Azure MapReduce

Thilina Gunarathne

Salsa group, Indiana University

Agenda

- Recap of Azure Cloud Services
- Recap of MapReduce
- Azure MapReduce Architecture
- Pairwise distance alignment implementation
- Next steps

Cloud Computing

- On demand computational services over web
  - Backed by massive commercial infrastructures giving economies of scale
  - Spiky compute needs of the scientists
- Horizontal scaling with no additional cost
  - Increased throughput
- Cloud infrastructure services
  - Storage, messaging, tabular storage
  - Cloud oriented services guarantees
  - Virtually unlimited scalability
- Future seems to be **CLOUDY!!!**

Azure Platform

- Windows Azure Compute
  - .net platform as a service
  - Worker roles & web roles
- Azure Storage
  - Blobs
  - Queues
  - Table
- Development SDK, fabric and storage

MapReduce

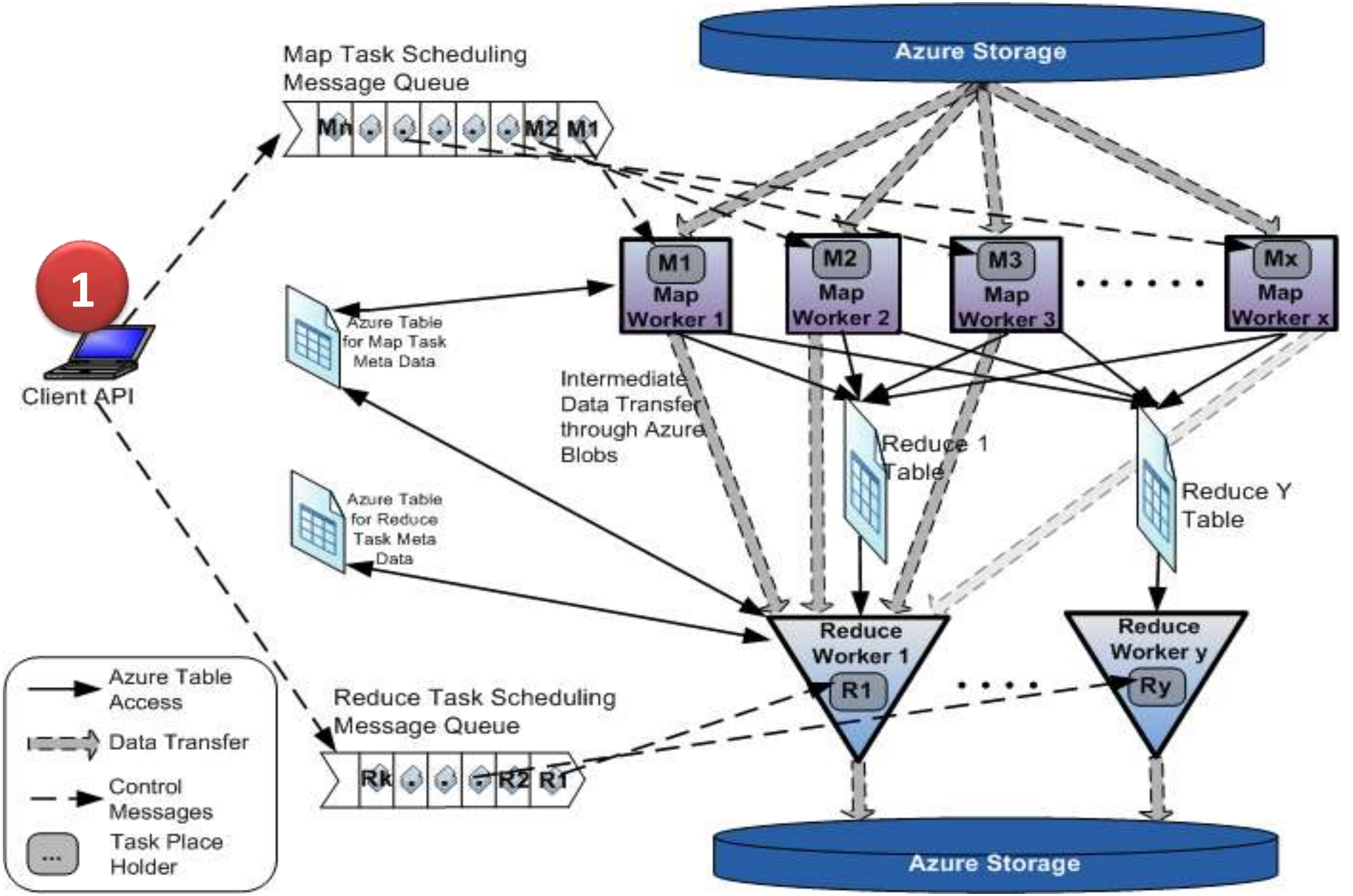
- Automatic parallelization & distribution
- Fault-tolerant
- Provides status and monitoring tools
- Clean abstraction for programmers
  - **`map  (in_key, in_value) -> (out_key, intermediate_value) list`**
  - **`reduce (out_key, intermediate_value list) -> out_value list`**

