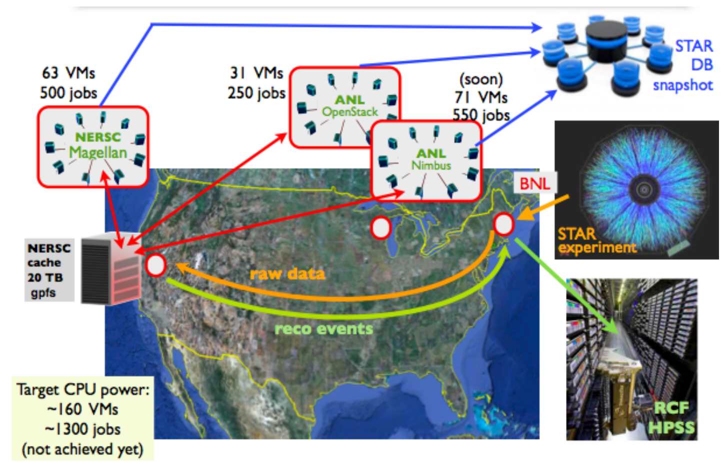
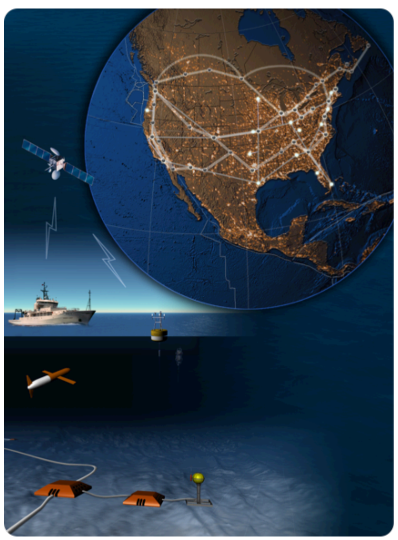
Virtual Observatories: A Facility for Online Data Analysis

Kate Keahey
Argonne National Laboratory
keahey@mcs.anl.gov

Supporting Online Data Analysis

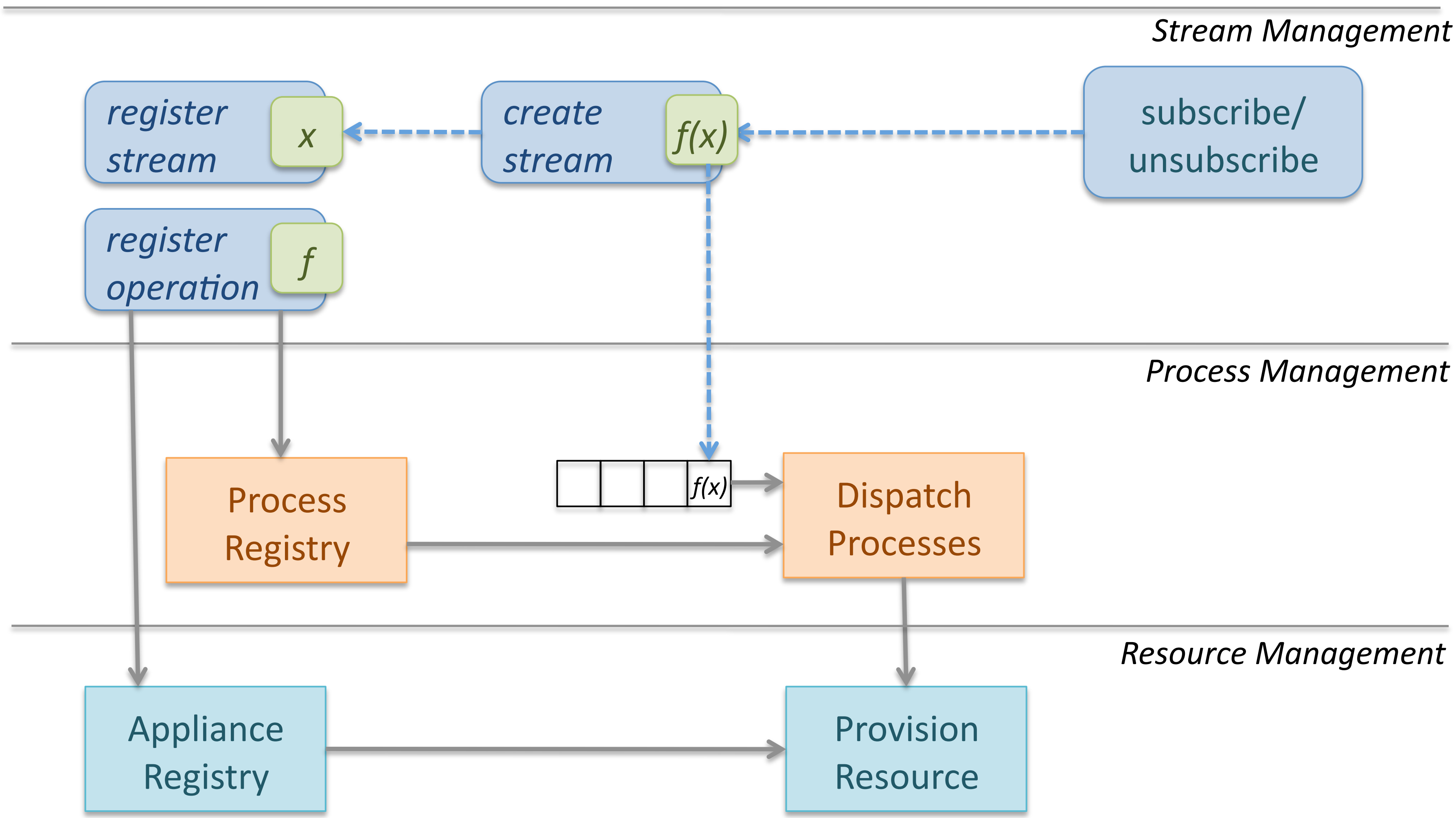
- Existing experimental platform
  - W boson reconstruction
  - Discrete experimental events
  - “Time to science” and feedback
  - Challenging but well understood data processing needs





- Emerging experimental platform
  - From exploratory to observatory science
  - “Always-on” service
  - Highly volatile processing needs
  - Real-time event-based data streaming
  - Services providing high availability and auto scaling

