Implications of NIST Big Data Application Classification for Benchmarking

IEEE SPEC Research Group on Big Data online meetings

See: http://hpc-abds.org/kaleidoscope/ for details

August 5 2015

Geoffrey Fox

gcf@indiana.edu

http://www.infomall.org, http://spidal.org/

School of Informatics and Computing

Digital Science Center

Indiana University Bloomington

NIST Big Data Initiative

Led by Chaitin Baru, Bob Marcus, Wo Chang

And

Big Data Application Analysis

NBD-PWG (NIST Big Data Public Working Group) Subgroups & Co-Chairs

- There were 5 Subgroups
  - Note mainly industry
- **Requirements and Use Cases Sub Group**
  - *Geoffrey Fox, Indiana U.; Joe Paiva, VA; Tsegereda Beyene, Cisco*
- **Definitions and Taxonomies SG**
  - *Nancy Grady, SAIC; Natasha Balac, SDSC; Eugene Luster, R2AD*
- **Reference Architecture Sub Group**
  - *Orit Levin, Microsoft; James Ketner, AT&T; Don Krapohl, Augmented Intelligence*
- **Security and Privacy Sub Group**
  - *Arnab Roy, CSA/Fujitsu Nancy Landreville, U. MD Akhil Manchanda, GE*
- **Technology Roadmap Sub Group**
  - *Carl Buffington, Vistronix; Dan McClary, Oracle; David Boyd, Data Tactics*
- See http://bigdatawg.nist.gov/usecases.php
- and http://bigdatawg.nist.gov/V1_output_docs.php

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

Use Case Template

- 26 fields completed for 51 apps
- **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**
- **Now an online form**

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

26 Features for each use case
Biased to science

- http://bigdatawg.nist.gov/usecases.php
- https://bigdatacoursespring2014.appspot.com/course (Section 5)
- **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

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,

http://hpc-abds.org/kaleidoscope/survey/

NIST BIG DATA PUBLIC WORKING GROU

BIG DATA USE CASE SURVEY

BETA - This survey was designed by the NIST Big Data Public Working Group to gather Big I use cases. To contact the group, refer to http://bigdatawg.nist.gov/home.php - v2.3.

* Required

SURVEY OUTLINE: Principal Big Data Use Case Details

- Overall Project Description
- Current Solutions
- Big Data Characteristics
- Big Data Science
- Overall Big Data Issues

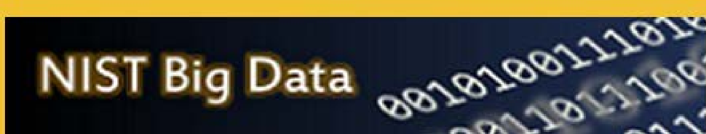
SURVEY OUTLINE: Other Use Case Information

- Tags to classify use case
- Workflow Processes (Basic, Extended)
- Abstract
- Author, Owner, Contact Info

SURVEY OUTLINE: Security and Privacy

- Roles
- PII
- Covenants, Liability
- Ownership, Identity, Distribution
- Risk Mitigation
- Provenance
- Data Life Cycle
- Audit and Traceability
- Application Provider Security
- Framework Provider Security
- System Health
- Permitted Use Cases

Online Use Case Form



Classify Use Cases with Tags

In next four groups, all questions are yes/no and generate tags we can use to classify use cases. See http://dsc.soic.indiana.edu/publications/OgrePaperv11.pdf (Towards an Understanding of Facets and Exemplars of Big Data Applications) for an example of how tags were used in initial 51 use cases

Data Tags - Application Style and Data sharing and acquisition

Check any number of items from this list.

- [ ] Uses Geographical Information Systems?
- [ ] Use case involves Internet of Things?
- [ ] Data comes from HPC or other simulations?
- [ ] Data Fusion important?
- [ ] Data is Real time Streaming?
- [ ] Data is Batched Streaming (e.g. collected remotely and uploaded every so often)?
- [ ] Important Data is in a Permanent Repository (Not streamed)?
- [ ] Transient Data important?
- [ ] Permanent Data Important?
- [ ] Data shared between different applications/users?
- [ ] Data largely dedicated to only this use case?

Data Tags - Management and Storage

Check any number of items from this list.

- [ ] Application data system based on Files
- [ ] Application data system based on Objects
- [ ] Uses HDFS style File System?
- [ ] Uses Wide area File System like Lustre?
- [ ] Uses HPC parallel file system like GPFS?
- [ ] Uses SQL?
- [ ] Uses NoSQL?
- [ ] Uses NewSQL?
- [ ] Uses Graph Database?

Are there other data acquisition/access/sharing/management/storage issues?

Specify in text box below.

Analytics Tags - Data Format and Nature of Algorithm used in Analytics

Principal Big Data Use Case Details

Overall questions about the use case covering all facets except security and privacy

Overall project description

http://hpc-abds.org/kaleidoscope/survey/

Use Case Title *

Your choice. Best if at most one line

Domain ("Vertical") *

What application area applies? There is no fixed ontology. See examples of existing use cases. "He Care" "Social Networking" "Financial" "Energy are examples

Actors / Stakeholders

Identify relevant stakeholder roles and responsibilities. (Note: Security and privacy roles are survey separate part of this survey.)

Project Goals or Objectives

Security and Privacy

Note that there are aspects of curation, provenance and governance that are not strictly speaking only security and privacy considerations. Refer to other sections of this survey for those aspects.

The S&P questions are grouped as follows:

- Roles
- Personally Identifiable Information
- Covenants and Liability
- Ownership, Distribution, Publication
- Risk Mitigation
- Audit and Traceability
- Data Life Cycle
- Dependencies
- Framework provider S&P
- Application Provider S&P
- Information Assurance | System Health
- Permitted Use CaSes

Security, privacy, provenance, governance, curation, and system health.

Roles

Roles may be associated with multiple functions within a big data ecosystem.

Investigator, Lead Analyst, Lead Scientists, Project Leader, Mgr Product Dev, VP Engineering

Identify the role associated with identifying the use case need, requirements and deployment

Investigator Affiliations

Features and Examples

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

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 (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

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

Local and Global Machine Learning

- **Many applications** use **LML or Local machine Learning** where machine learning (often from R) is run separately on every data item such as on every image
- **But others** are **GML** Global Machine Learning where machine learning is a single 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).
  - **Graph analytics** is typically GML
- Covering **clustering**/community detection, mixture models, topic determination, Multidimensional scaling, (**Deep**) **Learning Networks**
- **PageRank** is "just" parallel linear algebra
- Note many **Mahout** algorithms are sequential – partly as MapReduce limited; partly because parallelism unclear
  - **MLLib** (Spark based) better
- **SVM** and **Hidden Markov Models** do not use large scale parallelization in practice?

