MPI and MapReduce

CCGSC 2010 Flat Rock NC

September 8 2010

Geoffrey Fox

gcf@indiana.edu

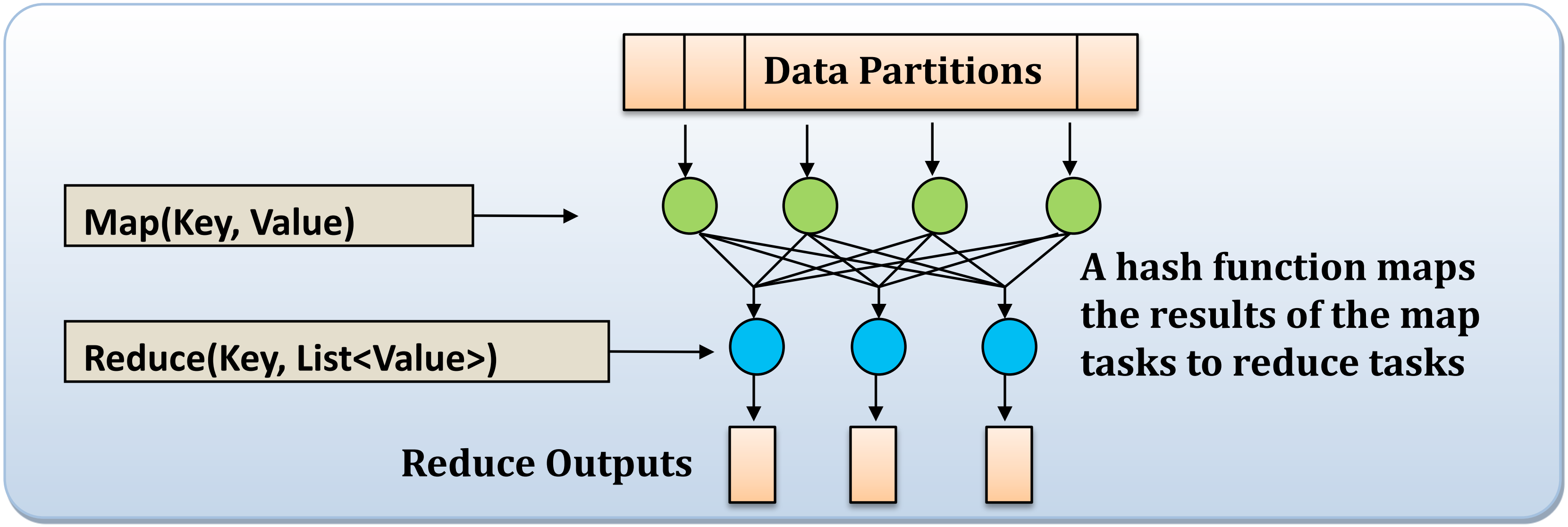
http://www.infomall.org http://www.futuregrid.org

Director, Digital Science Center, Pervasive Technology Institute

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

Indiana University Bloomington

MapReduce

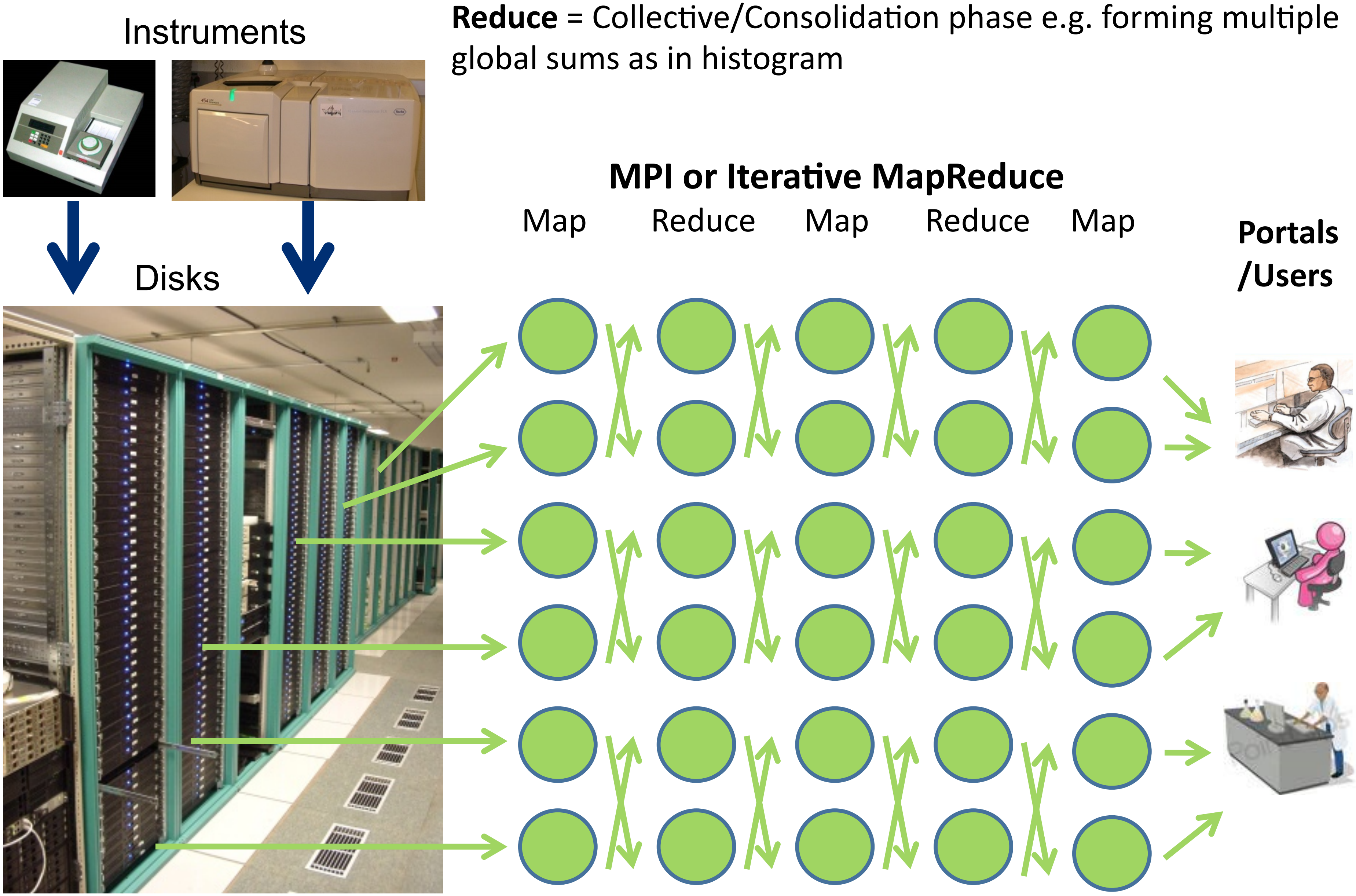


- Implementations (Hadoop – Java; Dryad – Windows) support:
  - Splitting of data with customized file systems
  - Passing the output of map functions to reduce functions
  - Sorting the inputs to the reduce function based on the intermediate keys
  - Quality of service
- 20 petabytes per day (on an average of 400 machines) processed by Google using MapReduce September 2007

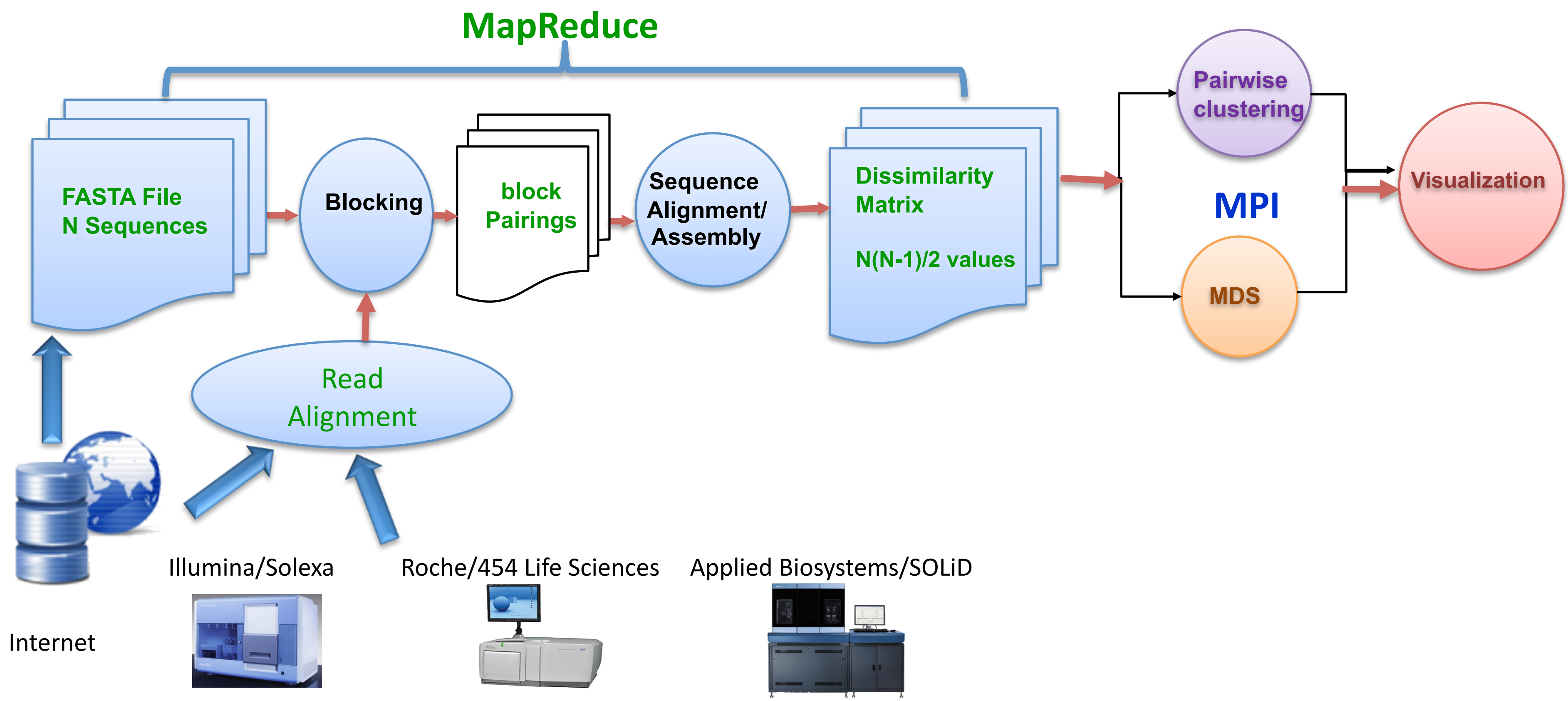
MapReduce "File/Data Repository" Parallelism

