Data Intensive Applications on Clouds

The Second International Workshop on

Data Intensive Computing in the Clouds (DataCloud-SC11)

at SC11

November 14 2011

Geoffrey Fox

gcf@indiana.edu

http://www.infomall.org http://www.salsahpc.org

Director, Digital Science Center, Pervasive Technology Institute

Associate Dean for Research and Graduate Studies, School of Informatics and Computing

Indiana University Bloomington

Work with Judy Qiu and several students

https://portal.futuregrid.org

Some Trends

- The **Data Deluge** is clear trend from Commercial (Amazon, transactions) , Community (Facebook, Search) and Scientific applications
- **Exascale** initiatives will continue drive to high end with a simulation orientation
- **Clouds** offer from different points of view
  - **NIST:** On-demand service (elastic); Broad network access; Resource pooling; Flexible resource allocation; Measured service
  - **Economies of scale**
  - Powerful new **software models**

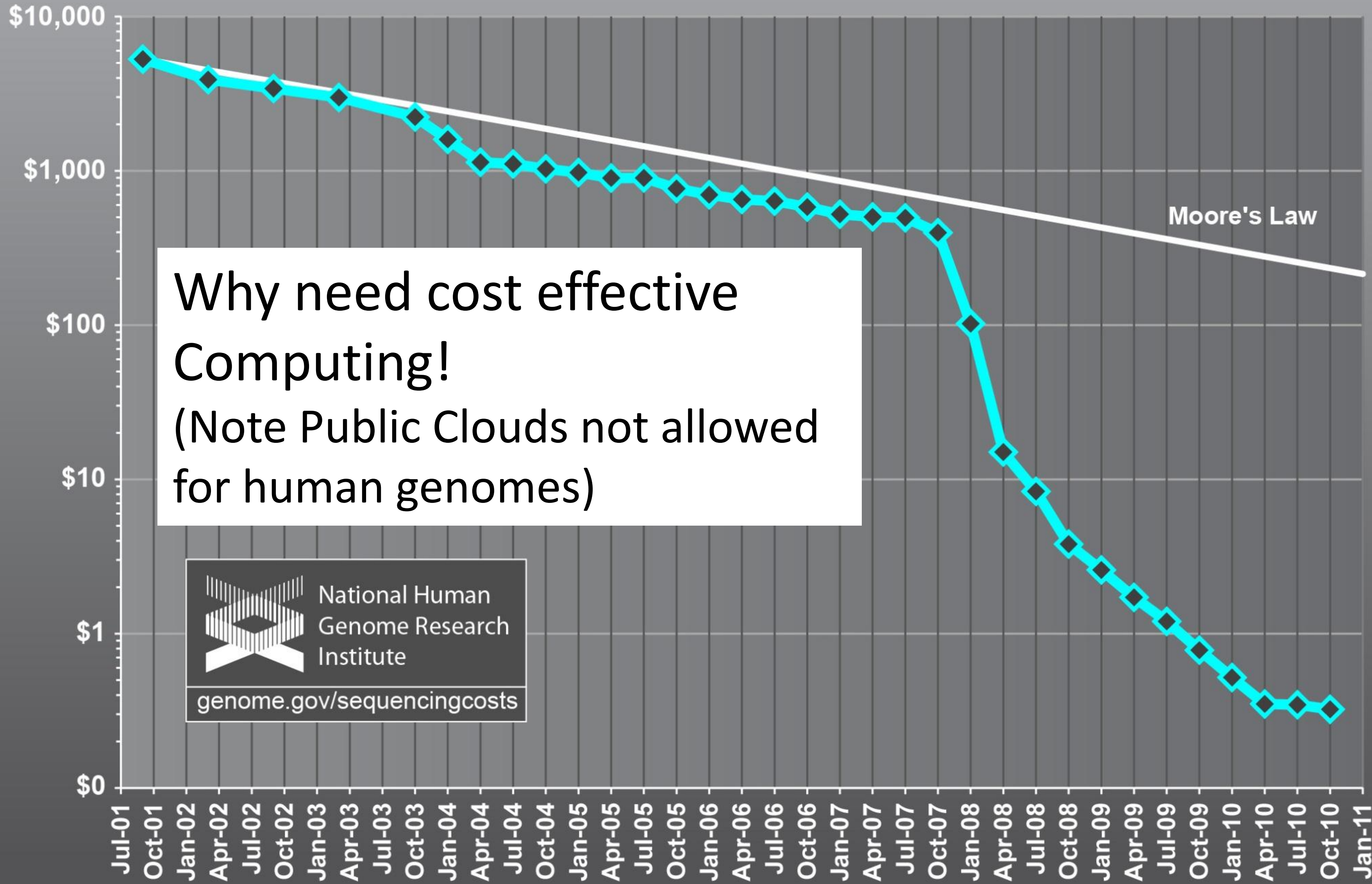
Some Data sizes

- ~40 10^9 Web pages at ~300 kilobytes each = 10 Petabytes
- Youtube 48 hours video uploaded per minute;
  - in 2 months in 2010, uploaded more than total NBC ABC CBS
  - ~2.5 petabytes per year uploaded?
- LHC 15 petabytes per year
- Radiology 69 petabytes per year
- Square Kilometer Array Telescope will be 100 terabits/second
- Earth Observation becoming ~4 petabytes per year
- Earthquake Science – few terabytes **total** today
- PolarGrid – 100's terabytes/year
- Exascale simulation data dumps – terabytes/second
- Not very quantitative

Genomics in Personal Health

- Suppose you measured everybody's genome every 2 years
- 30 petabits of new gene data per day
  - factor of 100 more for raw reads with coverage
- Data surely distributed
- 1.5*10^8 to 1.5*10^10 continuously running present day cores to perform a simple Blast analysis on this data
  - Amount depends on clever hashing and maybe Blast not good enough as field gets more sophisticated
- Analysis requirements not well articulated in many fields – See http://www.delsall.org for life sciences
  - LHC data analysis well understood – is it typical?
  - LHC Pleasing parallel (PP) – some in Life Sciences like Blast also PP

Cost per Megabase of DNA Sequence



Why need cost effective Computing!

(Note Public Clouds not allowed for human genomes)

Clouds and Grids/HPC

- Synchronization/communication Performance
  Grids > Clouds > HPC Systems
- Clouds appear to execute effectively Grid workloads but are not easily used for closely coupled HPC applications
- **Service Oriented Architectures** and **workflow** appear to work similarly in both grids and clouds
- Assume for immediate future, science supported by a mixture of
  - Clouds – data analysis (and pleasingly parallel)
  - Grids/High Throughput Systems (moving to clouds as convenient)
  - Supercomputers ("MPI Engines") going to exascale

Clouds and Jobs

- **Clouds** are a major industry thrust with a growing fraction of IT expenditure that IDC estimates will grow to **$44.2 billion direct investment in 2013** while 15% of IT investment in 2011 will be related to cloud systems with a 30% growth in public sector.
- Gartner also rates cloud computing high on list of critical emerging technologies with for example "Cloud Computing" and "Cloud Web Platforms" rated as transformational (their highest rating for impact) in the next 2-5 years.
- Correspondingly there is and will continue to be major opportunities for new jobs in cloud computing with a recent European study estimating there will be **2.4 million new cloud computing jobs in Europe alone by 2015**.
- Cloud computing spans research and economy and so attractive component of **curriculum** for students that mix "going on to PhD" or "graduating and working in industry" (as at Indiana University where most CS Masters students go to industry)

2 Aspects of Cloud Computing: Infrastructure and Runtimes

- Cloud infrastructure: outsourcing of servers, computing, data, file space, utility computing, etc..
- Cloud runtimes or Platform: tools to do data-parallel (and other) computations. Valid on Clouds and traditional clusters
  - Apache Hadoop, Google **MapReduce**, Microsoft Dryad, Bigtable, Chubby and others
  - MapReduce designed for information retrieval but is excellent for a wide range of science data analysis applications
  - Can also do much traditional parallel computing for data-mining if extended to support iterative operations
  - **Data Parallel File system** as in HDFS and Bigtable

Guiding Principles

- Clouds may not be suitable for everything but they are suitable for majority of data intensive applications
  - Solving partial differential equations on 100,000 cores probably needs classic MPI engines
- Cost effectiveness, elasticity and quality programming model will drive use of clouds in many areas such as genomics
- Need to solve issues of
  - Security-privacy-trust for sensitive data
  - How to store data – "data parallel file systems" (HDFS), Object Stores, or classic HPC approach with shared file systems with Lustre etc.
- Programming model which is likely to be **MapReduce** based
  - Look at high level languages
  - Compare with databases (SciDB?)
  - **Must support iteration to do "real parallel computing"**
  - **Need Cloud-HPC Cluster Interoperability**

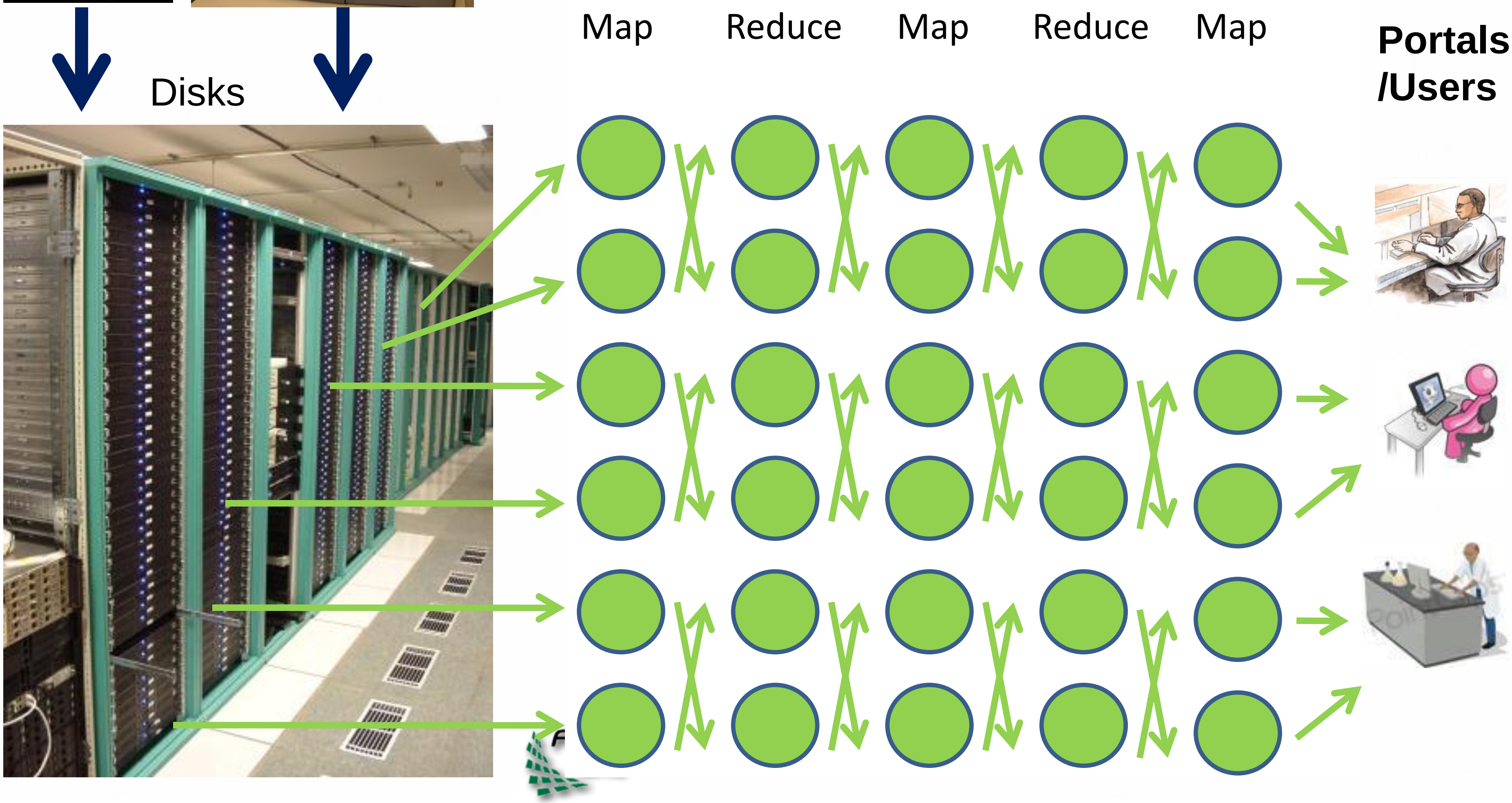
MapReduce “File/Data Repository” Parallelism

Map = (data parallel) computation reading and writing data

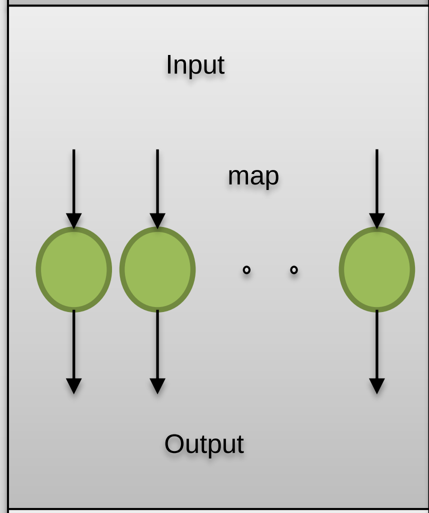
Reduce = Collective/Consolidation phase e.g. forming multiple global sums as in histogram

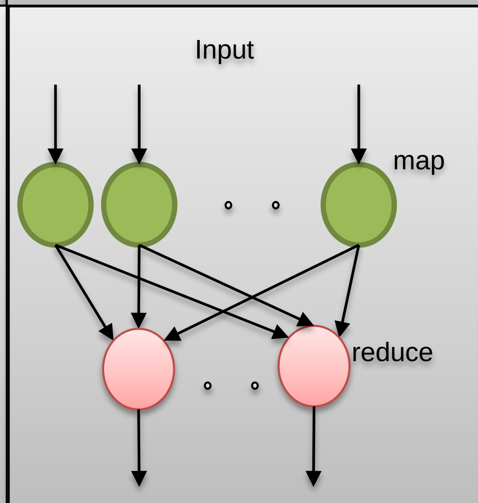
Instruments

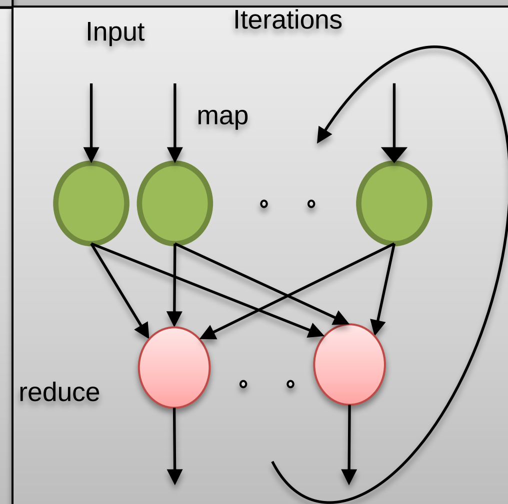
MPI or Iterative MapReduce

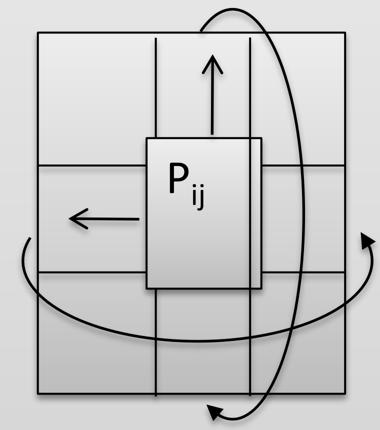


Application Classification









(a) Map Only	**(b) Classic MapReduce**	**(c) Iterative MapReduce**	**(d) Loosely Synchronous**
BLAST Analysis Smith-Waterman Distances Parametric sweeps PolarGrid Matlab data analysis	High Energy Physics (HEP) Histograms Distributed search Distributed sorting Information retrieval	Expectation maximization clustering e.g. Kmeans Linear Algebra Multimensional Scaling Page Rank	Many MPI scientific applications such as solving differential equations and particle dynamics
← **Domain of MapReduce and Iterative Extensions** →			**MPI**