13 Image-based Use Cases

- **13-15 Military Sensor Data Analysis/ Intelligence PP, LML, GIS, MR**
- **7:Pathology Imaging/ Digital Pathology: PP, LML, MR** for search becoming terabyte 3D images, Global Classification
- **18&35: Computational Bioimaging (Light Sources): PP, LML** Also materials
- **26: Large-scale Deep Learning: GML** Stanford ran 10 million images and 11 billion parameters on a 64 GPU HPC; vision (drive car), speech, and Natural Language Processing
- **27: Organizing large-scale, unstructured collections of photos: GML** Fit position and camera direction to assemble 3D photo ensemble
- **36: Catalina Real-Time Transient Synoptic Sky Survey (CRTS): PP, LML** followed by classification of events (**GML**)
- **43: Radar Data Analysis for CReSIS Remote Sensing of Ice Sheets: PP, LML** to identify glacier beds; **GML** for full ice-sheet
- **44: UAVSAR Data Processing, Data Product Delivery, and Data Services**: **PP** to find slippage from radar images
- **45, 46: Analysis of Simulation visualizations: PP LML ?GML** find paths, classify orbits, classify patterns that signal earthquakes, instabilities, climate, turbulence

Internet of Things and Streaming Apps

- It is projected that there will be **24 (Mobile Industry Group) to 50 (Cisco) billion devices** on the Internet by 2020.
- The **cloud** natural controller of and **resource provider for the Internet of Things.**
- Smart phones/watches, Wearable devices (Smart People), “Intelligent River” “Smart Homes and Grid” and “Ubiquitous Cities”, Robotics.
- Majority of use cases are streaming – experimental science gathers data in a stream – sometimes batched as in a field trip. Below is sample
- **10: Cargo Shipping Tracking** as in UPS, Fedex **PP GIS LML**
- **13: Large Scale Geospatial Analysis and Visualization PP GIS LML**
- **28: Truthy: Information diffusion research from Twitter Data PP MR** for Search, **GML** for community determination
- **39: Particle Physics: Analysis of LHC Large Hadron Collider Data: Discovery of Higgs particle PP** for event Processing, Global statistics
- **50: DOE-BER AmeriFlux and FLUXNET Networks PP GIS LML**
- **51: Consumption forecasting in Smart Grids PP GIS LML**

Big Data Patterns – the Ogres Benchmarking

Classifying Applications and Benchmarks

- "Benchmarks" "kernels" "algorithm" "mini-apps" can serve multiple purposes
- Motivate hardware and software features
  - e.g. collaborative filtering algorithm parallelizes well with MapReduce and suggests using Hadoop on a cloud
  - e.g. deep learning on images dominated by matrix operations; needs CUDA&MPI and suggests HPC cluster
- Benchmark sets designed cover key features of systems in terms of features and sizes of "important" applications
- Take 51 uses cases → derive specific features; each use case has multiple features
- Generalize and systematize as Ogres with features termed "facets"
- 50 Facets divided into 4 sets or views where each view has "similar" facets

7 Computational Giants of NRC Massive Data Analysis Report

http://www.nap.edu/catalog.php?record_id=18374

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 Benchmark Classics

- **Linpack** or HPL: Parallel LU factorization for solution of linear equations
- **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

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.
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!