Map = (data parallel) computation reading and writing data

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

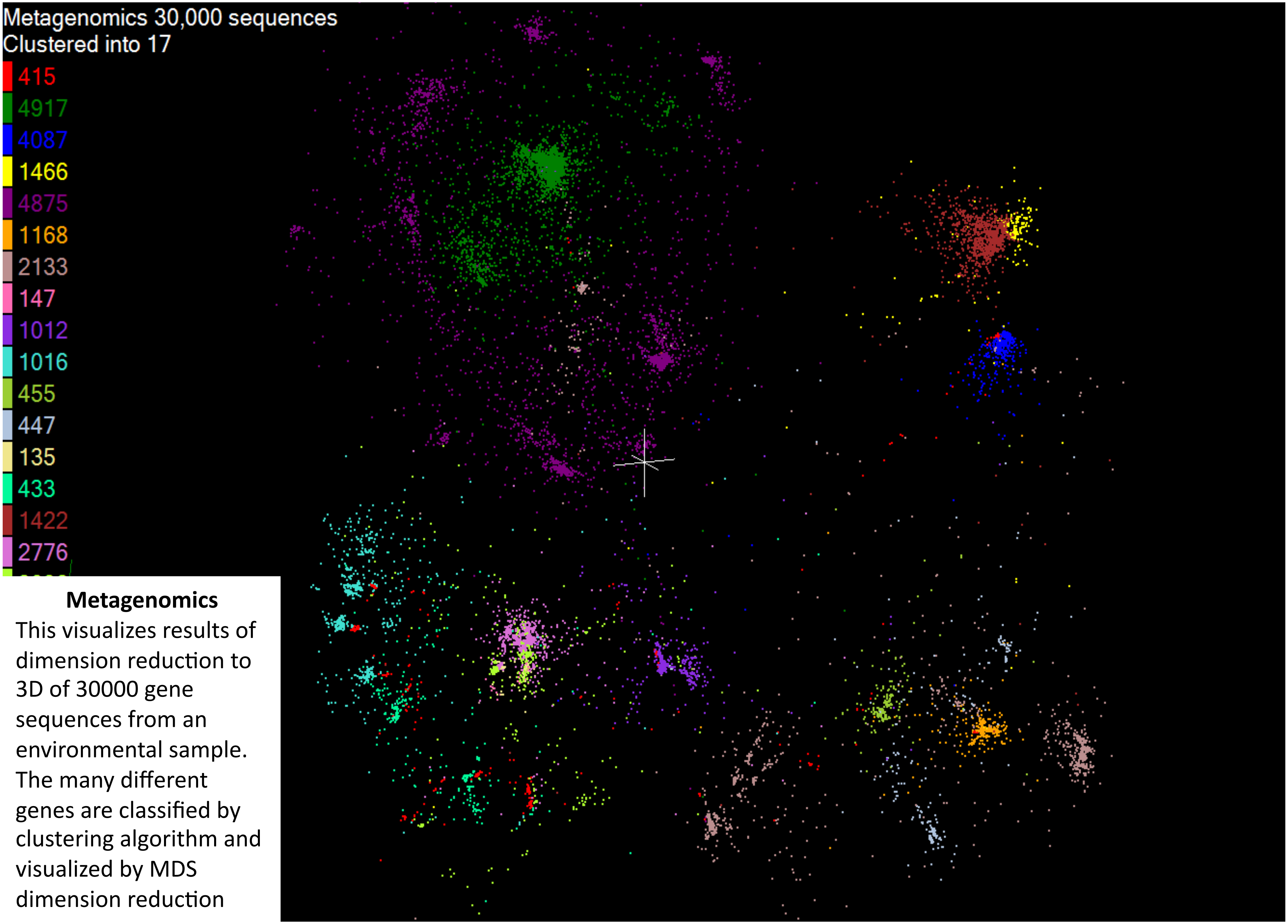


Typical Application Challenge:
DNA Sequencing Pipeline



Modern Commercial Gene Sequencers

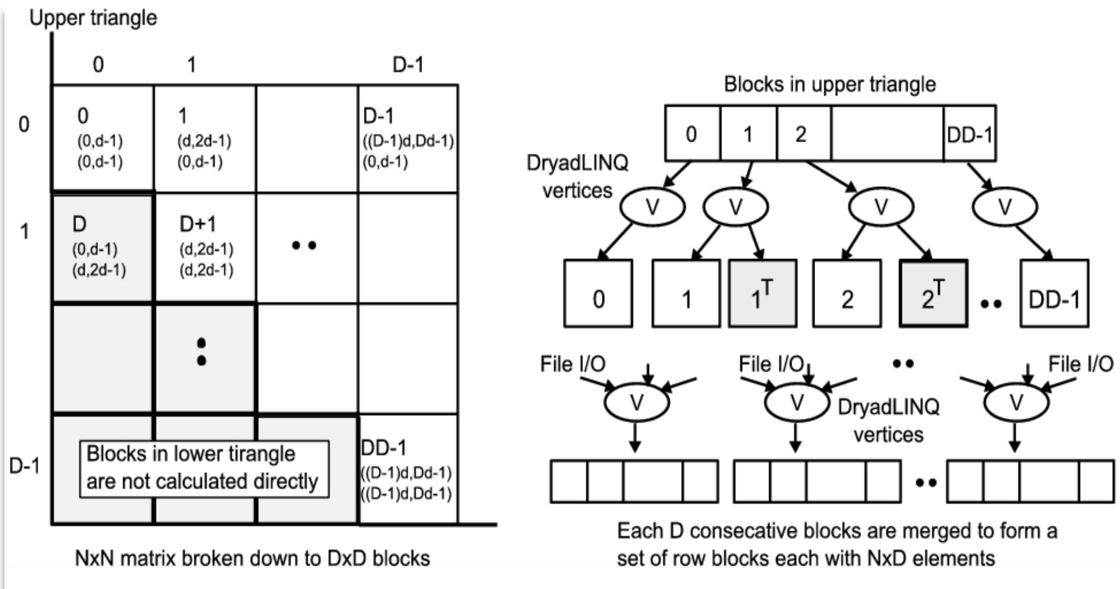
Linear Algebra or Expectation Maximization based data mining poor on MapReduce – equivalent to using MPI writing messages to disk and restarting processes each step/iteration of algorithm

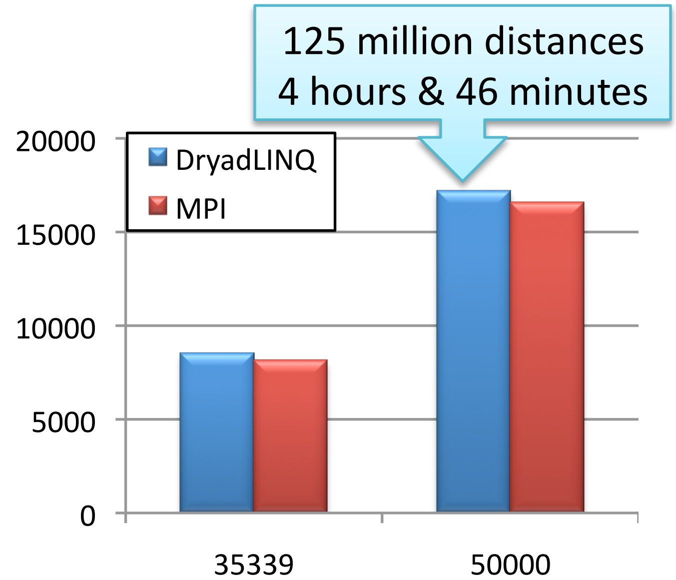


Metagenomics

This visualizes results of dimension reduction to 3D of 30000 gene sequences from an environmental sample. The many different genes are classified by clustering algorithm and visualized by MDS dimension reduction