Twister v0.9

March 15, 2011

New Interfaces for Iterative MapReduce Programming

http://www.iterativemapreduce.org/

SALSA Group

Bingjing Zhang, Yang Ruan, Tak-Lon Wu, Judy Qiu, Adam Hughes, Geoffrey Fox, **Applying Twister to Scientific Applications**, Proceedings of IEEE CloudCom 2010 Conference, Indianapolis, November 30-December 3, 2010

Twister4Azure released May 2011

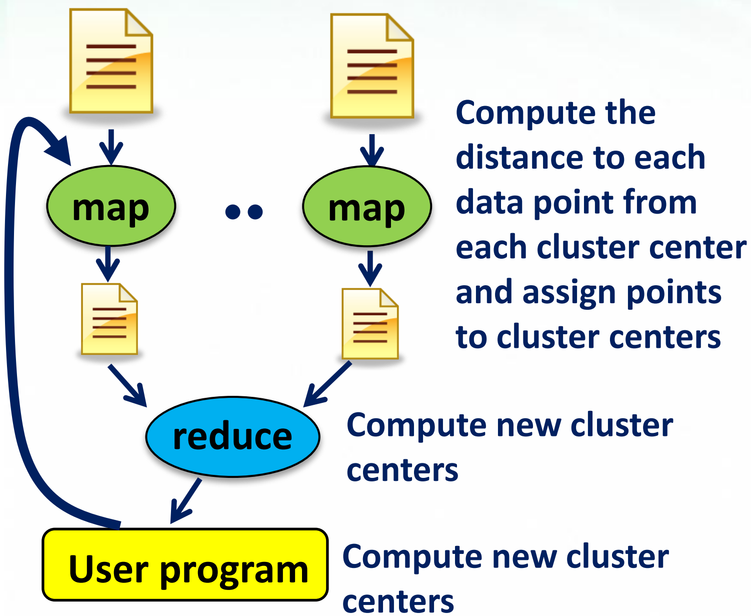
http://salsahpc.indiana.edu/twister4azure/

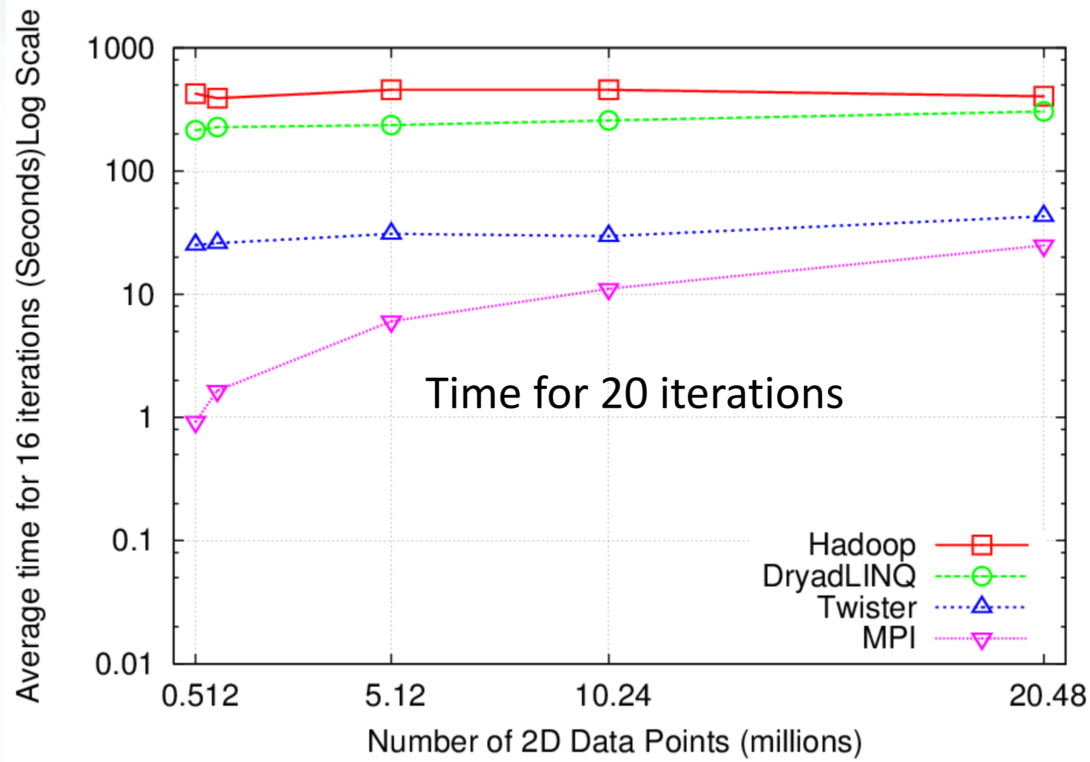
MapReduceRoles4Azure available for some time at

http://salsahpc.indiana.edu/mapreduceroles4azure/

Microsoft Daytona project July 2011 is Azure version

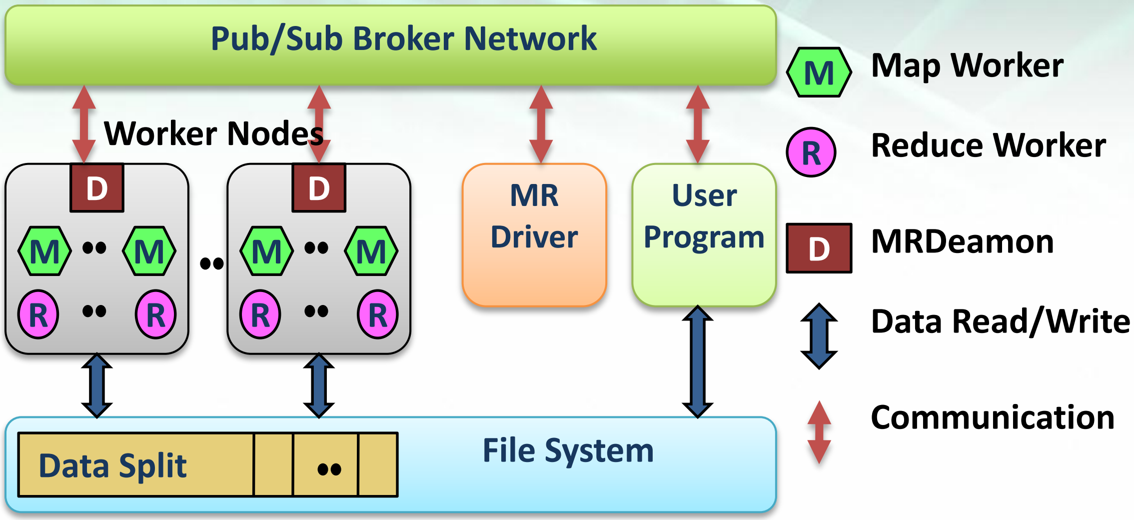
K-Means Clustering



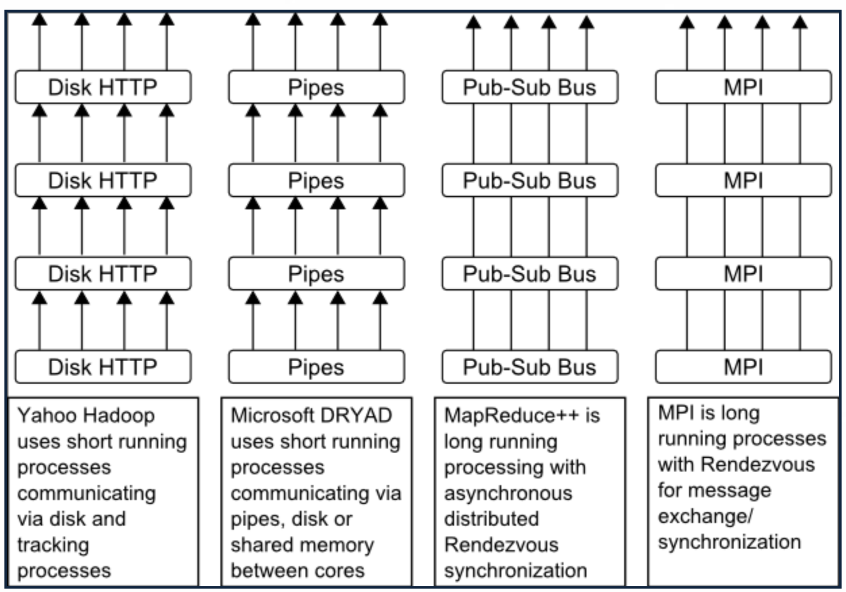


- Iteratively refining operation
- Typical MapReduce runtimes incur extremely high overheads
  - New maps/reducers/vertices in every iteration
  - File system based communication
- Long running tasks and faster communication in Twister enables it to perform close to MPI

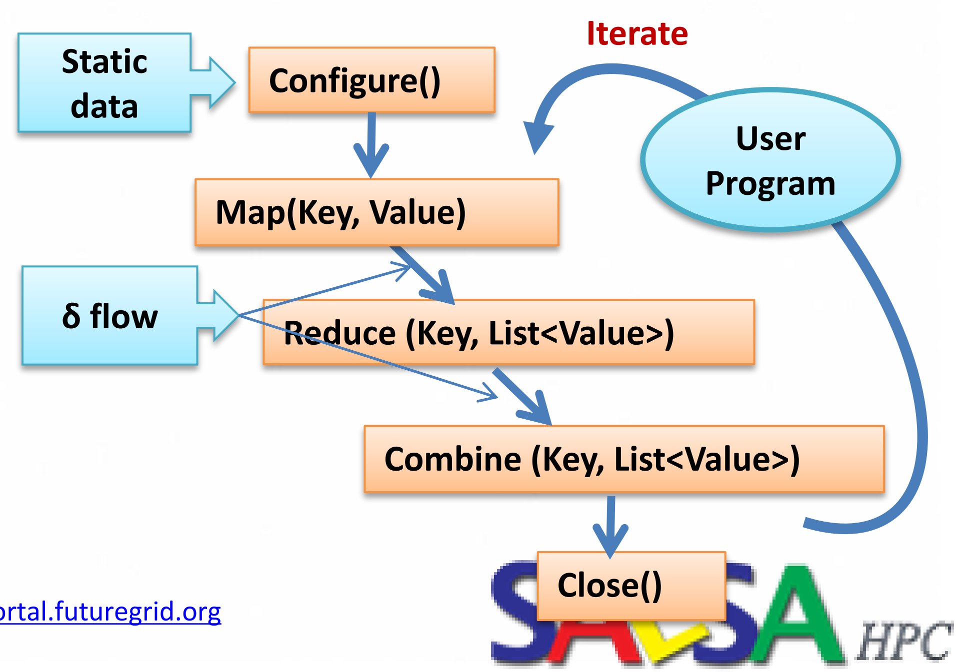
Twister



- Streaming based communication
- Intermediate results are directly transferred from the map tasks to the reduce tasks – **eliminates local files**
- Cacheable map/reduce tasks
  - Static data remains in memory
- Combine phase to combine reductions
- User Program is the **composer** of MapReduce computations
- **Extends the MapReduce model to iterative computations**

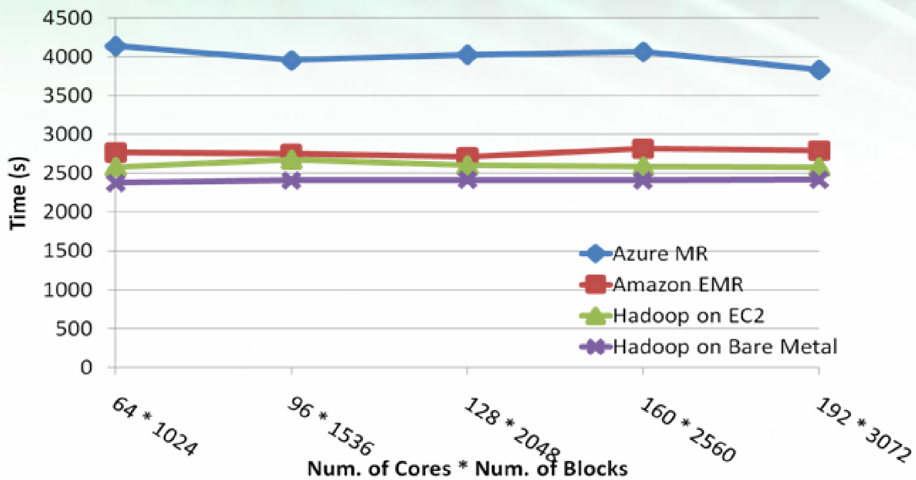


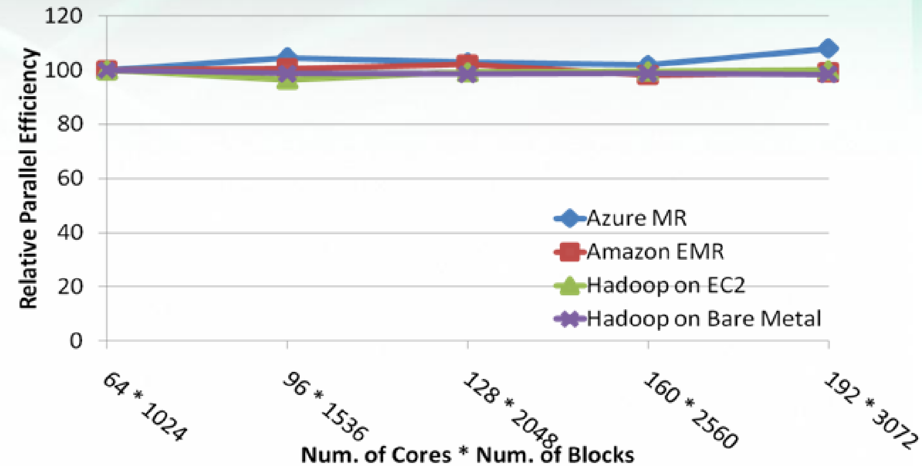
Different synchronization and intercommunication mechanisms used by the parallel runtimes