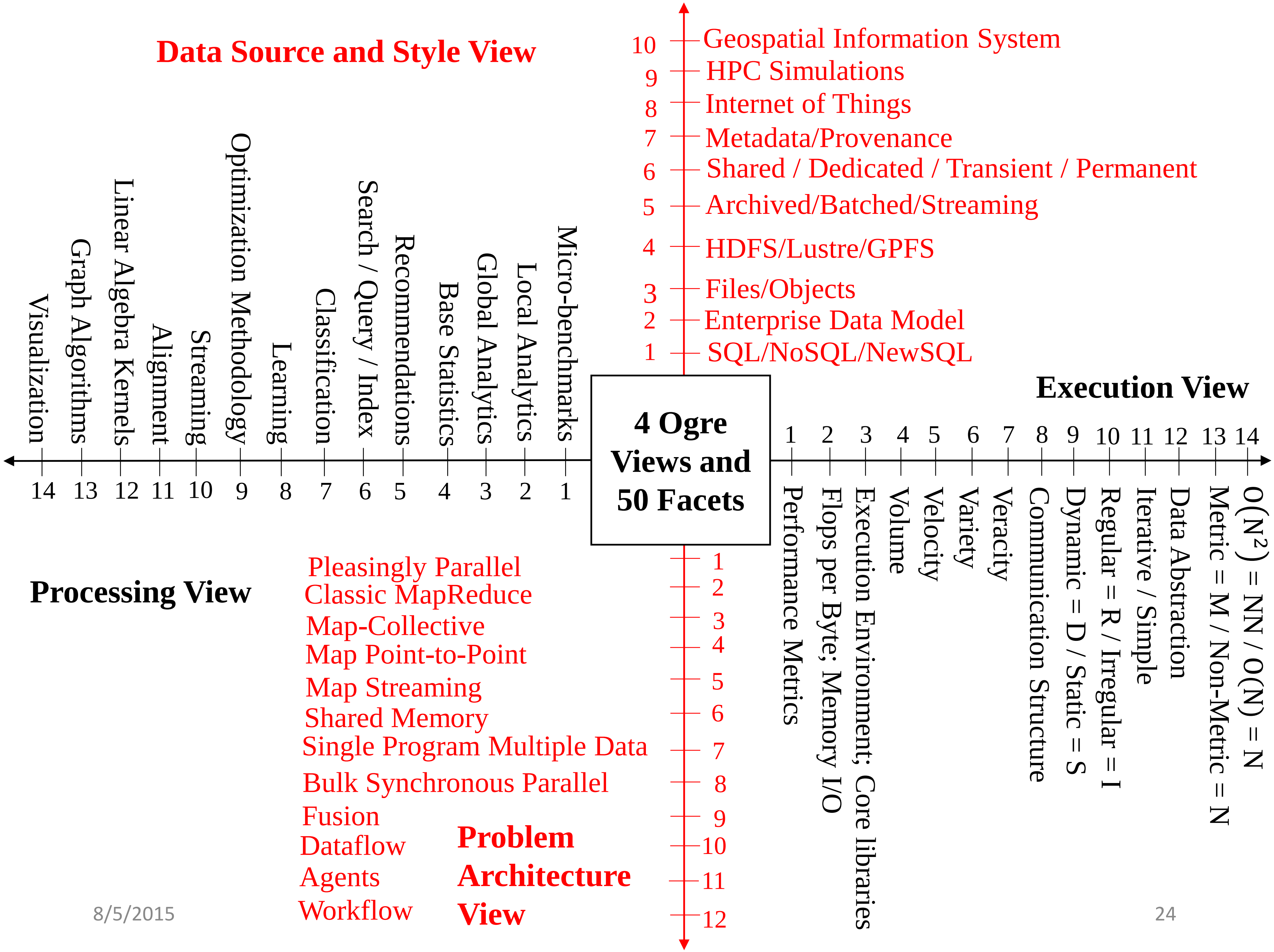
Facets of the Ogres

Introducing Big Data Ogres and their Facets I

- **Big Data Ogres** provide a systematic approach to understanding applications, and as such they have **facets** which represent key characteristics defined both from our experience and from a bottom-up study of features from several individual applications.
- The facets capture common characteristics (shared by several problems)which are inevitably multi-dimensional and often overlapping.
- Ogres characteristics are cataloged in four distinct dimensions or views.
- Each view consists of facets; when multiple facets are linked together, they describe classes of big data problems represented as an Ogre.
- Instances of Ogres are particular big data problems
- A set of Ogre instances that cover a rich set of facets could form a benchmark set
- Ogres and their instances can be atomic or composite

Introducing Big Data Ogres and their Facets II

- Ogres characteristics are cataloged in four distinct dimensions or views.
- Each view consists of facets; when multiple facets are linked together, they describe classes of big data problems represented as an Ogre.
- One view of an Ogre is the overall **problem architecture** which is naturally related to the machine architecture needed to support data intensive application while still being different.
- Then there is the **execution (computational) features** view, describing issues such as I/O versus compute rates, iterative nature of computation and the classic V's of Big Data: defining problem size, rate of change, etc.
- The **data source & style** view includes facets specifying how the data is collected, stored and accessed.
- The final **processing** view has facets which describe classes of processing steps including algorithms and kernels. Ogres are specified by the particular value of a set of facets linked from the different views.


Data Source and Style View
10 Geospatial Information System
9 HPC Simulations
8 Internet of Things
7 Metadata/Provenance
6 Shared / Dedicated / Transient / Permanent
5 Archived/Batched/Streaming
4 HDFS/Lustre/GPFS
3 Files/Objects
2 Enterprise Data Model
1 SQL/NoSQL/NewSQL
Execution View
1 Performance Metrics
2 Flops per Byte; Memory I/O
3 Execution Environment; Core libraries
4 Volume
5 Velocity
6 Variety
7 Veracity
8 Communication Structure
9 Dynamic = D / Static = S
10 Regular = R / Irregular = I
11 Iterative / Simple
12 Data Abstraction
13 Metric = M / Non-Metric = N
14 $O(N^2)$ = NN / $O(N)$ = N
4 Ogre Views and 50 Facets
Processing View
1 Micro-benchmarks
2 Local Analytics
3 Global Analytics
4 Base Statistics
5 Recommendations
6 Search / Query / Index
7 Classification
8 Learning
9 Optimization Methodology
10 Streaming
11 Alignment
12 Linear Algebra Kernels
13 Graph Algorithms
14 Visualization
Problem Architecture View
1 Pleasingly Parallel
2 Classic MapReduce
3 Map-Collective
4 Map Point-to-Point
5 Map Streaming
6 Shared Memory
7 Single Program Multiple Data
8 Bulk Synchronous Parallel
9 Fusion
10 Dataflow
11 Agents
12 Workflow

Facets of the Ogres

Problem Architecture

Meta or Macro Aspects of Ogres

Problem Architecture View of Ogres (Meta or MacroPatterns)

i. **Pleasingly Parallel** – as in BLAST, Protein docking, some (bio-)imagery including **Local Analytics or Machine Learning** – ML or filtering pleasingly parallel, as in bio-imagery, radar images (pleasingly parallel but sophisticated local analytics)
ii. **Classic MapReduce:** Search, Index and Query and Classification algorithms like collaborative filtering (G1 for MRStat in Features, G7)
iii. **Map-Collective:** Iterative maps + communication dominated by "collective" operations as in reduction, broadcast, gather, scatter. Common datamining pattern
iv. **Map-Point to Point:** Iterative maps + communication dominated by many small point to point messages as in graph algorithms
v. **Map-Streaming:** Describes streaming, steering and assimilation problems
vi. **Shared Memory:** Some problems are asynchronous and are easier to parallelize on shared rather than distributed memory – see some graph algorithms
vii. **SPMD:** Single Program Multiple Data, common parallel programming feature
viii. **BSP or Bulk Synchronous Processing:** well-defined compute-communication phases
ix. **Fusion:** Knowledge discovery often involves fusion of multiple methods.
x. **Dataflow:** Important application features often occurring in composite Ogres
xi. **Use Agents:** as in epidemiology (swarm approaches)
xii. **Workflow:** All applications often involve orchestration (workflow) of multiple components

