BERKELEY LAB
Bringing Science Solutions to the World

U.S. DEPARTMENT OF ENERGY

Scientific Data Services Framework for Exascale Infrastructure

John Wu
Lawrence Berkeley National Laboratory

http://crd.lbl.gov/sdm/

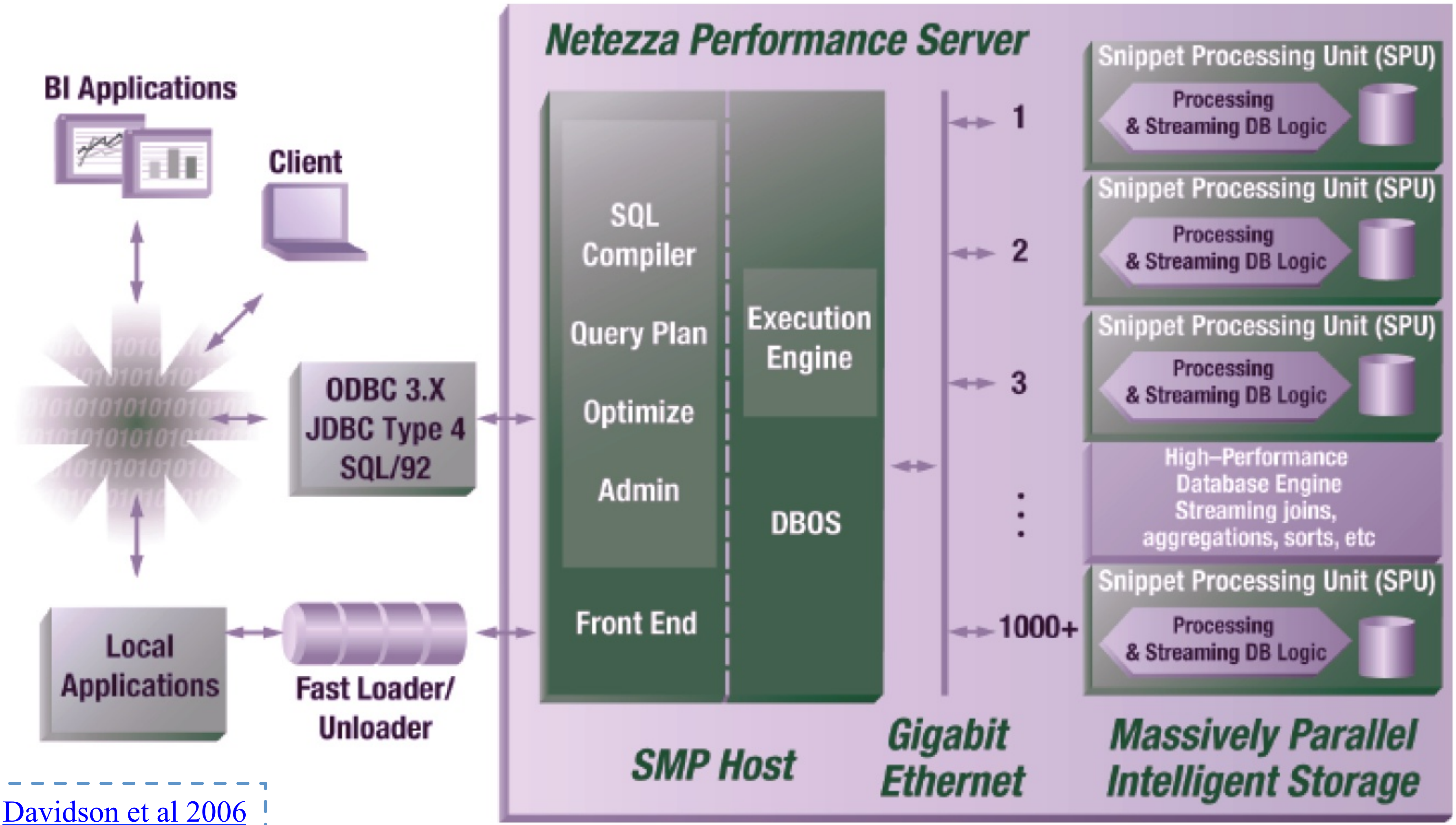
What do you think of when you hear Big Data?

Scientific Computing also Uses Many CPUs

BERKELEY LAB

Could scientific data analysis use Big Data software systems?

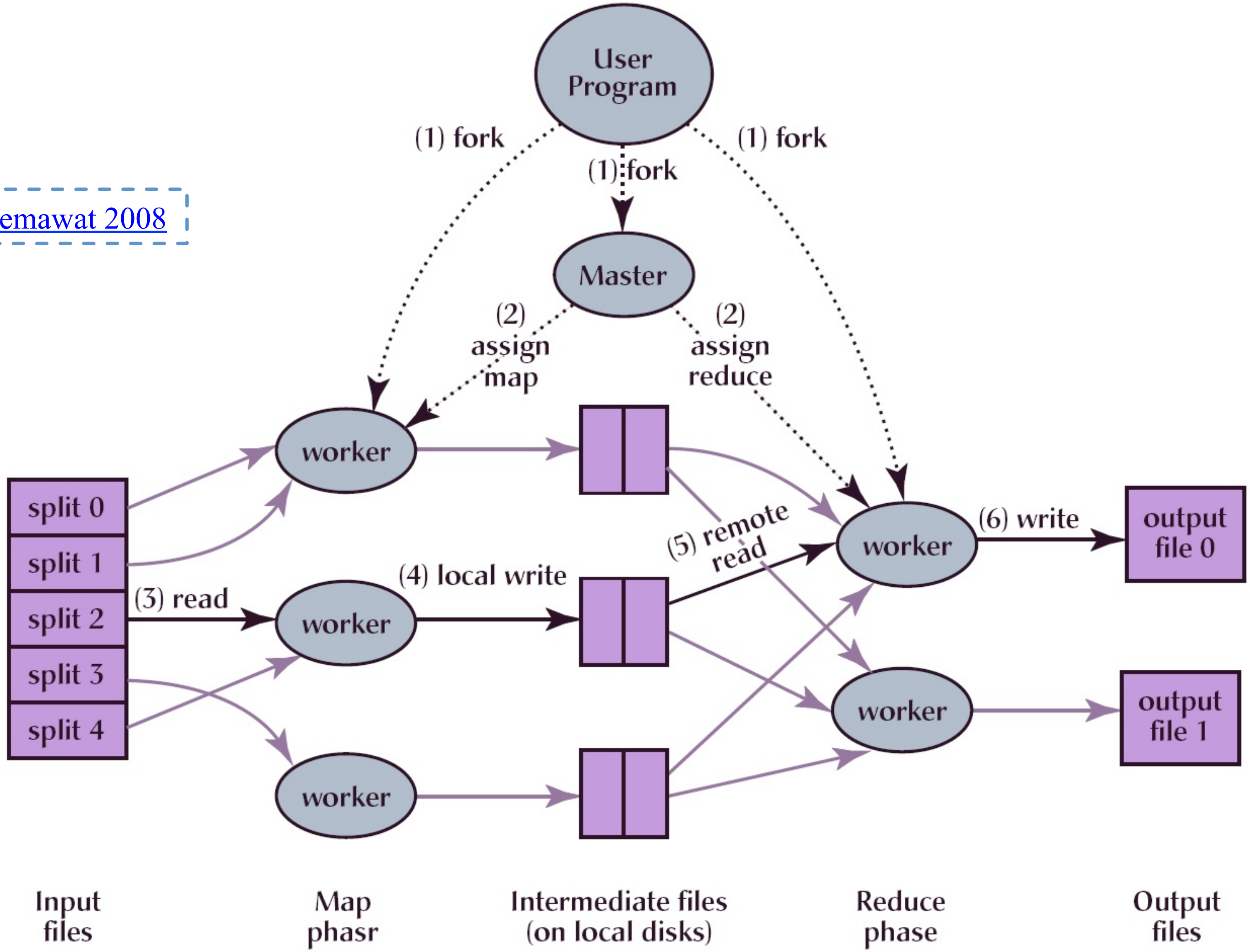
Big Data System
-- Parallel Database Systems



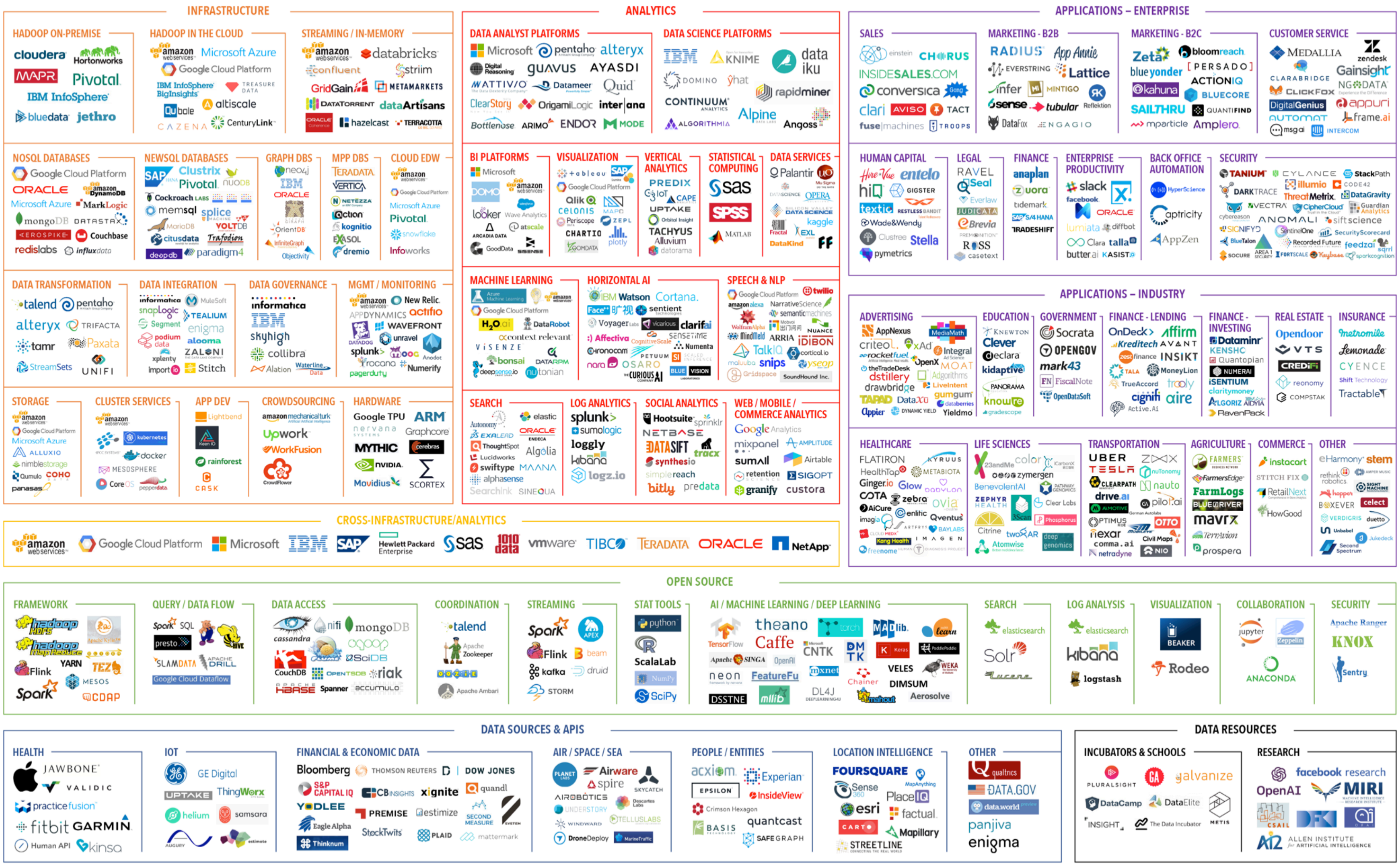
Davidson et al 2006

Big Data System -- MapReduce

Dean and Ghemawat 2008



Big Data Has a Vibrant Software Ecosystem



http://mattturck.com/bigdata2017/

Can scientific data analyses make effective uses of Big Data software?