Virtual Observatory Infrastructure



Challenges @Scale

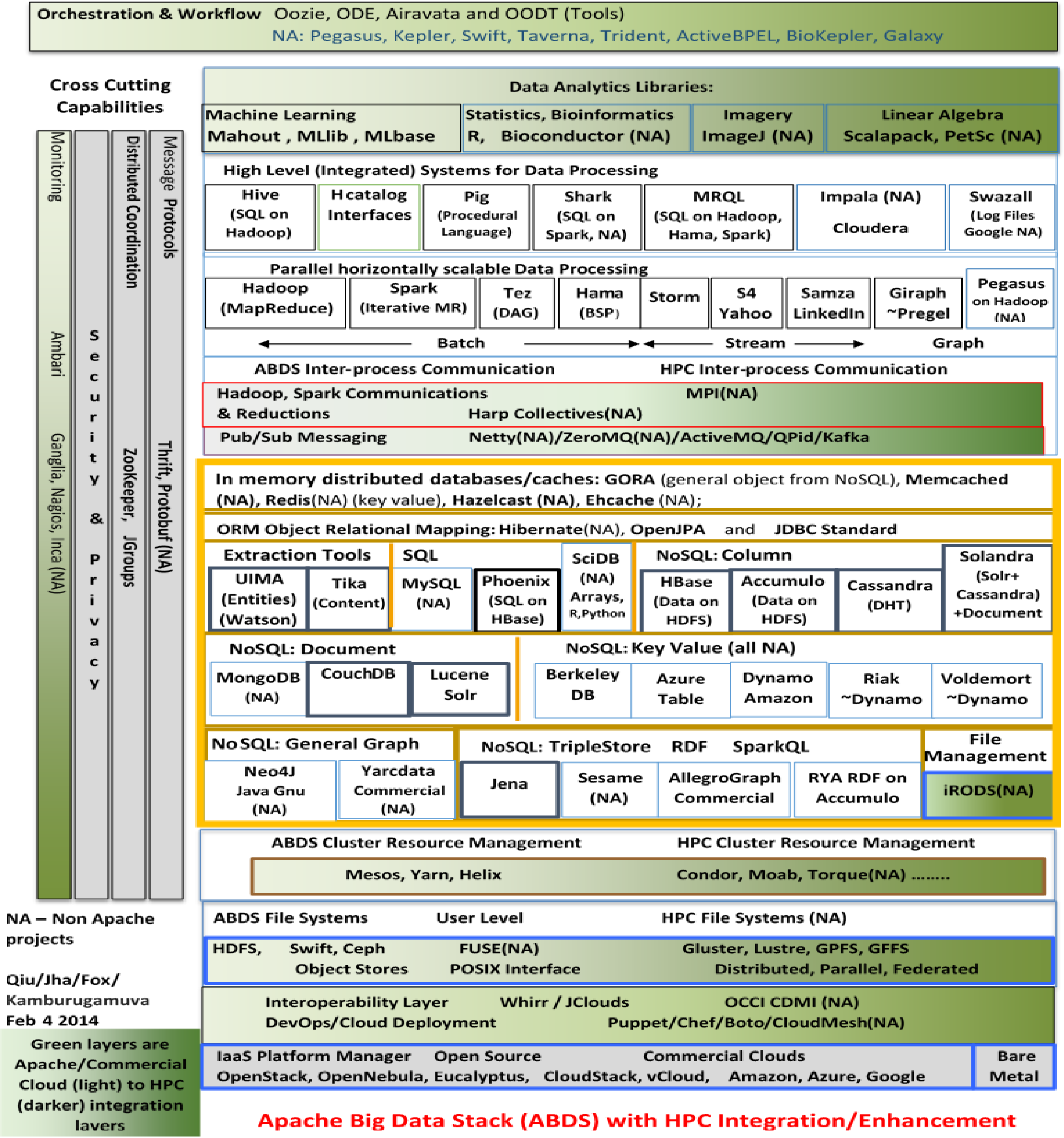
- Towards dynamic management
  - Extensibility: user operation definitions
  - Publish/subscribe: dynamic, interactive workflow
  - Late binding, late resource acquisition
  - Scalability and availability
  - Lightweight process dispatch
- Big Data
  - STAR adventures: holistic scheduling
  - Atlas adventures: better separation of compute and network
- Big Compute
  - On-demand versus on-availability in a resource rich environment
  - Big Compute pattern even in small compute

High Performance High Functionality Big Data Software Stack

Geoffrey Fox, Judy Qiu, Shantenu Jha
Indiana and Rutgers University

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

- 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



Enhanced **Apache Big Data Stack** ABDS+

- 114 Capabilities
- **Green layers have strong HPC Integration opportunities**

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

Big Data Ogres and Their Facets from 51 use cases

- **The first Ogre Facet captures different problem "architecture"**. Such as (i) **Pleasingly Parallel** – as in Blast, Protein docking, imagery (ii) **Local Machine Learning** – ML or filtering pleasingly parallel as in bio-imagery, radar (iii) **Global Machine Learning** seen in LDA, Clustering etc. with parallel ML over nodes of system (iii) **Fusion:** Knowledge discovery often involves fusion of multiple methods.
- **The second Ogre Facet captures source of data** (i) **SQL**, (ii) **NOSQL** based, (iii) Other Enterprise data systems (10 at NIST) (iv)**Set of Files** (as managed in iRODS), (v) **Internet of Things**, (vi) **Streaming** and (vii) **HPC simulations**.
- **The third Ogre Facet is distinctive system features** such as (i) **Agents**, as in epidemiology (swarm approaches) and (ii) **GIS** (Geographical Information Systems).
- **The fourth Ogre Facet captures Style of Big Data applications.** (i) Are data points in **metric or non-metric spaces** (ii) **Maximum Likelihood**, (iii) χ^2 minimizations, and (iv) **Expectation Maximization** (often Steepest descent).
- **The fifth Facet is Ogres themselves classifying core analytics kernels** (i) Recommender Systems (**Collaborative Filtering**) (ii) **SVM** and Linear Classifiers (Bayes, Random Forests), (iii) **Outlier Detection** (iORCA) (iv) **Clustering** (many methods), (v) **PageRank**, (vi) **LDA** (Latent Dirichlet Allocation), (vii) **PLSI** (Probabilistic Latent Semantic Indexing), (viii) **SVD** (Singular Value Decomposition), (ix) **MDS** (Multidimensional Scaling), (x) **Graph Algorithms** (seen in neural nets, search of RDF Triple stores), (xi) Learning Neural Networks (**Deep Learning**), and (xii) **Global Optimization** (Variational Bayes).