Relation of Problem and Machine Architecture

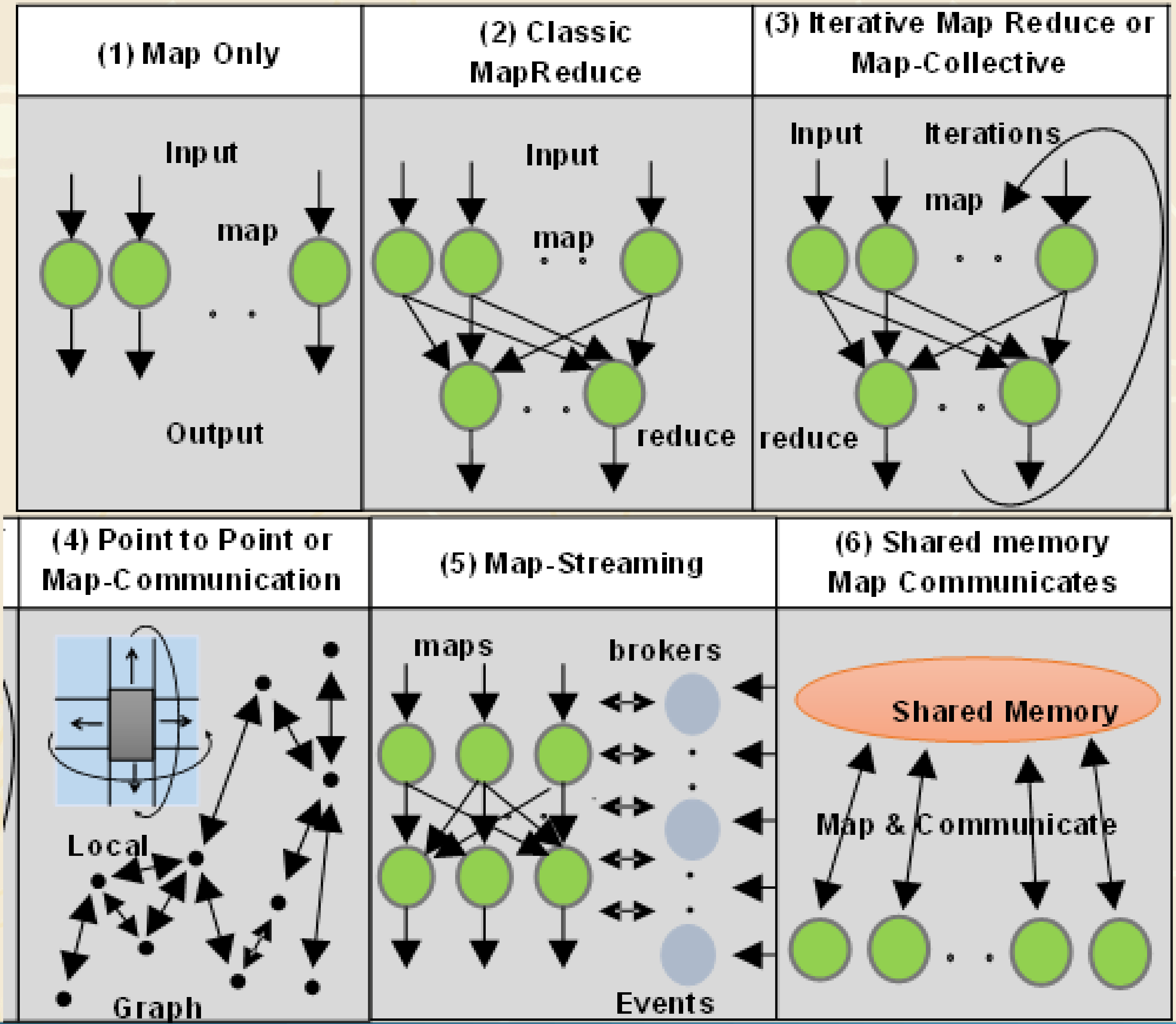
- In my old papers (especially book Parallel Computing Works!), I discussed computing as multiple complex systems mapped into each other

Problem → Numerical formulation → Software → Hardware

- Each of these 4 systems has an architecture that can be described in similar language
- One gets an easy programming model if architecture of problem matches that of Software
- One gets good performance if architecture of hardware matches that of software and problem
- So "MapReduce" can be used as architecture of software (programming model) or "Numerical formulation of problem"

6 Forms of MapReduce
cover "all" circumstances

Also an interesting software (architecture) discussion



Ogre Facets

Execution Features View

One View of Ogres has Facets that are micropatterns or Execution Features

i. **Performance Metrics;** property found by benchmarking Ogre
ii. **Flops per byte;** memory or I/O
iii. **Execution Environment; Core libraries needed:** matrix-matrix/vector algebra, conjugate gradient, reduction, broadcast; Cloud, HPC etc.
iv. **Volume:** property of an Ogre instance
v. **Velocity:** qualitative property of Ogre with value associated with instance
vi. **Variety:** important property especially of composite Ogres
vii. **Veracity:** important property of "mini-applications" but not kernels
viii. **Communication Structure;** Interconnect requirements; Is communication BSP, Asynchronous, Pub-Sub, Collective, Point to Point?
ix. Is application (graph) **static** or **dynamic?**
x. Most applications consist of a set of interconnected entities; is this **regular** as a set of pixels or is it a complicated **irregular graph?**
xi. Are algorithms **Iterative** or **not?**
xii. **Data Abstraction:** key-value, pixel, graph(G3), vector, bags of words or items
xiii. Are data points in **metric or non-metric spaces?**
xiv. Is algorithm **$O(N^2)$ or $O(N)$** (up to logs) for N points per iteration (G2)

Facets of the Ogres

Data Source and Style Aspects

Data Source and Style View of Ogres I

i. **SQL NewSQL or NoSQL:** NoSQL includes Document, Column, Key-value, Graph, Triple store; NewSQL is SQL redone to exploit NoSQL performance

ii. Other **Enterprise data systems:** 10 examples from NIST integrate SQL/NoSQL

iii. **Set of Files or Objects:** as managed in iRODS and extremely common in scientific research

iv. **File systems, Object, Blob and Data-parallel** (HDFS) raw storage: Separated from computing or colocated? HDFS v Lustre v. Openstack Swift v. GPFS

v. **Archive/Batched/Streaming:** Streaming is incremental update of datasets with new algorithms to achieve real-time response (G7); Before data gets to compute system, there is often an initial data gathering phase which is characterized by a block size and timing. Block size varies from month (Remote Sensing, Seismic) to day (genomic) to seconds or lower (Real time control, streaming)

Data Source and Style View of Ogres II

vi. **Shared/Dedicated/Transient/Permanent:** qualitative property of data; Other characteristics are needed for permanent auxiliary/comparison datasets and these could be interdisciplinary, implying nontrivial data movement/replication

vii. **Metadata/Provenance:** Clear qualitative property but not for kernels as important aspect of data collection process

