Returning to Java Grande: High Performance Architecture for Big Data

INTERNATIONAL ADVANCED RESEARCH WORKSHOP
ON HIGH PERFORMANCE COMPUTING
From Clouds and Big Data to Exascale and Beyond

Cetraro (Italy)

July 7 2014

Geoffrey Fox

gcf@indiana.edu

http://www.infomall.org

School of Informatics and Computing

Digital Science Center

Indiana University Bloomington

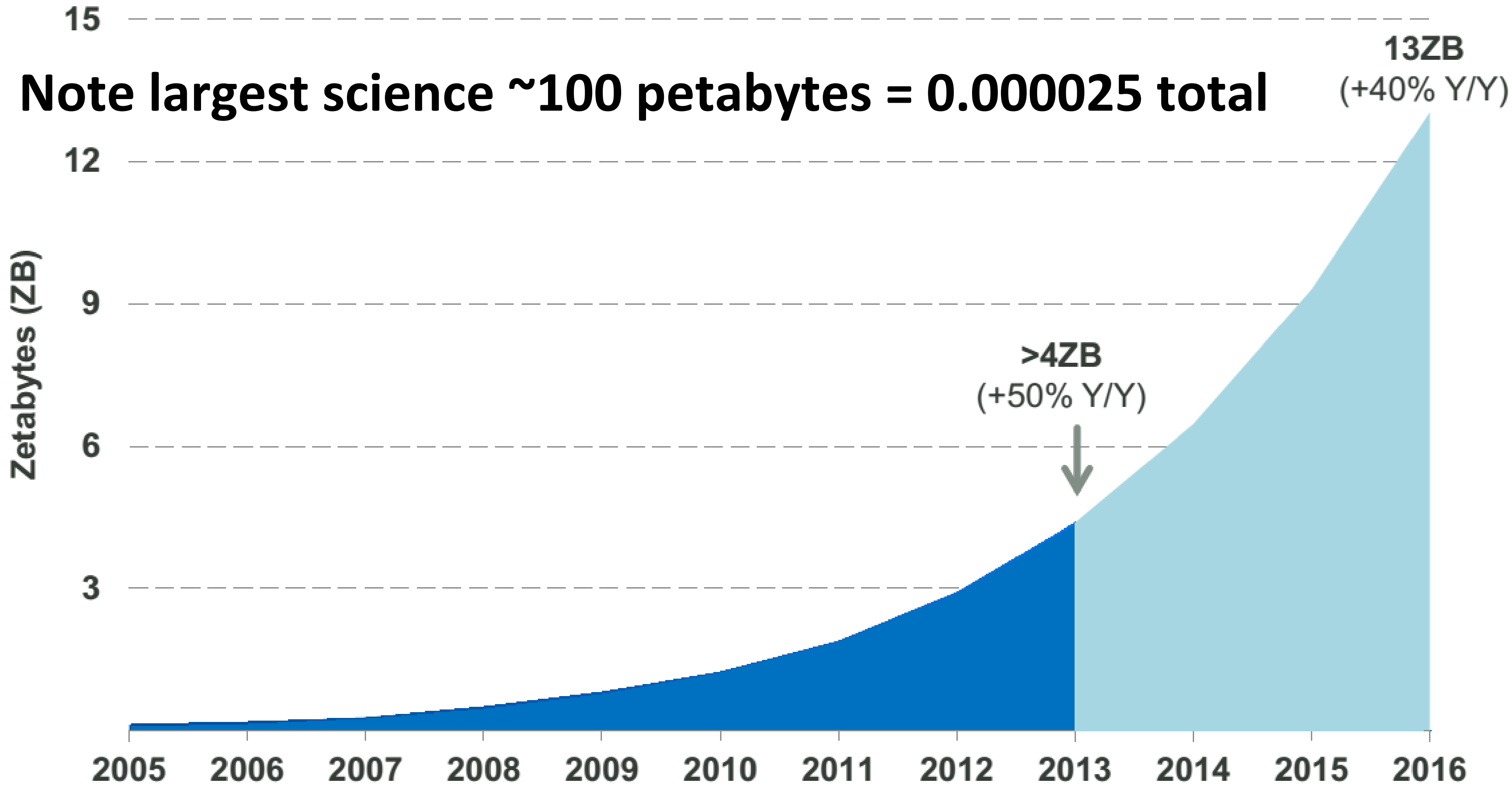
Abstract

- Here we use a sample of over 50 big data applications to identify characteristics of data intensive applications and to deduce needed runtime and architectures.
- We propose a big data version of the famous Berkeley dwarfs and NAS parallel benchmarks as the kernel big data applications.
- We suggest that one must unify HPC with the well known Apache software stack that is well used in modern cloud computing and surely is most widely used data processing framework in the "real world".
- We give some examples including clustering, deep-learning and multi-dimensional scaling. This work suggests the value of a high performance Java (Grande) runtime that supports simulations and big data.

'Digital Universe' Information Growth = Robust... +50%, 2013

2/3rd's of Digital Universe Content = Consumed / Created by Consumers ...Video Watching, Social Media Usage, Image Sharing...

Note largest science ~100 petabytes = 0.000025 total



@KPCB

Note: 1 petabyte = 1MM gigabytes, 1 zetabyte = 1MM petabytes.
Source: IDC Digital Universe, data as of 5/14.

http://www.kpcb.com/internet-trends

NIST Big Data Use Cases

Led by Chaitin Baru, Bob Marcus, Wo Chang

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 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**

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,

Big Data Patterns – the Ogres

Would like to capture “essence of these use cases”

“small” kernels, mini-apps
Or Classify applications into patterns

Do it from HPC background **not database** viewpoint
e.g. focus on cases with detailed analytics

Section 5 of my class
https://bigdatacoursespring2014.appspot.com/preview classifies
51 use cases with ogre facets

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

- Dense Linear Algebra
- Sparse Linear Algebra
- Spectral Methods
- N-Body Methods
- Structured Grids
- Unstructured Grids
- MapReduce
- Combinational Logic
- Graph Traversal
- Dynamic Programming
- Backtrack and Branch-and-Bound
- Graphical Models
- 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!

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

51 Use Cases: Low-Level (Run-time) Computational Types

- **PP(26):** Pleasingly Parallel or Map Only
- **MR(18 +7 MRStat):** Classic MapReduce
- **MRStat(7):** Simple version of MR where key computations are simple reduction as coming in statistical averages
- **MRIter(23):** Iterative MapReduce or MPI
- **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

(Count) out of 51

51 Use Cases: Higher-Level Computational Types or Features

Not Independent

- **Classification(30):** divide data into categories
- **S/Q/Index(12):** Search and Query
- **CF(4):** Collaborative Filtering
- **LML - Local ML(36):** Local Machine Learning (**Independent for each entity**)
- **GML - Global ML(23):** Deep Learning, Clustering, LDA, PLSI, MDS, Large Scale Optimizations as in Variational Bayes, Lifted Belief Propagation, Stochastic Gradient Descent, L-BFGS, Levenberg-Marquardt (Sometimes call **EGO** or Exascale Global Optimization – **scalable parallel algorithm**)
- **Workflow:** (Left out of analysis but ~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. generates big data
- **Agent(2):** Simulations of models of data-defined macroscopic entities represented as agents