Let’s take a look at some examples of scientific data analyses…

Example 1: Combustion Ignition Kernels

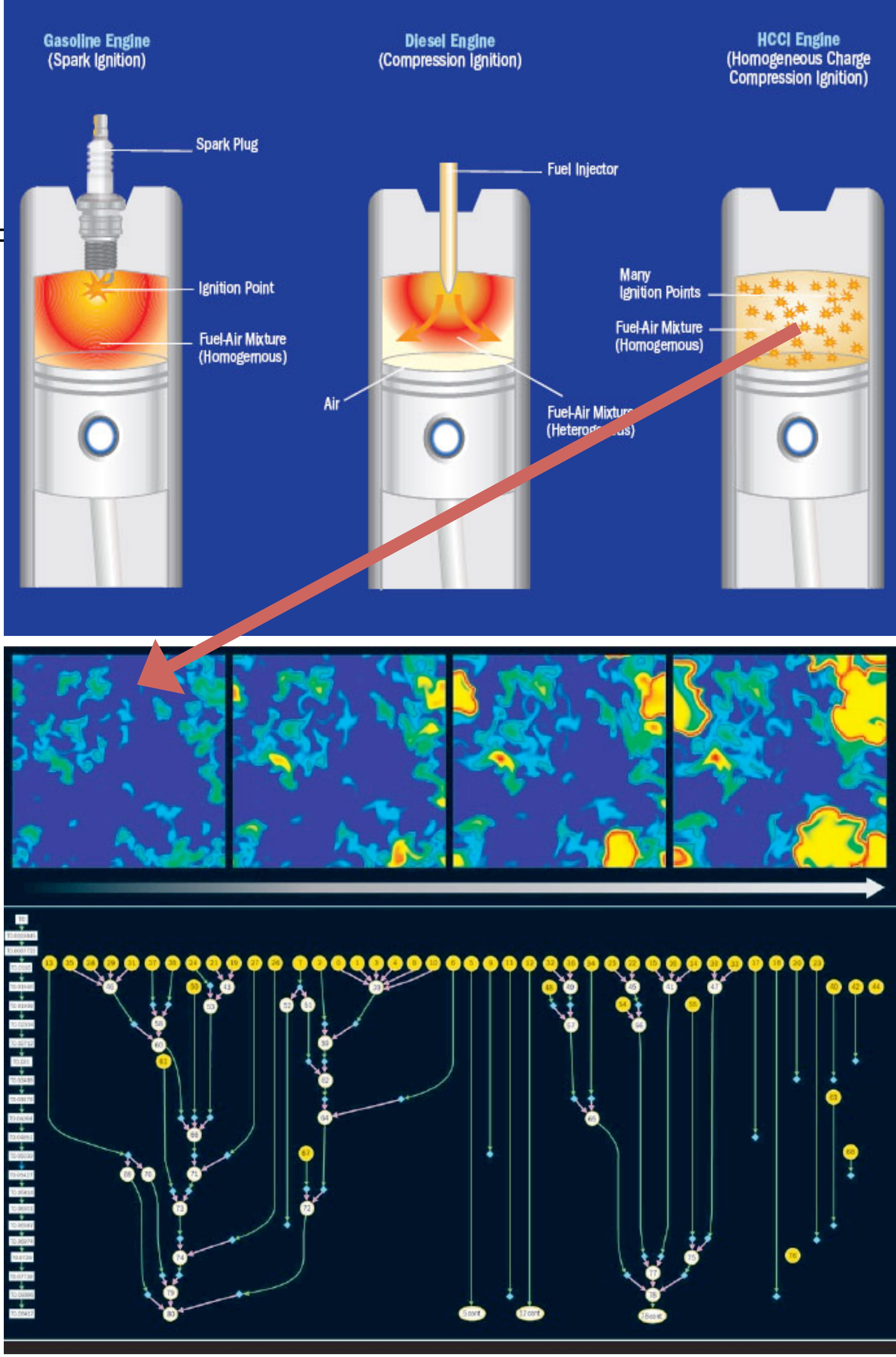
Simulation of Homogeneous Charge Compressed Ignition engine

1000x1000x1000 cube mesh

10000s time steps

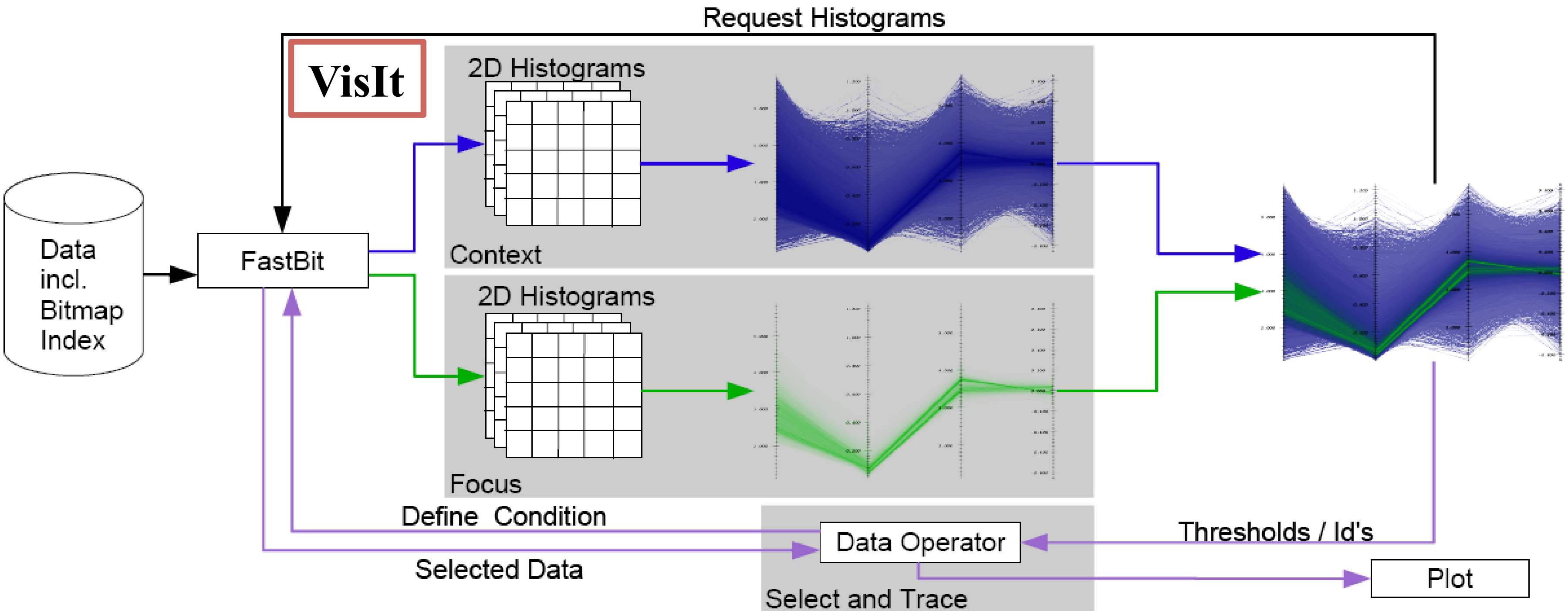
Hundreds of variables per mesh point to describe realistic diesel-air mixture

Example data analysis task: tracking the ignition over time



Wu, Koegler, Chen, Shoshani 2005

Example 2: Particles in Accelerator Modeling

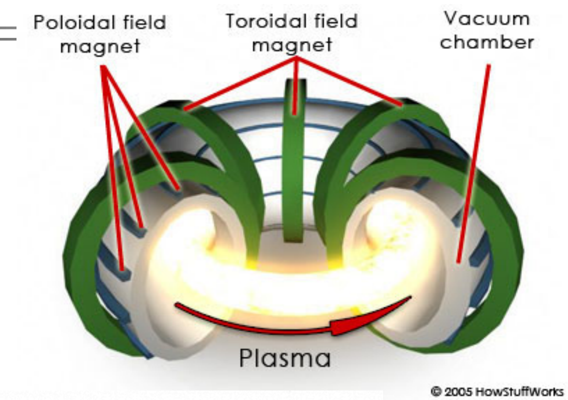


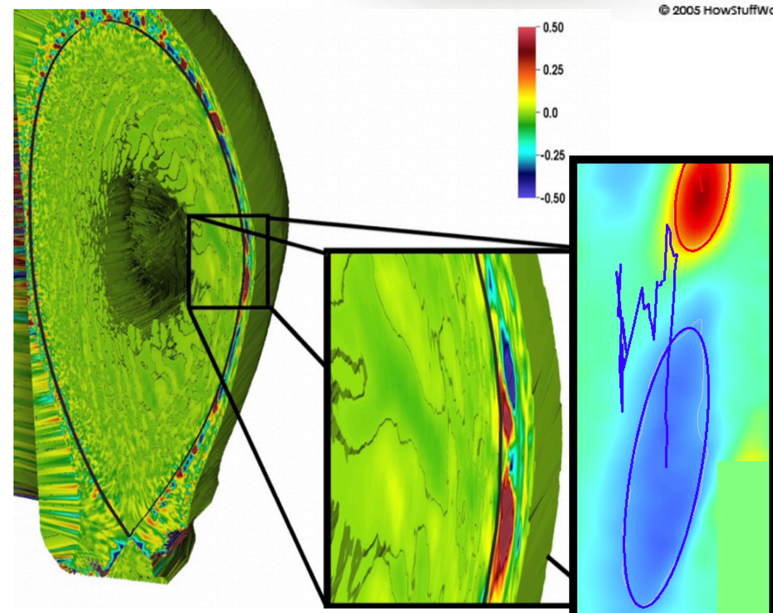
- ❑ Billions of particles produced from modeling of Laser Wakefield Particle Accelerators
- ❑ Sample analysis tasks:
  - ✧Find 1000s most energetic particles
  - ✧Track the progression of the particles