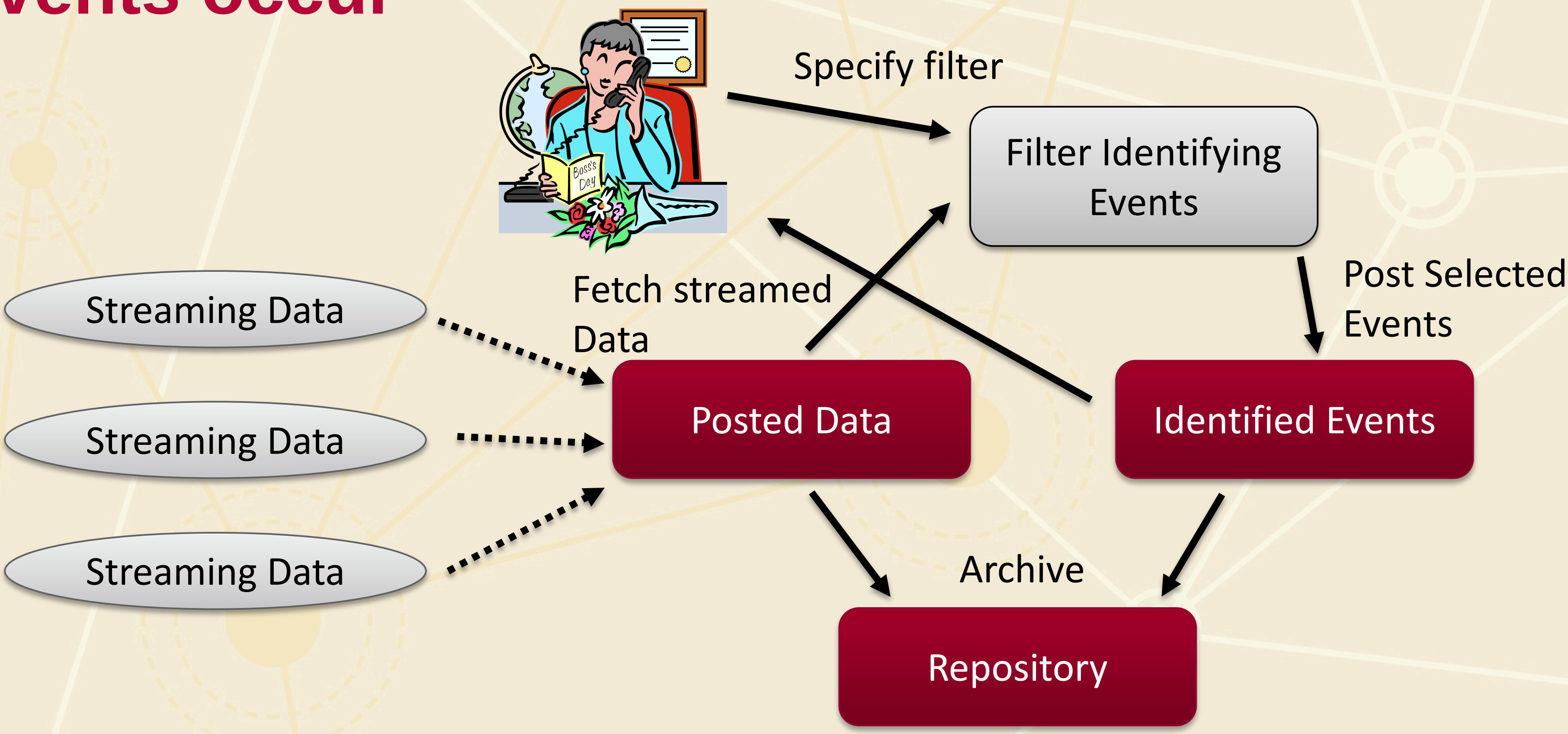
viii. **Internet of Things:** 24 to 50 Billion devices on Internet by 2020

ix. **HPC simulations:** generate major (visualization) output that often needs to be mined

x. Using **GIS:** Geographical Information Systems provide attractive access to geospatial data

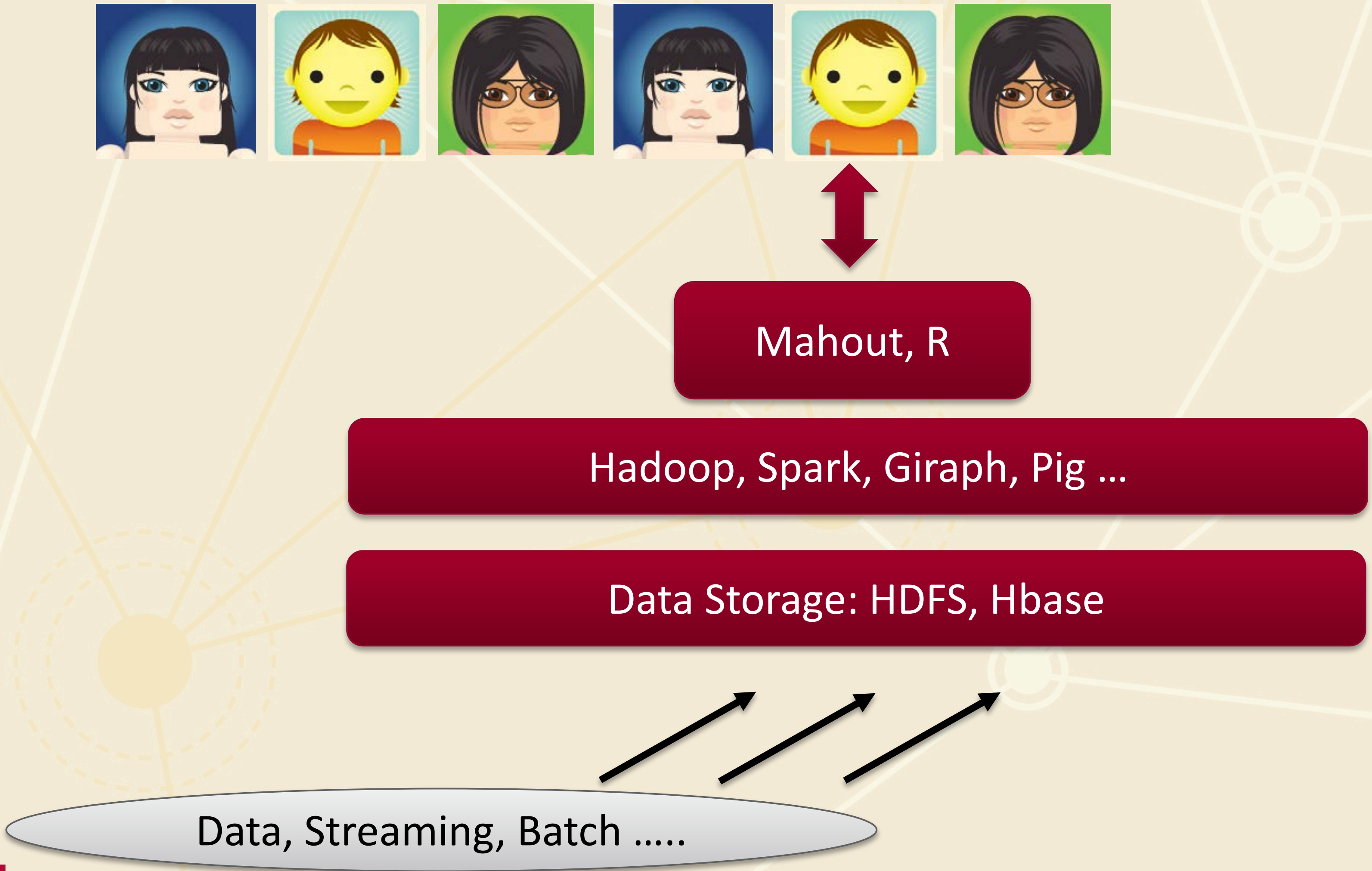
Note 10 Bob Marcus (led NIST effort) Use cases

