Scientific big data analytics challenges at large scale

G. Aloisio[a,b], S. Fiore[a,b], Ian Foster[c], D. Williams[d]

[a]Euro-Mediterranean Center on Climate Change, Italy

[b]University of Salento, Italy

[c]Computation Institute, University of Chicago and Argonne National Laboratory, Chicago, IL, USA

[d]Lawrence Livermore National Laboratory, Livermore, California, USA

Dr. Sandro Fiore, Prof. Giovanni Aloisio

Euro-Mediterranean Centre on Climate Change & University of Salento

enes European Network for Earth System Modelling

Università del Salento

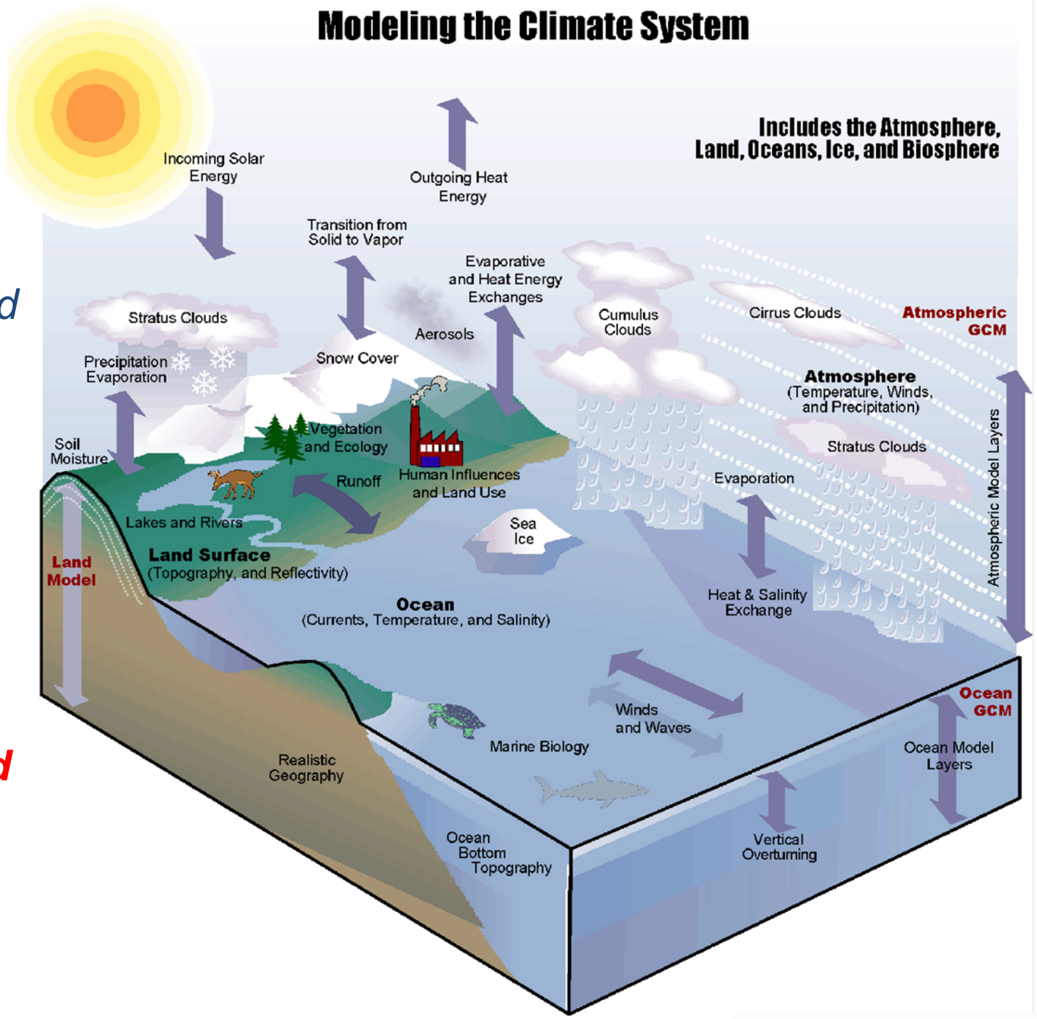
cmcc Centro Euro-Mediterraneo sui Cambiamenti Climatici

The University of Chicago

Lawrence Livermore National Laboratory

Modelling the Climate System - The big challenge

- *Several **complex processes** to be simulated*
- *Several **interacting processes***
- *Great range of **time scales** to be analyzed*
- *Great range of **spatial scales** to be considered*
- *Need **interdisciplinar sciences** (physics, chemistry, biology, geology,…)*
- *Inherently **non-linear governing equations***
- *Need **sophisticated numerics***
- ***Need huge computational resources***
- ***…and large volume of data is produced***

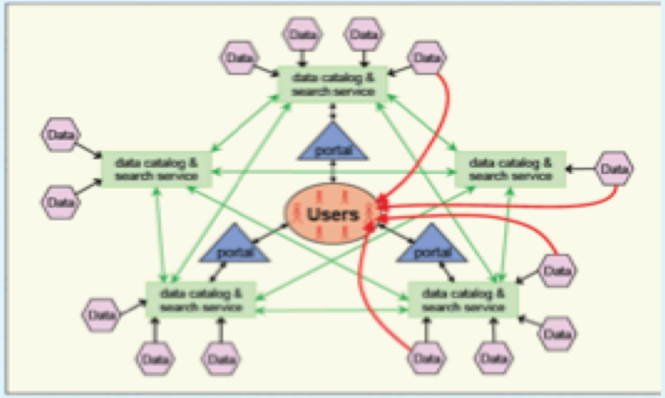


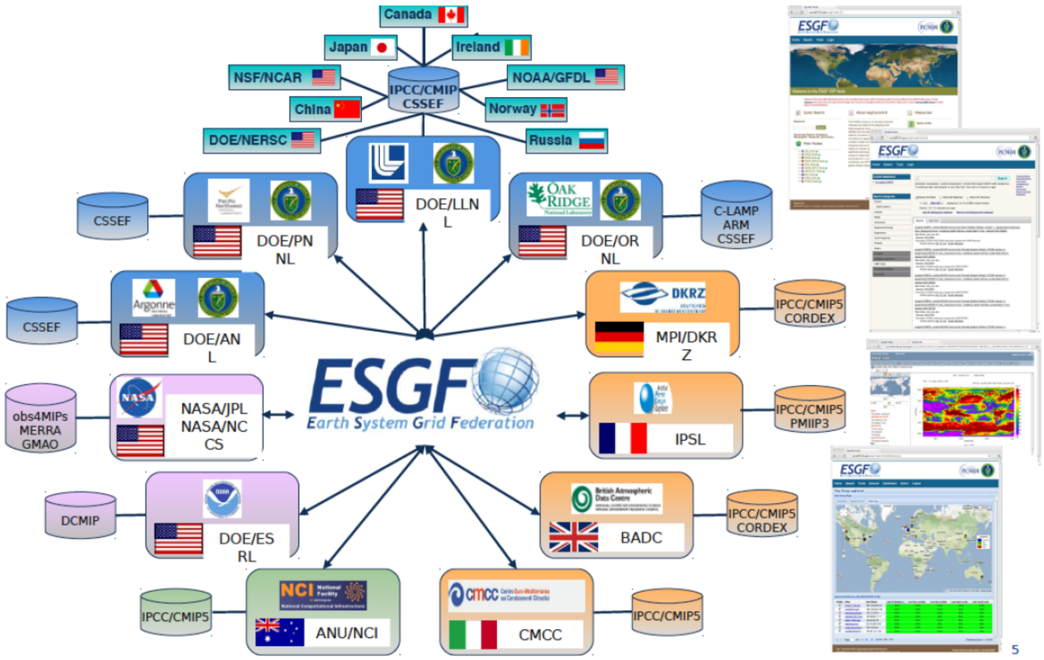
Office of Science
U.S. DEPARTMENT OF ENERGY

Climate data deluge: the CMIP5 experiment and ESGF

CMIP5 ≈ 2 to 3 PB
CMIP6 x 30 ?

Earth System Grid Federation



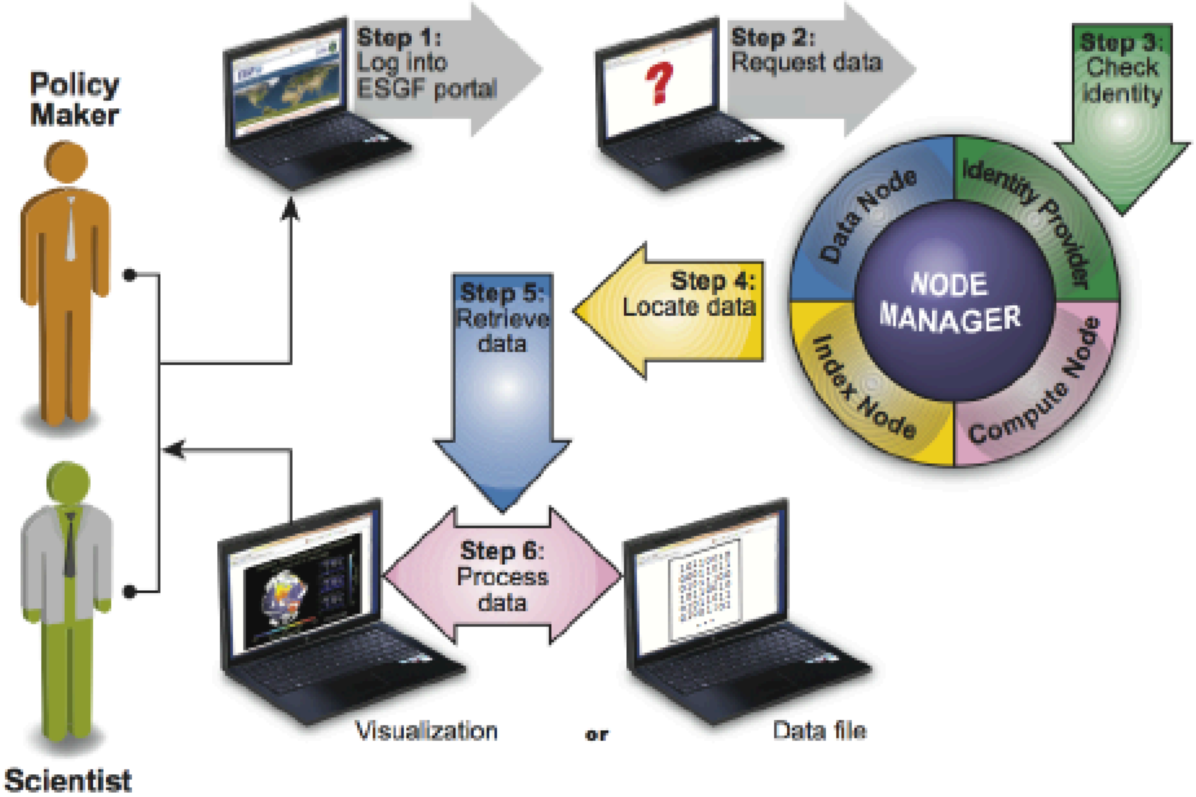


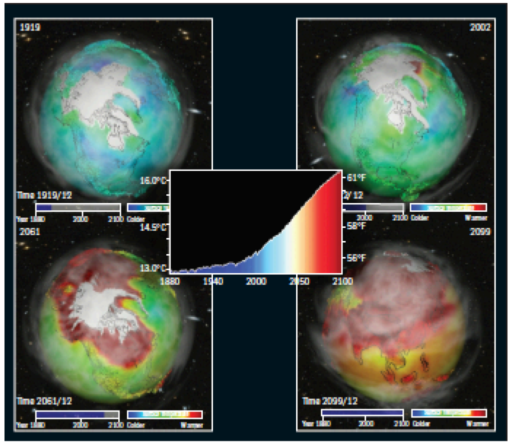
Model data expected to grow exponentially (resolution, number of simulations)

Strong demand from society : « Climate Services »

Need to have analysis and computation where data are

The current scientific workflow and the ESGF use case





Workflow: search, locate, download, analyze, display results

Earth System Grid Federation
ESGF

J. Chen, A. Choudhary, S. Feldman, B. Hendrickson, C.R. Johnson, R. Mount, V. Sarkar, V. White, D. Williams. "Synergistic Challenges in Data-Intensive Science and Exascale Computing," DOE ASCAC Data Subcommittee Report, Department of Energy Office of Science, March, 2013.

Software available, strenghts and weaknesses

Climate change **libraries** and **command line interfaces** today available:

- Climate Data Operators (**CDO**), the NetCDF Operators (**NCO**), the Grid Analysis and Display System (**GrADS**), the NCAR Command Language (**NCL**), …

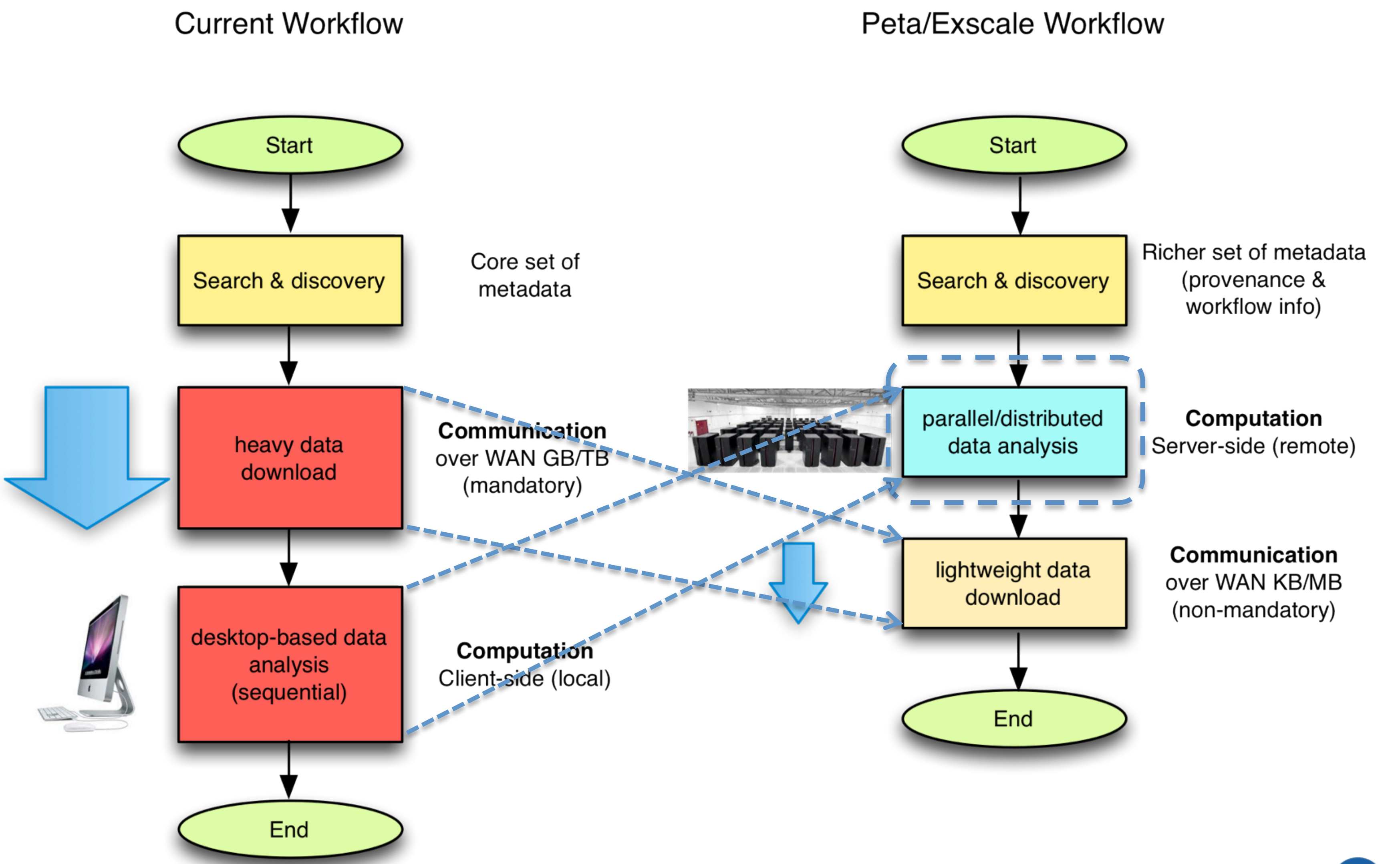
Strenghts:

- They address scientific needs and requirements coming from the climate community
- They provide complete and comprehensive set of climate data operators

Weaknesses:

- **download step** needed to get the raw data before starting locally any kind of analysis
- client-server paradigm exploiting **parallel implementations** of the needed “data primitives”.
- lack of **standardized declarative languages** to run **complex analytics tasks**

Rethinking the workflow…



Current Workflow

Start

Search & discovery

Core set of metadata

heavy data download

Communication over WAN GB/TB (mandatory)

desktop-based data analysis (sequential)

Computation Client-side (local)

End

Peta/Exscale Workflow

Start

Search & discovery

Richer set of metadata (provenance & workflow info)

parallel/distributed data analysis

Computation Server-side (remote)

lightweight data download

Communication over WAN KB/MB (non-mandatory)

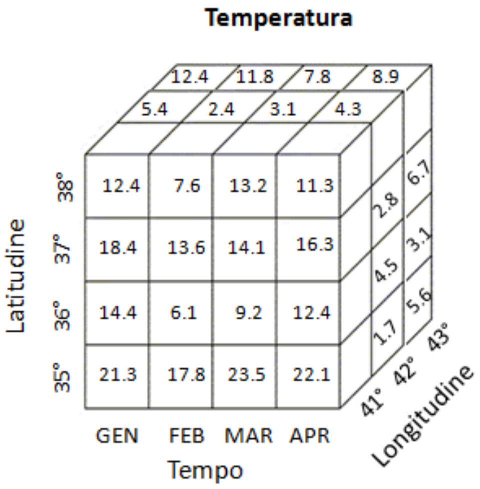
End

Multidimensional data model and the data cube abstraction

Climate data are **multidimensional** and require specific primitives for **subsetting** (slicing/ dicing), data **reduction**, **statistical** analysis, **time series analysis**, **roll-up/drill-down**.

The full data analytics stack needs:

- *New **data structures*** based on new ***storage models*** **(domain-independent and dimensional-independent)**
- ***Data partitioning, distribution*** *and* ***replication***
- ***n-dimensional array*** *primitives for scientific data management*
- ***data cube operators*** *performing analytics-based computations on "big data"(sets)*
- *new **programming models** for BDEC*

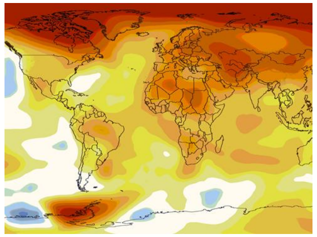


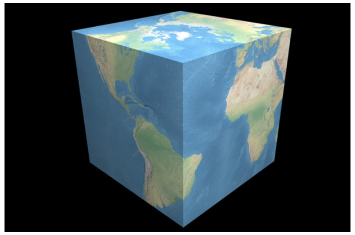
	GEN	FEB	MAR	APR
38°	12.4	7.6	13.2	11.3
37°	18.4	13.6	14.1	16.3
36°	14.4	6.1	9.2	12.4
35°	21.3	17.8	23.5	22.1

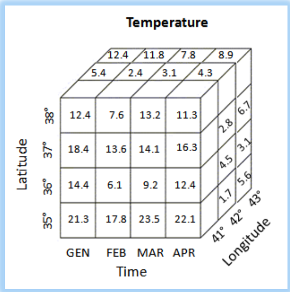
Introducing the Ophidia Project

The **Ophidia** project aims at addressing “big data” challenges, issues and requirements to support scientific data management in multiple domains.

Ophidia is an international effort among the **University of Salento**, the **Euro Mediterranean Centre on Climate Change** (CMCC), the **University of Chicago** and the **Lawrence Livermore National Laboratory** (LLNL)

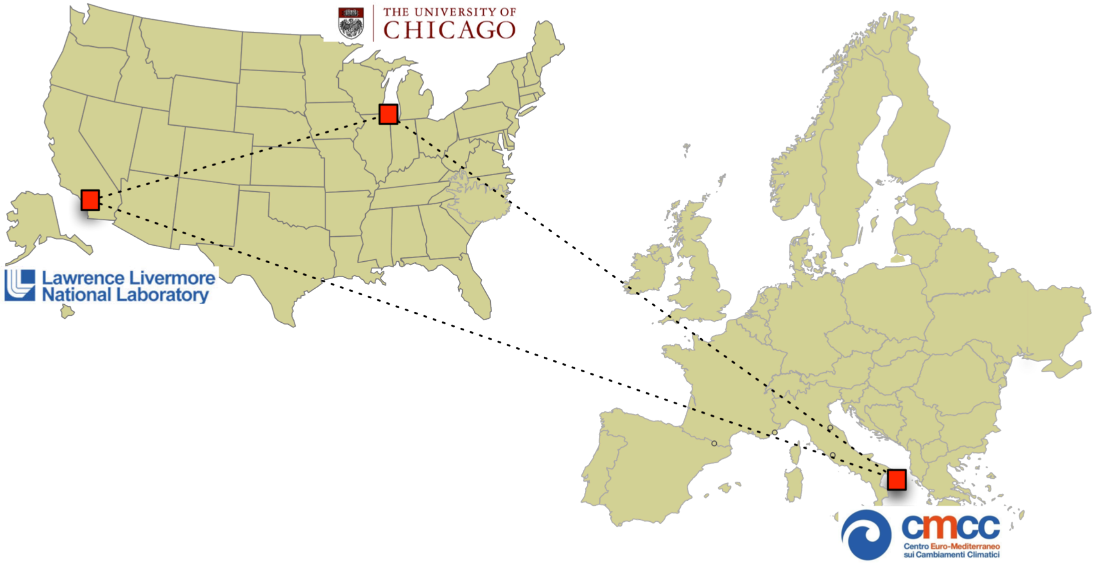






Temperature

Latitude	GEN	FEB	MAR	APR
38°	12.4	7.6	13.2	11.3
37°	18.4	13.6	14.1	16.3
36°	14.4	6.1	9.2	12.4
35°	21.3	17.8	23.5	22.1



THE UNIVERSITY OF CHICAGO
Lawrence Livermore National Laboratory
cmcc
Centro Euro-Mediterraneo sui Cambiamenti Climatici

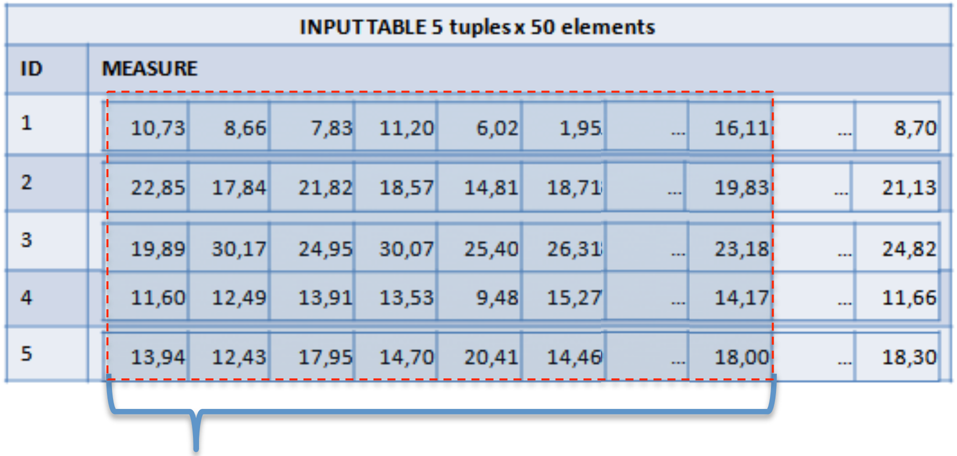
UNIVERSITÀ DEL SALENTO

[1] S. Fiore, A. D’Anca, C. Palazzo, I. Foster, D. N. Williams, G. Aloisio, “***Ophidia: toward bigdata analytics for eScience***”, ICCS2013 Conference, Procedia Elsevier, Barcelona, June 5-7, 2013.

Array based primitives: nesting feature (boxplot, su-barray, uncompress)

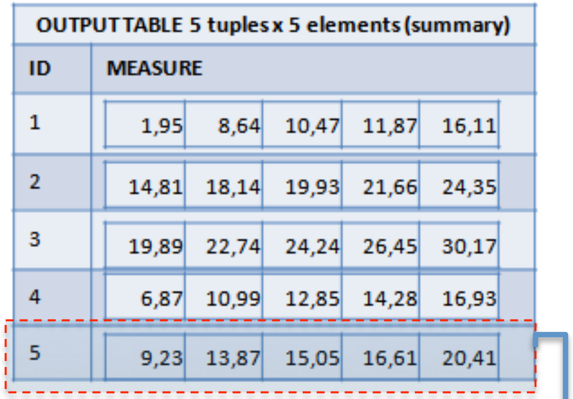
SELECT oph_boxplot(oph_subarray(oph_uncompress(measure), 1,18), "OPH_DOUBLE") AS measure FROM table;

Storage level view



INPUT TABLE 5 tuples x 50 elements

ID	MEASURE									
1	10,73	8,66	7,83	11,20	6,02	1,95	...	16,11	...	8,70
2	22,85	17,84	21,82	18,57	14,81	18,71	...	19,83	...	21,13
3	19,89	30,17	24,95	30,07	25,40	26,31	...	23,18	...	24,82
4	11,60	12,49	13,91	13,53	9,48	15,27	...	14,17	...	11,66
5	13,94	12,43	17,95	14,70	20,41	14,46	...	18,00	...	18,30

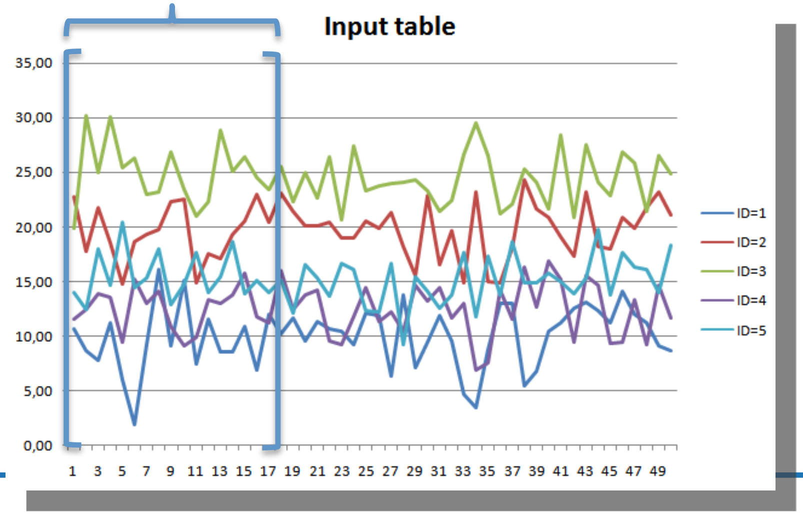


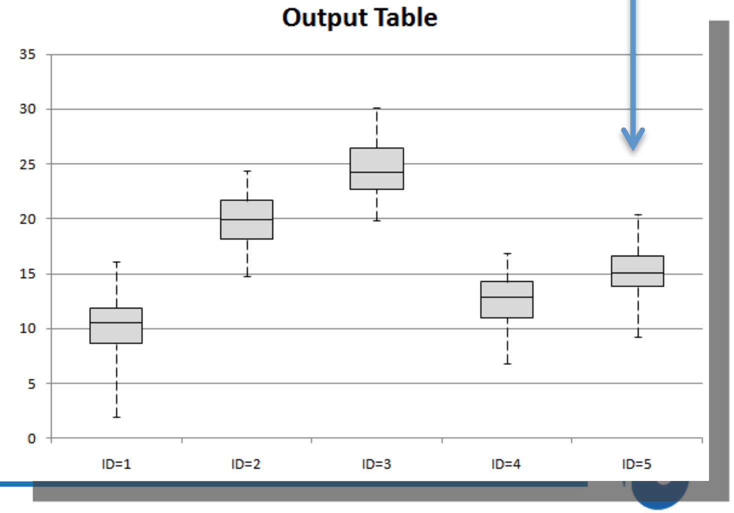
OUTPUT TABLE 5 tuples x 5 elements (summary)

ID	MEASURE				
1	1,95	8,64	10,47	11,87	16,11
2	14,81	18,14	19,93	21,66	24,35
3	19,89	22,74	24,24	26,45	30,17
4	6,87	10,99	12,85	14,28	16,93
5	9,23	13,87	15,05	16,61	20,41

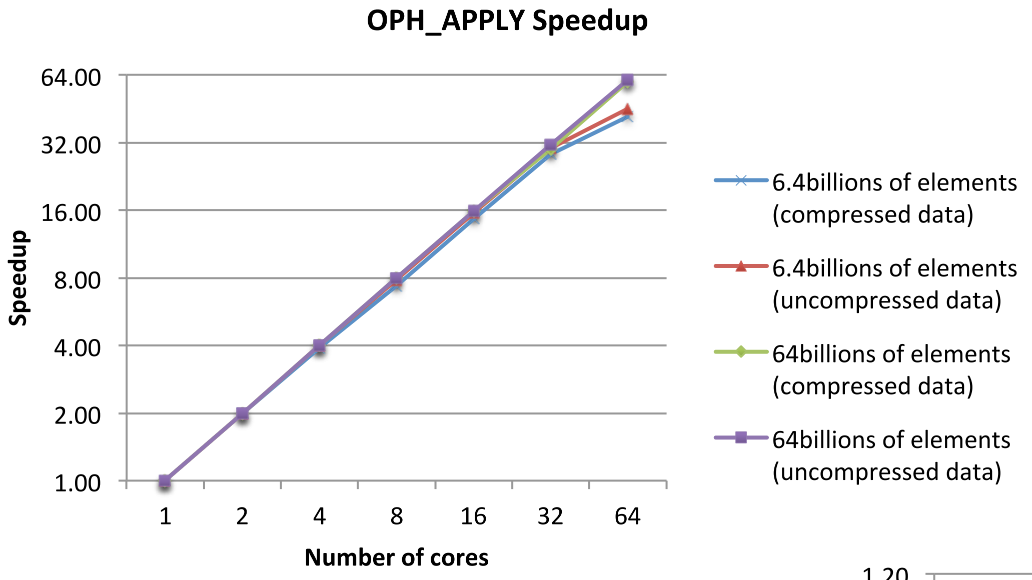
subarray(measure, 1,18)