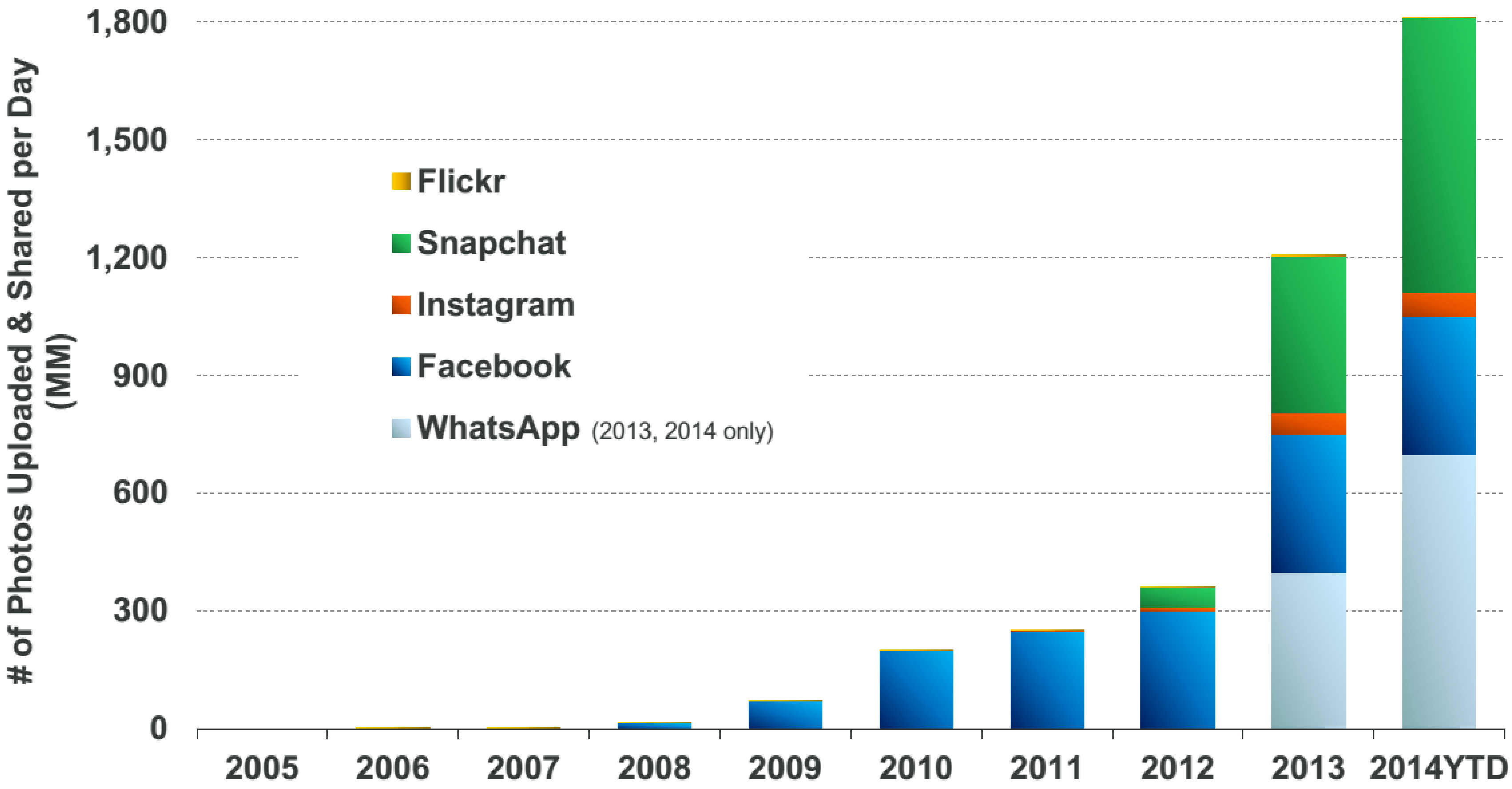
Global Machine Learning aka EGO – Exascale Global Optimization

- Typically 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). Usually it's a sum of positive numbers as in least squares
- 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?
- Detailed papers on particular parallel graph algorithms
- Name invented at Argonne-Chicago workshop

Image and Internet of Things based Applications

Photos Alone = 1.8B+ Uploaded & Shared Per Day...
Growth Remains Robust as New Real-Time Platforms Emerge

Daily Number of Photos Uploaded & Shared on Select Platforms, 2005 – 2014YTD



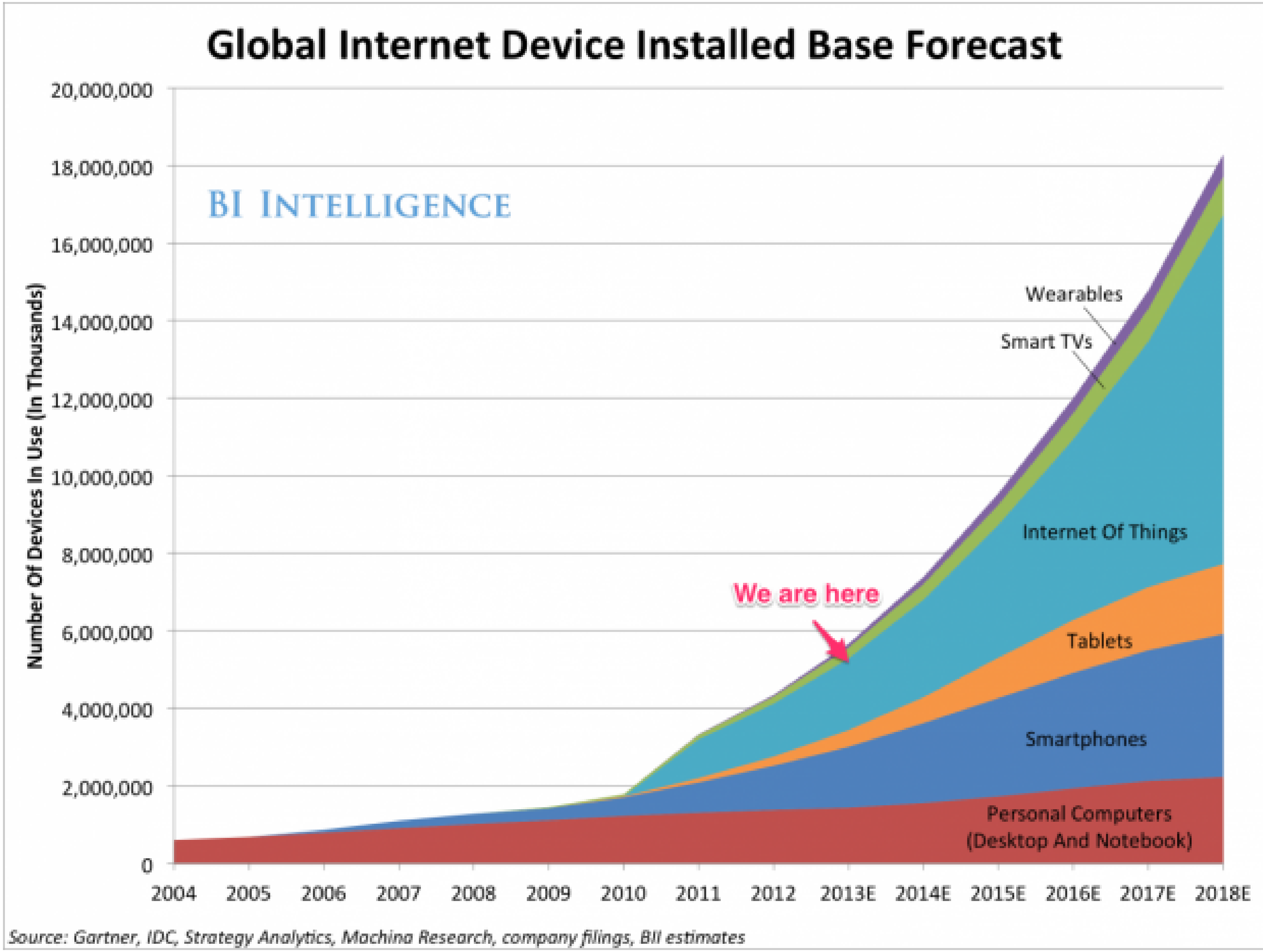
@KPCB

Source: KPCB estimates based on publicly disclosed company data, 2014 YTD data per latest as of 5/14.