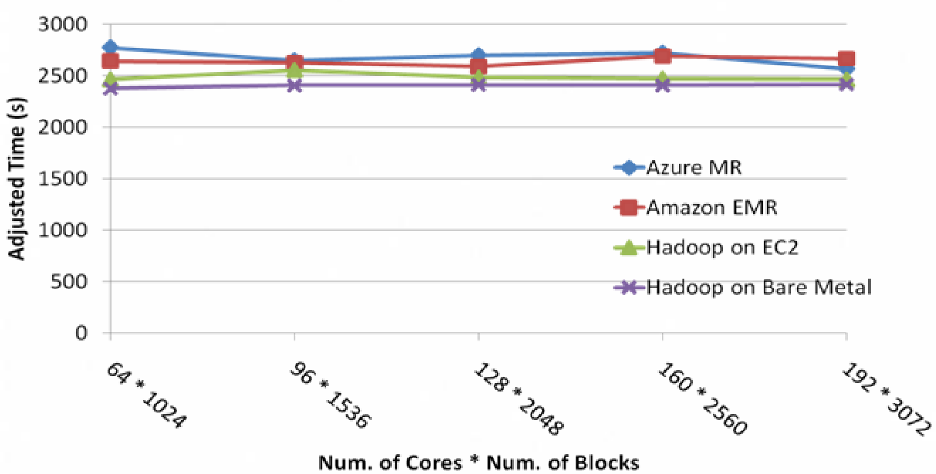


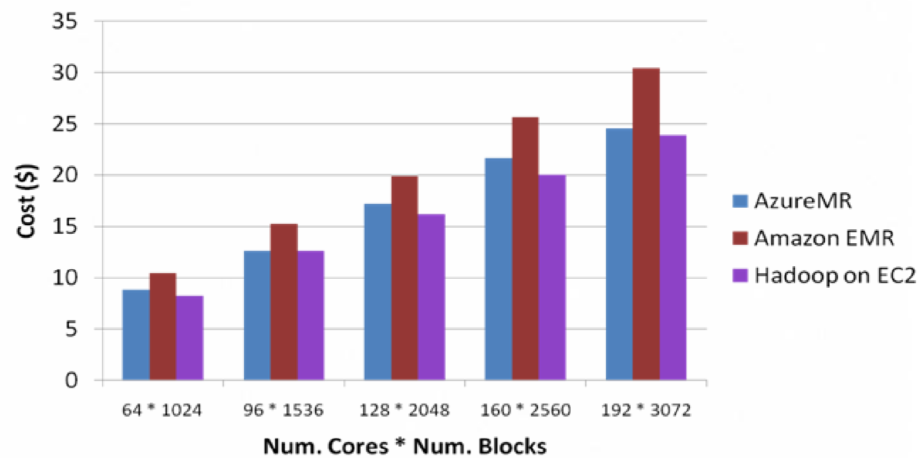
https://portal.futuregrid.org

SWG Sequence Alignment Performance









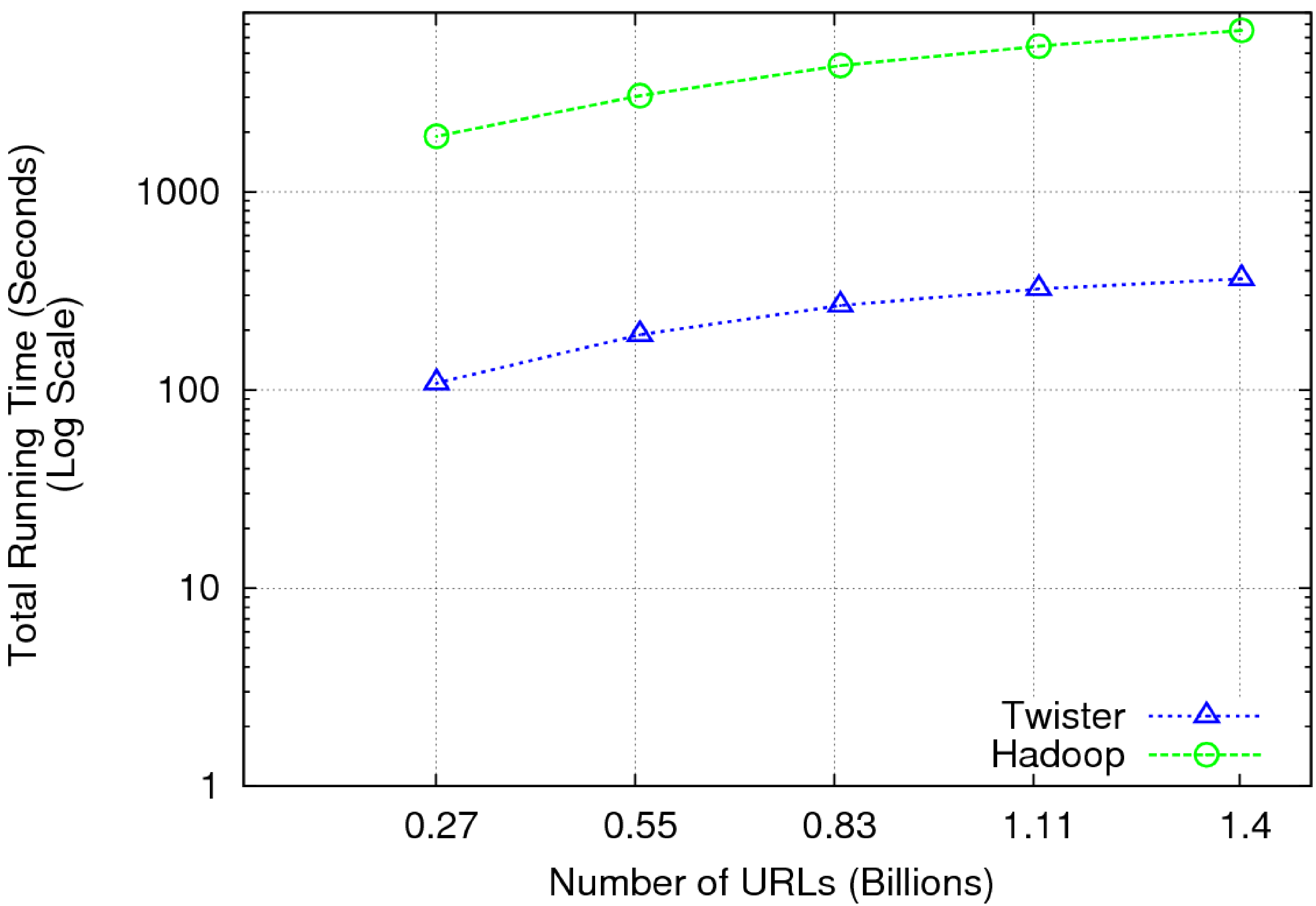
Smith-Waterman-GOTOH to calculate all-pairs dissimilarity

Future Grid https://portal.futuregrid.org

SALSA HPC

Performance of Pagerank using ClueWeb Data (Time for 20 iterations)

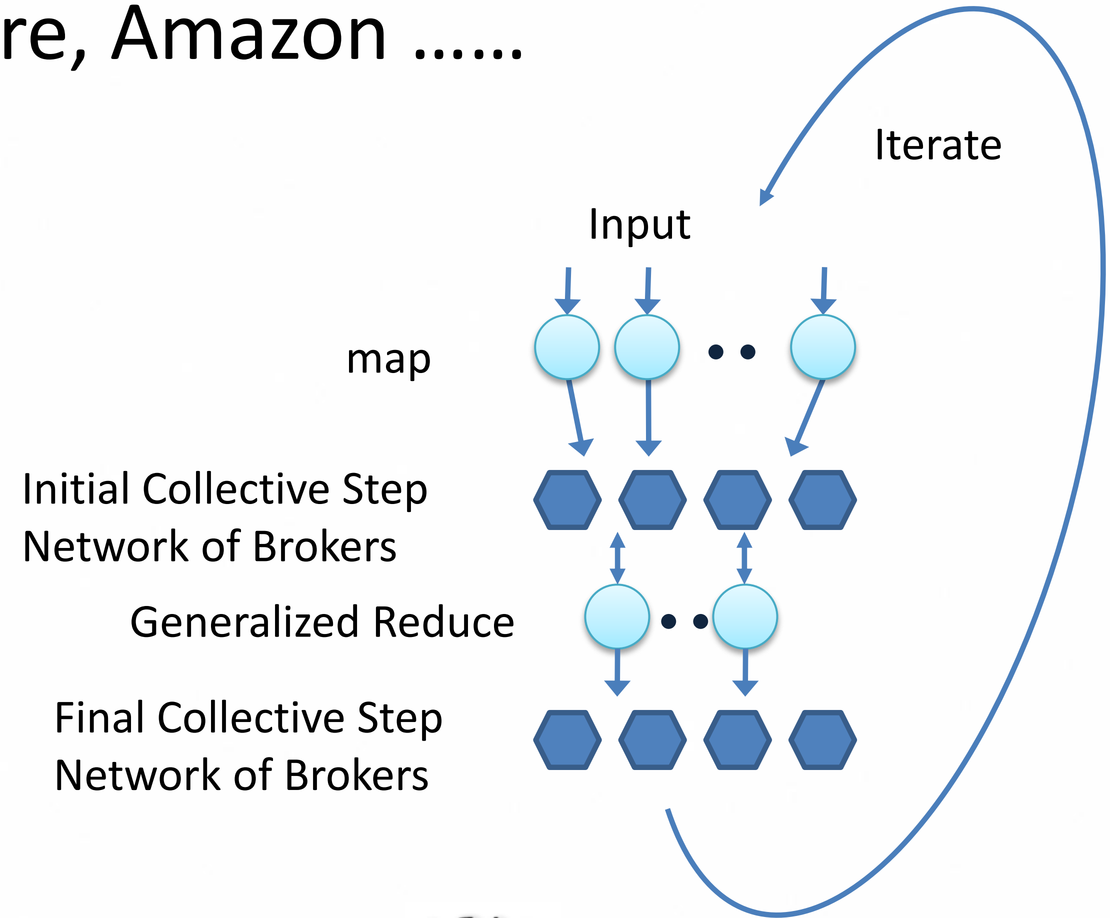
using 32 nodes (256 CPU cores) of Crevasse



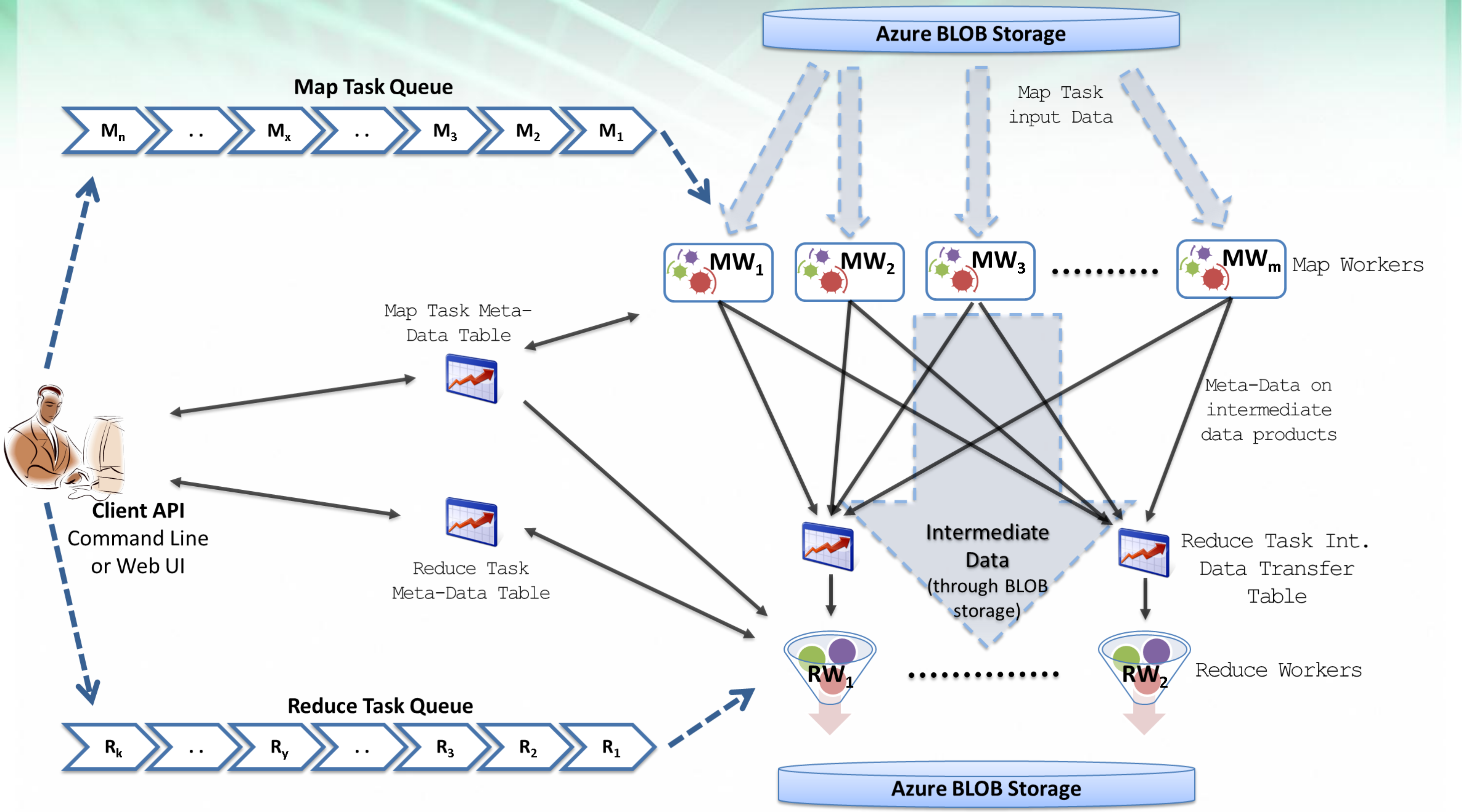
IPC

Map Collective Model (Judy Qiu)

- Combine MPI and MapReduce ideas
- Implement collectives optimally on Infiniband, Azure, Amazon ……



MapReduceRoles4Azure Architecture



Azure **Queues** for scheduling, **Tables** to store meta-data and monitoring data, **Blobs** for input/output/intermediate data storage.