Scientific point of view





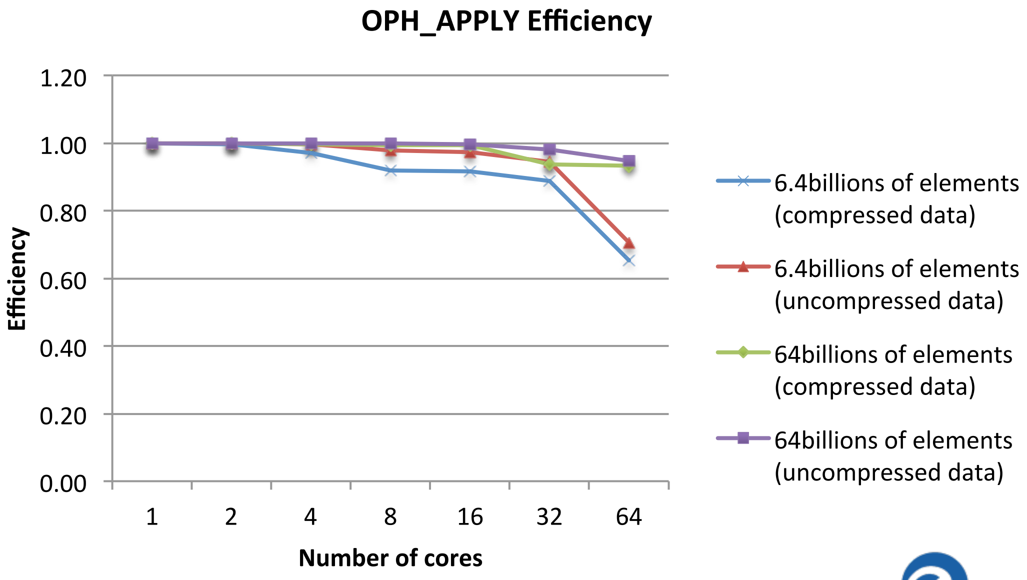
Analysis framework evaluation: OPH_APPLY benchmark



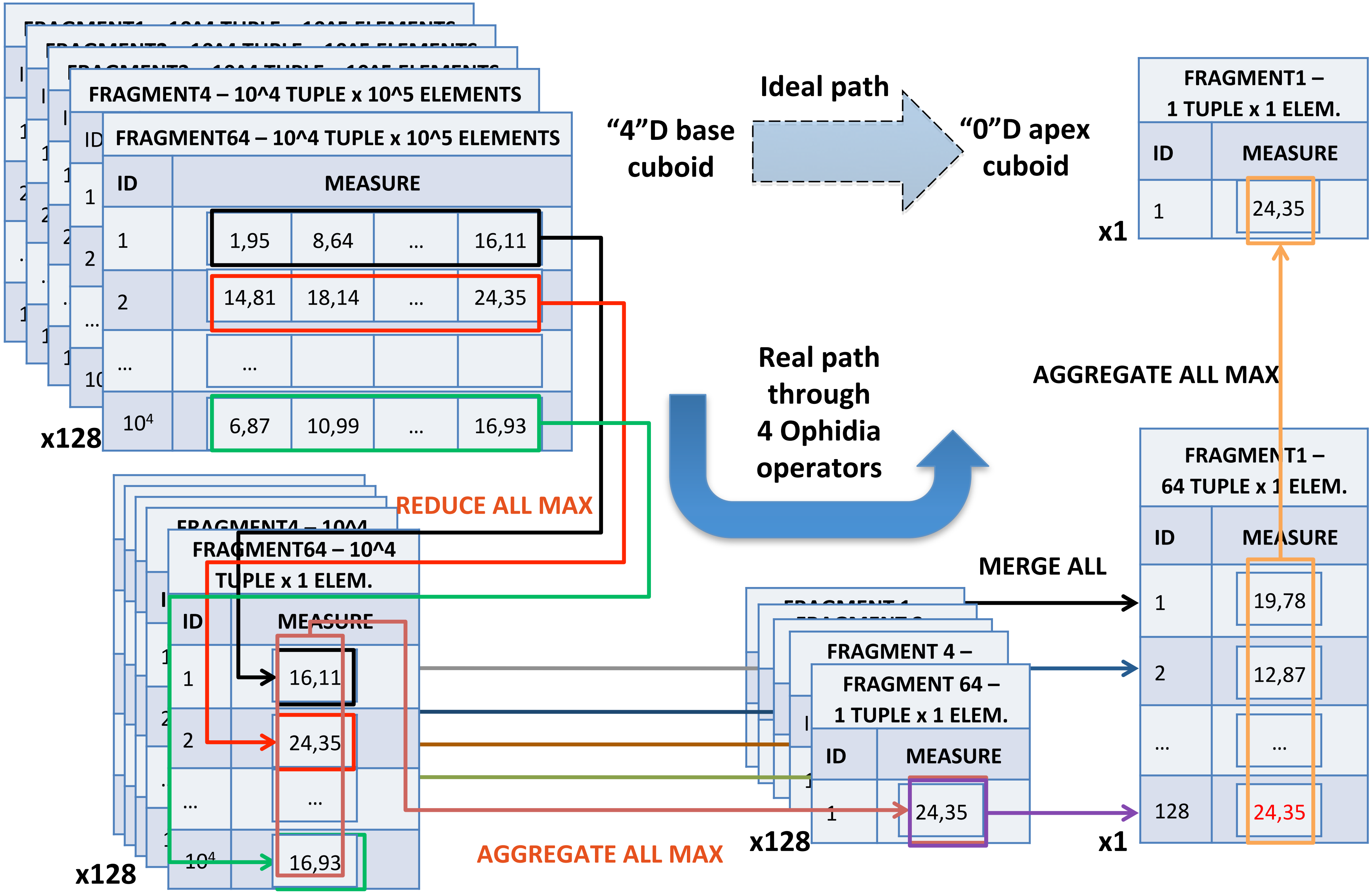
Four test cases:

- 2 different dataset sizes (6.4billions and 64billions of elements, ½ TBs)
- with/without compression

Efficiency gets up to 93%-95% with 64billions of elements on 64 parallel cores (speedup ≅60)



Running multiple operators: the apex cuboid use case



FRAGMENT64 – 10^4 TUPLE x 10^5 ELEMENTS

ID	MEASURE			
1	1,95	8,64	…	16,11
2	14,81	18,14	…	24,35
…	…			
10^4	6,87	10,99	…	16,93

FRAGMENT64 – 10^4 TUPLE x 1 ELEM.

ID	MEASURE
1	16,11
2	24,35
…	…
10^4	16,93

FRAGMENT 64 – 1 TUPLE x 1 ELEM.

ID	MEASURE
1	24,35

FRAGMENT1 – 64 TUPLE x 1 ELEM.

ID	MEASURE
1	19,78
2	12,87
…	…
128	24,35

FRAGMENT1 – 1 TUPLE x 1 ELEM.

ID	MEASURE
1	24,35

Interoperability challenges: metadata and data provenance

Metadata represents a **valuable source of information** for data discovery and data description.

In a data intensive context it will be important to:

- provide server-side **metadata management** capabilities,
- describe a dataset with **provenance metadata** information in terms of applied data analytics primitives,
- enrich this information with **descriptive metadata** and links to cross-related digital objects, that could be indexed as well, to improve the data search and discovery process,
- build **new community-oriented tools** to enrich metadata and provide, at the same time, a way to move this process towards much more **open, multi-level and collaborative forms**.

Provenance will allow a better understanding of past experiments. It will both

- avoid **re-running analysis**, and also…
- allow **reproducibility** of analysis and products.

Thanks

Life cycle management of big data for extreme-scale simulation

Kenji Ono

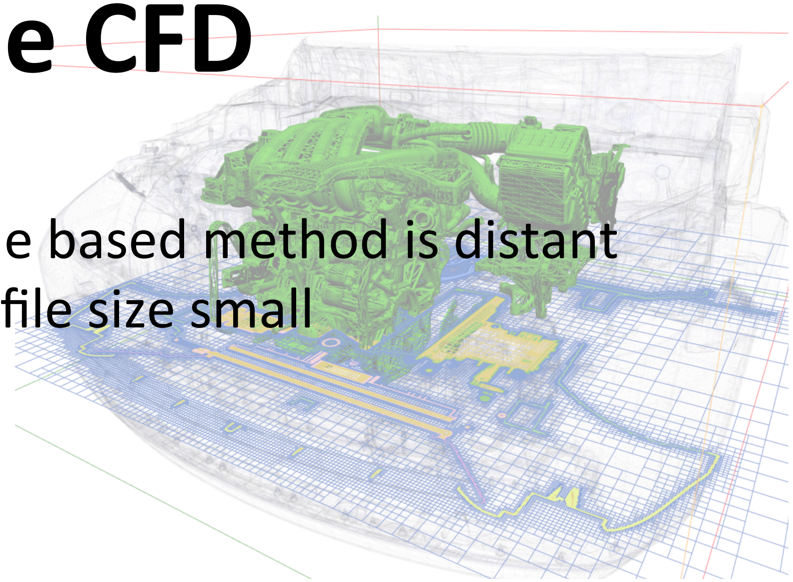
Advanced Institute for Computational Science, RIKEN

BDEC workshop, April 2013

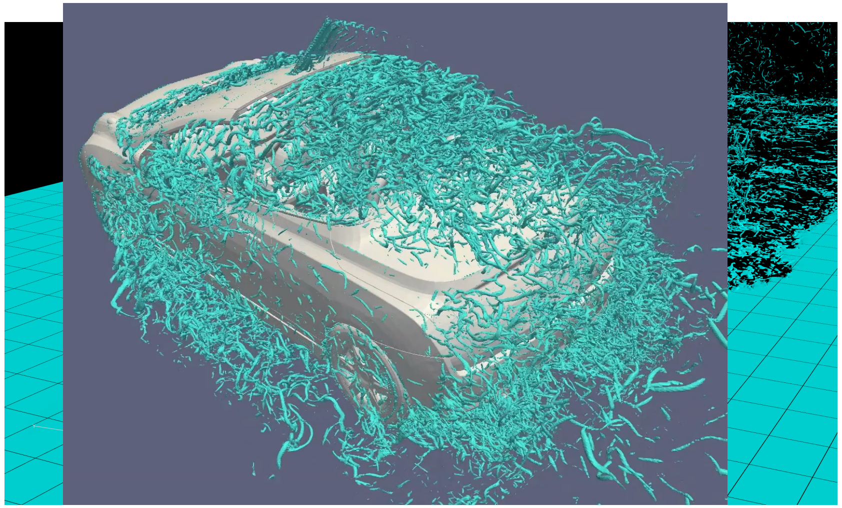
Impact of Extreme-Computing for Product Design

- **HPC will change a style of product design**
  - **Reduce time cost**
    - A solution in a short period of time
    - Many trials in shot turnaround time
      - Parametric study with details becomes feasible > MOO
  - **Increase reliability**
    - Reliability of the results becomes higher as the resolution increases with adequate solution method, e.g., LES.
  - **Tackle complicated phenomena**
    - More physics

Issues to be Addressed for Large-Scale CFD



- **Analysis model**
  - Grid generation 10G-100G range, file based method is distant
  - Compression/Decompression, keep file size small
- **Parallel computation**
  - Performance, load balancing
- **Post-processing**
  - Parallel visualization and data exploration for large-scale dataset
  - Data re-use
- **File handling**
  - Many files but a single file image
  - File I/O performance

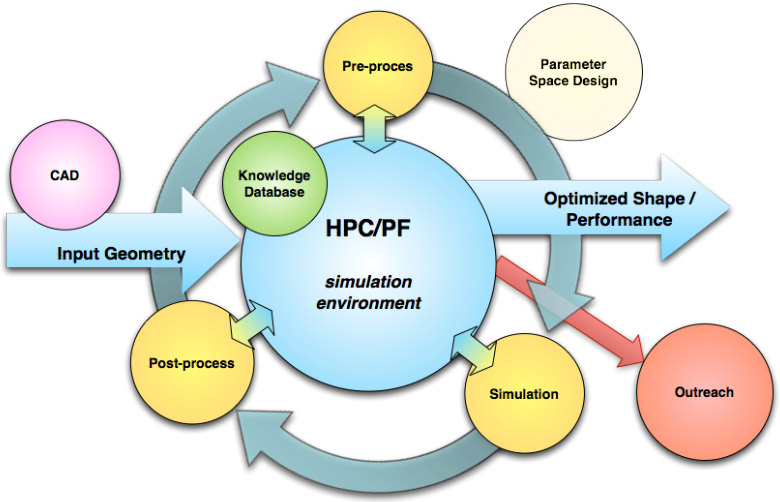


Vortex Structure on 30Billion Grids
Onishi(2012)

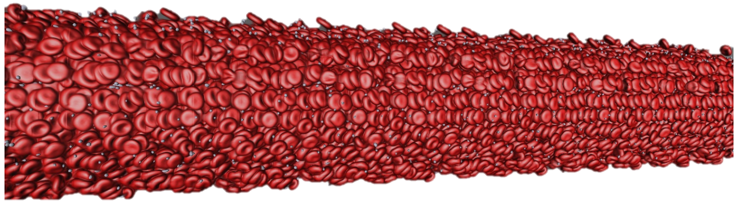
Research Topics

- **Large-scale CFD simulation** for industrial applications
  - Management of distributed files in application

- Developing an **execution environment** to support a design process of a product
  - Project management
  - Workflow
  - Simulators
  - Pre/Post processing
  - Database



- Development of a **visualization system on K computer** for large-scale datasets



Sugiyama@UT
1.4Billion cells, 45GB x 700 time slices

TOC

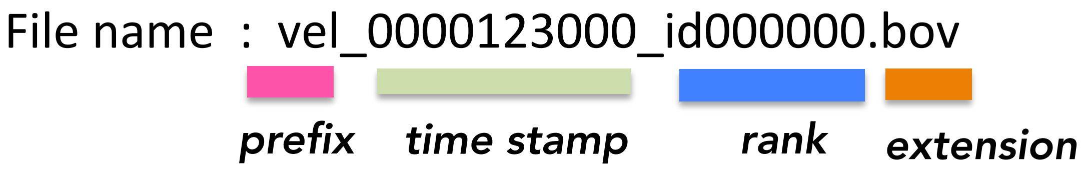
- **Application data management**
- **Project data management**
- **In-situ issue**
- **Database**

Application Data Management

- It is important **to design a way of management for domain specific applications**
  - Data structure
  - Use-case scenarios

- **Distributed file management for domain decomposition based simulation on Cartesian data structure**
  - Directory management
  - Restart
  - Mutual exploitation of file I/O between a simulator and a post processing

File Output Pattern



File name : vel_0000123000_id000000.bov

All together

```
~/hoge/vel_*_id*.bov
      /prs_*_id*.bov
```

Time slice directory

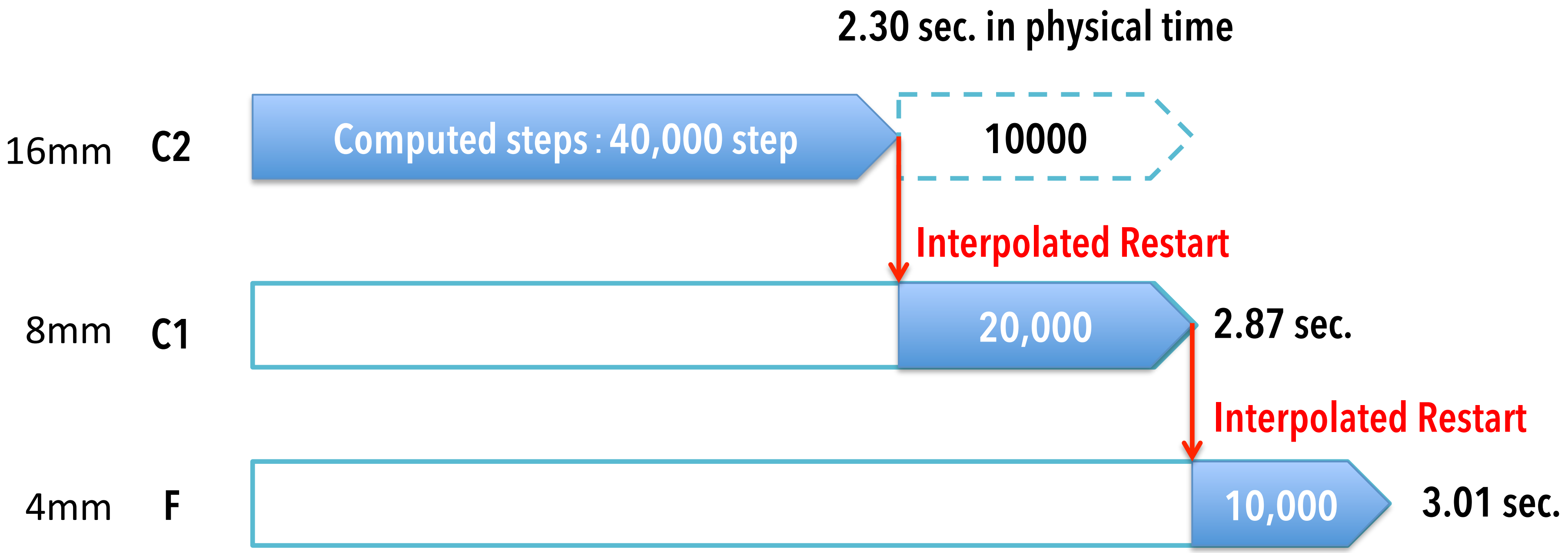
```
~/hoge/100/vel_0000000100_id*.bov
          /prs_0000000100_id*.bov

      /200/vel_0000000200_id*.bov
          /prs_0000000200_id*.bov
```

Collected file

```
~/hoge/vel_*.bov
      /prs_*.bov
```

To get solution in a short period time

Width (mm)	Model (# Grids)	Nodes (# Process)	Steps	Computed Time (H)	Physical Time (sec.)	Start
16	C2 (0.45G)	9,216	50,000	1.0	2.87	Initial
8	C1 (3.6G)	9,216	20,000	1.0	0.57	Interpolated
4	F (29G)	9,216	10,000	27.4	0.14	Interpolated



2.30 sec. in physical time

16mm C2

Computed steps : 40,000 step

10000

Interpolated Restart

8mm C1

20,000

2.87 sec.

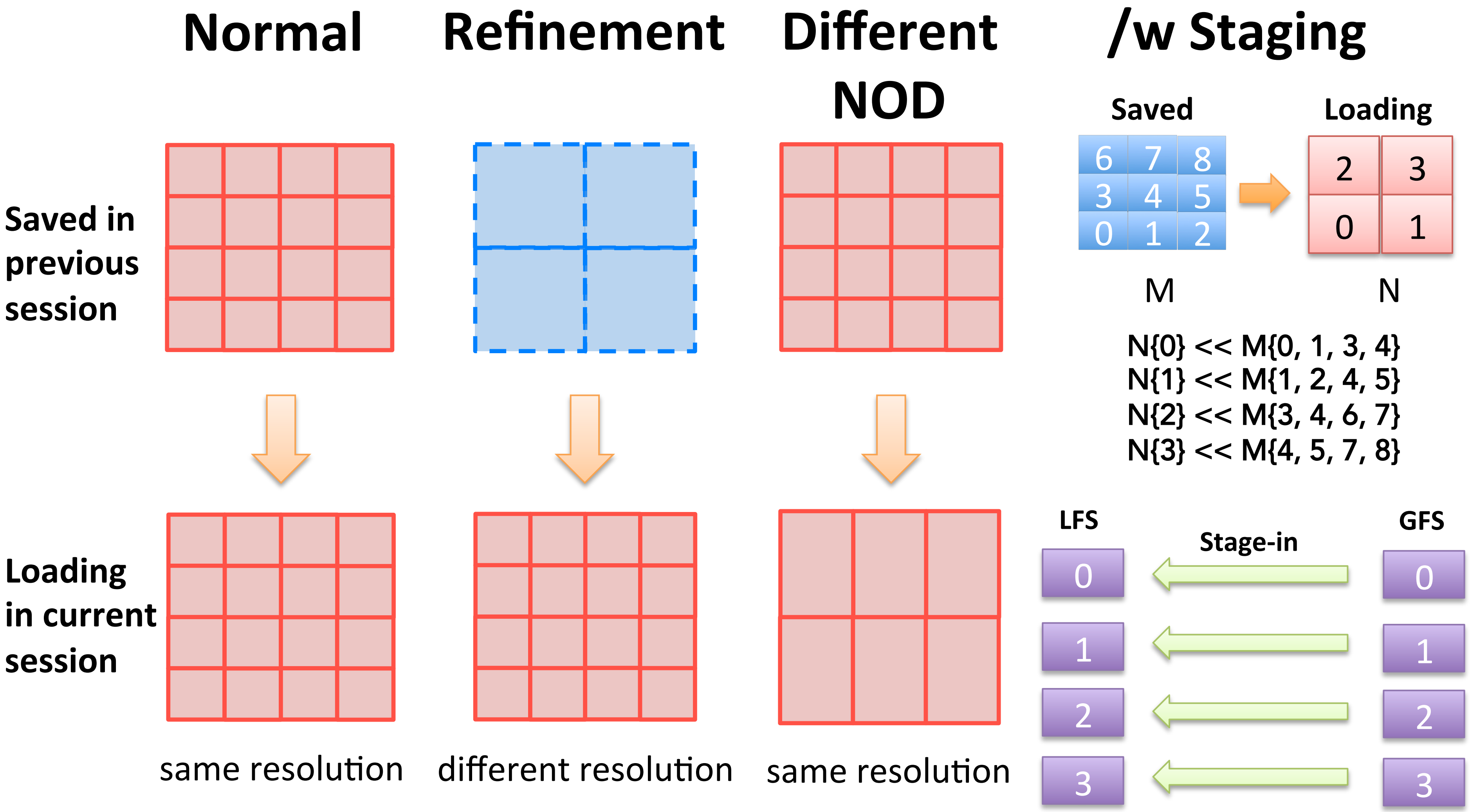
Interpolated Restart

4mm F

10,000

3.01 sec.

Restart Pattern



Normal

Refinement

Different NOD

/w Staging

Saved in previous session

Loading in current session

same resolution

different resolution

same resolution

Saved

Loading

M

N

N{0} << M{0, 1, 3, 4}
N{1} << M{1, 2, 4, 5}
N{2} << M{3, 4, 6, 7}
N{3} << M{4, 5, 7, 8}

LFS

Stage-in

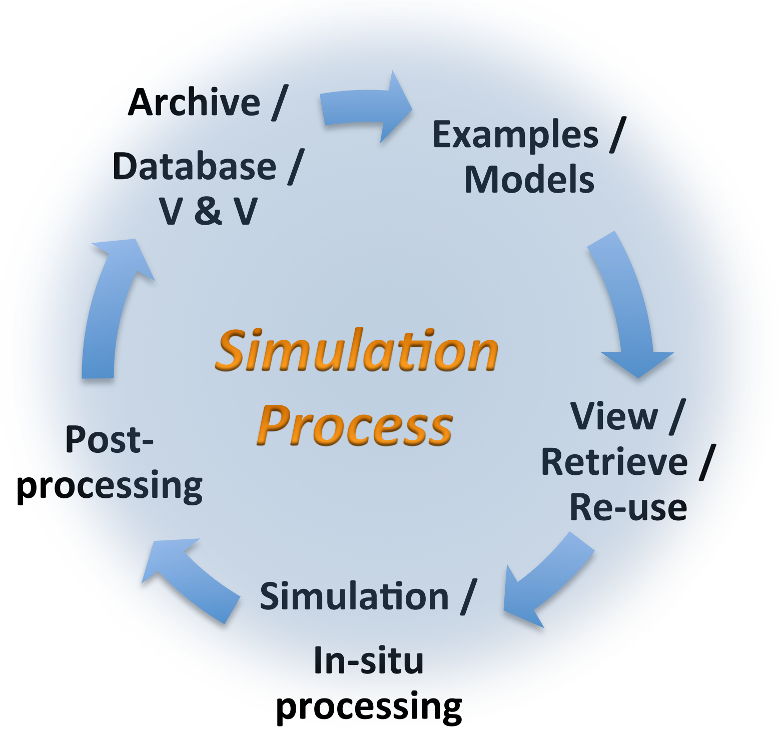
GFS

Does ADIOS already include these feature?

Project Data Management

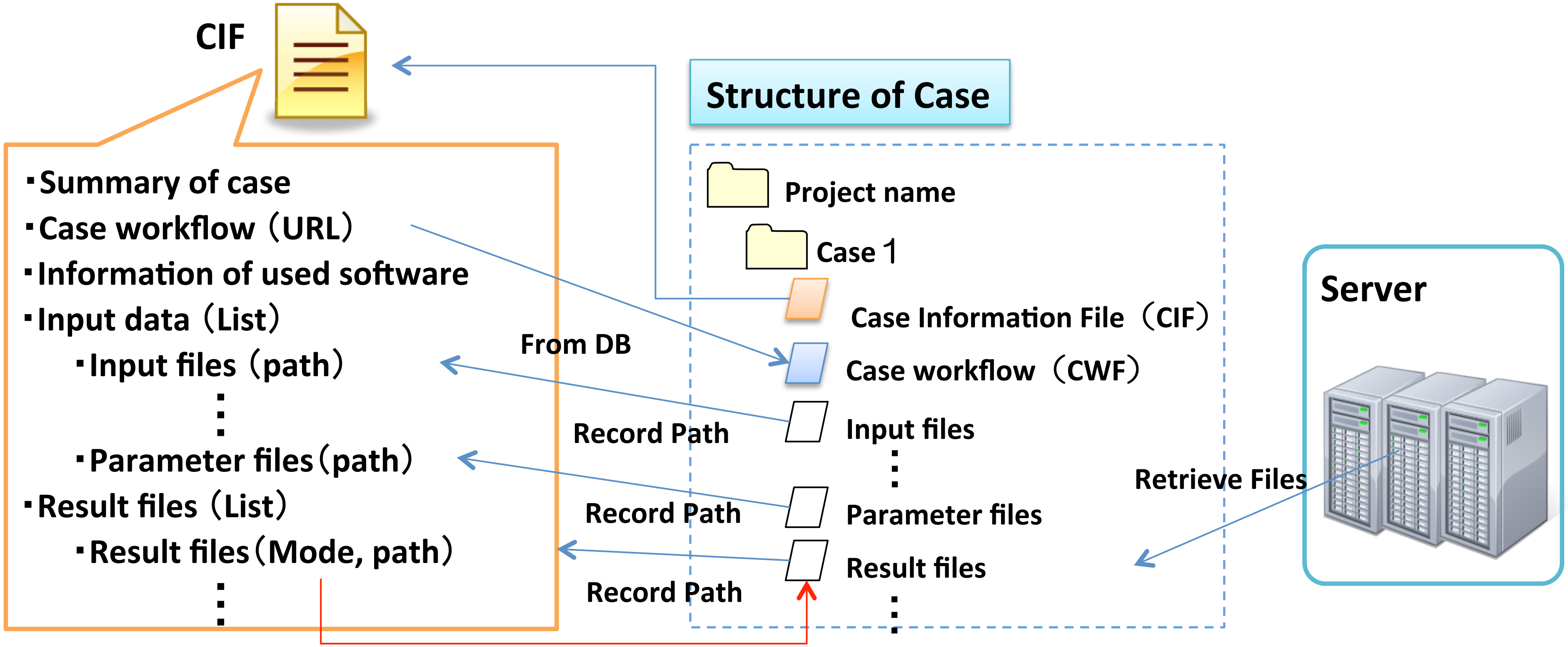
- **Resource management of a project**
  - all information; HW info., input files, calculated result files, and derived files
  - **Case**
    - a unit of execution of a simulation
  - **Project**
    - a set of cases

- **Data management enables us to**
  - automatic processing
  - collaboration with database
  - grid search
  - provenance tracking



Case Information File

- **Case**
  - a unit of execution of a simulation
  - Case Information File (CIF) describes contents



CIF

・Summary of case
・Case workflow（URL）
・Information of used software
・Input data（List）
　・Input files（path）
　⋮
　・Parameter files（path）
・Result files（List）
　・Result files（Mode, path）
⋮