Ruebel et al SC08

Example 3
Near Real Time Detection of Fusion Blobs

- ❖ Plasma blobs
  - Lead to the loss of stability and/or confinement of tokamak plasmas
  - Cause fast thermal and/or current quench
  - Could damage multi-billion tokamak
- ❖ The experimental facility may not have enough computing power for the necessary data processing
- ❖ Distributed in transient processing
  - Make more processing power available
  - Allow more scientists to participate in the data analysis operations and monitor the experiment remotely
  - Enable scientists to share knowledge and processes
- ❖ Wu, et al. 2016





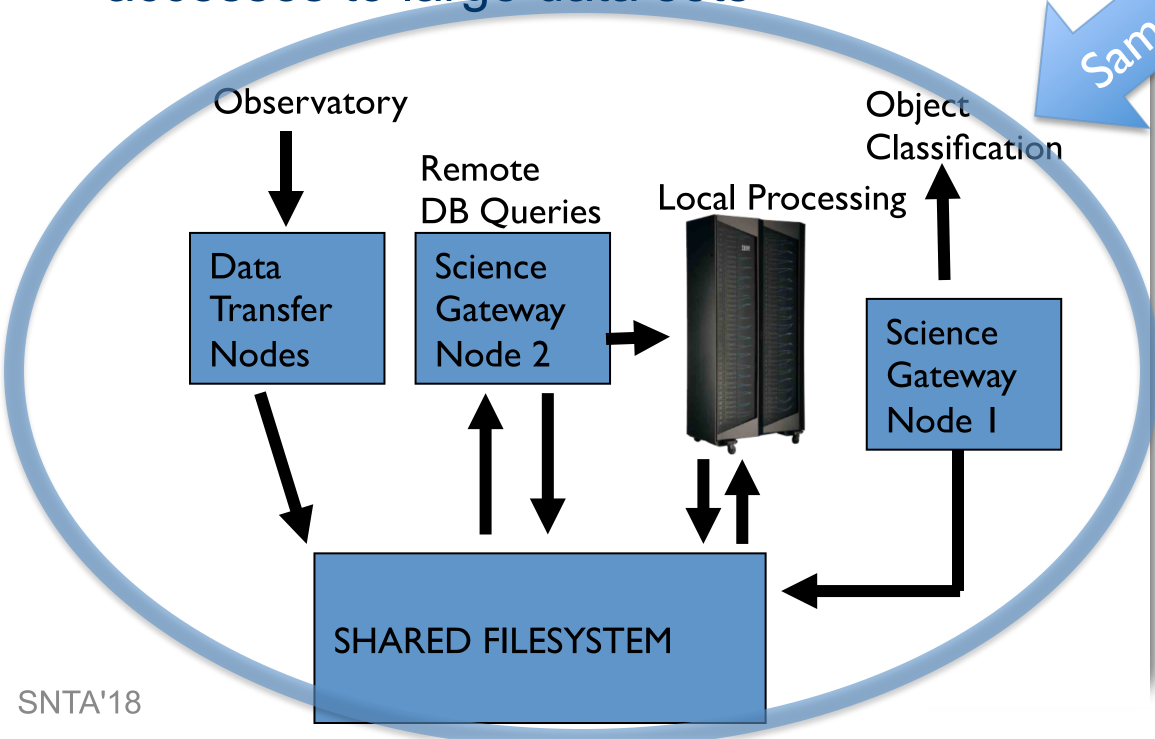
Blobs in fusion reaction (Source: EPSI project)

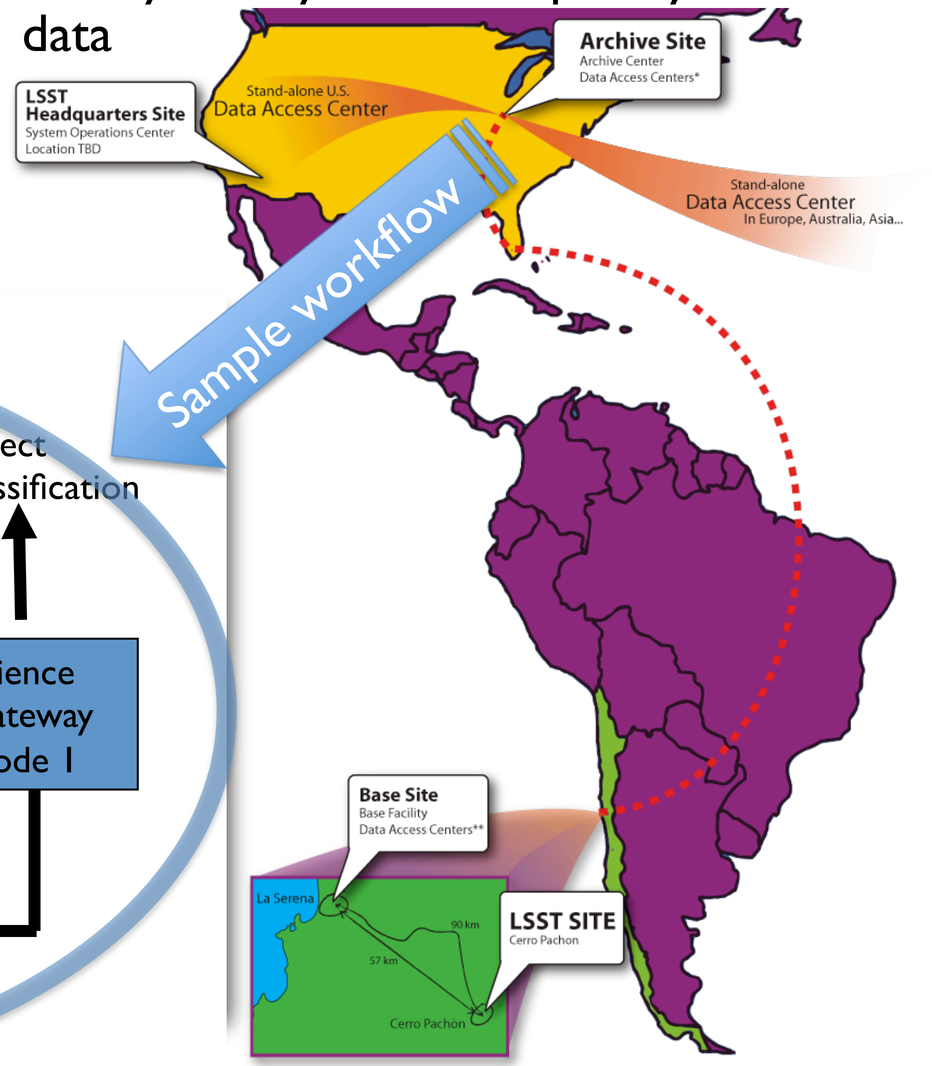
Blob trajectory

Example 4
Astronomic Observations

- Ease-of-use: scientifically-meaningful abstraction, intuitive API
- Functionality: support a wide variety of data access patterns
- Performance: efficient parallel data accesses to large data sets

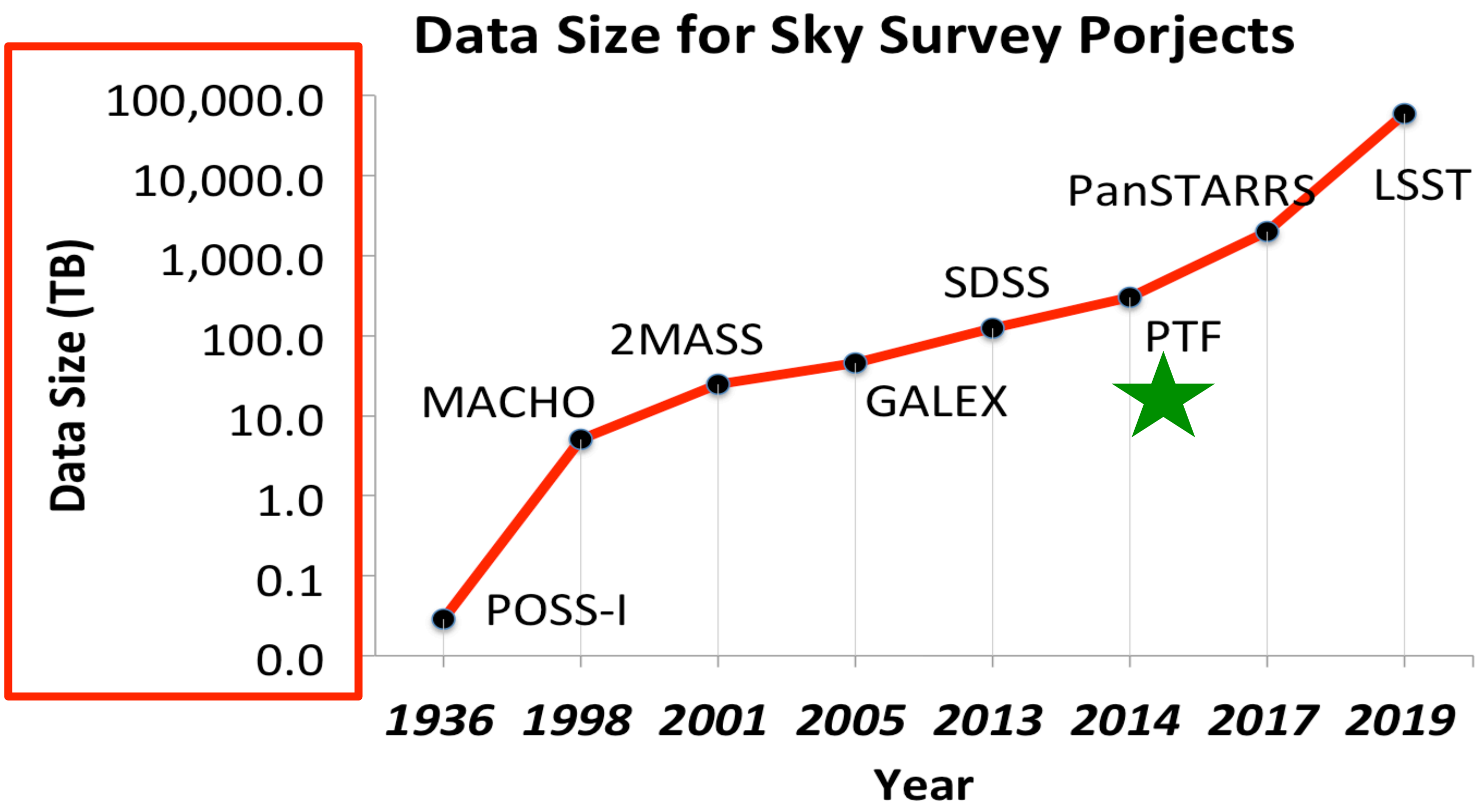
Large collaborations need to support a variety of way to access petabytes of data





Scientific activities evolve into big data analysis

Example: scientific projects for supernovae, dark matter/energy, etc.



Data source:
Rick White,
J. Hart,
R. Cutri,
Ian Foster,
C. J. Grillmair,
etc.

Data Management In Service Of Big Sciences
-- A Reflection on Needs of Exascale Program

Background: From ASCR Data Crosscutting Requirements Review (April, 2013)

Finding 1: The challenges associated with scientific data are diverse and often distinct from challenges in other data-intensive domains, such as web analytics and business intelligence.