Lessons / Insights

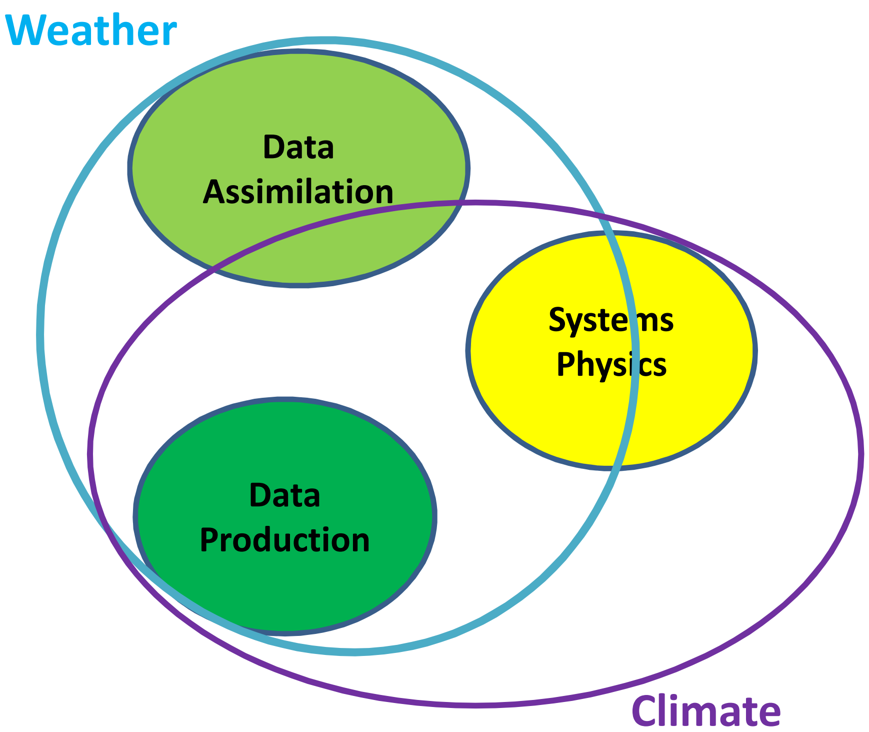
Geoffrey Fox , Judy Qiu (Indiana), Shantenu Jha (Rutgers)

- Please **add** to set of **51 use cases**
- **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 **ABDS+ has 114 members** – please improve!
  - There is a **lot more than Hadoop in ABDS**
  - 6 zettabytes total data; LHC is ~0.0001 zettabytes (100 petabytes)
- **HPC-ABDS+ Integration areas** include **file systems, cluster resource management, file and object data management, inter process and thread communication, analytics libraries, workflow and monitoring**
- **Ogres** classify Big Data applications by **five facets** – each with several exemplars
  - Guide to breadth and depth of Big Data
  - Does your architecture/software support all the ogres?

The role of mini-apps in weather and climate models performance optimization

Giovanni Aloisio, *Jean-Claude André*, Italo Epicoco, Silvia Mocavero
with special thanks to Serge GRATTON, Yann MEURDESOIF and Anthony WEAVER

Scientific and computational issues



Data assimilation
- Minimization
- Ensembles
- **Mini-app 1**

Data production
- I/O
- **Mini-app 2**

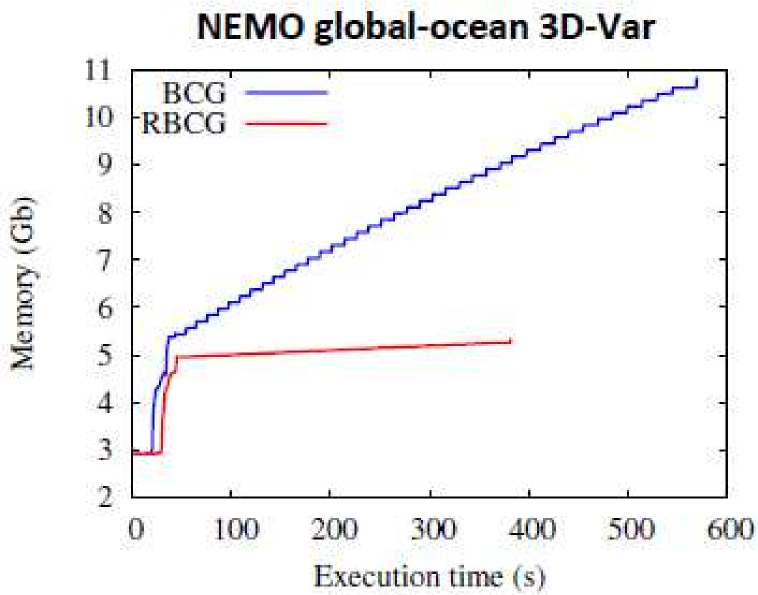
System physics
- Navier Stokes
- Solveurs
- **Mini-app 3**

and other issues
- …
- **Mini-app n**

Issue 1: **Data assimilation**

Estimate the trajectory of a time dependent system using observations: minimize, during an «assimilation window», the distance [cost function $J(\mathbf{x}_a)$] between observed and predicted values. **Issues** : efficiency and scalability of the computation and quantification of the errors on the estimated trajectory. **One way to go** : produce an ensemble of estimates.