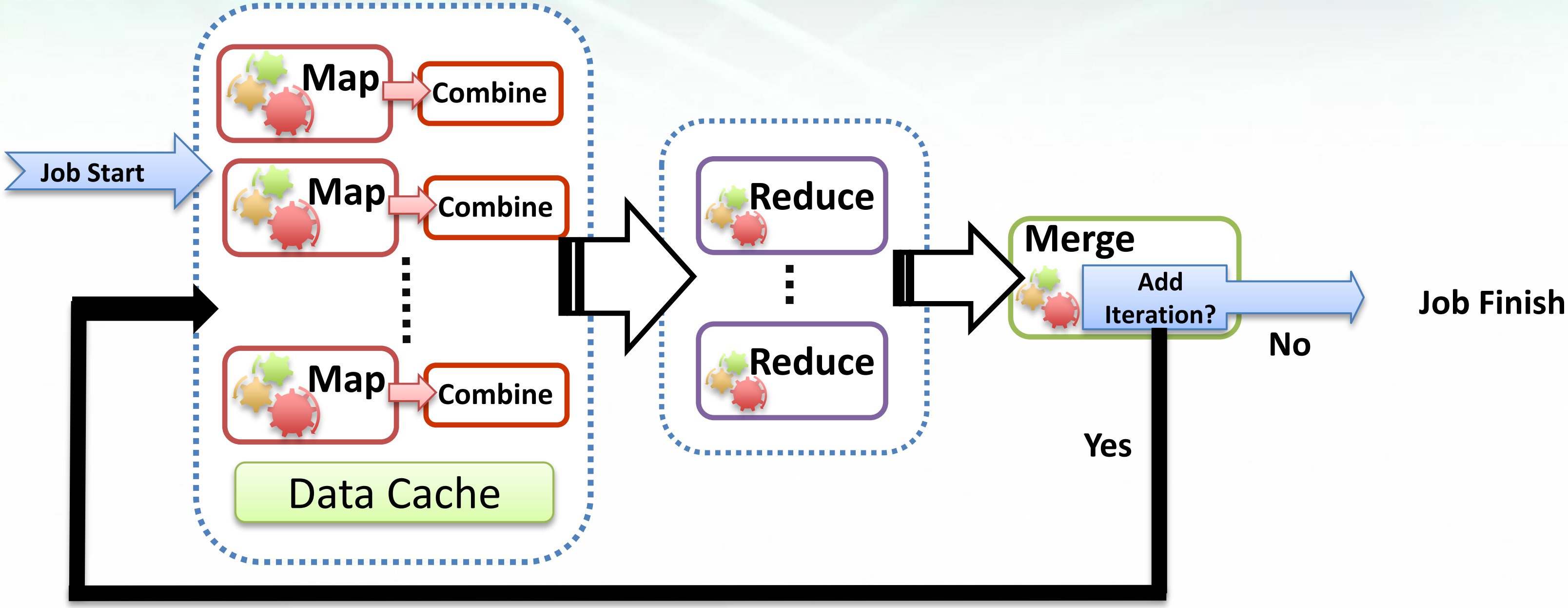
Future Grid https://portal.futuregrid.org

SALSA HPC

MapReduceRoles4Azure

- Use distributed, highly scalable and highly available cloud services as the building blocks.
  - **Azure Queues** for task scheduling.
  - **Azure Blob storage** for input, output and intermediate data storage.
  - **Azure Tables** for meta-data storage and monitoring
- Utilize eventually-consistent , high-latency cloud services effectively to deliver performance comparable to traditional MapReduce runtimes.
- Minimal management and maintenance overhead
- Supports dynamically scaling up and down of the compute resources.
- MapReduce fault tolerance
- http://salsahpc.indiana.edu/mapreduceroles4azure/

High Level Flow Twister4Azure

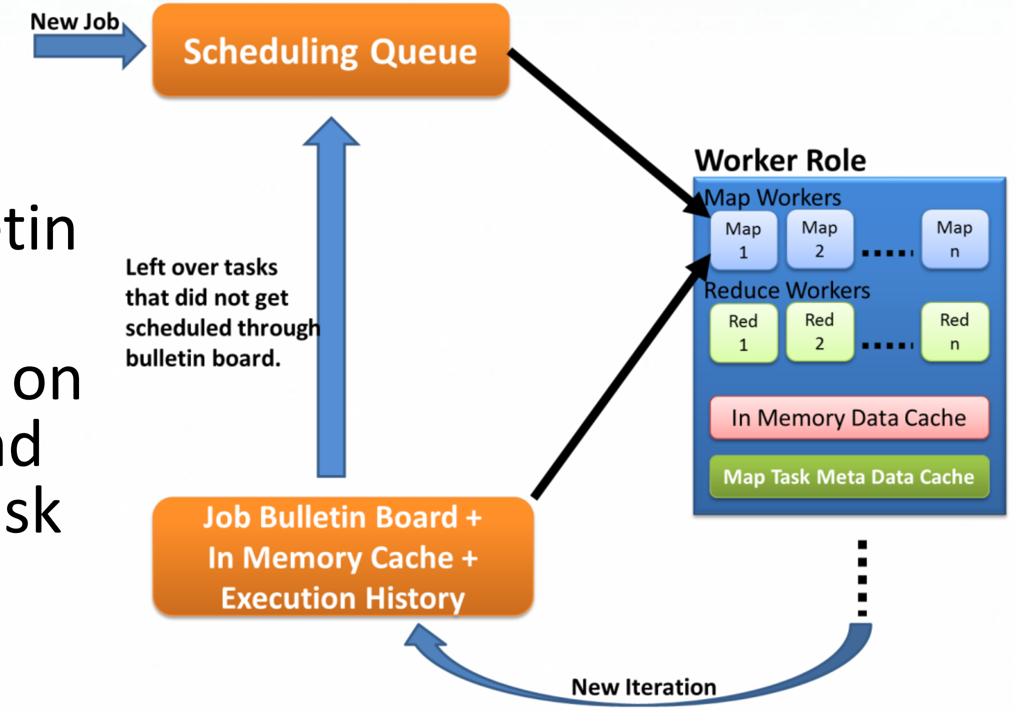


Hybrid scheduling of the new iteration

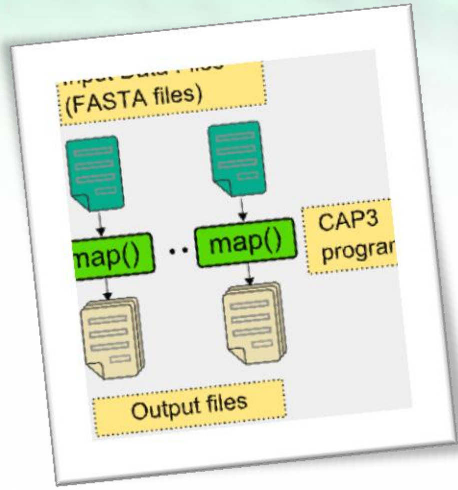
- Merge Step
- In-Memory Caching of static data
- Cache aware hybrid scheduling using Queues as well as using a bulletin board (special table)

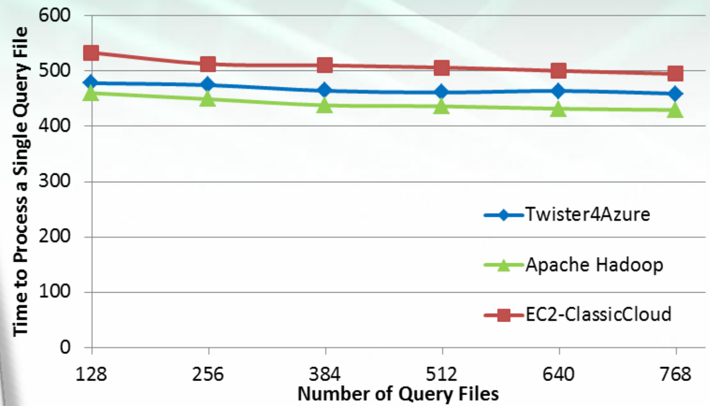
Cache aware scheduling

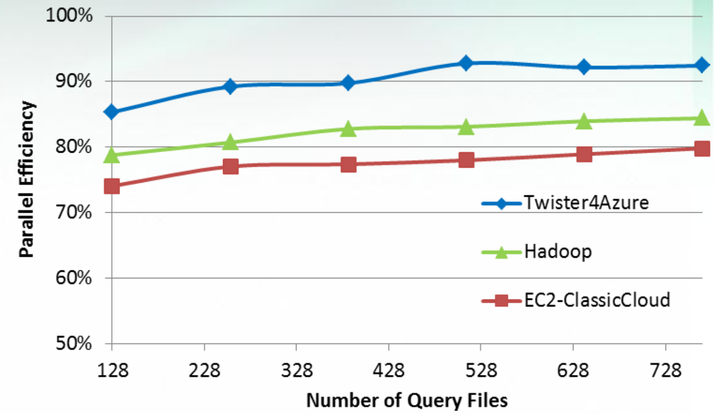
- New Job (1st iteration)
  - Through queues
- New iteration
  - Publish entry to Job Bulletin Board
  - Workers pick tasks based on in-memory data cache and execution history (MapTask Meta-Data cache)
  - Any tasks that do not get scheduled through the bulletin board will be added to the queue.



