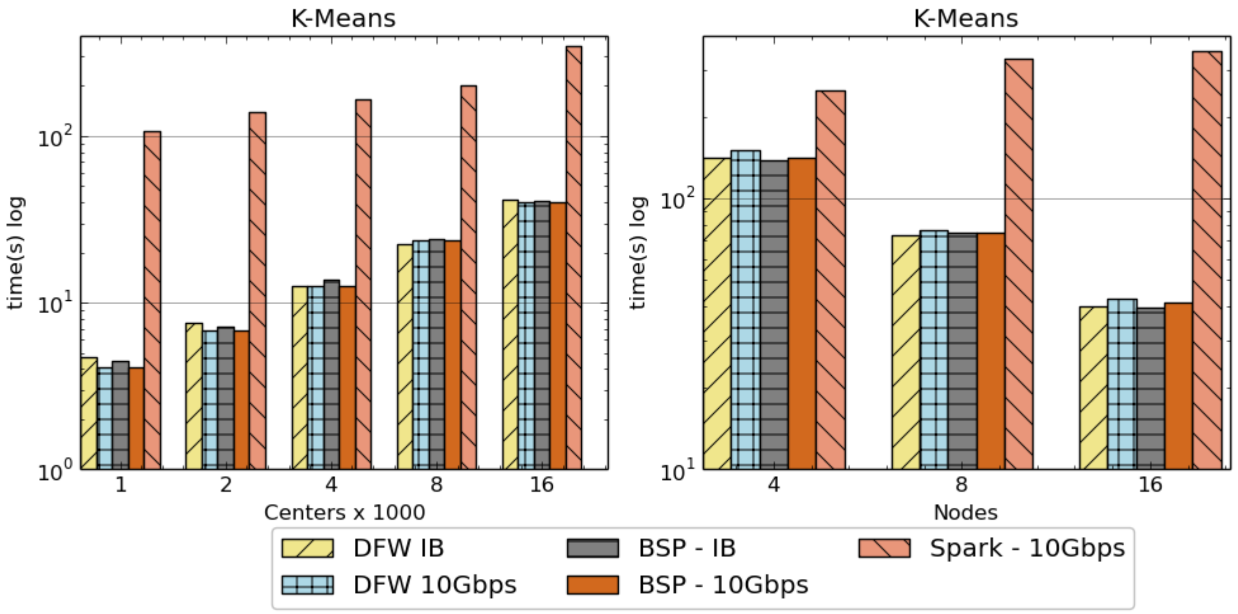
Twister2 Dataflow Communications

- **Twister:Net** offers two communication models
- **BSP** (Bulk Synchronous Processing) message-level communication using TCP or MPI separated from its task management plus extra Harp collectives
- **DFW** a new **Dataflow library** built using MPI software but at data movement not message level
  - Non-blocking
  - Dynamic data sizes
  - Streaming model
    - Batch case is modeled as a finite stream
  - The communications are between a set of tasks in an arbitrary task graph
  - Key based communications
  - Data-level Communications spilling to disks
  - Target tasks can be different from source tasks