http://www.kpcb.com/internet-trends

9 Image-based Use Cases

- **17:Pathology Imaging/ Digital Pathology:** **PP, LML, MR** for search becoming terabyte 3D images, Global Classification
- **18: 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


Global Internet Device Installed Base Forecast
BI INTELLIGENCE
Number Of Devices In Use (In Thousands)
20,000,000
18,000,000
16,000,000
14,000,000
12,000,000
10,000,000
8,000,000
6,000,000
4,000,000
2,000,000
0
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013E
2014E
2015E
2016E
2017E
2018E
Wearables
Smart TVs
Internet Of Things
We are here
Tablets
Smartphones
Personal Computers
(Desktop And Notebook)
Source: Gartner, IDC, Strategy Analytics, Machina Research, company filings, BII estimates

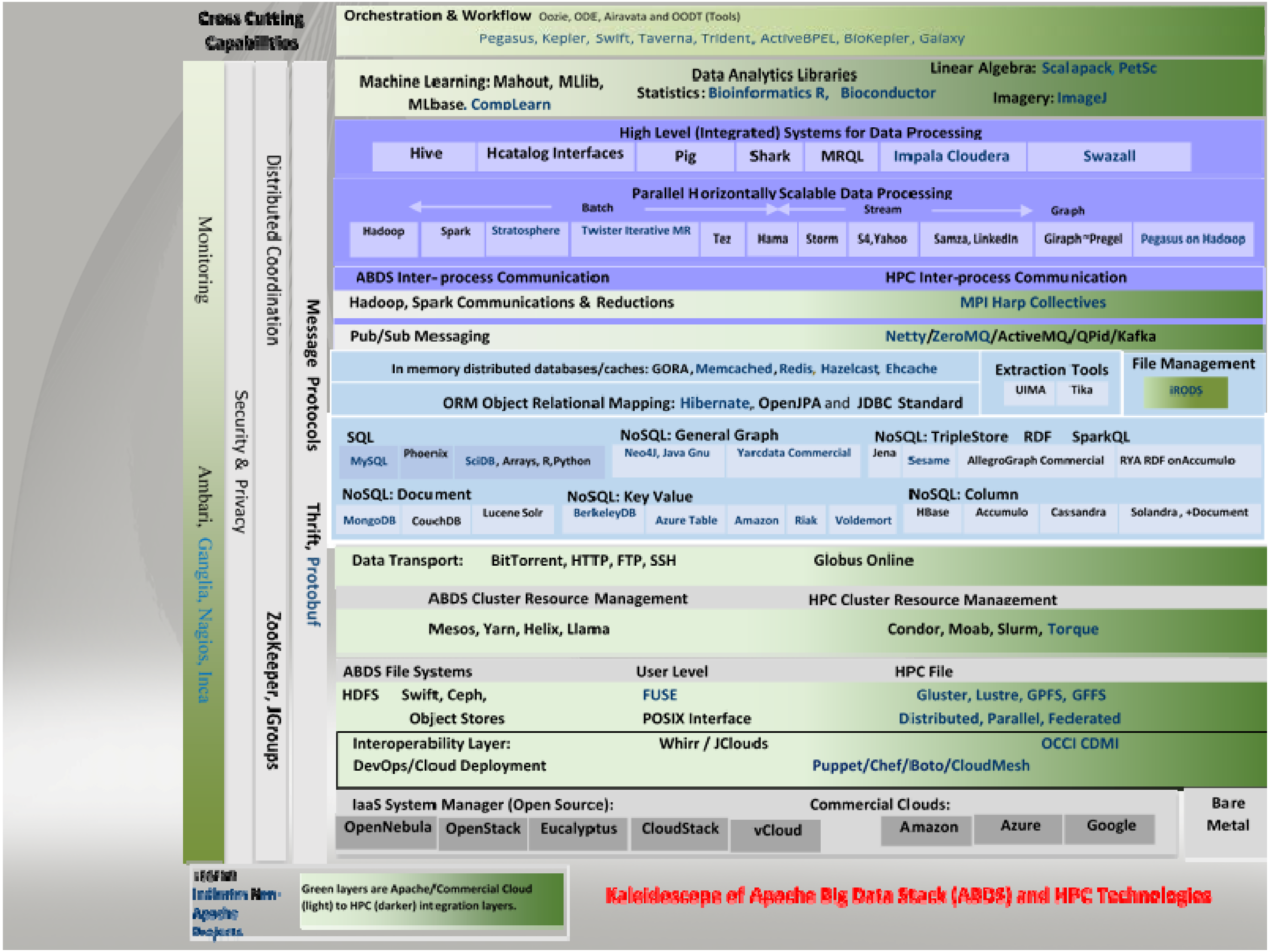
Internet of Things and Streaming Apps

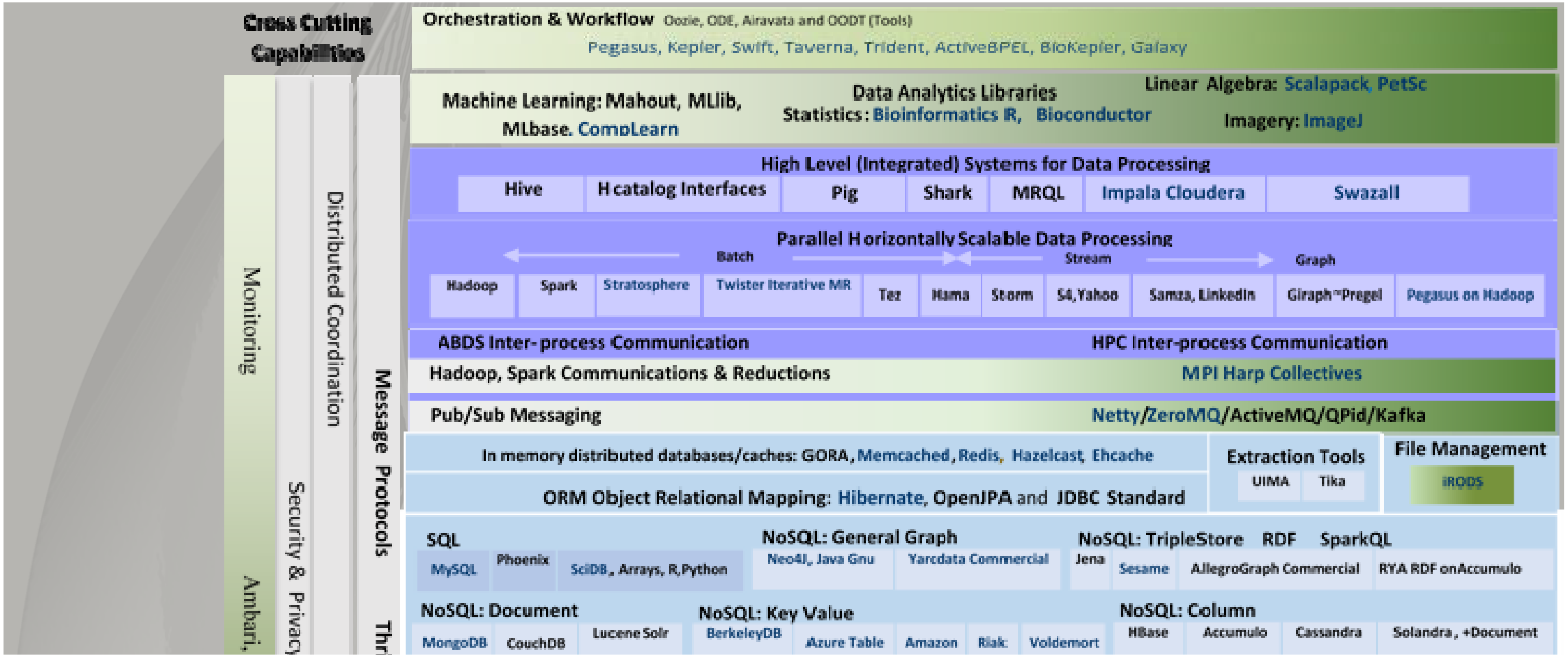
- It is projected that there will be **24 (Mobile Industry Group) to 50 (Cisco) billion devices** on the Internet by 2020.
- The **cloud** natural controller of and **resource provider for the Internet of Things.**
- Smart phones/watches, Wearable devices (Smart People), "Intelligent River" "Smart Homes 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 Local Processing Global statistics**
- **50: DOE-BER AmeriFlux and FLUXNET Networks PP GIS LML**
- **51: Consumption forecasting in Smart Grids PP GIS LML**