Finding 2: Research communities across the Office of Science have considerable expertise in the aspects of data science necessary for performing their science.

Finding 3: Many Office of Science experimental facilities anticipate rapid growth in data volume, velocity, and complexity.
[this applies to simulation data as well]

Finding 4: Currently, many scientific facilities expect users to manage their own data.

Finding 5: There is an urgent need for standards and community application programming interfaces (APIs) for storing, annotating, and accessing scientific data.

U.S. DEPARTMENT OF ENERGY
Office of Science

What can scientific data management research learn from Big Data software?

- Separate data management from data analyses
- Develop data model for science
- Support complex data access patterns

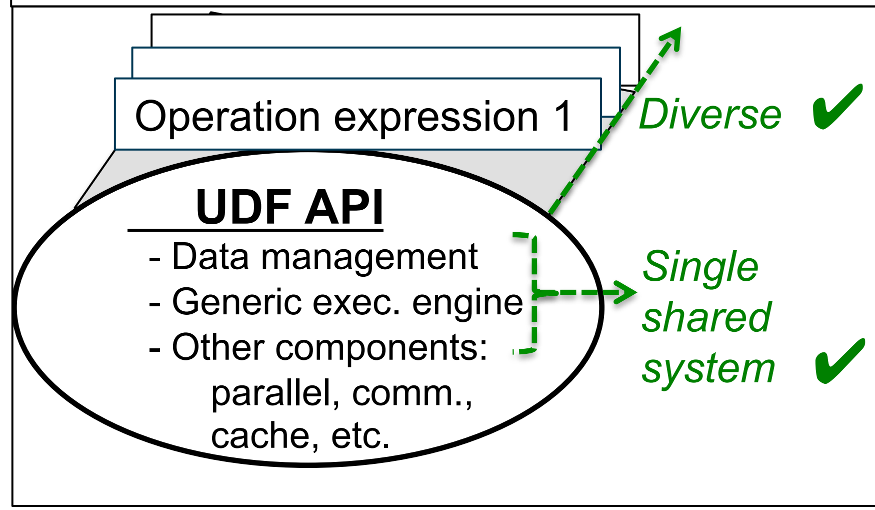
Lesson 1
Separate Data Management from Data Analyses

Customized Solutions

For each operation *P* **Do**
Develop *P*'s :
- Data management ---> *Redundant* ✗
- Expression execution ---> *Diverse* ✔
- Other components: ---> *Redundant* ✗
  parallel,
  communication
  cache,
  etc.

End For

User-defined Functions (UDF)



Scientific data analyses typically are custom programs ✗

Big Data systems separate data management from data analyses ✔

Key: UDF needs a well-defined data model, e.g., key-value pairs in MapReduce, and tuples in Database systems

What can scientific data analyses learn from Big Data software?

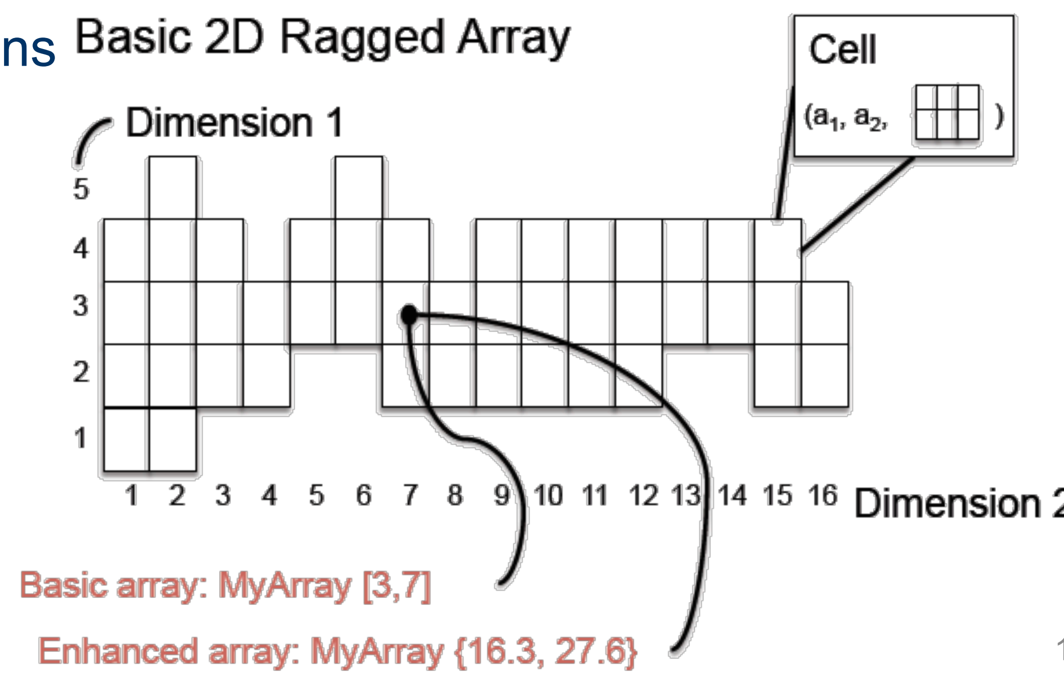
- Separate data management from data analyses
- Develop data model for science
- Support complex data access patterns

Scientific Data is Stored in Arrays

Approach 1: a database system for scientific applications, e.g., SciDB

SciDB features:

- Array-oriented data model
- Append-only storage
- First-class support for user-defined functions
- Massively parallel computations

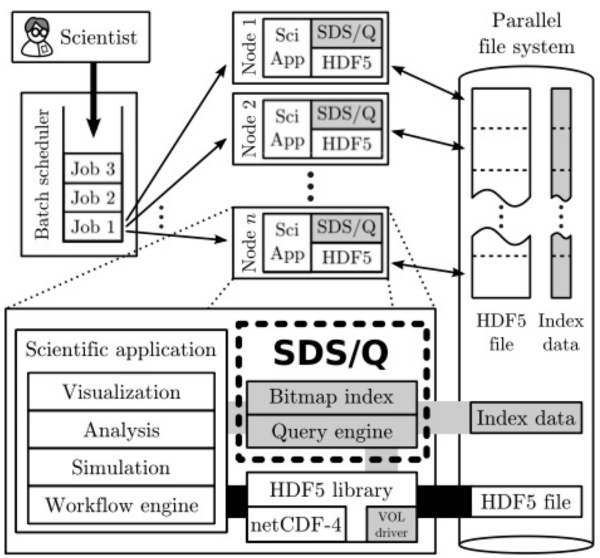


Cudre-Mauroux et al 2009

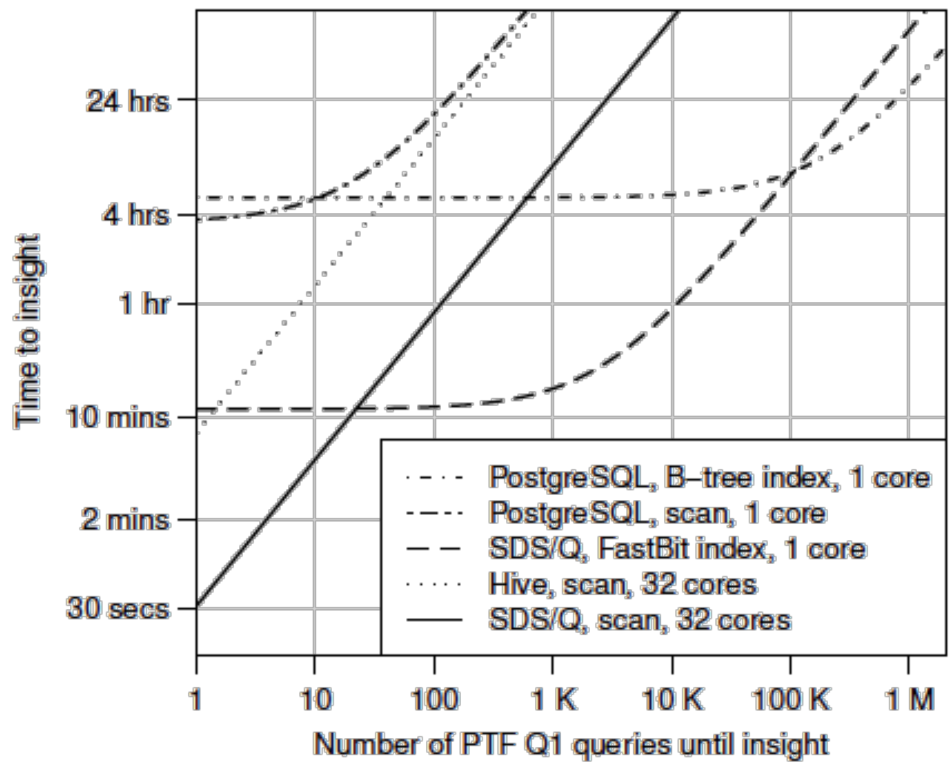
Scientific Data is Stored in Files

Approach 2

- **Relational parallel query processing directly on scientific file formats**
- **Using database technology requires costly loading of data and converting results**



Overview of SDS/Q, the querying component of the Scientific Data Services framework.



Time to insight for a PTF query: 150X faster than PostgreSQL and 10X faster than Hive

S. Blanas, K. Wu, S. Byna, D. Bin, A. Shoshani, SIGMOD 2014

What can scientific data analyses learn from Big Data software?

