FUTURE WORLD 2011 2031

Iterative MapReduce Enabling HPC-Cloud Interoperability

IIT, Chicago, November 4, 2011

SALSA HPC Group

http://salsahpc.indiana.edu

Indiana University

Twister4Azure Iterative MapReduce for Azure Cloud

Twister Cross Platform Iterative MapReduce

PlotViz

**A New Book from Morgan Kaufmann Publishers, an imprint of Elsevier, Inc.,
Burlington, MA 01803, USA. (ISBN: 9780123858801)**

Distributed Systems and Cloud Computing:

From Parallel Processing to the Internet of Things

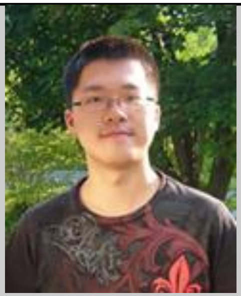
Kai Hwang, Geoffrey Fox, Jack Dongarra

SALSA HPC Group

Twister

Bingjing Zhang, Richard Teng

Funded by Microsoft, Indiana University's Faculty Research Support Program and NSF OCI-1032677 Grant



Twister4Azure

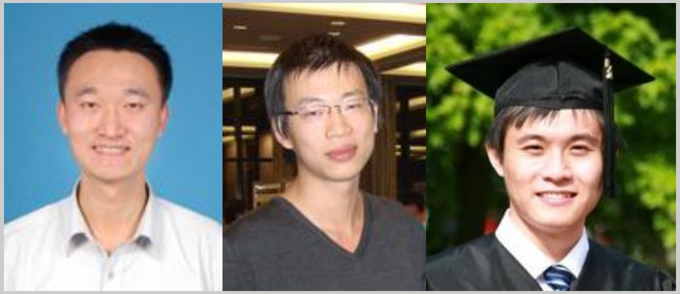
Thilina Gunarathne

Funded by Microsoft

High-Performance Visualization Algorithms For Data-Intensive Analysis

Seung-Hee Bae and Jong Youl Choi

Funded by NIH Grant 1RC2HG005806-01



DryadLINQ CTP Evaluation

Hui Li, Yang Ruan, and Yuduo Zhou

Funded by Microsoft

Cloud Storage, FutureGrid

Xiaoming Gao, Stephen Wu

Funded by Indiana University's Faculty Research Support Program and Natural Science Foundation Grant 0910812

Million Sequence Challenge

Saliya Ekanayake, Adam Hughs, Yang Ruan

Funded by NIH Grant 1RC2HG005806-01

Cyberinfrastructure for Remote Sensing of Ice Sheets

Jerome Mitchell

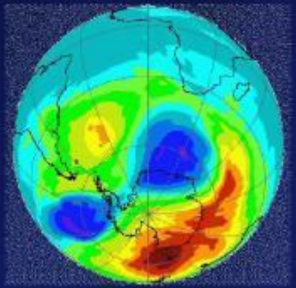
Funded by NSF Grant OCI-0636361

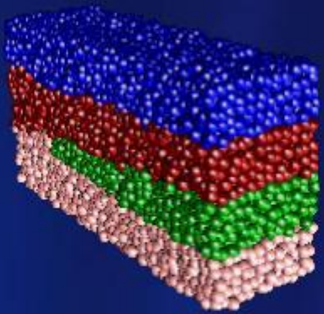
SALSA HPC

Science 2020

"In the last two decades advances in computing technology, from processing speed to network capacity and the Internet, have revolutionized the way scientists work.

From sequencing genomes to monitoring the Earth's climate, many recent scientific advances would not have been possible without a parallel increase in computing power - and with revolutionary technologies such as the quantum computer edging towards reality, *what will the relationship between computing and science bring us over the next 15 years?"*





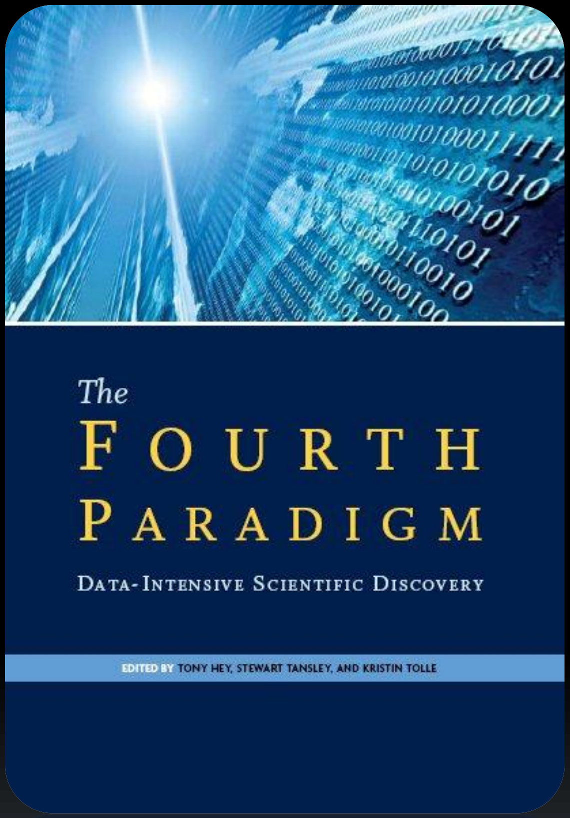
Evolving Science

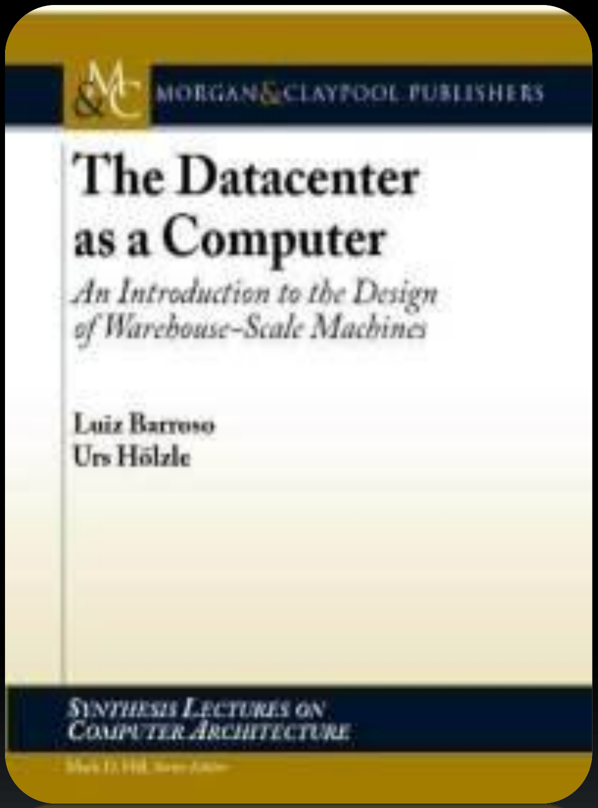
- Thousand years ago:
  science was **empirical**
  describing natural phenomena
- Last few hundred years:
  theoretical branch
  using models, generalizations
- Last few decades:
  a **computational** branch
  simulating complex phenomena
- Today:
  data exploration (eScience)
  synthesizing theory, experiment and computation with advanced data management and statistics
  → new algorithms!

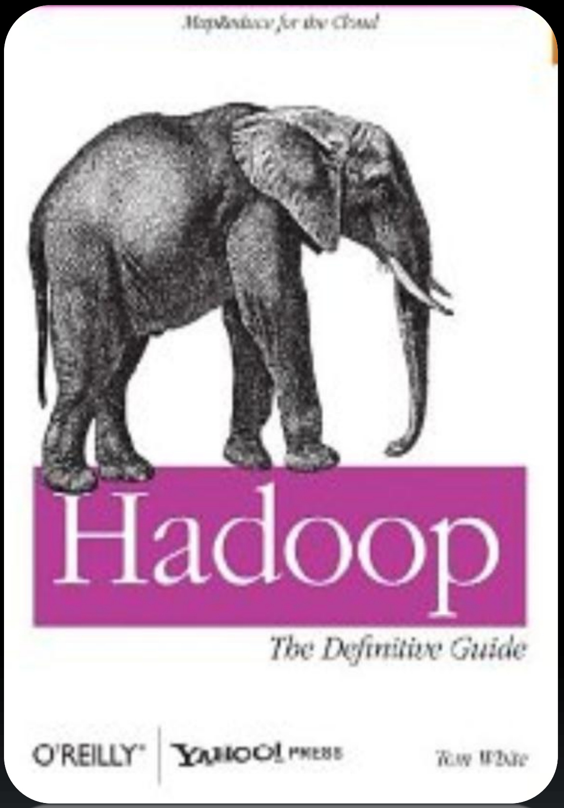
$$\left(\frac{\dot{a}}{a}\right)^2 = \frac{4\pi G\rho}{3} - K\frac{c^2}{a^2}$$

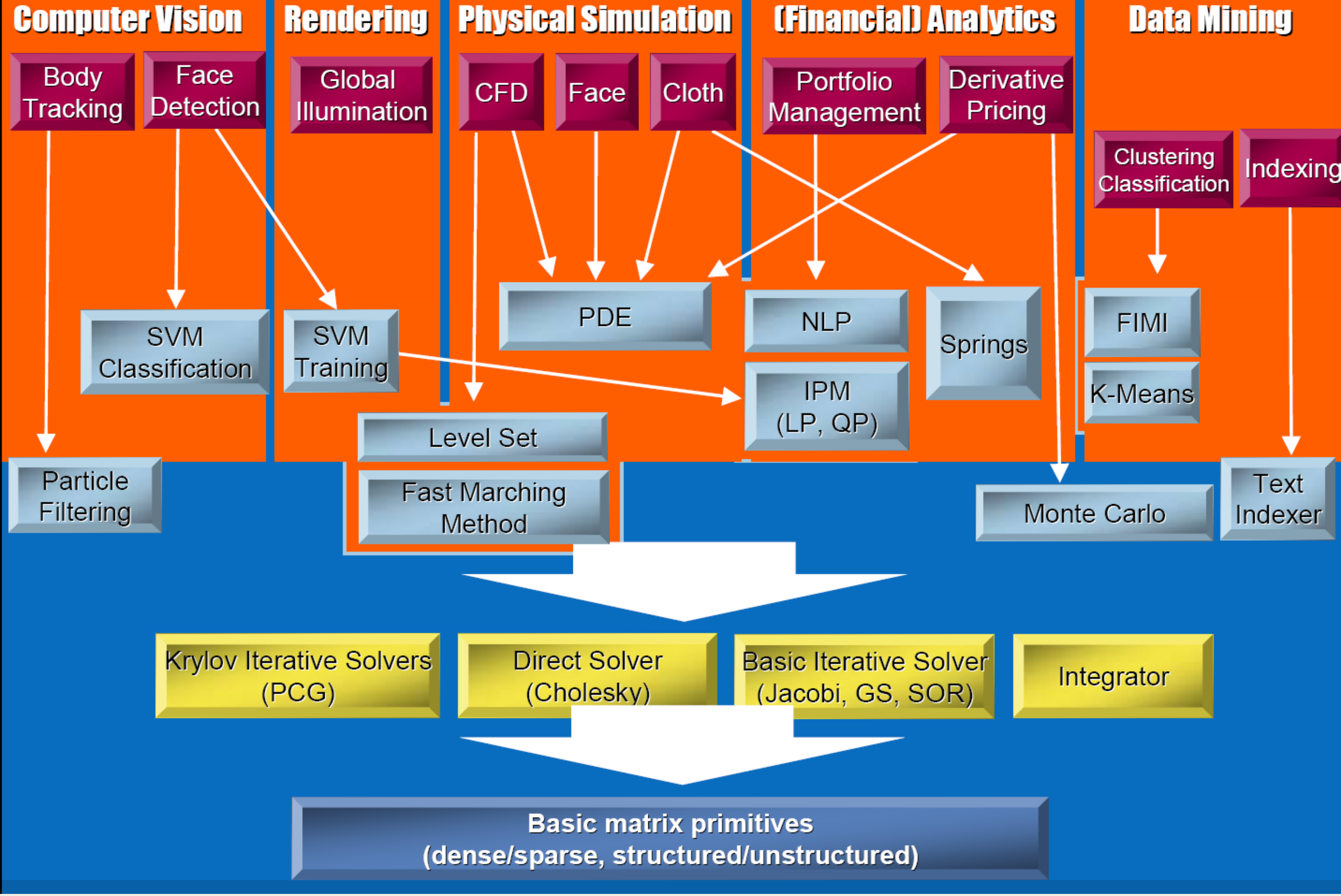
Alex Szalay, The Johns Hopkins University

Paradigm Shift in Data Intensive Computing



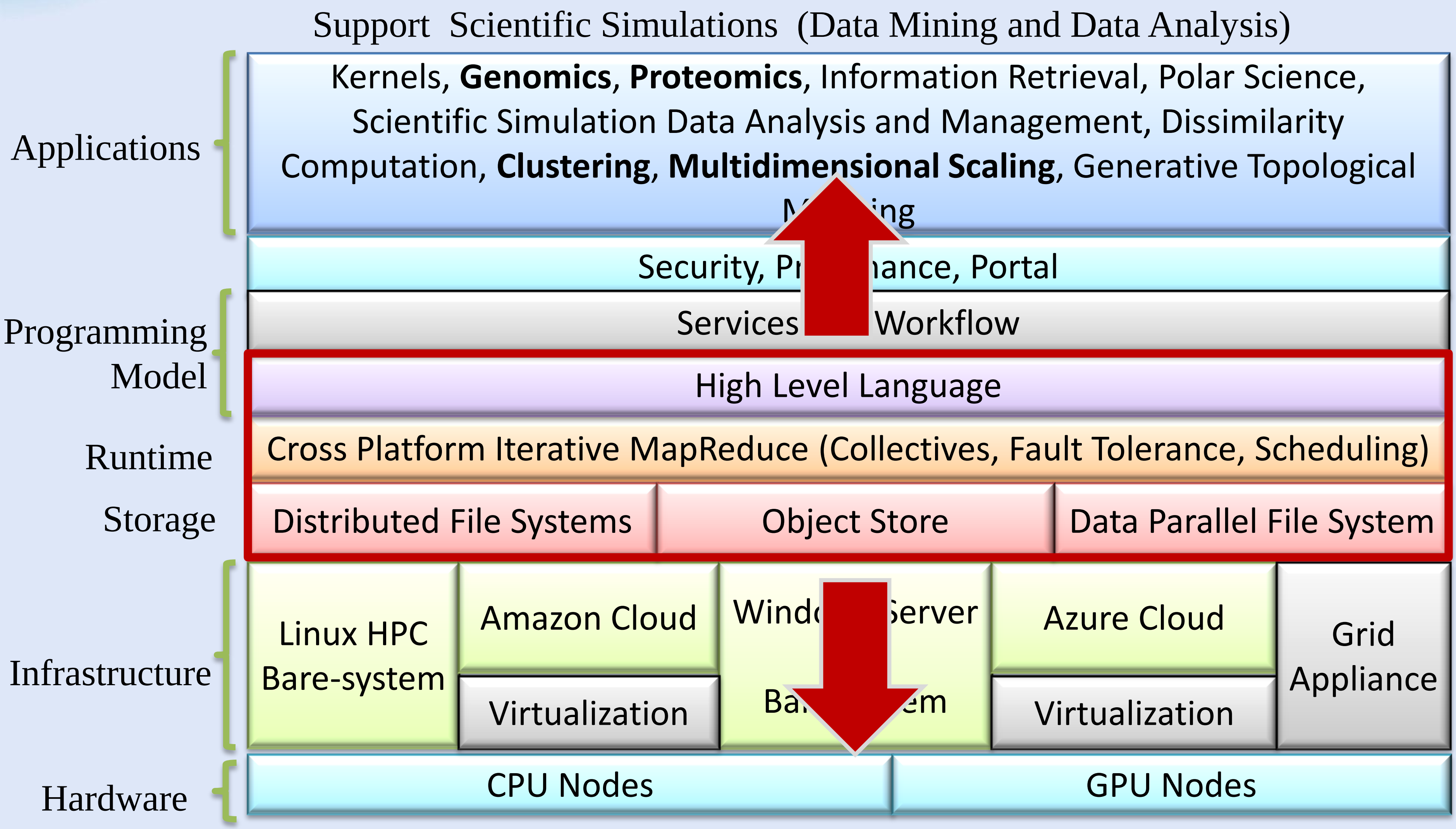






Intel’s Application Stack

(Iterative) MapReduce in Context



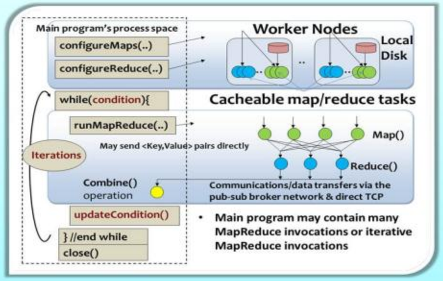
SALSA

Iterative MapReduce Enabling HPC-Cloud Interoperability

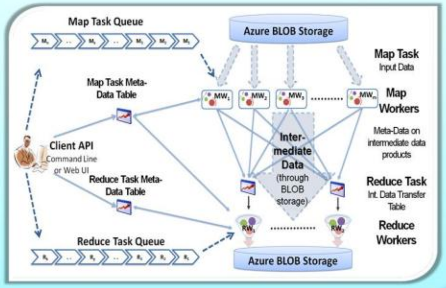
Scientific Applications

Iterative MapReduce Runtime

Twister
Cross Platform Iterative MapReduce



Twister4Azure
Iterative MapReduce for Azure Cloud



Large Scale Infrastructure

HPC Cluster

Windows Azure

amazon web services

What are the challenges?

These challenges must be met for both ***computation and storage****. If computation and storage are separated, it's not possible to bring computing to data.*

Providing both cost effective and powerful parallel programming paradigms that is capable of handling the incredible increases in dataset sizes.
(large-scale data analysis for Data Intensive applications)

Research issues

- portability between HPC and Cloud systems
- scaling performance
- fault tolerance

Data locality

- its impact on performance;
- the factors that affect data locality;
- the maximum degree of data locality that can be achieved.

Factors beyond data locality to improve performance

To achieve the best data locality is not always the optimal scheduling decision. For instance, if the node where input data of a task are stored is overloaded, to run the task on it will result in performance degradation.

Task granularity and load balance

In MapReduce , task granularity is fixed.
This mechanism has two drawbacks

1) limited degree of concurrency
2) load unbalancing resulting from the variation of task execution time.

Data Center vs Supercomputers

Scale

- Blue Waters = 40K 8-core "servers"
- Road Runner = 13K cell + 6K AMD servers
- MS Chicago Data Center = 50 containers = 100K 8-core servers.