Performance Comparisons

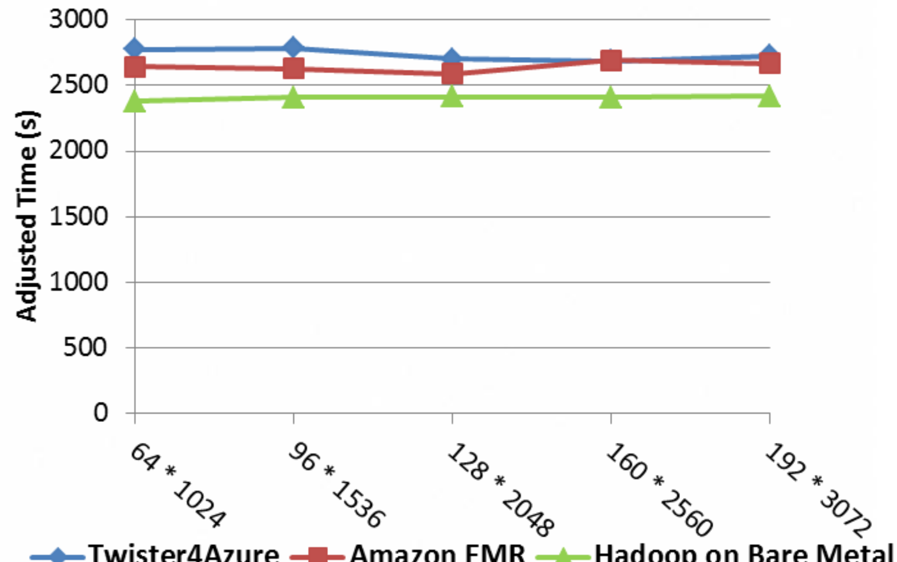




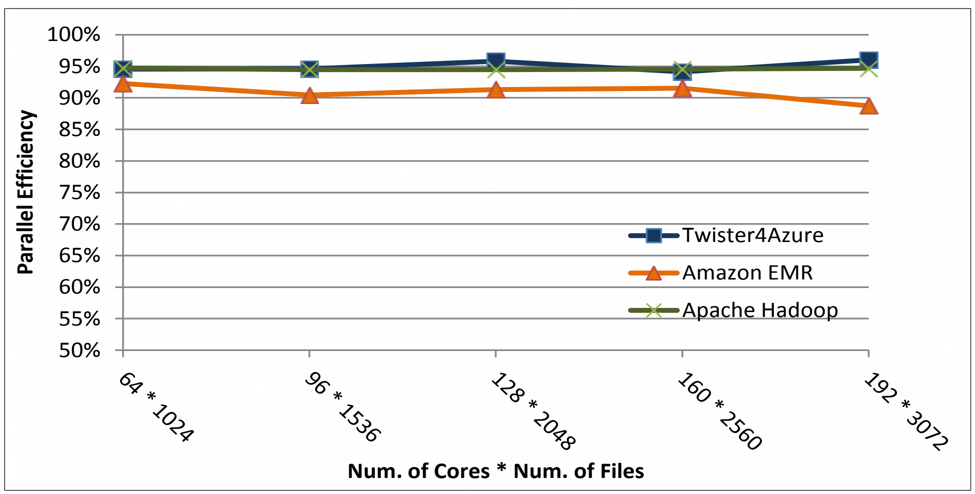


BLAST Sequence Search

Smith Waterman Sequence Alignment



Cap3 Sequence Assembly

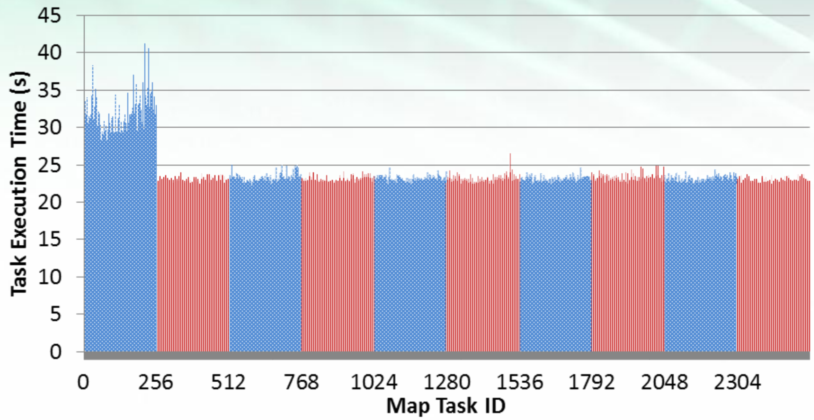


Future Grid

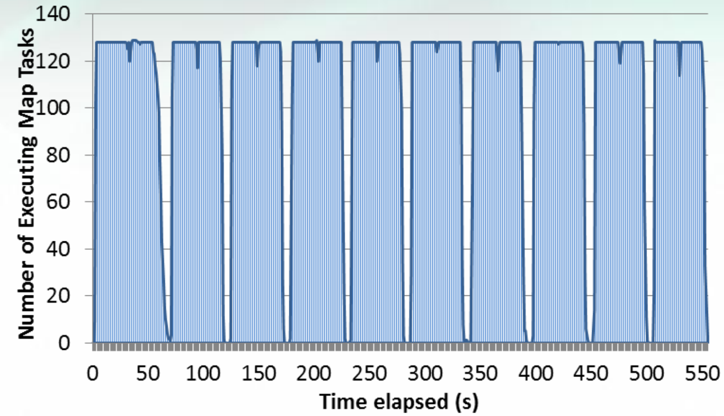
https://portal.futuregrid.org

SALSA HPC

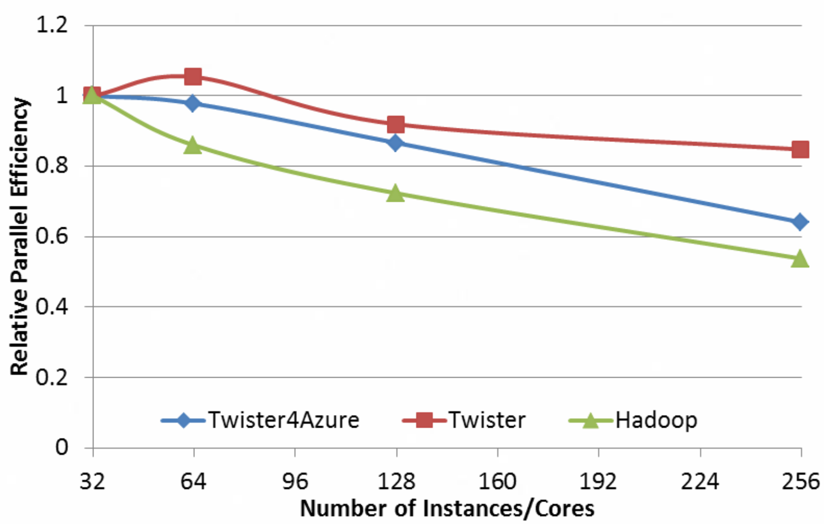
Performance – Kmeans Clustering



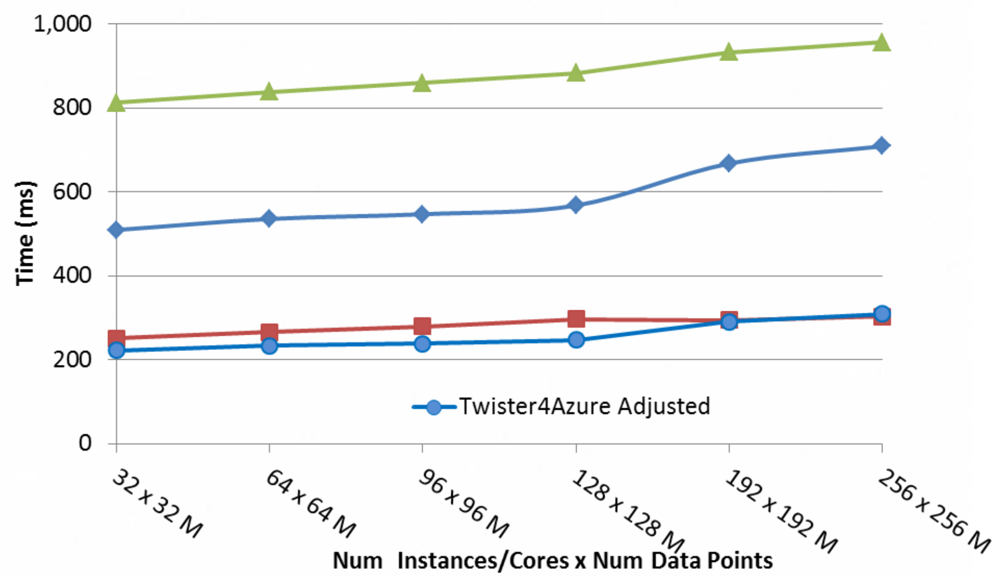
Task Execution Time Histogram



Number of Executing Map Task Histogram



Strong Scaling with 128M Data Points

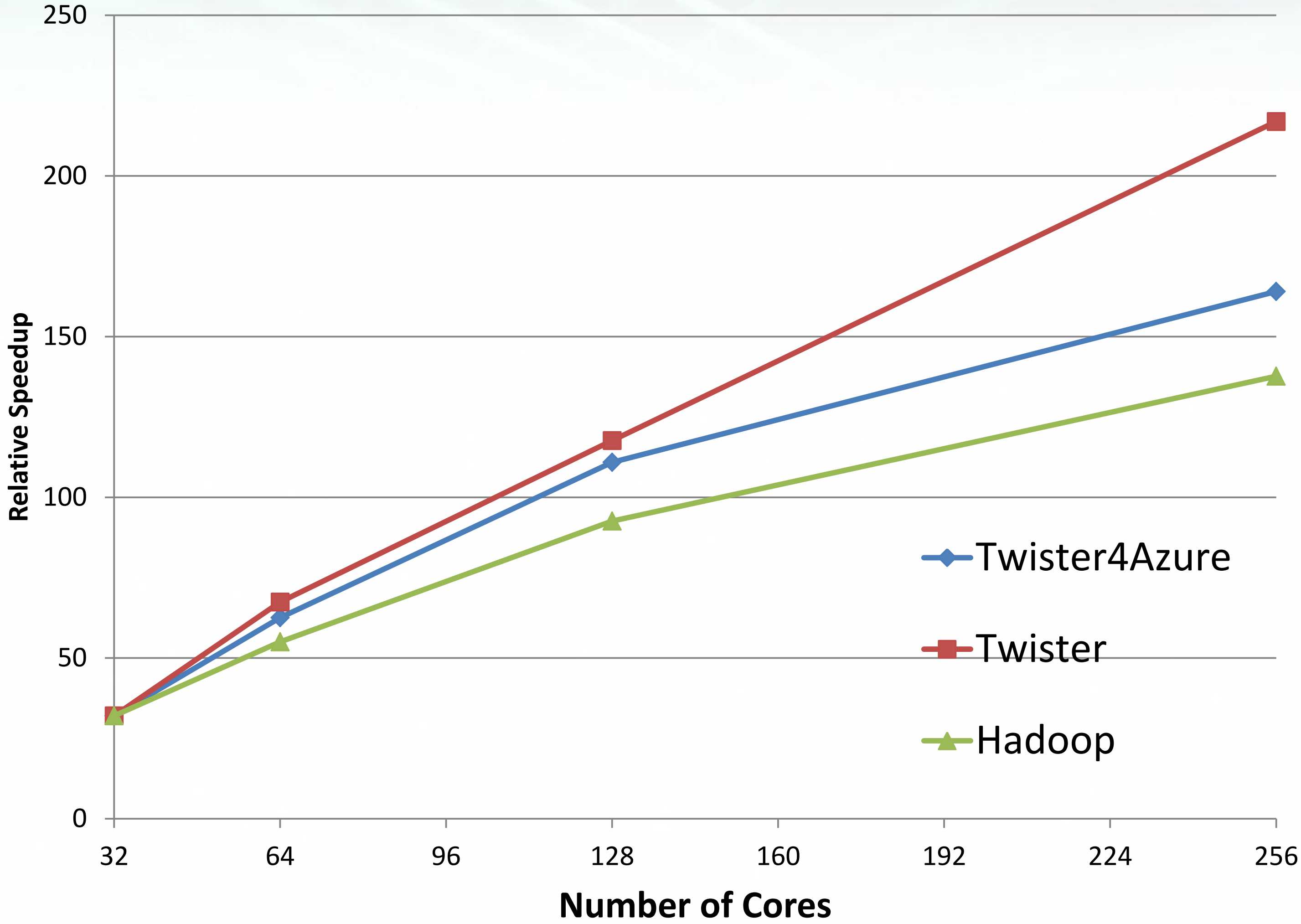


Weak Scaling

Future Grid https://portal.futuregrid.org

SALSA HPC

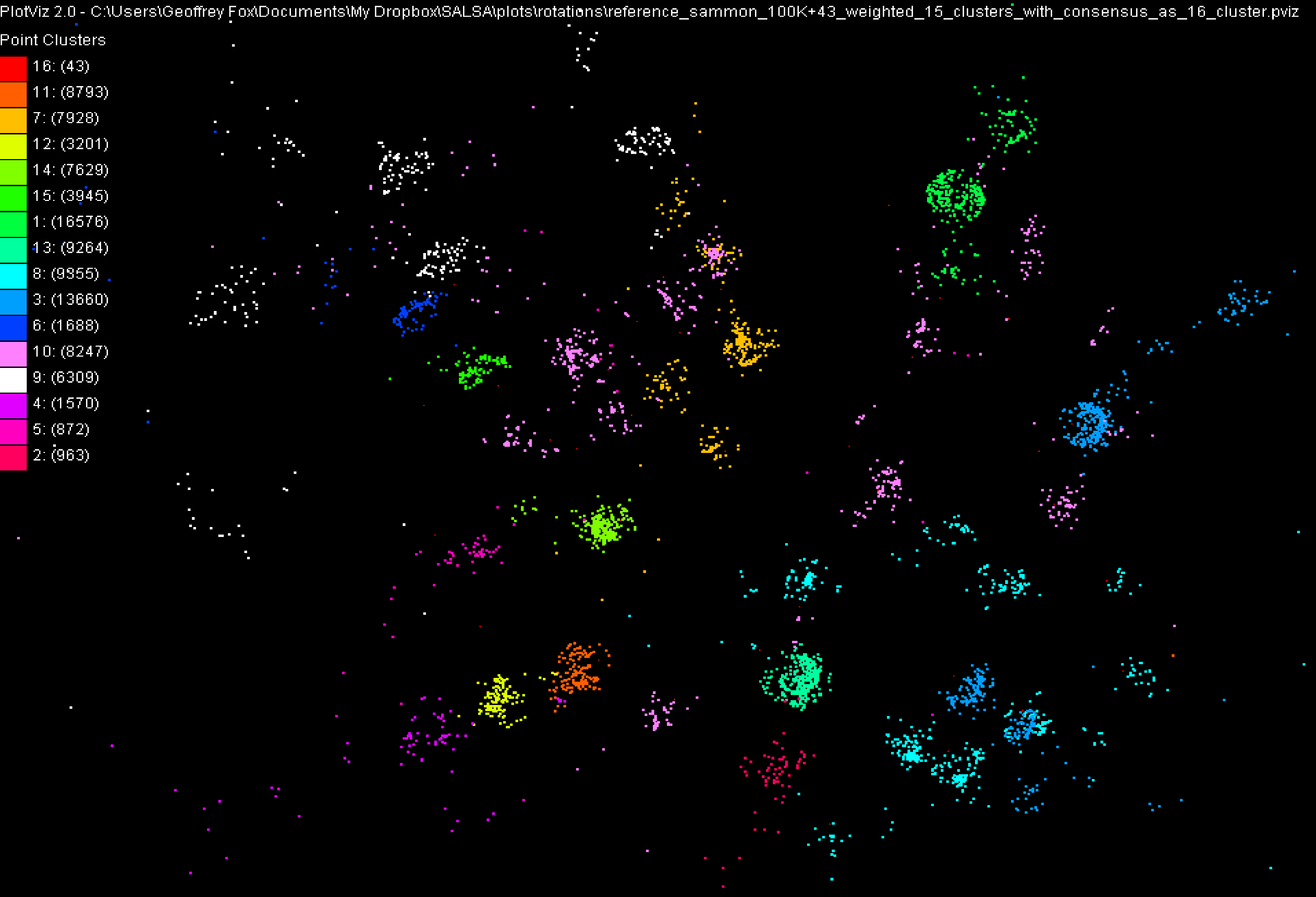
Kmeans Speedup from 32 cores



https://portal.futuregrid.org

Look at one problem in detail

- Visualizing Metagenomics where sequences are ~1000 dimensions
- Map sequences to 3D so you can visualize
- Minimize Stress $$\sigma(\boldsymbol{X}) = \sum_{i<j\leq N} w_{ij}(d_{ij}(\boldsymbol{X}) - \delta_{ij})^2$$
- Improve with deterministic annealing (gives lower stress with less variation between random starts)
- Need to iterate Expectation Maximization
- N^2 dissimilarities (Smith Waterman, Needleman-Wunsch, Blast) δ_{ij}
- Communicate N positions **X** between steps



100,043 Metagenomics Sequences mapped to 3D

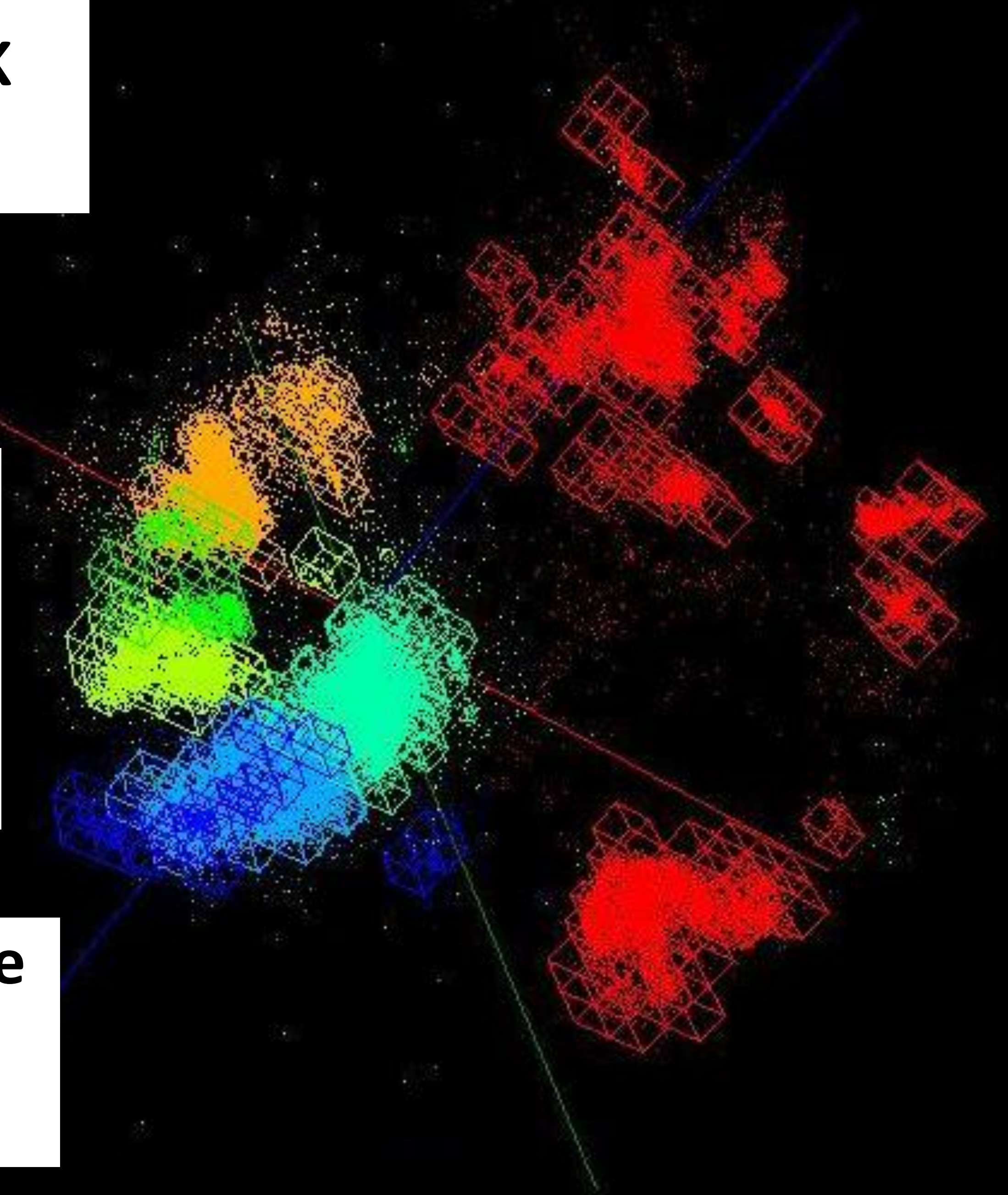
Its an $O(N^2)$ Problem

- 100,000 sequences takes a few days on 768 cores 32 nodes Windows Cluster Tempest
- Could just run 680K on 6.8^2 larger machine but lets try to be “cleverer” and use hierarchical methods
- Start with 100K sample run fully
- Divide into “megaregions” using 3D projection
- Interpolate full sample into megaregions and analyze latter separately
- See http://salsahpc.org/millionseq/16SrRNA_index.html