HPC-ABDS

Integrating High Performance Computing with Apache Big Data Stack

Shantenu Jha, Judy Qiu, Andre Luckow


Cross Cutting Capabilities
Monitoring
Ambari, Ganglia, Nagios, Inca
Security & Privacy
Distributed Coordination
ZooKeeper, JGroups
Message Protocols
Thrift, Protobuf
Orchestration & Workflow Oozie, ODE, Airavata and OODT (Tools)
Pegasus, Kepler, Swift, Taverna, Trident, ActiveBPEL, BioKepler, Galaxy
Machine Learning: Mahout, MLlib, MLbase, CompLearn
Data Analytics Libraries
Statistics: Bioinformatics R, Bioconductor
Linear Algebra: Scalapack, PetSc
Imagery: ImageJ
High Level (Integrated) Systems for Data Processing
Hive
Hcatalog Interfaces
Pig
Shark
MRQL
Impala Cloudera
Swazall
Parallel Horizontally Scalable Data Processing
Batch
Stream
Graph
Hadoop
Spark
Stratosphere
Twister Iterative MR
Tez
Hama
Storm
S4,Yahoo
Samza, LinkedIn
Giraph~Pregel
Pegasus on Hadoop
ABDS Inter-process Communication
HPC Inter-process Communication
Hadoop, Spark Communications & Reductions
MPI Harp Collectives
Pub/Sub Messaging
Netty/ZeroMQ/ActiveMQ/QPid/Kafka
In memory distributed databases/caches: GORA, Memcached, Redis, Hazelcast, Ehcache
ORM Object Relational Mapping: Hibernate, OpenJPA and JDBC Standard
Extraction Tools
UIMA
Tika
File Management
iRODS
SQL
MySQL
Phoenix
SciDB, Arrays, R,Python
NoSQL: General Graph
Neo4J, Java Gnu
Yarcdata Commercial
NoSQL: TripleStore RDF SparkQL
Jena
Sesame
AllegroGraph Commercial
RYA RDF onAccumulo
NoSQL: Document
MongoDB
CouchDB
Lucene Solr
NoSQL: Key Value
BerkeleyDB
Azure Table
Amazon
Riak
Voldemort
NoSQL: Column
HBase
Accumulo
Cassandra
Solandra, +Document
Data Transport: BitTorrent, HTTP, FTP, SSH
Globus Online
ABDS Cluster Resource Management
HPC Cluster Resource Management
Mesos, Yarn, Helix, Llama
Condor, Moab, Slurm, Torque
ABDS File Systems
User Level
HPC File
HDFS
Swift, Ceph, Object Stores
FUSE
POSIX Interface
Gluster, Lustre, GPFS, GFFS
Distributed, Parallel, Federated
Interoperability Layer:
Whirr / JClouds
OCCI CDMI
DevOps/Cloud Deployment
Puppet/Chef/Boto/CloudMesh
IaaS System Manager (Open Source):
OpenNebula
OpenStack
Eucalyptus
CloudStack
vCloud
Commercial Clouds:
Amazon
Azure
Google
Bare Metal
Legend
Indicates Non-Apache Projects
Green layers are Apache/Commercial Cloud (light) to HPC (darker) integration layers.
Kaleidoscope of Apache Big Data Stack (ABDS) and HPC Technologies



- HPC-ABDS
- ~120 Capabilities
- >40 Apache
- **Green layers have strong HPC Integration opportunities**

- **Goal**
- **Functionality of ABDS**
- **Performance of HPC**

Cross-Cutting Functionalities

- Message Protocols
- Distributed Coordination
- Security & Privacy
- Monitoring

~120 HPC-ABDS Software capabilities in 17 functionalities

- Workflow-Orchestration
- Application and Analytics: Mahout, MLlib, R…
- High level Programming
- Basic Programming model and runtime SPMD, Streaming, MapReduce, MPI
- Inter process communication Collectives, point-to-point, publish-subscribe
- In-memory databases/caches
- Object-relational mapping
- SQL and NoSQL, File management
- Data Transport
- Cluster Resource Management
- File systems
- DevOps
- IaaS Management from HPC to hypervisors

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

SPIDAL (Scalable Parallel Interoperable Data Analytics Library)

Getting High Performance on Data Analytics

- On the systems side, we have two principles:
  - The Apache Big Data Stack with ~120 projects has important broad functionality with a vital large support organization
  - HPC including MPI has striking success in delivering high performance, however with a fragile sustainability model
- There are key systems abstractions which are levels in HPC-ABDS software stack where Apache approach needs careful integration with HPC
  - Resource management
  - Storage
  - Programming model -- horizontal scaling parallelism
  - Collective and Point-to-Point communication
  - Support of iteration
  - Data interface (not just key-value)
- In application areas, we define application abstractions to support:
  - Graphs/network
  - Geospatial
  - Genes
  - Images, etc.

HPC-ABDS Hourglass

HPC ABDS
System (Middleware)

120 Software Projects

System Abstractions/standards
•Data format
•Storage

- **HPC Yarn for Resource management**
- **Horizontally scalable parallel programming model**
- **Collective and Point-to-Point communication**
- **Support of iteration (in memory databases)**

Application Abstractions/standards
Graphs, Networks, Images, Geospatial ….

High performance
Applications

SPIDAL (Scalable Parallel Interoperable Data Analytics Library) or High performance Mahout, R, Matlab…

Useful Set of Analytics Architectures

- **Pleasingly Parallel:** including **local machine learning** as in parallel over images and apply image processing to each image

  - Hadoop could be used but many other HTC, Many task tools
- **Search:** including collaborative filtering and motif finding implemented using **classic MapReduce** (Hadoop)
- **Map-Collective or Iterative MapReduce** using Collective Communication (clustering) – Hadoop with Harp, Spark …..
- **Map-Communication or Iterative Giraph:** (MapReduce) with point-to-point communication (most graph algorithms such as maximum clique, connected component, finding diameter, community detection)
  – Vary in difficulty of finding partitioning (classic parallel load balancing)
- **Shared memory: thread-based** (event driven) graph algorithms (shortest path, Betweenness centrality)

Ideas like workflow are "orthogonal" to this

Facets of the Ogres

Application Class Facet of Ogres

- **Classification (30)** divide data into categories
- **Search Index** and **query (12)**
- **Maximum Likelihood** or χ^2 minimizations
- **Expectation Maximization** (often Steepest descent)
- **Local (pleasingly parallel) Machine Learning (36)** contrasted to
- **(Exascale) Global Optimization (23)** (such as Learning Networks, Variational Bayes and Gibbs Sampling)
- Do they **Use Agents (2)** as in epidemiology (swarm approaches)?