All-Pairs Using MPI or DryadLINQ



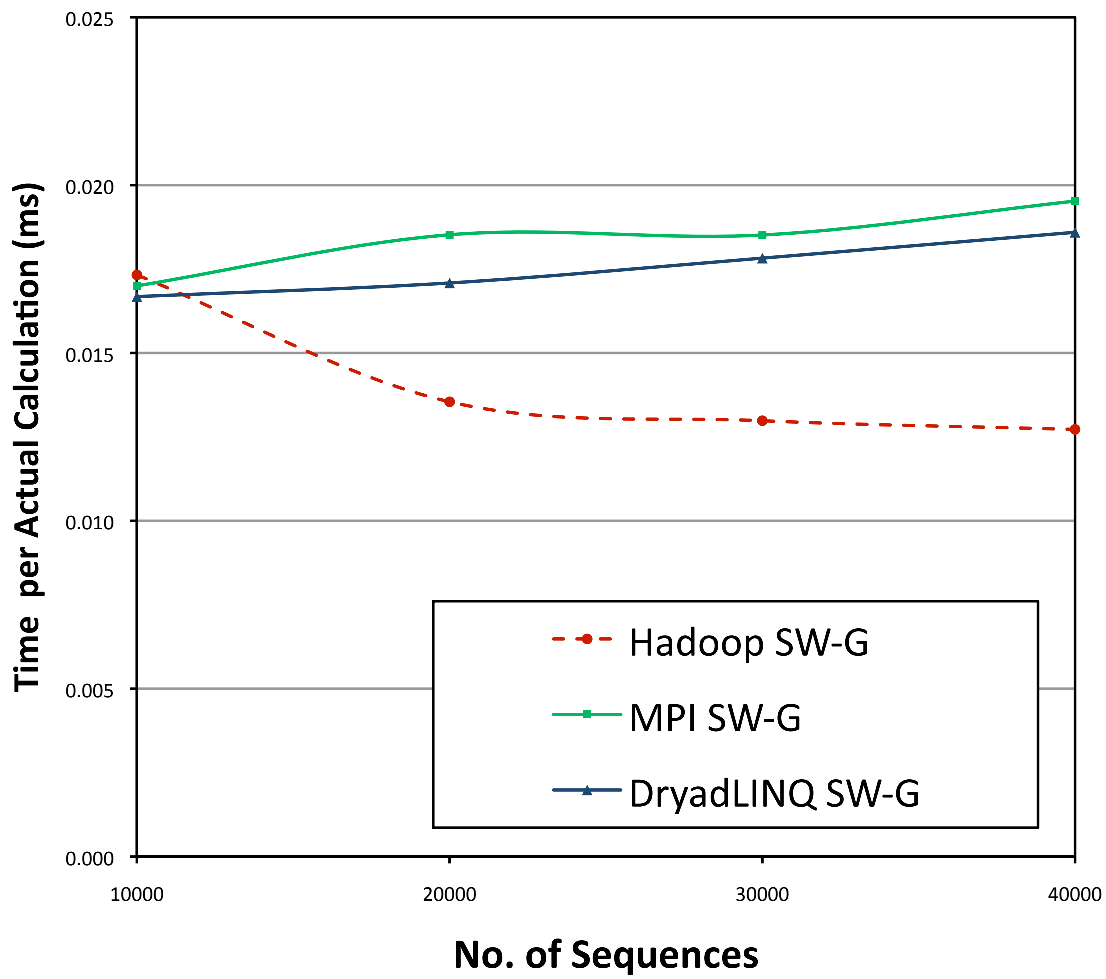


Calculate Pairwise Distances (Smith Waterman Gotoh)

- Calculate pairwise distances for a collection of genes (used for clustering, MDS)
- Fine grained tasks in MPI
- Coarse grained tasks in DryadLINQ
- Performed on 768 cores (Tempest Cluster)

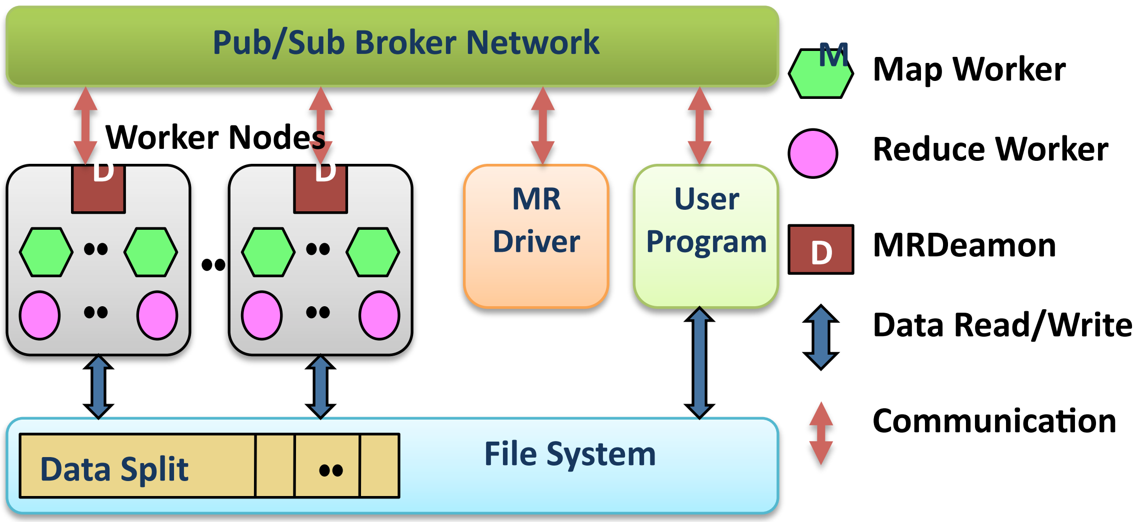
Moretti, C., Bui, H., Hollingsworth, K., Rich, B., Flynn, P., & Thain, D. (2009). All-Pairs: An Abstraction for Data Intensive Computing on Campus Grids. *IEEE Transactions on Parallel and Distributed Systems , 21*, 21-36.

Smith Waterman
MPI DryadLINQ Hadoop

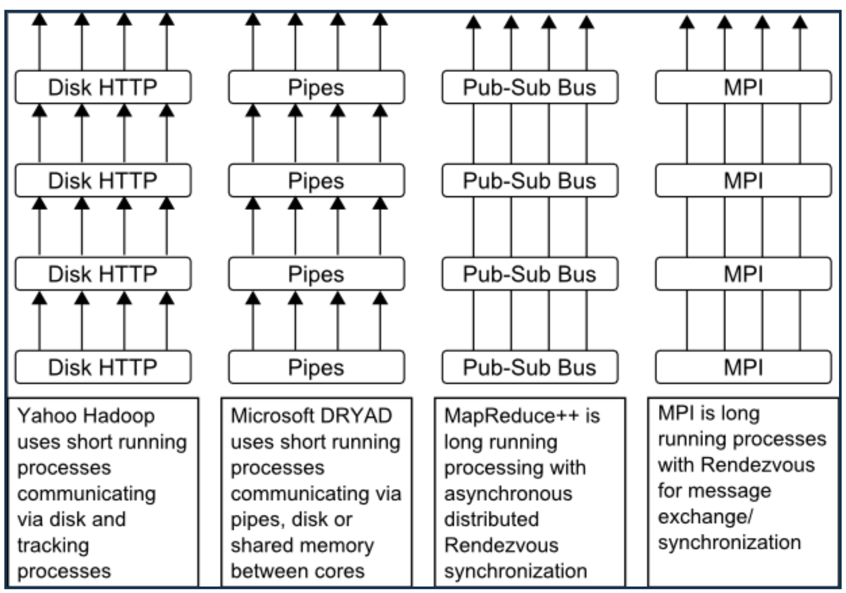


Hadoop is Java; MPI and Dryad are C#

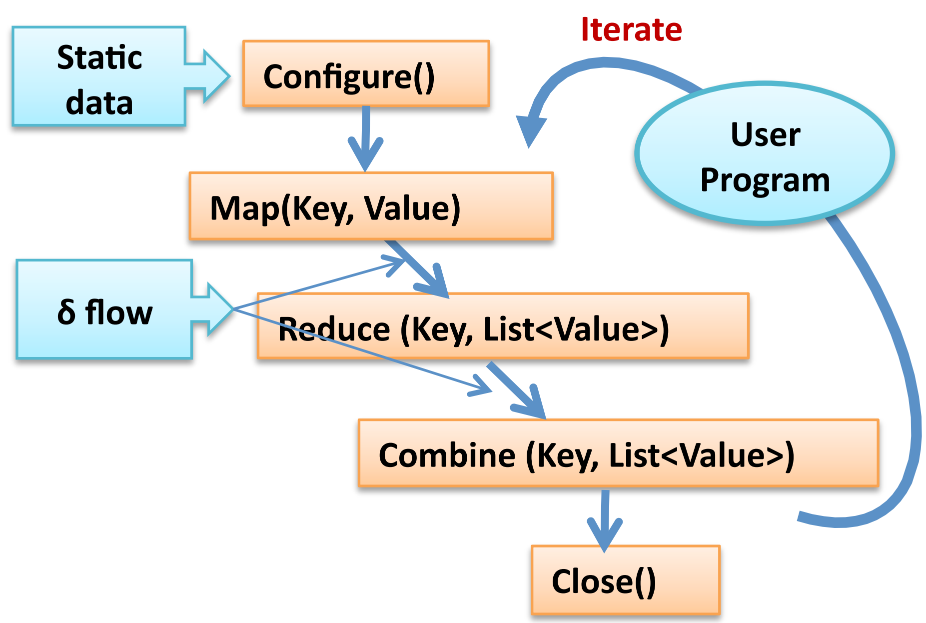
Twister(MapReduce++)



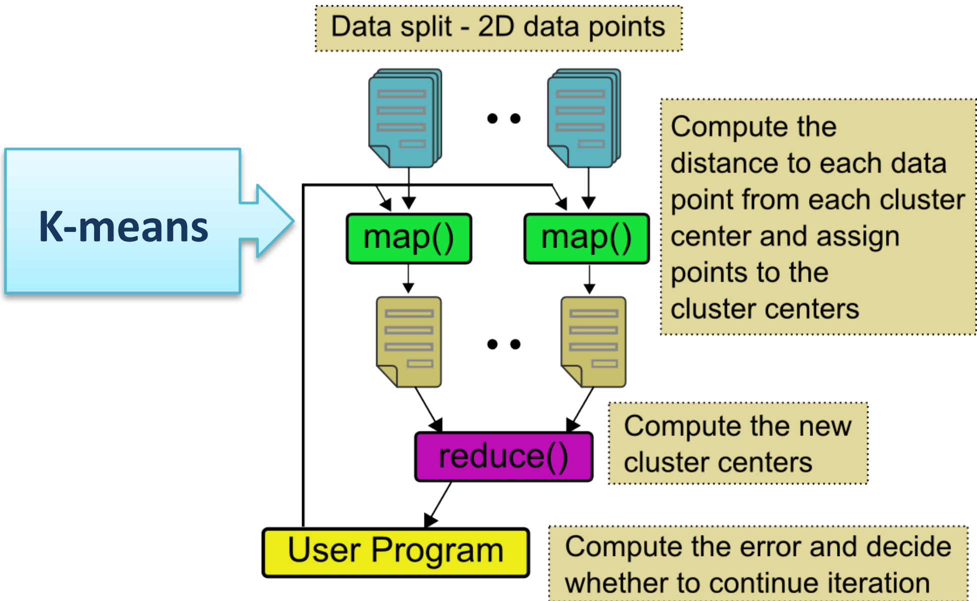
- Streaming based communication
- Intermediate results are directly transferred from the map tasks to the reduce tasks – **eliminates local files**
- Cacheable map/reduce tasks
  - Static data remains in memory
- Combine phase to combine reductions
- User Program is the **composer** of MapReduce computations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



Different synchronization and intercommunication mechanisms used by the parallel runtimes