- Separate data management from data analyses
- Develop data model for science
- Support complex data access patterns
  - Accessing neighbors
  - Selective special records

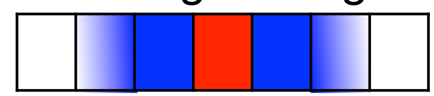
MapReduce Not Optimal for Scientific Data Analyses

Reason 1: most scientific data are multi-dimensional arrays

→ Converting array to (key, value) is expensive

Reason 2: most scientific data analysis operations need to access neighbors

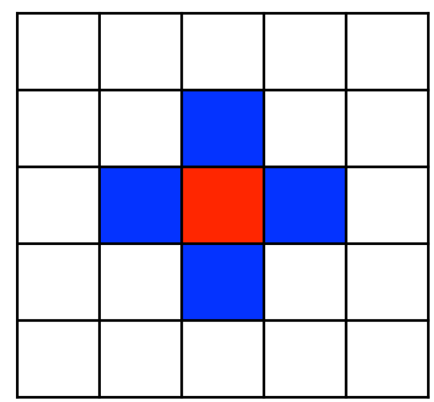
Moving Average



Structure locality:

The analysis operation on a single cell accesses its neighborhood cells

2D Poisson Equation Solver (Discrete)



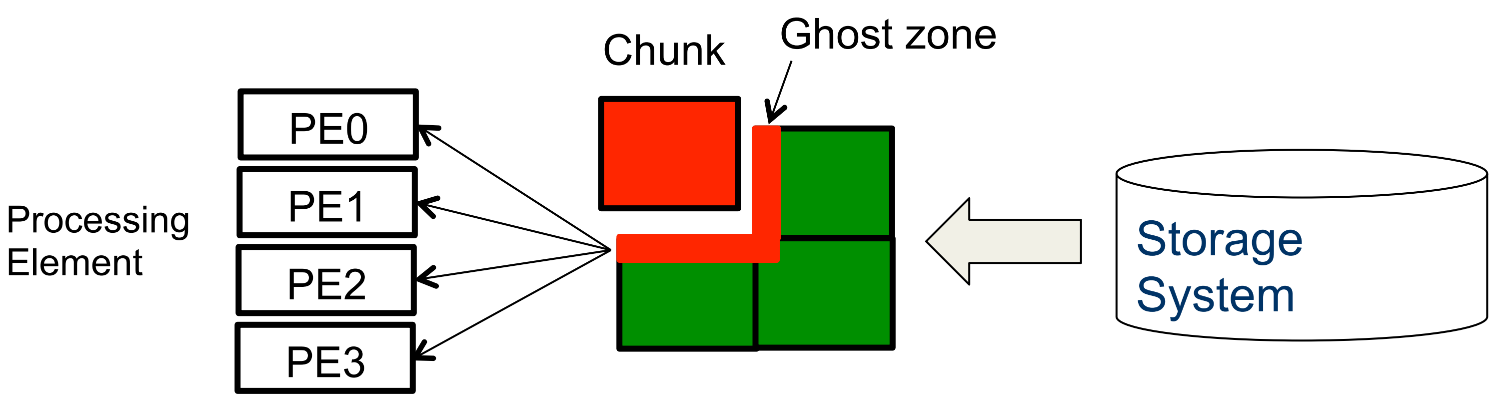
→ Map deals with a single element at a time

→ Reduce requires to duplicate each cell for all neighborhood cells

→ Reduce only happens after expensive shuffle

ArrayUDF: user-defined scientific data analysis on arrays

- Stencil-based user-defined function
  ⇨ Structural locality aware array operations
- Native multidimensional array data model
  ⇨ In-situ data processing in scientific data formats, e.g., HDF5
- Optimal and automatic chunking and ghost zone handling method
  ⇨ Fast large array processing in parallel & out-of-core manner

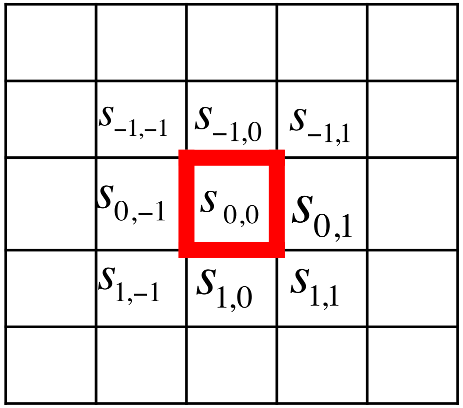


Stencil-based UDF

- Stencil is a set (S) of neighborhood cells
  - *The S has a* center where computing happens
  - The size of $|S|$ is not fixed
  - Notations for set member

    $s_{\delta 1, \delta 1, \cdots}$ stands for the cell at *offset* $\delta_1, \delta_2, \cdots$ *from center point* $i, j, \cdots$

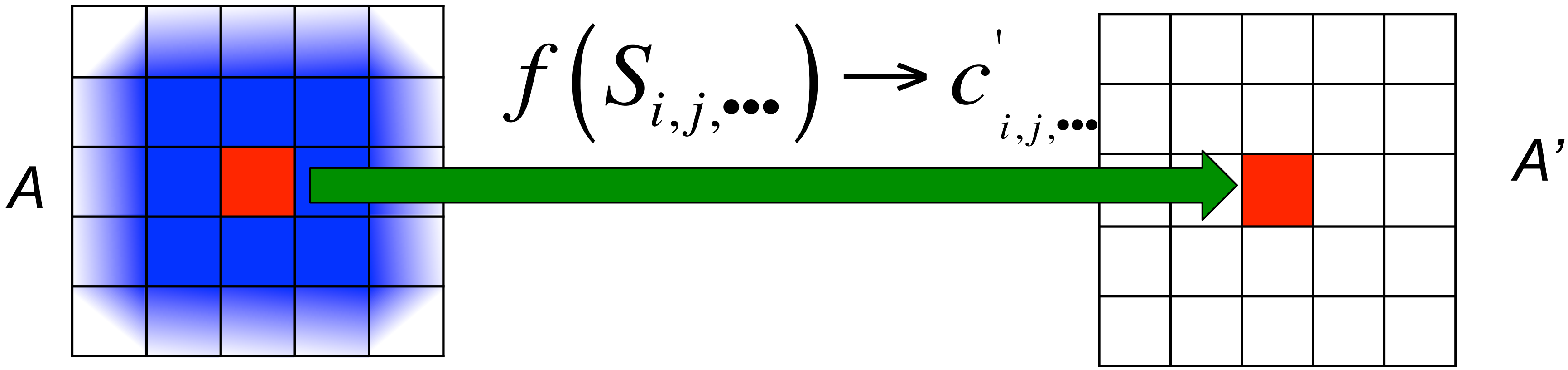
2D Example:



➡ Materialized structure locality

➡ Flexible UDF expression by manipulating each neighborhood cell independently

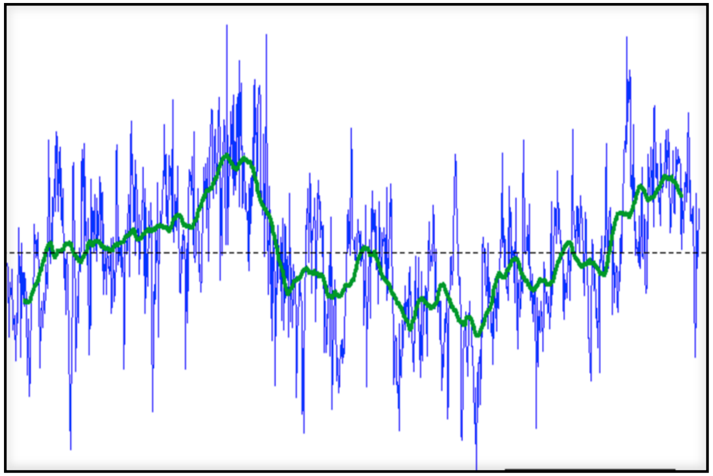
Stencil-based UDF(continued)



- f is arbitrary user-defined function
- *Input S* is Stencil representing set of neighborhood cells
  - |*S*| = 1, user-defined function of a single cell
    i.e., map in MapReduce
  - |*S*| > 1, user-defined aggregation of a set of cells,
    i.e., reduce in MapReduce

Examples of using ArrayUDF

Example 1: moving average in time series data



Global temperature trend filtered by moving average at 60 years' interval from 1908 to 2008

Three steps by using ArrayUDF:

Step 1: Initialize data

Array *T*("data location pointer")

Step 2: Define operation on Stencil

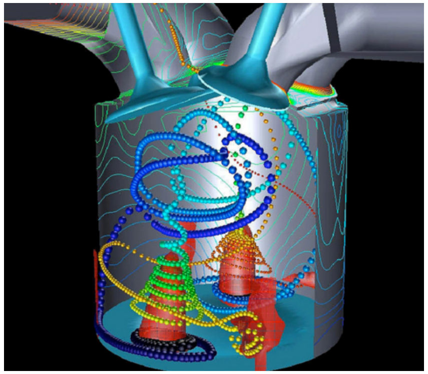
Tem_avg(Stencil *t*):
  return (*t*(-30)+ … *t*(30))/60

Step 3: Run & get result *T'*

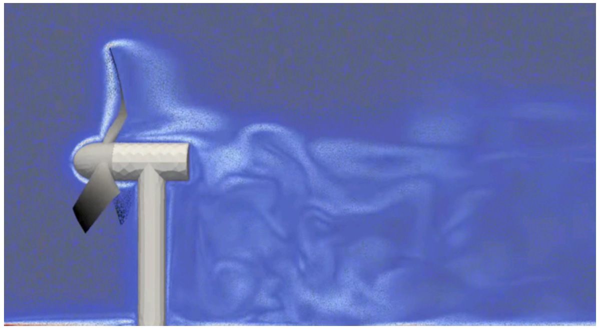
T.Apply(*Tem_avg*, *T'*)

Examples of using ArrayUDF (Continued)

Example 2: vorticity computation in fluid flow



Combustion engines



Modeling renewable energy

Pictures credit to: LANL, Frank Fritz Michael Milthaler, *etc.*

Three steps by using ArrayUDF :

Step 1: Initialize data (2D example)

Array *V_X*("data location pointer")
Array *V_y*("data location pointer")

Step 2: Define operation on Stencil

VC_X(Stencil u):
    return u(0,1)- u(0, -1)
VC_Y(Stencil v):
    return v(1,0)- u(-1, 0)

Step 3: Run & get result

V_X.Apply(VC_X, *V_X'*)
V_Y.Apply(VC_Y , *V_Y'*)
V_X'+*V_Y'* as vorticity

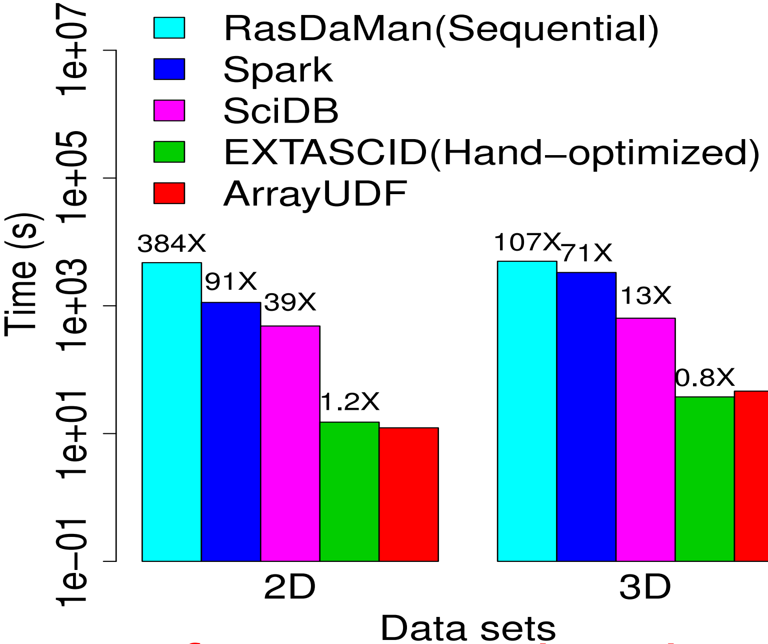
Evaluations

- Hardware:
  - Edison, a Cray XC30 supercomputer at NERSC
  - 5576 computing nodes, 24 cores/node, 64GB DDR3 Memory
- Software
  - ArrayUDF - RasDaMan 9.5 (sequential version)
  - Spark 1.5.0 - EXTASCID(hand-optimized version)
  - SciDB 16.9 - Hand-optimized C/C++ code
- Workloads
  - Two synthetic data sets (i.e., 2D and 3D) for micro benchmarks
    - Window operators, chunking strategy, trail-run, etc.
  - Four real scientific data sets (i.e., S3D, MSI , VPIC , CoRTAD)
    - Overall performance tests /w generic UDF interface