From DB

Record Path

Record Path

Record Path

Structure of Case

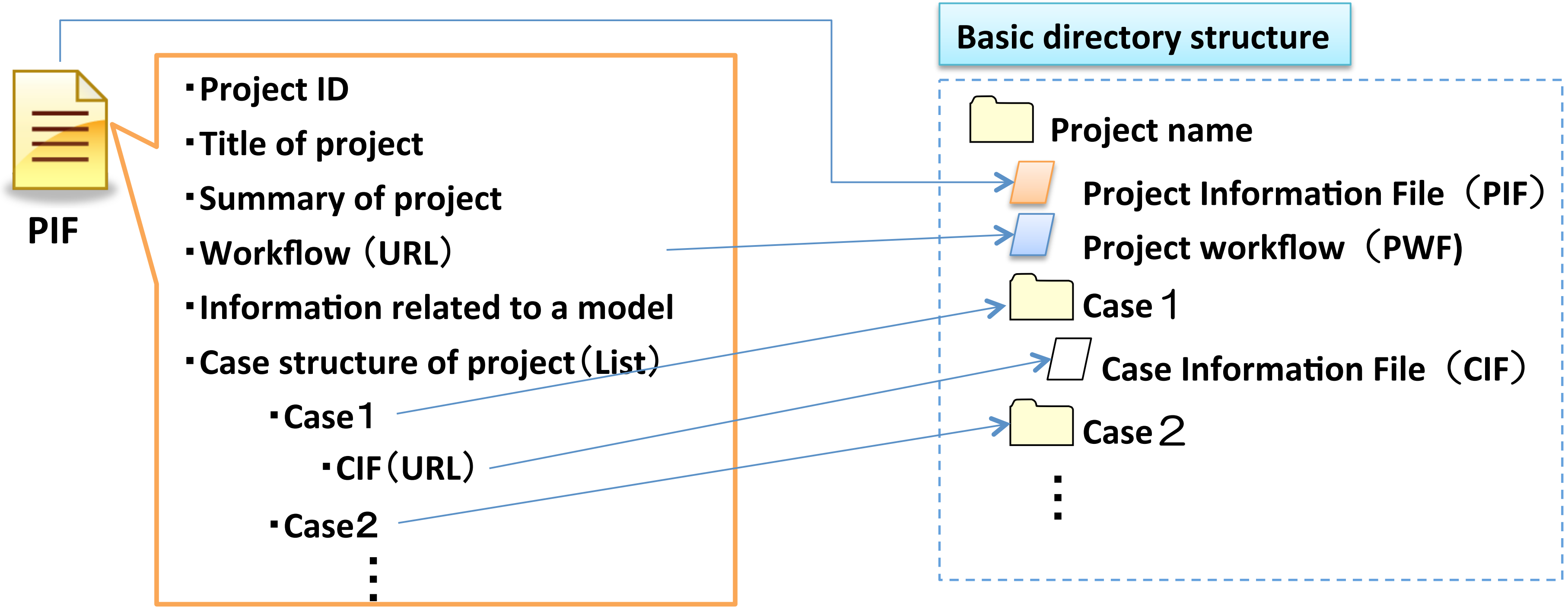
Project name
Case 1
Case Information File（CIF）
Case workflow（CWF）
Input files
⋮
Parameter files
Result files
⋮

Retrieve Files

Server

Project Information File

- **Project**
  - a set of cases
  - Project Information File (PIF) describes contents


PIF
・Project ID
・Title of project
・Summary of project
・Workflow（URL）
・Information related to a model
・Case structure of project（List）
・Case1
・CIF（URL）
・Case2
⋮
Basic directory structure
Project name
Project Information File（PIF）
Project workflow（PWF）
Case１
Case Information File（CIF）
Case２
⋮

Workflow

- Workflow is described by **basic and commonly used technology**
  - Shell and Perl

- Introduction of **Xcrypt**
  - Xcrypt allows us to control batch job submission and retrieve results from server.
  - http://super.para.media.kyoto-u.ac.jp/xcrypt/index.html

Workflow

- **Choose basic languages to describe**
  - To take into account interoperability, technology dead is good choice because a new machine environment may not have high-level language set
  - Combine several scripts
    - For instance, Shell + Perl

Post processing

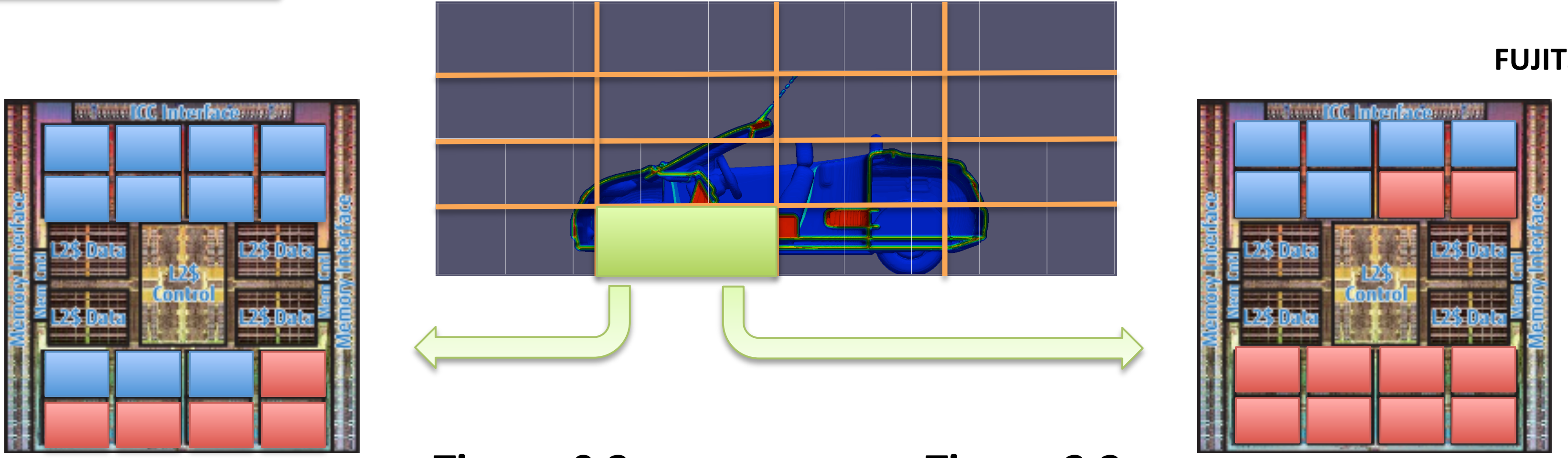
- **In-situ processing**
  - Resource assignment between simulator and data processing
- **Rendering on supercomputer**
  - Rendering API, image compositing, large image
- **Multi-modal data processing**
  - Agent-based approach

Dynamic Resource Assignment for In-Situ case

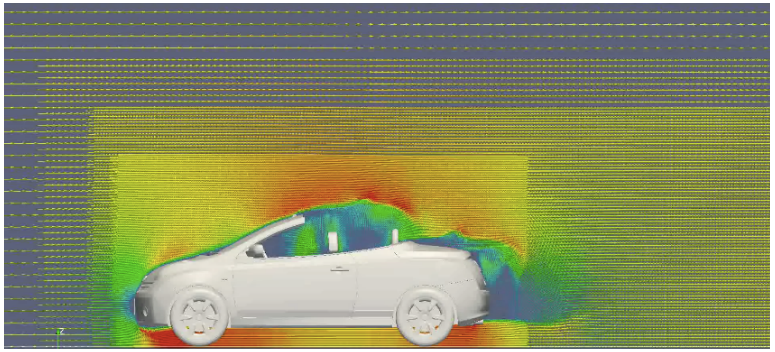
MPI or OpenMP?

Domain Decomposition

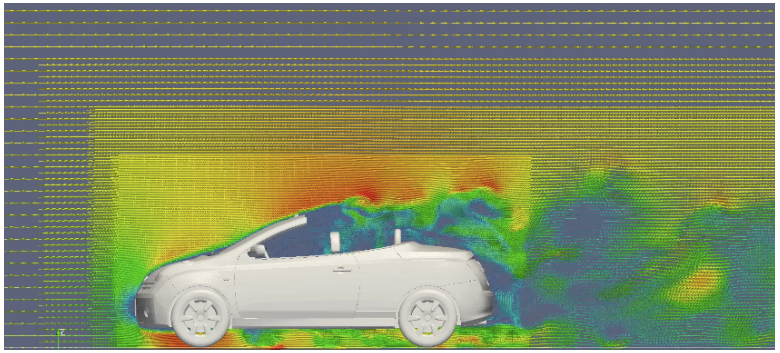
FUJITSU FX10

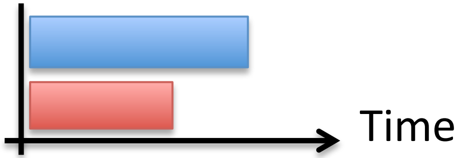


Time = 0.3



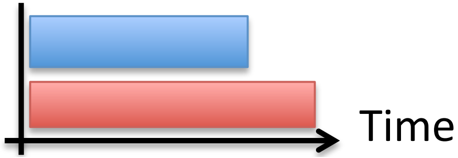
Time = 2.2





🟦 Flow Simulation

🟥 Data Processing



Database Collaboration

- **Repository**
  - All meta data derived from raw data are stored
  - Linking between meta data and raw data
  - Automatic registration by workflow
- **Scenario**
  - Simulation examples, V&V
  - Experience of archived contents
  - Zero design cycle time
- **Curation service**
  - Content curation by Baysian filter, SOM,...

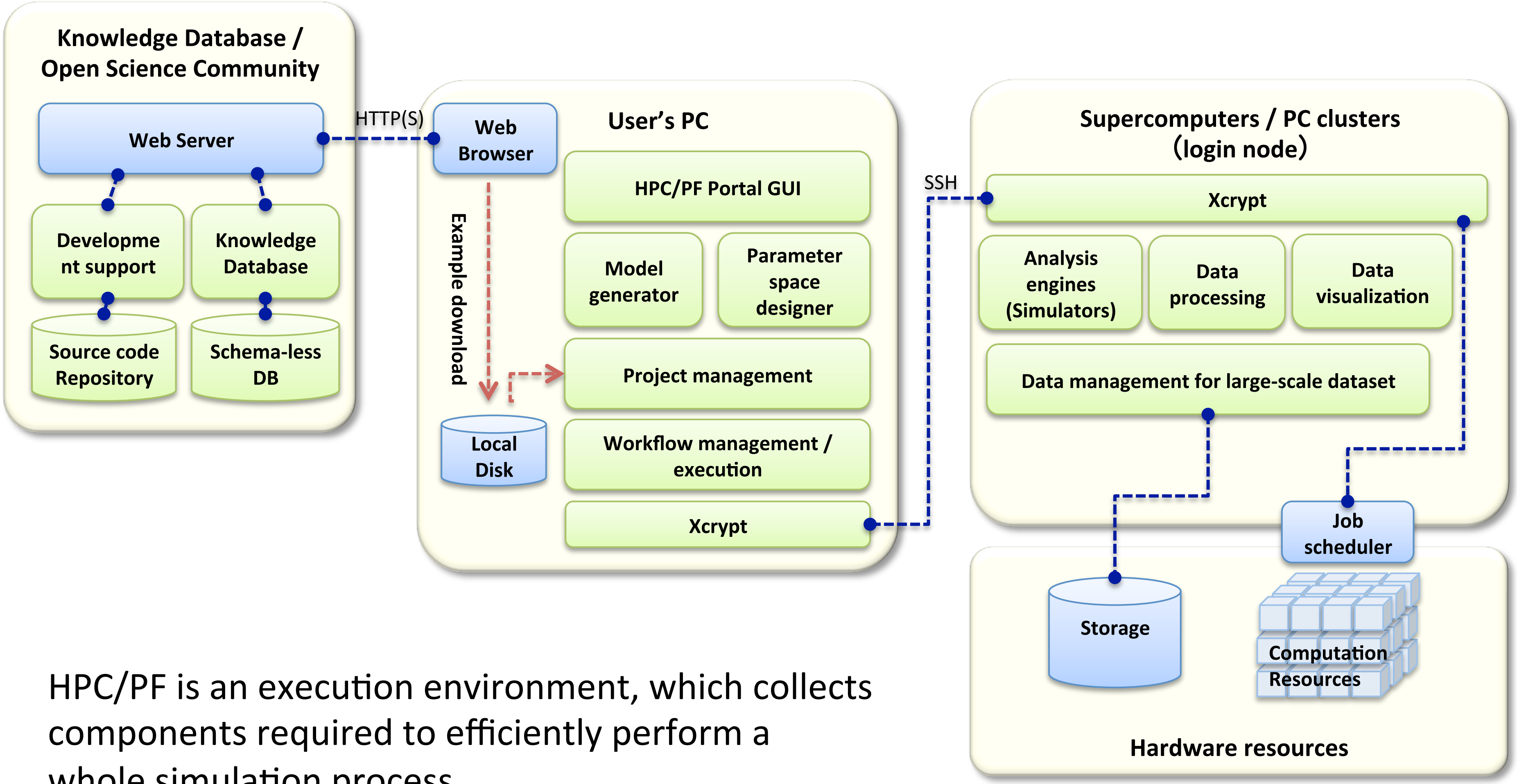
Zero Design Cycle Time

Pratt & Whitney

- **Compress leading time of design**
  - Compute all cases in parameter space
  - Register results of all cases in DB
  - Then, DB can provide data that is required to design in real-time

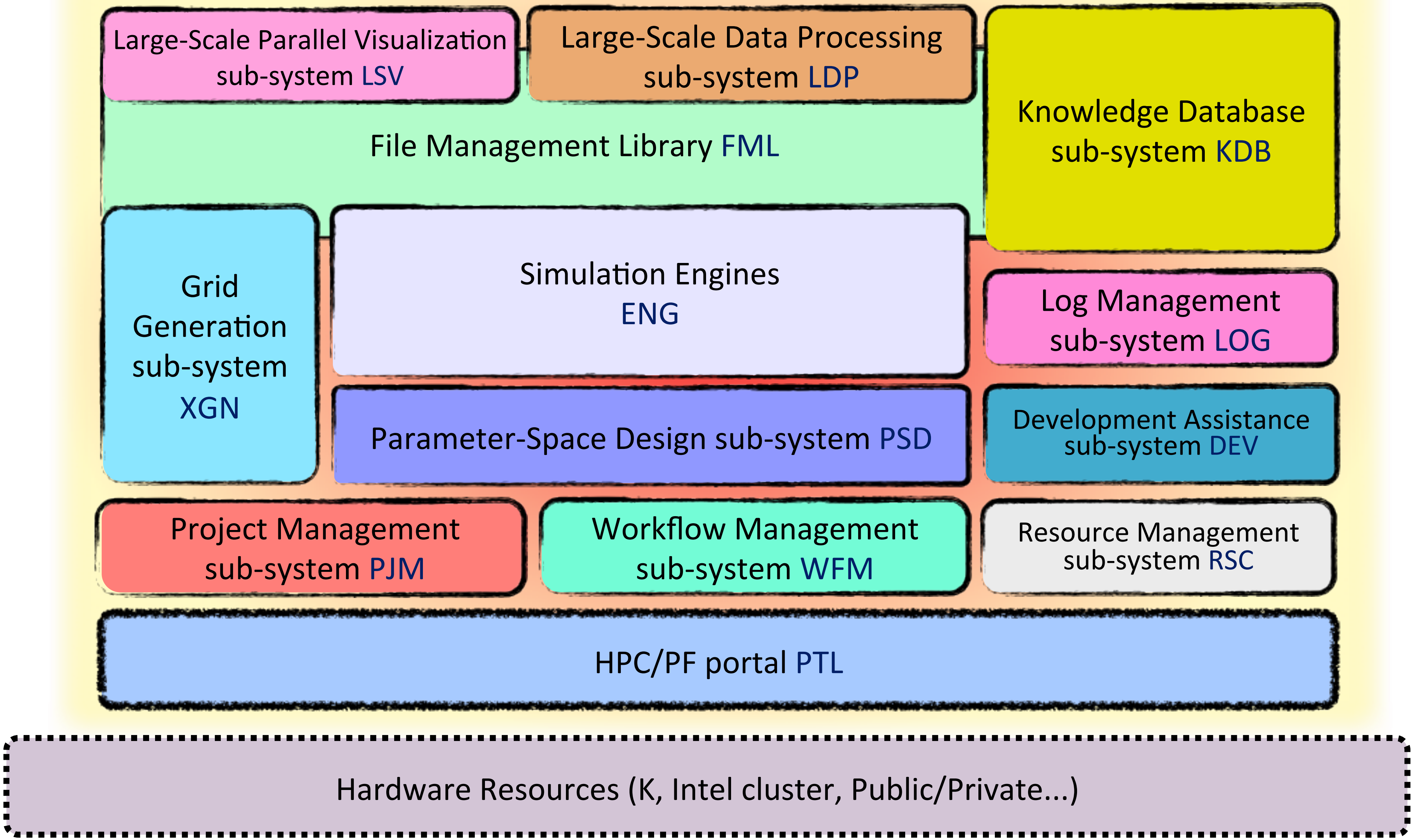
- **New paradigm of design**
  - demands EC and BD

Structure of HPC/PF



HPC/PF is an execution environment, which collects components required to efficiently perform a whole simulation process.

Components of HPC/PF



Large-Scale Parallel Visualization sub-system LSV

Large-Scale Data Processing sub-system LDP

Knowledge Database sub-system KDB

File Management Library FML

Grid Generation sub-system XGN

Simulation Engines ENG

Log Management sub-system LOG

Parameter-Space Design sub-system PSD

Development Assistance sub-system DEV

Project Management sub-system PJM

Workflow Management sub-system WFM

Resource Management sub-system RSC

HPC/PF portal PTL

Hardware Resources (K, Intel cluster, Public/Private...)

Statement : Software

- **Software libraries/tools need development and improvement**
  - Management of both HW resources(execution cores) and tasks all at once is required for in-situ data processing
  - A framework to describe multiple programs with good load-balancing

- **A middleware to efficiently build applications is demanded**
  - A middleware allows us to describe algorithm in higher-level and to avoid machine dependent code.

Statement : Software

- **Design of a system** that enables data-centric computation
  - Modular design for each component
  - Define a common information and an API to be shared with other components

Statement : Interoperability

- **Two points of view for provenance**
  - Inner-process
    - Inner-process provenance is managed by a process.
    - For instance, VisTrail
  - Inter-process
    - Inter-process provenance is managed by project level.
    - What is best way?

Summary

- **Design scenario**
- **Domain-specific approach** is straight forward way
  - Data structure and taxonomy of parallelization
- **Resource and task management** is essential
  - A framework is demanded
- **System design** for BD and EC

Remarks on Big Data Clustering (and its visualization)

Big Data and Extreme-scale Computing (BDEC)
Charleston SC May 1 2013

Geoffrey Fox
gcf@indiana.edu
http://www.infomall.org/

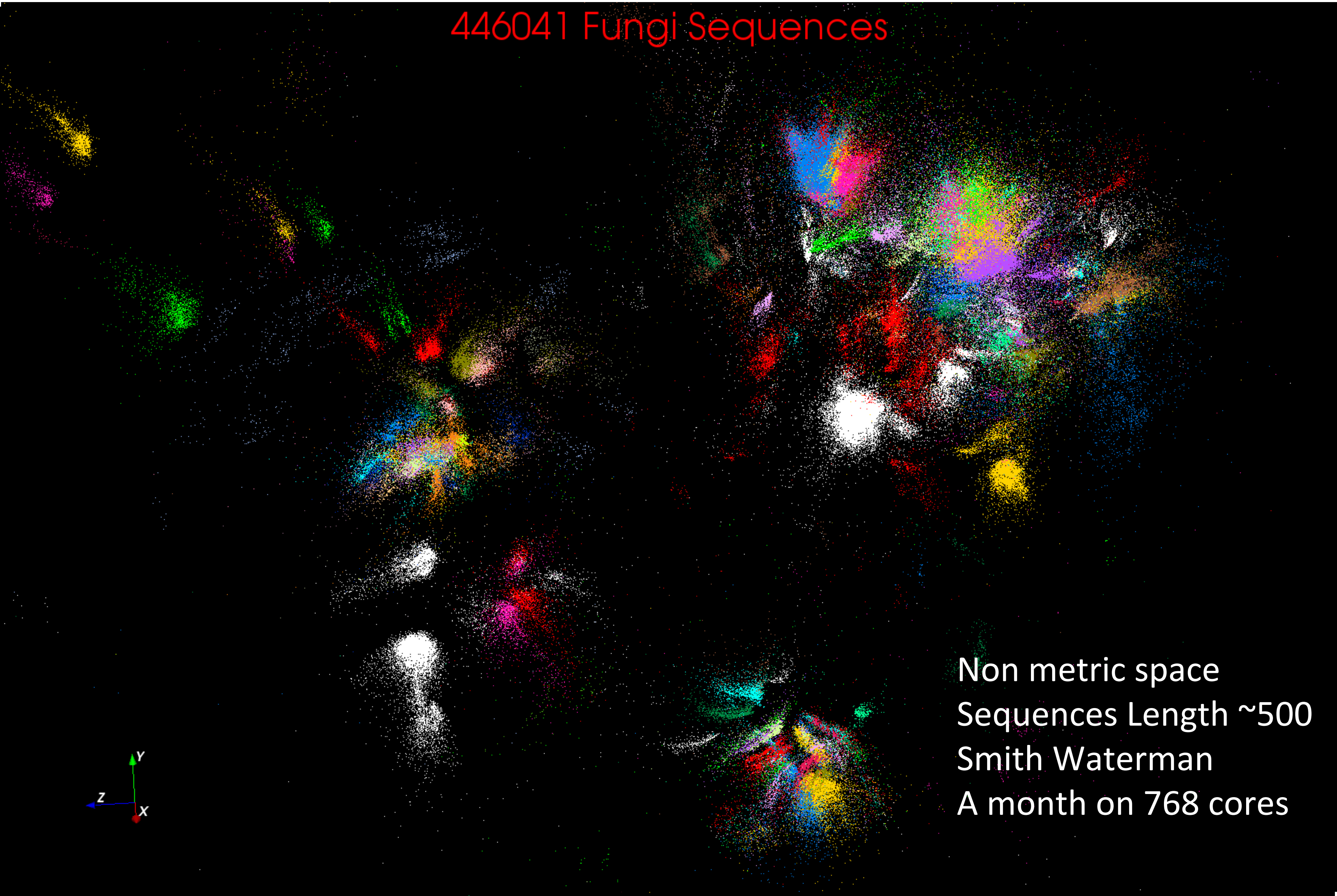
School of Informatics and Computing
Indiana University Bloomington

2013

Remarks on Clustering and MDS

- The standard data libraries (**R**, **Matlab**, **Mahout**) do not have best algorithms/software in either functionality or scalable parallelism
- A lot of algorithms are built around "classic **full matrix**" kernels
- **Clustering**, **Gaussian Mixture Models**, **PLSI** (probabilistic latent semantic indexing), **LDA** (Latent Dirichlet Allocation) similar
- **Multi-Dimensional Scaling** (MDS) classic information visualization algorithm for high dimension spaces (map preserving distances)
- **Vector** O(N) and **Non Vector semimetric** O(N^2) space cases for N points; "all" apps are points in spaces – not all "Proper linear spaces"
- Trying to release ~most powerful (in features/performance) available Clustering and MDS library although unfortunately in C#
- **Supported Features:** Vector, Non-Vector, Deterministic annealing, Hierarchical, sharp (trimmed) or general cluster sizes, Fixed points and general weights for MDS, (generalized Elkans algorithm)

~125 Clusters from Fungi sequence set



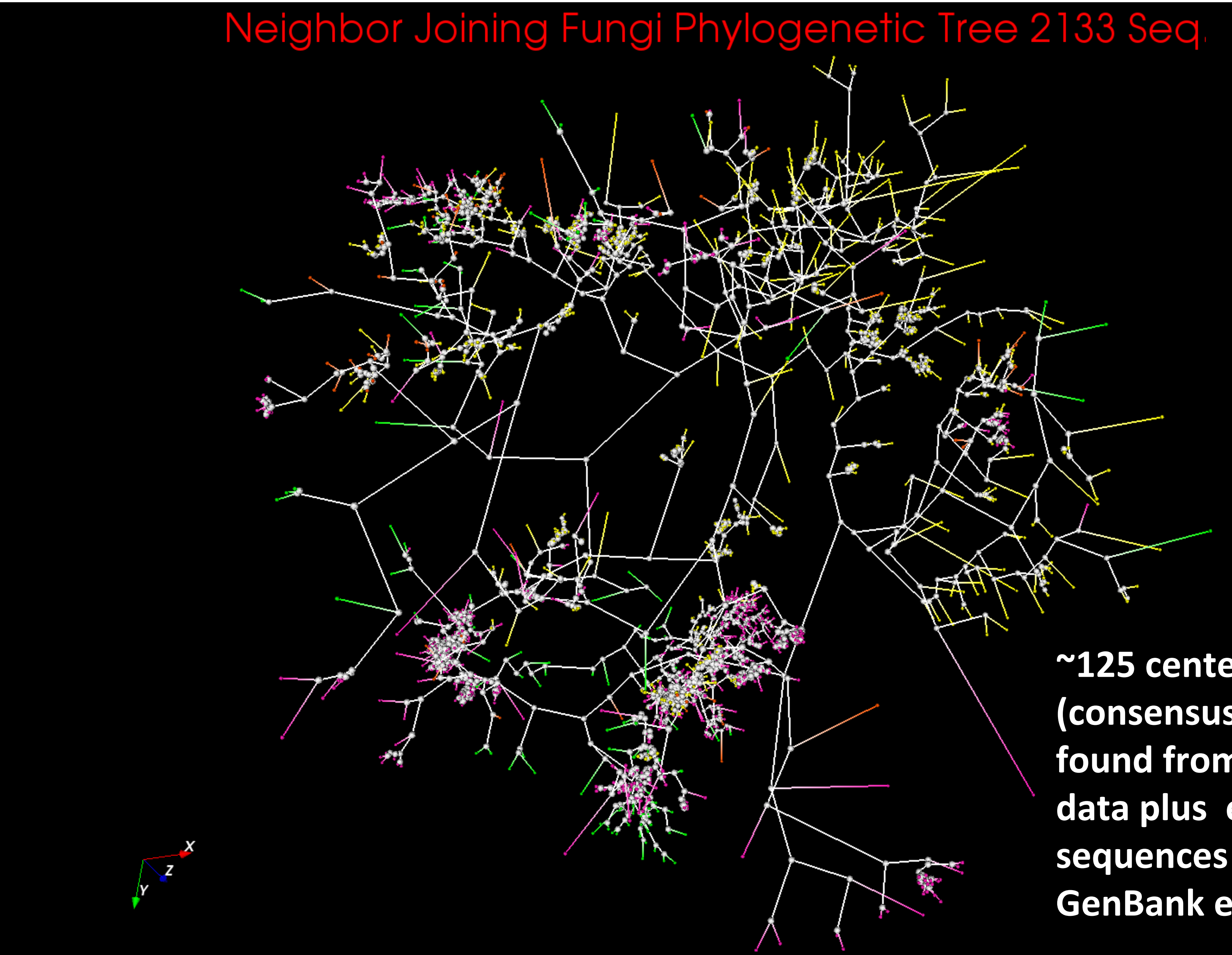
Non metric space

Sequences Length ~500

Smith Waterman

A month on 768 cores

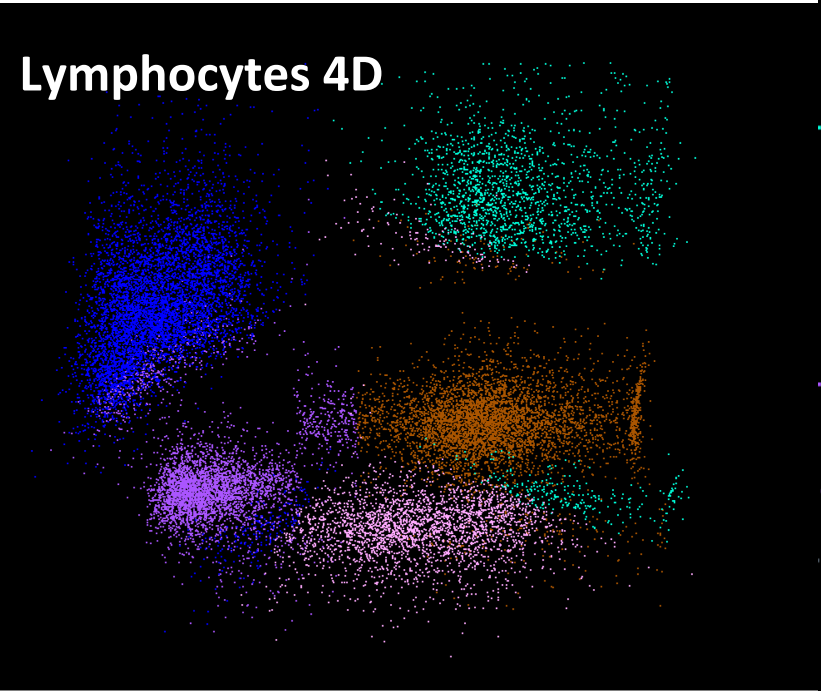
Phylogenetic Trees in 3D (usual 1D)

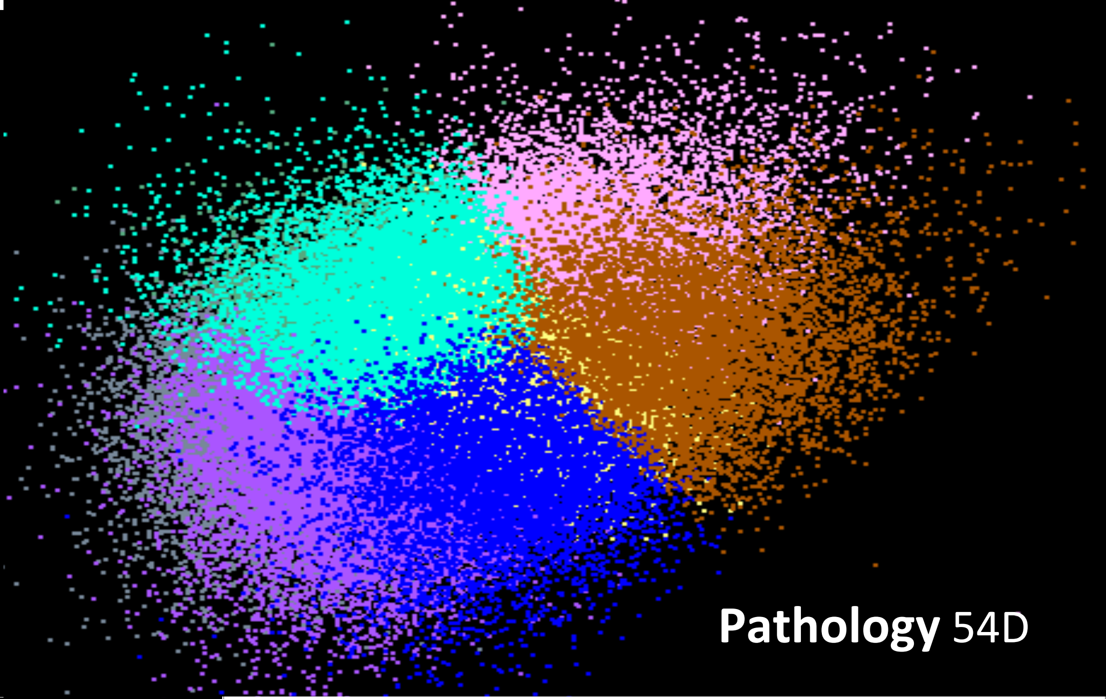


~125 centers (consensus vectors) found from Fungi data plus existing sequences from GenBank etc.