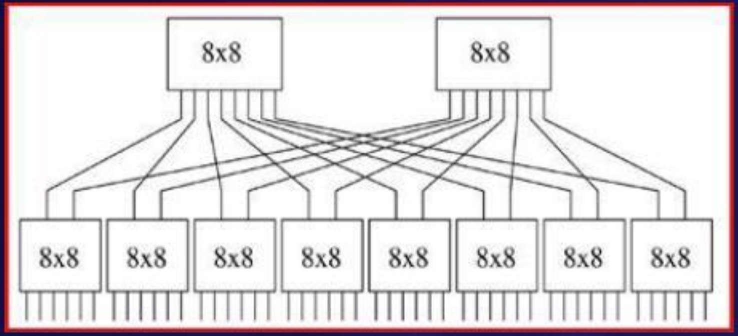
Network Architecture

- Supercomputers: CLOS "Fat Tree" infiniband
  - Low latency – high bandwidth
  - protocols
- Data Center: IP based
  - Optimized for Internet Access

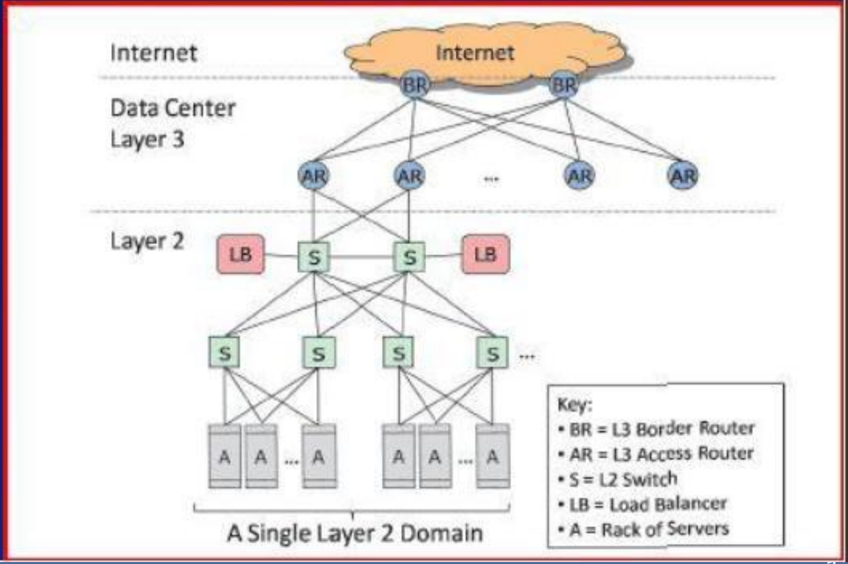
Data Storage

- Supers: separate data farm
  - GPFS or other parallel file system
- DCs: use disk on node + memcache

Fat tree network

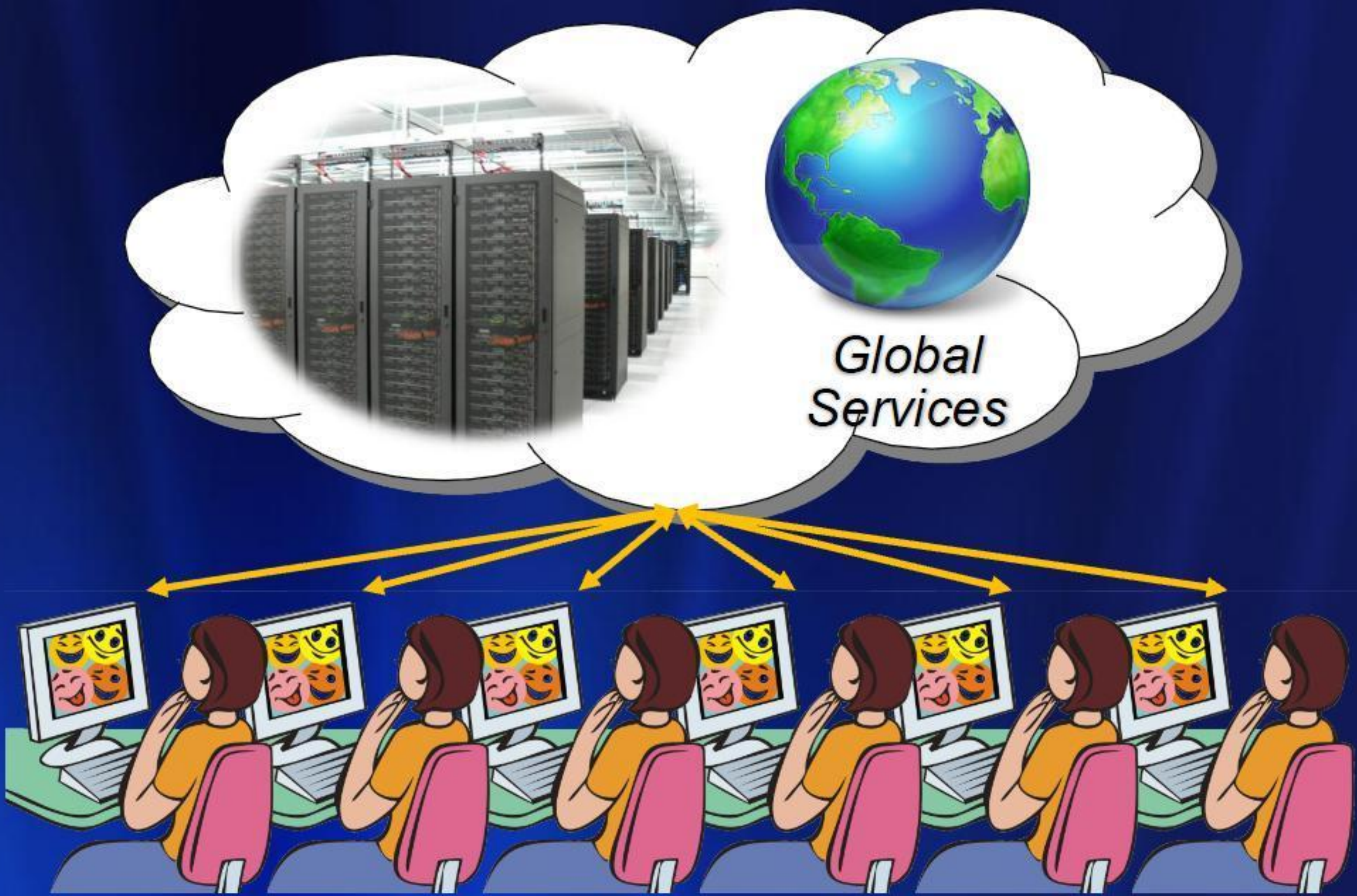


Standard Data Center Network



MICROSOFT

New Software Architecture



Clouds hide Complexity

Cyberinfrastructure
Is "Research as a Service"

SaaS: Software as a Service

(e.g. Clustering is a service)

PaaS: Platform as a Service

IaaS plus core software capabilities on which you build SaaS

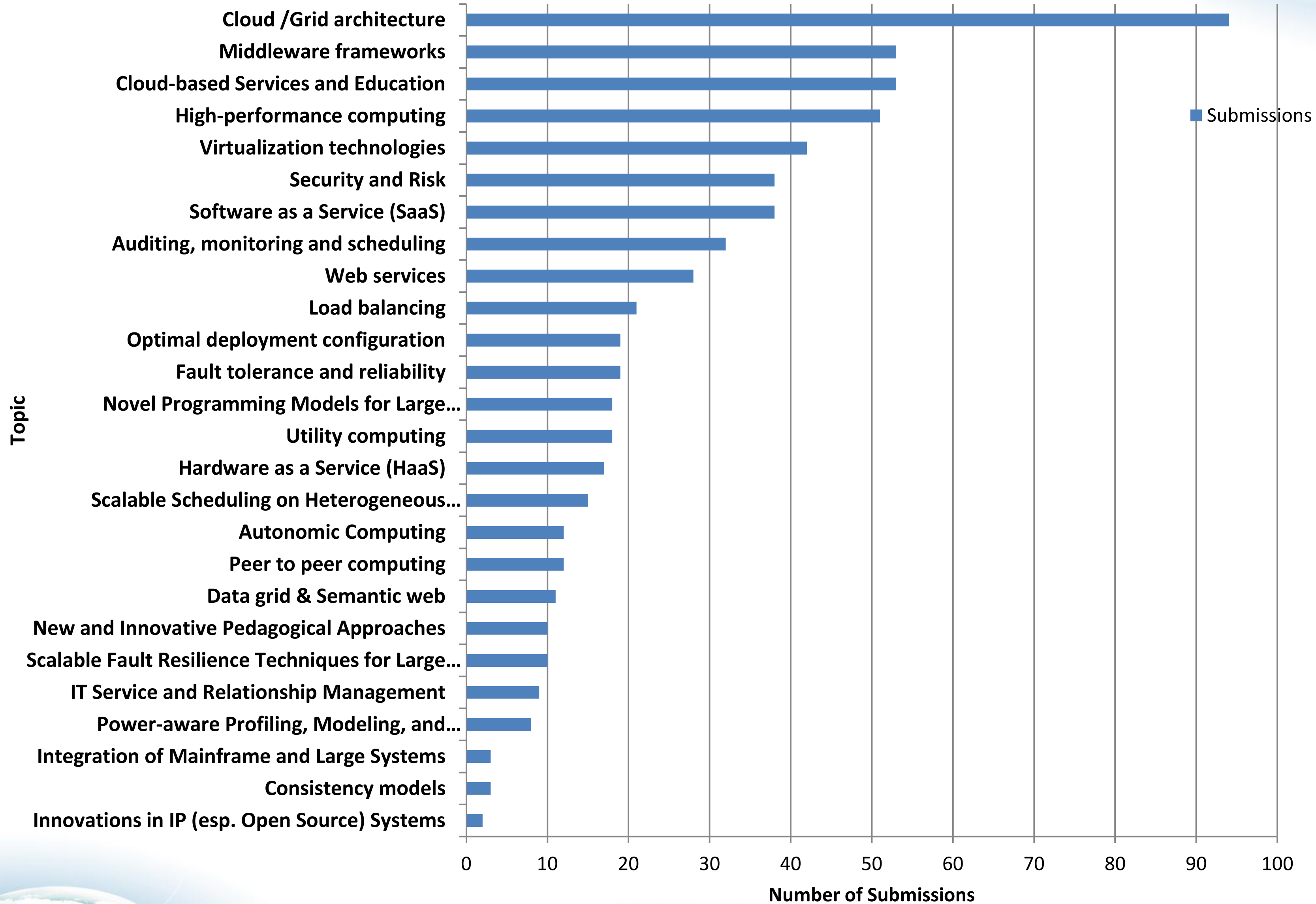
(e.g. Azure is a PaaS; MapReduce is a Platform)

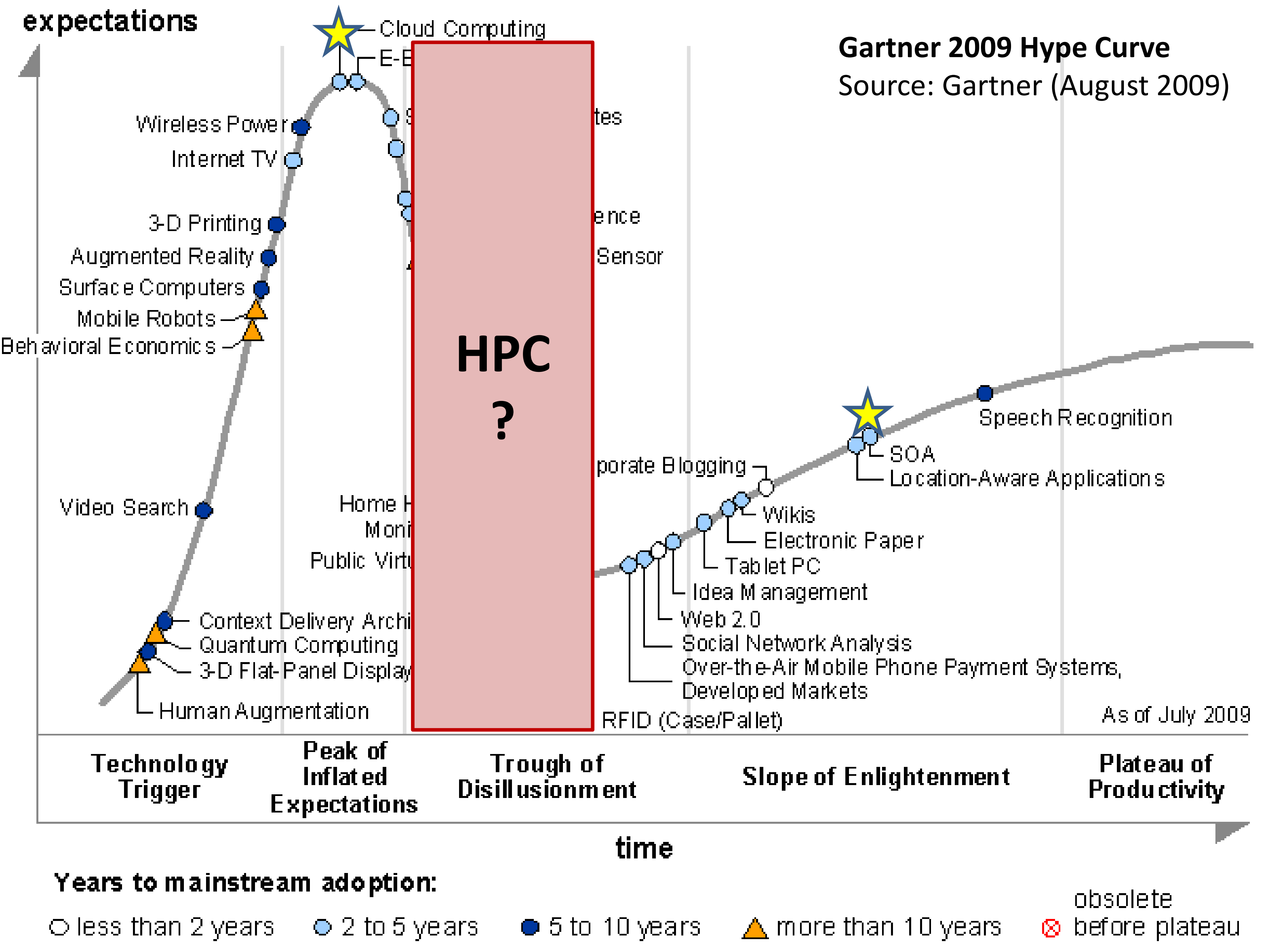
IaaS (**HaaS**): Infrasturcture as a Service

(get computer time with a credit card and with a Web interface like EC2)

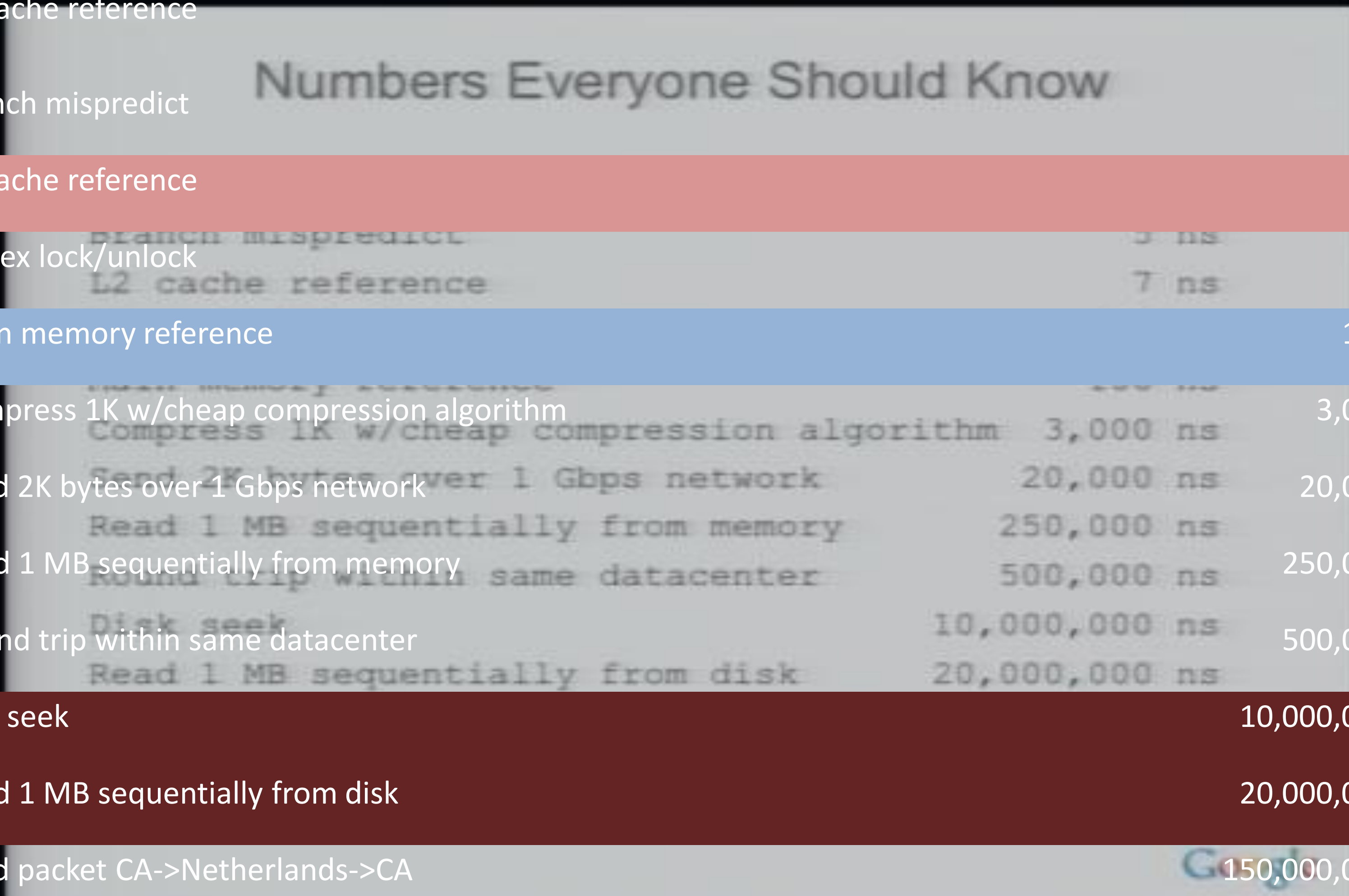
Cloud Computing 2010

Poster and Demo List





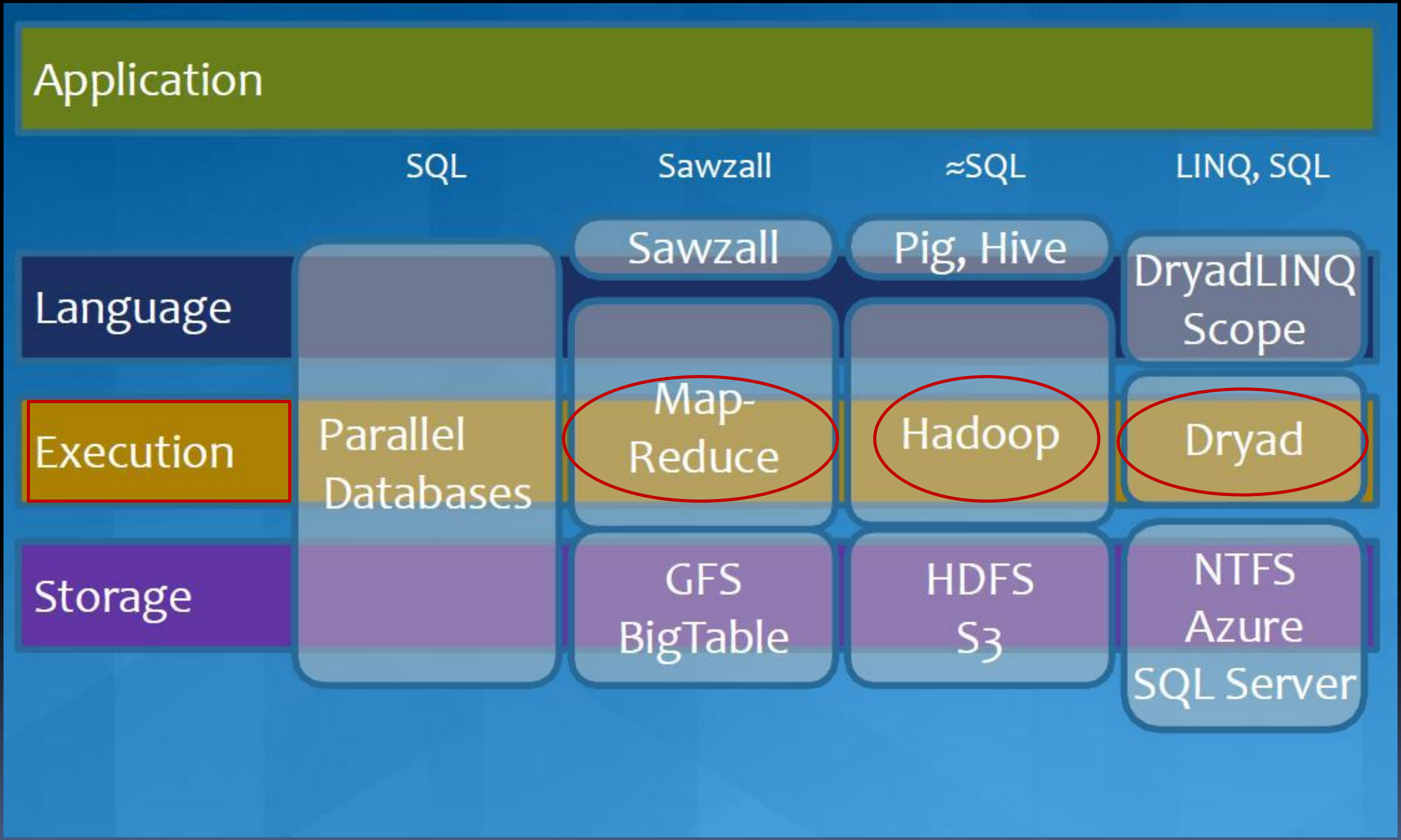
Gartner 2009 Hype Curve
Source: Gartner (August 2009)
expectations
Cloud Computing
E-B
tes
ence
Sensor
Wireless Power
Internet TV
3-D Printing
Augmented Reality
Surface Computers
Mobile Robots
Behavioral Economics
Video Search
Home H
Moni
Public Virtu
Context Delivery Archi
Quantum Computing
3-D Flat-Panel Display
Human Augmentation
HPC
?
porate Blogging
Wikis
Electronic Paper
Tablet PC
Idea Management
Web 2.0
Social Network Analysis
Over-the-Air Mobile Phone Payment Systems,
Developed Markets
RFID (Case/Pallet)
SOA
Location-Aware Applications
Speech Recognition
As of July 2009
Technology Trigger
Peak of Inflated Expectations
Trough of Disillusionment
Slope of Enlightenment
Plateau of Productivity
time
Years to mainstream adoption:
less than 2 years
2 to 5 years
5 to 10 years
more than 10 years
obsolete before plateau



L1 cache reference	0.5 ns
Branch mispredict	5 ns
L2 cache reference	7 ns
Mutex lock/unlock	25 ns
Main memory reference	100 ns
Compress 1K w/cheap compression algorithm	3,000 ns
Send 2K bytes over 1 Gbps network	20,000 ns
Read 1 MB sequentially from memory	250,000 ns
Round trip within same datacenter	500,000 ns
Disk seek	10,000,000 ns
Read 1 MB sequentially from disk	20,000,000 ns
Send packet CA->Netherlands->CA	150,000,000 ns

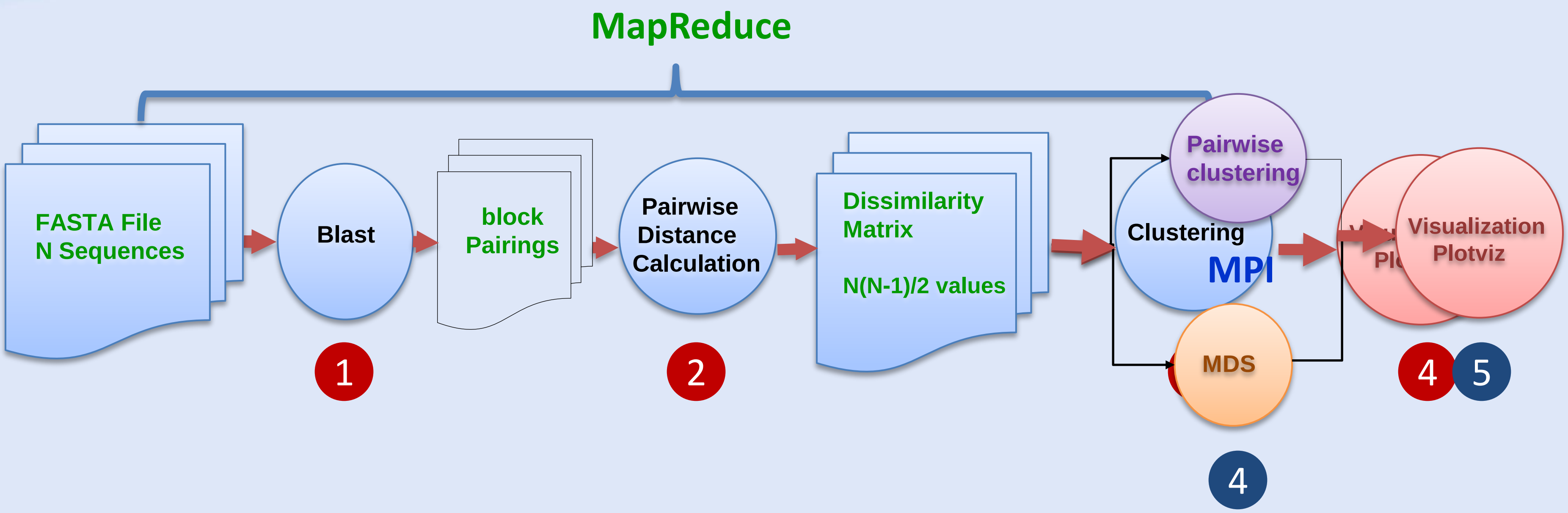
Programming Models and Tools

MapReduce in Heterogeneous Environment



MICROSOFT

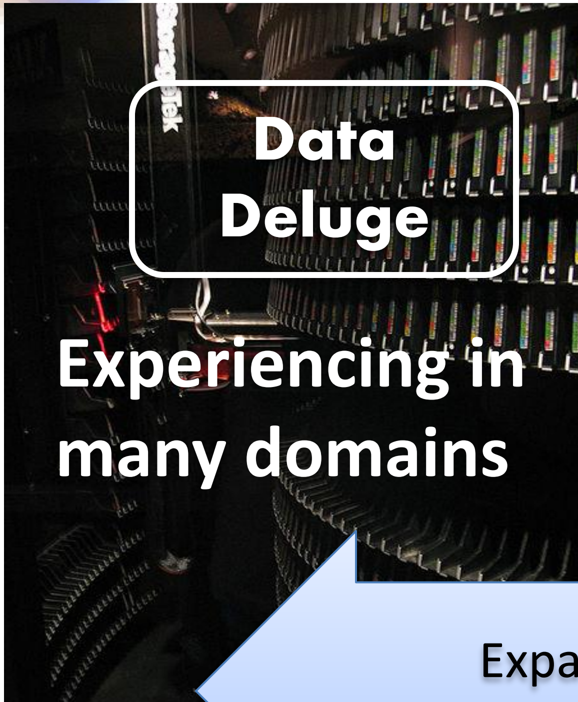
Next Generation Sequencing Pipeline on Cloud



- Users submit their jobs to the pipeline and the results will be shown in a visualization tool.
- This chart illustrate a hybrid model with MapReduce and MPI. Twister will be an unified solution for the pipeline mode.
- The components are services and so is the whole pipeline.
- We could research on which stages of pipeline services are suitable for private or commercial Clouds.