Higher-Level Computational Types or Features in earlier slide also has
CF(4): Collaborative Filtering in **Core Analytics Facet**
and two categories in data source and style
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. generates big data

Problem Architecture Facet of Ogres (Meta or MacroPattern)

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 sophisticat

ii. **Classic MapReduce** for Search anc

iii. **Global Analytics or Machine Lear** programming models

iv. **Problem set up as a graph** as opp

v. **SPMD (Single Program Multiple D**

vi. **Bulk Synchronous Processing:** we communication phases

vii. **Fusion:** Knowledge discovery ofte methods.

viii. **Workflow** (often used in fusion)

Note problem and machine architectu

Slight expansion of an earlier slides on:

Major Analytics Architectures in Use Cases

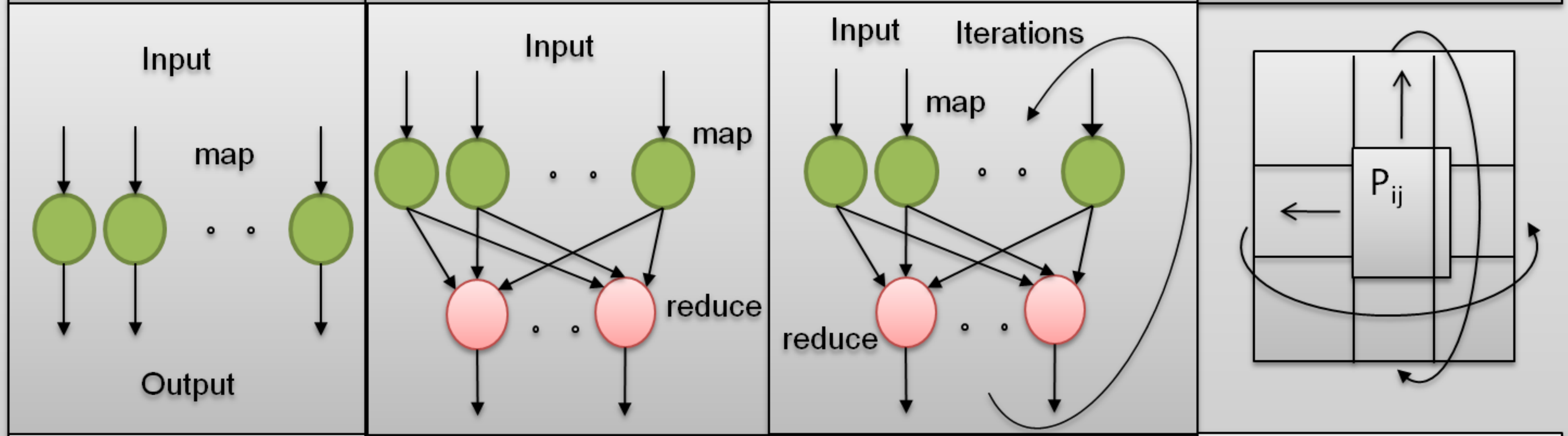
Pleasingly parallel
Search (MapReduce)
Map-Collective
Map-Communication as in MPI
Shared Memory

Low-Level (Run-time) Computational Types used to label 51 use cases

PP(26): Pleasingly Parallel
MR(18 +7 MRStat): Classic MapReduce
MRStat(7)
MRIter(23)
Graph(9)
Fusion(11)
Streaming(41) In data source

4 Forms of MapReduce

(a) Map Only	(b) Classic MapReduce	(c) Iterative Map Reduce or Map-Collective	(d) Point to Point



BLAST Analysis Local Machine Learning Pleasingly Parallel	High Energy Physics (HEP) Histograms Distributed search	Expectation maximization Clustering e.g. K-means Linear Algebra, PageRank	Classic MPI PDE Solvers and particle dynamics
← Domain of MapReduce and Iterative Extensions →			MPI Giraph

All of them are Map-Communication?

One Facet of Ogres has Computational Features

a) Flops per byte;
b) Communication Interconnect requirements;
c) Is application (graph) **constant** or **dynamic?**
d) Most applications consist of a set of interconnected entities; is this **regular** as a set of pixels or is it a complicated **irregular graph?**
e) Is communication **BSP** or **Asynchronous?** In latter case **shared memory** may be attractive;
f) Are algorithms **Iterative** or **not?**
g) Data Abstraction: key-value, pixel, graph, vector
  - Are data points in **metric or non-metric spaces?**

h) Core libraries needed: matrix-matrix/vector algebra, conjugate gradient, reduction, broadcast

Data Source and Style Facet of Ogres

- (i) **SQL**
- (ii) **NOSQL** based
- (iii) Other Enterprise data systems (10 examples from Bob Marcus)
- (iv) **Set of Files** (as managed in iRODS)
- (v) **Internet of Things**
- (vi) **Streaming** and
- (vii) **HPC simulations**
- (viii) Involve **GIS** (Geographical Information Systems)
- 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)
- There are **storage/compute system styles:** Shared, Dedicated, Permanent, Transient
- Other characteristics are needed for permanent **auxiliary/comparison datasets a**nd these could be interdisciplinary, implying nontrivial data movement/replication

Analytics Facet (kernels) of the Ogres

Core Analytics Facet of Ogres (microPattern) I

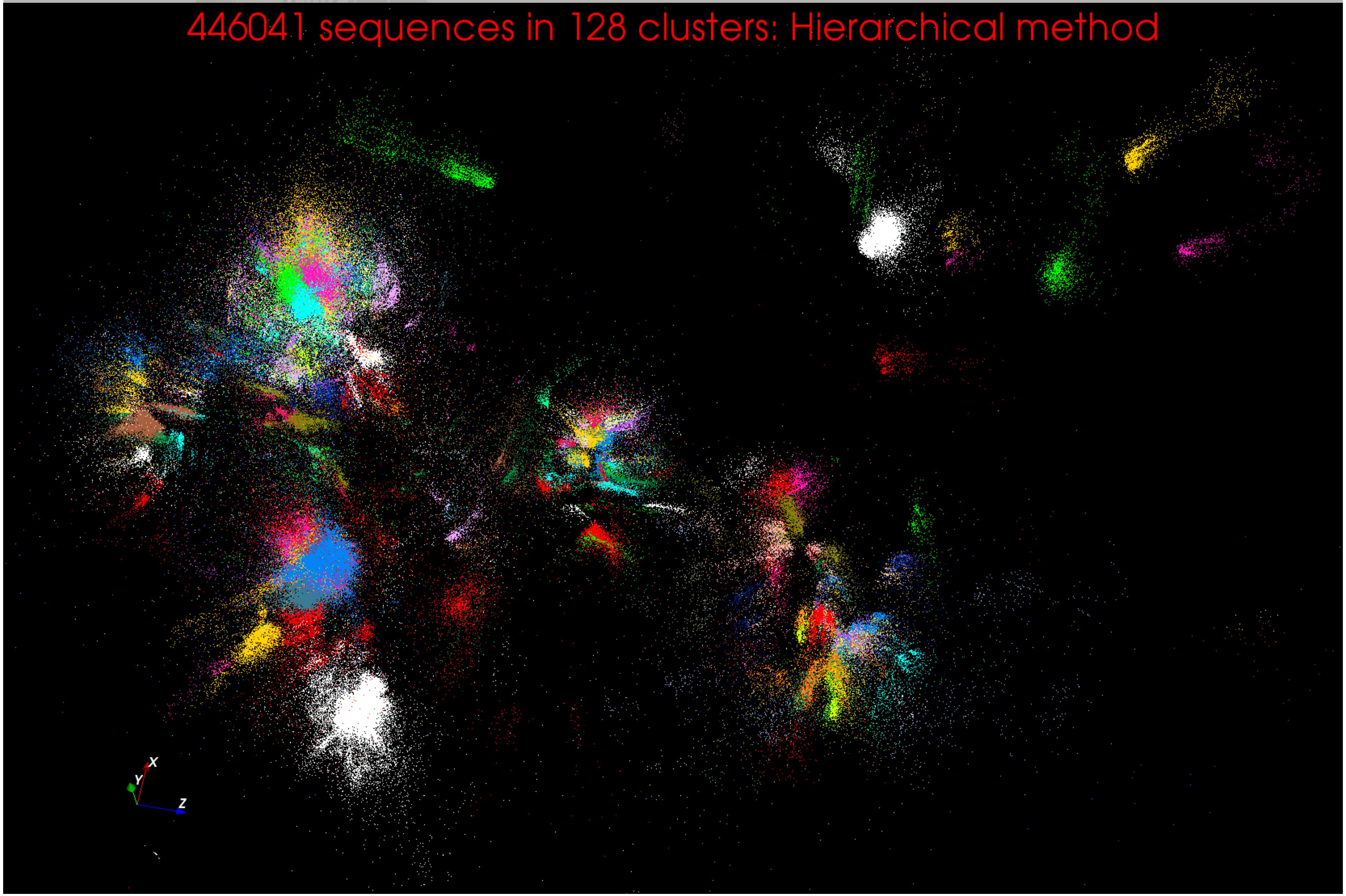
- **Map-Only**
- Pleasingly parallel - **Local Machine Learning**
- **MapReduce: Search/Query**
- Summarizing **statistics** as in LHC Data analysis (histograms)
- Recommender Systems (**Collaborative Filtering**)
- Linear Classifiers (**Bayes, Random Forests**)
- **Global Analytics**
- **Nonlinear Solvers (structure depends on objective function)**
  - Stochastic Gradient Descent SGD
  - (L-)BFGS approximation to Newton's Method
  - Levenberg-Marquardt solver
- **Map-Collective I (need to improve/extend Mahout, MLlib)**
- **Outlier Detection**, **Clustering** (many methods),
- **Mixture Models, LDA** (Latent Dirichlet Allocation), **PLSI** (Probabilistic Latent Semantic Indexing)