For each estimate, use of Krylov subspace methods; reduction of computational cost and memory can be achieved through dual (*observation space*) as opposed to primal (*model space*) methods (efficiency)

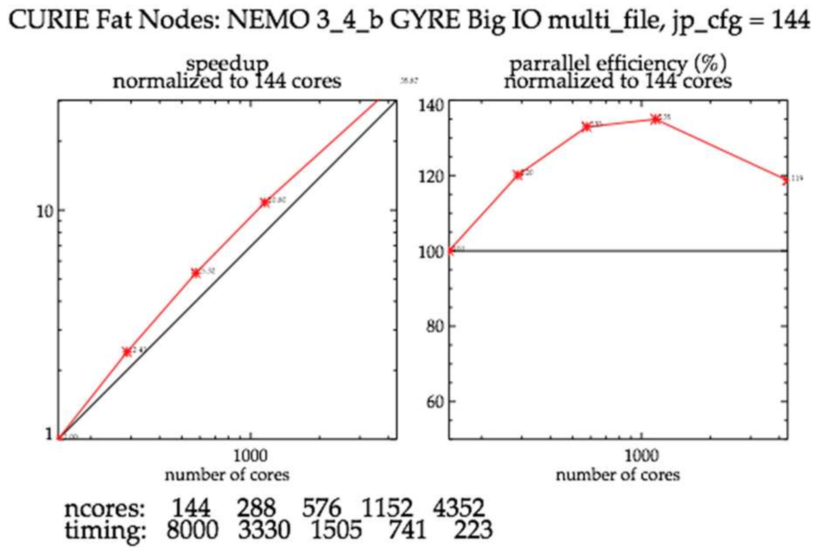


Future developments: Use a perturbed model and perturbed data assimilation system to simulate the evolution of state errors in the system. The ensemble of states provides, among others, a flow-dependent estimate of background error for the data assimilation system (uncertainty quantification)

Issue 2: **I/O**

Need for flexibility (simplification, modularity, …) and performance (more than 10^4 cores, no slowing-down of the computation, …)

Client/server approach, using asynchronous call for outsourcing I/O definition and minimizing I/O calls, number of calling arguments, …



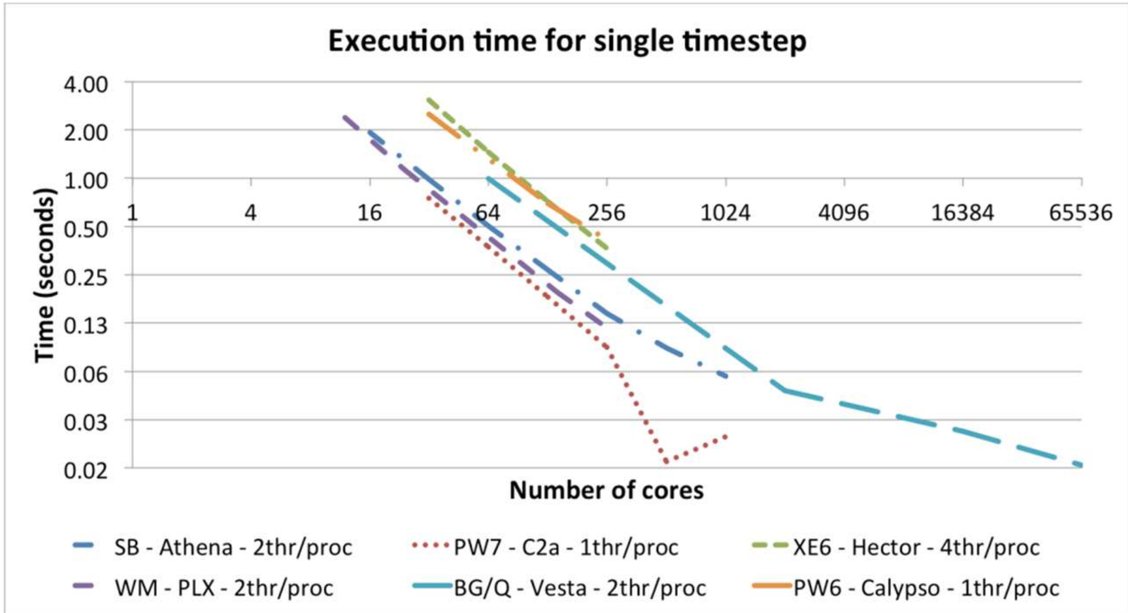
Nemo-XIOS : scaling up to 8160 core with 128 XIOS servers

Future developments :
Online post-processing to benefit from the full parallel ressources and to reduce field combination, remapping, means and interpolation

Issue 3: **Solvers**

Example: Advection schemes for evaluating the divergence of the tracers' advective fluxes. It is commonly implemented with finite different method with a 5-points stencil communication pattern

Reduction of the communication overhead of the MPI implementation exploiting the shared memory -> hybrid parallel approach (MPI/OpenMP)



Future develpments: hybrid approaches can exploit high-end machines equipped with 'accelerators'

High-performance Software Stacks for Extremely Large-scale Graph Analysis System

Katsuki Fujisawa, Toyotaro Suzumura, Hitoshi Sato, Toshio Endo

High performance Software Stacks for Extremely Large-scale Graph Analysis System by K.Fujisawa et al.

- **The extremely large-scale graphs that have recently emerged in various application fields**
  - US Road network : 58 **million** edges
  - Twitter fellow-ship : 1.47 **billion** edges
  - Neuronal network : 100 **trillion** edges
- **Fast and scalable graph processing** by using **HPC**