Motivation

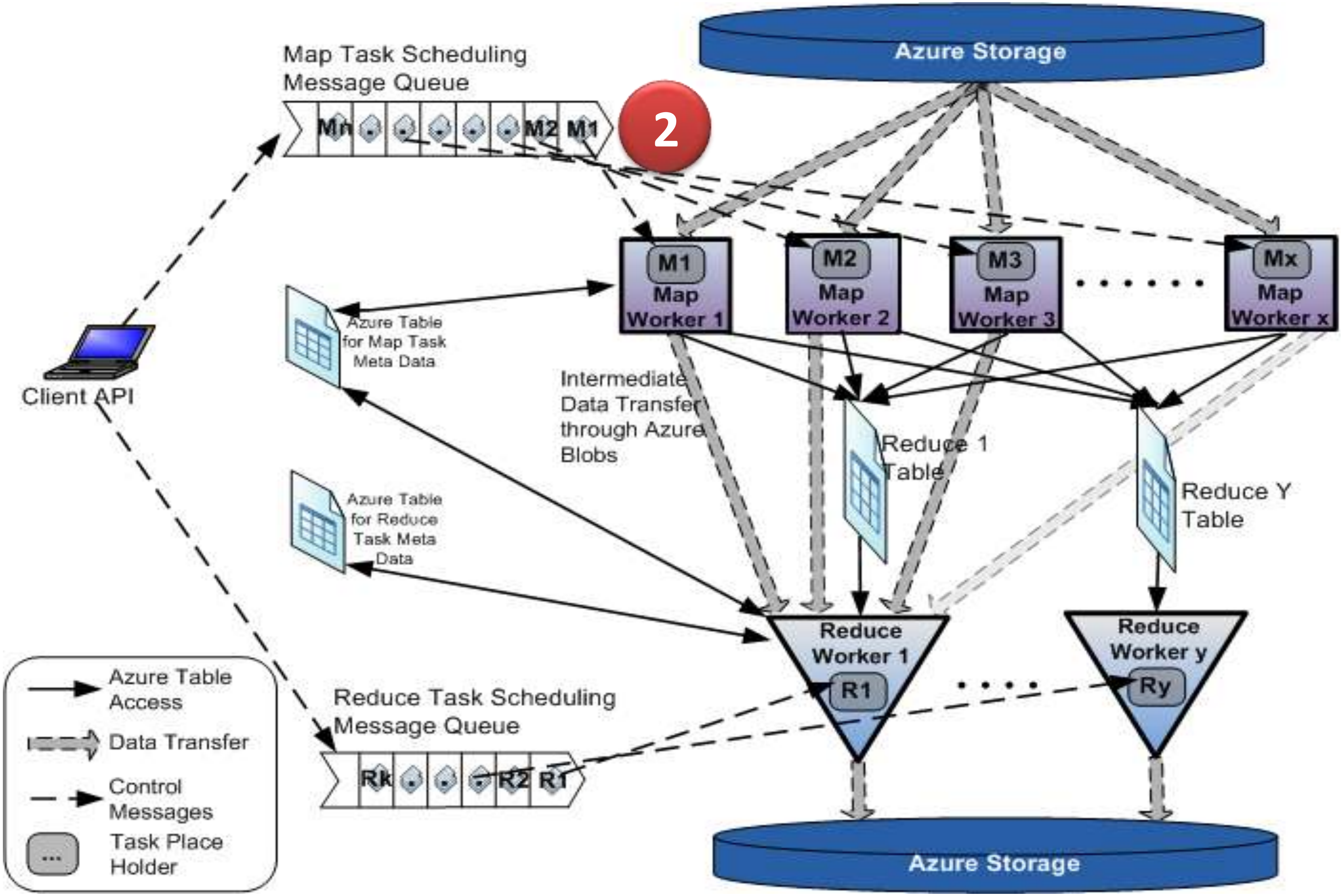
- Currently no parallel programming framework on Azure
  - No MPI, No Dryad
- Well known, easy to use programming model
- Cloud nodes are not as reliable as conventional cluster nodes