Core Analytics Facet of Ogres (microPattern) II

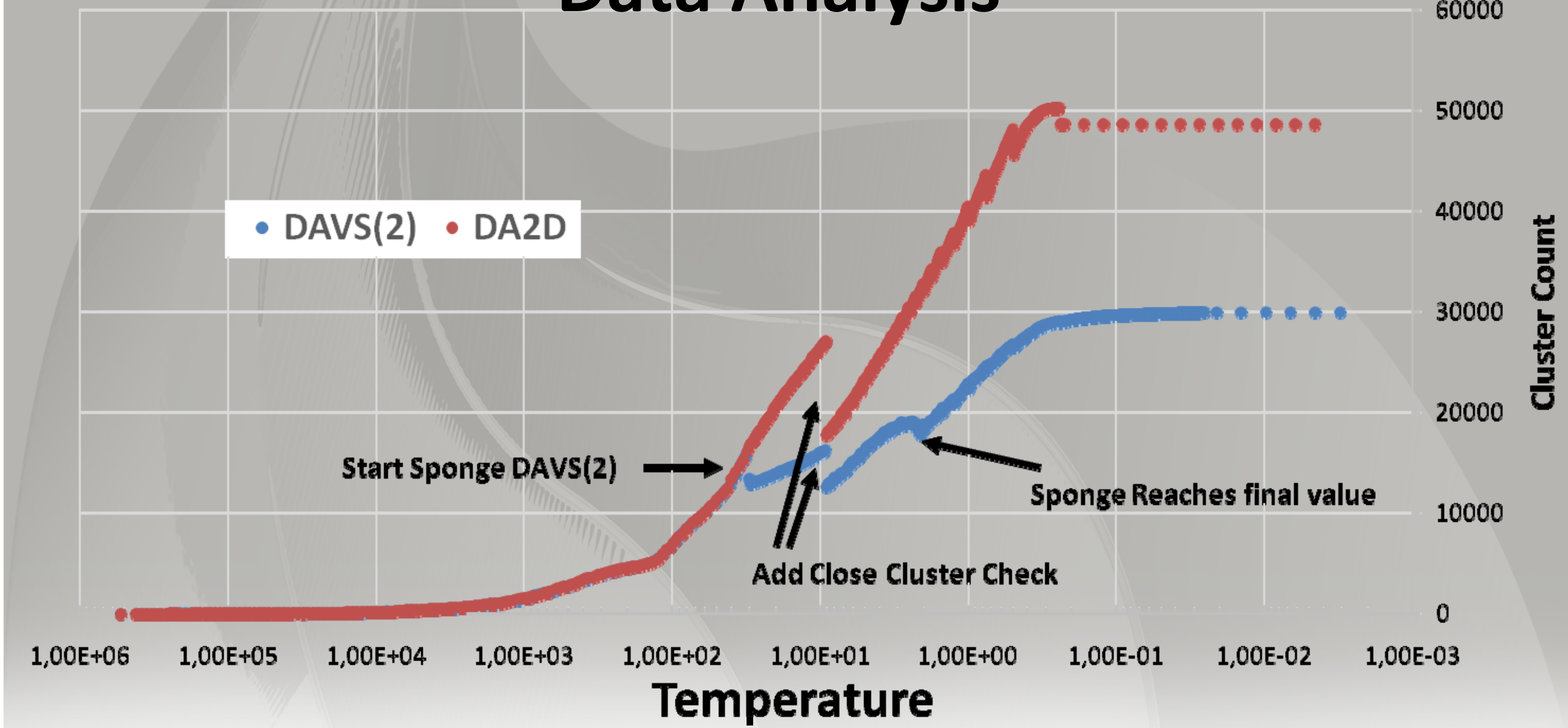
- Map-Collective II
- Use matrix-matrix,-vector operations, solvers (conjugate gradient)
- **SVM** and **Logistic Regression**
- **PageRank**, (find leading eigenvector of sparse matrix)
- **SVD** (Singular Value Decomposition)
- **MDS** (Multidimensional Scaling)
- Learning Neural Networks (**Deep Learning**)
- **Hidden Markov Models**
- Map-Communication
- **Graph Structure (Communities, subgraphs/motifs, diameter, maximal cliques, connected components)**
- **Network Dynamics - Graph simulation Algorithms** (epidemiology)
- Asynchronous Shared Memory
- **Graph Structure (Betweenness centrality, shortest path)**