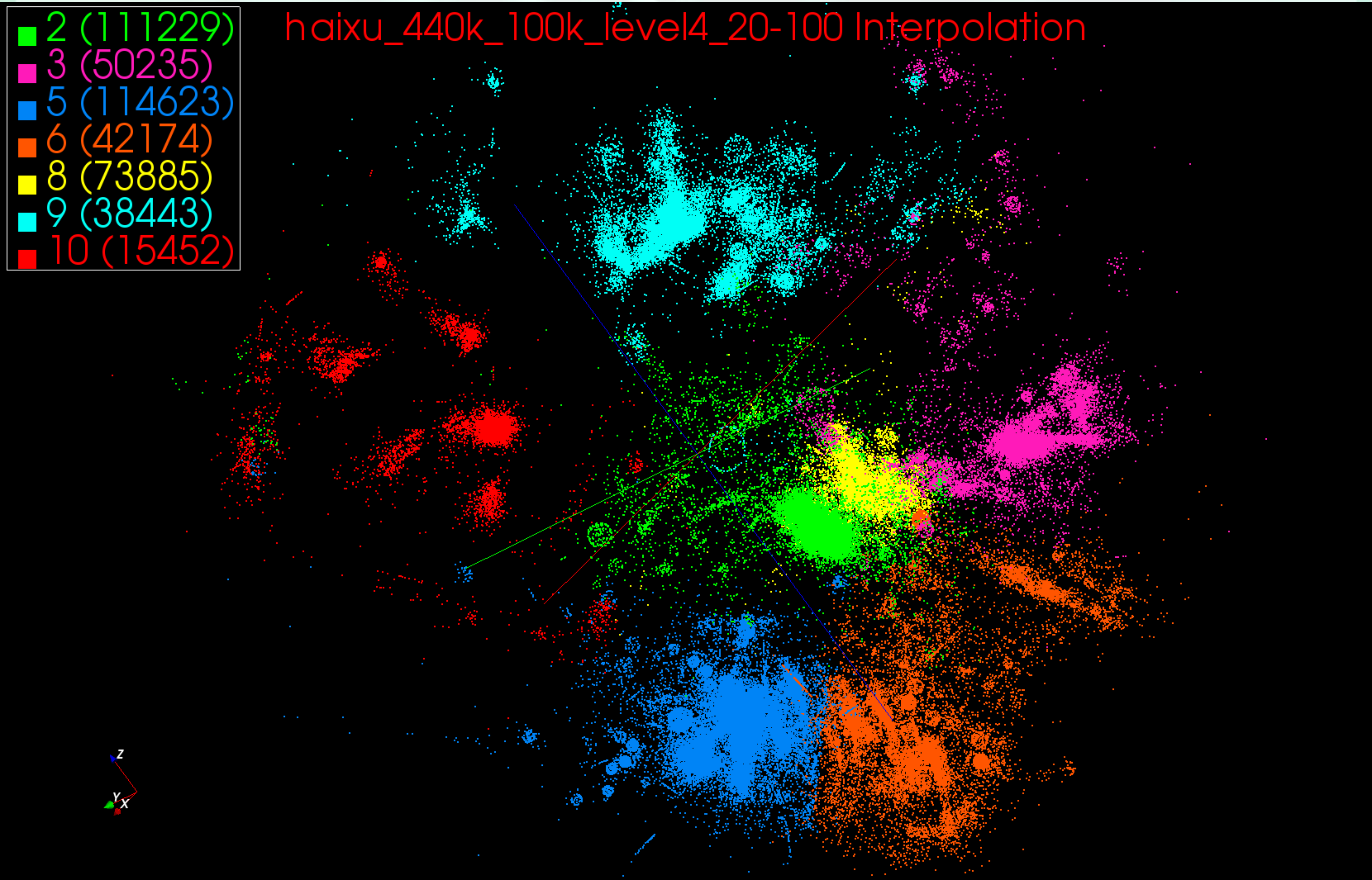
OctTree for 100K sample of Fungi

Use Barnes Hut OctTree originally developed to make $O(N^2)$ astrophysics $O(N\log N)$

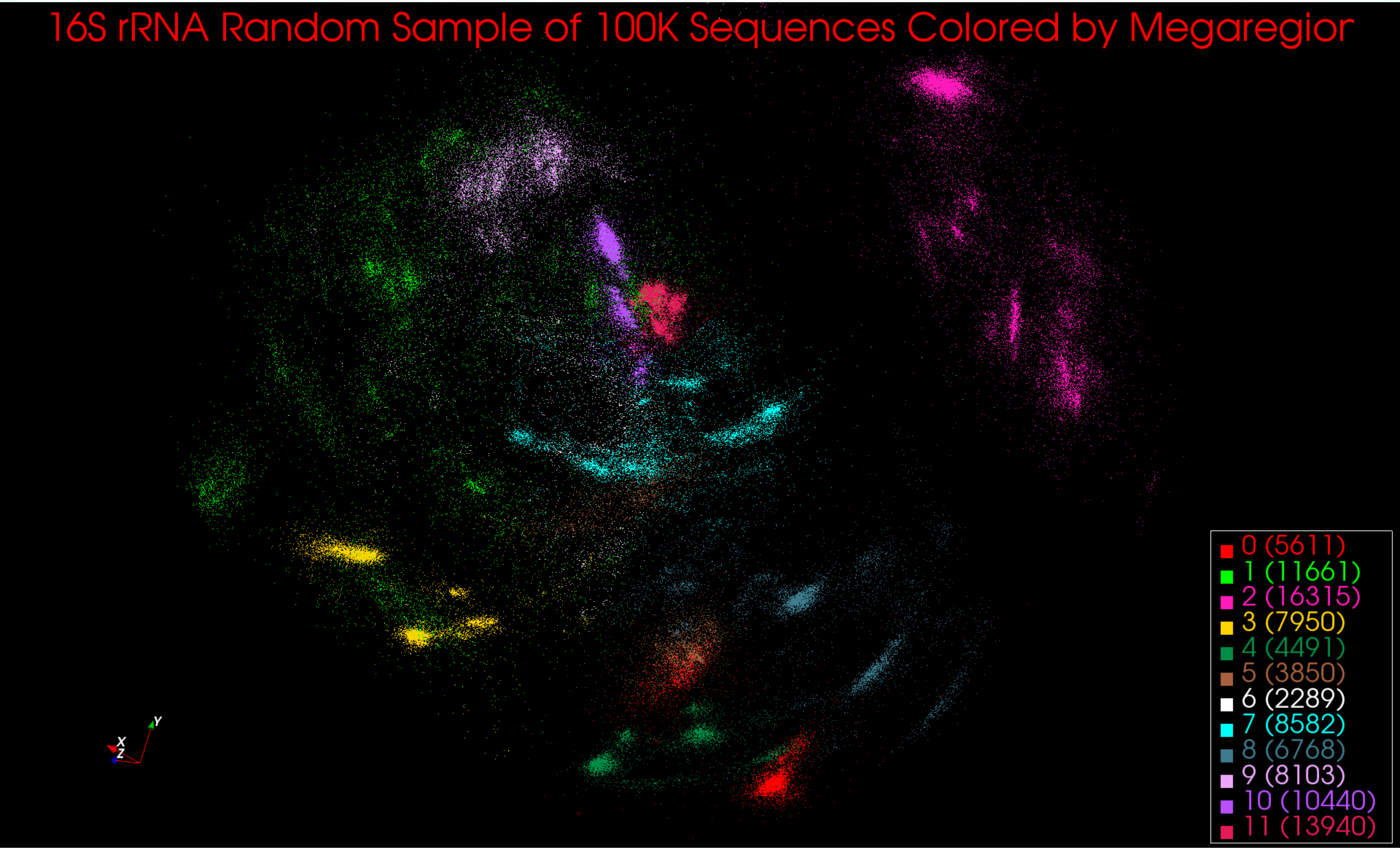
We will use OctTree for logarithmic interpolation



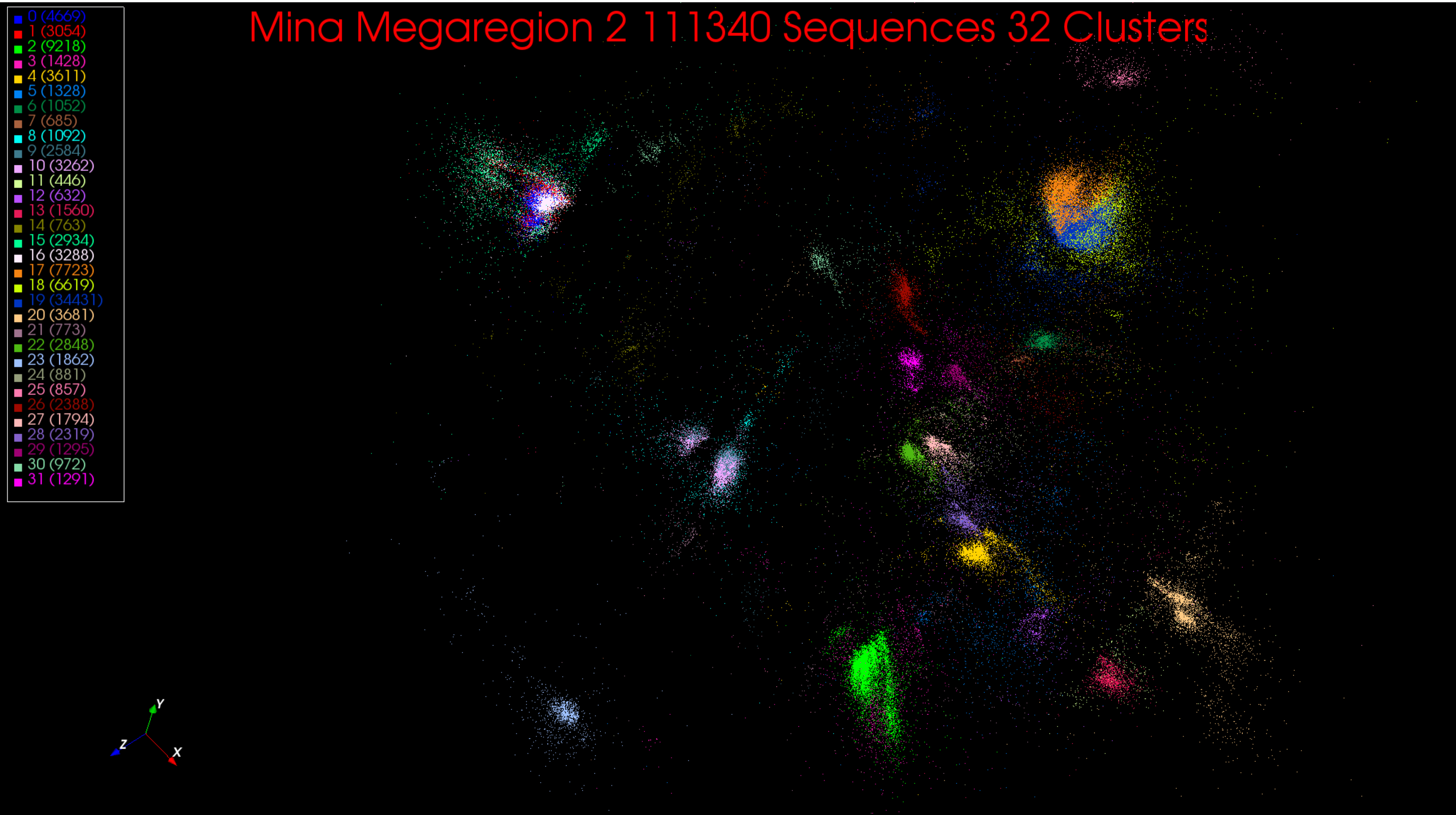
440K Interpolated



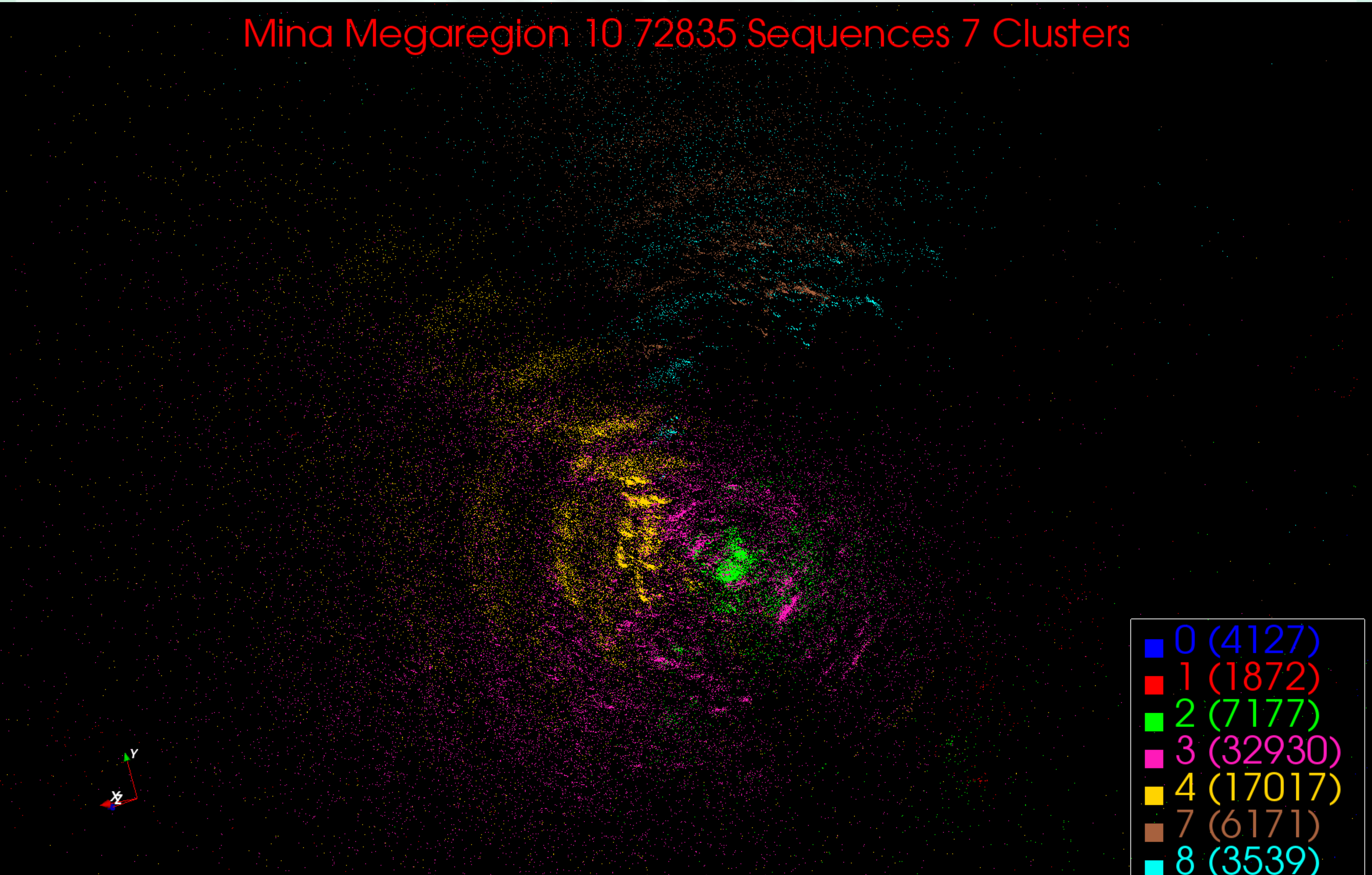
12 Megaregions defined from initial sample