Azure MapReduce Concepts

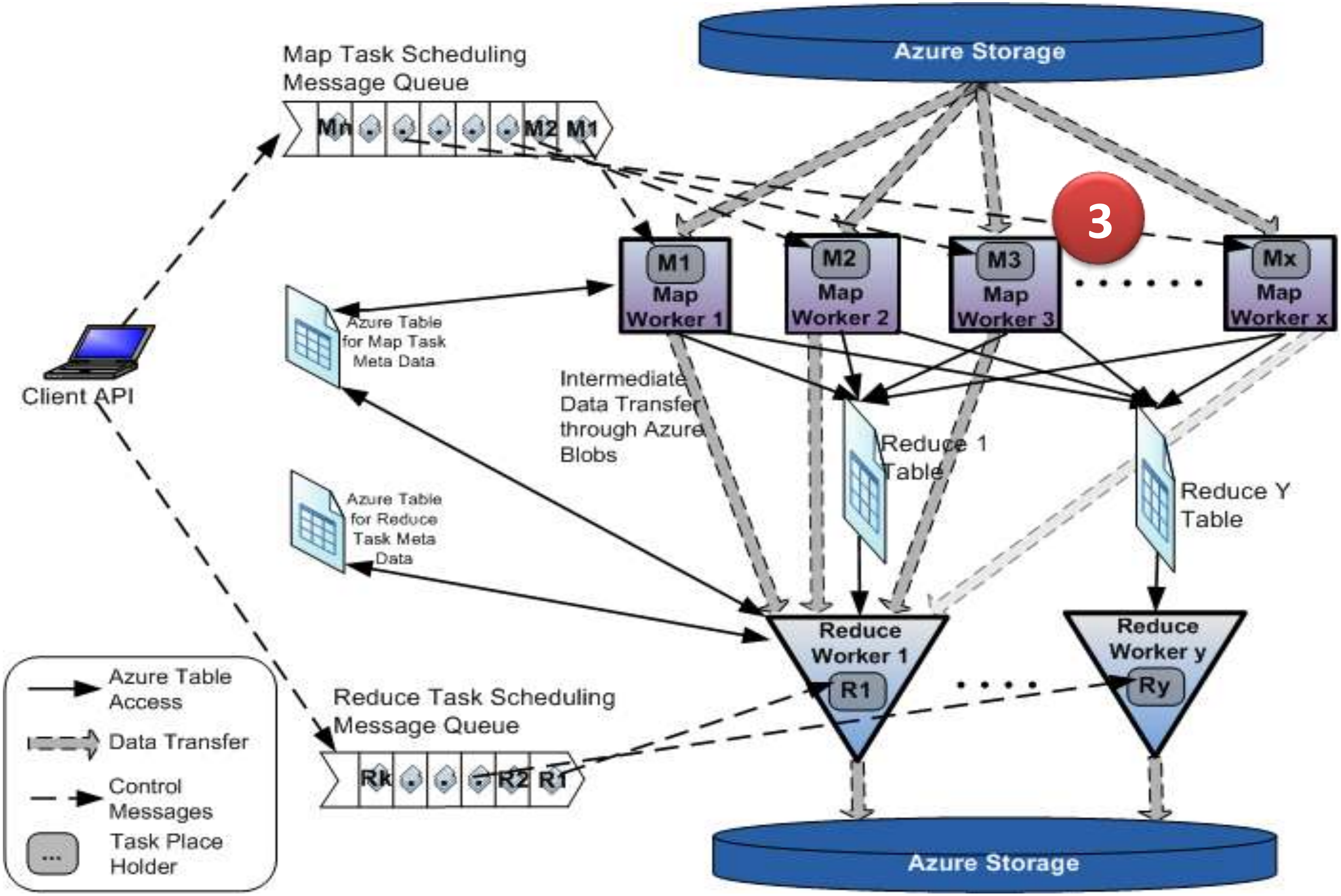
- Take advantage of the cloud services
  - Distributed services, Unlimited scalability
  - Backed by industrial strength data centers and technologies
- Decentralized control
- Dynamically scale up/down
- Eventual consistency
- Large latencies
  - Coarser grained map tasks
- Global queue based scheduling