Social network

Twitter

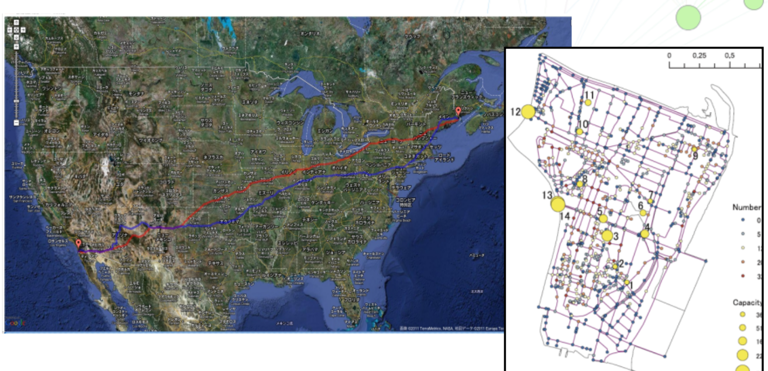
61.6 million nodes & 1.47 billion edges

Neuronal network @ Human Brain Project

89 billion nodes & 100 trillion edges

US road network

24 million nodes & 58 million edges



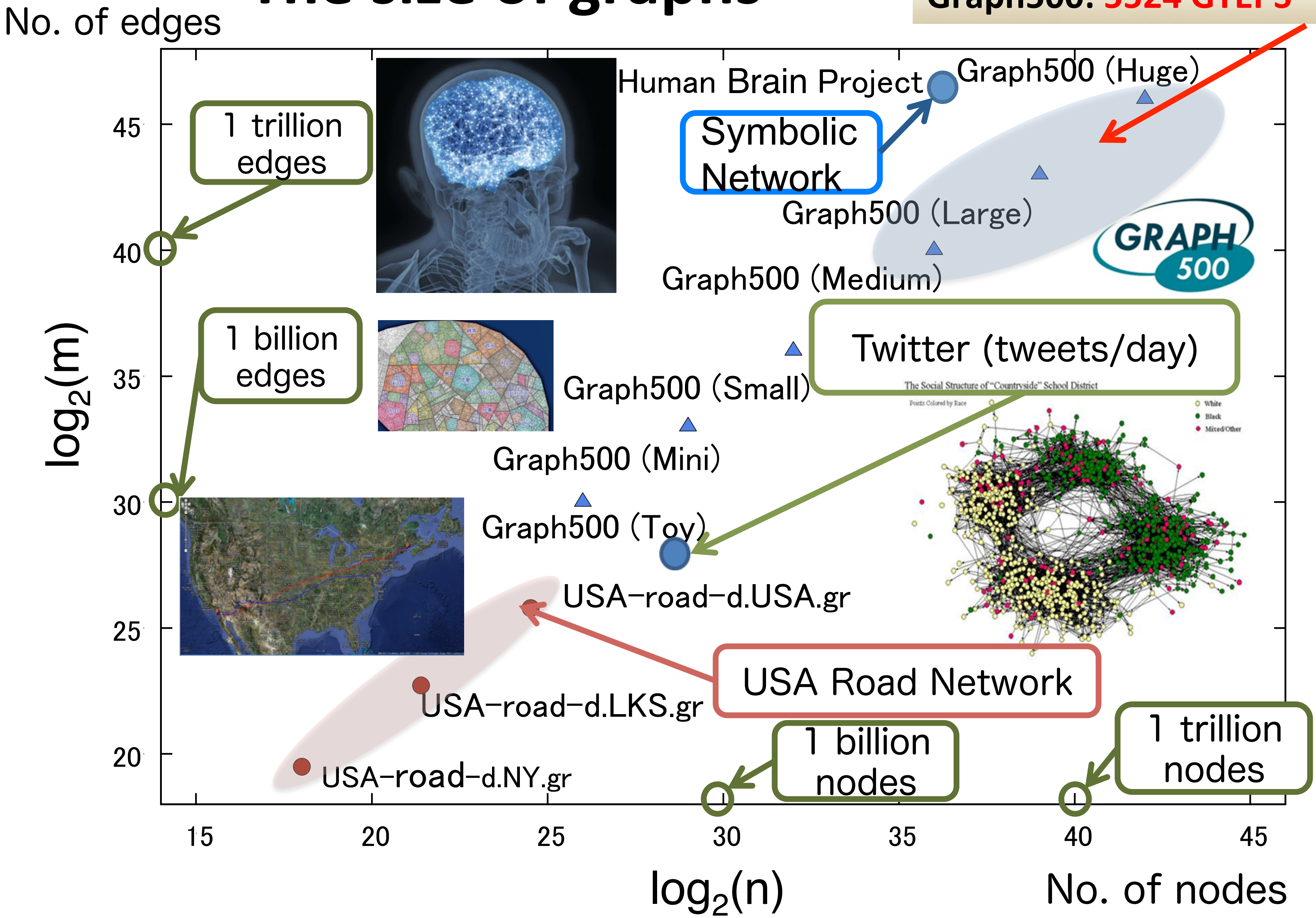
Cyber-security

15 billion log entries / day

Image: Illustration by Mirko Ilic

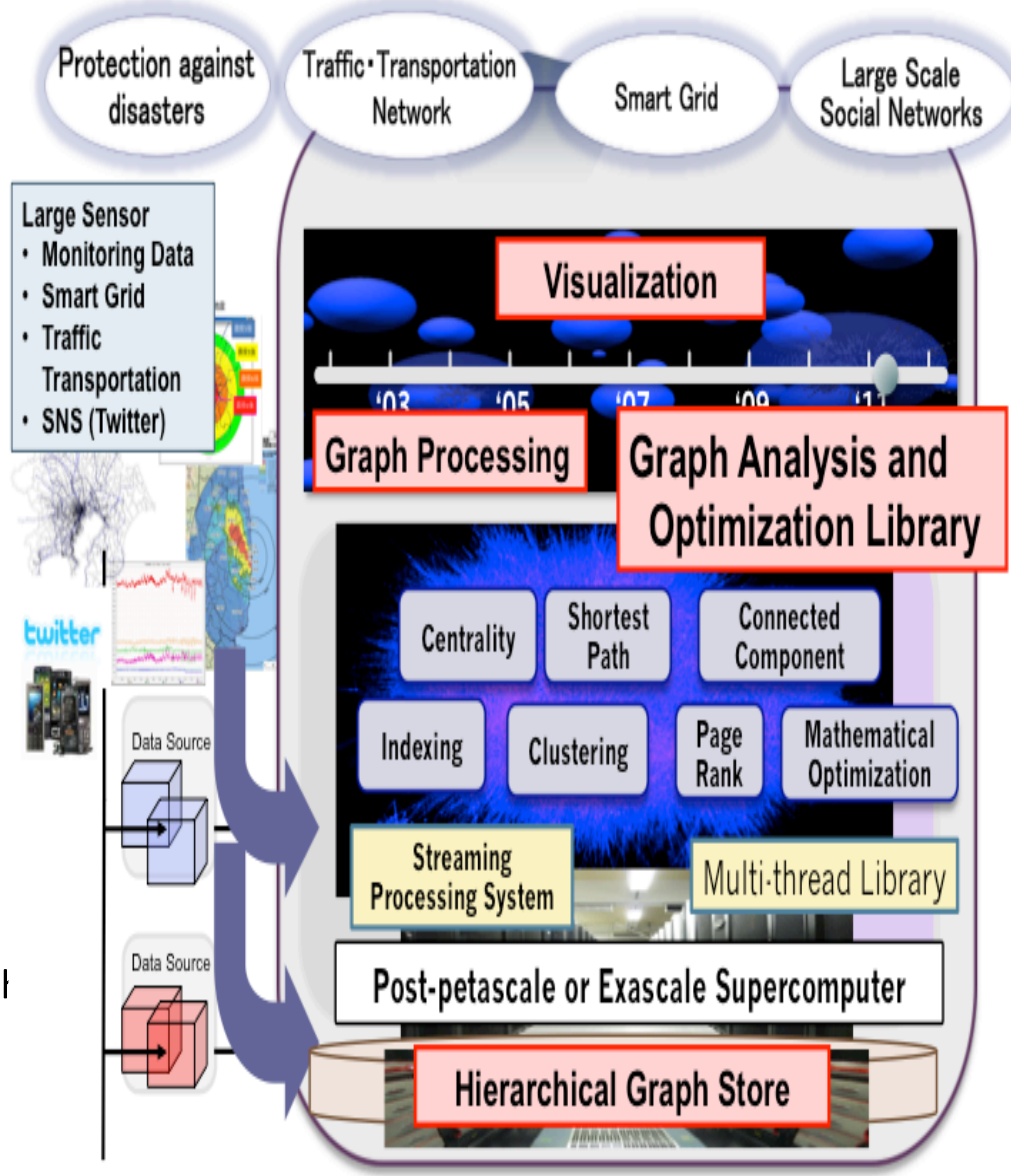
The size of graphs

K computer: 65536nodes
Graph500: 5524 GTEPS



Software stacks for an extremely large-scale graph analysis system

- **Hierarchal Graph Store:**
- Utilizing emerging NVM devices as extended semi-external memory volumes for processing extremely large-scale graphs that exceed the DRAM capacity of the compute nodes
- **Graph Analysis and Optimization Library:**
- Perform graph analysis and search algorithms, such as the BFS kernel for Graph500, on multiple CPUs, GPUs, and Xeon Phis.
- **Graph Processing and Visualization:**
- We aim to perform an interactive operation for large-scale graphs with hundreds of million of nodes and tens of billion of edges.



Large — Comp. time — small

Small — Data size — Large

Upper layer : Optimization algorithms for NP-hard problems

MIP(Mixed integer problem) : No. of 0-1 integer variables = n → $O(2^n)$

1. Parallel branch and cut (bound) algorithm and MPI + pthreads parallel computation
2. Data size : $n <= 10^5$
3. Facility location problem, Set covering (partitioning) problem, Scheduling