SALSA

Motivation



MapReduce

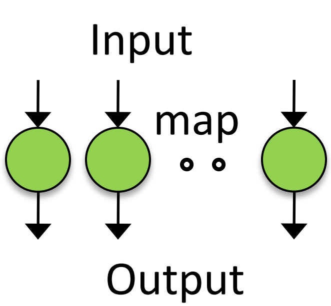
Data Centered, QoS

Classic Parallel Runtimes (MPI)

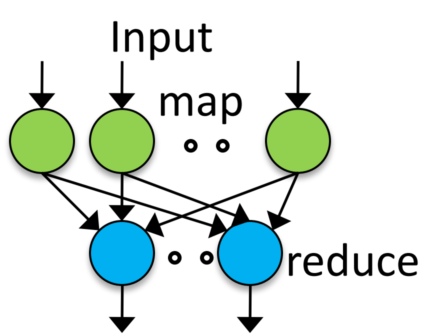
Efficient and Proven techniques

Expand the Applicability of MapReduce to more **classes** of Applications

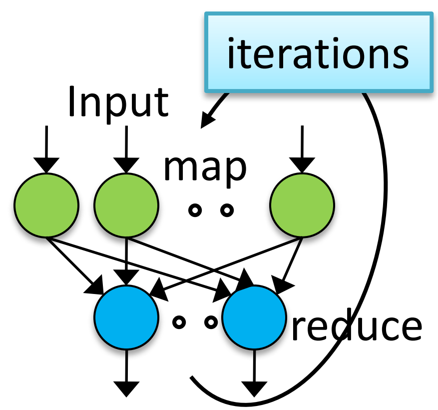
Map-Only



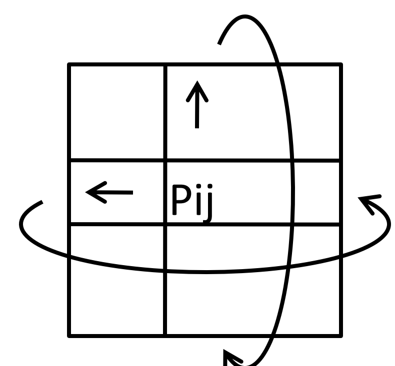
MapReduce



Iterative MapReduce



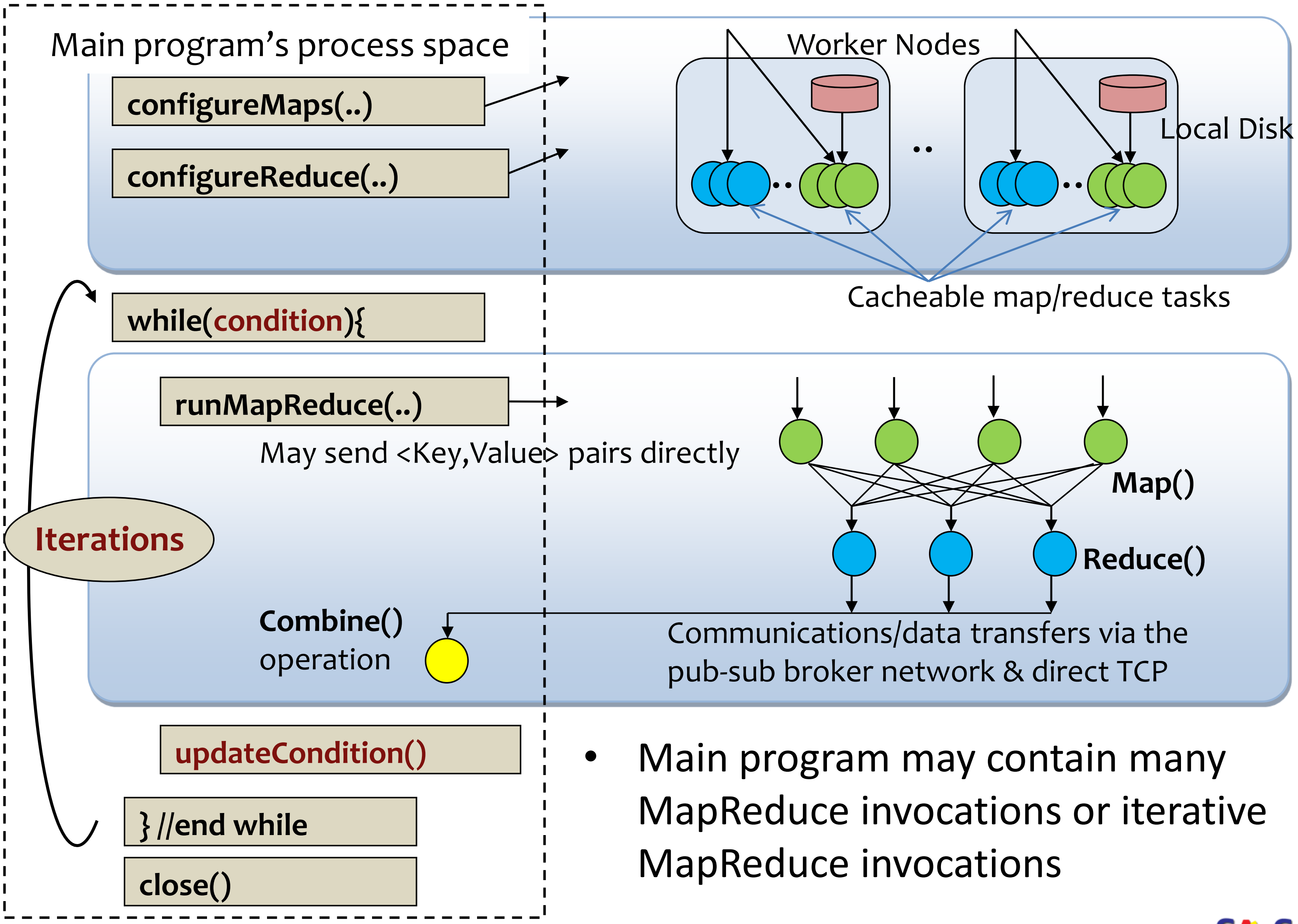
More Extensions



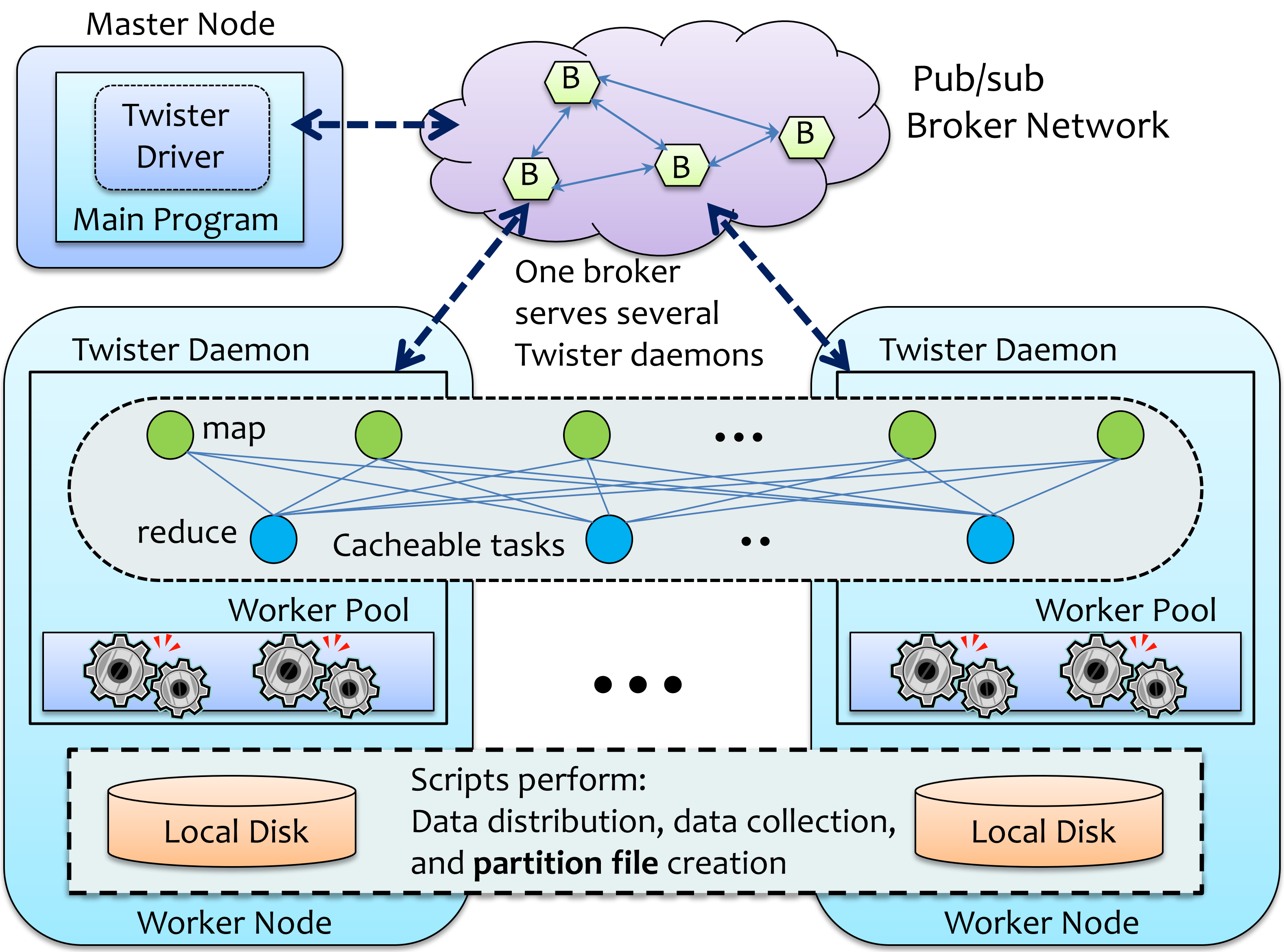
Twister v0.9

New Infrastructure for Iterative MapReduce Programming

- *Distinction on static and variable data*
- *Configurable long running (cacheable) map/reduce tasks*
- *Pub/sub messaging based communication/data transfers*
- *Broker Network for facilitating communication*



- Main program may contain many MapReduce invocations or iterative MapReduce invocations


Master Node
Twister Driver
Main Program
Pub/sub Broker Network
B
B
B
B
One broker serves several Twister daemons
Twister Daemon
Twister Daemon
map
reduce
Cacheable tasks
Worker Pool
Worker Pool
Scripts perform:
Data distribution, data collection, and **partition file** creation
Local Disk
Local Disk
Worker Node
Worker Node

Microsoft Research

Search Microsoft Research

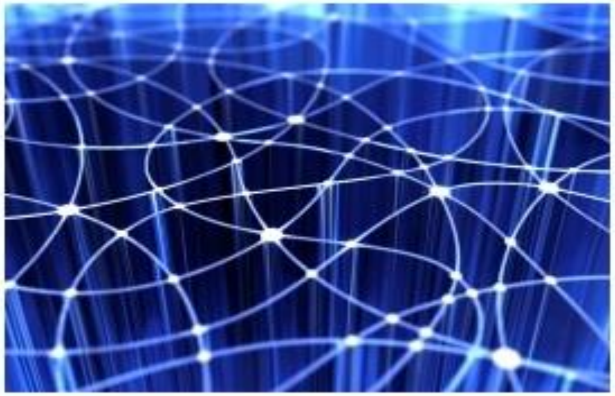
Home | Our Research | Connections | Careers

Worldwide Labs | Research Areas | Research Groups

> Projects > Daytona

Daytona

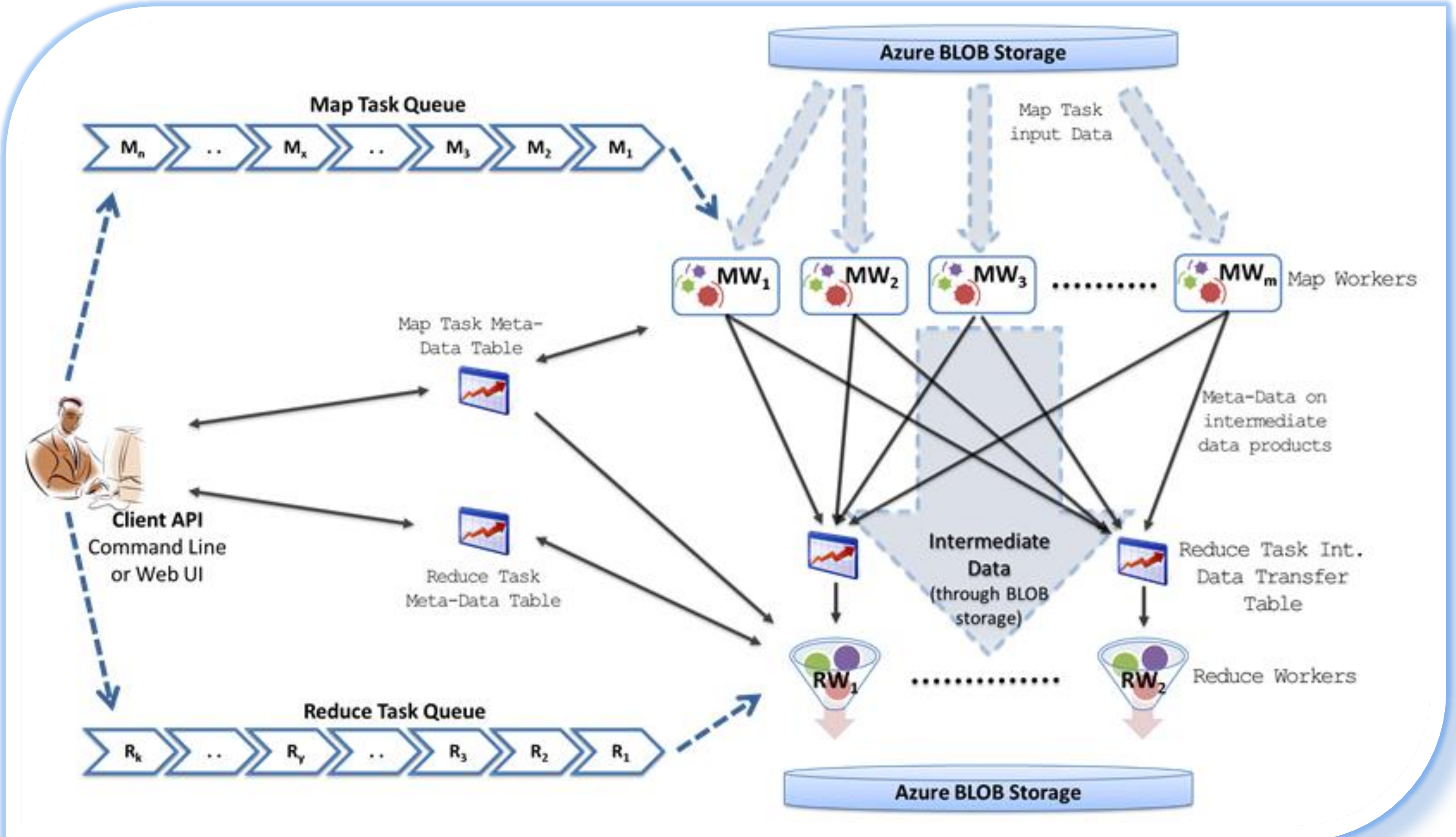
Iterative MapReduce on Windows Azure



Microsoft has developed an iterative MapReduce runtime for Windows Azure, code-named "Daytona." Project Daytona is designed to support a wide class of data analytics and machine learning algorithms. It can scale out to hundreds of server cores for analysis of distributed data.