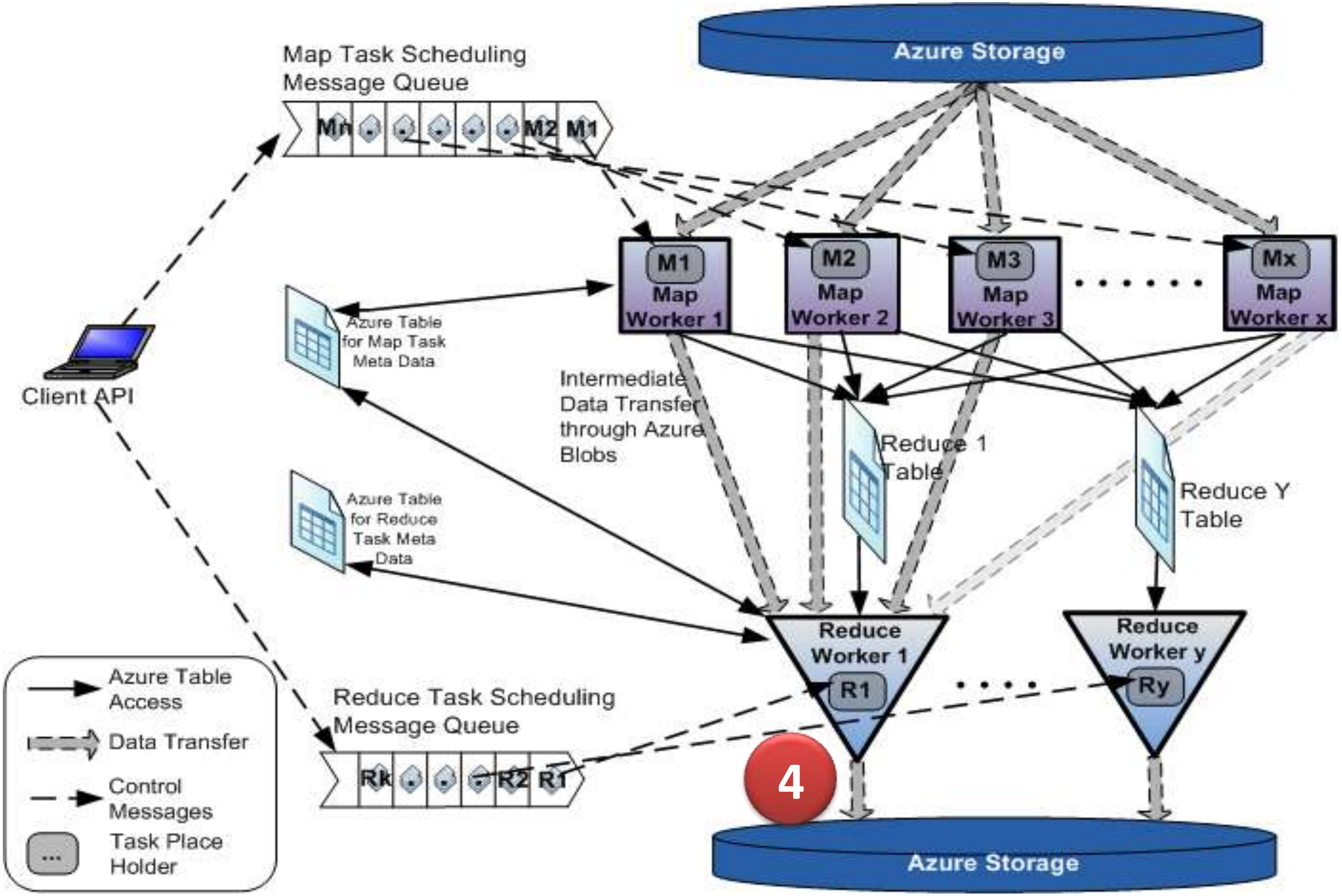
1.Client driver loads the map & reduce tasks to the queues