2. Perform real time analytics on data source streams and notify users when specified events occur



Storm, Kafka, Hbase, Zookeeper

5. Perform interactive analytics on data in analytics-optimized database



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

Local Accumulate and initial computing

Record Scientific Data in “field”

NIST examples include LHC, Remote Sensing, Astronomy and Bioinformatics

Facets of the Ogres

Processing View

Facets in Processing (runtime) View of Ogres I

i. **Micro-benchmarks** ogres that exercise simple features of hardware such as communication, disk I/O, CPU, memory performance
ii. **Local Analytics** executed on a single core or perhaps node
iii. **Global Analytics** requiring iterative programming models (G5,G6) across multiple nodes of a parallel system
iv. **Optimization Methodology:** overlapping categories
    i. **Nonlinear Optimization (G6)**
    ii. **Machine Learning**
    iii. **Maximum Likelihood** or χ^2 minimizations
    iv. **Expectation Maximization** (often Steepest descent)
    v. **Combinatorial Optimization**
    vi. **Linear/Quadratic Programming (G5)**
    vii. **Dynamic Programming**
v. **Visualization** is key application capability with algorithms like MDS useful but it itself part of “mini-app” or composite Ogre
vi. **Alignment (G7)** as in BLAST compares samples with repository

Facets in Processing (run time) View of Ogres II

vii. **Streaming divided into 5 categories depending on event size and synchronization and integration**
  - Set of independent events where precise time sequencing unimportant.
  - Time series of connected small events where time ordering important.
  - Set of independent large events where each event needs parallel processing with time sequencing not critical
  - Set of connected large events where each event needs parallel processing with time sequencing critical.
  - Stream of connected small or large events to be integrated in a complex way.

viii. **Basic Statistics (G1):** MRStat in NIST problem features

ix. **Search/Query/Index:** Classic database which is well studied (Baru, Rabl tutorial)

x. **Recommender Engine:** core to many e-commerce, media businesses; collaborative filtering key technology

xi. **Classification:** assigning items to categories based on many methods
  - MapReduce good in Alignment, Basic statistics, S/Q/I, Recommender, Calssification

xii. **Deep Learning** of growing importance due to success in speech recognition etc.

xiii. **Problem set up as a graph** (G3) as opposed to vector, grid, bag of words etc.

xiv. Using **Linear Algebra Kernels**: much machine learning uses linear algebra kernels

Benchmarks and Ogres

Benchmarks/Mini-apps spanning Facets

- **Look at** NSF SPIDAL Project, NIST 51 use cases, Baru-Rabl review
- **Catalog facets** of benchmarks and choose entries to cover "all facets"
- **Micro Benchmarks:** SPEC, EnhancedDFSIO (HDFS), Terasort, Wordcount, Grep, MPI, Basic Pub-Sub ….
- **SQL and NoSQL Data systems, Search, Recommenders:** TPC (-C to x–HS for Hadoop), BigBench, Yahoo Cloud Serving, Berkeley Big Data, HiBench, BigDataBench, Cloudsuite, Linkbench
  - includes MapReduce cases Search, Bayes, Random Forests, Collaborative Filtering
- **Spatial Query:** select from image or earth data
- **Alignment:** Biology as in BLAST
- **Streaming:** Online classifiers, Cluster tweets, Robotics, Industrial Internet of Things, Astronomy; BGBenchmark; choose to cover all 5 subclasses
- **Pleasingly parallel (Local Analytics):** as in initial steps of LHC, Pathology, Bioimaging (differ in type of data analysis)
- **Global Analytics:** Outlier, Clustering, LDA, SVM, Deep Learning, MDS, PageRank, Levenberg-Marquardt, Graph 500 entries
- **Workflow** and **Composite** (analytics on xSQL) linking above

Facets for some General (Core) Machine Learning

Algorithm	Applications	Problem Arch View	Execution View	Processing View
DA Vector Clustering	Accurate Clusters	3, 7, 8	9D, 10I, 11, 12V, 13M, 14N	9ML, 9EM, 12
DA Non-metric Clustering	Accurate Clusters, Biology, Web	3, 7, 8	9S, 10R, 11, 12BI, 13N, 14NN	9ML, 9EM, 12
Kmeans; Basic, Fuzzy and Elkan	Fast Clustering	3, 7, 8	9D, 10I(Elkan), 11, 12V, 13M, 14N	9ML, 9EM
Levenberg-Marquardt Optimization	Non-linear Gauss-Newton, use in MDS	3, 7, 8	9D, 10R, 11, 12V, 14NN	9ML, 9NO, 9LS, 9EM, 12
DA, Weighted SMACOF	MDS with general weights	3, 7, 8	9S, 10R, 11, 12BI, 13N, 14NN	9ML, 9NO, 9LS, 9EM, 12, 14
TFIDF Search	Find nearest neighbors in document corpus	1	9S, 10R, 12BI, 13N, 14N	2, 9ML
All-pairs similarity search	Find pairs of documents with TFIDF distance below a threshold	3, 7, 8	9S, 10R, 12BI, 13N, 14NN	9ML
Support Vector Machine SVM	Learn and Classify	3, 7, 8	9S, 10R, 11, 12V, 13M, 14N	7, 8, 9ML
Random Forest	Learn and Classify	1	9S, 10R, 12BI, 13N, 14N	2, 7, 8, 9ML
Gibbs sampling (MCMC)	Solve global inference problems	3, 7, 8	9S, 10R, 11, 12BW, 13N, 14N	9ML, 9NO, 9EM
Latent Dirichlet Allocation LDA with Gibbs sampling or Var. Bayes	Topic models (Latent factors)	3, 7, 8	9S, 10R, 11, 12BW, 13N, 14N	9ML, 9EM
Singular Value Decomposition SVD	Dimension Reduction and PCA	3, 7, 8	9S, 10R, 11, 12V, 13M, 14NN	9ML, 12
Hidden Markov Models (HMM)	Global inference on sequence models	3, 7, 8	9S, 10R, 11, 12BI	2, 9ML, 12