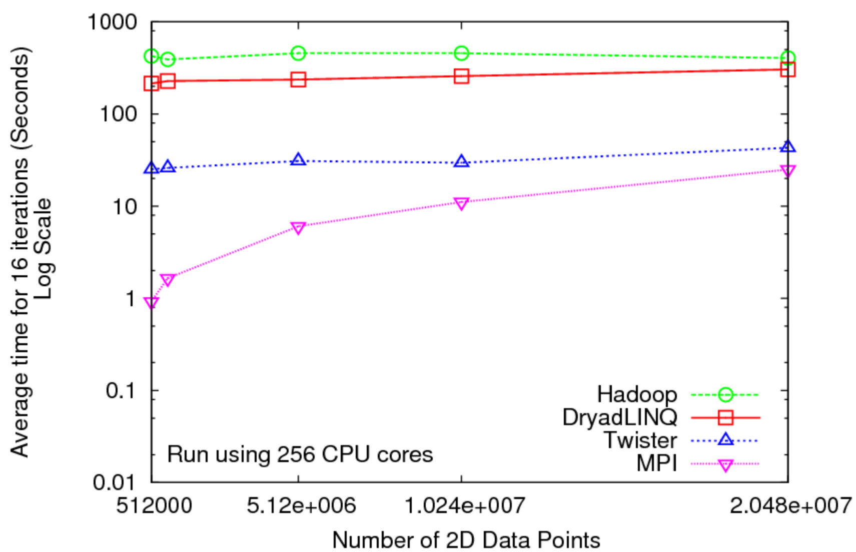


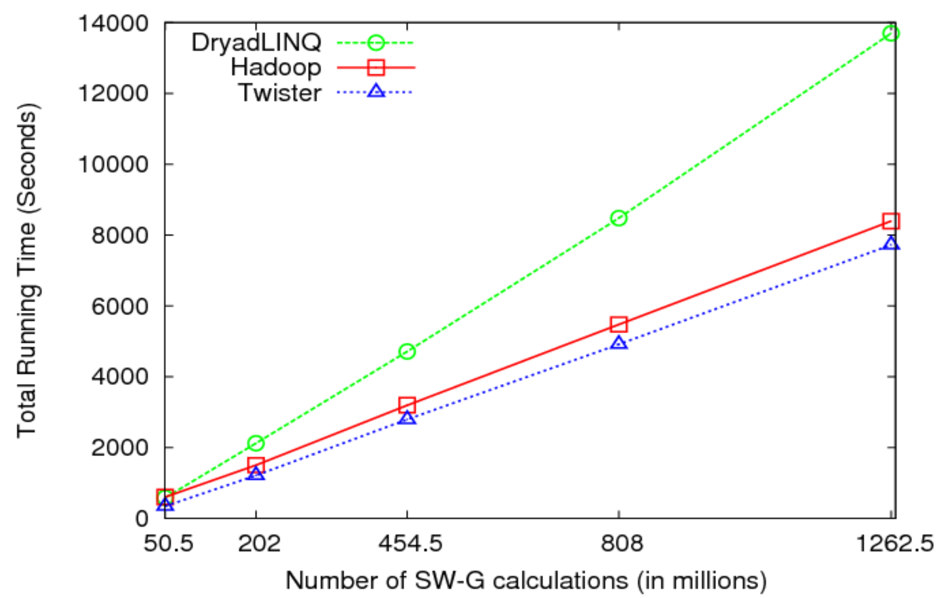
Iterative and non-Iterative Computations



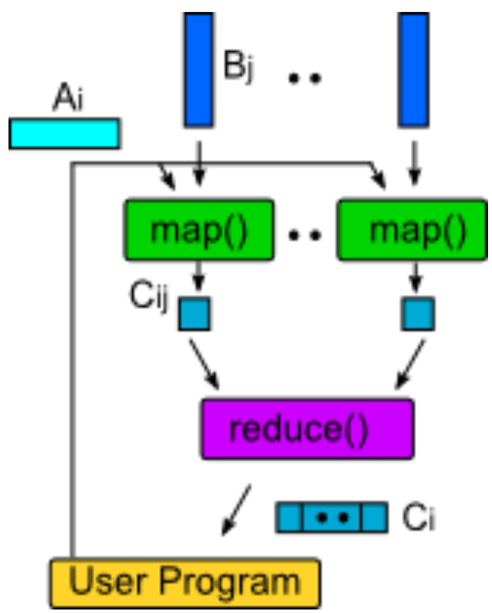
Smith Waterman is a non iterative case and of course runs fine