Middle layer : Polynomial time optimization algorithms

SDP(Semidefinite programming problem) : n = matrix size, m = no. of constraints → $O(n^3 + m^3)$

1. Exploiting sparsity, MPI + OpenMP parallel computation using multiple CPUs and GPUs
2. Data size : $n <= 10^8$, $m <= 10^6$
3. Graph partitioning problem, Sensor allocation, Data mining (Support vector machine)

Lower layer : Graph and network analysis algorithms

Dijkstra algorithm (Single source shortest path problem with 2-heap) : n = no. of nodes, m = no. of edges → $O((n + m)\log n)$

BFS (Breath first search algorithm) : n = no. of nodes, m = no. of edges → $O(m)$

1. Data size : n and $m <= 10^{12\sim14}$
2. Shortest path, Centrality(BC etc.), Clustering problem

On Next Generation Big Data Analytics Systems

Volker Markl
http://www.user.tu-berlin.de/marklv/

Data & Analysis: More and More Complex!

scalability / DM	data volume too large data rate too fast data too heterogeneous	***V**olume* ***V**elocity* ***V**ariability*	Reporting Ad-Hoc Queries ETL/Integration	aggregation, selection SQL, XQuery map/reduce	DM / scalability
algorithms / ML	data too uncertain	***V**eracity*	Data Mining Predictive/Prescripive	Matlab, R, Python Matlab, R, Python	ML / algorithms

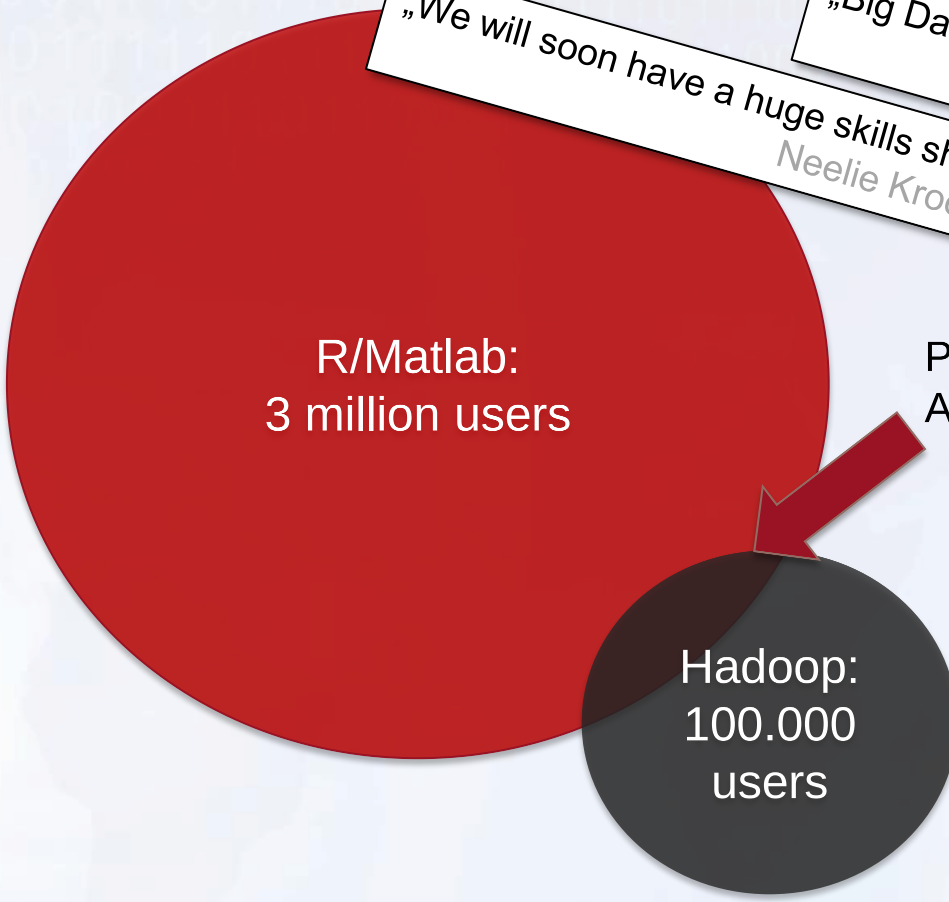
Data | **Analysis**

The real scalability problem of Big Data:
Big Data Analytics requires Systems Programming!