One Megaregion divided into many clusters

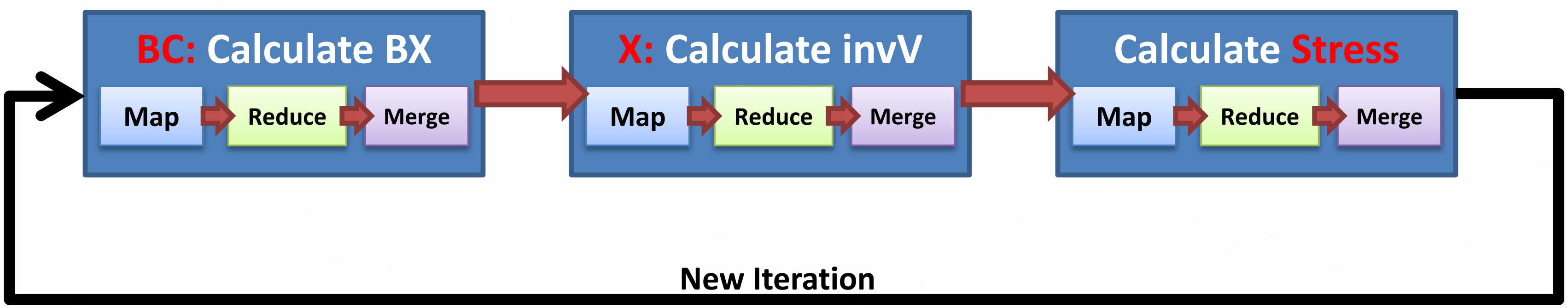


A more compact Megaregion

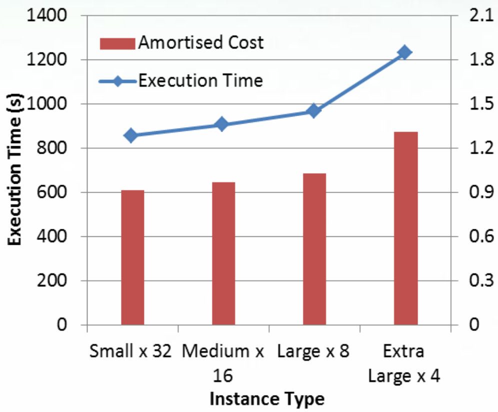


Multi-Dimensional-Scaling

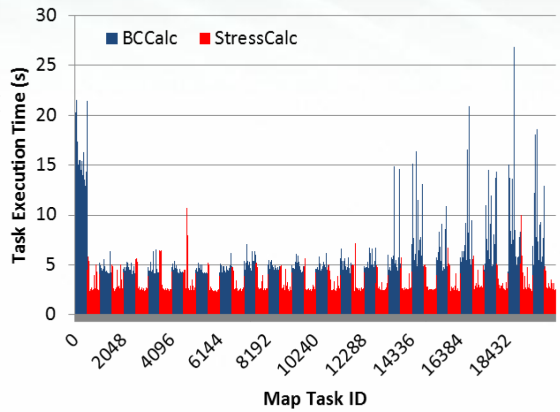
- Many iterations
- Memory & Data intensive
- 3 Map Reduce jobs per iteration
- $\underline{X}_k = invV * B(\underline{X}_{(k-1)}) * \underline{X}_{(k-1)}$
- 2 matrix vector multiplications termed BC and X



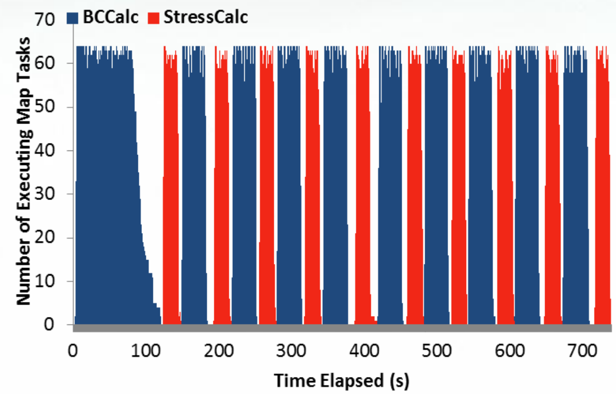
Performance – Multi Dimensional Scaling



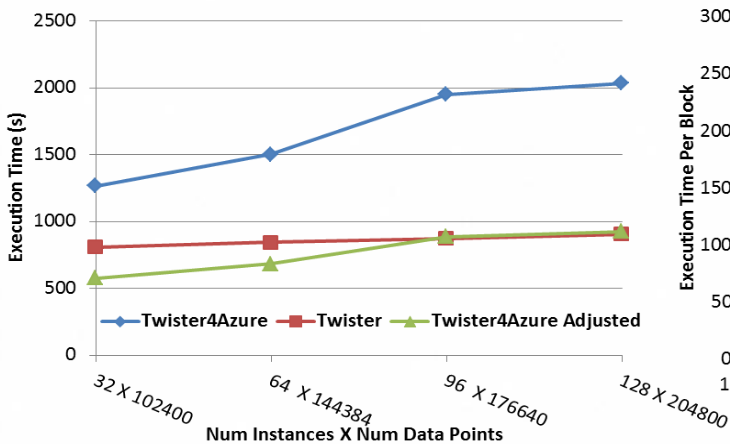
Azure Instance Type Study



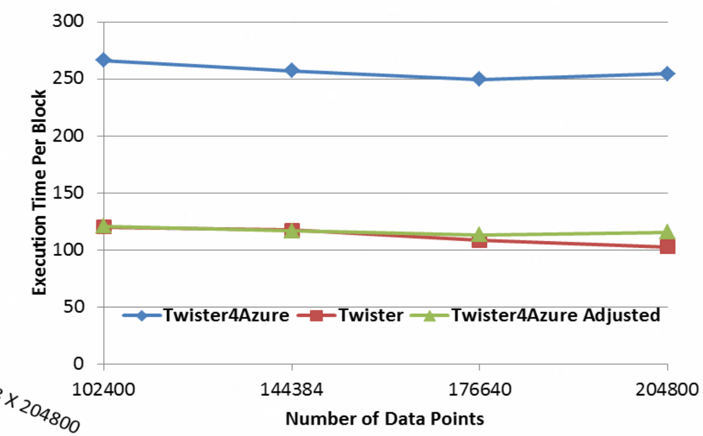
Task Execution Time Histogram



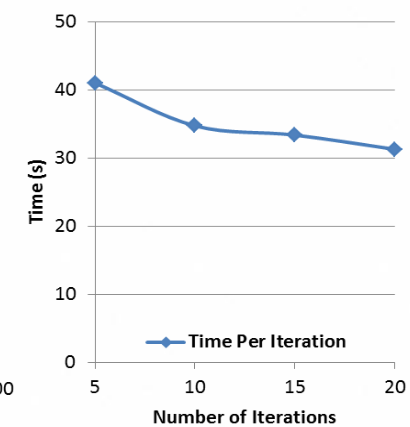
Number of Executing Map Task Histogram



Weak Scaling



Data Size Scaling



Increasing Number of Iterations

Future Grid

https://portal.futuregrid.org

SALSA HPC

Twister4Azure Conclusions

- Twister4Azure enables users to easily and efficiently perform large scale iterative data analysis and scientific computations on Azure cloud.
  - Supports classic and iterative MapReduce
  - Non pleasingly parallel use of Azure
- Utilizes a hybrid scheduling mechanism to provide the caching of static data across iterations.
- Should integrate with workflow systems
- Plenty of testing and improvements needed!
- Open source: Please use http://salsahpc.indiana.edu/twister4azure

What was/can be done where?

- **Dissimilarity Computation** (largest time)
  - Done using Twister on HPC
  - Have running on Azure and Dryad
  - Used Tempest with MPI as well (MPI.NET failed(!), Twister didn't)
- **Full MDS**
  - Done using MPI on Tempest
  - Have running well using Twister on HPC clusters and Azure
- **Pairwise Clustering**
  - Done using MPI on Tempest
  - Probably need to change algorithm to get good efficiency on cloud
- **Interpolation** (smallest time)
  - Done using Twister on HPC
  - Running on Azure

Expectation Maximization and Iterative MapReduce

- **Clustering** and **Multidimensional Scaling** are both EM (**expectation maximization**) using deterministic annealing for improved performance
- **EM** tends to be **good** for **clouds** and **Iterative MapReduce**
  – Quite **complicated computations** (so compute largish compared to communicate)
  – Communication is **Reduction** operations (global sums in our case)
  – See also **Latent Dirichlet Allocation** and related Information Retrieval algorithms similar structure

DA-PWC EM Steps (E is red, M Black)

k runs over clusters; i,j, α points

1) $A(k) = -0.5 \sum_{i=1}^{N} \sum_{j=1}^{N} \delta(i, j) \langle M_i(k)\rangle \langle M_j(k)\rangle / \langle C(k)\rangle^2$
2) $B_\alpha(k) = \sum_{i=1}^{N} \delta(i, \alpha) \langle M_i(k)\rangle / \langle C(k)\rangle$
3) $\varepsilon_\alpha(k) = (B_\alpha(k) + A(k))$
4) $\langle M_i(k)\rangle = p(k) \exp(-\varepsilon_i(k)/T) / \sum_{k'=1}^{K} p(k') \exp(-\varepsilon_i(k')/T)$
5) $C(k) = \sum_{i=1}^{N} \langle M_i(k)\rangle$
6) $p(k) = C(k) / N$

Steps 1 global sum (reduction)
Step 1, 2, 5 local sum if $\langle M_i(k)\rangle$ broadcast

- Loop to converge variables; decrease T from ∞; split centers by halving p(k)

May Need New Algorithms

- **DA-PWC** (Deterministically Annealed Pairwise Clustering) splits clusters automatically as temperature lowers and reveals clusters of size O(√T)
- Two approaches to splitting
  1. Look at correlation matrix and see when becomes singular which is a separate parallel step
  2. Formulate problem with multiple centers for each cluster and perturb ever so often spitting centers into 2 groups; unstable clusters separate
- Current MPI code uses first method which will run on Twister as matrix singularity analysis is the usual "power eigenvalue method" (as is page rank)
  - However not very good compute/communicate ratio
- Experiment with second method which "just" EM with better compute/communicate ratio (simpler code as well)

What can we learn?

- There are many **pleasingly parallel data analysis** algorithms which are super for clouds
  - Remember SWG computation longer than other parts of analysis
- There are interesting data mining algorithms needing **iterative parallel run times**
- There are **linear algebra** algorithms with flaky compute/communication ratios
- **Expectation Maximization** good for Iterative MapReduce

Research Issues for (Iterative) MapReduce

- Quantify and Extend that Data analysis for Science seems to work well on Iterative MapReduce and clouds so far.
  - Iterative MapReduce (Map Collective) spans all architectures as unifying idea
- Performance and Fault Tolerance Trade-offs;
  - Writing to disk each iteration (as in Hadoop) naturally lowers performance but increases fault-tolerance
  - Integration of GPU's
- Security and Privacy technology and policy essential for use in many biomedical applications
- Storage: multi-user data parallel file systems have scheduling and management
  - NOSQL and SciDB on virtualized and HPC systems
- Data parallel Data analysis languages: Sawzall and Pig Latin more successful than HPF?
- Scheduling: How does research here fit into scheduling built into clouds and Iterative MapReduce (Hadoop)
  - important load balancing issues for MapReduce for heterogeneous workloads

Authentication and Authorization: Provide single sign in to All system architectures
Workflow: Support workflows that link job components between Grids and Clouds.
Provenance: Continues to be critical to record all processing and data sources
Data Transport: Transport data between job components on Grids and Commercial Clouds respecting custom storage patterns like Lustre v HDFS
Program Library: Store Images and other Program material
Blob: Basic storage concept similar to Azure Blob or Amazon S3
DPFS Data Parallel File System: Support of file systems like Google (MapReduce), HDFS (Hadoop) or Cosmos (dryad) with compute-data affinity optimized for data processing
Table: Support of Table Data structures modeled on Apache Hbase/CouchDB or Amazon SimpleDB/Azure Table. There is “Big” and “Little” tables – generally NOSQL
SQL: Relational Database
Queues: Publish Subscribe based queuing system
Worker Role: This concept is implicitly used in both Amazon and TeraGrid but was (first) introduced as a high level construct by Azure. Naturally support **Elastic Utility Computing**
MapReduce: Support MapReduce Programming model including Hadoop on Linux, Dryad on Windows HPCS and Twister on Windows and Linux. Need Iteration for Datamining
Software as a Service: This concept is shared between Clouds and Grids
Web Role: This is used in Azure to describe user interface and can be supported by portals in Grid or HPC systems

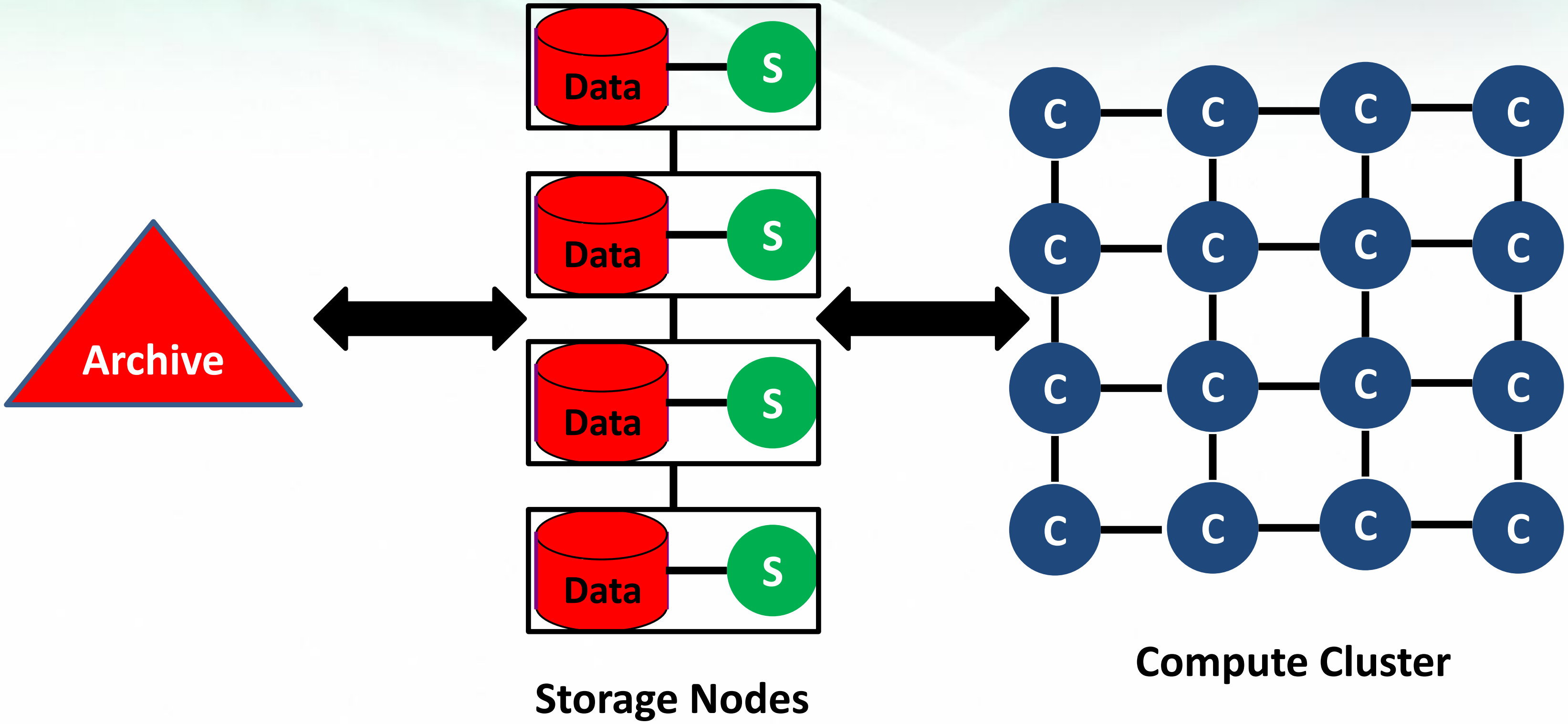
Architecture of Data Repositories?