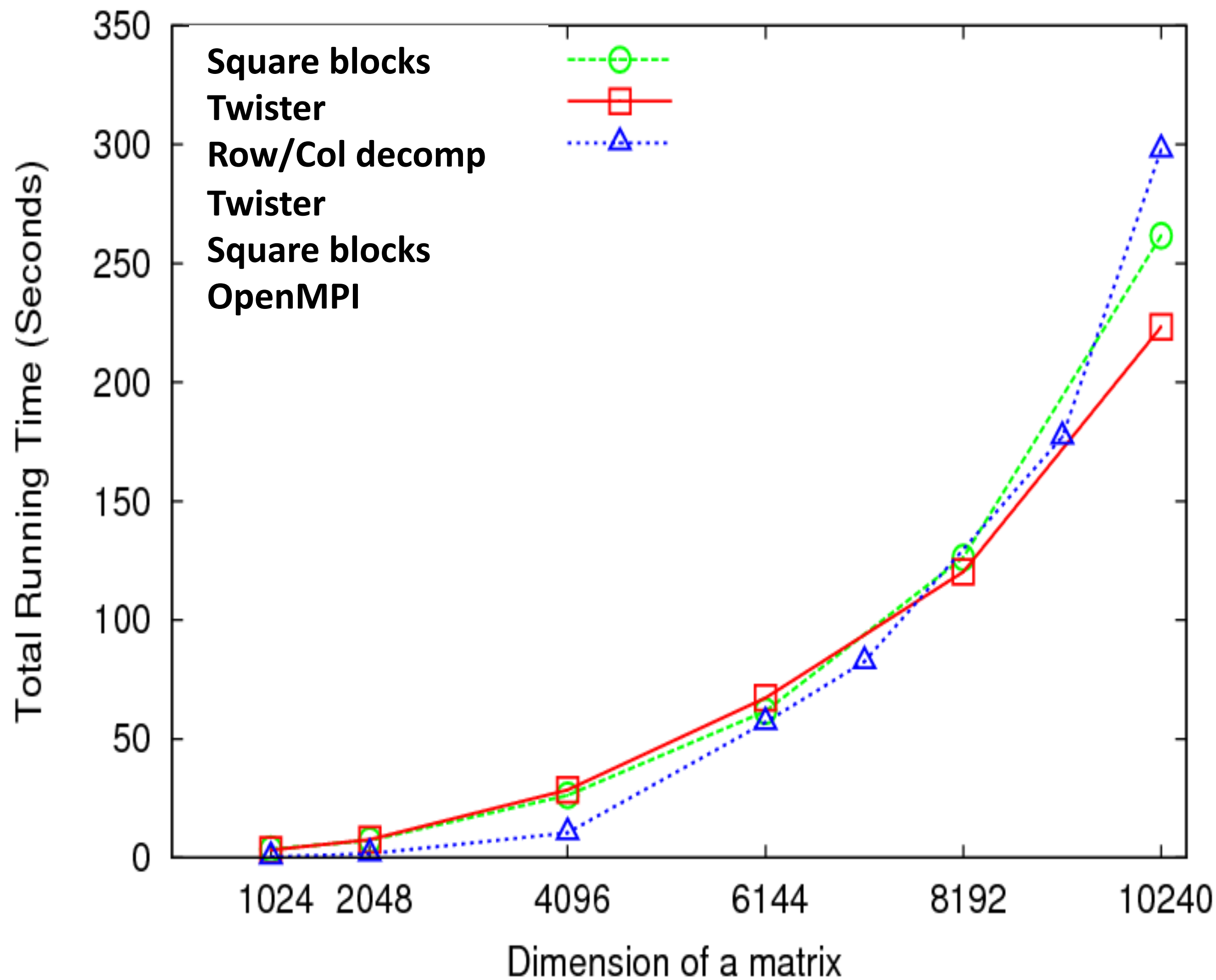
Performance of K-Means





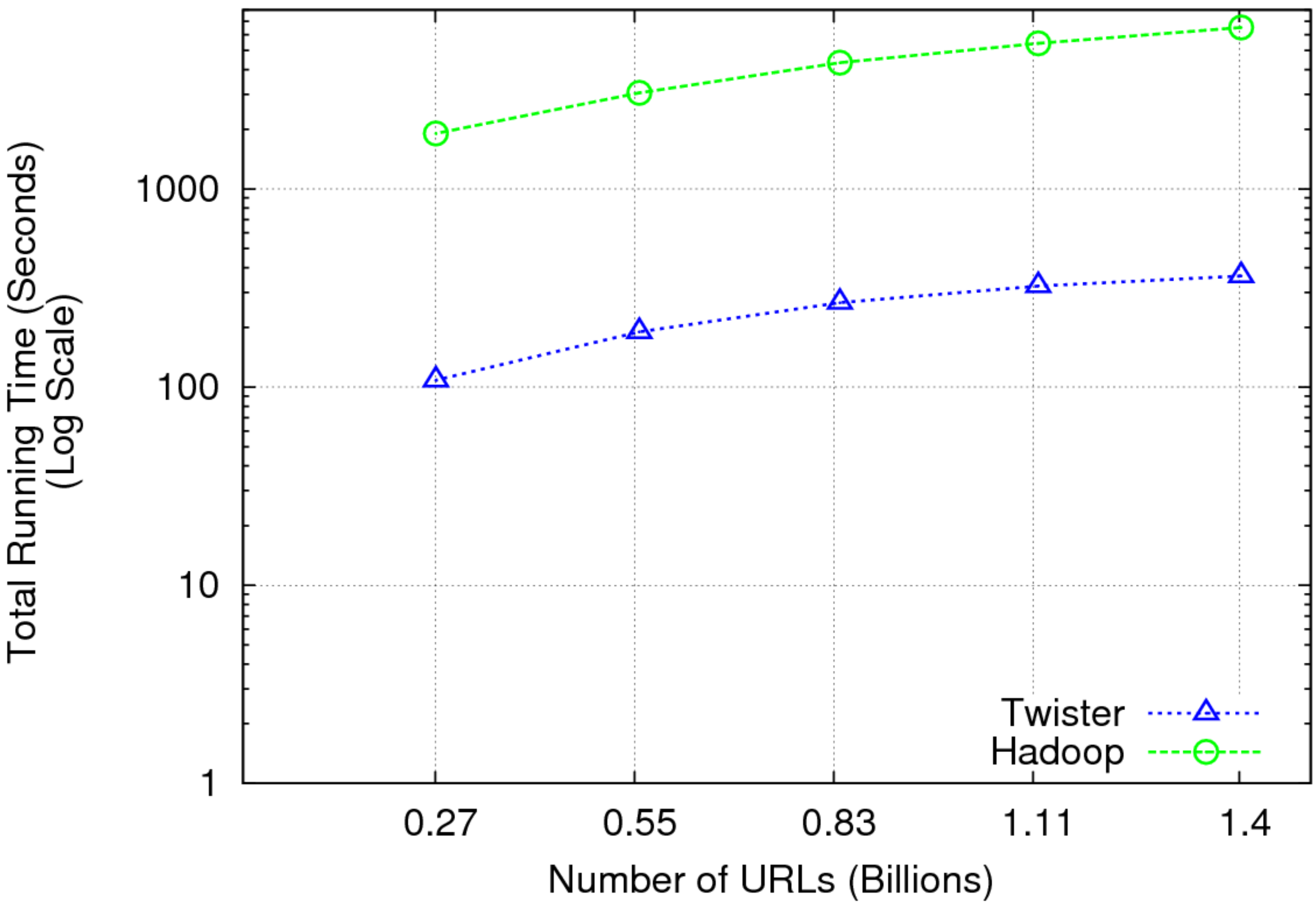
Matrix Multiplication 64 cores





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

using 32 nodes (256 CPU cores) of Crevasse



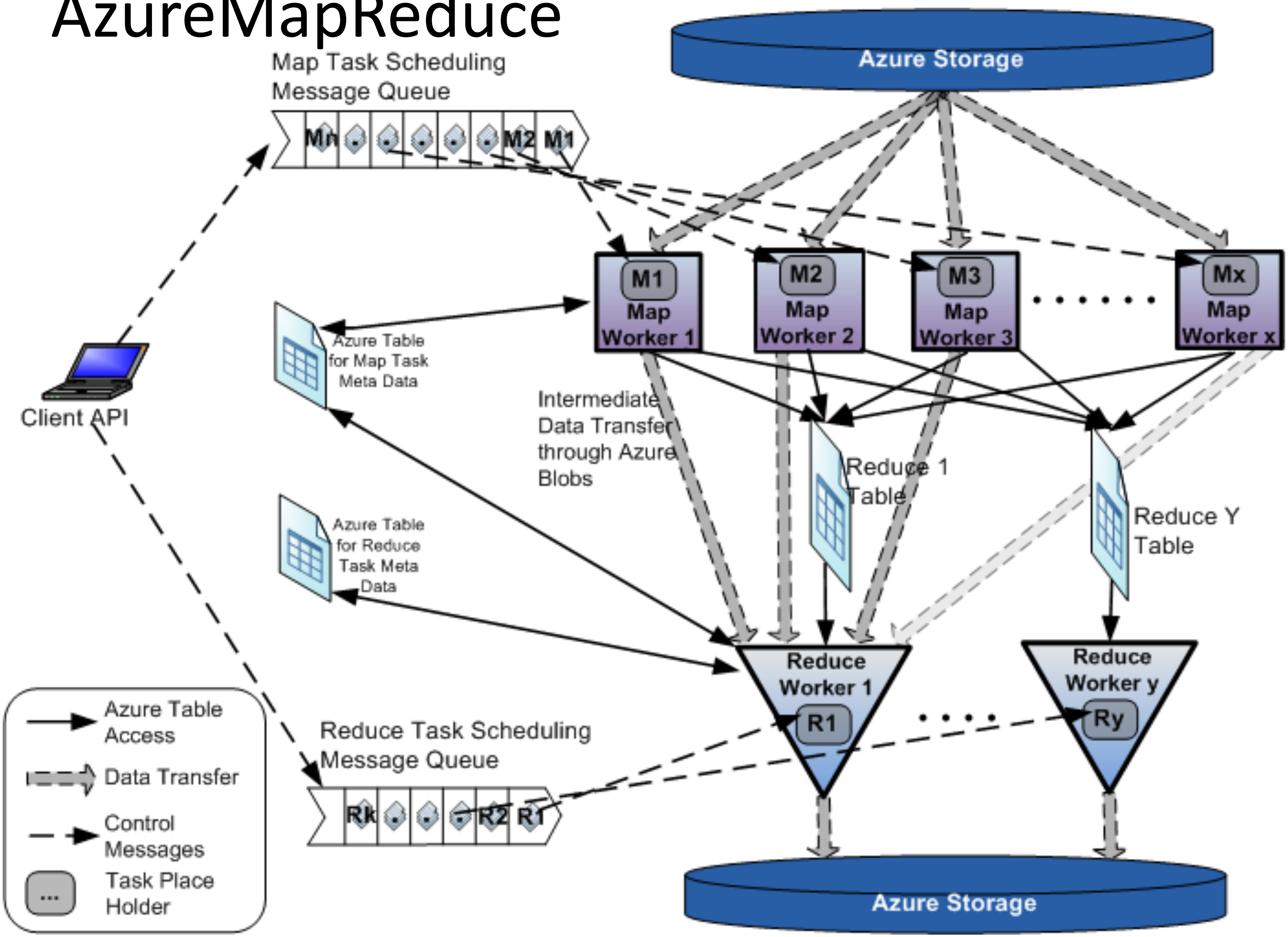
Fault Tolerance and MapReduce

- MPI does "maps" followed by "communication" including "reduce" but does this iteratively
- There must (for most communication patterns of interest) be a strict synchronization at end of each communication phase
  - Thus if a process fails then everything grinds to a halt
- In MapReduce, all Map processes and all reduce processes are independent and stateless and read and write to disks
  - As 1 or 2 (reduce+map) iterations, no difficult synchronization issues
- Thus failures can easily be recovered by rerunning process without other jobs hanging around waiting
- Re-examine MPI fault tolerance in light of MapReduce
  - Relevant for Exascale?
- Re-examine MapReduce in light of MPI experience …..