Comparison with peer systems with standard "window" operators

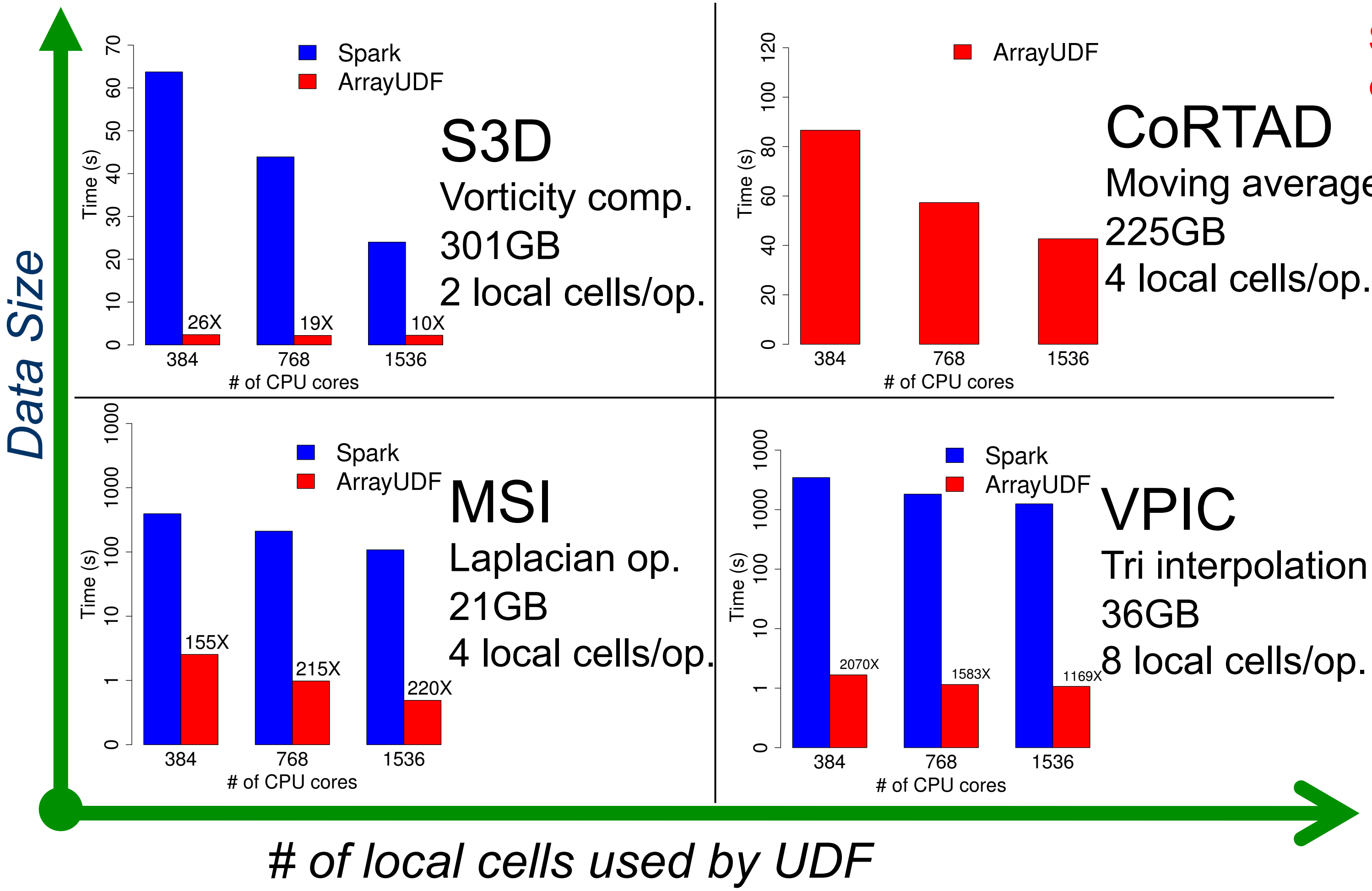
- "window" comes from SciDB and RasDaMan, where a operator is applied to all window members uniformly

Average on
- window 2x2 for 2D
- window 2x2x2 for 3D



- ArrayUDF has close performance to hand-optimized code
- ArrayUDF is as much as 384X faster than peer systems

Comparison with Spark in real scientific data analysis with generic UDF interface



Spark experiences out-of-memory:

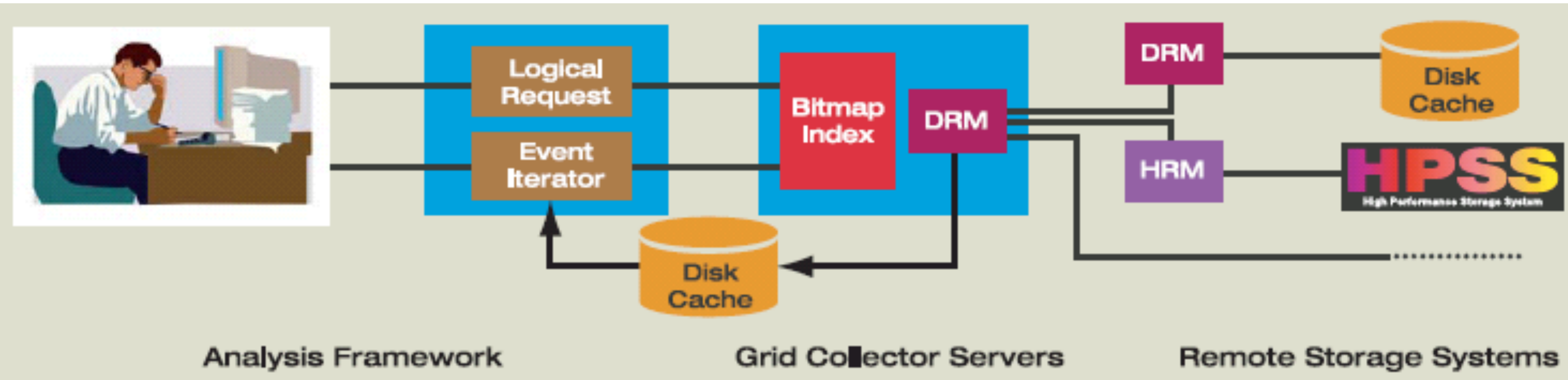
- **large data size**
- **more local cells**

We observed ArrayUDF is 2070X faster

What can scientific data analyses learn from Big Data software?

- Separate data management from data analyses
- Develop data model for science
- Support complex data access patterns
  - Accessing neighbors
  - Selective special records

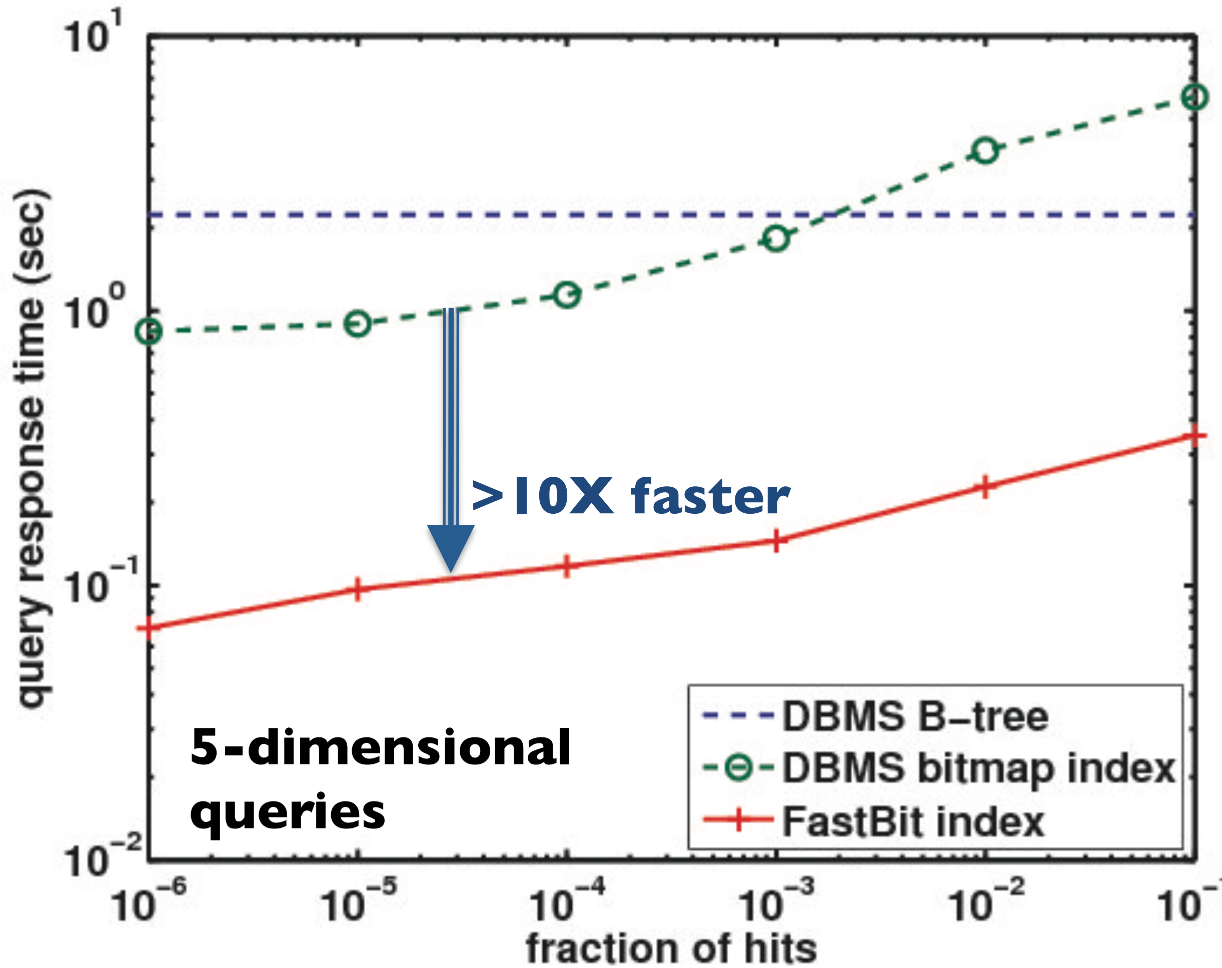
Selective Access Example: Grid Collector



- High-Energy Experiment STAR was to search for Quark Gluon Plasma (QGP)
- A small number (~hundreds) of collision events may contain the clearest evidence of QGP
- Using high-level summary data, one found 80 special events
  - Have track distributions that may indicate presence of QGP
- Further analysis needs to access more detailed data
  - Detailed data are large (terabytes) and reside on tape archive
  - May take many weeks to manually migrate to disk
- Grid Collector retrieved the 80 events in 15 minutes (2005)
- Key technology in Grid Collector: **indexing**