„Big Data's Big Problem: Little Talent“
Wall Street Journal

„We will soon have a huge skills shortage for data-related jobs.“
Neelie Kroes (ICT 2013, Nov. 7, Vilnius)

Data Analysis
Statistics
Algebra
Optimization
Machine Learning
Clustering
Regression
SVM
Dim. Reduction
NLP
Signal Processing
Image Analysis
Audio-,Video Analysis
Information Integration
Information Extraction



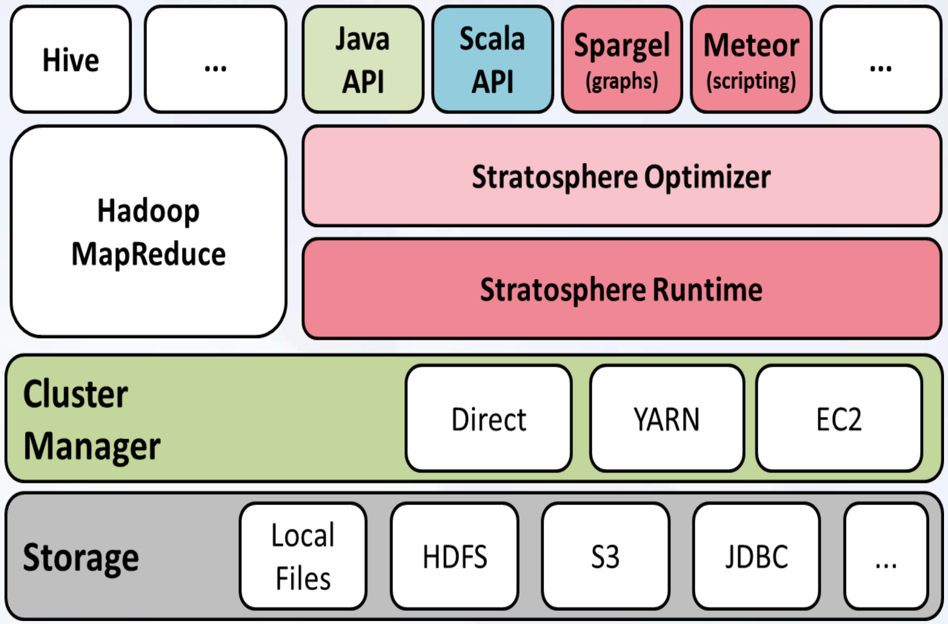
People with Big Data Analytics Skills

Indexing
Parallelization
Communication
Memory Management
Query Optimization
Efficient Algorithms
Resource Management
Fault Tolerance
Numerical Stability

Big Data is now where database systems were in the 70s (before relational algebra and SQL)!

We need a declarative language with automatic optimization, parallelization and hardware adaptation

Stratosphere is a next generation Big Data Analytics platform with automatic parallelization, optimization and hardware adaptation!



http://www.stratosphere.eu

- Many APIs
  - Generic Java, Scala
  - Graph
  - Scripting
  - Under development: Python, SQL
- Iterative Programs
  - Bulk (batch-to-batch in memory) and Incremental (Delta Updates)
- Automatic data flow optimization
  - For iterations: automatic caching and cross-loop optimizations
- Fast runtime (in-memory and out-of-core)
- In-situ analytics
- Plugs easily into Hadoop ecosystem (HDFS, YARN)
- Apache Open Source 2.0
- Growing user Community in Europe and beyond

"Science Automation using Workflows in the Big Data and Extreme Scale Computing Era"

Key idea:
Need for multiple, customized, collaborating WMS for BD (ex-situ) and EC (in-situ)