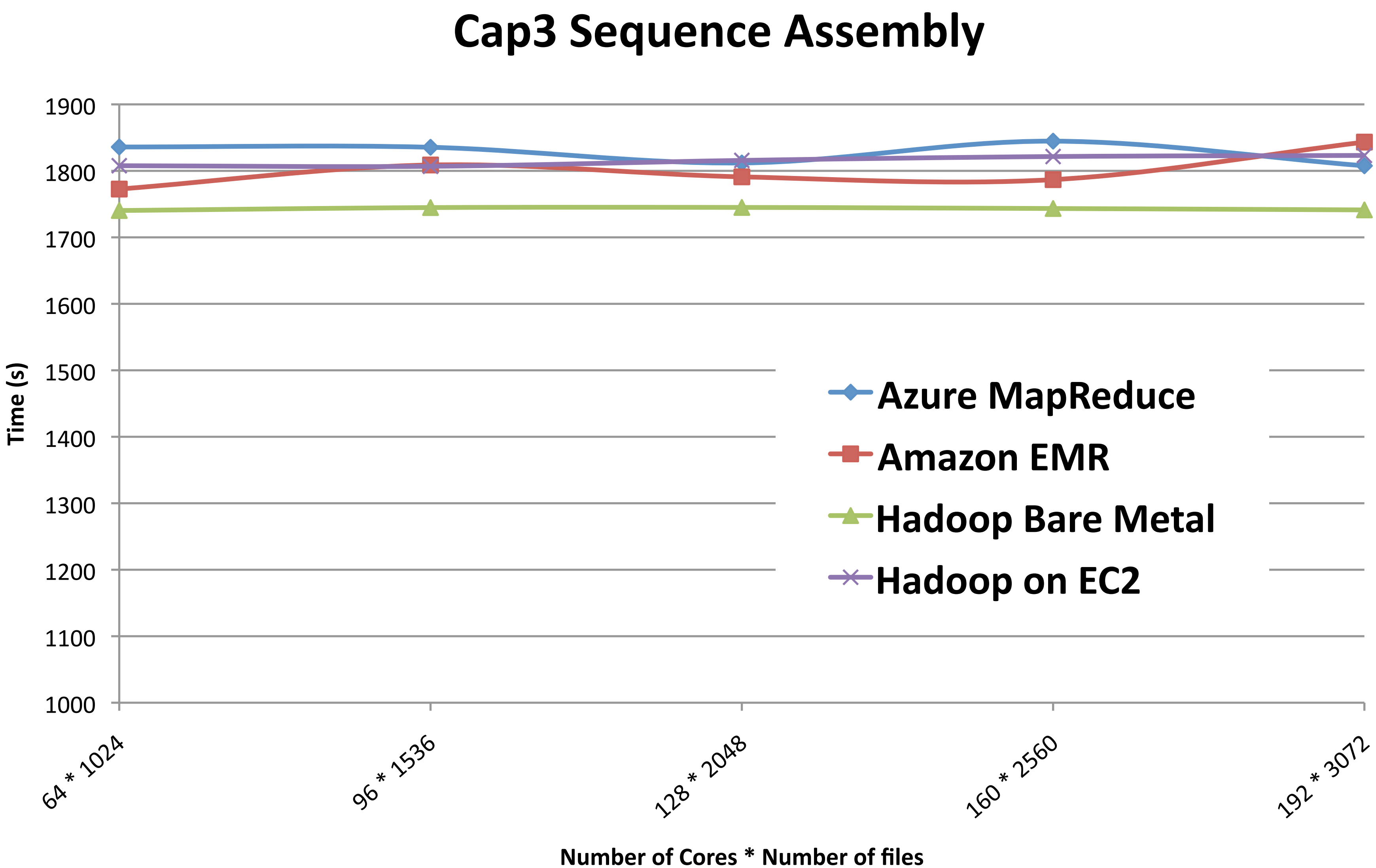
MPI & Iterative MapReduce papers

- **MapReduce on MPI** Torsten Hoefler, Andrew Lumsdaine and Jack Dongarra, Towards Efficient MapReduce Using MPI, *Recent Advances in Parallel Virtual Machine and Message Passing Interface* Lecture Notes in Computer Science, 2009, Volume 5759/2009, 240-249
- **MPI with generalized MapReduce**
- Jaliya Ekanayake, Hui Li, Bingjing Zhang, Thilina Gunarathne, Seung-Hee Bae, Judy Qiu, Geoffrey Fox Twister: A Runtime for Iterative MapReduce, *Proceedings of the First International Workshop on MapReduce and its Applications of ACM HPDC 2010* conference, Chicago, Illinois, June 20-25, 2010 http://grids.ucs.indiana.edu/ptliupages/publications/twister__hpdc_mapreduce.pdf http://www.iterativemapreduce.org/
- Grzegorz Malewicz, Matthew H. Austern, Aart J. C. Bik, James C. Dehnert, Ilan Horn, Naty Leiser, and Grzegorz Czajkowski Pregel: A System for Large-Scale Graph Processing, *Proceedings of the 2010 international conference on Management of data* Indianapolis, Indiana, USA Pages: 135-146 2010
- Yingyi Bu, Bill Howe, Magdalena Balazinska, Michael D. Ernst HaLoop: Efficient Iterative Data Processing on Large Clusters, *Proceedings of the VLDB Endowment, Vol. 3, No. 1, The 36th International Conference on Very Large Data Bases*, September 1317, 2010, Singapore.
- Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, Ion Stoica Spark: Cluster Computing with Working Sets poster at http://radlab.cs.berkeley.edu/w/upload/9/9c/Spark-retreat-poster-s10.pdf