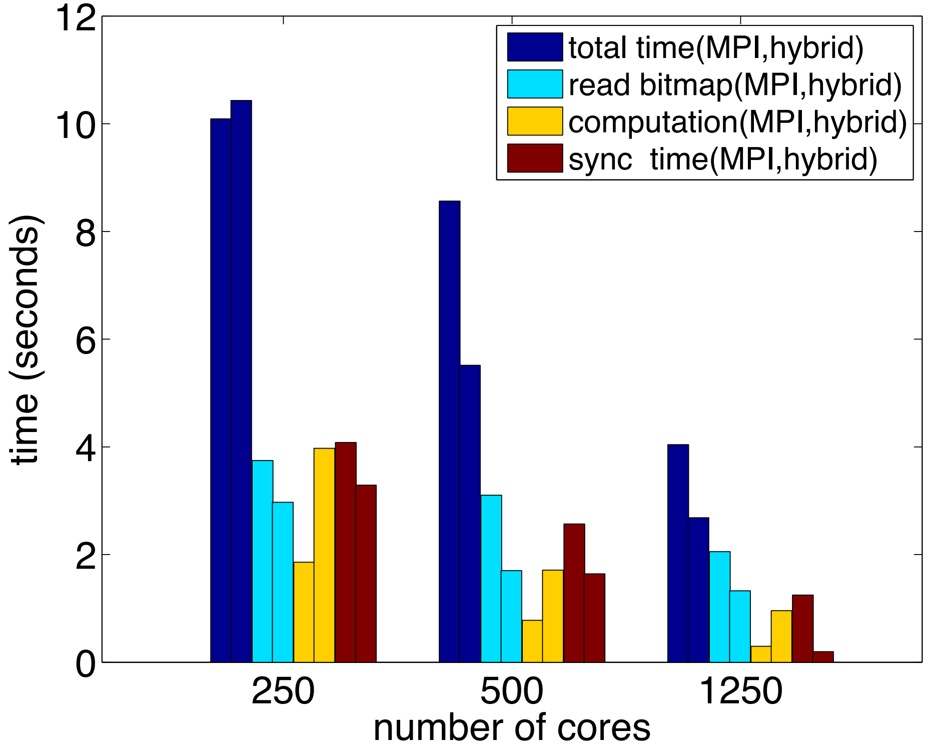
Multi-Dimensional Query with FastBit

- Queries 5 out of 12 most popular variables from STAR (2.2 million records)
- Average attribute cardinality (distinct values): 222,000
- FastBit uses WAH compression
- DBMS uses BBC compression
- FastBit >10X faster than DBMS
- FastBit indexes are 30% of raw data sizes



[Wu, Otoo and Shoshani 2002]

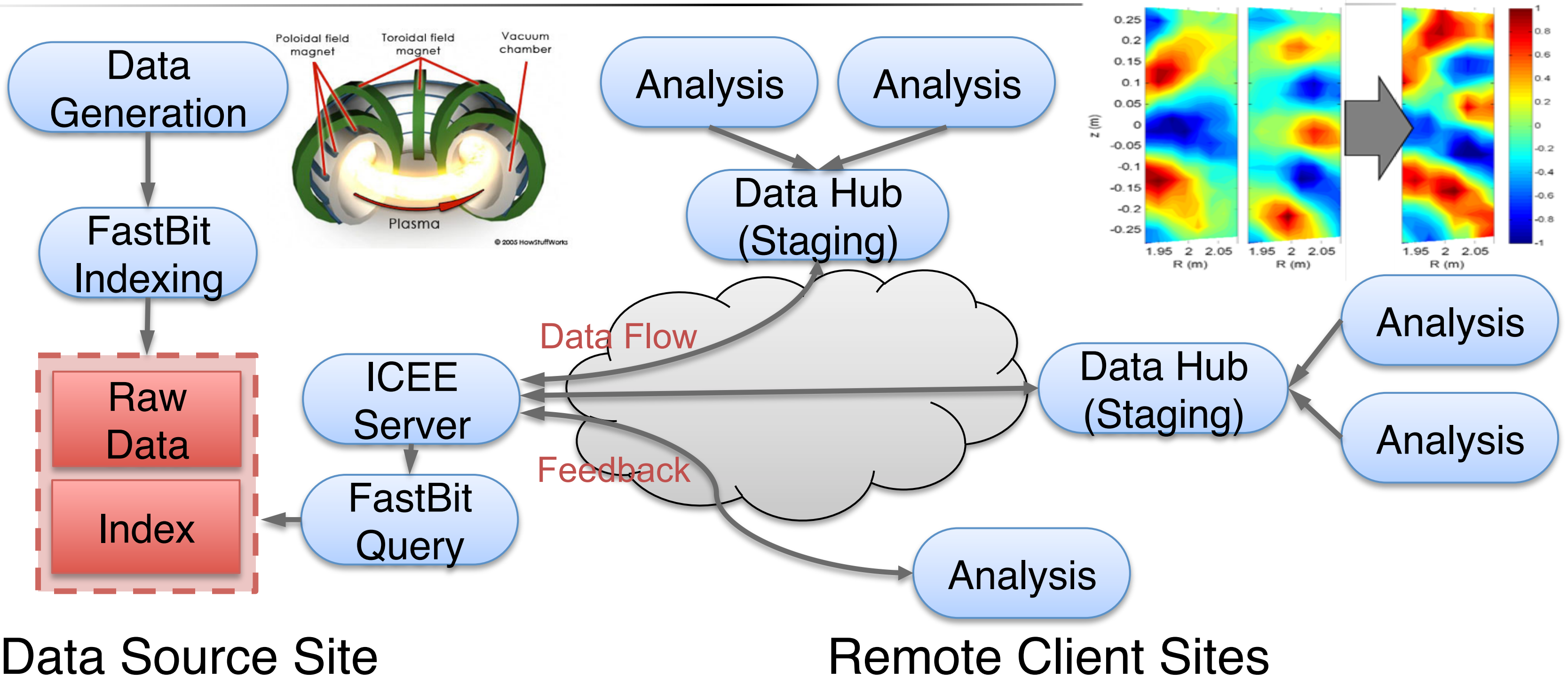
Performance of Querying with Hybrid Parallelism



#cores	scan	MPI-alone	hybrid
250	975	10.1	10.8
500	532	8.6	5.5
1250	266	4.1	2.7

- Queried for particles where ‘Energy > 1.3’ from the trillion-particle dataset
- Took **less than three seconds** to sift through 1 trillion particles
- Better than MPI-only

Indexing in a Distributed Analysis Framework



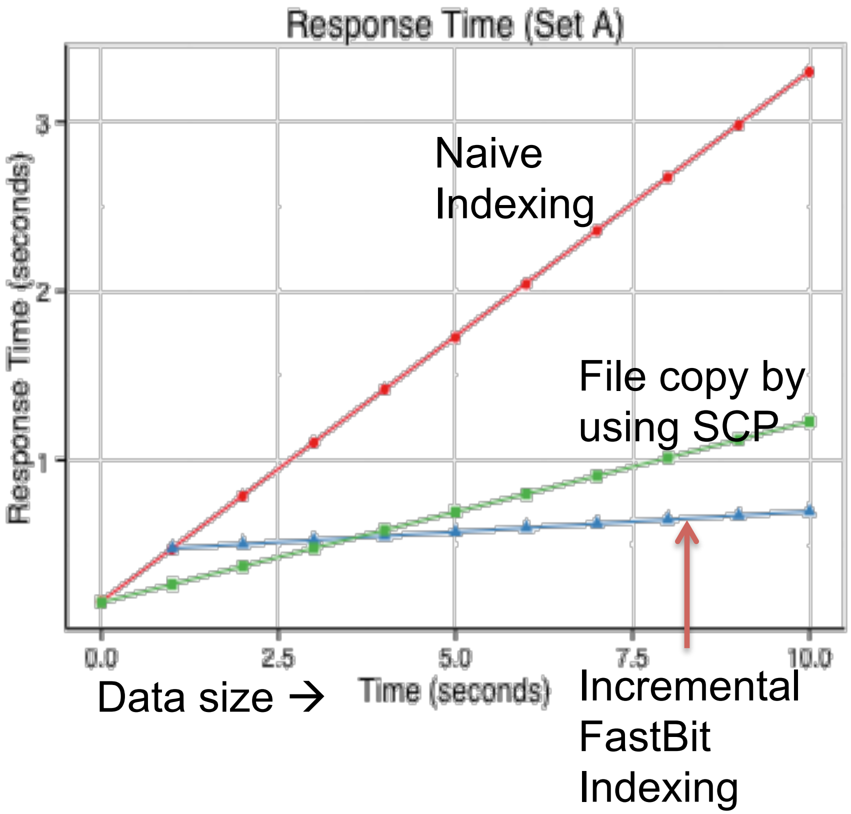
ICEE in situ analysis system

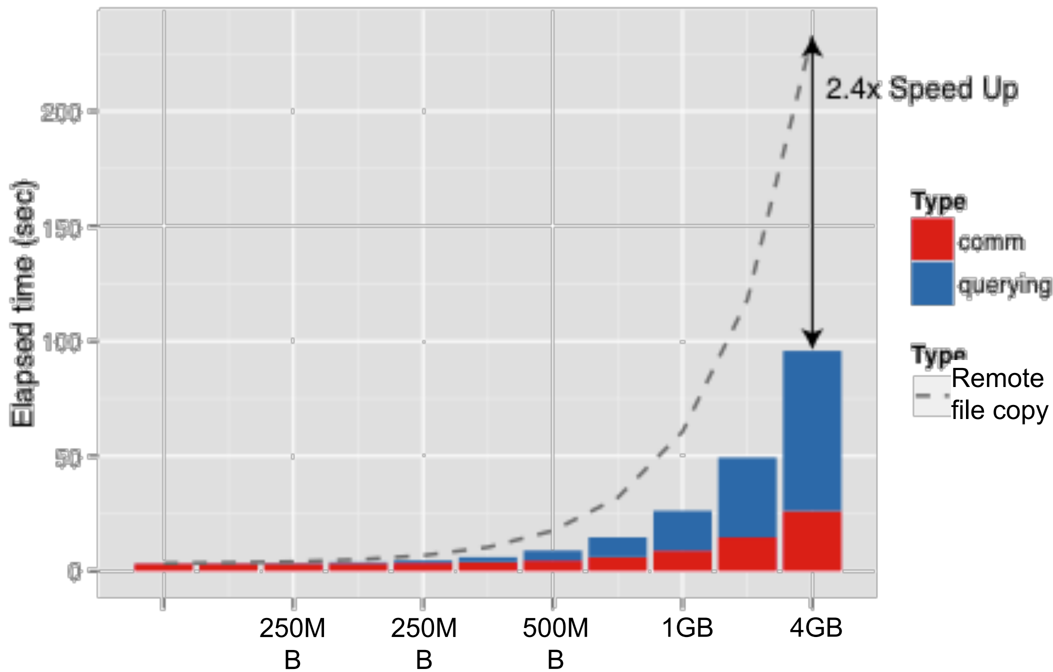
- ADIOS provides an overlay network to share data and give feedbacks
- Stream data processing – supports stream-based IO to process pulse data
- In transit processing – provides remote memory-to-memory mapping between data source (data generator) and client (data consumer)
- Indexing and querying with FastBit technology