Parallel Global Machine Learning Examples
Initial SPIDAL entries

Clustering and MDS Large Scale $O(N^2)$ GML



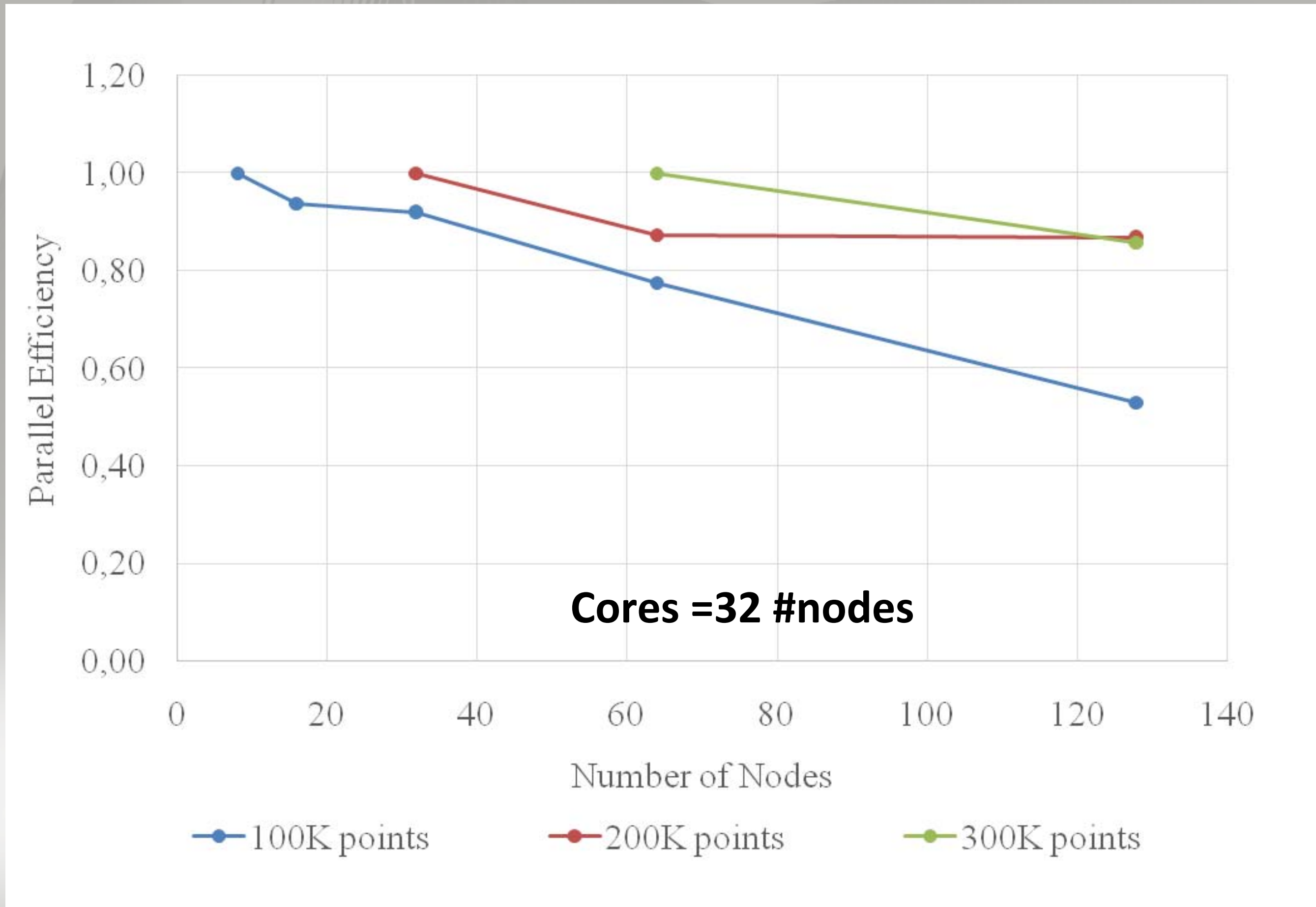
Cluster Count v. Temperature for LC-MS Data Analysis



- All start with one cluster at far left
- T=1 special as measurement errors divided out
- DA2D counts clusters with 1 member as clusters. DAVS(2) does not

WDA SMACOF MDS (Multidimensional Scaling) using Harp on IU Big Red 2

Parallel Efficiency: on 100-300K sequences

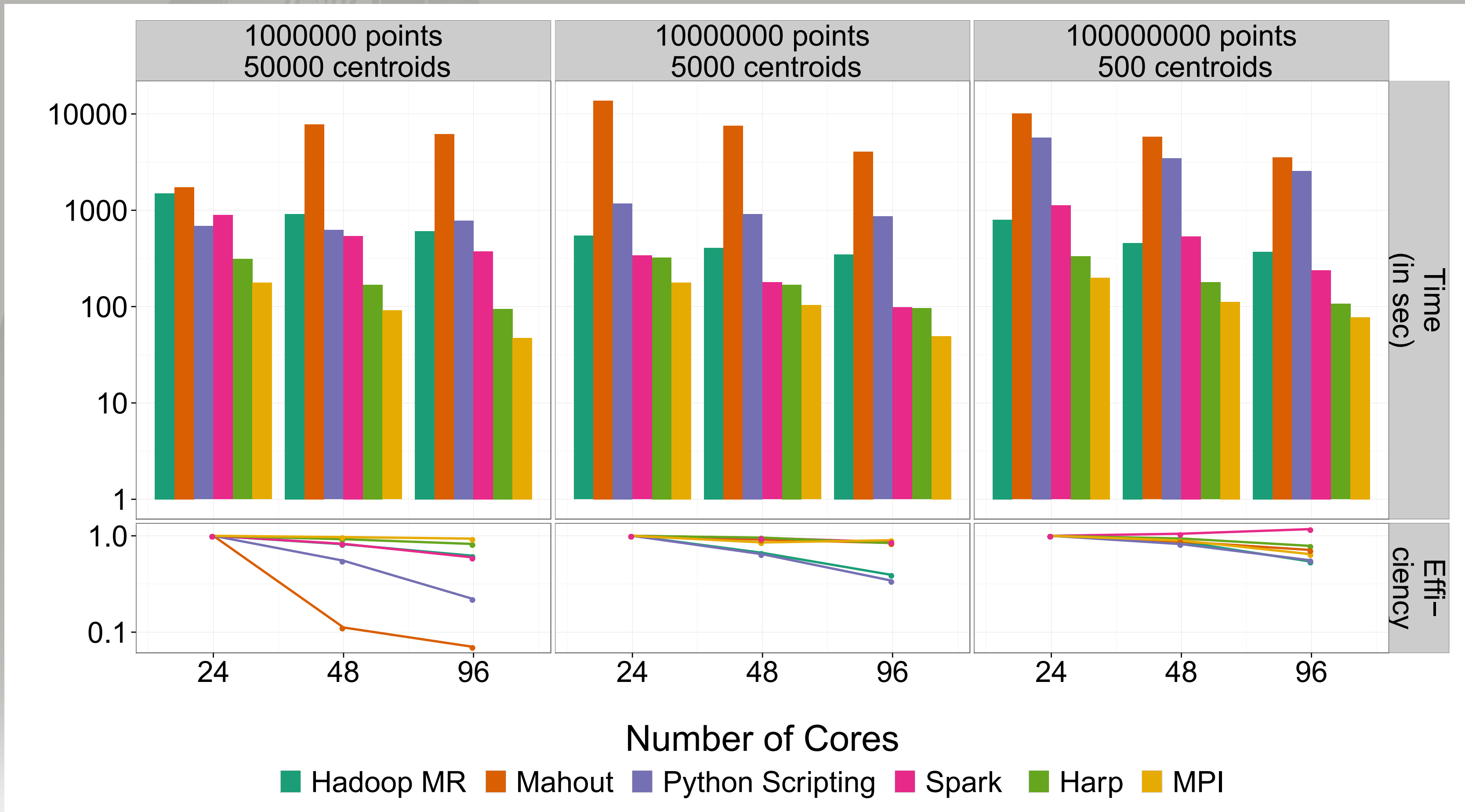


Best available MDS (much better than that in R)
Java

Harp (Hadoop plugin) described by Qiu later

Conjugate Gradient (dominant time) and Matrix Multiplication

Increasing Communication **Identical Computation**



Mahout and Hadoop MR – Slow due to MapReduce

Python slow as Scripting; **MPI** fastest

Spark Iterative MapReduce, non optimal communication

Harp Hadoop plug in with ~MPI collectives

Comparing Data Intensive and Simulation Problems

Comparison of Data Analytics with Simulation I

- **Pleasingly parallel** often important in both
- Both are often **SPMD** and **BSP**
- **Non-iterative MapReduce** is major big data paradigm
  - not a common simulation paradigm except where "Reduce" summarizes pleasingly parallel execution
- Big Data often has **large collective communication**
  - Classic simulation has a lot of smallish point-to-point messages
- Simulation dominantly **sparse** (nearest neighbor) data structures
  - "Bag of words (users, rankings, images..)" algorithms are sparse, as is PageRank
  - Important data analytics involves full matrix algorithms