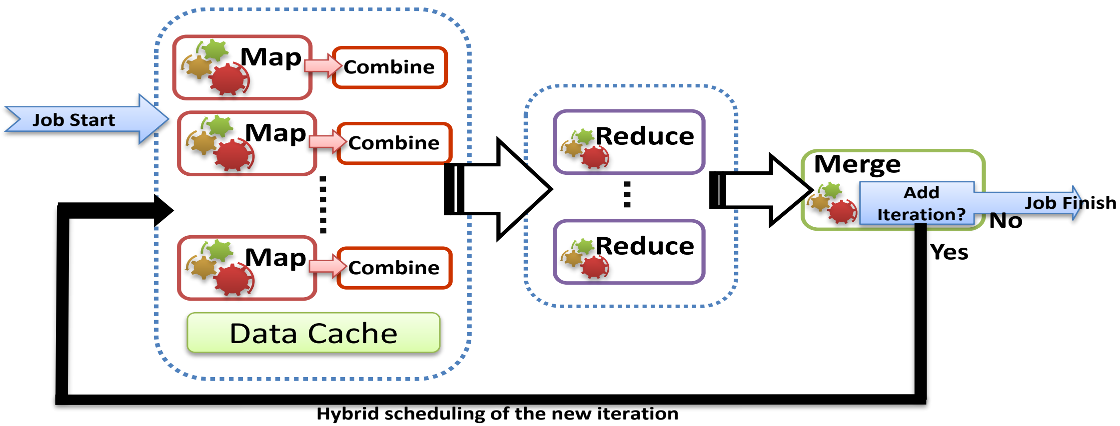
Project Daytona was developed as part of the eXtreme Computing Group's Cloud Research Engagement Initiative, and made its debut at the Microsoft Research Faculty Summit. One of the most common requests we have received from the community of researchers in our program is for a data analysis and processing framework. Increasingly, researchers in a wide range of domains—such as healthcare, education, and environmental science—have large and growing data collections and they need simple tools to help them find signals in their data and uncover insights. We are making the Project Daytona MapReduce Runtime for Windows Azure download freely available, along with sample codes and instructional materials that researchers can use to set up their own large-scale,

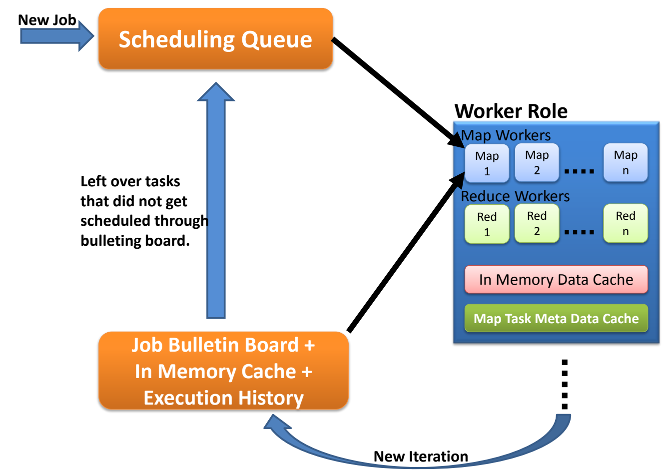
MRRoles4Azure



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

Iterative MapReduce for Azure



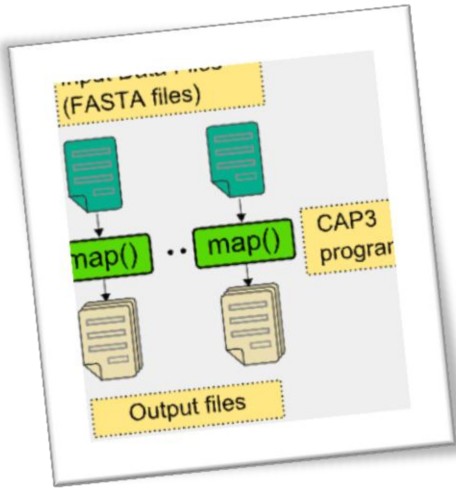


- Programming model extensions to support broadcast data
- Merge Step
- In-Memory Caching of static data
- Cache aware hybrid scheduling using Queues, bulletin board (special table) and execution histories
- Hybrid intermediate data transfer

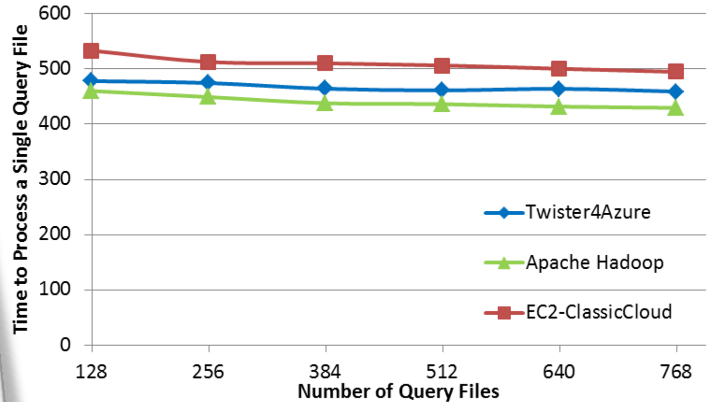
MRRoles4Azure

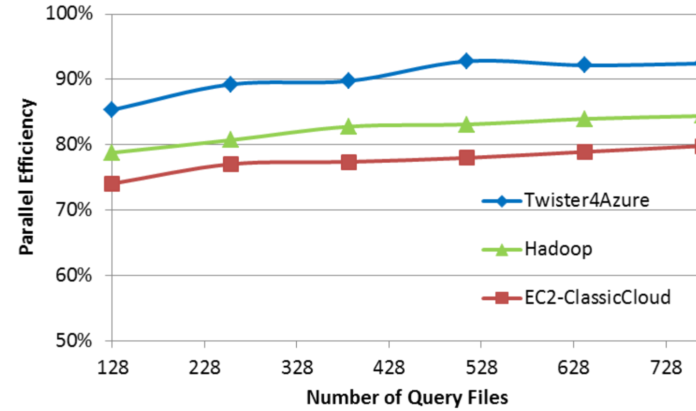
- Distributed, highly scalable and highly available cloud services as the building blocks.
- Utilize eventually-consistent , high-latency cloud services effectively to deliver performance comparable to traditional MapReduce runtimes.
- Decentralized architecture with global queue based dynamic task scheduling
- Minimal management and maintenance overhead
- Supports dynamically scaling up and down of the compute resources.
- MapReduce fault tolerance

Performance Comparisons

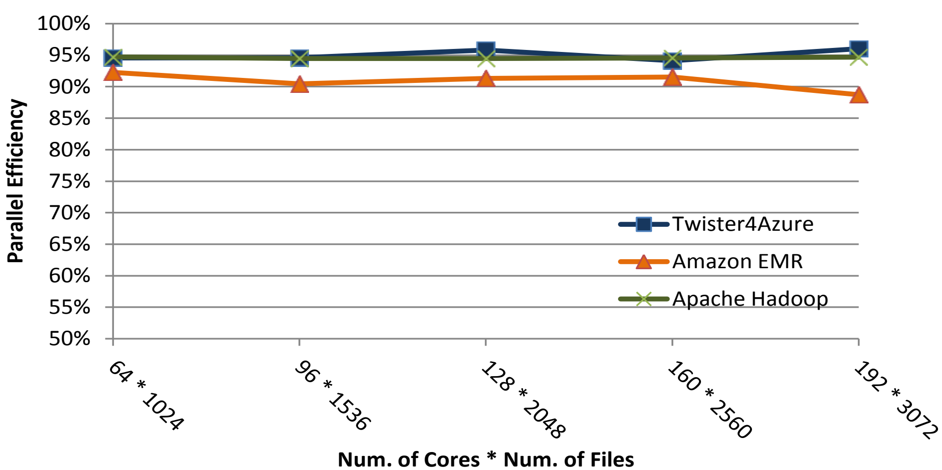


BLAST Sequence Search

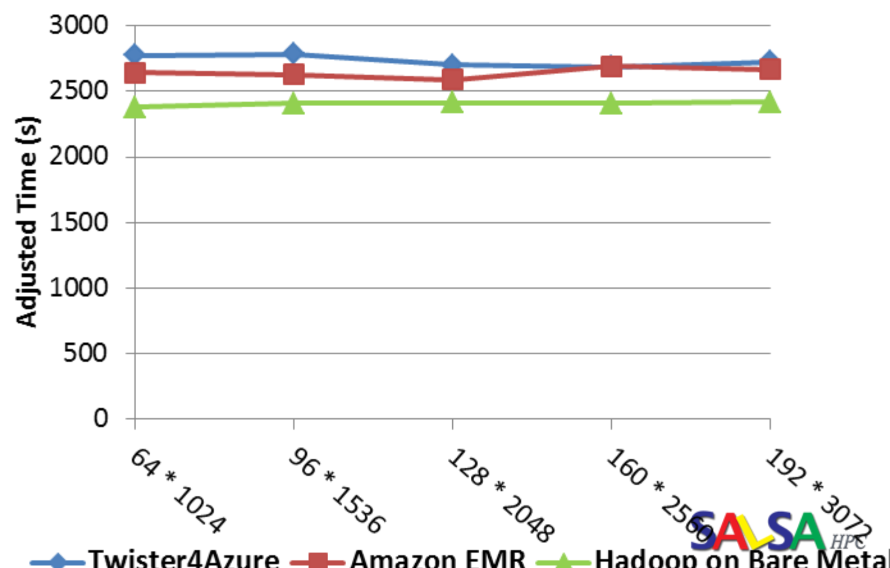




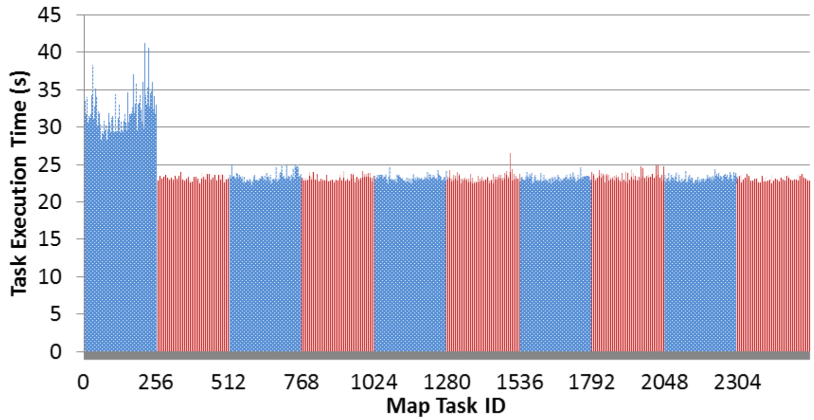
Cap3 Sequence Assembly



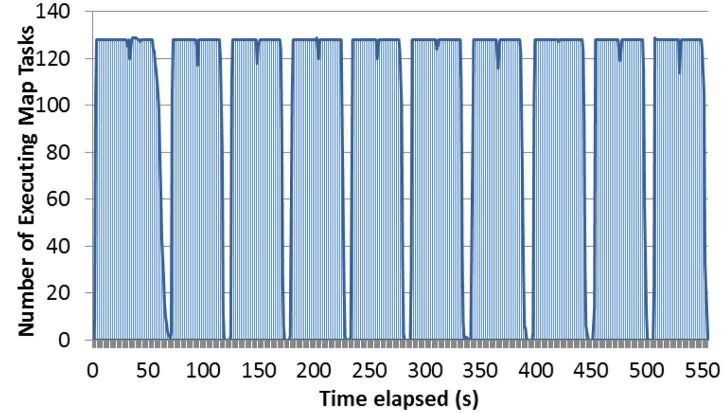
Smith Waterman Sequence Alignment



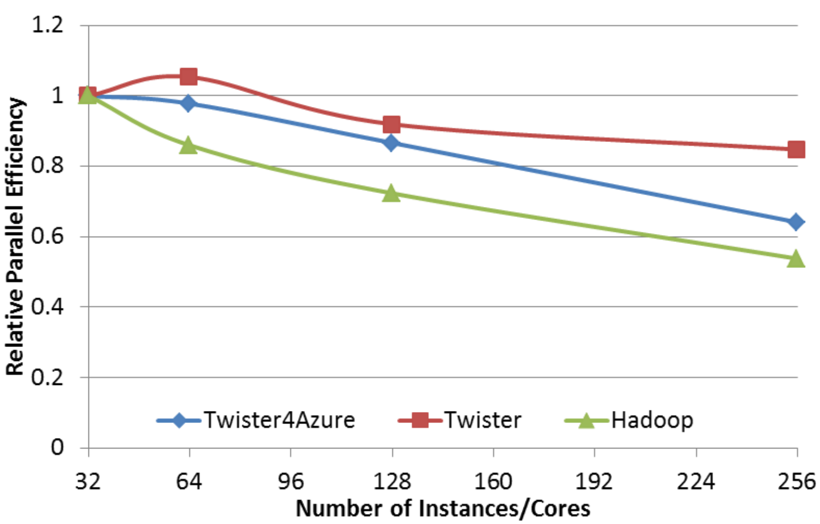
Performance – Kmeans Clustering



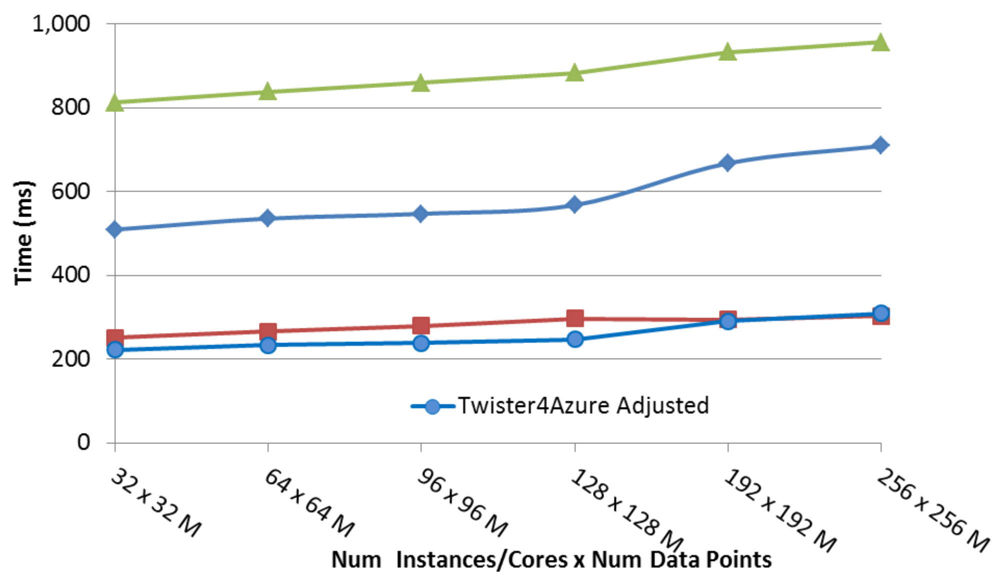
Task Execution Time Histogram



Number of Executing Map Task Histogram

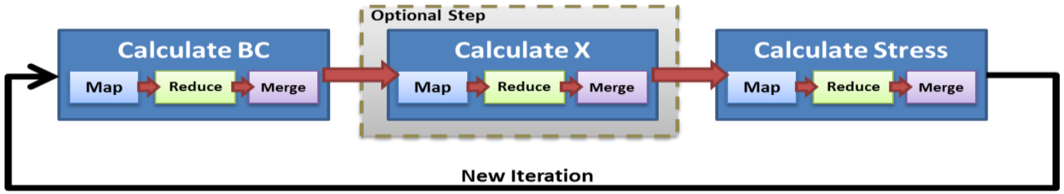


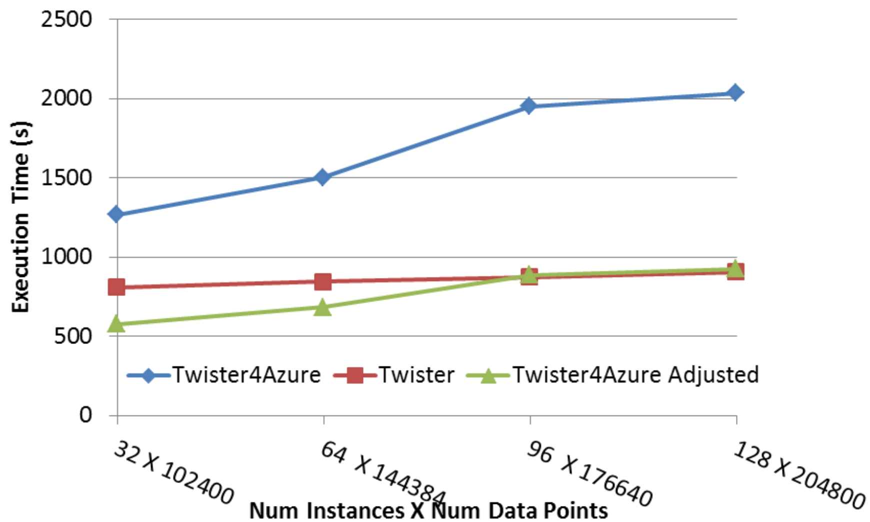
Strong Scaling with 128M Data Points



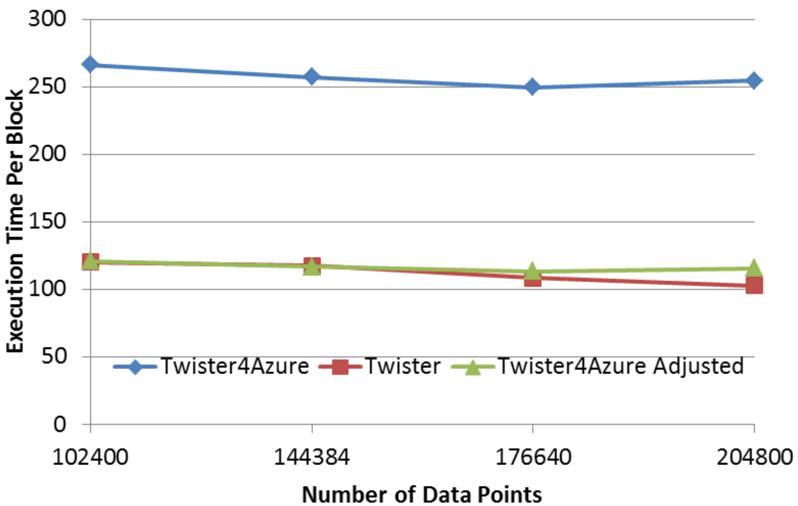
Weak Scaling

Performance – Multi Dimensional Scaling

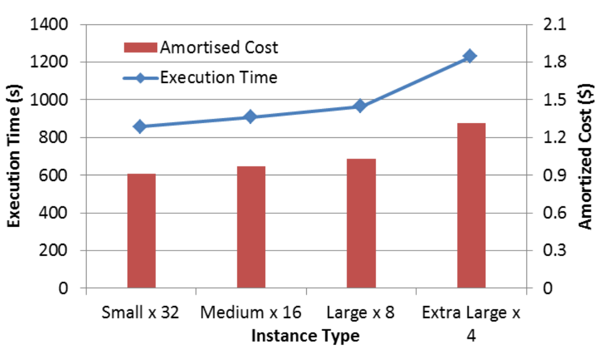




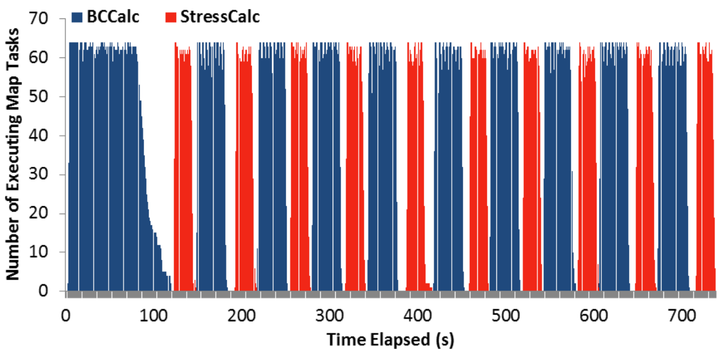
Weak Scaling



Data Size Scaling



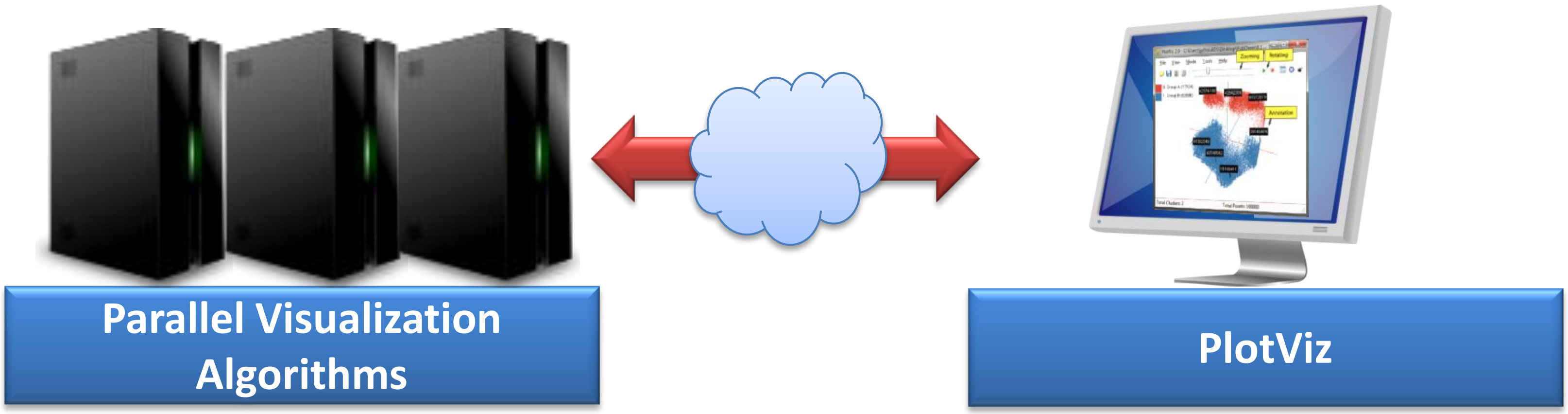
Azure Instance Type Study



Number of Executing Map Task Histogram

SALSA HPC

PlotViz, Visualization System



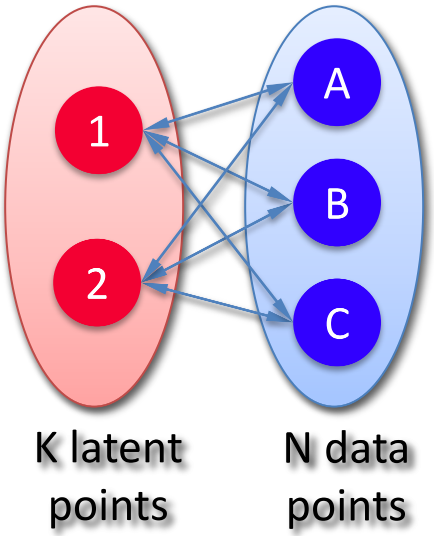
- Parallel visualization algorithms (GTM, MDS, …)
- Improved quality by using DA optimization
- Interpolation
- Twister Integration (Twister-MDS, Twister-LDA)

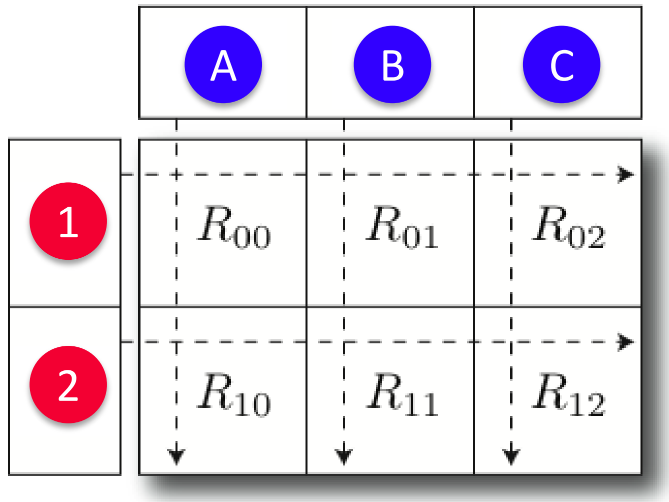
- Provide Virtual 3D space
- Cross-platform
- Visualization Toolkit (VTK)
- Qt framework