	Facet and View	Comments	SP	DB	NI
		Facets in Problem Architecture View (AV)			
1	Pleasingly Parallel	Clear qualitative property overlapping Local Analytics	M	S	H
2	Classic MapReduce	Clear qualitative property of non-iterative algorithms	M	H	H
3	Map-Collective	Clear qualitative property of much machine learning	H	S	H
4	Map Point-to-Point (graphs)	Clear qualitative property of graphs and simulation	H	S	M
5	Map Streaming	Property of growing importance. Not well benchmarked	N	N	H
6	Shared memory (as opposed to distributed parallel algorithm)	Corresponds to problem where shared memory implementations important. Tend to be dynamic asynchronous	S	N	S
7	Single Program Multiple Data SPMD	Clear qualitative property famous in parallel computing	H	M	H
8	Bulk Synchronous Processing BSP	Needs to be defined but reasonable qualitative property	H	M	H
9	Fusion	Only present for composite Ogres	N	N	H
10	Dataflow	Only present for composite Ogres	N	N	H
11	Agents	Clear but uncommon qualitative property	N	N	S
12	Orchestration (workflow)	Only present for composite Ogres	N	H	H

Facet and View		Comments	SP	DB	NI
		Facets in Execution View (EV)			
1	Performance Metrics	Result of Benchmark	-	-	-
2	Flops per Byte (Memory or I/O). Flops per watt (power).	I/O Not needed for “pure in memory” benchmark. Value needs detailed quantitative study. Could depend on implementation	-	-	-
3	Execution Environment (LN = Libraries needed, C= Cloud, HPC = HPC, T=Threads, MP= Message Passing)	Depends on how benchmark set up. Could include details of machine used for benchmarking here	-	-	-
4	Volume	Depends on data size. Benchmark measure	-	M	-
5	Velocity	Associated with streaming facet but value depends on particular problem	N	S	H
6	Variety	Most useful for composite Ogres	N	S	H
7	Veracity	Most problems would not discuss but potentially important	N	N	M
8	Communication Structure (D=Distributed, I=Interconnect, S=Synchronization)	Qualitative property – related to BSP (Bulk Synchronous Processing) and Shared memory	U	U	U
9	D=Dynamic or S=Static	Clear qualitative properties. Importance familiar from parallel computing	H	H	H
10	R=Regular or I=Irregular		H	H	H
11	Iterative?	Clear qualitative property. Highlighted by Iterative MapReduce and always present in classic parallel computing	H	S	H
12	Data Abstraction(K= key-value, BW= bag of words, BI = bag of items, P= pixel/spatial, V= vectors/matrices, S= sequence, G= graph)	Clear quantitative property although important data abstractions not agreed upon. All should be supported by Programming model and run time	H	M	H
13	M= Metric Space or N= not?	Clear qualitative property discussed in [69]	H	N	H
14	NN= O(N2) or N= O(N)?	Clear qualitative property highlighted in [2]	H	N	H

	Facet and View	Comments	SP	DB	NI
Facets in Data Source & Style View (DV)					
1	SQL/NoSQL/NewSQL?	Clear qualitative property. Can add NoSQL sub-categories such as key-value, graph, document …	N	H	H
2	Enterprise data model (warehouses)	Clear qualitative property of data model highlighted in database community / industry benchmarks	N	H	M
3	Files/Objects?	Clear qualitative property of data model where files important in Science; objects in industry	N	S	H
4	HDFS/Lustre/GPFS?	Clear qualitative property where HDFS important in Apache stack but not much used in science	N	H	H
5	Archive/Batched/Streaming	Clear qualitative property but not for kernels as it describes how data is collected	N	N	H
6	Shared/Dedicated/Transient/Permanent	Clear qualitative property of data whose importance is not well studied	N	N	H
7	Metadata/Provenance	Clear qualitative property but not for kernels as important aspect of data collection process	N	N	H
8	Internet of Things	Dominant source of commodity data in future	N	N	H
9	HPC Simulations	Important in science research especially at exascale	N	N	H
10	Geographic Information Systems	Clear property but not for kernels	S	N	H

Facet and View		Comments	SP	DB	NI
Facets in Processing View (PV)					
1	Micro-benchmarks	Important subset of small kernels	N	H	N
2	Local Analytics or Informatics	Well defined but overlaps Pleasingly Parallel	H	H	H
3	Global Analytics or Informatics	Clear qualitative property that includes parallel Mahout (E.g. Kmeans) and Hive (database)	H	H	H
4	Base Statistics	Describes simple statistical averages needing simple MapReduce. MRStat in [6]	N	N	M
5	Recommender Engine	Clear type of machine learning of especial importance commercially	N	M	H
6	Search/Query/Index	Clear important class of algorithms in industry	S	H	H
7	Classification	Clear important class of algorithms	S	M	H
8	Learning	Includes deep learning as category	S	S	H
9	Optimization Methodology (ML= Machine Learning, NO = Nonlinear Optimization, LS = Least Squares, EM = expectation maximization, LQP = Linear/Quadratic Programming, CO = Combinatorial Optimization)	LQP and CO overshadowed by machine learning but important where used. ML includes many analytics which are often NO and EM and sometimes LS (or similar Maximum Likelihood)	H	M	H
10	Streaming	Clear important class of algorithms associated with Internet of Things. Can be called DDDAS Dynamic Data-Driven Application Systems	N	N	H
11	Alignment	Clear important class of algorithms as in BLAST	N	S	M
12	Linear Algebra Kernels	Important property of some analytics	H	S	H
13	Graph Algorithms	Clear important class of algorithms – often hard	H	M	M
14	Visualization	Clearly important aspect of data analysis but different in character to most other facets	S	N	H

Conclusions

- Collected 51 use cases; useful although certainly incomplete and biased (to research and against energy for example)
- Improved (especially in security and privacy) and available as online form
- Identified 50 features called facets divided into 4 sets (views) used to classify applications
- Used to derive set of hardware architectures
  - Could discuss software (se papers)
- Surveyed some benchmarks
- Could be used to identify missing benchmarks
  - Noted streaming a dominant feature of use cases but not common in benchmarks