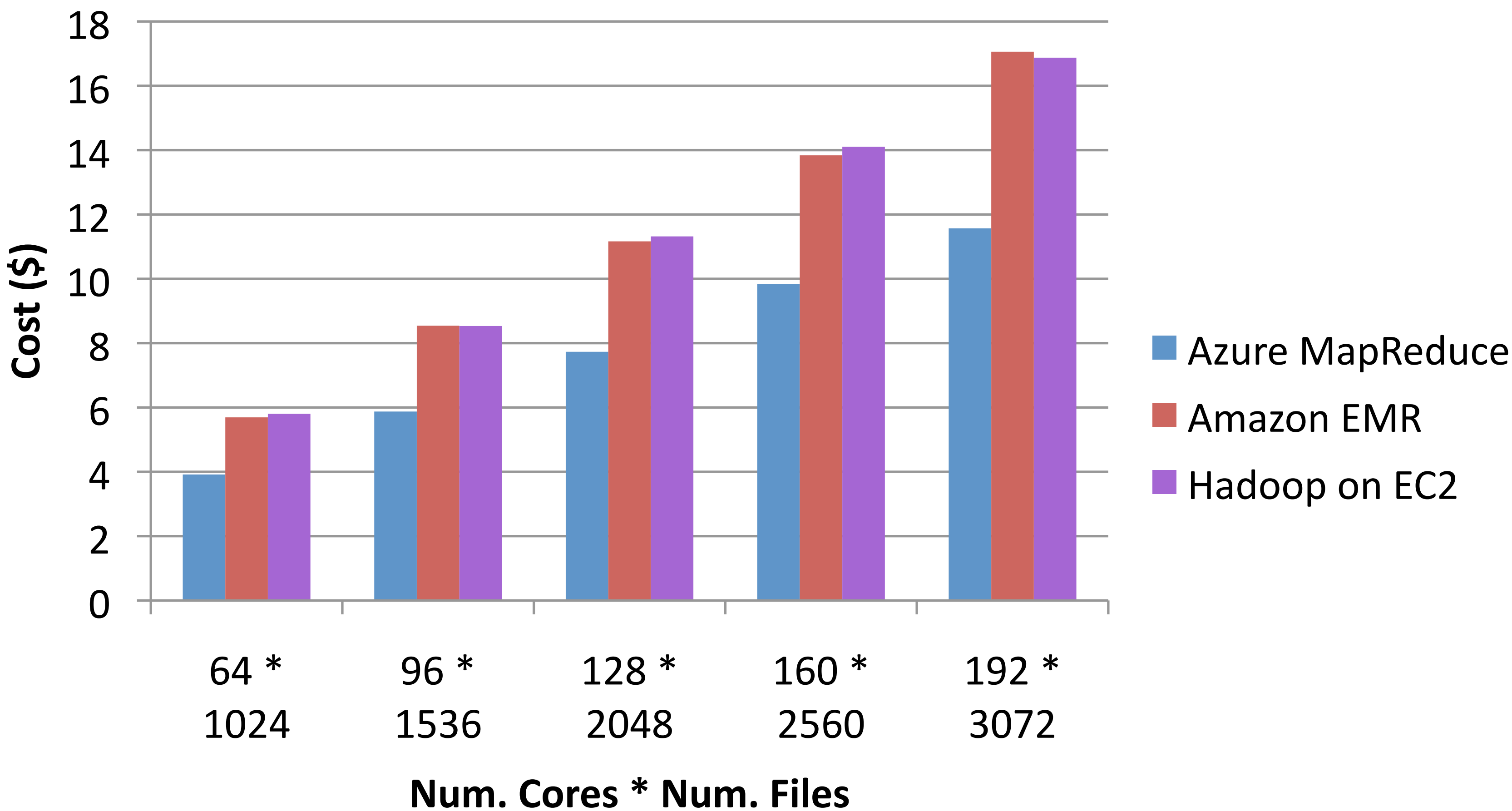
AzureMapReduce



Scaled Timing with Azure/Amazon MapReduce

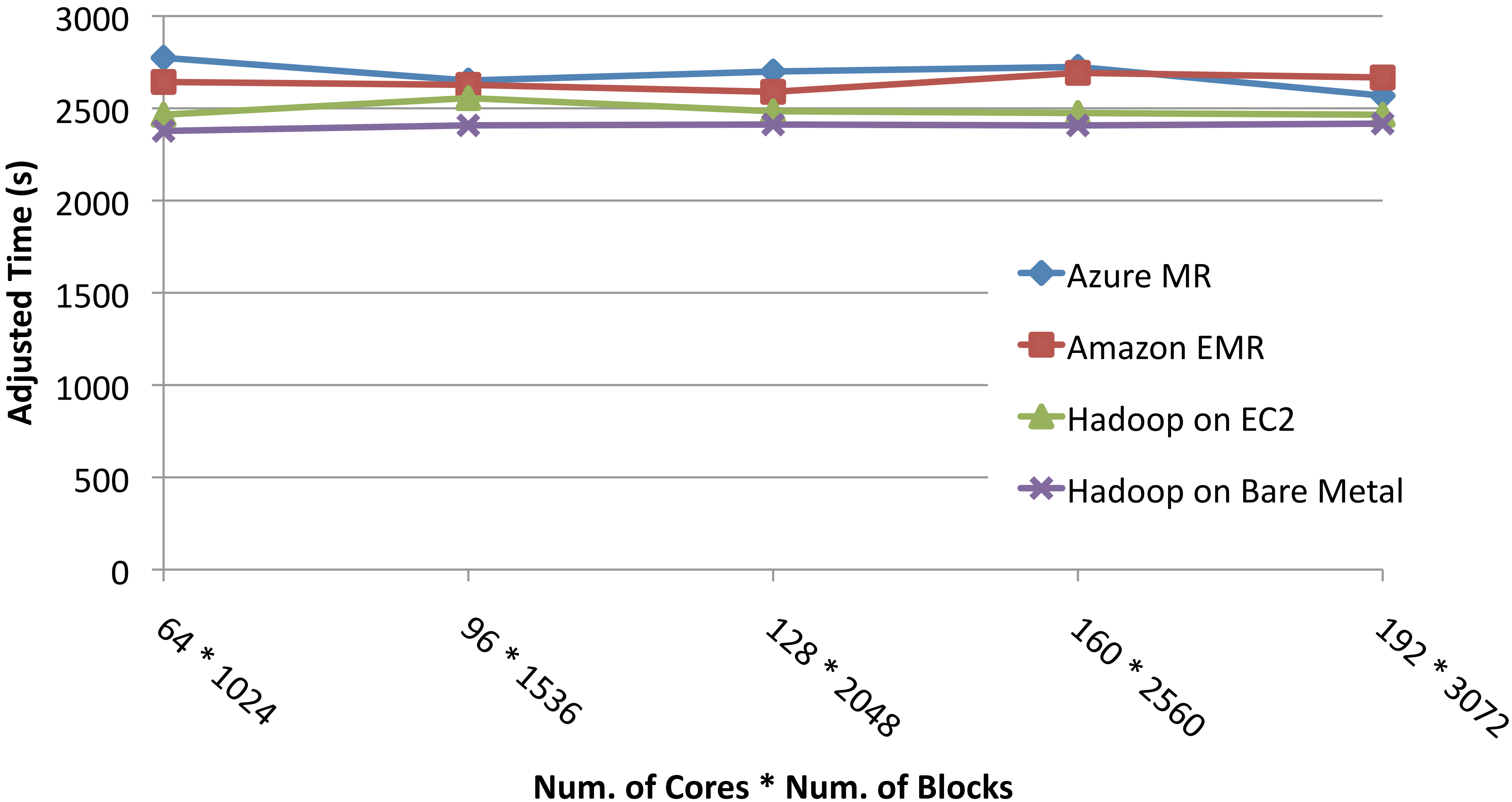


Cap3 Cost

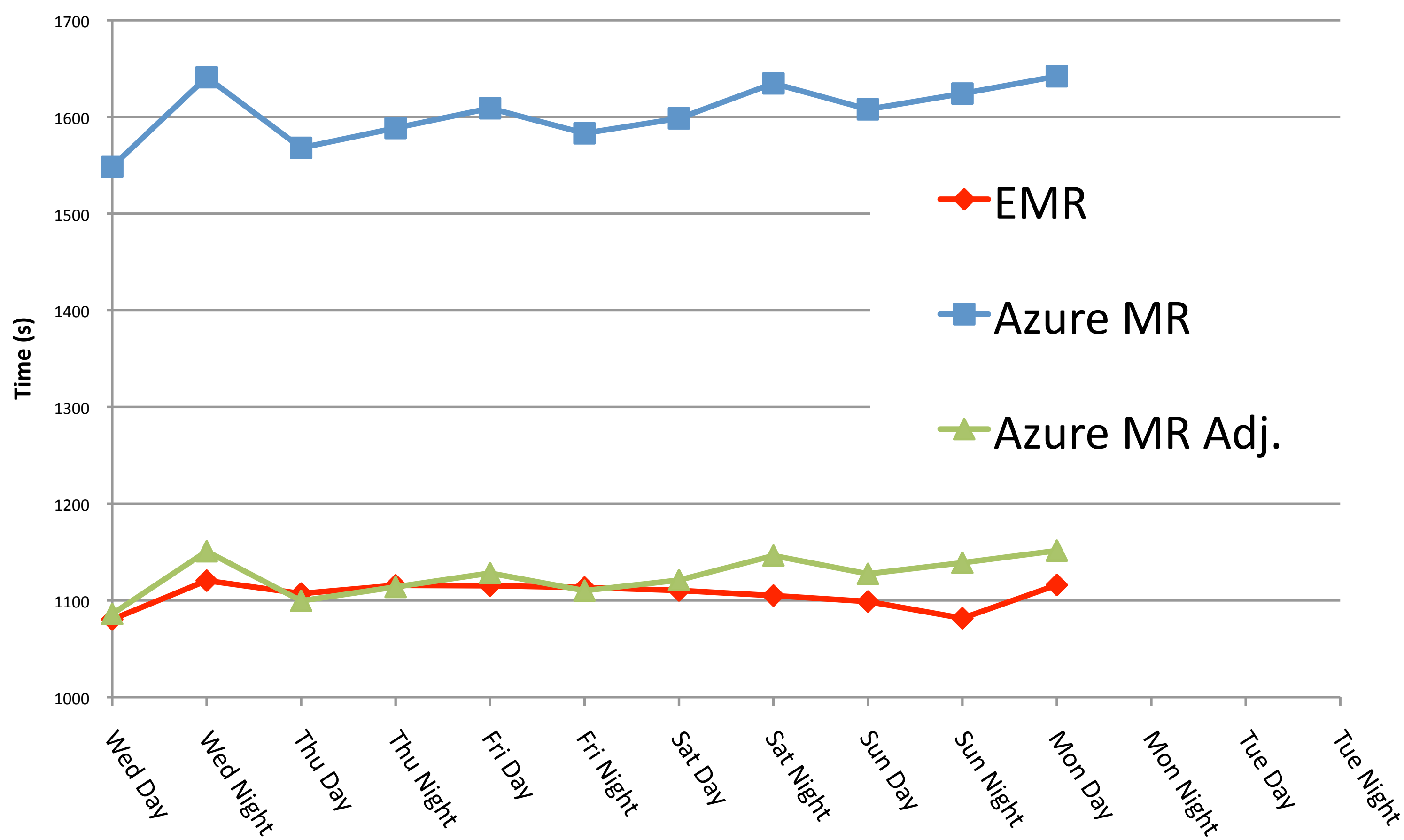


Cost ($)
18
16
14
12
10
8
6
4
2
0
64 * 1024
96 * 1536
128 * 2048
160 * 2560
192 * 3072
Num. Cores * Num. Files
Azure MapReduce
Amazon EMR
Hadoop on EC2

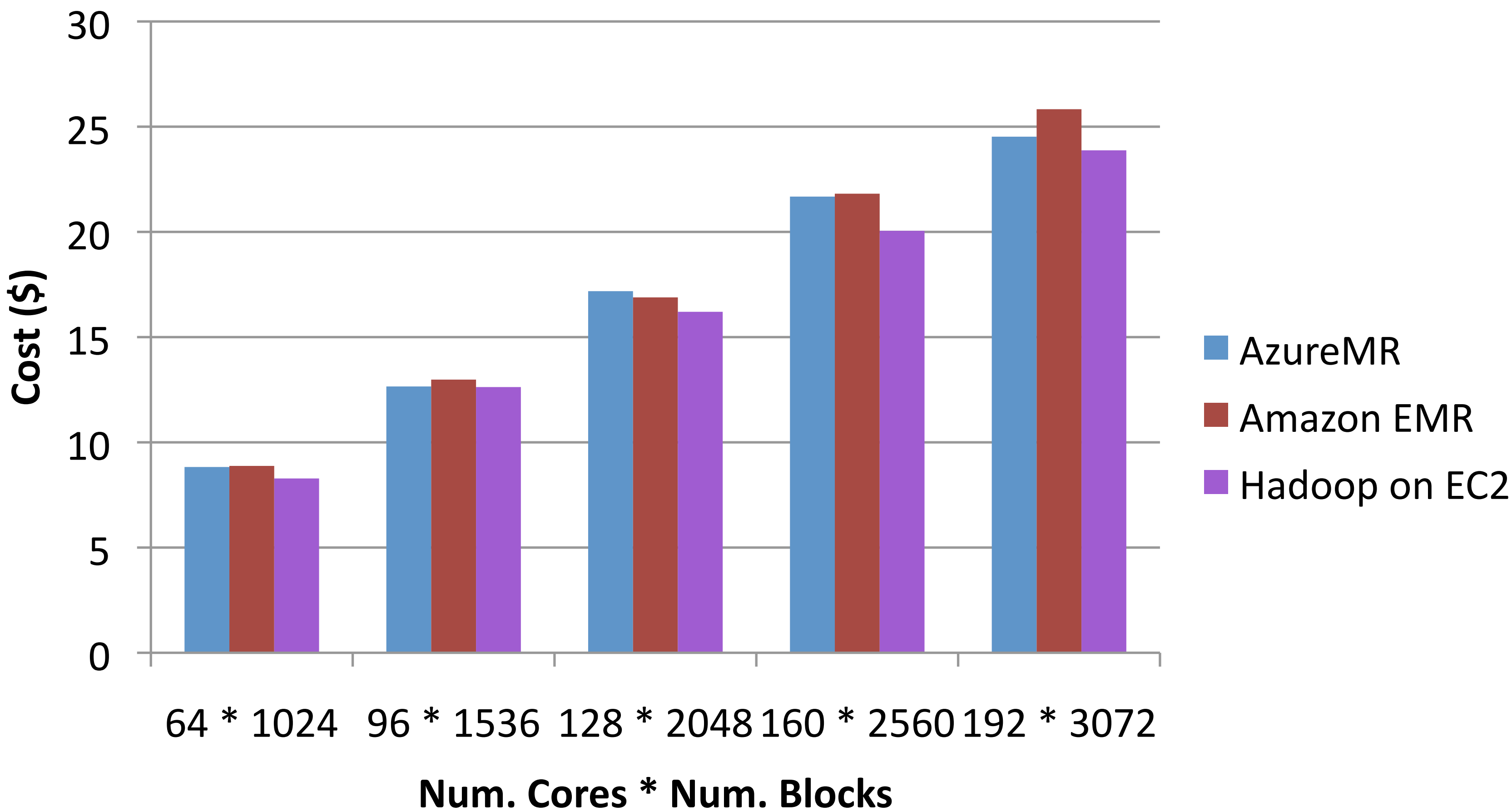
Smith Waterman: "Scaled Speedup" Timing



Smith Waterman: daily effect

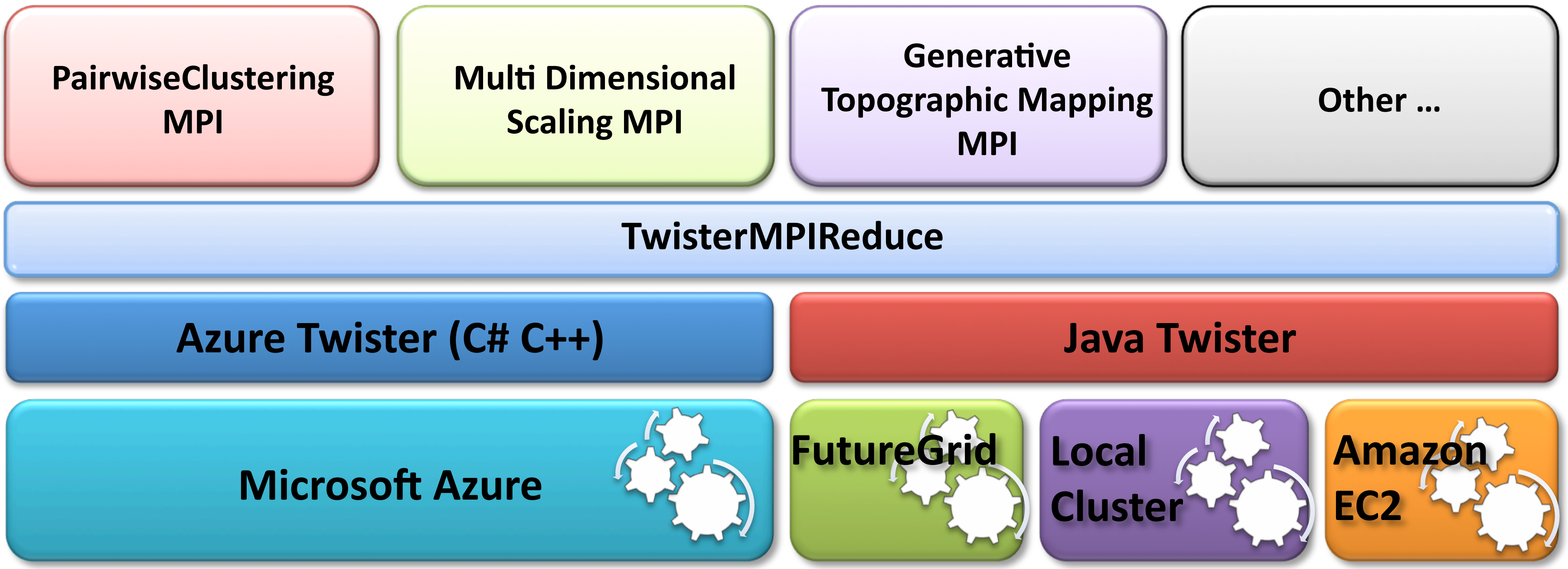


SWG Cost



30
25
20
15
10
5
0
Cost ($)
64 * 1024
96 * 1536
128 * 2048
160 * 2560
192 * 3072
Num. Cores * Num. Blocks
AzureMR
Amazon EMR
Hadoop on EC2