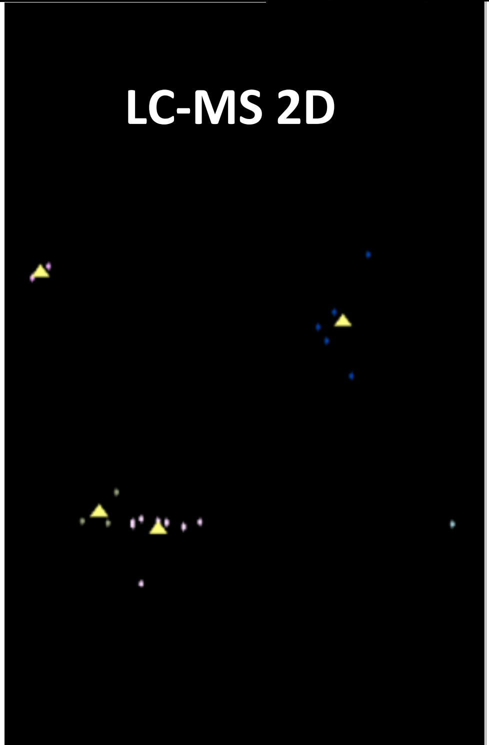
Clustering + MDS Applications

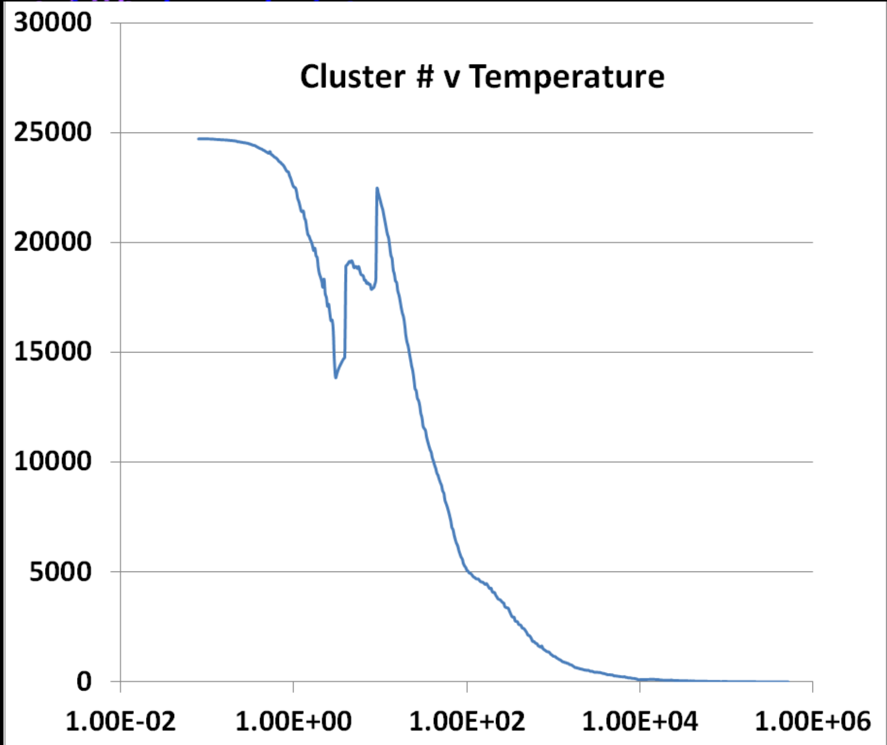
- Cases where “**real clusters**” as in genomics
- Cases as in pathology, proteomics, deep learning and recommender systems (Amazon, Netflix ….) where used for unsupervised **classification** of related items
- Recent “deep learning” papers either use Neural networks with **40 million- 11 billion parameters (10-50 million YouTube images)** or (Kmeans) Clustering with up to **1-10 million clusters**
  - Applications include automatic (Face) recognition; Autonomous driving; Pathology detection (Saltz)
  - Generalize to χ^2 fit of all (Internet) data to a model
  - Internet offers **“infinite” image** and **text** data
- **MDS** (map all points to 3D for visualization) can be used to verify “correctness” of analysis and/or to browse data as in **Geographical Information Systems**
- **Mini-app** of Joel Saltz
- **Ab-initio** (hardest, compute dominated) and **Update** (streaming, interpolation)





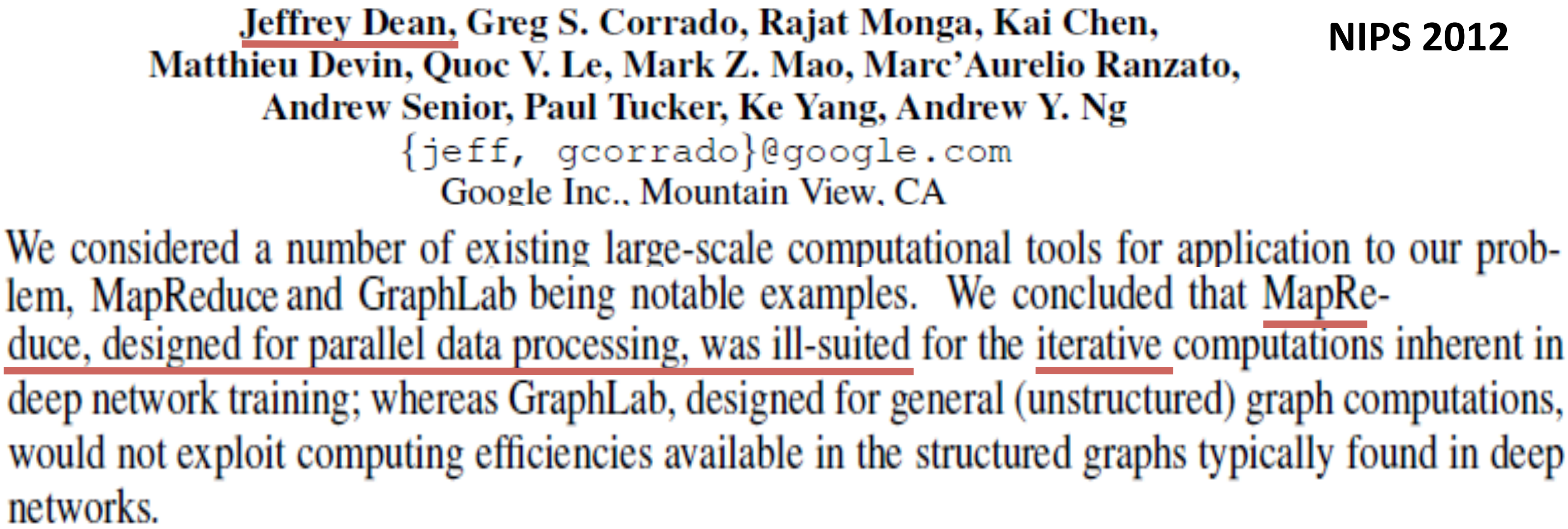
- Comparison of clustering and classification (top right)
- LC-MS Mass Spectrometry Sharp Clusters as known error in measurement





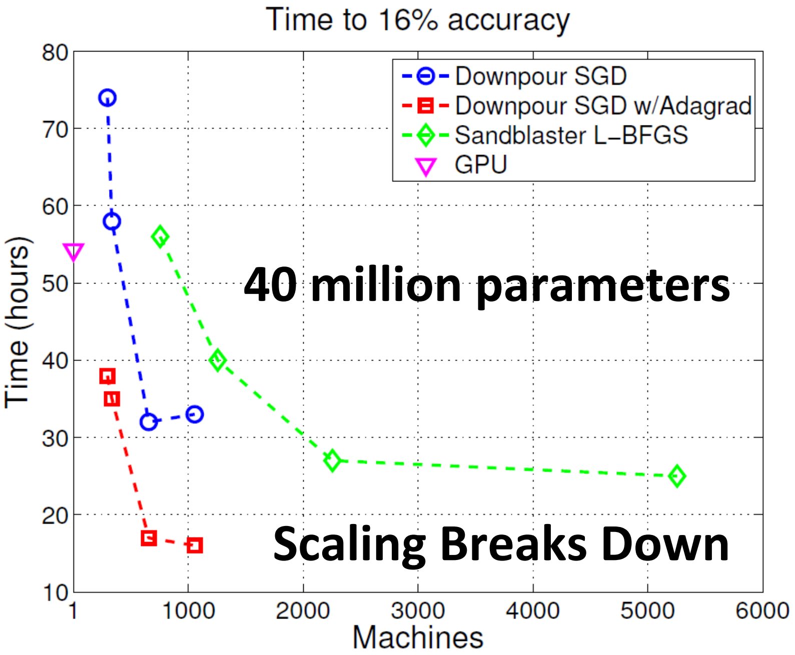
Large Scale Distributed Deep Networks

NIPS 2012



Jeffrey Dean, Greg S. Corrado, Rajat Monga, Kai Chen,
Matthieu Devin, Quoc V. Le, Mark Z. Mao, Marc'Aurelio Ranzato,
Andrew Senior, Paul Tucker, Ke Yang, Andrew Y. Ng
{jeff, gcorrado}@google.com
Google Inc., Mountain View, CA

We considered a number of existing large-scale computational tools for application to our problem, MapReduce and GraphLab being notable examples. We concluded that MapReduce, designed for parallel data processing, was ill-suited for the iterative computations inherent in deep network training; whereas GraphLab, designed for general (unstructured) graph computations, would not exploit computing efficiencies available in the structured graphs typically found in deep networks.



40 million parameters

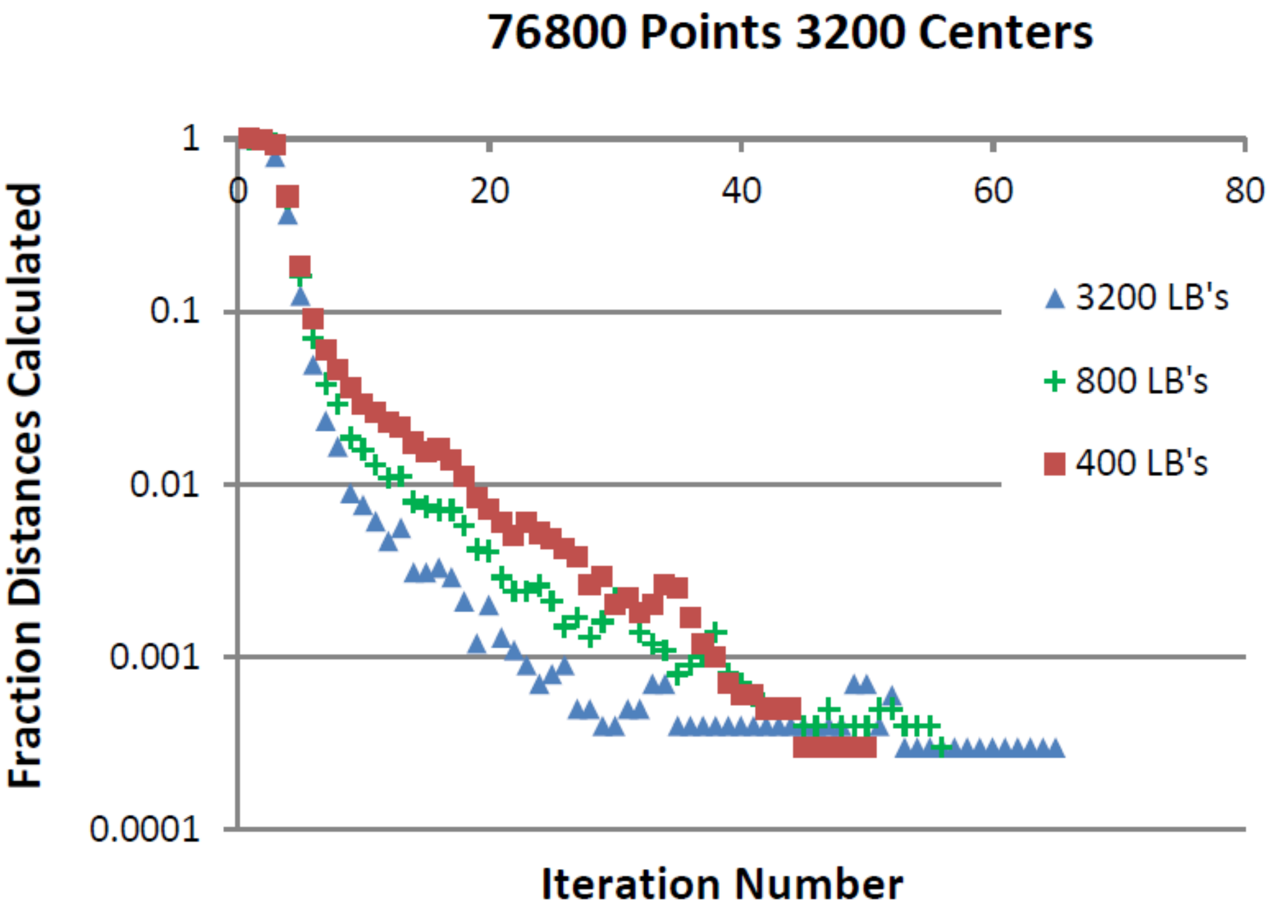
Scaling Breaks Down

- **DistBelief** (Google) rejected MapReduce but still didn't work well
- Coates and Ng (Stanford) et al. redid much larger problem on HPC cluster with Infiniband with 16 nodes and 64 GPU's
- Could use Iterative MapReduce (Twister) with GPU's

Triangle Inequality and Kmeans

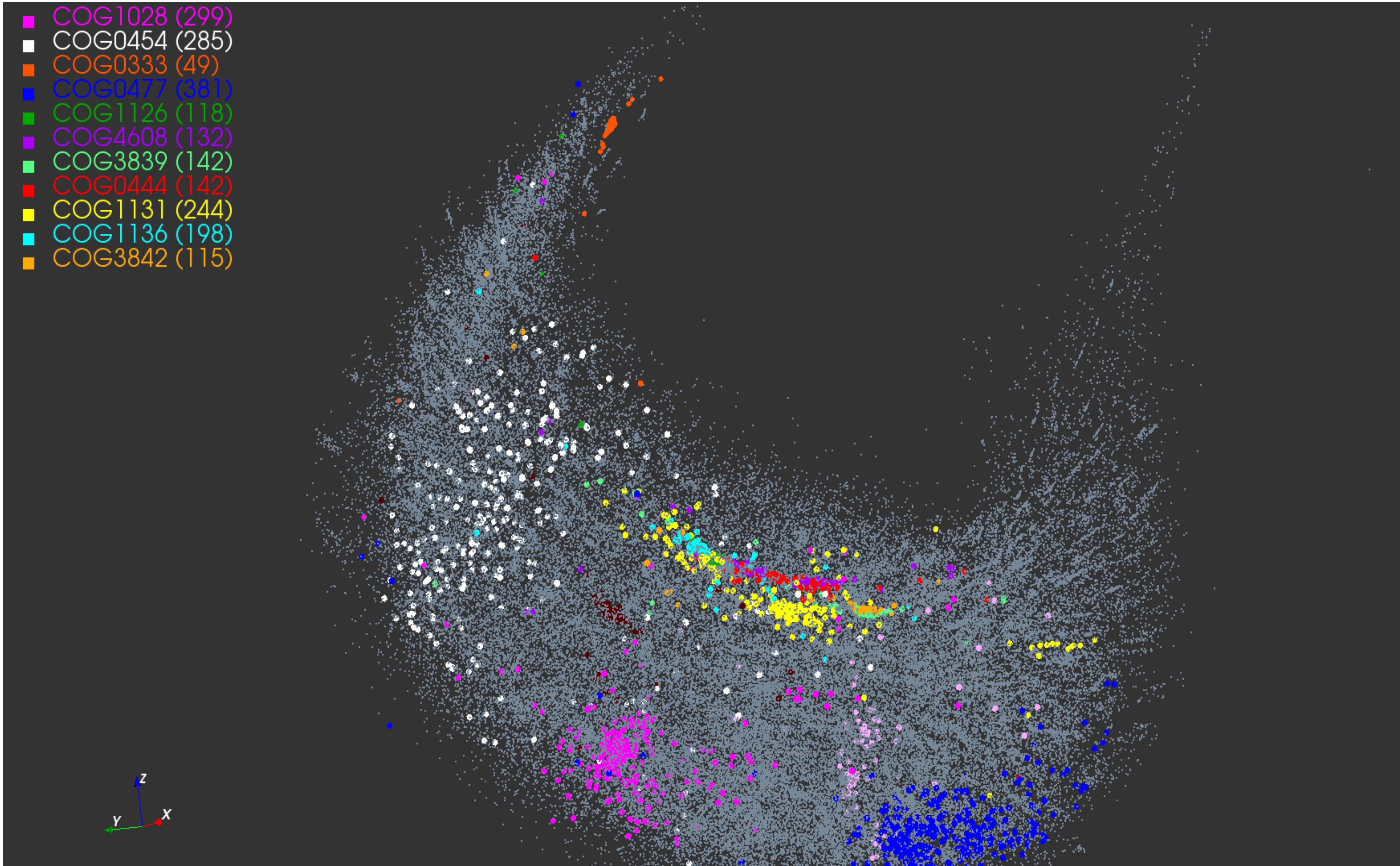
- Dominant part of Kmeans algorithm is finding nearest center to each point
  O(#Points * #Clusters * Vector Dimension)
- Simple algorithms finds
  min over centers c: d(x, c) = distance(point x, center c)
- But most of d(x, c) calculations are wasted as much larger than minimum value
- Elkan (2003) showed how to use triangle inequality to speed up using relations like
  d(x, c) >= d(x,c-last) – d(c, c-last)
  c-last position of center at last iteration
- So compare **d(x,c-last) – d(c, c-last)** with **d(x, c-best)** where c-best is nearest cluster at last iteration
- Complexity reduced by a factor = Vector Dimension and so this important in clustering high dimension spaces such as social imagery with 512 or more features per image
- GPU performance unclear

Fraction of Point-Center Distances Calculated in Kmeans D=2048



Fraction of Point-Center Distances calculated for 3 versions of the algorithm for 76800 points and 3200 centers in a 2048 dimensional space for three choices of lower bounds LB kept per point

Protein Universe Browser for COG Sequences with a few illustrative biologically identified clusters



I apologize that I come from other end of problem …..

Undergraduate X-Informatics Class

http://www.infomall.org/X-InformaticsSpring2013/

Big data MOOC http://x-informatics.appspot.com/preview

Mantra of class

Big Data Ecosystem in One Sentence

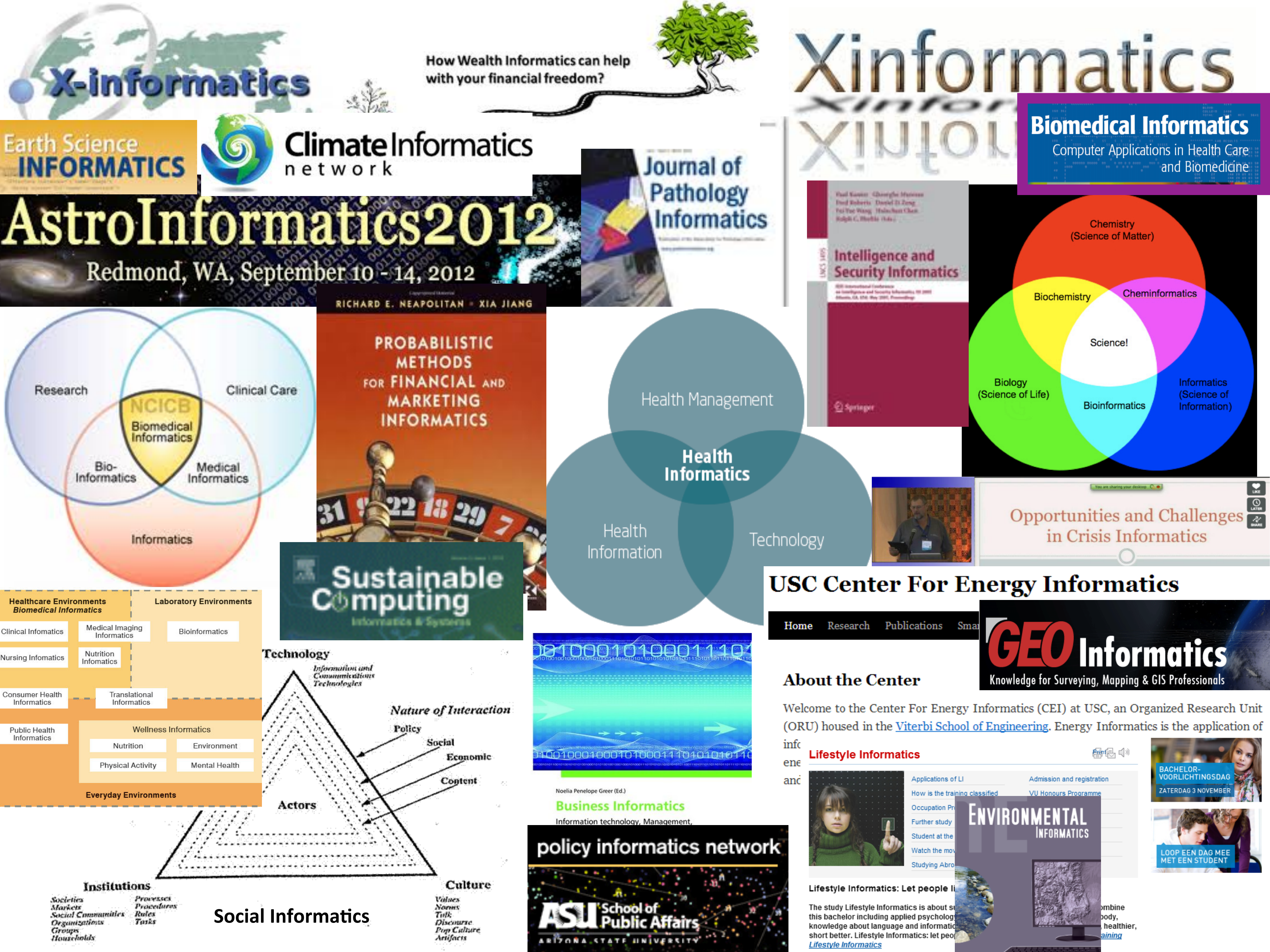
Use Clouds running Data Analytics processing Big Data to solve problems in X-Informatics (or e-X)

X = Astronomy, Biology, Biomedicine, Business, Chemistry, Climate, Crisis, Earth Science, Energy, Environment, Finance, Health, Intelligence, Lifestyle, Marketing, Medicine, Pathology, Policy, Radar, Security, Sensor, Social, Sustainability, Wealth and Wellness with more fields (physics) defined implicitly

Spans Industry and Science (research)

Education: Data Science see recent New York Times articles

http://datascience101.wordpress.com/2013/04/13/new-york-times-data-science-articles/



About the Center

Welcome to the Center For Energy Informatics (CEI) at USC, an Organized Research Unit (ORU) housed in the Viterbi School of Engineering. Energy Informatics is the application of
infc
ene
and

Lifestyle Informatics: Let people li

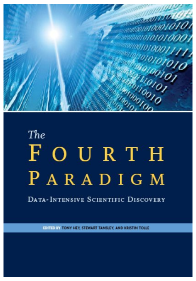
The study Lifestyle Informatics is about su ombine
this bachelor including applied psycholog body,
knowledge about language and informatic healthier,
short better. Lifestyle Informatics: let peo aining
Lifestyle Informatics

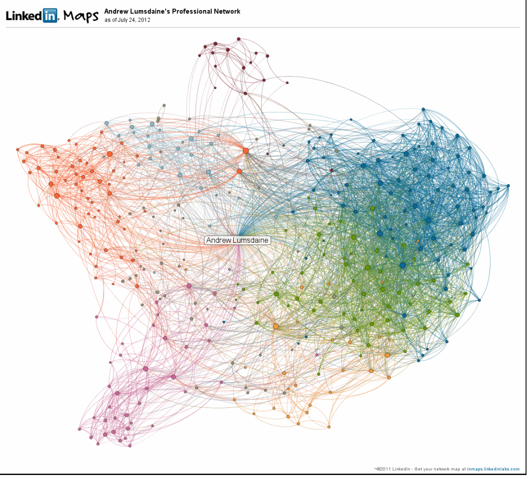
New Execution Models Are Required for Big Data at Exascale

Andrew Lumsdaine
Center for Research in Extreme Scale Technologies
Indiana University

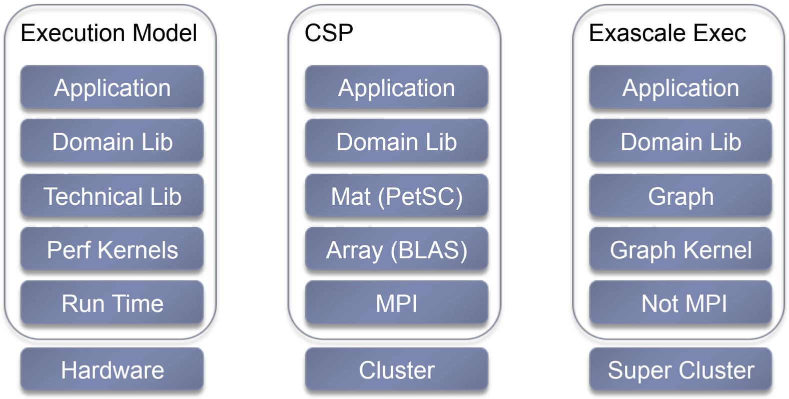
Extreme-Scale Computing

- Not just for PDEs anymore
- Graph abstraction important for Big Data problems


The
FOURTH
PARADIGM
DATA-INTENSIVE SCIENTIFIC DISCOVERY
EDITED BY TONY HEY, STEWART TANSLEY, AND KRISTIN TOLLE


LinkedIn Maps
Andrew Lumsdaine's Professional Network
as of July 24, 2012
Andrew Lumsdaine
¬©2011 LinkedIn - Get your network map at inmaps.linkedinlabs.com

Big Data and the Extreme Scale Ecosystem



Big Data at Extreme Scale

ss is data dependent

cation is data

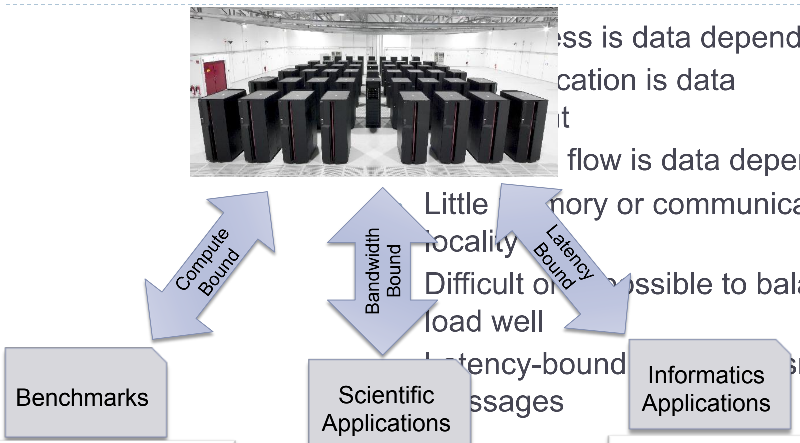
t

flow is data dependent

Little nory or communication locality

Difficult or ossible to balance load well

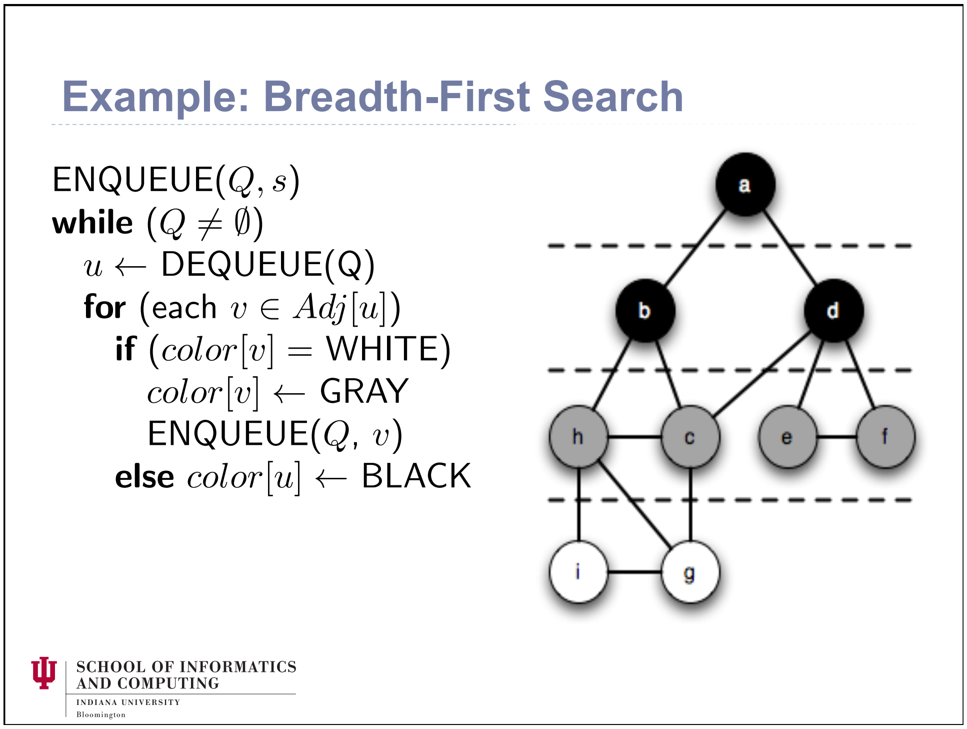
tency-bound small ssages



```
for (int i = 0; i < M; ++i)
  for (int j = 0; j < N; ++j)
    for (int k = 0; k < K; ++k)
      C[i][j] += A[i][k] * B[k][j];
```

```
for (int i = 0; i < M; ++i)
  for (int j = row[i]; j < row[i+1]; ++j)
    Y[i] += A[j] * X[col[j]];
```