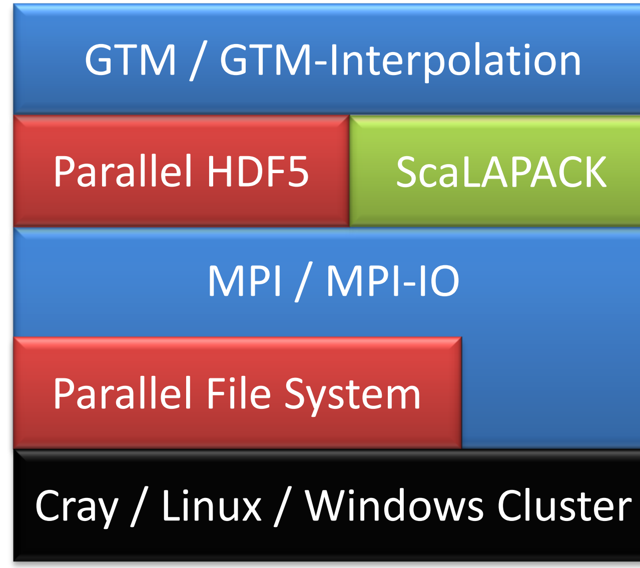
GTM vs. MDS

	GTM	MDS (SMACOF)
Purpose	• Non-linear dimension reduction • Find an optimal configuration in a lower-dimension • Iterative optimization method	
Input	Vector-based data	Non-vector (Pairwise similarity matrix)
Objective Function	Maximize Log-Likelihood	Minimize STRESS or SSTRESS
Complexity	O(KN) (K << N)	$O(N^2)$
Optimization Method	EM	Iterative Majorization (EM-like)

Parallel GTM







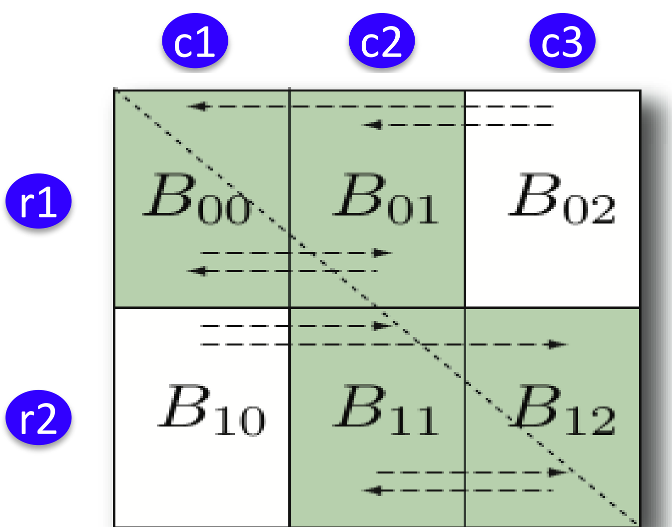
GTM SOFTWARE STACK

- Finding K clusters for N data points
  - Relationship is a bipartite graph (bi-graph)
  - Represented by K-by-N matrix (K << N)
- Decomposition for P-by-Q compute grid
  - Reduce memory requirement by 1/PQ

Scalable MDS

Parallel MDS

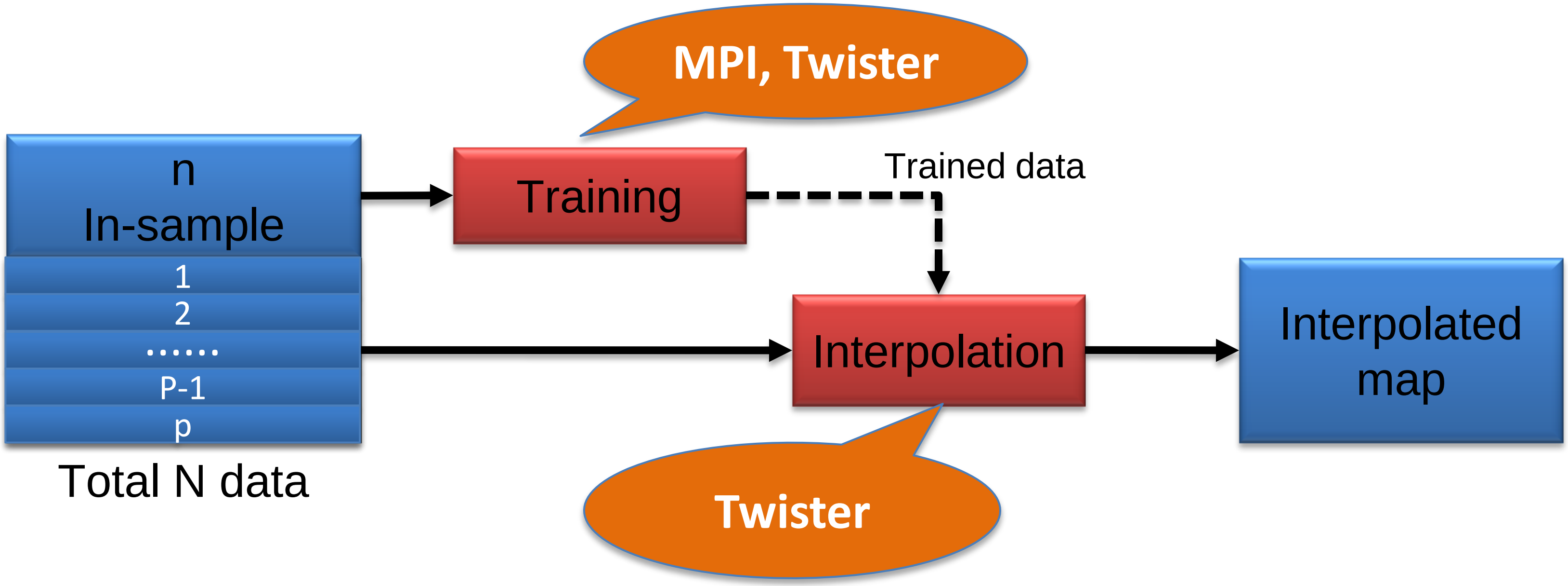
- $O(N^2)$ memory and computation required.
  - 100k data → 480GB memory
- Balanced decomposition of NxN matrices by P-by-Q grid.
  - Reduce memory and computing requirement by 1/PQ
- Communicate via MPI primitives



MDS Interpolation

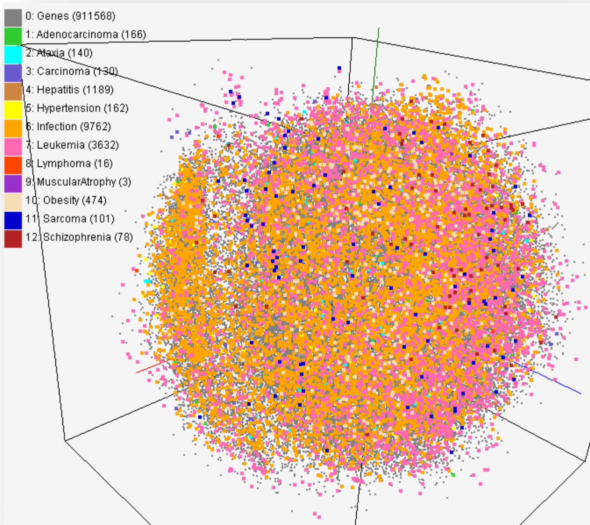
- Finding approximate mapping position w.r.t. k-NN's prior mapping.
- Per point it requires:
  - O(M) memory
  - O(k) computation
- Pleasingly parallel
- Mapping 2M in 1450 sec.
  - vs. 100k in 27000 sec.
  - 7500 times faster than estimation of the full MDS.

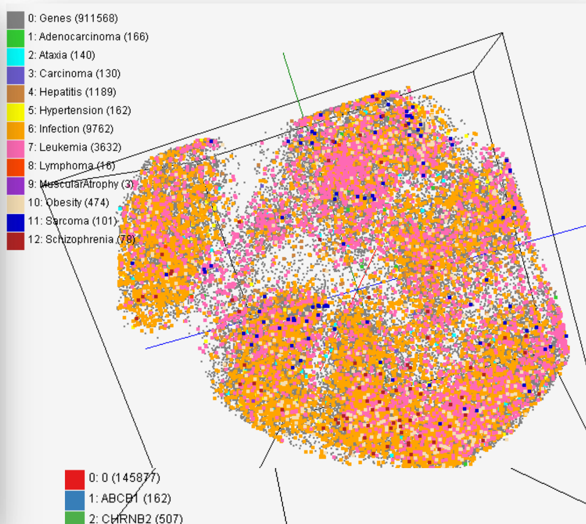
Interpolation extension to GTM/MDS



- Full data processing by GTM or MDS is computing- and memory-intensive
- Two step procedure
  - *Training* : training by M samples out of N data
  - *Interpolation* : remaining (N-M) out-of-samples are approximated without training

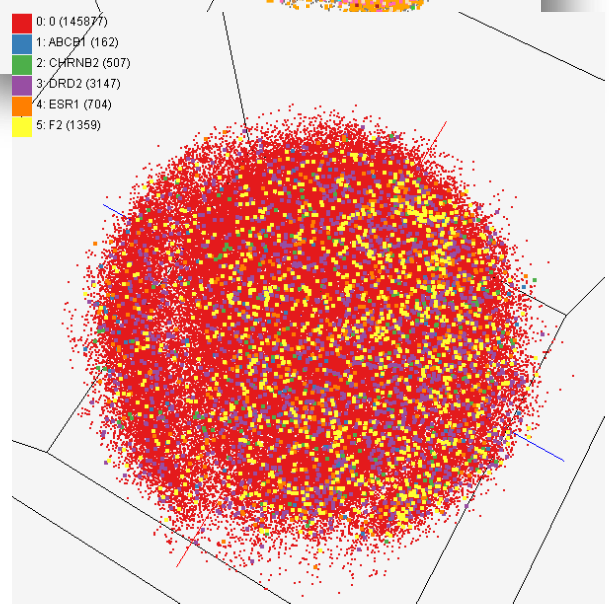
GTM/MDS Applications

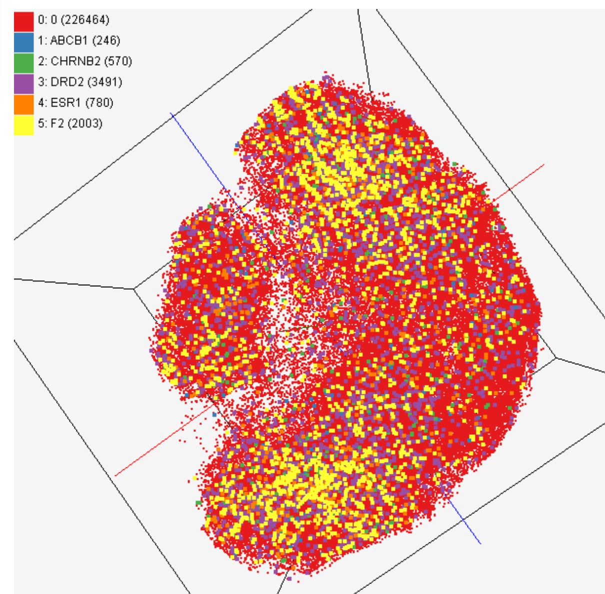




PubChem data with CTD visualization by using MDS (left) and GTM (right)

About 930,000 chemical compounds are visualized as a point in 3D space, annotated by the related genes in Comparative Toxicogenomics Database (CTD)





Chemical compounds shown in literatures, visualized by MDS (left) and GTM (right)

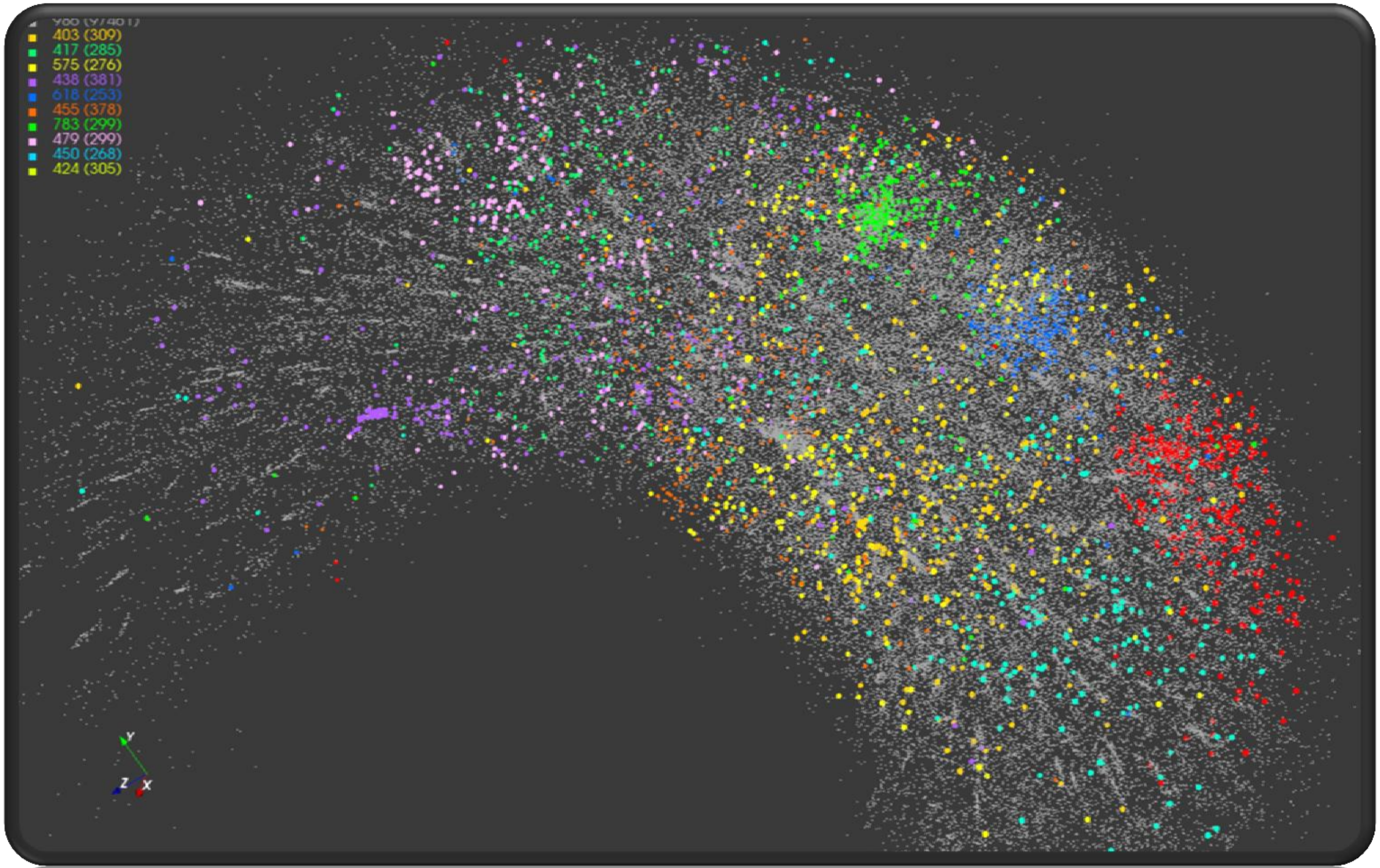
Visualized 234,000 chemical compounds which may be related with a set of 5 genes of interest (ABCB1, CHRNB2, DRD2, ESR1, and F2) based on the dataset collected from major journal literatures which is also stored in Chem2Bio2RDF system.

SALSA HPC

Twister-MDS Demo

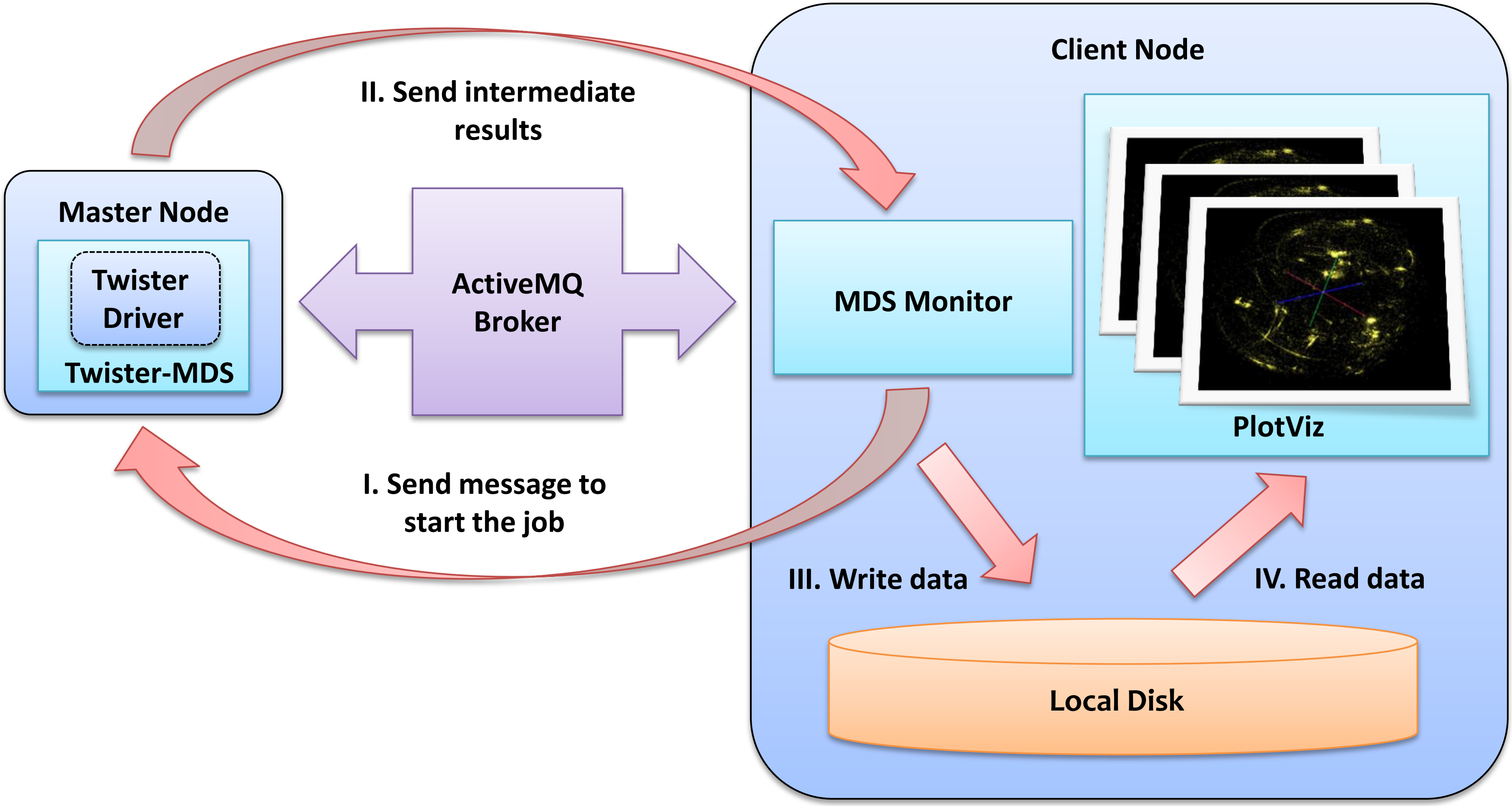
- This demo is for real time visualization of the process of multidimensional scaling(MDS) calculation.
- We use Twister to do parallel calculation inside the cluster, and use PlotViz to show the intermediate results at the user client computer.
- The process of computation and monitoring is automated by the program.