Comparison of Data Analytics with Simulation II

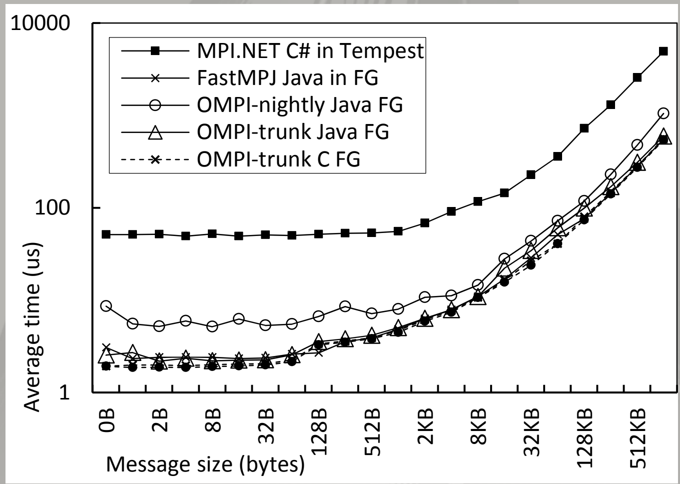
- There are similarities between some **graph problems and particle simulations** with a **strange cutoff force.**
  - Both **Map-Communication**
- Note many big data problems are "**long range force**" as all points are linked.
  - Easiest to parallelize. Often full matrix algorithms
  - e.g. in DNA sequence studies, distance $\delta(i, j)$ defined by BLAST, Smith-Waterman, etc., between all sequences *i*, *j.*
  - Opportunity for "fast multipole" ideas in big data.
- In image-based **deep learning**, neural network weights are block sparse (corresponding to links to pixel blocks) but can be formulated as full matrix operations on GPUs and MPI in blocks.
- In HPC benchmarking, Linpack being challenged by a new sparse conjugate gradient benchmark HPCG, while I am diligently using **non-sparse conjugate gradient solvers** in clustering and Multi-dimensional scaling.

Java Grande

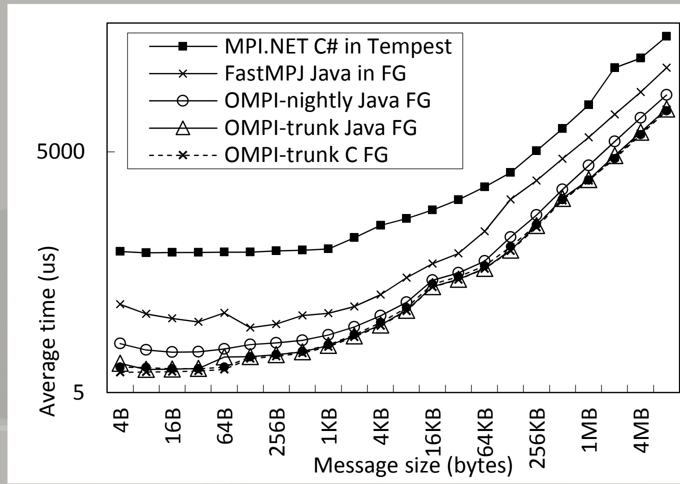
Java Grande

- We once tried to encourage use of Java in HPC with Java Grande Forum but Fortran, C and C++ remain central HPC languages.
  - Not helped by .com and Sun collapse in 2000-2005
- The pure Java CartaBlanca, a 2005 R&D100 award-winning project, was an early successful example of HPC use of Java in a simulation tool for non-linear physics on unstructured grids.
- Of course Java is a major language in ABDS and as data analysis and simulation are naturally linked, should consider broader use of Java
- Using Habanero Java (from Rice University) for Threads and mpiJava or FastMPJ for MPI, gathering collection of high performance parallel Java analytics
  - Converted from C# and sequential Java faster than sequential C#
- So will have either Hadoop+Harp or classic Threads/MPI versions in Java Grande version of Mahout

Performance of MPI Kernel Operations

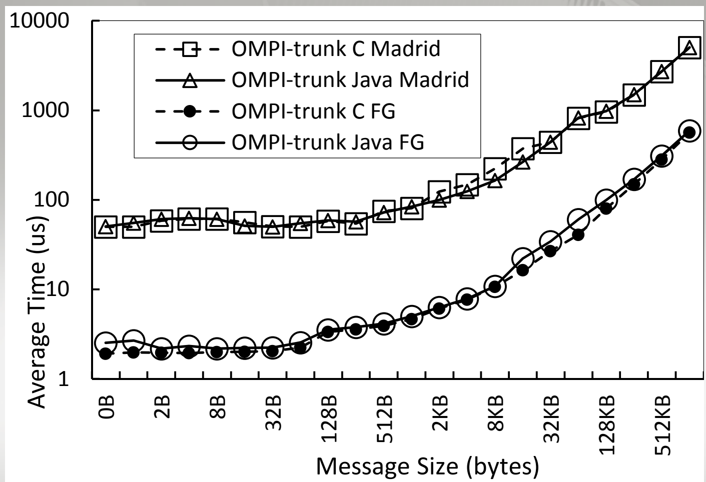


Performance of MPI send and receive operations

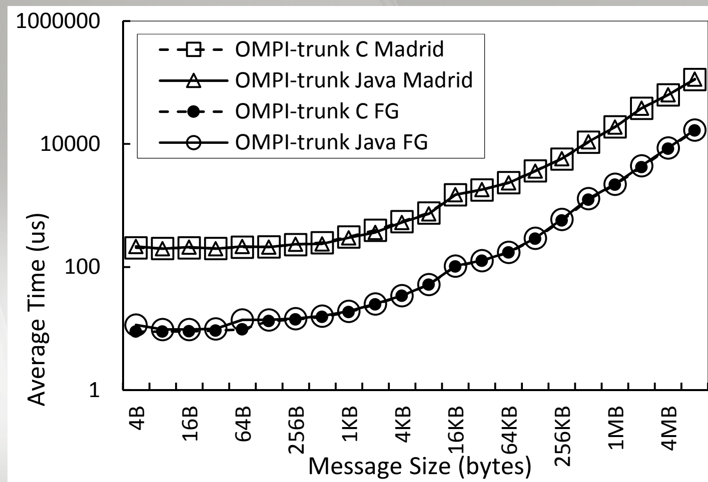


Performance of MPI allreduce operation

Pure Java as in FastMPJ slower than Java interfacing to C version of MPI



Performance of MPI send and receive on Infiniband and Ethernet



Performance of MPI allreduce on Infiniband and Ethernet

Lessons / Insights

- **Integrate** (don't compete) **HPC with "Commodity Big data"** (Google to Amazon to Enterprise Data Analytics)
  - i.e. **improve Mahout**; don't compete with it
  - Use **Hadoop plug-ins** rather than replacing Hadoop
- Enhanced Apache Big Data Stack **HPC-ABDS has ~120 members**
- Opportunities at Resource management, Data/File, Streaming, Programming, monitoring, workflow layers for HPC and ABDS integration
- Data intensive algorithms do not have the well developed **high performance libraries** familiar from HPC
- **Global Machine Learning** or (Exascale Global Optimization) particularly challenging
- Strong case for high performance Java (Grande) run time supporting all forms of parallelism