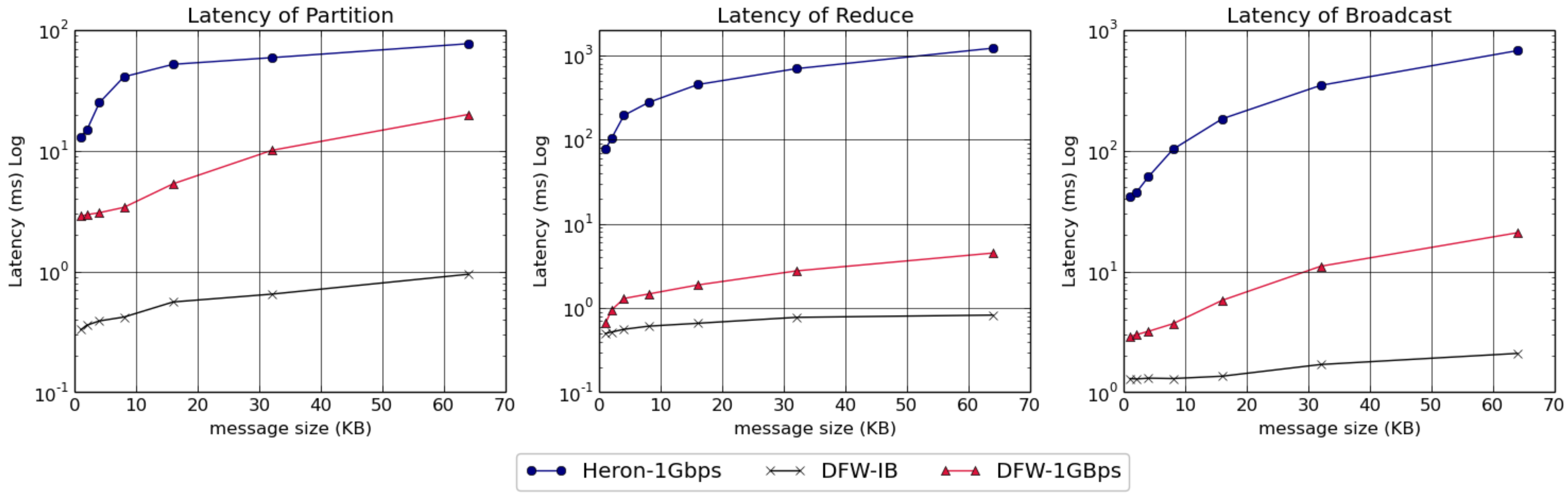


Twister:Net and Apache Heron and Spark

Left: K-means job execution time on 16 nodes with varying centers, 2 million points with 320-way parallelism. Right: K-Means wth 4,8 and 16 nodes where each node having 20 tasks. 2 million points with 16000 centers used.



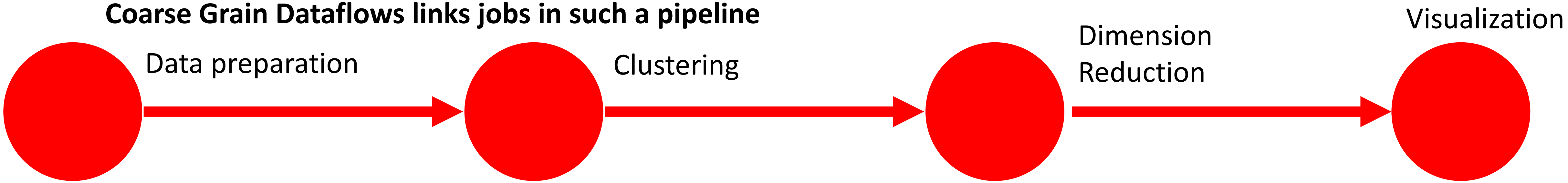
Latency of Apache Heron and Twister:Net DFW (Dataflow) for Reduce, Broadcast and Partition operations in 16 nodes with 256-way parallelism