Twister-MDS Output

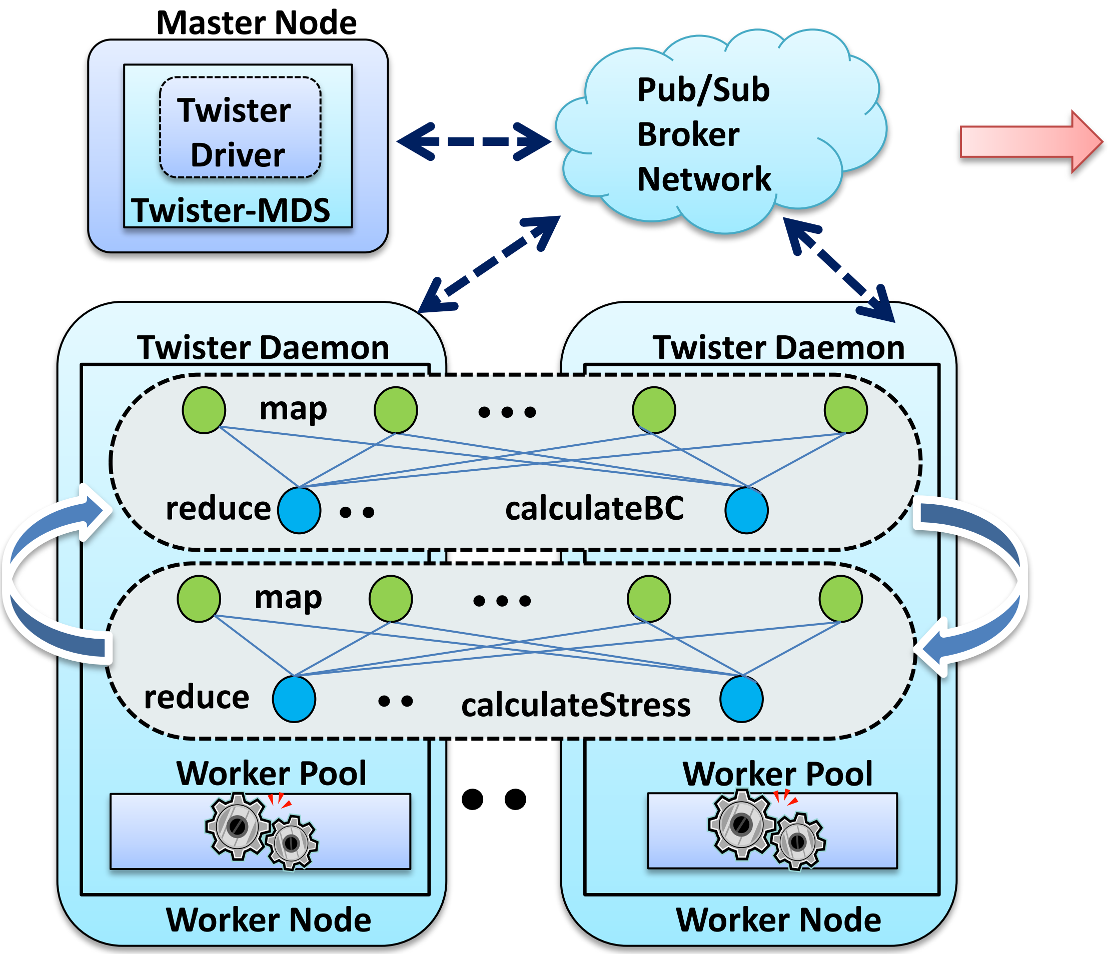


MDS projection of 100,000 protein sequences showing a few experimentally identified clusters in preliminary work with Seattle Children's Research Institute

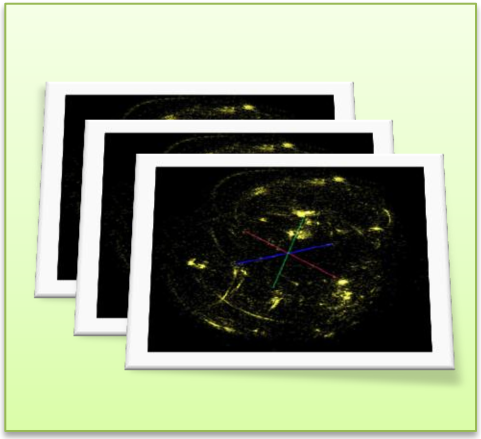
Twister-MDS Work Flow



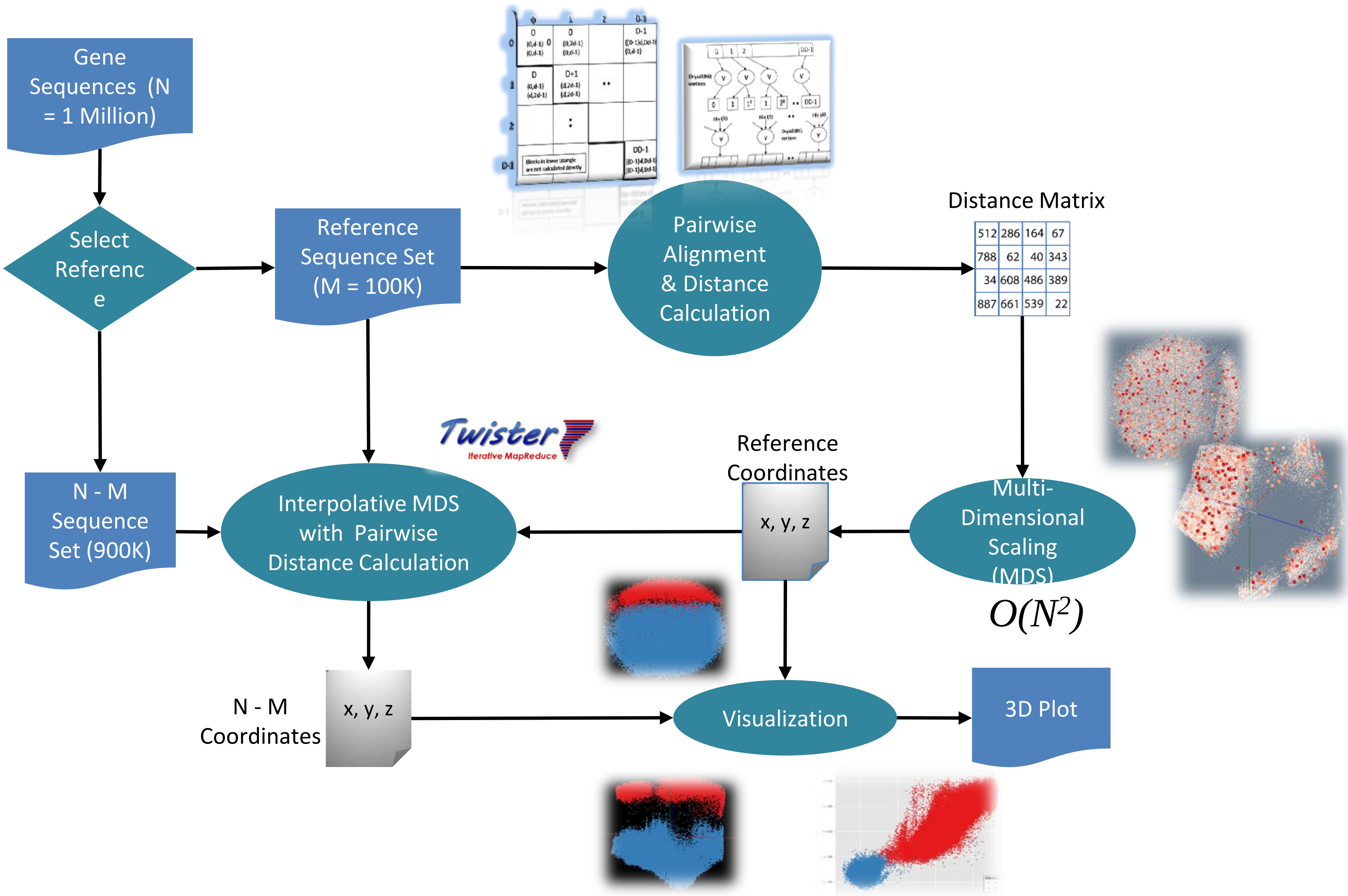
Twister-MDS Structure



MDS Output Monitoring Interface



Bioinformatics Pipeline

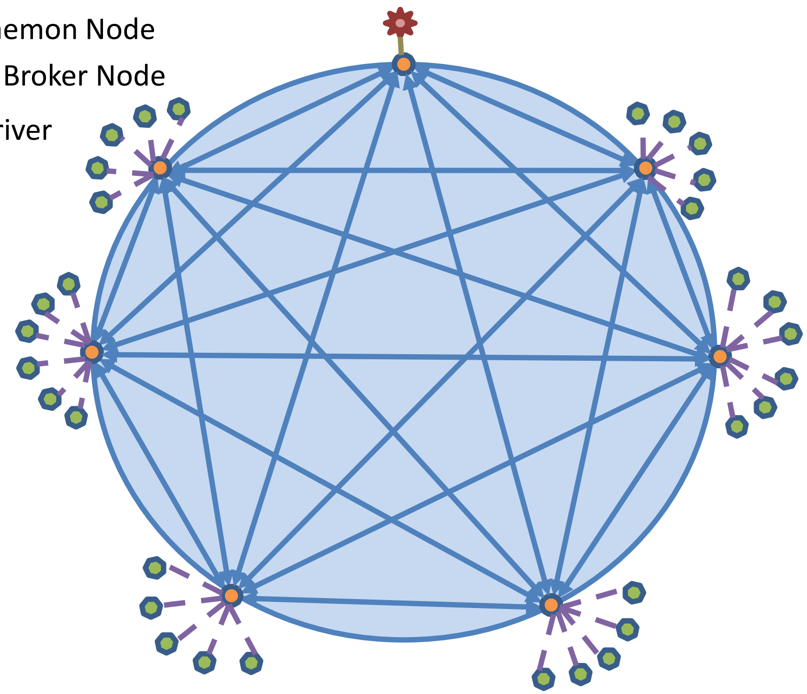


New Network of Brokers

Twister Daemon Node

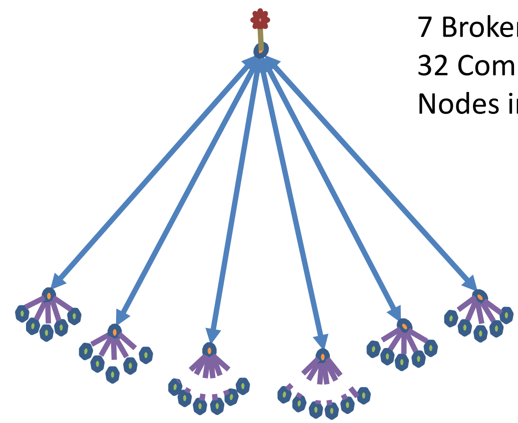
ActiveMQ Broker Node

Twister Driver Node

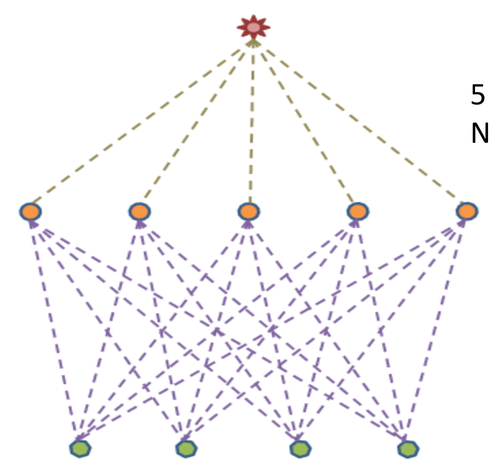


A. Full Mesh Network

B. Hierarchical Sending



7 Brokers and 32 Computing Nodes in total



5 Brokers and 4 Computing Nodes in total

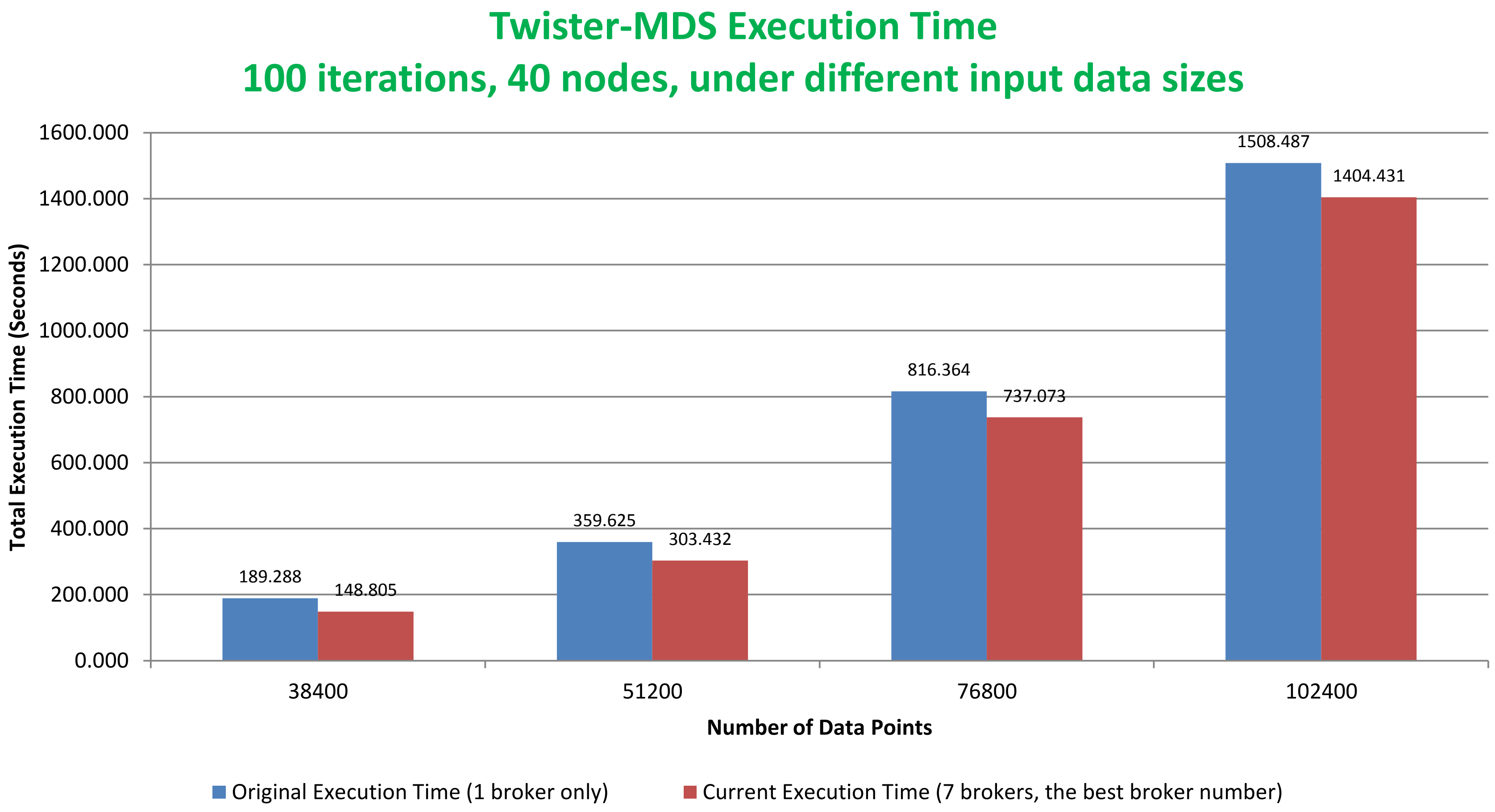
C. Streaming

Broker-Driver Connection

Broker-Daemon Connection

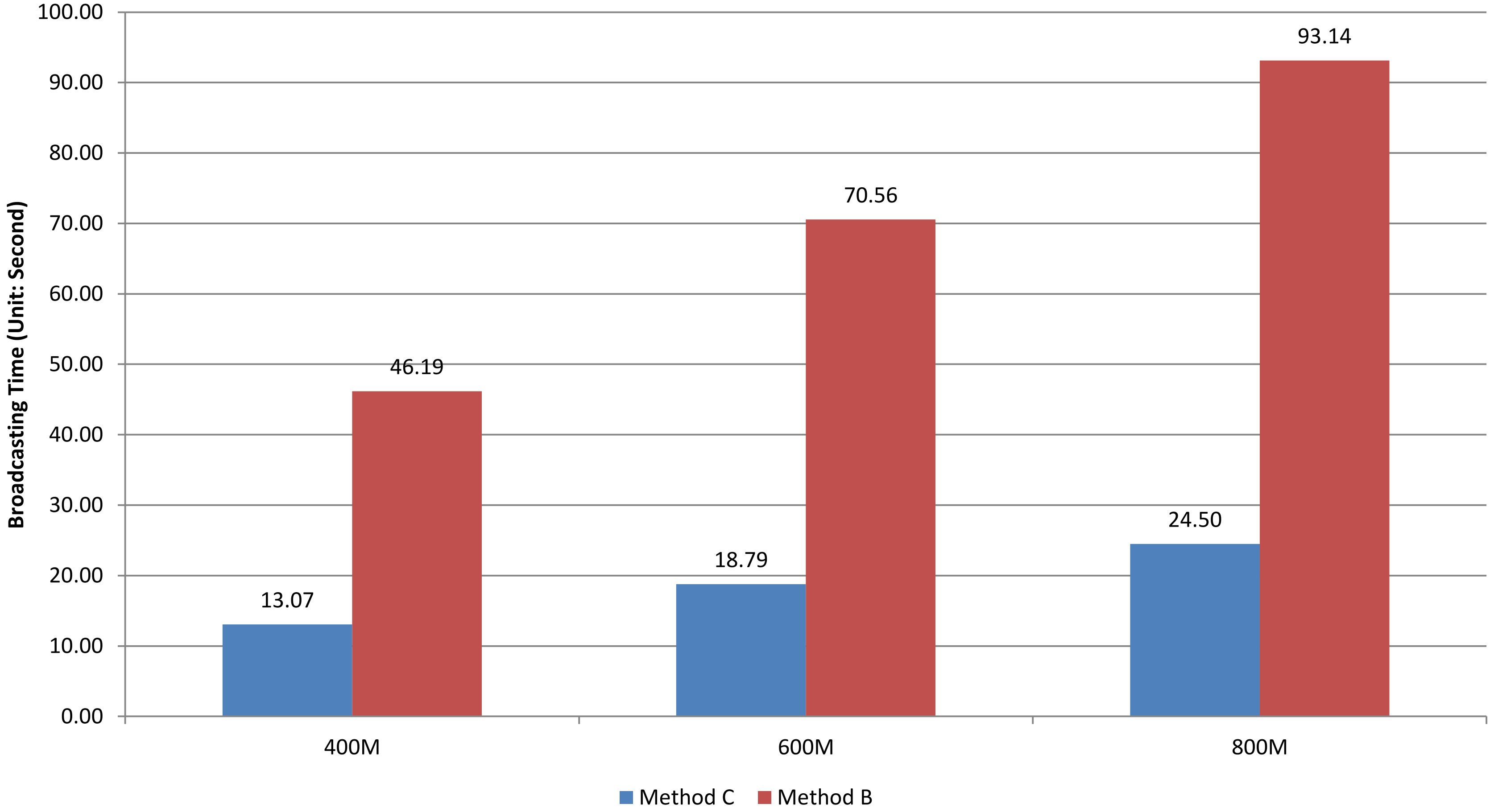
Broker-Broker Connection

Performance Improvement

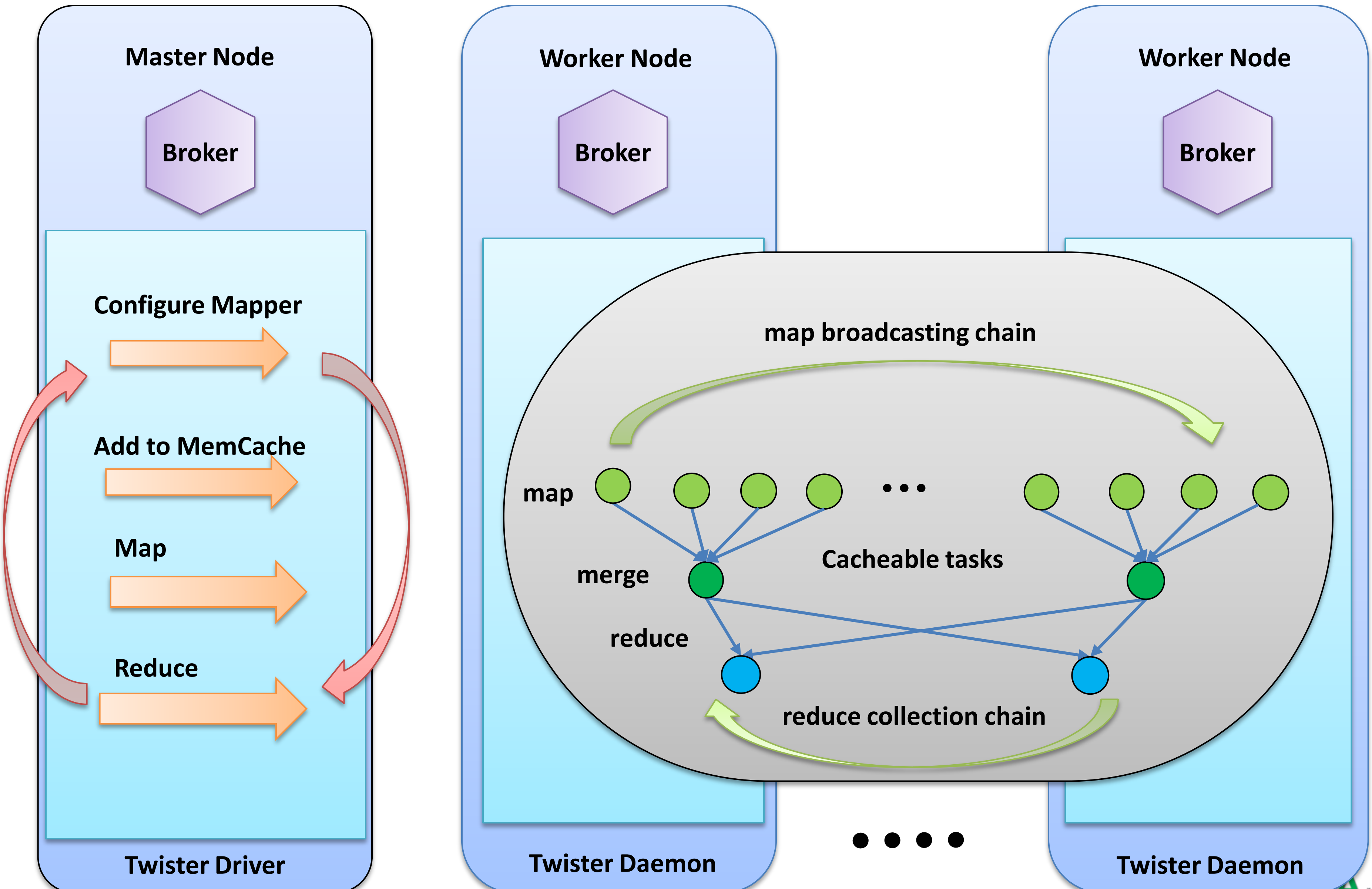


Broadcasting on 40 Nodes

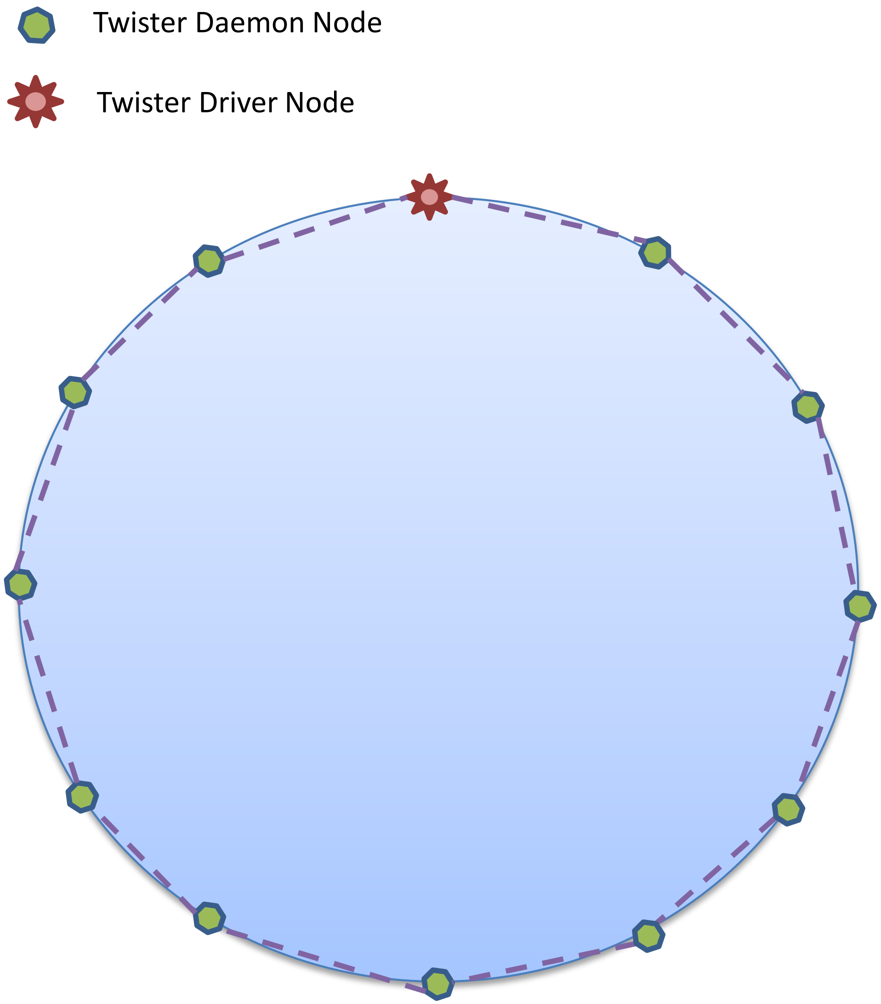
(In Method C, centroids are split to 160 blocks, sent through 40 brokers in 4 rounds)