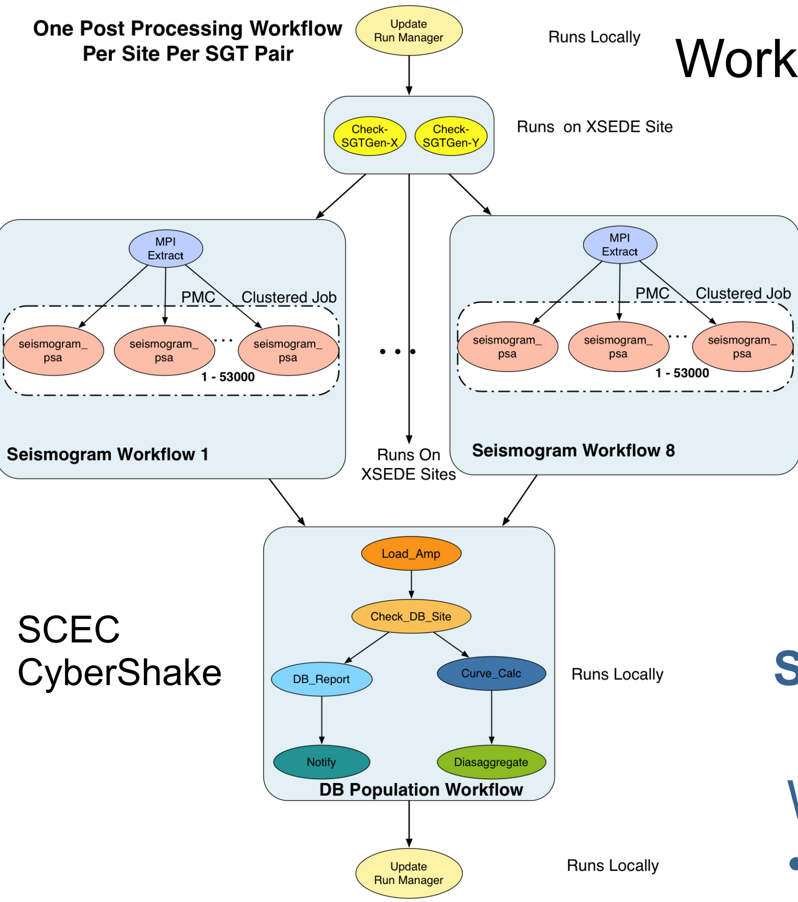
Ewa Deelman

University of Southern California

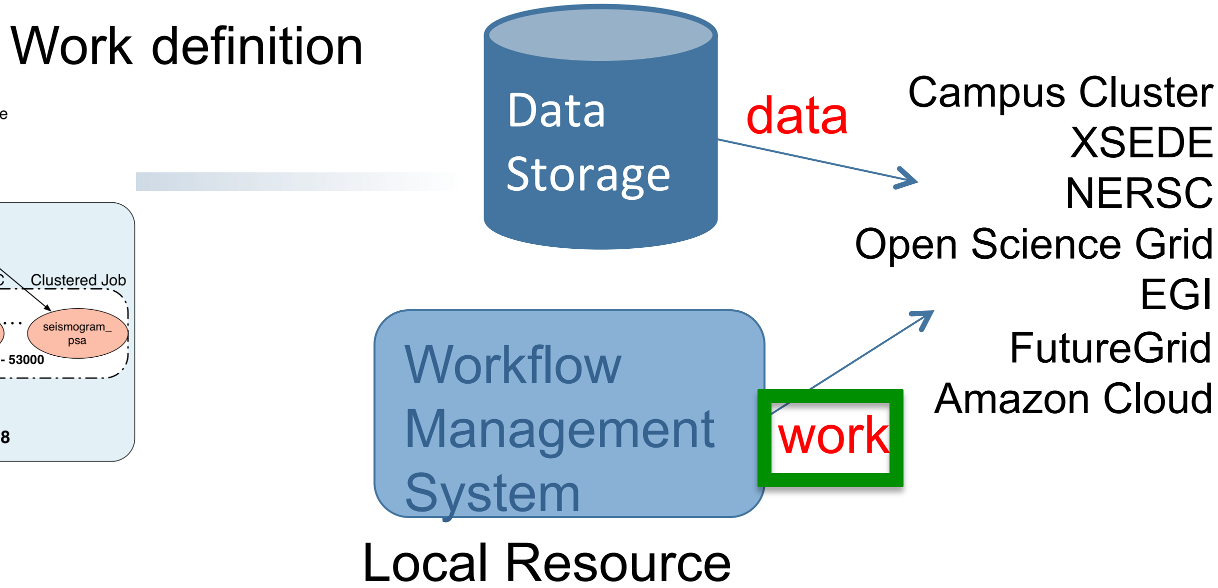
Information Sciences Institute

deelman@isi.edu
http://www.isi.edu/~deelman

Pegasus Workflow Management System
http://pegasus.isi.edu



SCEC
CyberShake



Separation of Workflow Definition and Execution

1,144 geographic locations
Uses Pegasus with execution on TACC's Stampede
~ 470 million tasks total
Over 739 hours of computing
~ 636,000 tasks per hour
45 TB of data
12 TB being staged back for archiving

Workflow ensembles:

- Today workflows are managed on an individual basis
- As science is scaling up, it is necessary to manage entire workflow ensembles.
- Opportunity to optimize data transfers, reuse, and storage, across the wide area and inside EC systems.

Tom Jordan, USC

Applications will be managed by multiple Workflow Management Systems

- Workflow Management Systems can potentially bridge the gap between big data and extreme scale computing
- Data needs to be staged to the EC resources and staged back
- Computations can involve multiple EC resources
- For efficiency a workflow management system may need to work *in situ* on an EC resource, coordinating fine-grained computation scheduling and data movement across the machine
- There needs to be a delegation of work or collaboration on workload management between BD WMS and EC WMS
- Each WMS needs to tailor and optimize the workflow execution to each specific environment
  - data and computation management decisions that occur inside an EC need to take into account energy efficiency, and thus data locality among others
- Need to worry about reliability and reproducibility
- Need to worry about interactivity with both types of WMSs

Interplay between BD and EC WMS needs to be explored

- Restructuring of the workflows for different environments
- Common capabilities that need to work together:
  - provenance capture (and linking), reliability, and performance
  - need tools for efficient provenance storage and query
- Data management at different scales
  - EC WMS
    - may deal with data in memory
    - potentially streaming data from/to the EC resource
    - Makes use of HPC libraries
  - BD WMS may
    - select to the best replica from a set of possible data repositories, select services, ECs
    - consider the proximity of computing to these storage resources.
    - trigger computations based on the influx of new data products
  - EC WMS may provide hints to BD WMS on how to stage the data into the extreme scale system
  - EC WMS may also give hints about how the output data is structured, or how it is streamed so that the BD WMS can reconstruct the results of the computation.

Workflows need to be easy to compose, reuse, launch, monitor, and interpret --- all from scientist's desktop.

Indiana/swany/BDEC/*ideas*

- Explore and document the problems
- Consider and expand the space of possible solutions

Much of the I/O gap is due to software

Unbundle, rework the notion of "files"

Data(Management+ Manipulation) => Efficiency is the critical aspect

Files and Programmer Productivity

- We must revisit basic abstractions around files and filesystems as the current file model is often less than ideal
- Programmer productivity is aided by thinking in terms of files and folders (and filing cabinets and banker's boxes on shelves and sticky notes)
- One set of abstractions doesn't "fit all" needs

Metadata

- Pages and blocks, cached and manipulated in various memories
- Long term storage with perhaps replication, compression, coding
- Metadata needs change over time given factors like fragmentation, metadata overhead, false sharing, real sharing, TLB

The Exofiles Manifesto

- Unbundle and reinvent the file by moving as much functionality as possible into a runtime
- Change the execution model to include a dramatically different notion of data: memory/storage/files/filesystems
- Potential to remove barriers to performance while increasing productivity with a rich, "unified view" of data