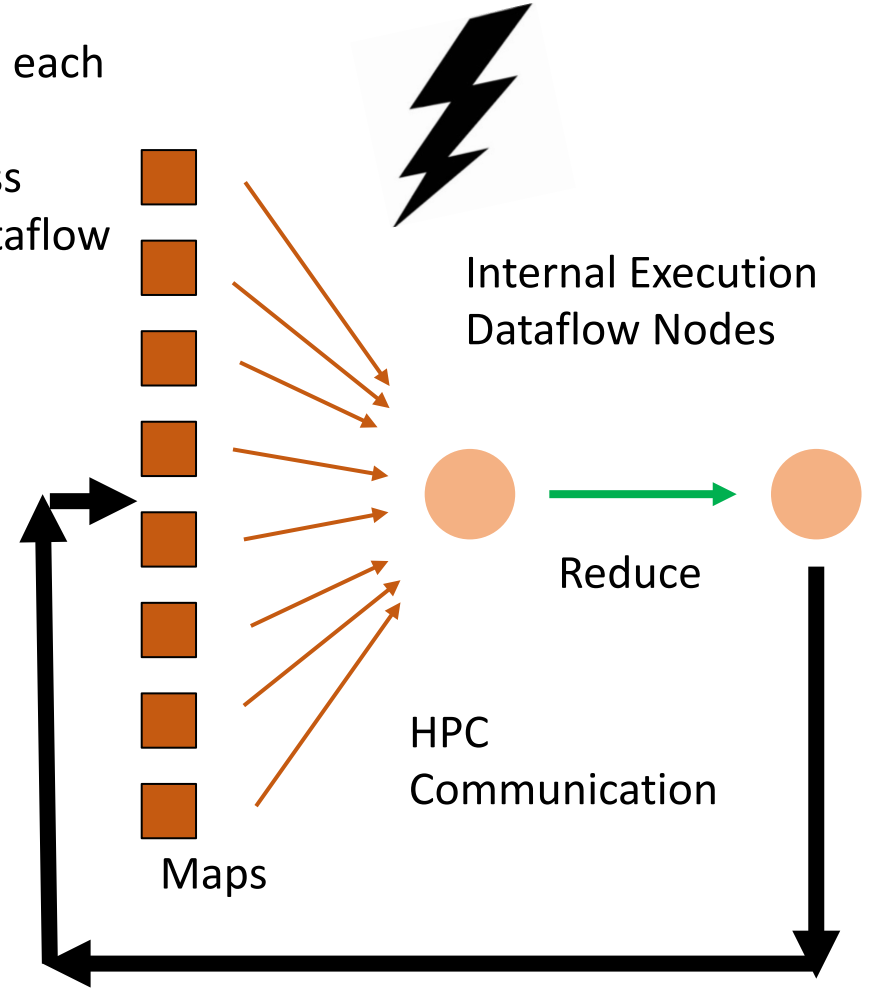
Intelligent Dataflow Graph

- The dataflow graph specifies the distribution and interconnection of job components
  - Hierarchical and Iterative
- Allow ML wrapping of component at each dataflow node
- Checkpoint **after each node of the dataflow graph**
  - Natural synchronization point
  - Let's allows user to choose when to checkpoint (not every stage)
  - Save state as user specifies; Spark just saves Model state which is insufficient for complex algorithms
- Intelligent nodes support customization of checkpointing, ML, communication
- Nodes can be coarse (large jobs) or fine grain requiring different actions

Coarse Grain Dataflows links jobs in such a pipeline



But internally to each job you can also elegantly express algorithm as dataflow but with more stringent performance constraints



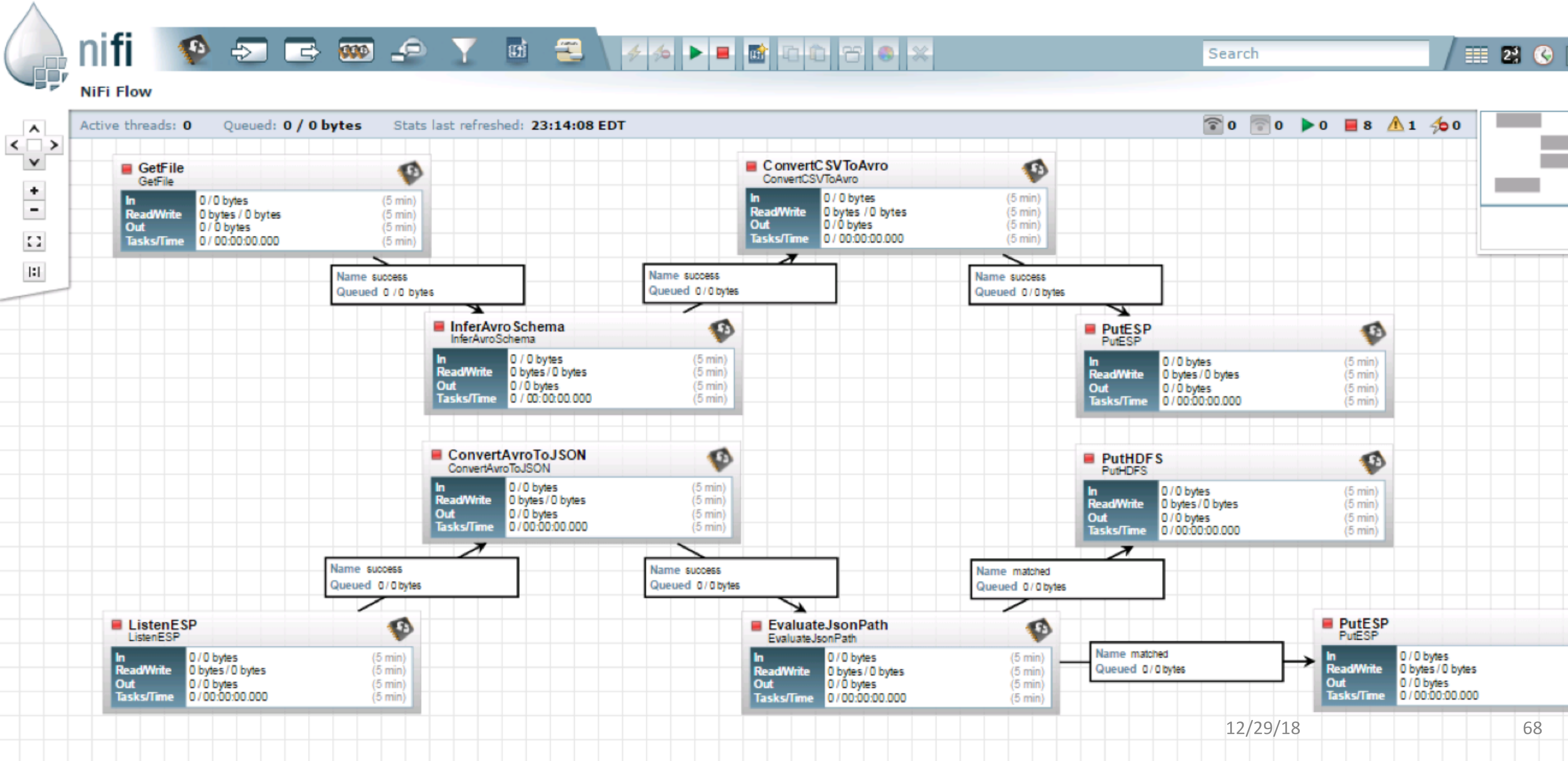
Corresponding to classic Spark K-means Dataflow