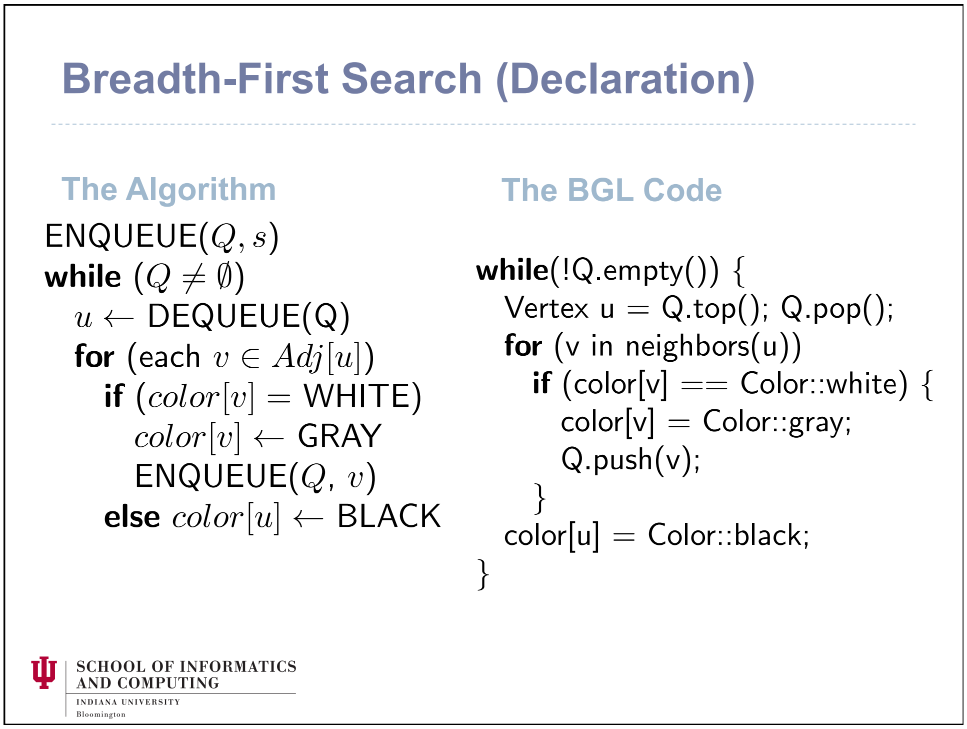
```
while (! Q.empty()) {
  Vertex u = Q.top(); Q.pop();
  for (v in neighbors(u)
    if (color[v] == Color::white()) {
      color[v] = Color::gray());
      Q.push(v);
    }
  color[u] = Color::black());
}
```



Example: Breadth-First Search

```
ENQUEUE($Q, s$)
while ($Q \neq \emptyset$)
    $u \leftarrow$ DEQUEUE(Q)
    for (each $v \in Adj[u]$)
        if ($color[v]$ = WHITE)
            $color[v] \leftarrow$ GRAY
            ENQUEUE($Q$, $v$)
        else $color[u] \leftarrow$ BLACK
```

SCHOOL OF INFORMATICS AND COMPUTING
INDIANA UNIVERSITY
Bloomington



Breadth-First Search (Declaration)

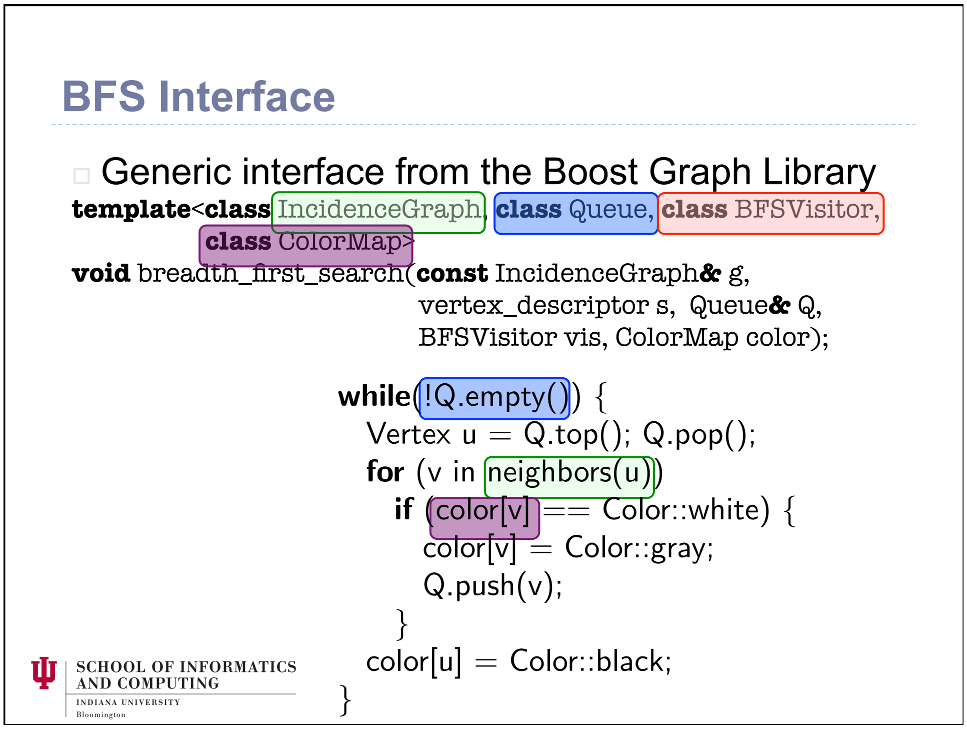
The Algorithm

```
ENQUEUE($Q, s$)
while ($Q \neq \emptyset$)
    $u \leftarrow$ DEQUEUE(Q)
    for (each $v \in Adj[u]$)
        if ($color[v]$ = WHITE)
            $color[v] \leftarrow$ GRAY
            ENQUEUE($Q$, $v$)
        else $color[u] \leftarrow$ BLACK
```

The BGL Code

```
while(!Q.empty()) {
  Vertex u = Q.top(); Q.pop();
  for (v in neighbors(u))
    if (color[v] == Color::white) {
      color[v] = Color::gray;
      Q.push(v);
    }
  color[u] = Color::black;
}
```

SCHOOL OF INFORMATICS AND COMPUTING
INDIANA UNIVERSITY
Bloomington

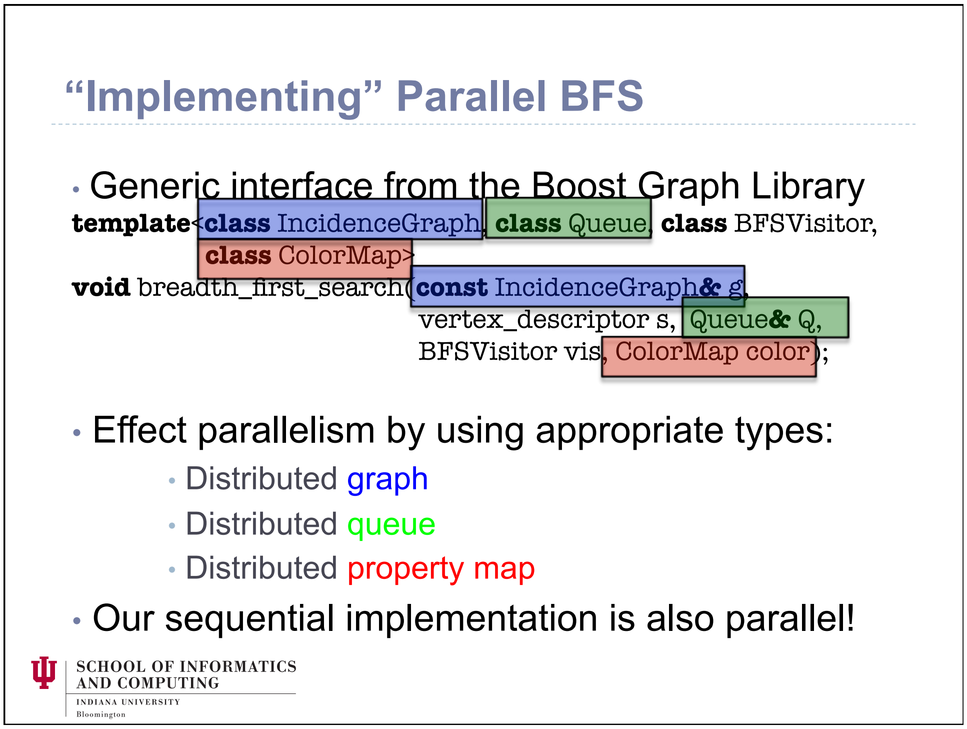


BFS Interface

□ Generic interface from the Boost Graph Library

```
template<class IncidenceGraph, class Queue, class BFSVisitor,
         class ColorMap>
void breadth_first_search(const IncidenceGraph& g,
                          vertex_descriptor s,  Queue& Q,
                          BFSVisitor vis, ColorMap color);

                while(!Q.empty()) {
                  Vertex u = Q.top(); Q.pop();
                  for (v in neighbors(u))
                    if (color[v] == Color::white) {
                      color[v] = Color::gray;
                      Q.push(v);
                    }
                  color[u] = Color::black;
                }
```

SCHOOL OF INFORMATICS
AND COMPUTING
INDIANA UNIVERSITY
Bloomington



“Implementing” Parallel BFS

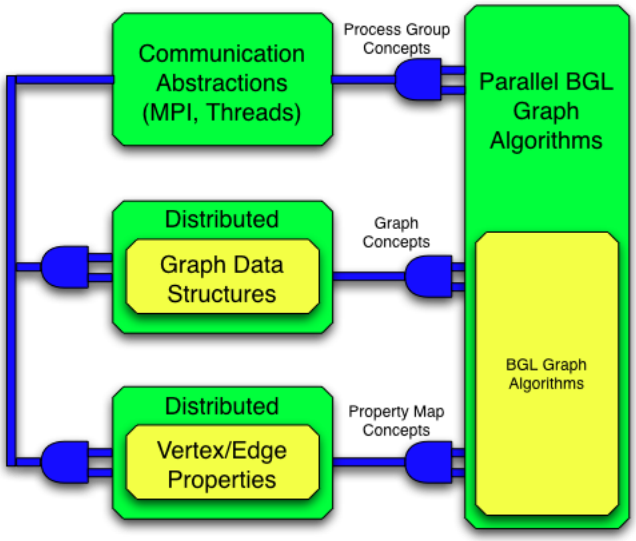
- Generic interface from the Boost Graph Library

```
template<class IncidenceGraph, class Queue, class BFSVisitor,
         class ColorMap>
void breadth_first_search(const IncidenceGraph& g,
                          vertex_descriptor s, Queue& Q,
                          BFSVisitor vis, ColorMap color);
```

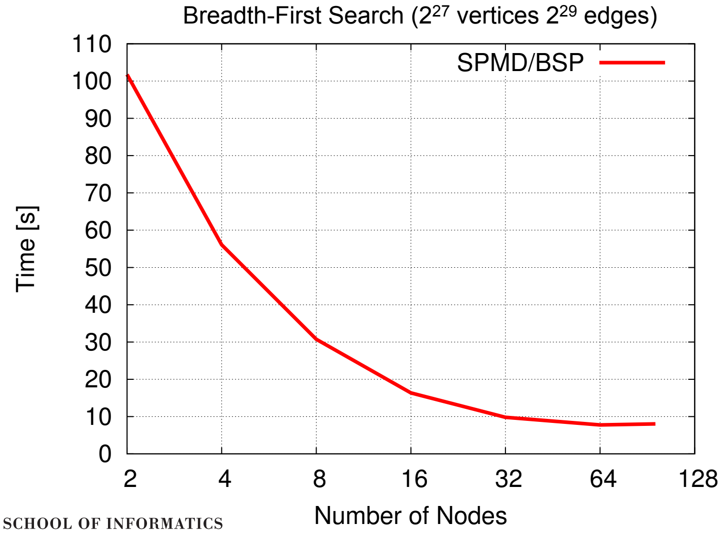
- Effect parallelism by using appropriate types:
    - Distributed graph
    - Distributed queue
    - Distributed property map
- Our sequential implementation is also parallel!

SCHOOL OF INFORMATICS
AND COMPUTING
INDIANA UNIVERSITY
Bloomington

Parallel BGL Architecture (CSP Model)

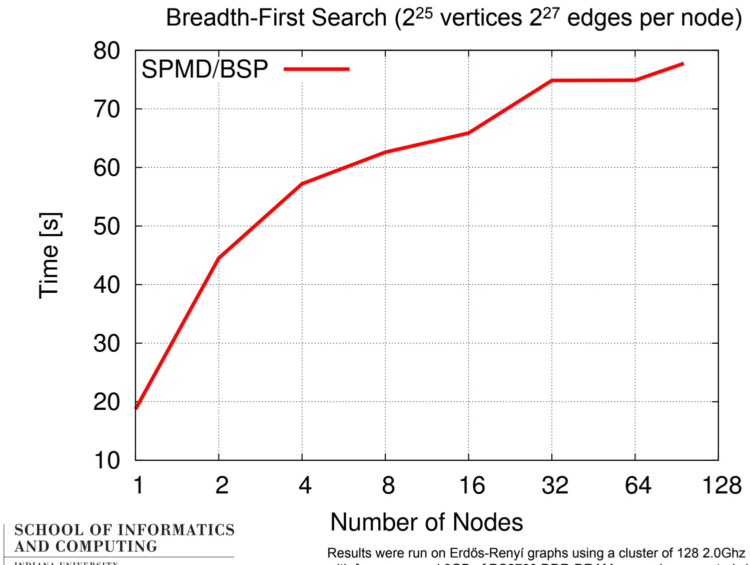


CSP Breadth-First Search (Strong Scaling)



Results were run on Erdős-Renyí graphs using a cluster of 128 2.0Ghz Opteron 270 processors with four cores and 8GB of PC2700 DDR-DRAM per node connected via SDR Infiniband.

CSP Breadth-First Search (Weak Scaling)

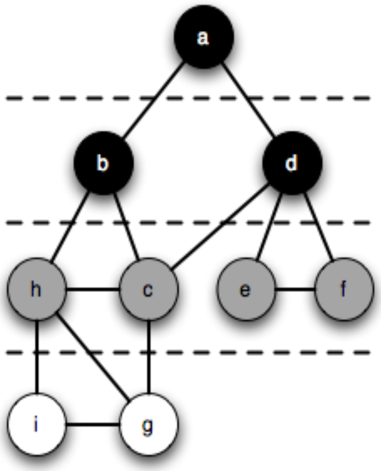


Results were run on Erdős-Renyí graphs using a cluster of 128 2.0Ghz Opteron 270 processors with four cores and 8GB of PC2700 DDR-DRAM per node connected via SDR Infiniband.

SCHOOL OF INFORMATICS AND COMPUTING
INDIANA UNIVERSITY
Bloomington

Find the Sequential Trap

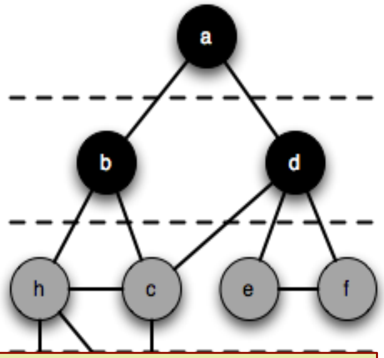
```
ENQUEUE($Q, s$)
while ($Q \neq \emptyset$)
    $u \leftarrow$ DEQUEUE(Q)
    for (each $v \in Adj[u]$)
        if ($color[v]$ = WHITE)
            $color[v] \leftarrow$ GRAY
            ENQUEUE($Q$, $v$)
        else $color[u] \leftarrow$ BLACK
```



SCHOOL OF INFORMATICS AND COMPUTING
INDIANA UNIVERSITY
Bloomington

Find the Synchronization Trap

```
ENQUEUE($Q, s$)
while ($Q \neq \emptyset$)
  $u \leftarrow$ DEQUEUE(Q)
  for (each $v \in Adj[u]$)
    if ($color[v]$ = WHITE)
      $color[v] \leftarrow$ GRAY
      ENQUEUE($Q$, $v$)
    else $color[u] \leftarrow$ BLACK
```

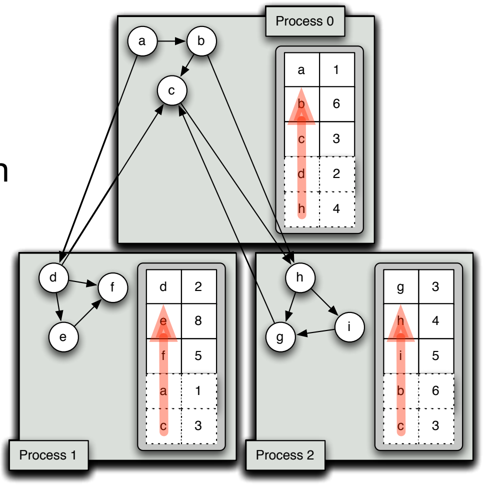


for *i* in *ranks*: start receiving *in_queue*[*i*] from rank *i*
for *j* in *ranks*: start sending *out_queue*[*j*] to rank *j*
synchronize and finish communications

SCHOOL OF INFORMATICS AND COMPUTING
INDIANA UNIVERSITY
Bloomington

Data Storage and Data Movement Trap

- Perform remote data access
- Barrier
- Use received data
- Barrier
- Full network RTT on every message
- Data reuse unlikely



a	1
b	6
c	3
d	2
h	4

d	2
e	8
f	5
a	1
c	3

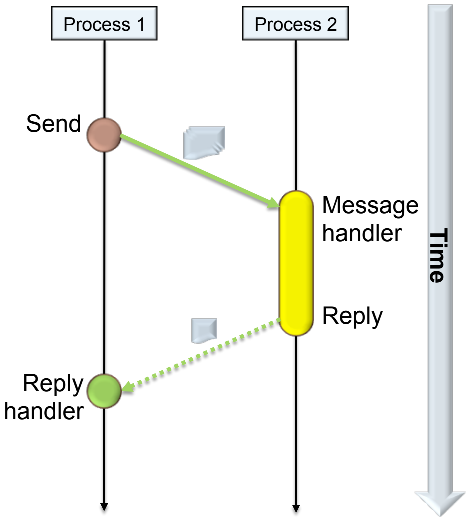
g	3
h	4
i	5
b	6
c	3

SCHOOL OF INFORMATICS AND COMPUTING
INDIANA UNIVERSITY
Bloomington

21

Active Messages

- Created by von Eicken et al, for Split-C (1992)
- Messages sent explicitly
- Receivers register handlers but are not involved with individual messages
- Messages typically asynchronous for higher throughput

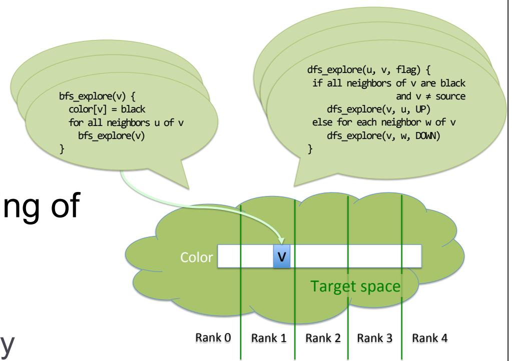


Active Pebbles

- Programming model
  - Active messages (*pebbles*)
  - Fine-grained addressing (*targets*)
- Execution model
  - Flexible message coalescing
  - Message reductions
  - Active routing
  - Termination detection
- Features are synergistic
- AM++ is our implementation of Active Pebbles model

Programming Model

- Program with natural granularity
  - No need to artificially coarsen object granularity
- Transparent addressing
  - Static and dynamic
  - Local and remote
- Bulk, anonymous handling of messages and targets
- Epoch model
  - Enforce message delivery
  - Controlled relaxed consistency

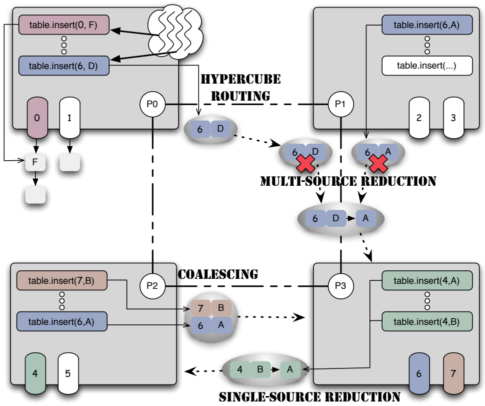


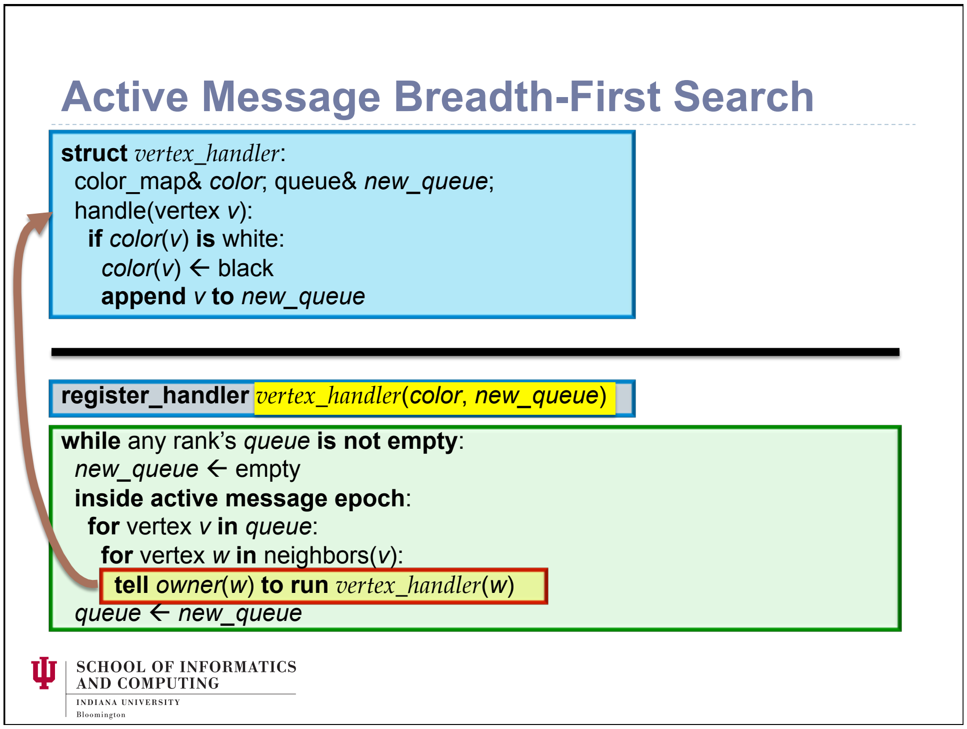
```
bfs_explore(v) {
  color[v] = black
  for all neighbors u of v
    bfs_explore(v)
}
```

```
dfs_explore(u, v, flag) {
 if all neighbors of v are black
                    and v ≠ source
   dfs_explore(v, u, UP)
 else for each neighbor w of v
   dfs_explore(v, w, DOWN)
}
```

Execution Model

- Message coalescing
  - Amortize latency
- Message reduction
  - Eliminate redundant computation
  - Distributed computation into network
- Active routing
  - Exploit physical topology
- Termination detection
  - Handlers send messages
  - Detect quiescence





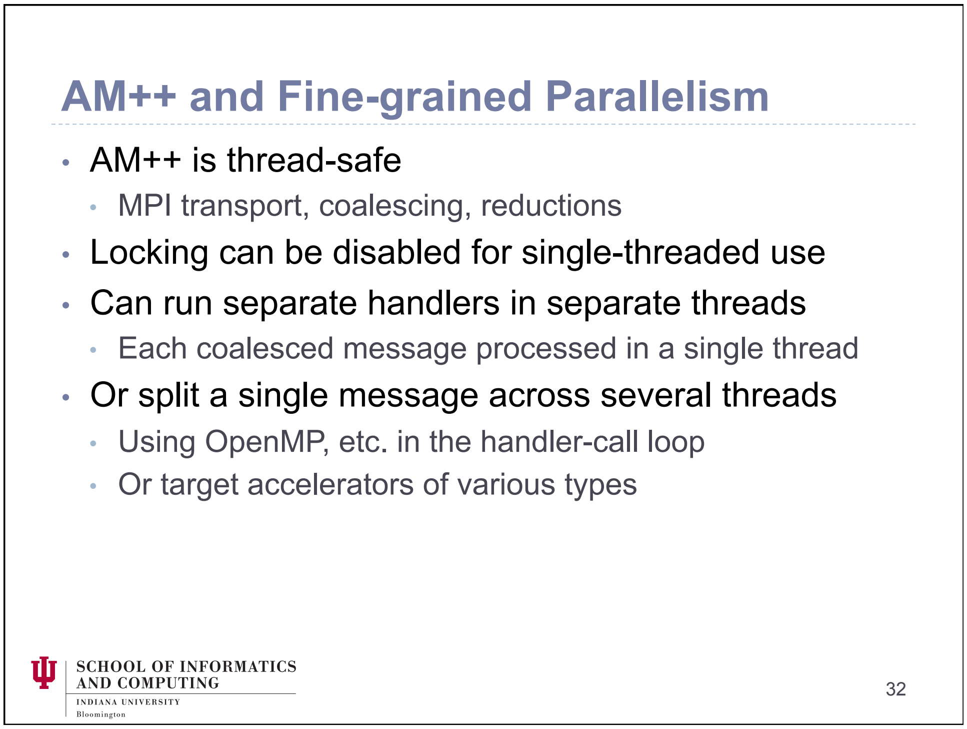
Active Message Breadth-First Search

```
struct vertex_handler:
  color_map& color; queue& new_queue;
  handle(vertex v):
    if color(v) is white:
      color(v) ← black
      append v to new_queue
```

```
register_handler vertex_handler(color, new_queue)
```

```
while any rank's queue is not empty:
  new_queue ← empty
  inside active message epoch:
    for vertex v in queue:
      for vertex w in neighbors(v):
        tell owner(w) to run vertex_handler(w)
  queue ← new_queue
```

SCHOOL OF INFORMATICS AND COMPUTING
INDIANA UNIVERSITY
Bloomington

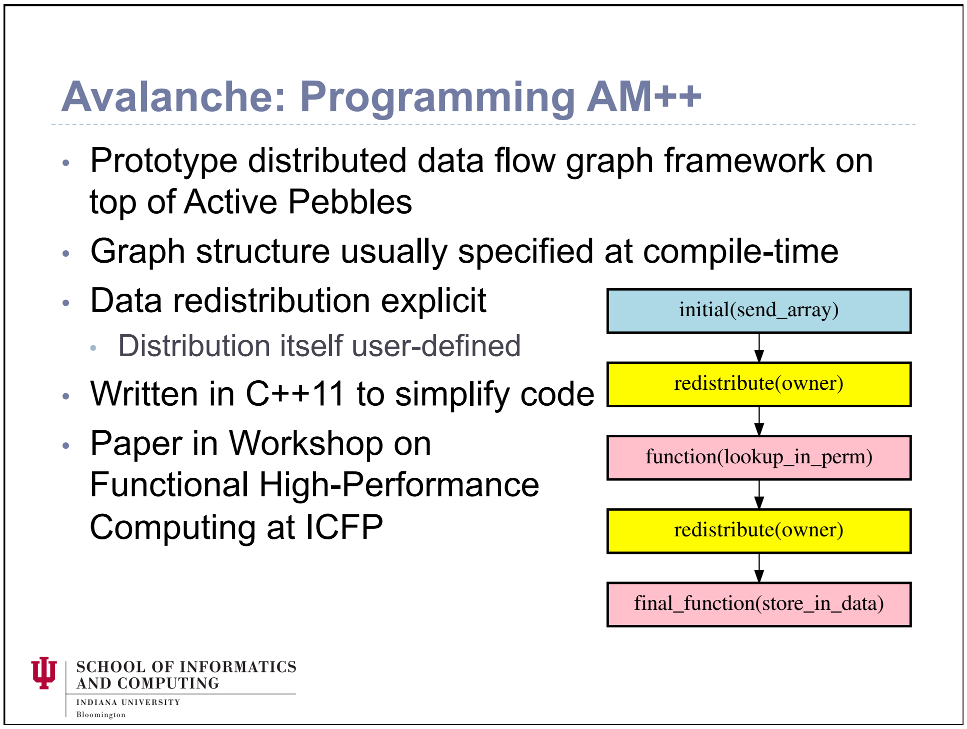


AM++ and Fine-grained Parallelism

- AM++ is thread-safe
  - MPI transport, coalescing, reductions
- Locking can be disabled for single-threaded use
- Can run separate handlers in separate threads
  - Each coalesced message processed in a single thread
- Or split a single message across several threads
  - Using OpenMP, etc. in the handler-call loop
  - Or target accelerators of various types