Twister New Architecture



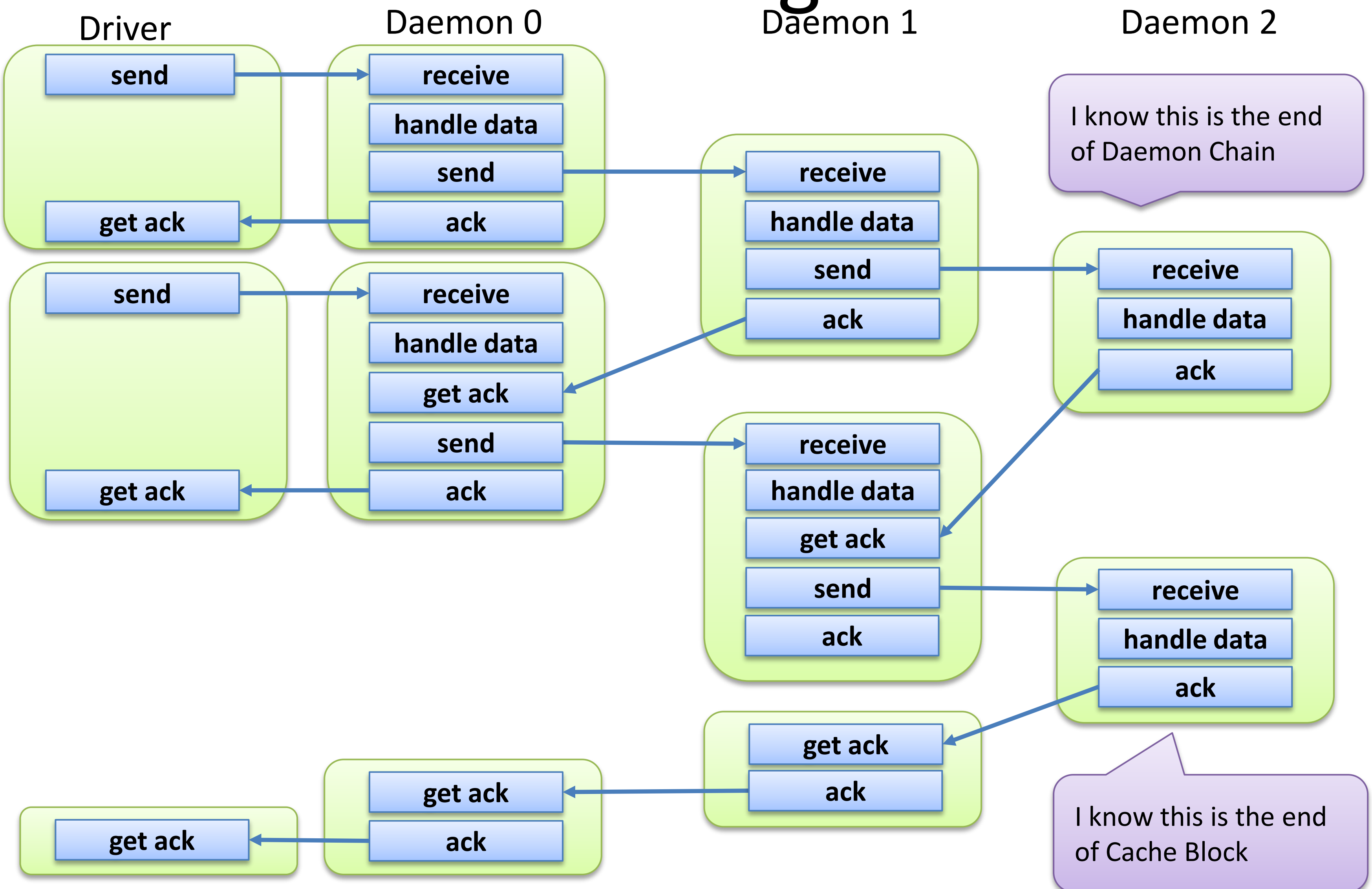
Chain/Ring Broadcasting



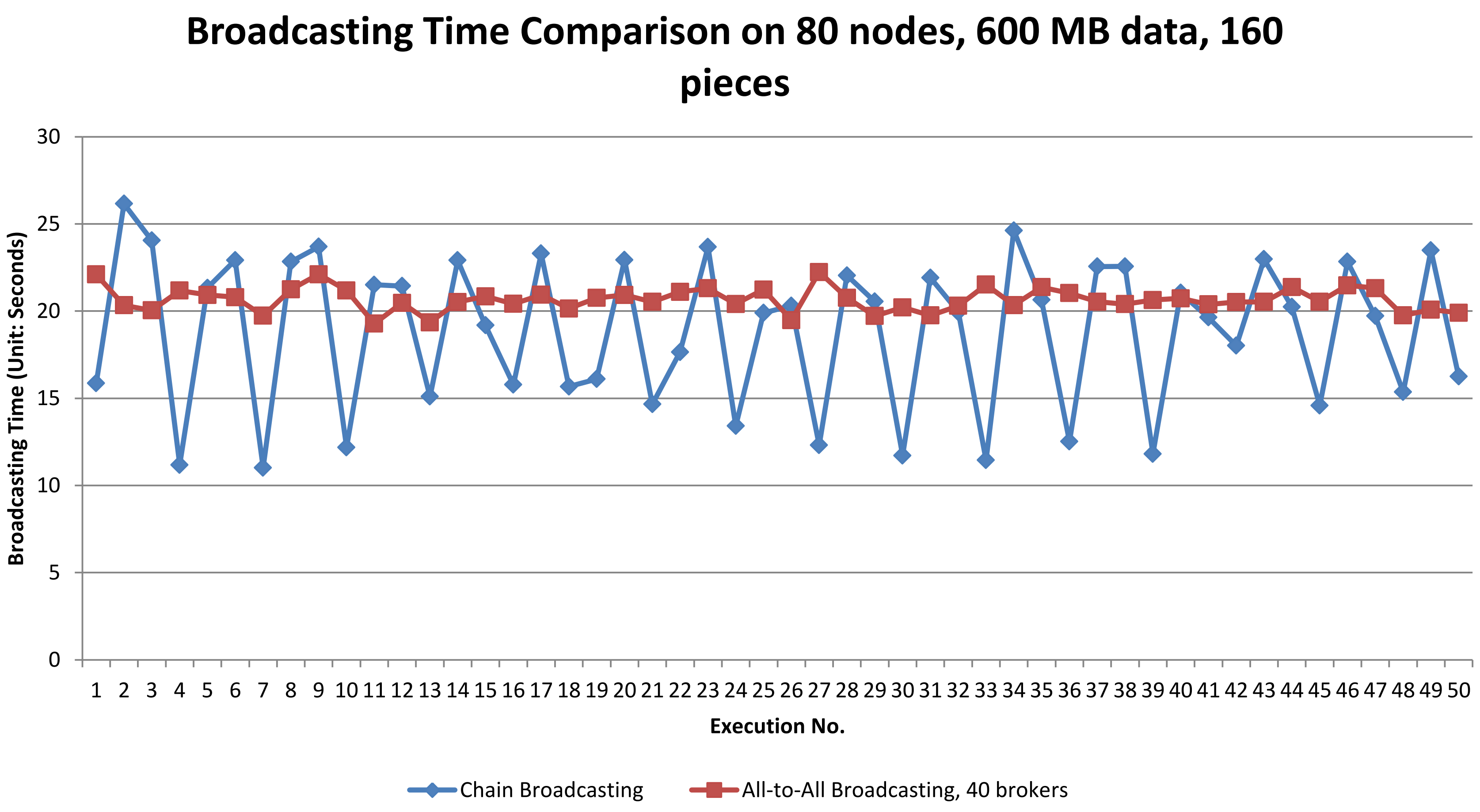
- Driver sender:
  - send broadcasting data
  - get acknowledgement
  - send next broadcasting data
  - …

- Daemon sender:
  - receive data from the last daemon (or driver)
  - cache data to daemon
  - Send data to next daemon (waits for ACK)
  - send acknowledgement to the last daemon

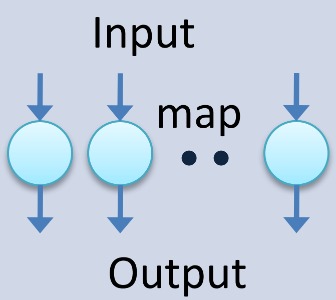
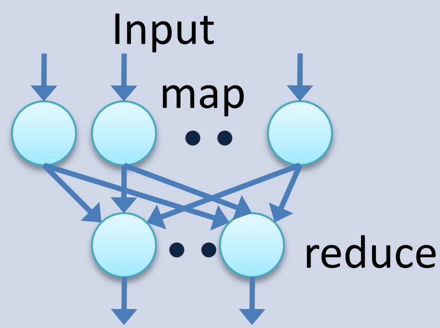
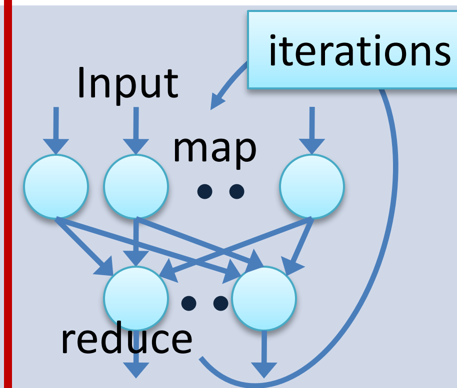
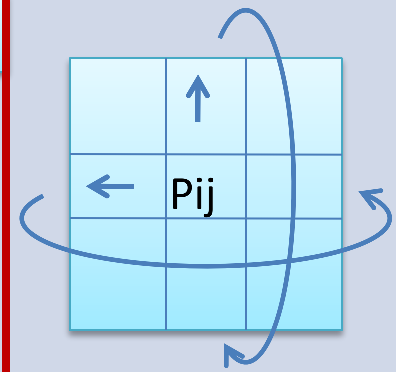
Chain Broadcasting Protocol



Broadcasting Time Comparison



Applications & Different Interconnection Patterns

Map Only	Classic MapReduce	Iterative Reductions Twister	Loosely Synchronous
			
CAP3 Analysis Document conversion (PDF -> HTML) Brute force searches in cryptography Parametric sweeps	High Energy Physics (**HEP**) Histograms **SWG** gene alignment Distributed search Distributed sorting Information retrieval	Expectation maximization algorithms Clustering Linear Algebra	Many MPI scientific applications utilizing wide variety of communication constructs including local interactions
- CAP3 Gene Assembly - PolarGrid Matlab data analysis	- Information Retrieval - HEP Data Analysis - Calculation of Pairwise Distances for ALU Sequences	- Kmeans - **Deterministic Annealing Clustering** - Multidimensional Scaling **MDS**	- Solving Differential Equations and - particle dynamics with short range forces

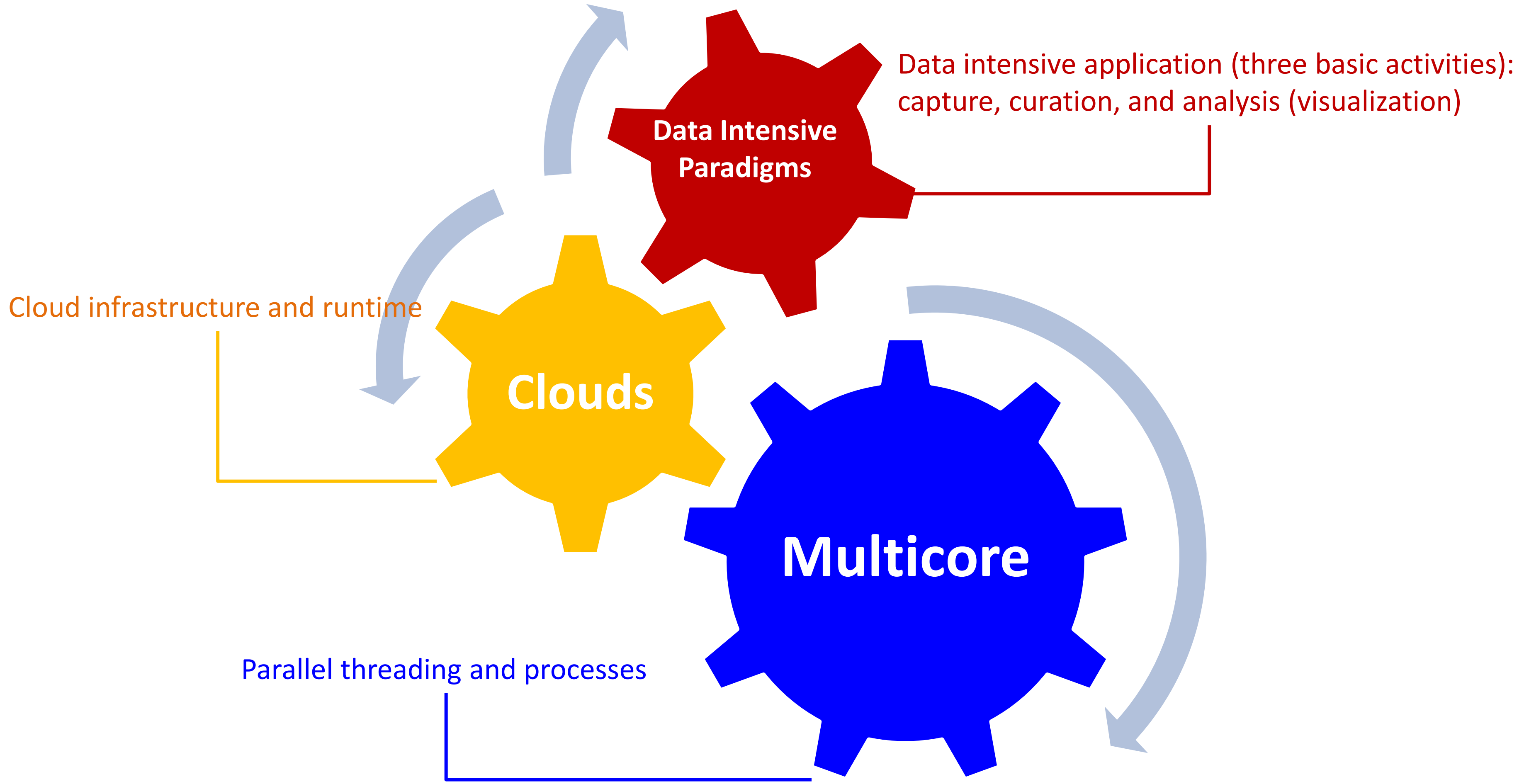
← Domain of MapReduce and Iterative Extensions → MPI

SALSA

Twister Futures

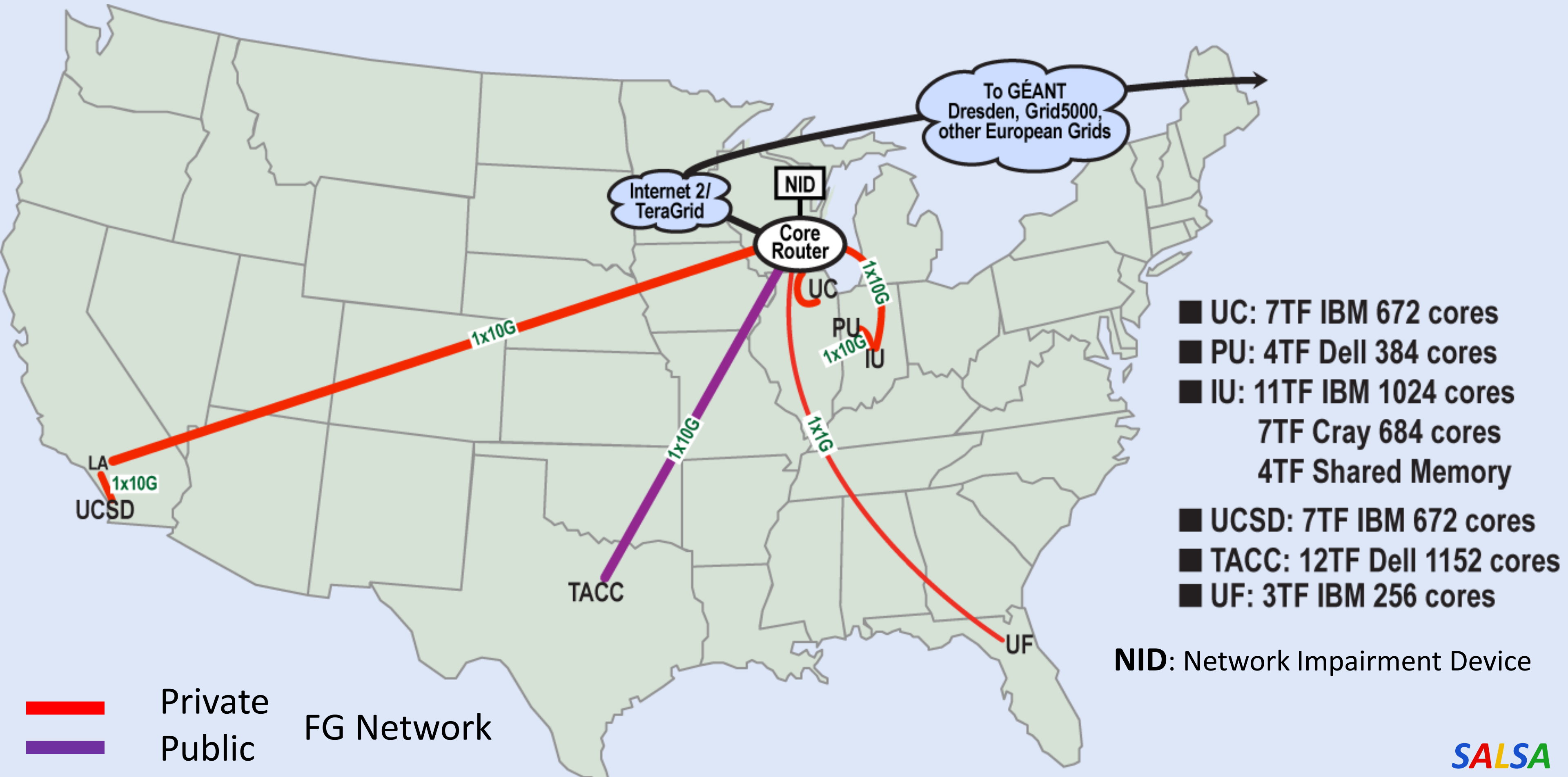
- Development of **library of Collectives** to use at Reduce phase
  - Broadcast and Gather needed by current applications
  - Discover other important ones
  - Implement efficiently on each platform – especially Azure
- Better **software message routing** with broker networks using asynchronous I/O with communication fault tolerance
- Support **nearby location of data and computing** using data parallel file systems
- Clearer application **fault tolerance** model based on implicit synchronizations points at iteration end points
- Later: Investigate **GPU** support
- Later: run time for **data parallel languages** like Sawzall, Pig Latin, LINQ

Convergence is Happening



FutureGrid: a Grid Testbed

- **IU** Cray operational, **IU** IBM (iDataPlex) completed stability test May 6
- **UCSD** IBM operational, **UF** IBM stability test completes ~ May 12
- **Network**, **NID** and **PU** HTC system operational
- **UC** IBM stability test completes ~ May 27; **TACC** Dell awaiting delivery of components



NID: Network Impairment Device

Private
Public
FG Network

SALSA

SC Portland, OR 2009
Computing For A Changing World
November 14-20, 2009
Oregon Convention Center
Portland, Oregon