Spare Slides

8 Data Analysis Problem Architectures

- 1) Pleasingly Parallel PP or “map-only” in MapReduce
  - BLAST Analysis; Local Machine Learning
- 2A) Classic MapReduce MR, Map followed by reduction
  - High Energy Physics (HEP) Histograms; Web search; Recommender Engines
- 2B) Simple version of classic MapReduce MRStat
  - Final reduction is just simple statistics
- 3) Iterative MapReduce MRIter
  - Expectation maximization Clustering Linear Algebra, PageRank
- 4A) Map Point to Point Communication
  - Classic MPI; PDE Solvers and Particle Dynamics; Graph processing Graph
- 4B) GPU (Accelerator) enhanced 4A) – especially for deep learning
- 5) Map + Streaming + Communication
  - Images from Synchrotron sources; Telescopes; Internet of Things IoT
- 6) Shared memory allowing parallel threads which are tricky to program but lower latency
  - Difficult to parallelize asynchronous parallel Graph Algorithms

http://hpc-abds.org/kaleidoscope/

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

Cross-Cutting Functions

1) **Message and Data Protocols:** Avro, Thrift, Protobuf

2) **Distributed Coordination:** Google Chubby, Zookeeper, Giraffe, JGroups

3) **Security & Privacy:** InCommon, Eduroam OpenStack Keystone, LDAP, Sentry, Sqrrl, OpenID, SAML OAuth

4) **Monitoring:** Ambari, Ganglia, Nagios, Inca

21 layers
Over 300 Software Packages

May 2 2015

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

16) **Application and Analytics:** Mahout , MLlib , MLbase, DataFu, R, pbdR, Bioconductor, ImageJ, OpenCV, Scalapack, PetSc, 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, MapGraph, IBM System G, GraphBuilder(Intel), TinkerPop, Google Fusion Tables, CINET, NWB, Elasticsearch, Kibana, Logstash, Graylog, Splunk, Tableau, D3.js, three.js, Potree

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, Google MillWheel, Amazon Kinesis, LinkedIn Databus, Facebook Puma/Ptail/Scribe/ODS, Azure Stream Analytics

14A) **Basic Programming model and runtime**, **SPMD, MapReduce:** Hadoop, Spark, Twister, Stratosphere (Apache Flink), Reef, Hama, Giraph, Pregel, Pegasus, Ligra, GraphChi

13) **Inter process communication Collectives, point-to-point, publish-subscribe:** MPI, Harp, Netty, ZeroMQ, ActiveMQ, RabbitMQ, NaradaBrokering, QPid, Kafka, Kestrel, JMS, AMQP, Stomp, MQTT, **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

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

11B) **NoSQL:** Lucene, Solr, Solandra, Voldemort, Riak, 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, 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

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

Green implies HPC Integration

6) **DevOps:** Docker, Puppet, Chef, Ansible, SaltStack, Boto, Cobbler, Xcat, Razor, CloudMesh, Juju, Foreman, OpenStack Heat, Rocks, Cisco Intelligent Automation for Cloud, Ubuntu MaaS, Facebook Tupperware, AWS OpsWorks, OpenStack Ironic, Google Kubernetes, Buildstep, Gitreceive

5) **IaaS Management from HPC to hypervisors:** Xen, KVM, Hyper-V, VirtualBox, OpenVZ, LXC, Linux-Vserver, OpenStack, OpenNebula, Eucalyptus, Nimbus, CloudStack, CoreOS, VMware ESXi, vSphere and vCloud, Amazon, Azure, Google and other public Clouds,
Networking: Google Cloud DNS, Amazon Route 53

Functionality of 21 HPC-ABDS Layers

1) Message Protocols:
2) Distributed Coordination:
3) Security & Privacy:
4) Monitoring:
5) IaaS Management from HPC to hypervisors:
6) DevOps:
7) Interoperability:
8) File systems:
9) Cluster Resource Management:
10) Data Transport:
11) A) File management
    B) NoSQL
    C) SQL
12) In-memory databases&caches / Object-relational mapping / Extraction Tools
13) Inter process communication Collectives, point-to-point, publish-subscribe, MPI:
14) A) Basic Programming model and runtime, SPMD, MapReduce:
    B) Streaming:
15) A) High level Programming:
    B) Frameworks
16) Application and Analytics:
17) Workflow-Orchestration:

Here are 21 functionalities. (including 11, 14, 15 subparts)

4 Cross cutting at top
17 in order of layered diagram starting at bottom

	Big Data ABDS	HPC, Cluster
17. Orchestration	Crunch, Tez, Cloud Dataflow	Kepler, Pegasus, Taverna
16. Libraries	MLlib/Mahout, R, Python	ScaLAPACK, PETSc, Matlab
15A. High Level Programming	Pig, Hive, Drill	Domain-specific Languages
15B. Platform as a Service	App Engine, BlueMix, Elastic Beanstalk	XSEDE Software Stack
Languages	Java, Erlang, Scala, Clojure, SQL, SPARQL, Python	Fortran, C/C++, Python
14B. Streaming	Storm, Kafka, Kinesis	
13,14A. Parallel Runtime	Hadoop, MapReduce	MPI/OpenMP/OpenCL CUDA, Exascale Runtime
2. Coordination	Zookeeper	
12. Caching	Memcached	
11. Data Management	Hbase, Accumulo, Neo4J, MySQL	iRODS
10. Data Transfer	Sqoop	GridFTP
9. Scheduling	Yarn	Slurm
8. File Systems	HDFS, Object Stores	Lustre
1, 11A Formats	Thrift, Protobuf	FITS, HDF
5. IaaS	OpenStack, Docker	Linux, Bare-metal, SR-IOV
Infrastructure	CLOUDS	SUPERCOMPUTERS

HPC-ABDS Integrated Software

Software for a Big Data Initiative

- **Functionality of ABDS and Performance of HPC**
- **Workflow:** Apache Crunch, Python or Kepler
- **Data Analytics:** Mahout, R, ImageJ, Scalapack
- **High level Programming:** Hive, Pig
- **Batch Parallel Programming model:** Hadoop, Spark, Giraph, Harp, MPI;
- **Streaming Programming model:** Storm, Kafka or RabbitMQ
- **In-memory:** Memcached
- **Data Management:** Hbase, MongoDB, MySQL
- **Distributed Coordination:** Zookeeper
- **Cluster Management:** Yarn, Slurm
- **File Systems:** HDFS, Object store (Swift),Lustre
- **DevOps:** Cloudmesh, Chef, Puppet, Docker, Cobbler
- **IaaS:** Amazon, Azure, OpenStack, Docker, SR-IOV
- **Monitoring:** Inca, Ganglia, Nagios

Ogres PROBLEM ARCHITECTURE View