SCHOOL OF INFORMATICS AND COMPUTING
INDIANA UNIVERSITY
Bloomington

32



Avalanche: Programming AM++

- Prototype distributed data flow graph framework on top of Active Pebbles
- Graph structure usually specified at compile-time
- Data redistribution explicit
  - Distribution itself user-defined
- Written in C++11 to simplify code
- Paper in Workshop on Functional High-Performance Computing at ICFP

initial(send_array)

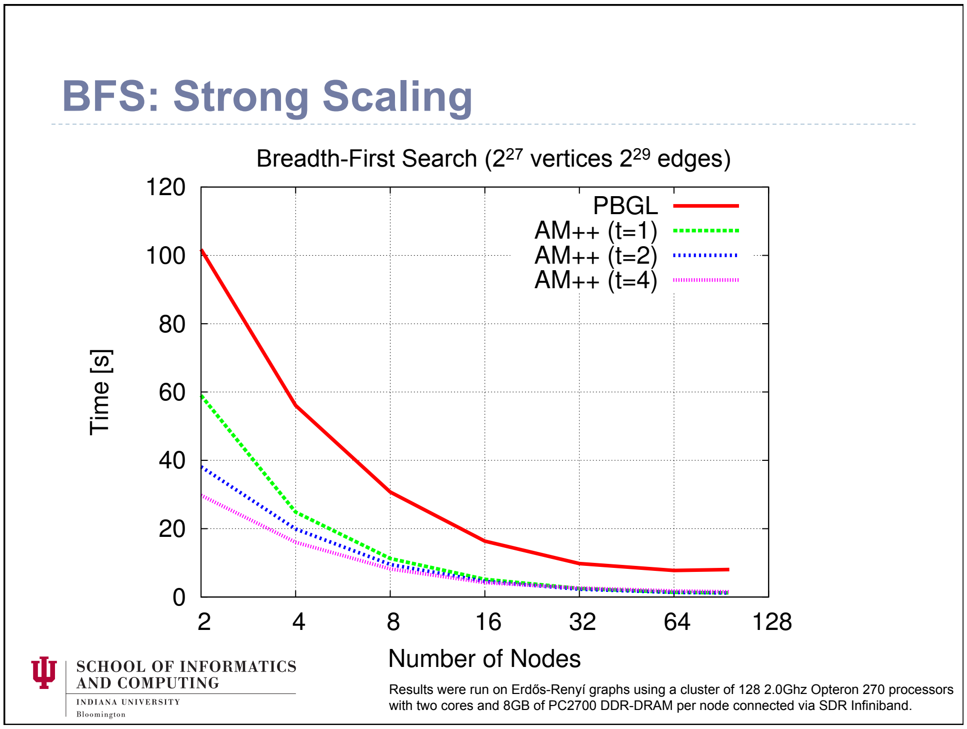
redistribute(owner)

function(lookup_in_perm)

redistribute(owner)

final_function(store_in_data)

SCHOOL OF INFORMATICS AND COMPUTING
INDIANA UNIVERSITY
Bloomington

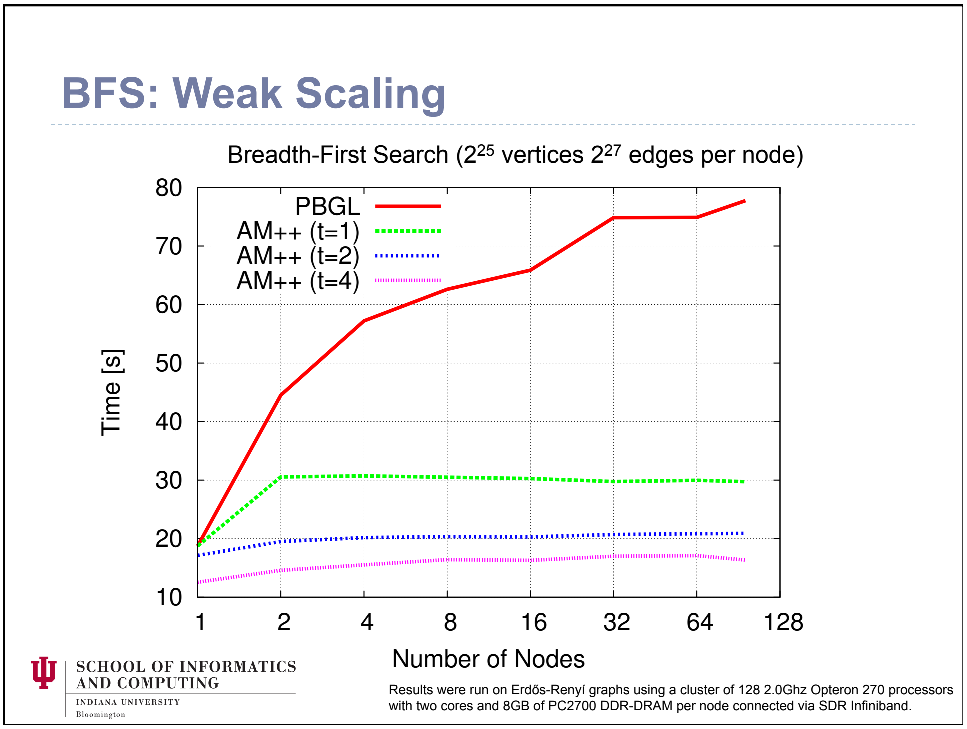


BFS: Strong Scaling

Breadth-First Search (2^{27} vertices 2^{29} edges)

Results were run on Erdős-Renyí graphs using a cluster of 128 2.0Ghz Opteron 270 processors with two cores and 8GB of PC2700 DDR-DRAM per node connected via SDR Infiniband.

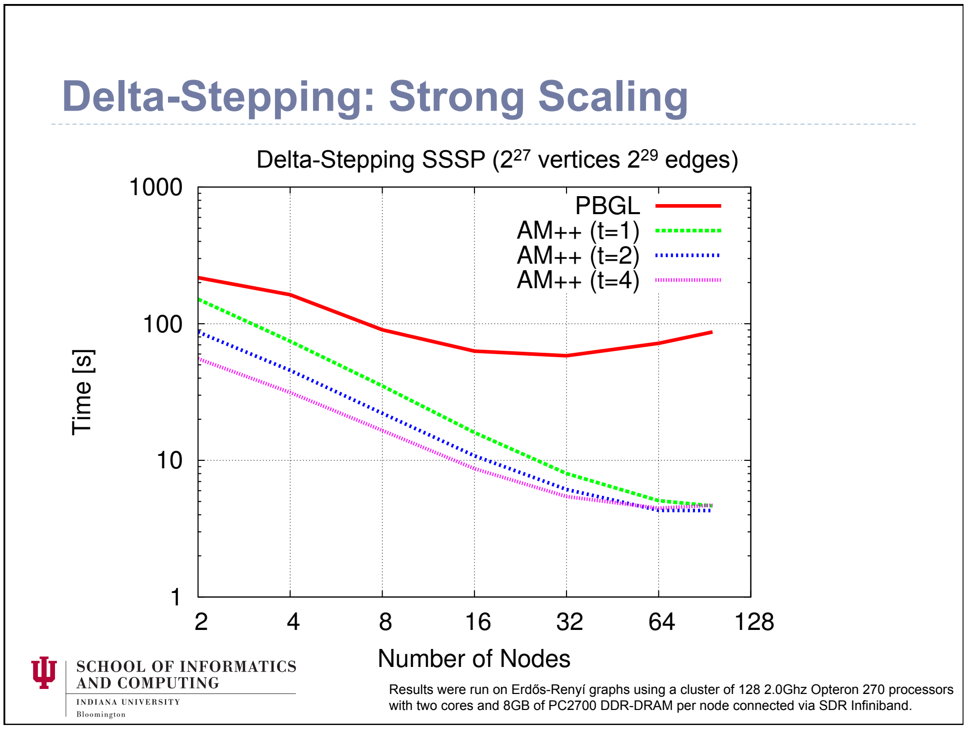
SCHOOL OF INFORMATICS AND COMPUTING
INDIANA UNIVERSITY
Bloomington



BFS: Weak Scaling

SCHOOL OF INFORMATICS AND COMPUTING
INDIANA UNIVERSITY
Bloomington

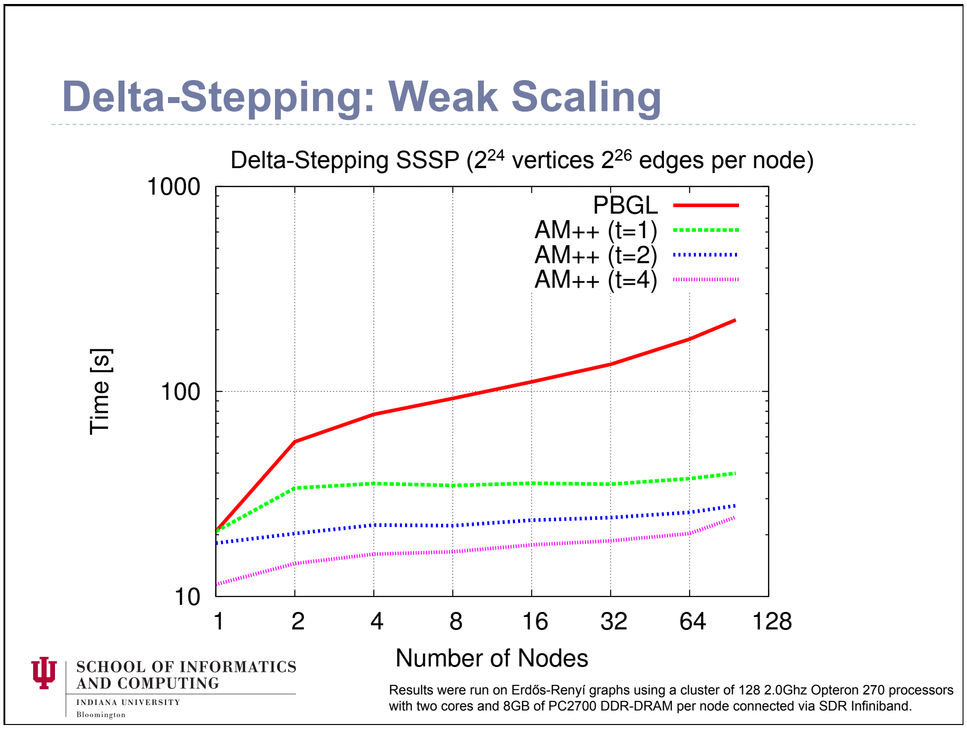
Results were run on Erdős-Renyí graphs using a cluster of 128 2.0Ghz Opteron 270 processors with two cores and 8GB of PC2700 DDR-DRAM per node connected via SDR Infiniband.



Delta-Stepping: Strong Scaling

SCHOOL OF INFORMATICS AND COMPUTING
INDIANA UNIVERSITY
Bloomington

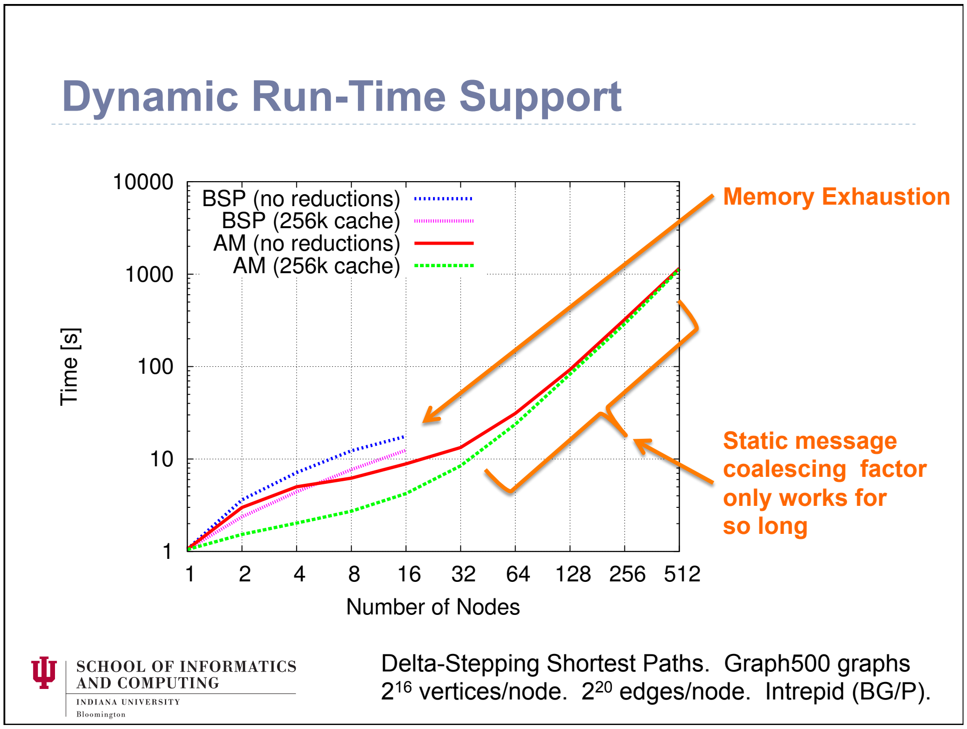
Results were run on Erdős-Renyí graphs using a cluster of 128 2.0Ghz Opteron 270 processors with two cores and 8GB of PC2700 DDR-DRAM per node connected via SDR Infiniband.



Delta-Stepping: Weak Scaling

SCHOOL OF INFORMATICS AND COMPUTING
INDIANA UNIVERSITY
Bloomington

Results were run on Erdős-Renyí graphs using a cluster of 128 2.0Ghz Opteron 270 processors with two cores and 8GB of PC2700 DDR-DRAM per node connected via SDR Infiniband.



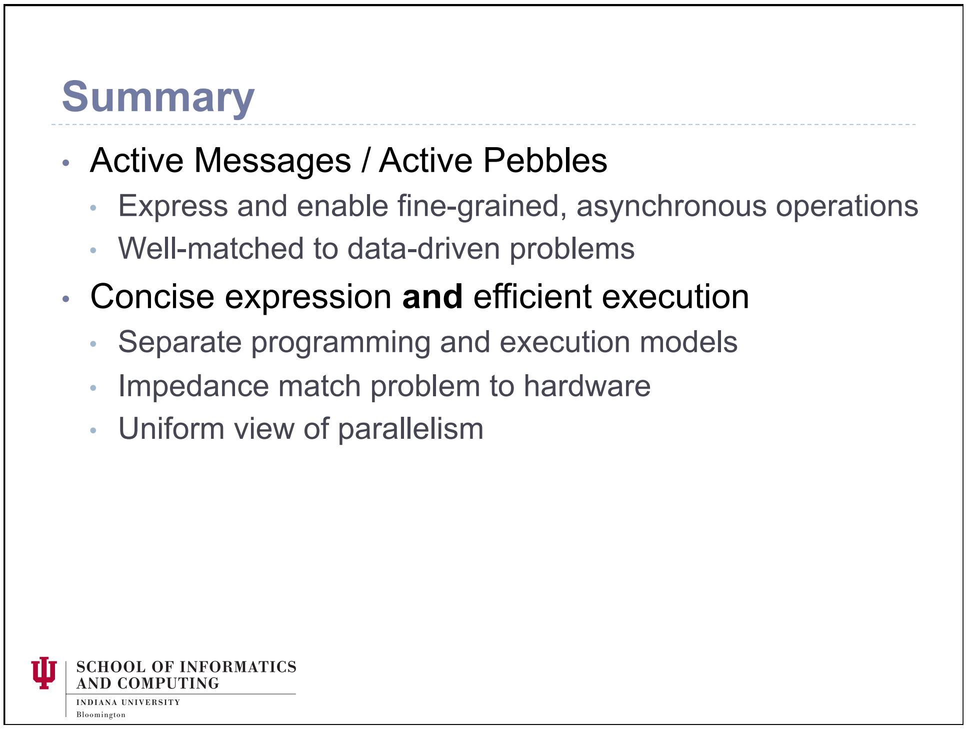
Dynamic Run-Time Support

Memory Exhaustion

Static message coalescing factor only works for so long

SCHOOL OF INFORMATICS AND COMPUTING
INDIANA UNIVERSITY
Bloomington

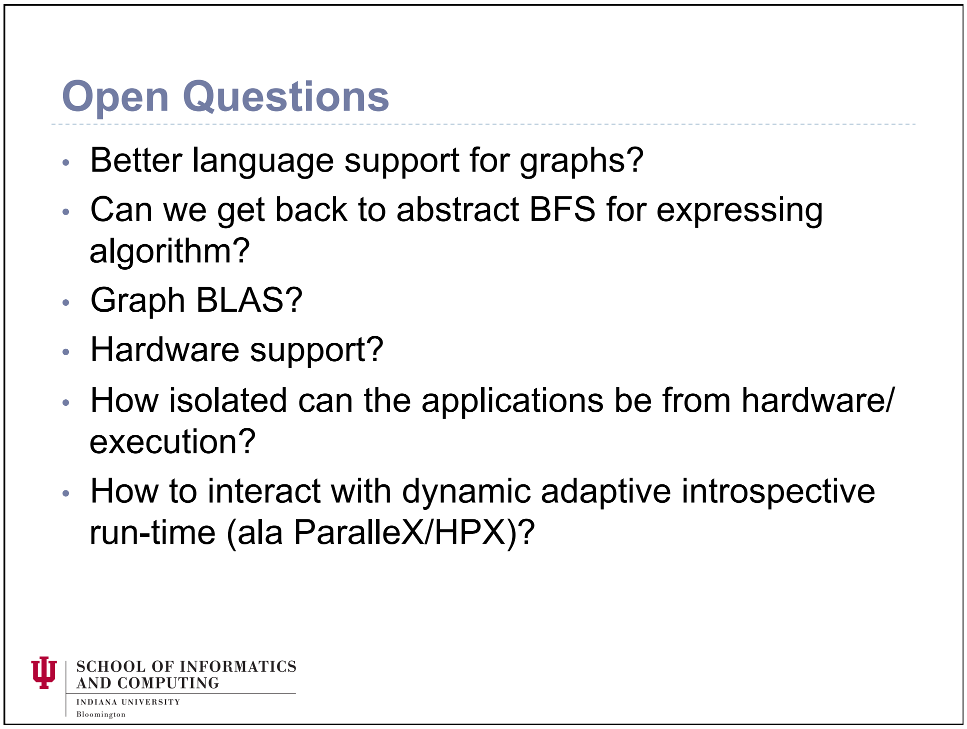
Delta-Stepping Shortest Paths. Graph500 graphs 2^{16} vertices/node. 2^{20} edges/node. Intrepid (BG/P).



Summary

- Active Messages / Active Pebbles
  - Express and enable fine-grained, asynchronous operations
  - Well-matched to data-driven problems
- Concise expression **and** efficient execution
  - Separate programming and execution models
  - Impedance match problem to hardware
  - Uniform view of parallelism

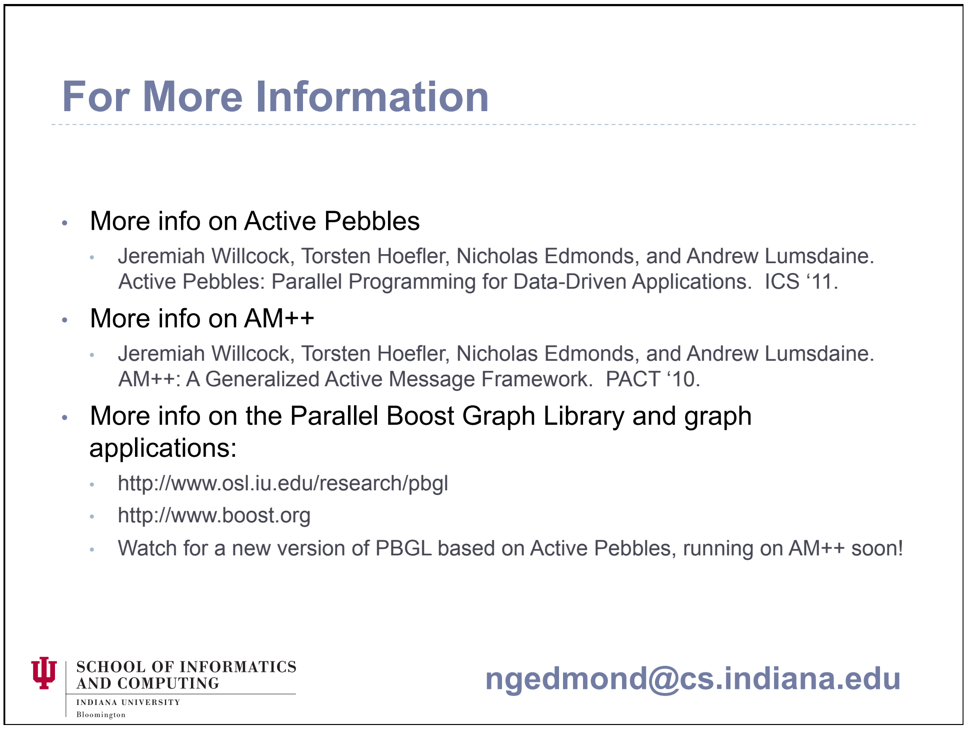
SCHOOL OF INFORMATICS AND COMPUTING
INDIANA UNIVERSITY
Bloomington



Open Questions

- Better language support for graphs?
- Can we get back to abstract BFS for expressing algorithm?
- Graph BLAS?
- Hardware support?
- How isolated can the applications be from hardware/ execution?
- How to interact with dynamic adaptive introspective run-time (ala ParalleX/HPX)?

SCHOOL OF INFORMATICS AND COMPUTING
INDIANA UNIVERSITY
Bloomington



For More Information

- More info on Active Pebbles
  - Jeremiah Willcock, Torsten Hoefler, Nicholas Edmonds, and Andrew Lumsdaine. Active Pebbles: Parallel Programming for Data-Driven Applications. ICS ‘11.
- More info on AM++
  - Jeremiah Willcock, Torsten Hoefler, Nicholas Edmonds, and Andrew Lumsdaine. AM++: A Generalized Active Message Framework. PACT ‘10.
- More info on the Parallel Boost Graph Library and graph applications:
  - http://www.osl.iu.edu/research/pbgl
  - http://www.boost.org
  - Watch for a new version of PBGL based on Active Pebbles, running on AM++ soon!

SCHOOL OF INFORMATICS
AND COMPUTING
INDIANA UNIVERSITY
Bloomington

ngedmond@cs.indiana.edu

Geospatial Analytics for Big Spatiotemporal Data

Ranga Raju Vatsavai and Budhendra Bhaduri

Geographic Information Science and Technology

Computational Sciences and Engineering Division

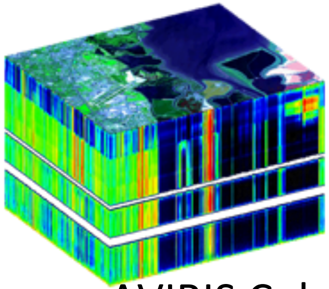
Oak Ridge National Laboratory

Big Data and Extreme-Scale Computing (BDEC)
April 30 – May 1, 2013.

Big Spatiotemporal Data

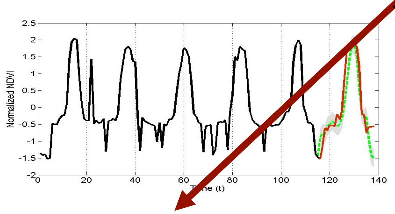
- What is Big Data?
  - V^4: Volume, Velocity, Variety, Veracity
- Many domains are becoming data driven
- Simulations
  - CMIP3 (AR4, 35TB, 2007), CMIP5 (~6PB, 2011)
- Observations
  - NASA EOSDIS (3PB, 2005), 5TB/day
- Social Media
  - 12TB of tweets/day

Remote Sensing: 1970-Present

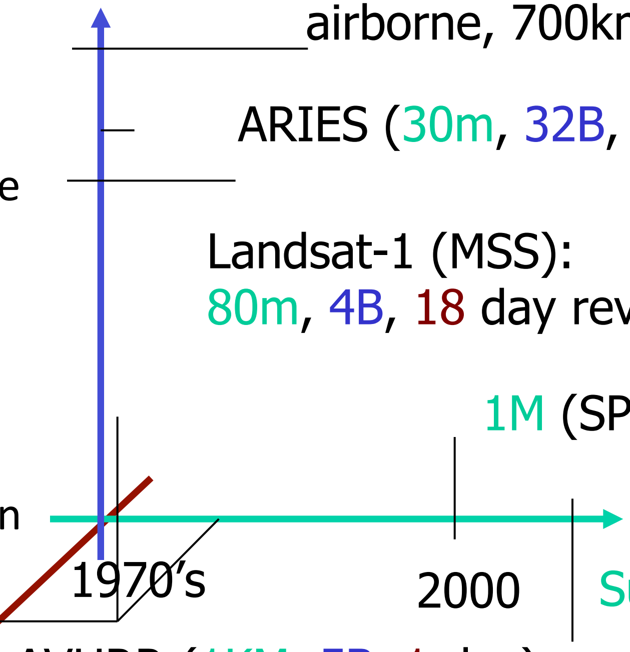


AVIRIS Cube

High-resolution Image



Temporal



Spectral

AVIRIS (20m, 224B): on demand, airborne, 700km/hr.

ARIES (30m, 32B, 7 day)

Landsat-1 (MSS):
80m, 4B, 18 day revisit

1M (SPOT, IKONOS, WorldView)

Spatial

1970's

2000

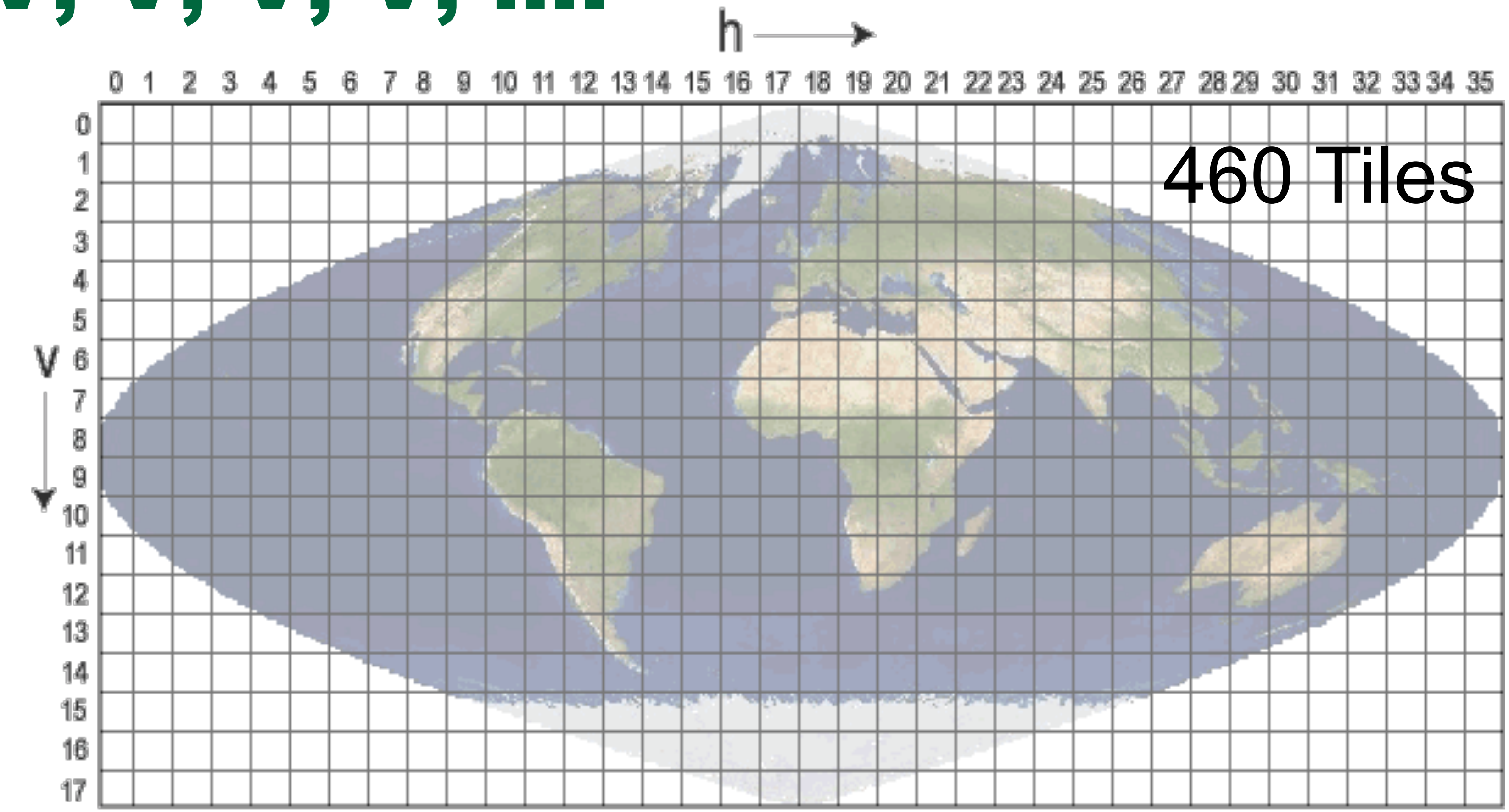
Sub-meter (Aerial, WV2…)

AVHRR (1KM, 5B, 1 day)

MODIS (250m-1KM, 36B, 1-2 days)

5TB/day – Heterogeneous data

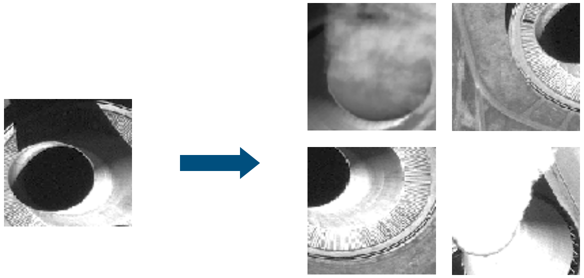
V, V, V, V, ...