Demonstrate the concept of Science on Clouds on FutureGrid

Dynamic Cluster Architecture

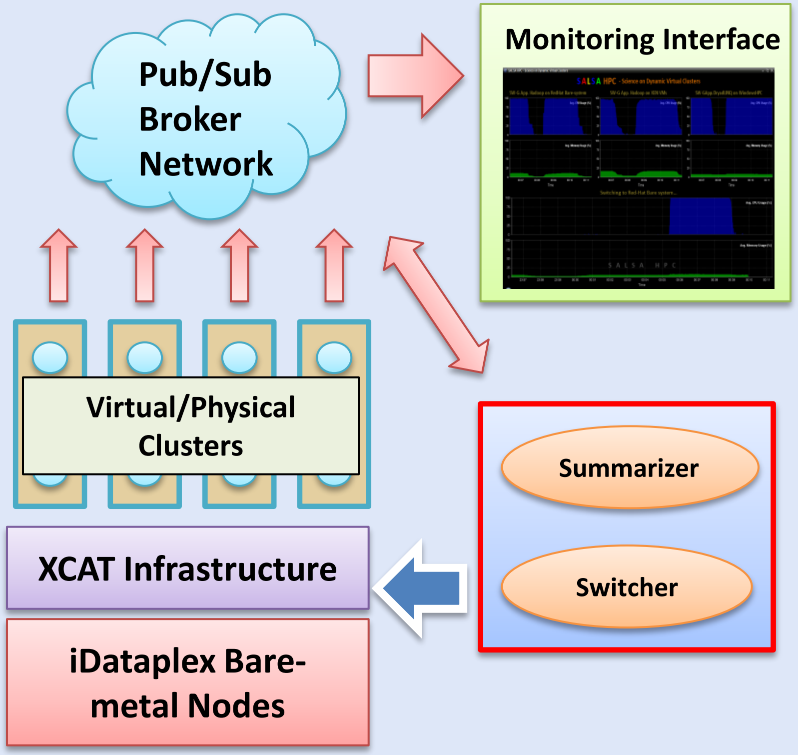
Monitoring Infrastructure

SW-G Using Hadoop	SW-G Using Hadoop	SW-G Using DryadLINQ
Linux Bare-system	Linux on Xen	Windows Server 2008 Bare-system

XCAT Infrastructure

iDataplex Bare-metal Nodes (32 nodes)

Monitoring & Control Infrastructure



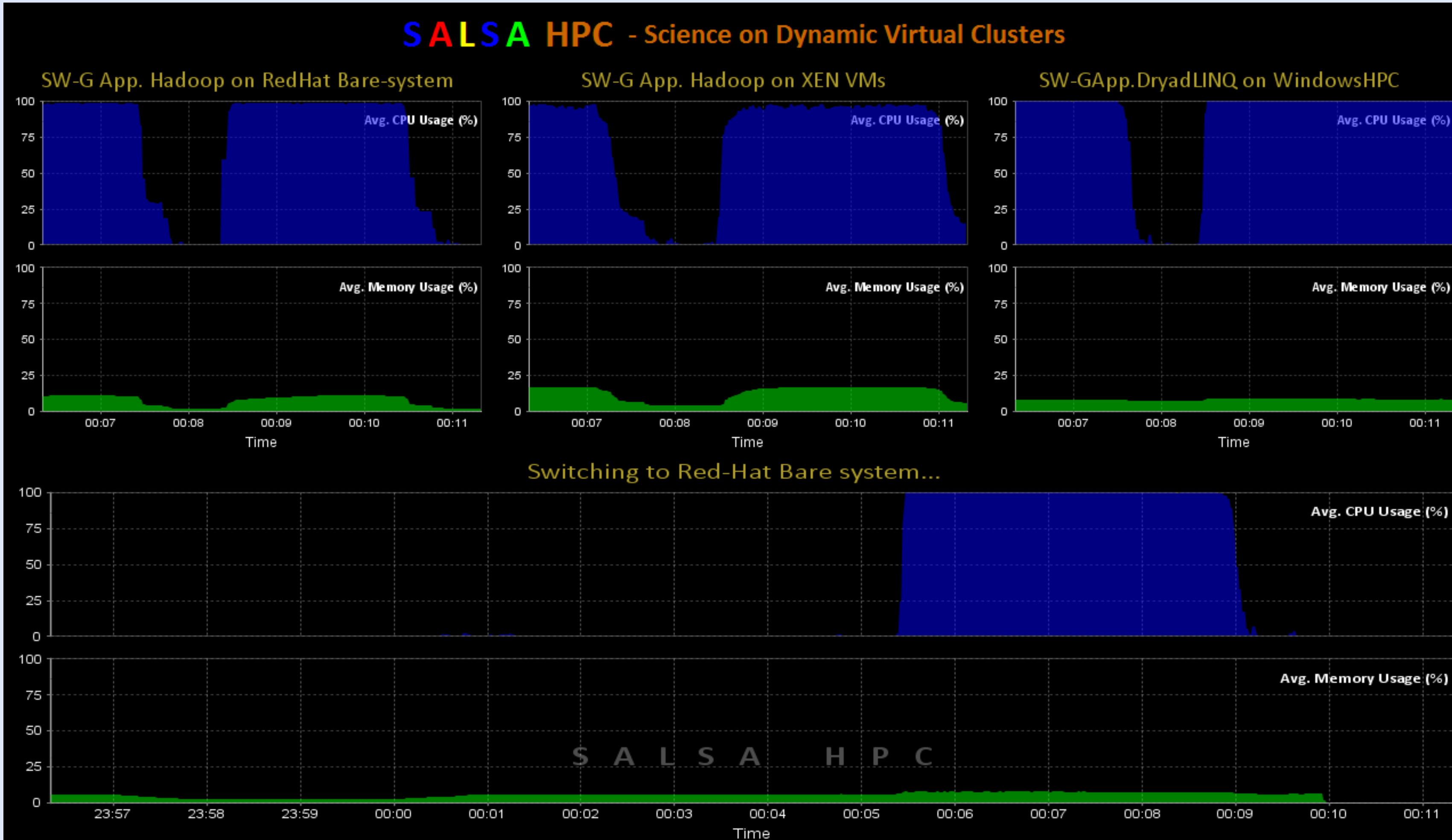
- Switchable clusters on the same hardware (~5 minutes between different OS such as Linux+Xen to Windows+HPCS)
- Support for virtual clusters
- SW-G : Smith Waterman Gotoh Dissimilarity Computation as an pleasingly parallel problem suitable for MapReduce style applications

Computing For A Changing World.

Portland, OR 2009

November 14-20, 2009
Oregon Convention Center
Portland, Oregon

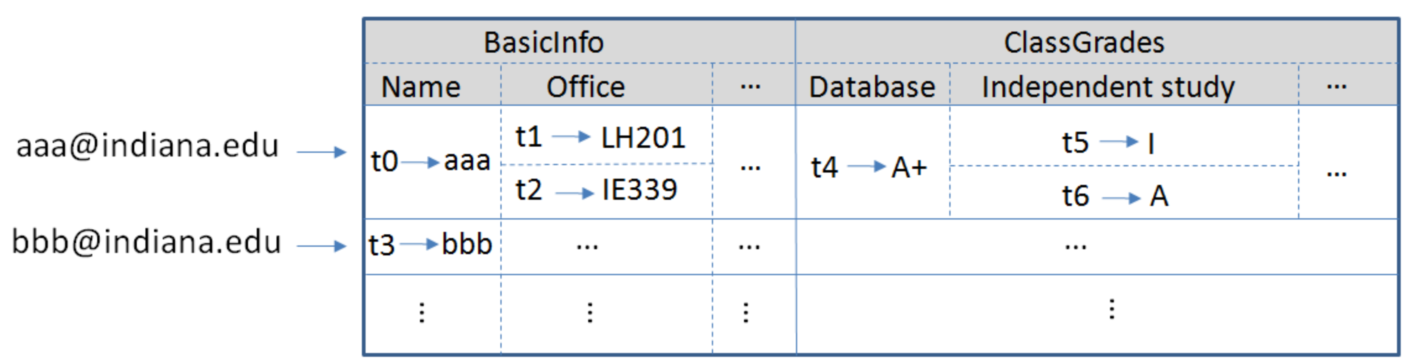
Demonstrate the concept of Science on Clouds using a FutureGrid cluster



- Top: 3 clusters are switching applications on fixed environment. Takes approximately 30 seconds.
- Bottom: Cluster is switching between environments: Linux; Linux +Xen; Windows + HPCS.
  Takes approxomately 7 minutes
- SALSAHPC Demo at SC09. This demonstrates the concept of Science on Clouds using a FutureGrid iDataPlex. SALSA

Experimenting Lucene Index on HBase in an HPC Environment

- Background: data intensive computing requires storage solutions for huge amounts of data
- One proposed solution: HBase, Hadoop implementation of Google's BigTable



Column families: BasicInfo, ClassGrades
Qualifiers: Name, Office, Database, Independent Study
Row keys: aaa@indiana.edu, bbb@indian.edu
Version timestamps: t0, t1, t2, t3, t4, t5, t6

SALSA

System design

- Table schemas:

  - title index table: <term value> --> {frequencies:[<doc id>, <doc id>, ...]}

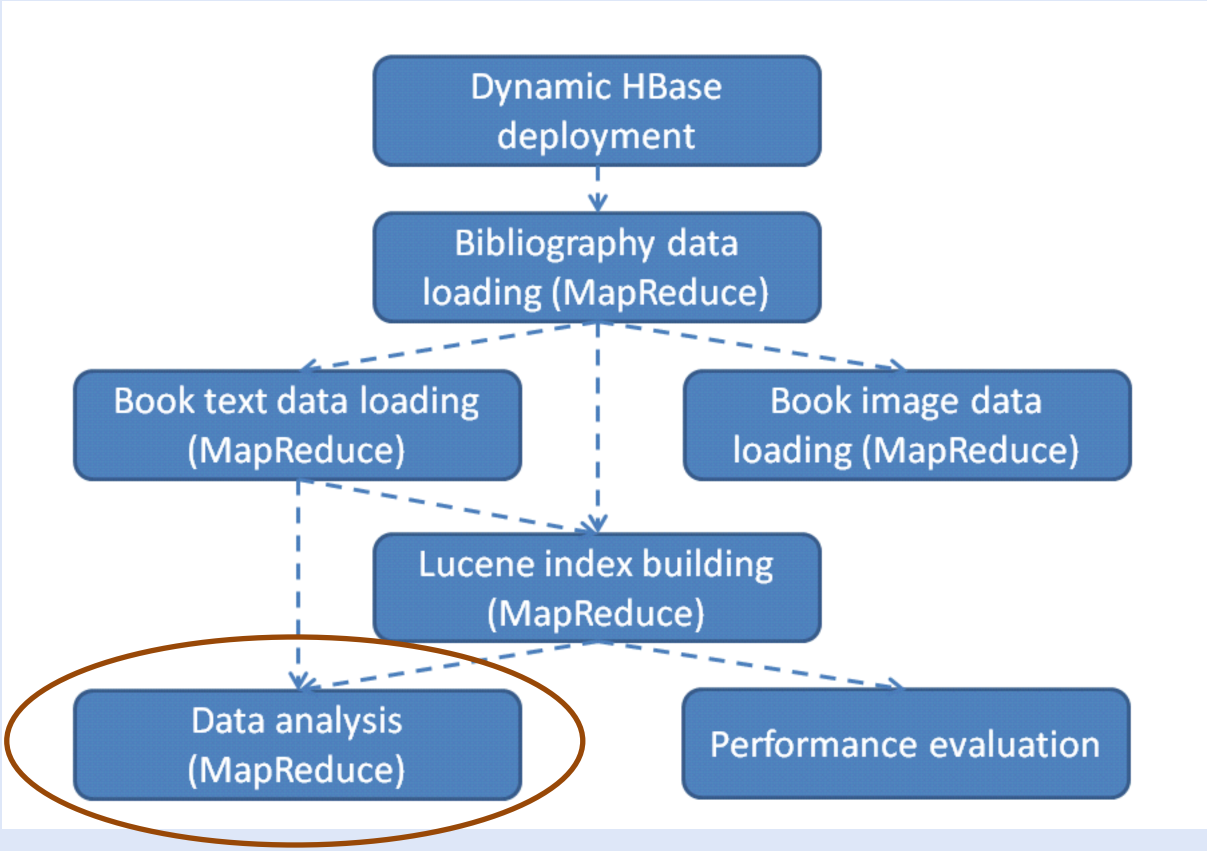
  - texts index table: <term value> --> {frequencies:[<doc id>, <doc id>, ...]}

  - texts term position vector table: <term value> --> {positions:[<doc id>, <doc id>, ...]}
- Natural integration with HBase
- Reliable and scalable index data storage
- Real-time document addition and deletion
- MapReduce programs for building index and analyzing index data

SALSA

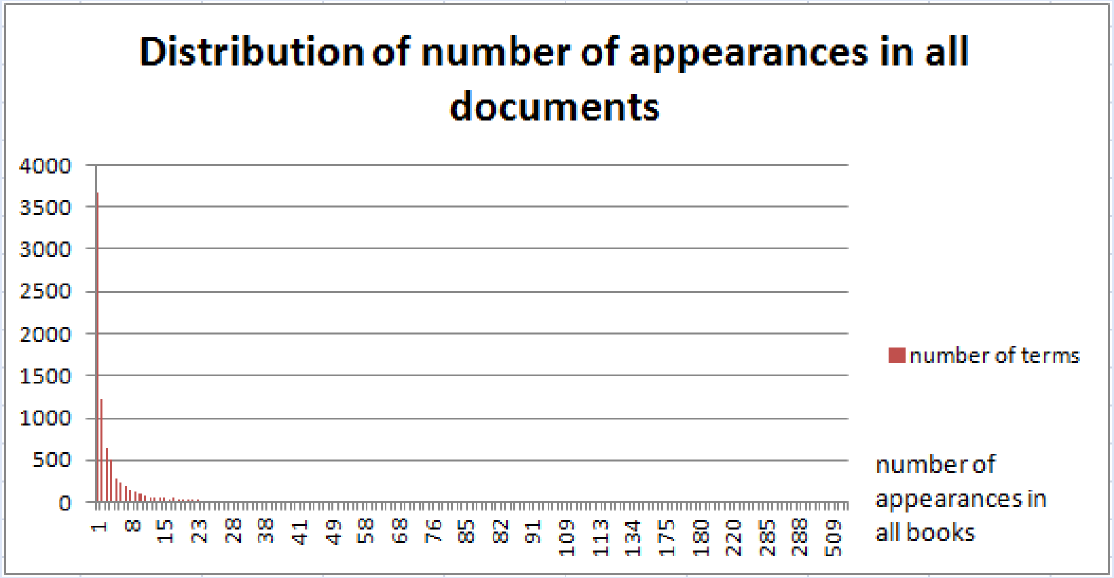
System implementation

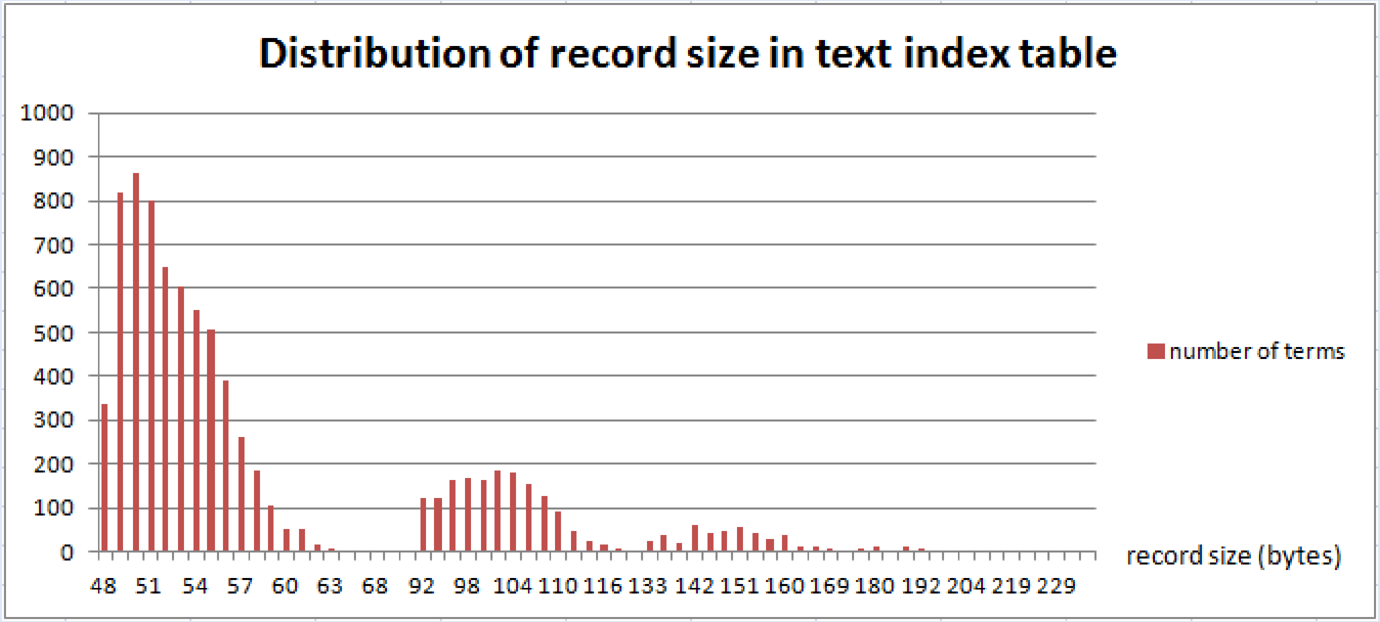
- Experiments completed in the Alamo HPC cluster of FutureGrid
- MyHadoop -> MyHBase
- Workflow:



SALSA

Index data analysis





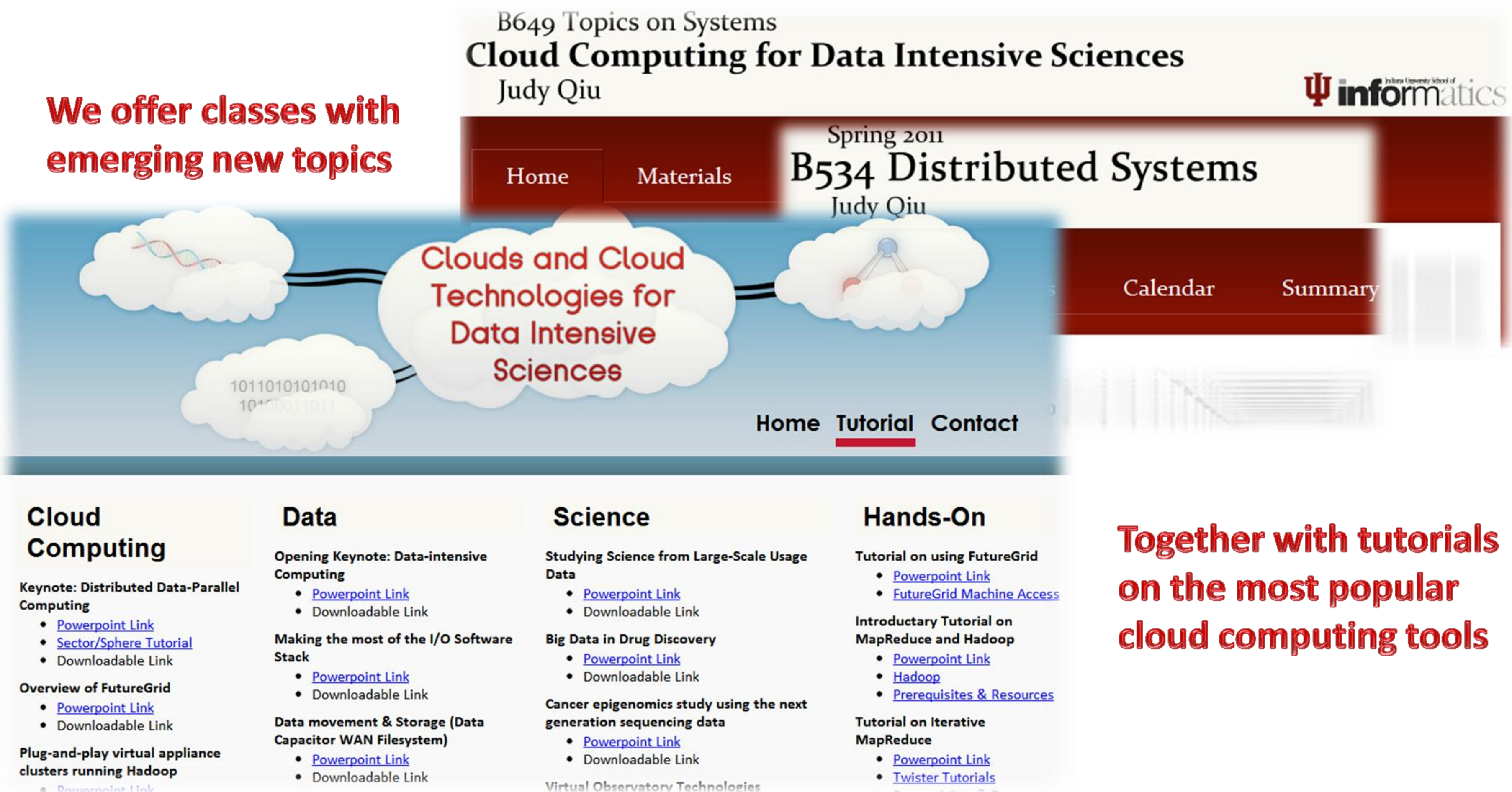
SALSA

Education and Broader Impact

We devote a lot to guide students
who are interested in computing

Education

We offer classes with emerging new topics



Together with tutorials on the most popular cloud computing tools

Broader Impact

Hosting workshops and spreading our technology across the nation

Giving students unforgettable research experience

Acknowledgement

SALSA HPC Group
Indiana University

http://salsahpc.indiana.edu

Twister4Azure
Iterative MapReduce for Azure Cloud

Twister
Cross Platform Iterative MapReduce

PlotViz

Microsoft

INDIANA UNIVERSITY

NSF

NATIONAL INSTITUTES OF HEALTH

Future Grid

SCHOOL OF INFORMATICS AND COMPUTING
INDIANA UNIVERSITY
Bloomington