- P = loadPoints()
- C = loadInitCenters()
- for (int i = 0; i < 10; i++) {
- T = P.map().withBroadcast(C)
- C = T.reduce() }

Iterate

Dataflow at Different Grain sizes

NiFi Coarse-grain Workflow



Twister2

Iterative MapReduce

http://www.iterativemapreduce.org/

Futures
Implementing Twister2
for Global AI and Modeling Supercomputer

Twister2 Timeline: Current Release (End of September 2018)

- Twister:Net Dataflow Communication API
  - Dataflow communications with MPI or TCP
- Data access
  - Local File Systems
  - HDFS Integration
- Task Graph
- Streaming and Batch analytics – Iterative jobs
- Data pipelines
- Deployments on Docker, Kubernetes, Mesos (Aurora), Slurm
- Harp for Machine Learning (Custom BSP Communications)
  - Rich collectives
  - Around 30 ML algorithms

Twister2 Timeline: January 2018

- DataSet API similar to Spark batch and Heron streaming with Tset realization
  - Can use Tsets for writing RDD/Streamlet style datasets
- Fault tolerance as in Heron and Spark
- Storm API for Streaming
- Hierarchical Dynamic Heterogeneous Task Graph
  - Coarse grain and fine grain dataflow
- Cyclic task graph execution
- Dynamic scaling of resources and heterogeneous resources (at the resource layer) for streaming and heterogeneous workflow
- Link to Pilot Jobs

Twister2 Timeline: July 1, 2018

- Naiad model based Task system for Machine Learning
- Native MPI integration to Mesos, Yarn
- Dynamic task migrations
- RDMA and other communication enhancements
- Integrate parts of Twister2 components as big data systems enhancements (i.e. run current Big Data software invoking Twister2 components)
  - Heron (easiest), Spark, Flink, Hadoop (like Harp today)
  - Tsets become compatible with RDD (Spark) and Streamlet (Heron)
- Support different APIs (i.e. run Twister2 looking like current Big Data Software),Hadoop, Spark (Flink), Storm
- Refinements like Marathon with Mesos etc.
- Function as a Service and Serverless
- Support higher level abstractions
  - Twister:SQL (major Spark use case)
- Graph API

Conclusions

- Can make use case collections to motivate benchmarks
  - NIST and BDEC have templates
  - Could helpfully fill in templates for benchmarks
- Research applications have some similarities but many differences from commercial use cases
- Increasing importance of integration of simulation and Machine Learning
- Increasing importance of distributed Edge applications
- Should benchmark dataflow and BSP style communication
- Twister2 will combine Heron and Spark with built in HPC performance