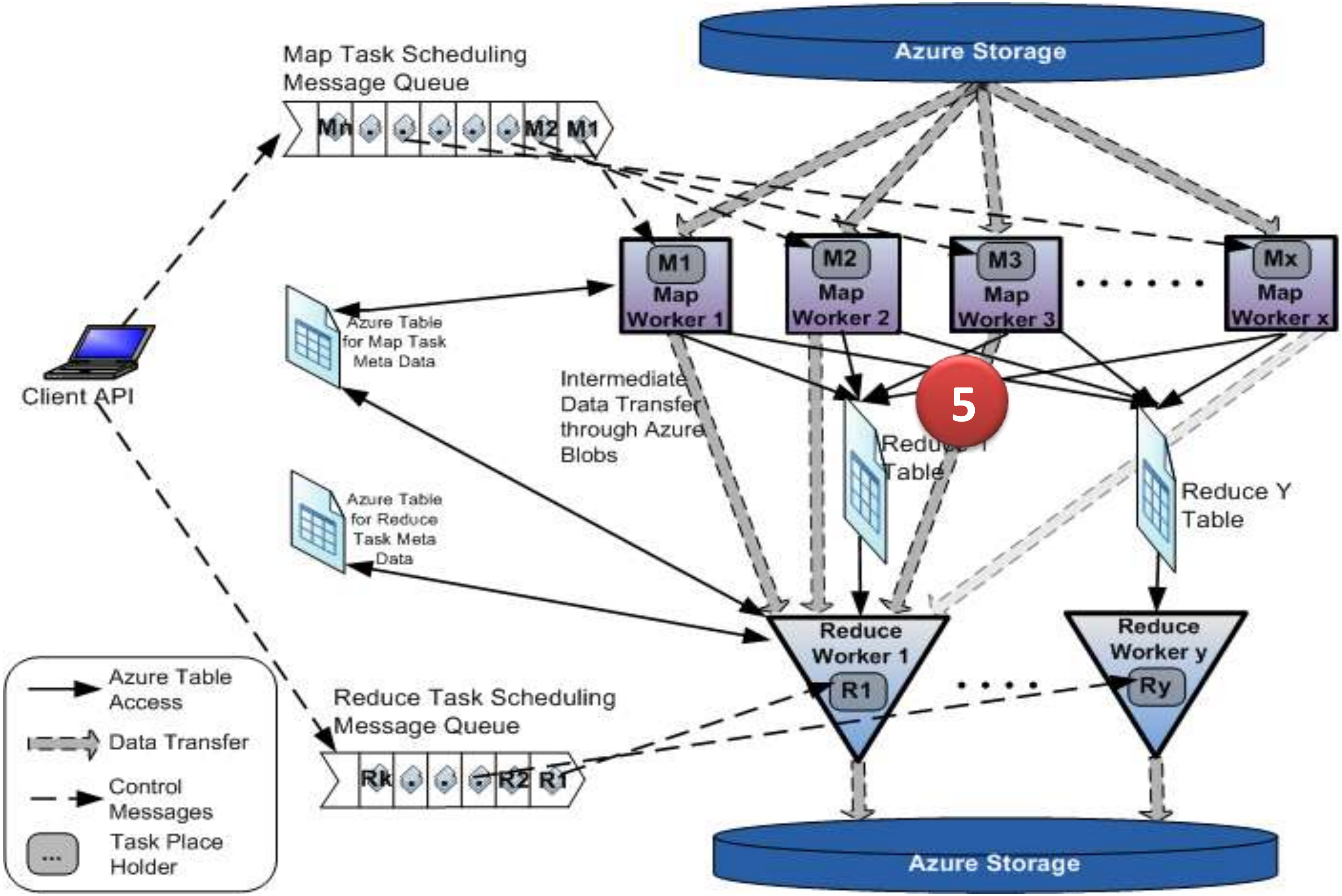
2. Map workers retrieve map tasks from the queue