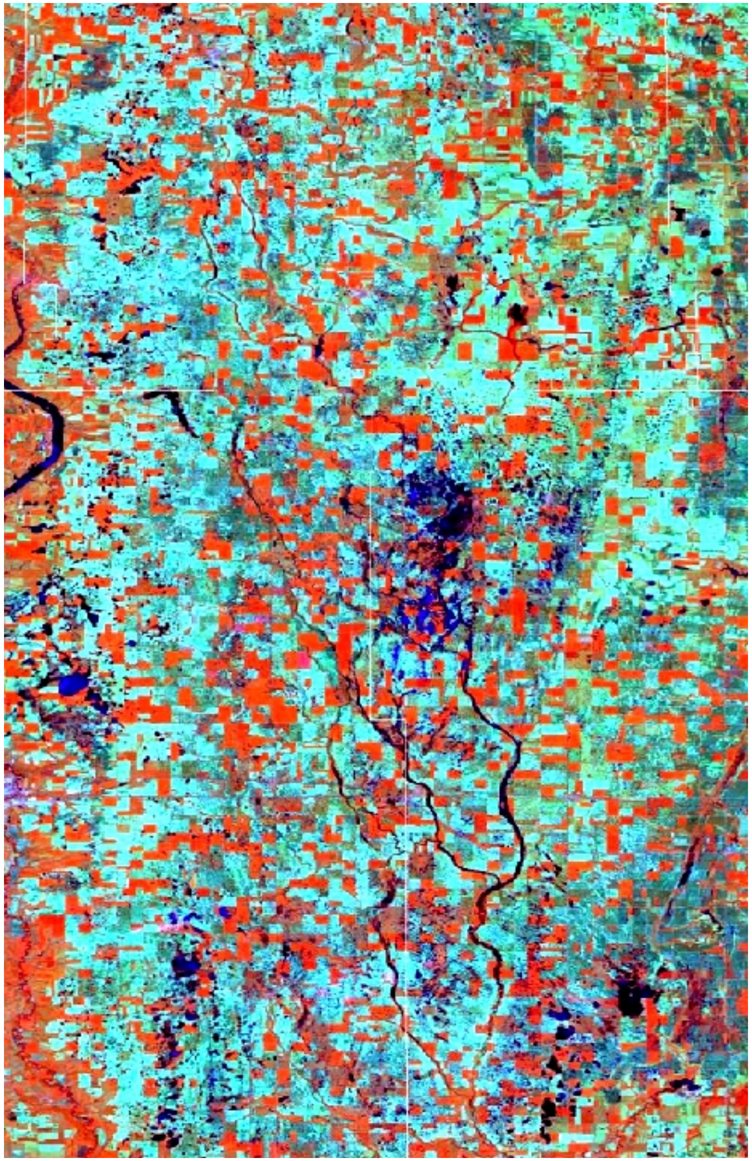
- Each Tile = 4800 x 4800 = 23,040,000 (250m)
- 16-bit, 1 Band = 44 MB
- (1m) => 1,440,000,000,000 = 1,373,291 MB
- Bands = 1 ~ 240; Derived Features ~ 250
- Temporal ~ 1 day to 22 days; 10’s of satellites

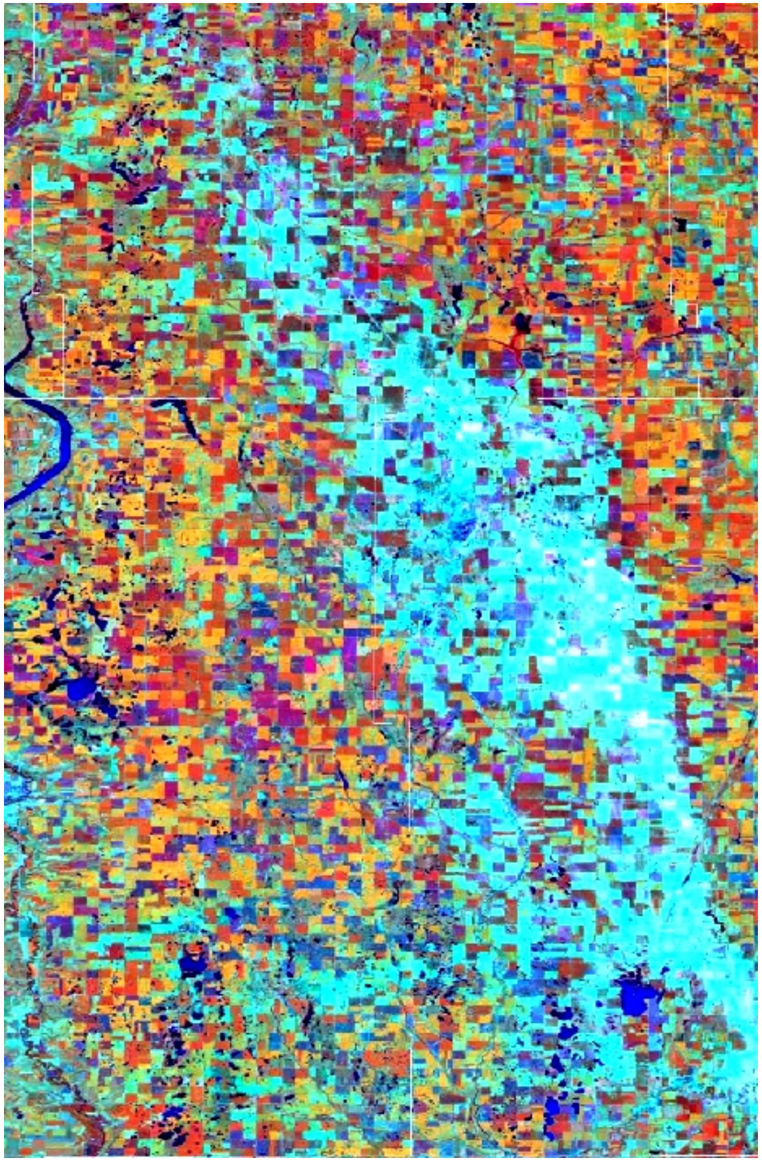
Searching for patterns

- Single Category Detection
  - Predict if a given visual category is present in a given image
- Content based image retrieval
  - Given query image, find similar images
- Structure Recognition
  - Structurally distinct objects within one class



Finding change patterns: Veg. damages





AWiFS (56 m, 4B, 5d)

- Moderate spatial, Moderate temporal
- Used for crop type and condition extraction
- Not good for changes at building level

Finding change patterns: infrastructure damages

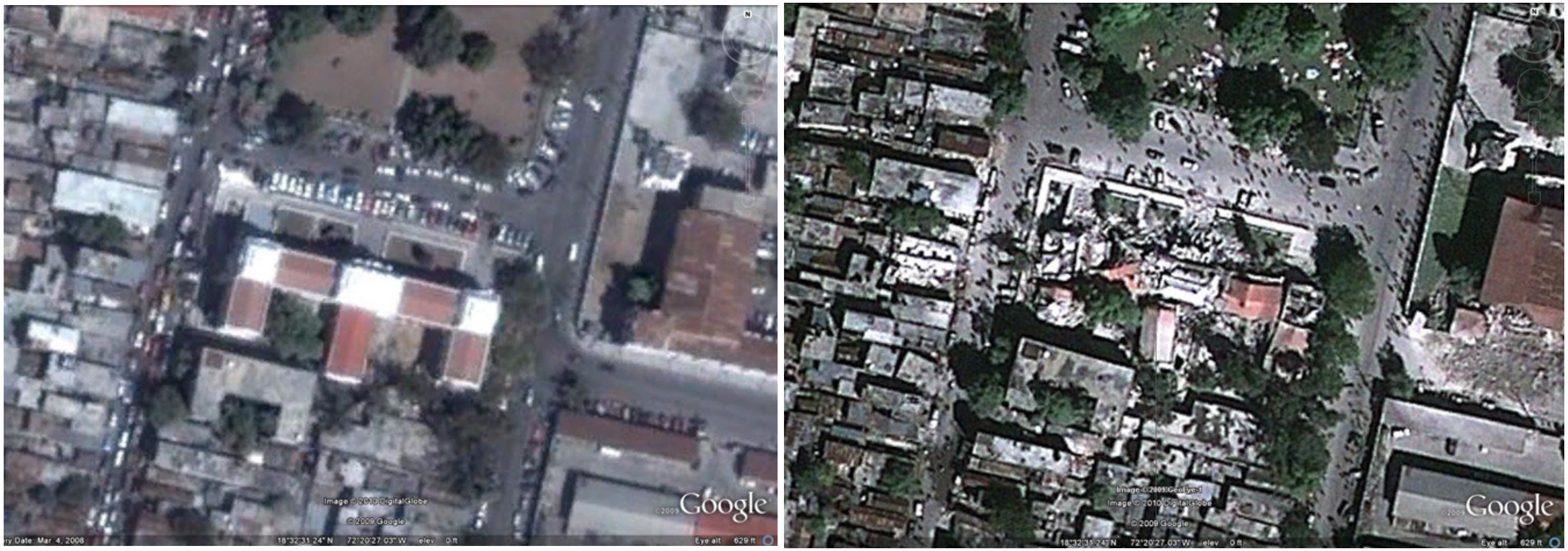
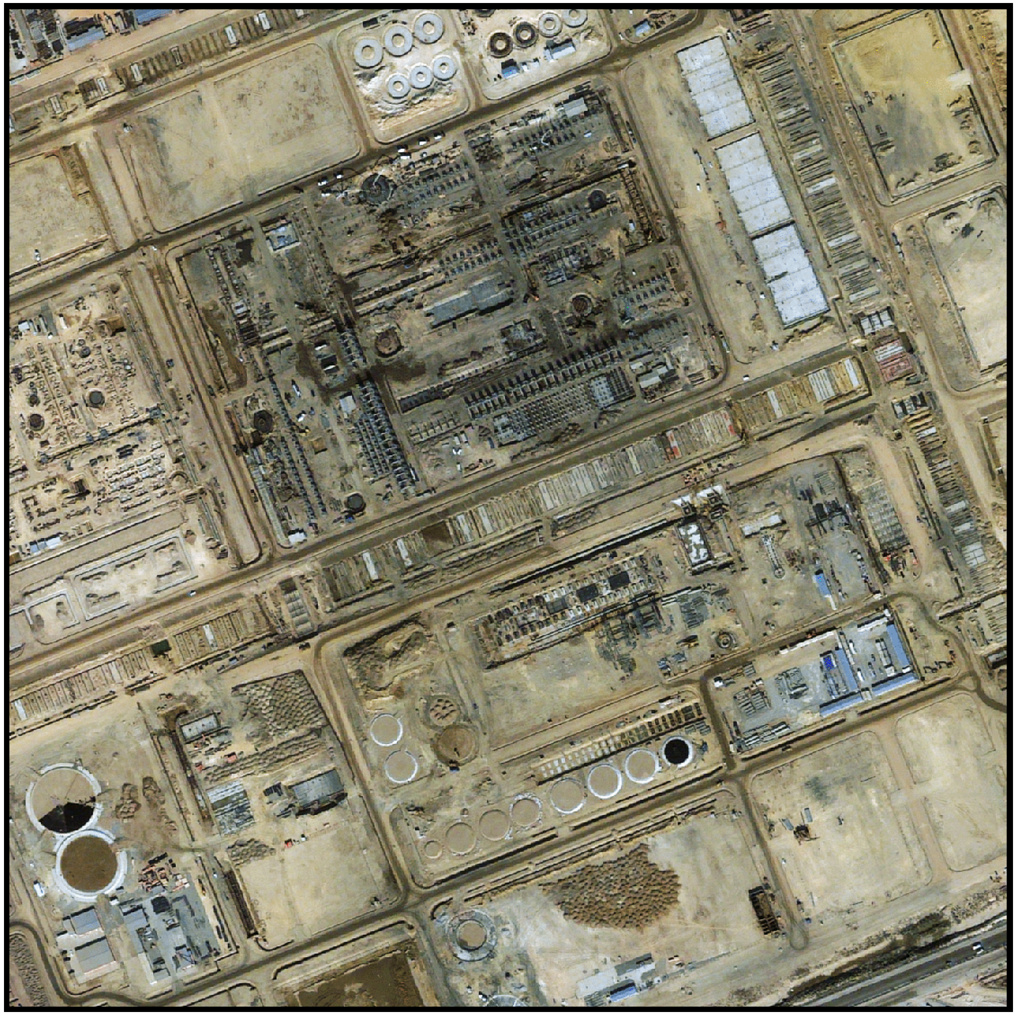


Image © 2010 DigitalGlobe
© 2009 Google
©2009 Google
ry Date: Mar 4, 2008
18°32'31.24" N 72°20'27.03" W elev 0 ft
Eye alt 629 ft
Image © 2009 GeoEye-1
Image © 2010 DigitalGlobe
© 2009 Google
©2009 Google
18°32'31.24" N 72°20'27.03" W elev 0 ft
Eye alt 629 ft

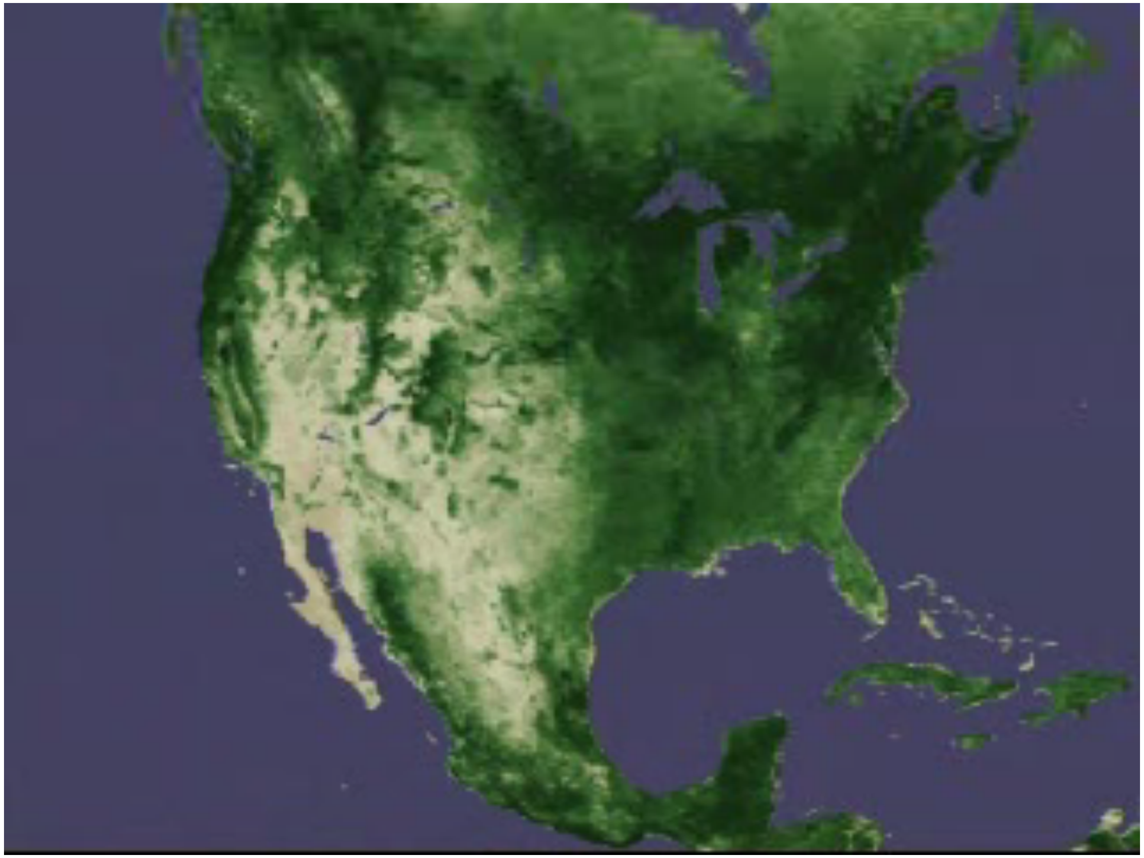
Haiti Earthquake Damages

Finding change patterns: new construction



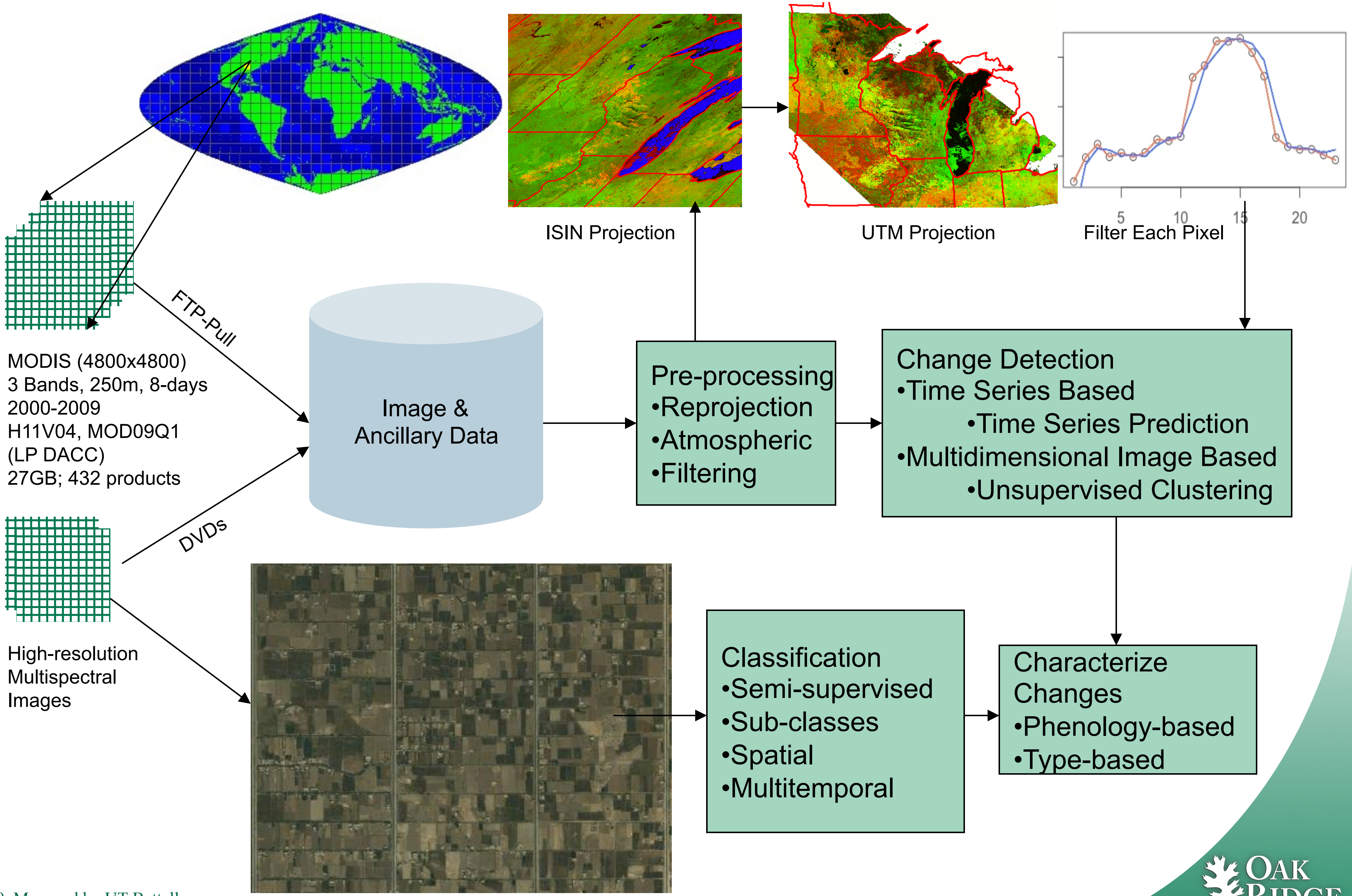
China – New Construction (QuickBird)

Understanding seasonal patterns



AVHRR NDVI 1KM (1981-2000)

Biomass monitoring framework



MODIS (4800x4800)
3 Bands, 250m, 8-days
2000-2009
H11V04, MOD09Q1
(LP DACC)
27GB; 432 products

High-resolution
Multispectral
Images

Image &
Ancillary Data

ISIN Projection

UTM Projection

Filter Each Pixel

Pre-processing
- Reprojection
- Atmospheric
- Filtering

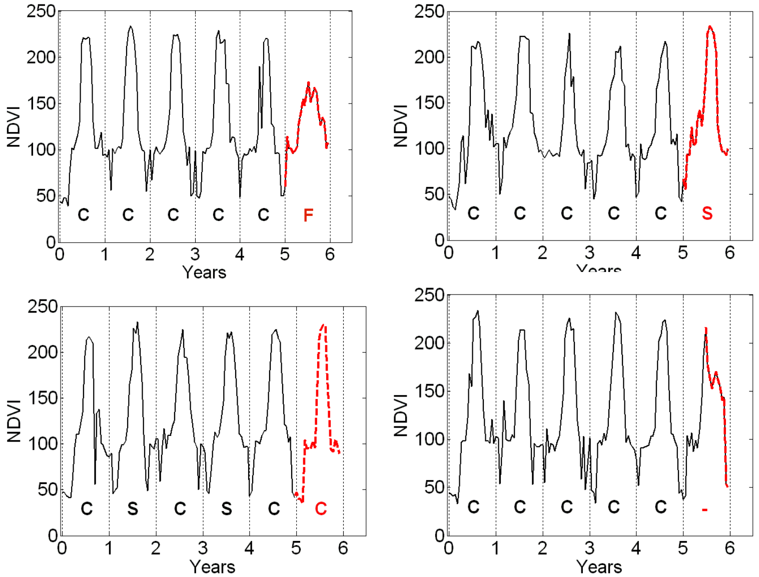
Change Detection
- Time Series Based
  - Time Series Prediction
- Multidimensional Image Based
  - Unsupervised Clustering

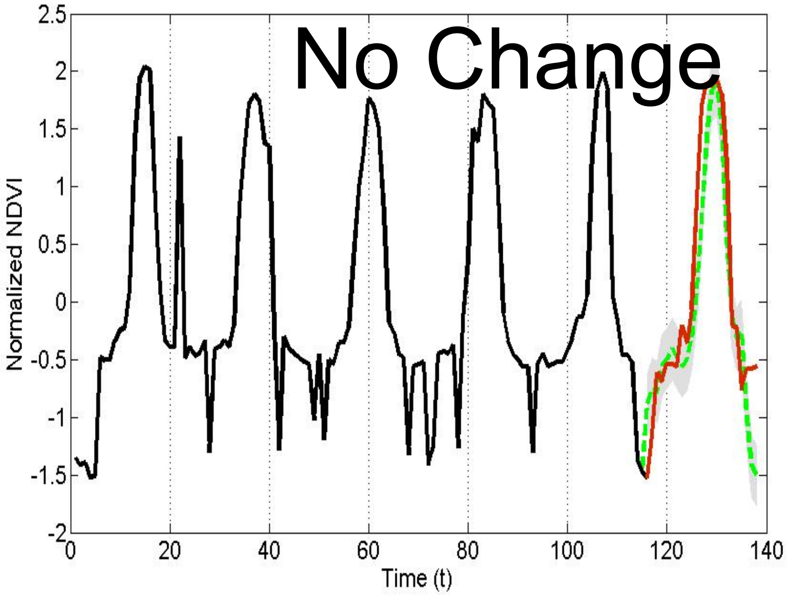
Classification
- Semi-supervised
- Sub-classes
- Spatial
- Multitemporal

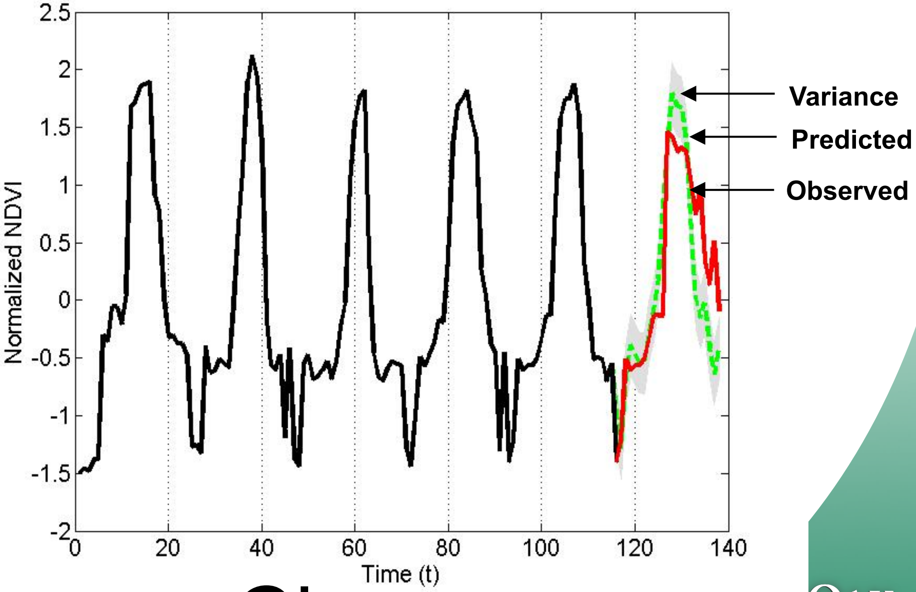
Characterize Changes
- Phenology-based
- Type-based

Change detection using Gaussian Process model

- **MODIS NDVI Time Series from Iowa**
  - **6 years (2001 – 2006)**
  - **23 observations per year**
- **Trained for first 5 years and monitored last year**
- ***Accuracy was 88% on a validation set consisting of 97 labeled time series with 13 true changes***