TwisterMPIReduce



- Runtime package supporting subset of MPI mapped to Twister
- Set-up, Barrier, Broadcast, Reduce

Some Issues with AzureTwister and AzureMapReduce

- Transporting data to Azure: Blobs (HTTP), Drives (GridFTP etc.), Fedex disks
- Intermediate data Transfer: Blobs (current choice) versus Drives (should be faster but don't seem to be)
- Azure Table v Azure SQL: Handle all metadata
- Messaging Queues: Use real publish-subscribe system in place of Azure Queues to get scaling (?) with multiple brokers – especially AzureTwister
- Azure Affinity Groups: Could allow better data-compute and compute-compute affinity

Research Issues

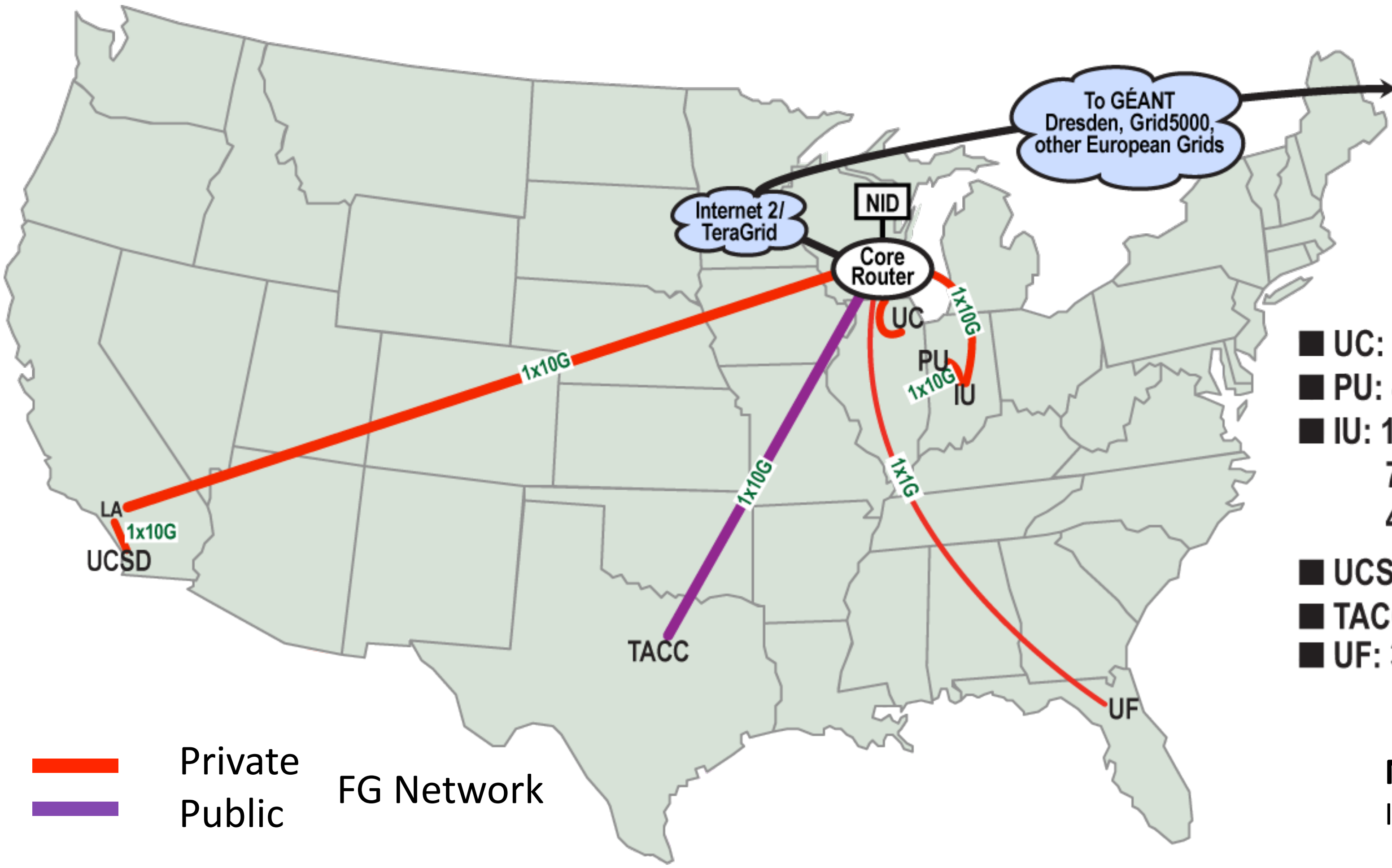
- Clouds are suitable for “Loosely coupled” data parallel applications
- “Map Only” (really pleasingly parallel) certainly run well on clouds (subject to data affinity) with many programming paradigms
- Parallel FFT and adaptive mesh PDE solver very bad on MapReduce but suitable for classic MPI engines.
- MapReduce is more dynamic and fault tolerant than MPI; it is simpler and easier to use
- Is there an intermediate class of problems for which Iterative MapReduce useful?
  - Long running processes?
  - Mutable data small in size compared to fixed data(base)?
  - Only support reductions?
  - Is it really different from a fault tolerant MPI?
  - Multicore implementation
  - Link to HDFS or equivalent data parallel file system
  - Will AzureTwister run satisfactorily?

FutureGrid in a Nutshell

- FutureGrid provides a testbed with a wide variety of computing services to its users
  - Supporting users developing new applications and new middleware using Cloud, Grid and Parallel computing (Hypervisors – Xen, KVM, ScaleMP, Linux, Windows, Nimbus, Eucalyptus, Hadoop, Globus, Unicore, MPI, OpenMP …)
  - Software supported by FutureGrid or users
  - ~5000 dedicated cores distributed across country
- The FutureGrid testbed provides to its users:
  - A rich development and testing platform for middleware and application users looking at interoperability, functionality and performance
  - A rich education and teaching platform for advanced cyberinfrastructure classes
- Each use of FutureGrid is an experiment that is reproducible
- Cloud infrastructure supports loading of general images on Hypervisors like Xen; FutureGrid dynamically provisions software as needed onto "bare-metal" using Moab/xCAT based environment

FutureGrid: a Grid/Cloud Testbed

- **Operational: IU** Cray operational; **IU** , **UCSD**, **UF** & **UC** IBM iDataPlex operational
- **Network**, **NID** operational
- **TACC** Dell running acceptance tests – ready ~September 15

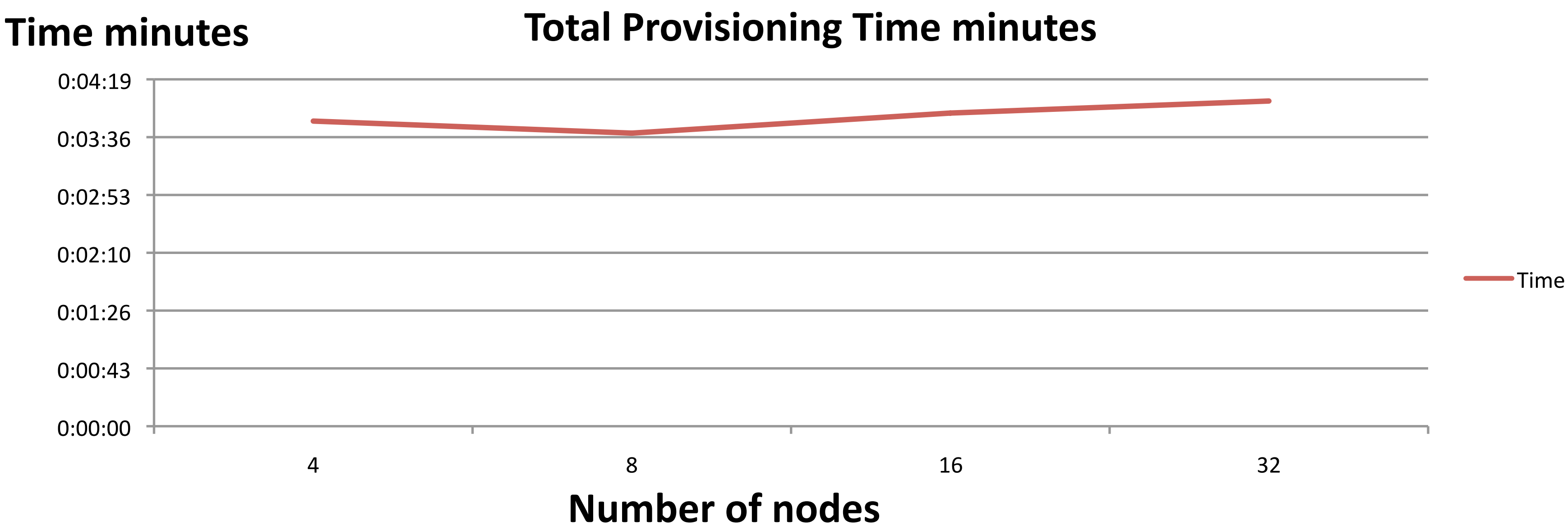


- UC: 7TF IBM 672 cores
- PU: 4TF Dell 384 cores
- IU: 11TF IBM 1024 cores
  7TF Cray 684 cores
  4TF Shared Memory
- UCSD: 7TF IBM 672 cores
- TACC: 12TF Dell 1152 cores
- UF: 3TF IBM 256 cores

Private
Public
FG Network

NID: Network Impairment Device

FutureGrid Dynamic Provisioning Results



Time elapsed between requesting a job and the jobs reported start time on the provisioned node. The numbers here are an average of 2 sets of experiments.

200 papers submitted to main track; 4 days of tutorials