- Traditionally governments set up repositories for data associated with particular missions
  - For example EOSDIS, GenBank, NSIDC, IPAC for Earth Observation , Gene, Polar Science and Infrared astronomy
  - LHC/OSG computing grids for particle physics
- This is complicated by volume of data deluge, distributed instruments as in gene sequencers (maybe centralize?) and need for complicated intense computing

Clouds as Support for Data Repositories?

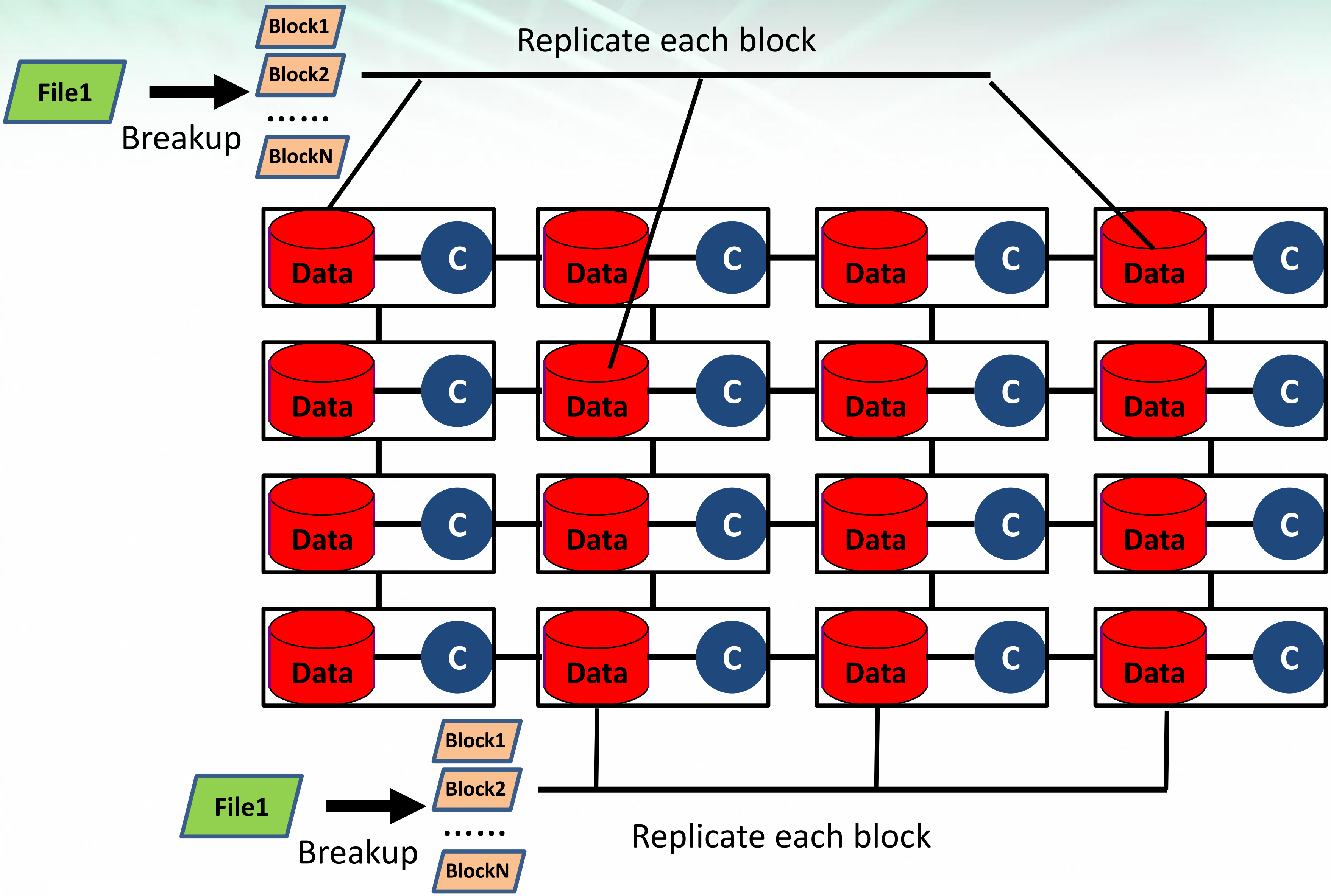
- The data deluge needs cost effective computing
  - Clouds are by definition cheapest
- Shared resources essential (to be cost effective and large)
  - Can't have every scientists downloading petabytes to personal cluster
- Need to reconcile distributed (initial source of) data with shared computing
  - Can move data to (disciple specific) clouds
  - How do you deal with multi-disciplinary studies

Traditional File System?



- Typically a shared file system (Lustre, NFS …) used to support high performance computing
- Big advantages in flexible computing on shared data but doesn't "**bring computing to data**"
- Object stores similar to this?

Data Parallel File System?



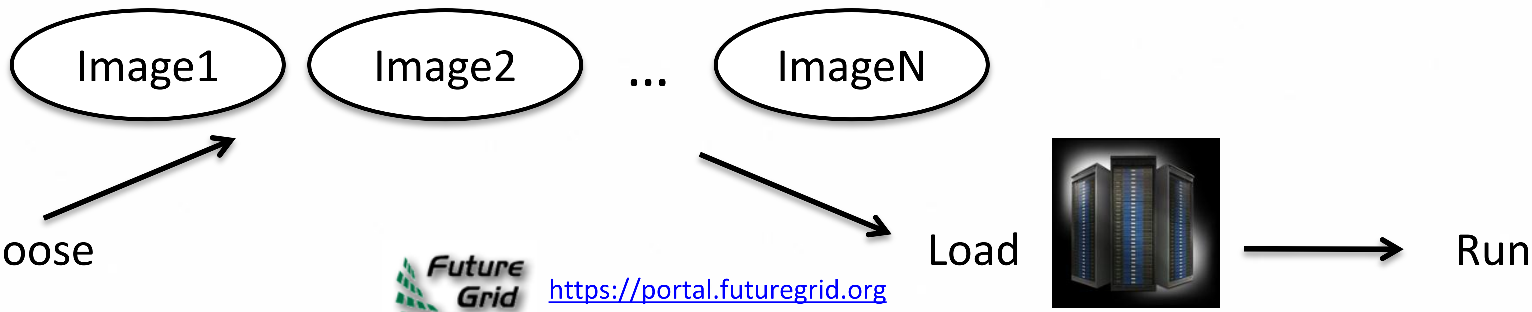
- No archival storage and computing brought to data

FutureGrid key Concepts I

- FutureGrid is an international testbed modeled on Grid5000
- Supporting international Computer Science and Computational Science research in cloud, grid and parallel computing (HPC)
  - Industry and Academia
  - Note much of current use Education, Computer Science Systems and Biology/Bioinformatics
- The FutureGrid testbed provides to its users:
  - A flexible development and testing platform for middleware and application users looking at interoperability, functionality, performance or evaluation
  - Each use of FutureGrid is an experiment that is reproducible
  - A rich education and teaching platform for advanced cyberinfrastructure (computer science) classes

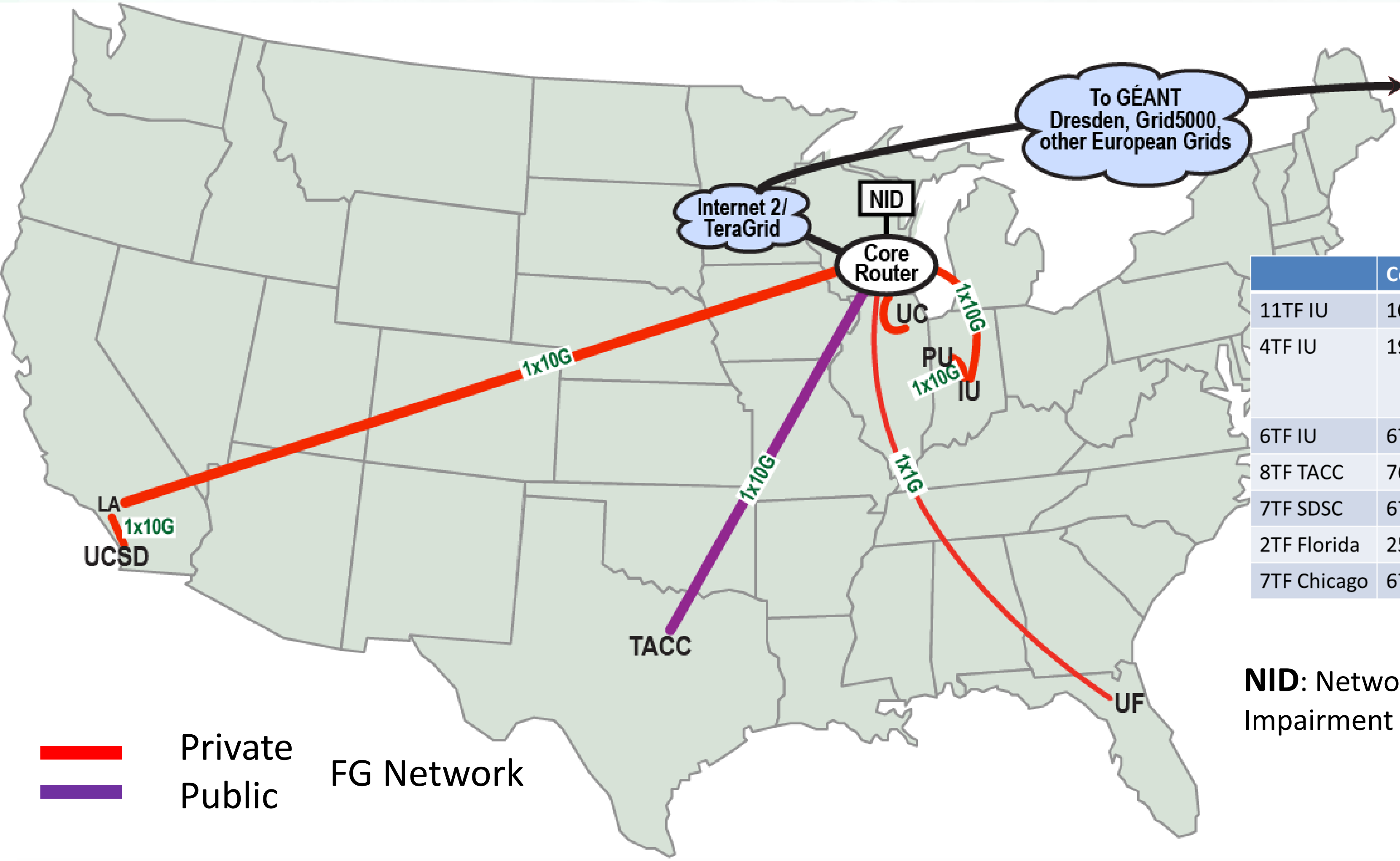
FutureGrid key Concepts II

- Rather than loading images onto VM’s, FutureGrid supports Cloud, Grid and Parallel computing environments by dynamically provisioning software as needed onto “bare-metal” using Moab/xCAT
  - Image library for MPI, OpenMP, Hadoop, Dryad, gLite, Unicore, Globus, Xen, ScaleMP (distributed Shared Memory), Nimbus, Eucalyptus, OpenNebula, KVM, Windows …..
- Growth comes from users depositing novel images in library
- FutureGrid has ~4000 (will grow to ~5000) distributed cores with a dedicated network and a Spirent XGEM network fault and delay generator



https://portal.futuregrid.org

FutureGrid: a Grid/Cloud/HPC Testbed



	Cores	
11TF IU	1024	IBM
4TF IU	192	12 TB Disk 192 GB mem, GPU on 8 nodes
6TF IU	672	Cray XT5M
8TF TACC	768	Dell
7TF SDSC	672	IBM
2TF Florida	256	IBM
7TF Chicago	672	IBM

NID: Network Impairment Device

https://portal.futuregrid.org

FutureGrid Partners

- Indiana University (Architecture, core software, Support)
- Purdue University (HTC Hardware)
- San Diego Supercomputer Center at University of California San Diego (INCA, Monitoring)
- University of Chicago/Argonne National Labs (Nimbus)
- University of Florida (ViNE, Education and Outreach)
- University of Southern California Information Sciences (Pegasus to manage experiments)
- University of Tennessee Knoxville (Benchmarking)
- University of Texas at Austin/Texas Advanced Computing Center (Portal)
- University of Virginia (OGF, Advisory Board and allocation)
- Center for Information Services and GWT-TUD from Technische Universtität Dresden. (VAMPIR)
- Red institutions have FutureGrid hardware

5 Use Types for FutureGrid

- **~122** approved projects over last 10 months
- **Training Education and Outreach (11%)**
  - Semester and short events; promising for non research intensive universities
- **Interoperability test-beds (3%)**
  - Grids and Clouds; **Standards**; Open Grid Forum OGF really needs
- **Domain Science applications (34%)**
  - Life sciences highlighted (17%)
- **Computer science (41%)**
  - Largest current category
- **Computer Systems Evaluation (29%)**
  - TeraGrid (TIS, TAS, XSEDE), OSG, EGI, Campuses
- Clouds are meant to need less support than other models; FutureGrid needs more **user support** …….

Software Components

- Portals including “Support” “use FutureGrid” “Outreach”
- Monitoring – INCA, Power (GreenIT)
- **Experiment Manager**: specify/workflow
- Image Generation and Repository
- Intercloud Networking ViNE
- Virtual Clusters built with virtual networks
- Performance library
- **Rain** or **R**untime **A**daptable **I**nsertio**N** Service for images
- Security Authentication, Authorization,

“Research”

Above and below

Nimbus OpenStack Eucalyptus