Index-and-Query Reduces Execution Time

Remote file copy VS. index-and-query

- Measured between LBL and ORNL to simulate KSTAR-LBL-ORNL connection
- Indexed by FastBit. Observed a linear performance (i.e., indexing cost increased by data size) ➔ Expensive indexing cost
- However, once we have index built, index-and-query can be a better choice over remote file copy



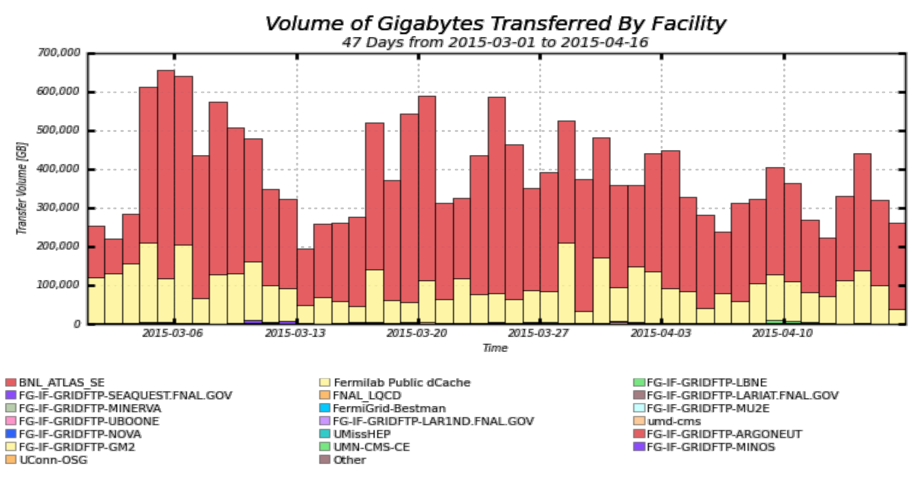


Unifying Distributed Storage Systems

BERKELEY LAB

Storage Resource Manager (SRM)

- Unify API for accessing storage systems
- Supports multiple transfer protocols and load balancing for multiple transfer servers
- Implements Storage Resource Management (SRM) interface v2.2, and compatible and interoperable with other 4 SRM implementations in WLCG



Daily data transfer volume in OSG from 3/1/2015 to 4/15/2015. BeStMan is used to transfer 100s TB/day in OSG.

Accomplishments

- Open source under BSD license, distributed with OSG software
- Scalable performance on many file systems and storages, such as Xrootd and Hadoop
- Organized an international standard through OFG - GFD.129, 2008
- Co-scheduling of network resource provisioning and host-to-host bandwidth reservation on high-performance network and storage systems

Impacts

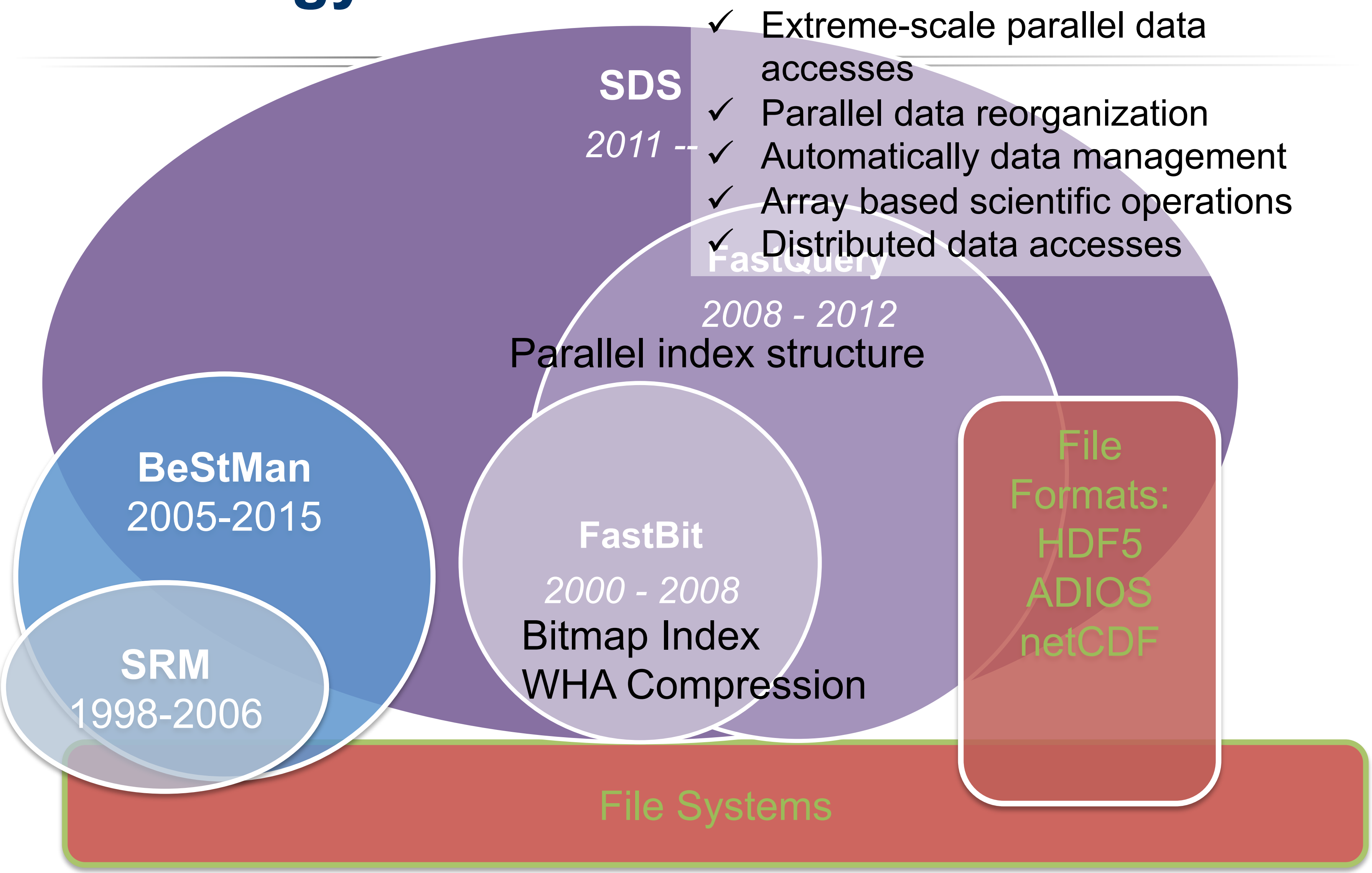
- Improve user productivity with a unified API for many storage systems
- 43 BeStMan deployments worldwide and 5 backend deployments for CERN EOS system, as of 2015
- Being used in scientific collaborations such as ESGF, OSG, and WLCG

SDS Framework Summary

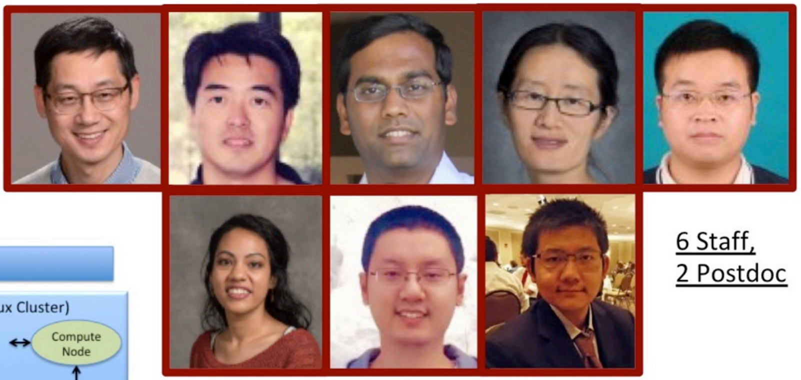
- **Extracting** information from large data sets is the key to scientific discoveries, e.g., finding supernova from astronomical observations and Higgs boson from particle collisions.
- Database systems (DBMS) support some versions of such analysis tasks, but, are often ineffective for complex scientific analyses.
- Furthermore, most scientific data sets are in **formatted data files** (HDF5, NetCDF, ADIOS BP, FITS, etc.) stored in parallel file system, not in DBMS.
- **Scientific Data Services (SDS)** framework is designed to extract information directly on formatted data files, without a DBMS.

Use Case	DBMS	SDS
Astronomy -- Palomar Transient Factory (PTF): finding supernova candidates	**~ 200s** (PostgreSQL, Hive)	**~ 5s** (new join algorithm)
Biology -- Gene Context Analysis (GCA): determine gene function from similarity of neighborhoods	**> 15 minutes** (commercial DBMS)	**< 10s** (new index)
Plasma – VPIC Data Analysis: Extracting accelerated particles near X-line of magnetic reconnection	**> 16 minutes** (SciDB)	**~ 10s** (data reorganization)

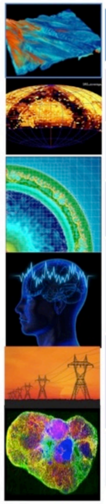
Technology Behind Scientific Data Services



Scientific Data Management

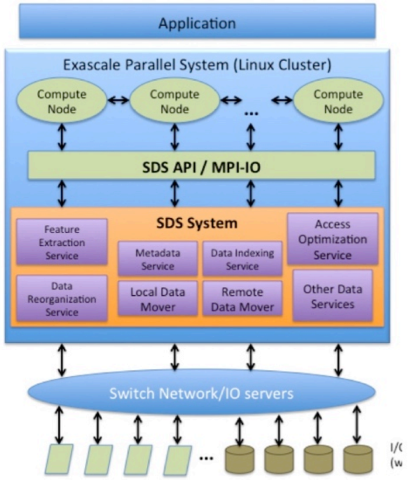


Core capability for simulation and experiments



Application	Approach	Size	Speed
VPIC Plasmas	Block Index	100x	5x
Cosmology	Block Index	100x	5x
AMR	AMR-Index		500x
Brain EEG	Statistical Similarity	106x	
Power Grid	Statistical Similarity	198x	
Mass Spec	Multilayout		90x

Parallel algorithms for data layout, access, management have huge science impact



Algorithms delivered in Scientific Data Service

6 Staff,
2 Postdoc

Collaborative and interoperable

Future: Automated algorithm selection, higher level abstraction, "elevators" for increased hierarchy

Data management is key in simulation and analysis for speed, memory and storage efficiency

- **Capability:** Premier group in algorithms and data structures, for data management, I/O and storage
- **Impact:** Improve applications performance, enable new science problems, more productive science
- **Stakeholders:** Application scientists and facilities; all SC, EERE, Health, and more

Acknowledgments

- Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, support for the SDS project and a DOE Career award under contract number DE-AC02-05CH11231

- National Energy Research Scientific Computing Center

Questions?

Contact information:

John Wu
John.Wu@nersc.gov
SDM group
http://crd.lbl.gov/sdm/