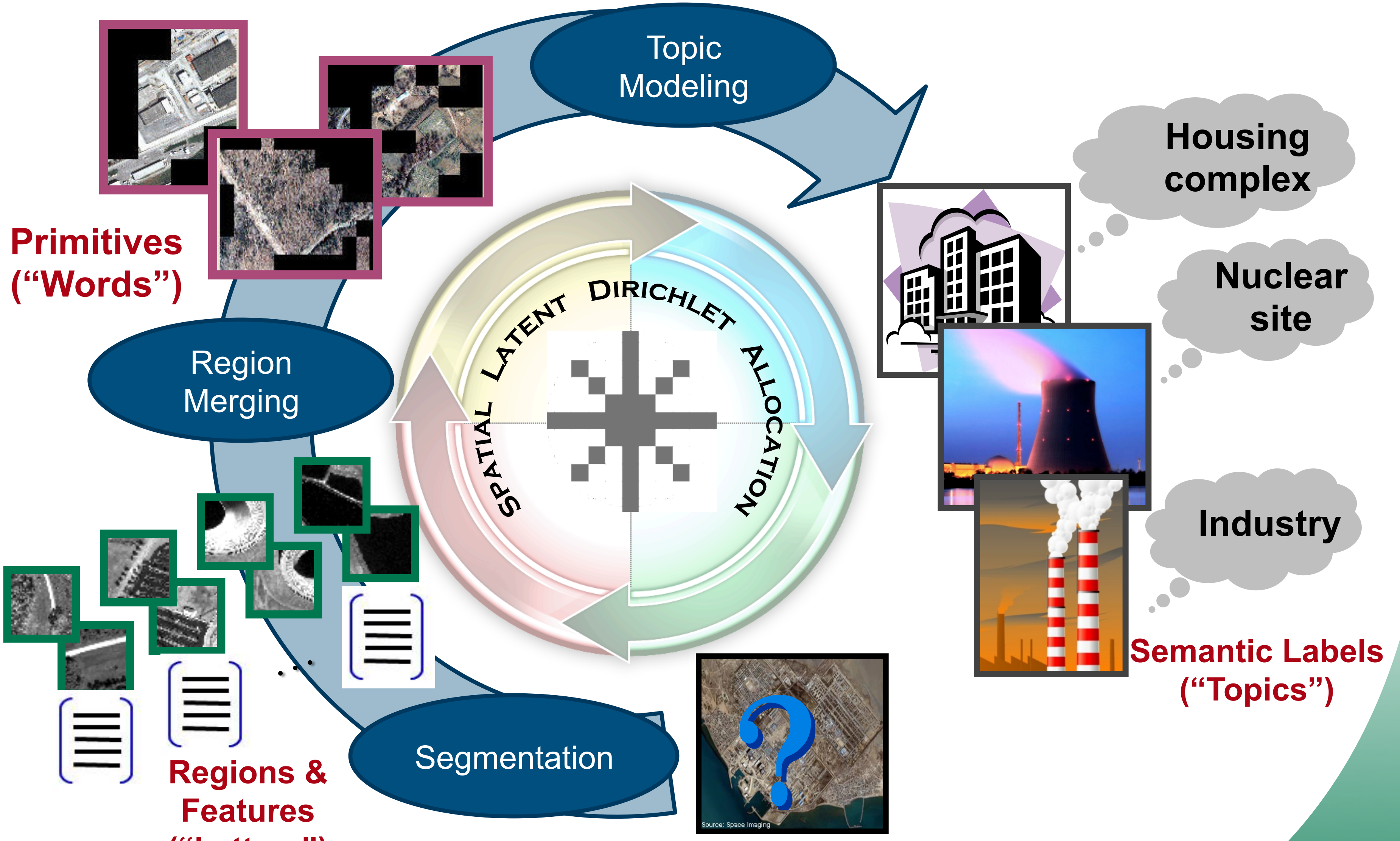






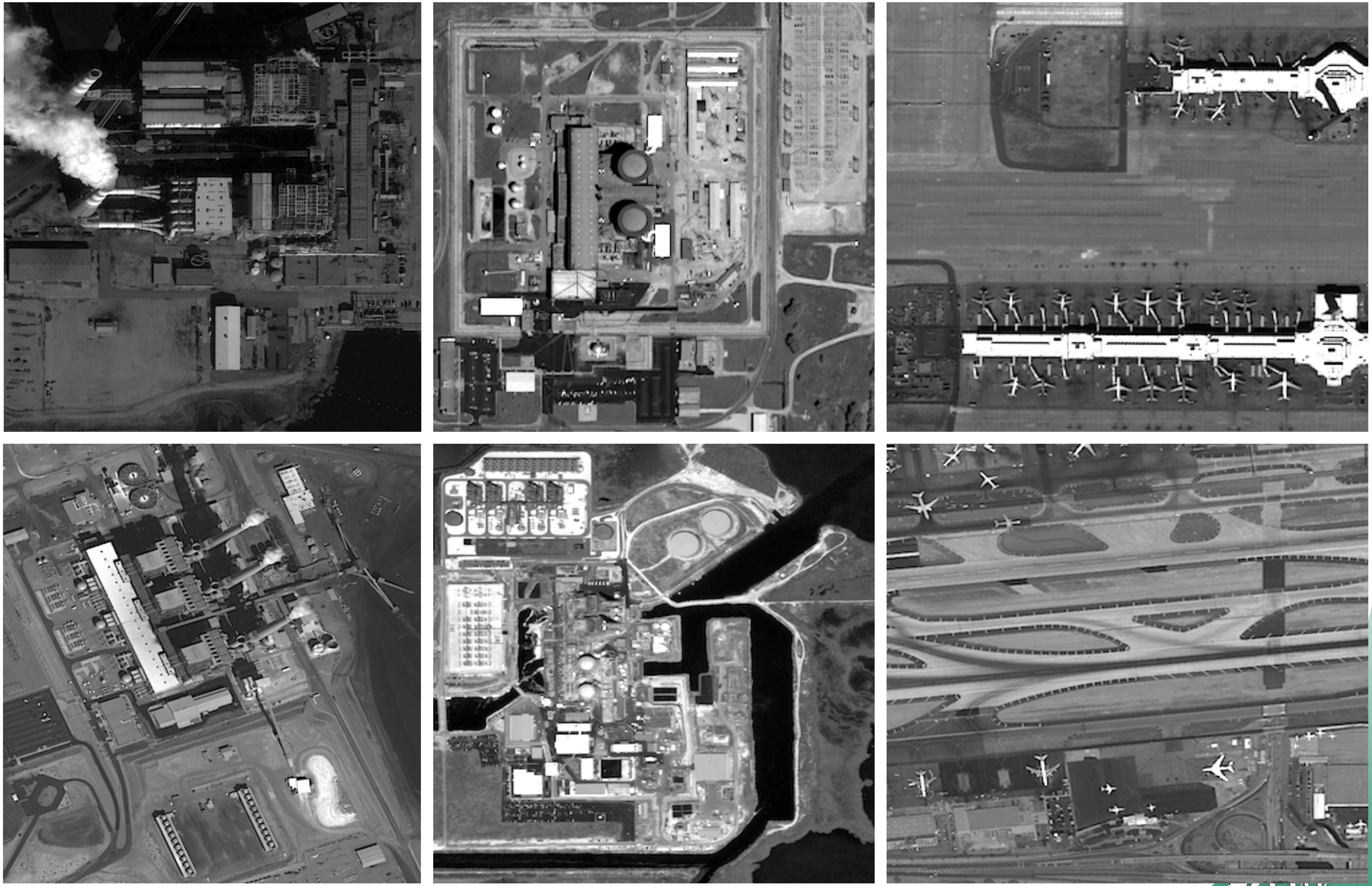
Change

Goal: Turn image pixels into semantic information for the analyst…

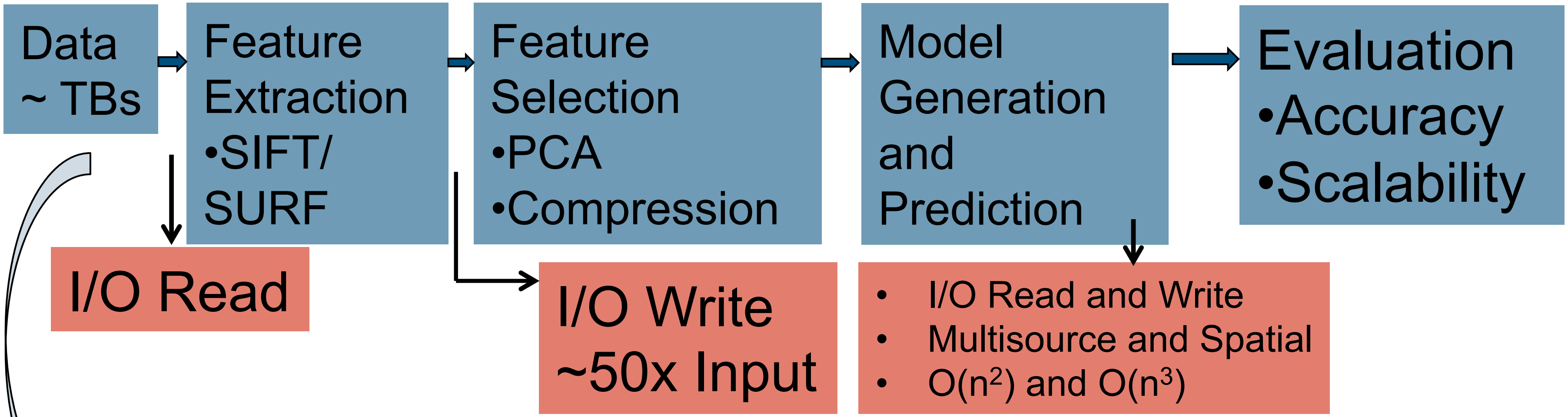


Unknown Images ("Documents")

Predict: Coal, Nuclear, Airports



Computational and I/O challenges



Data ~ TBs

Feature Extraction
- SIFT/ SURF

Feature Selection
- PCA
- Compression

Model Generation and Prediction

Evaluation
- Accuracy
- Scalability

I/O Read

I/O Write ~50x Input

- I/O Read and Write
- Multisource and Spatial
- $O(n^2)$ and $O(n^3)$

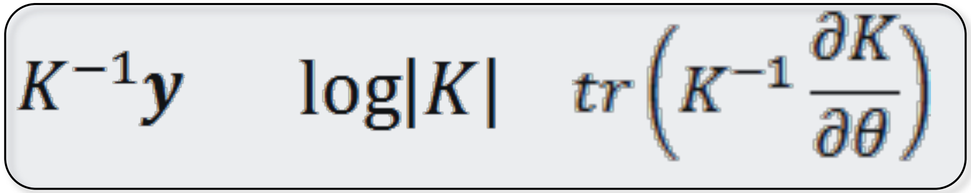
Source	Dataset Characteristics	Volume
Overhead Images	•Resolution: High (0.6 to 30 m) and moderate (56 m to 1 km)	0.5 PB (image size with features ranges from a GB to TB)
Terrestrial Images	•Small sized photographs: 12 million images (web scale: ~1 Billion images)	2 TB (images range from few KB to 0.5 MB)

Computational Primitives

- Gaussian Process Learning
  - Time-series based change detection
  - Spatial Classification/Prediction
  - GMM Clustering (X-Means, G-Means, GX-Means)

GP Change Detection – Computational Challenges

- Size of the covariance matrix grows quadratic with length of time series
- Need to compute



$$K^{-1}\mathbf{y} \qquad \log|K| \qquad tr\left(K^{-1}\frac{\partial K}{\partial \theta}\right)$$

- Standard methods are $O(t^3)$ and require $O(t^2)$ memory
- Not suitable for big time series
- Hyper-parameter estimation for p time series simultaneously is $O(p^*t^3)$

- AWiFS Satellite Data – Global spatial : 56m, Temporal: 5 days
- MODIS – 250m Temporal: 1 day
- Eddy Flux Sensors – Temporal: 15 minutes
- ECG Time Series – Temporal: ~ 0.2sec

Efficient Implementation by Exploiting Structure of Covariance Matrix

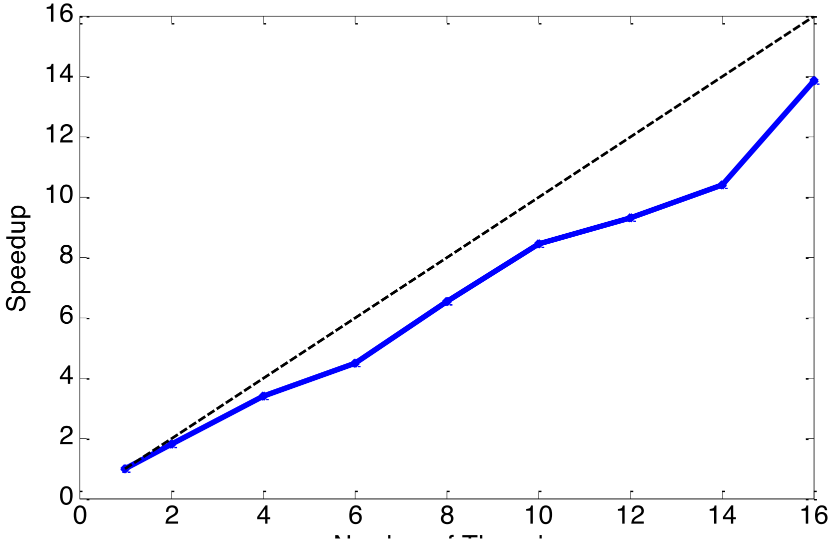
$$k(t_1,t_2) = \sigma_f^2 exp\left(-\frac{\Delta t}{2l^2}\right) exp\left(\frac{1-\cos\frac{2\pi\Delta t}{\omega}}{a}\right) + \sigma_n^2$$

- **Toeplitz**
- **Bi-symmetric**
- **Positive Definite**
- **Straightaway memory efficient (O(n))**
- **Inverse: O(n^2)**

k_0	k_1	k_2	k_3	k_4	k_5	k_6
k_1	k_0	k_1	k_2	k_3	k_4	k_5
k_2	k_1	k_0	k_1	k_2	k_3	k_4
k_3	k_2	k_1	k_0	k_1	k_2	k_3
k_4	k_3	k_2	k_1	k_0	k_1	k_2
k_5	k_4	k_3	k_2	k_1	k_0	k_1
k_6	k_5	k_4	k_3	k_2	k_1	k_0

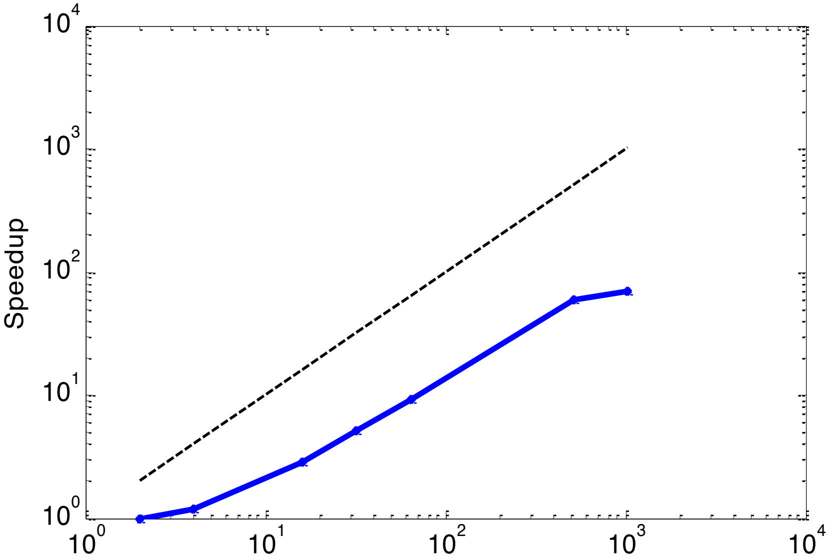
Parallelization Results

- Experiments done on FROST – A SGI Altix ICE 8200 cluster at ORNL
  - 128 compute nodes each having 16 virtual cores and 24GB of memory
- Task is to estimate hyper-parameters for 1 million NDVI time series



Number of Threads

Multi-threaded

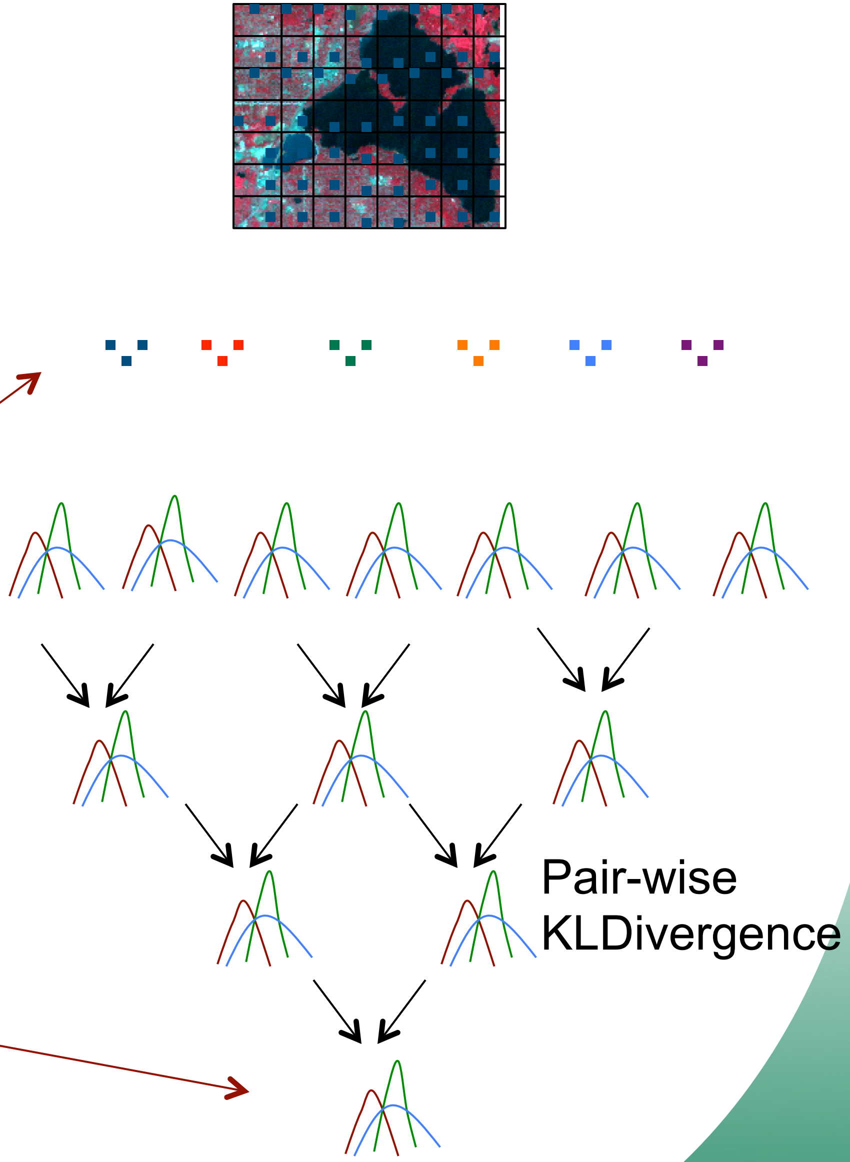


Number of Cores

MPI

GMM Clustering

- Expectation Maximization is a local optimization algorithm
    - Different initialization
- Multiple sampling
- Local model at each node
- Global model from local models



Pair-wise KLDivergence

Modeling Spatial Context

- i.i.d. assumptions are not valid
- MAP/MRF model
- SAR model $\mathbf{y} = \rho \mathbf{W} \mathbf{y} + \mathbf{x}\beta + \varepsilon$

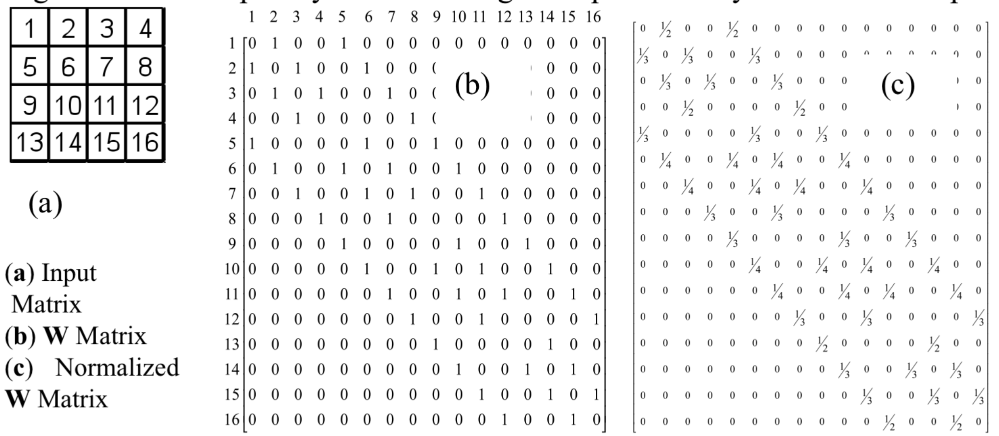
	X	
X	*S*	X
	X	

X	X	X
X	*S*	X
X	X	X

		X		
	X	X	X	
X	X	*S*	X	X
	X	X	X	
		X		

Challenge: Communication

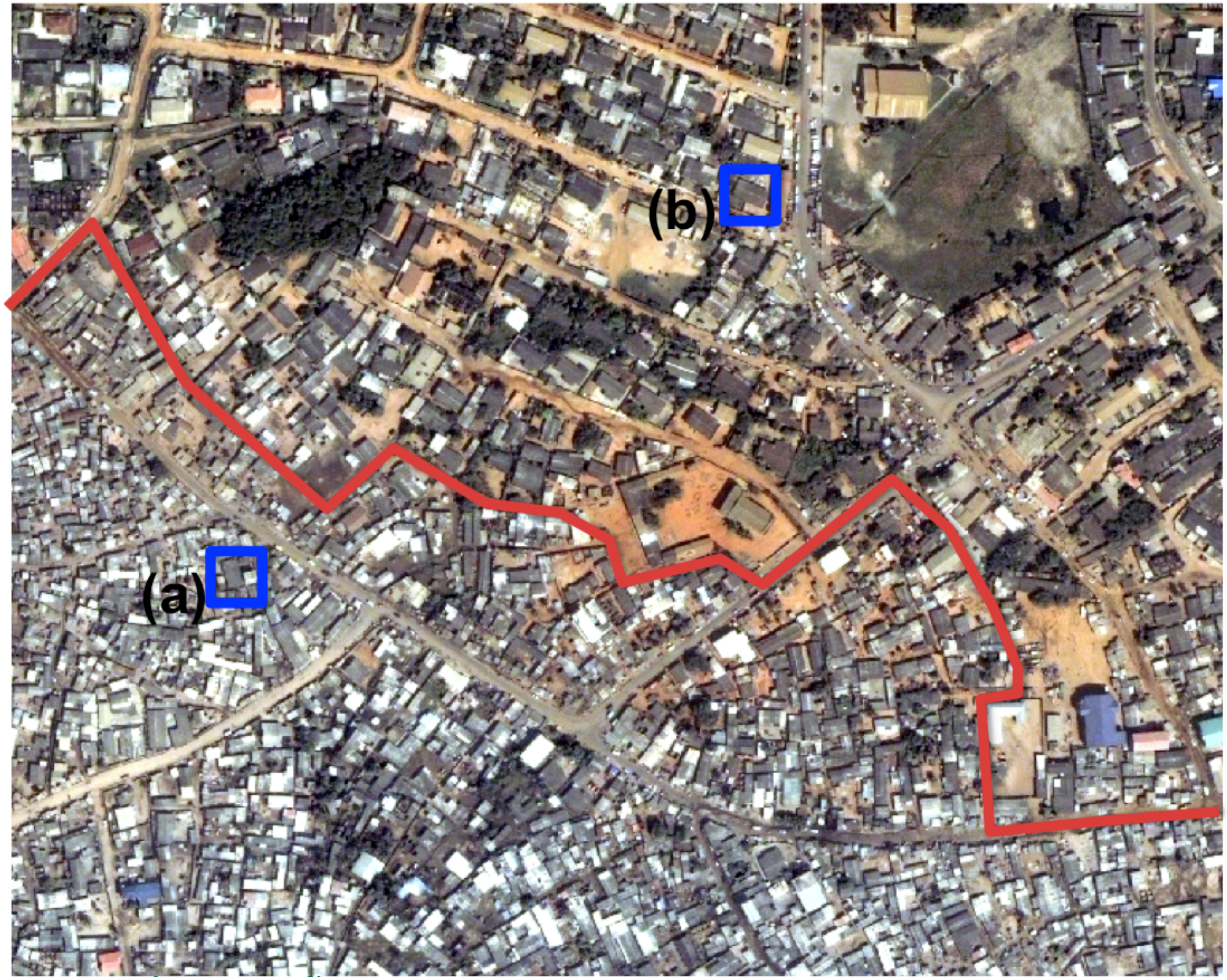
Figure 2. The complexity of W matrix grows quadratically in the size of input.



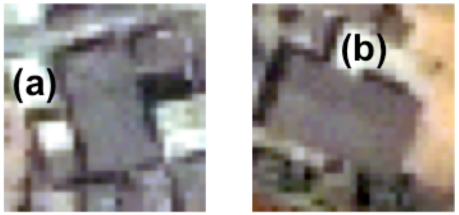
(a) Input Matrix
(b) **W** Matrix
(c) Normalized **W** Matrix

Complex Patterns

- Classes that cannot be separated by looking at pixels in isolation



Single-pixel (zoomed)



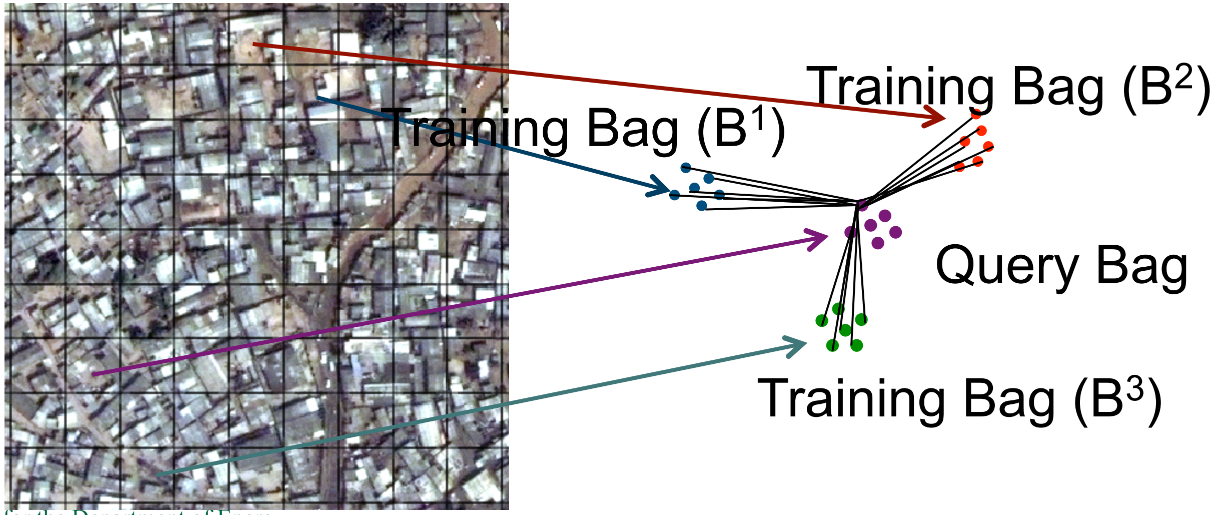
- Objects may be same (e.g., Buildings, Roads, …), but not the neighborhoods

Matching Segments

- Key
  - Define the distance between bags (min Hausdorff dist)

$$Dist(A,B) = \underset{\substack{1\le i\le n\\1\le j\le n}}{Min}\left(Dist(a_i,b_j)\right) = \underset{a\in A}{Min}\,\underset{b\in B}{Min}\|a-b\|$$

  - A, B: Bags; a_i, b_j: Instances from corresponding bags
- kNN: O(nd); Segment match: O(n^2Nd)

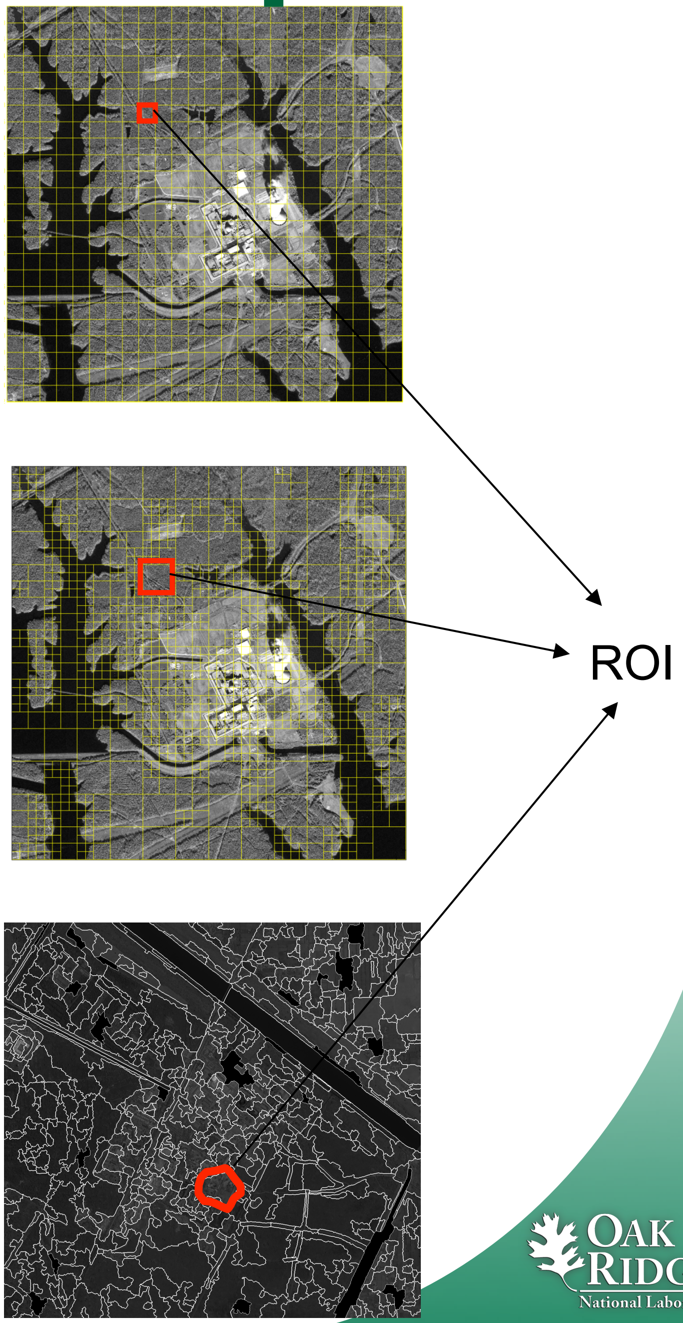


- Data
  - 1 Km^2, 1m pixel resolution, 3 bands
  - 1,000x1,000: 1M pixels
  - 10x10 block: 10K blocks
- Sequential Performance
  - 27.8 Hours
- Parallel (1-node; 16 threads)
  - 2.62 Hours

Feature Extraction Techniques

Low-Level Features

- Spectral/Intensity feature
- Local Binary Pattern (LBP)
- Local Edge Pattern (LEP)
- Edge Orientation
- SIFT
- Objective is to generate a feature vector representing the spectral and structural characteristics of the region-of-interest (ROI).
- ROI’s can be fixed size tile, variable size tile or irregular polygon.



What we have and what’s missing

- What we have?
  - Linear Algebra: ScaLAPACK, PLASMA, MAGMA, …
  - Parallel I/O: Parallel-NetCDF, ADIOS, …
  - Indexing: Bitmaps (FastBit), …
- What’s missing?
  - No similar libraries for spatial and spatiotemporal data mining, machine learning, and geospatial analytics

What's Needed?

- Community supported "mini-app" (Joel, Geoffrey, …)
- Library of core primitives tailored for heterogeneous architectures
  - Distance measures (e.g., Mahalanobis distance, KL Divergence, Bergman Divergence, Hausdorff distance, …)
  - Optimization (LP, IP, DP, …)
  - Search (*-first, branch-and-bound, iterative deepening, gradient descent, simulated annealing, nearest neighbor, …)
  - Pattern matching (linear/nonlinear temporal alignment, subsequence, dynamic time warping, …)
- Core data access/communication patterns

Conclusions

- Spatial and spatiotemporal applications
  - Big Data: Volume, Velocity, Variety, Veracity
  - Big Compute: O(n^3) and O(n^2)
- Diverse community
  - Remote Sensing and GIS
  - Climate Change
  - Medical Imagining
- Wish list
  - Community supported "mini-app"
  - Scalable library consisting of "core computational primitives"
  - Core set of data access/communication primitives

Acknowledgements

- DOE/NNSA/NA22: Simulations, Modeling, and Algorithms Program
- ORNL LDRD Program
- DOE/SDAV
- V. Chandola, A. Cheriyadat, S. Gleason, J. Grasser (ORNL); S. Shekhar (UMN), J. Ghosh (UT-Austin)
- Prepared by Oak Ridge National Laboratory, P.O. Box 2008, Oak Ridge, Tennessee 37831-6285, managed by UT-Battelle, LLC for the U. S. Department of Energy under contract no. DEAC05-00OR22725.

Questions

?