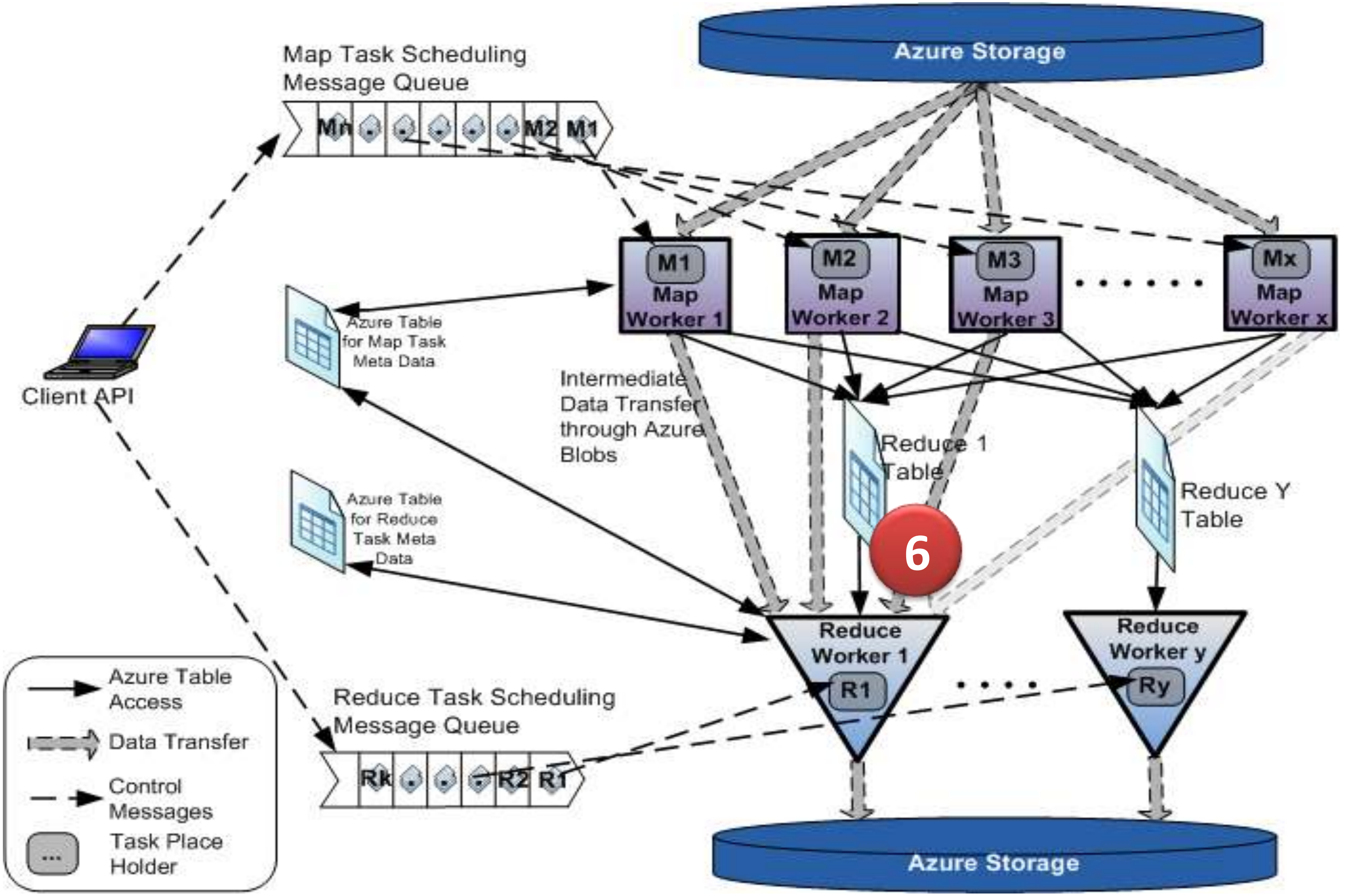
3. Map workers download data from the Blob storage and start processing



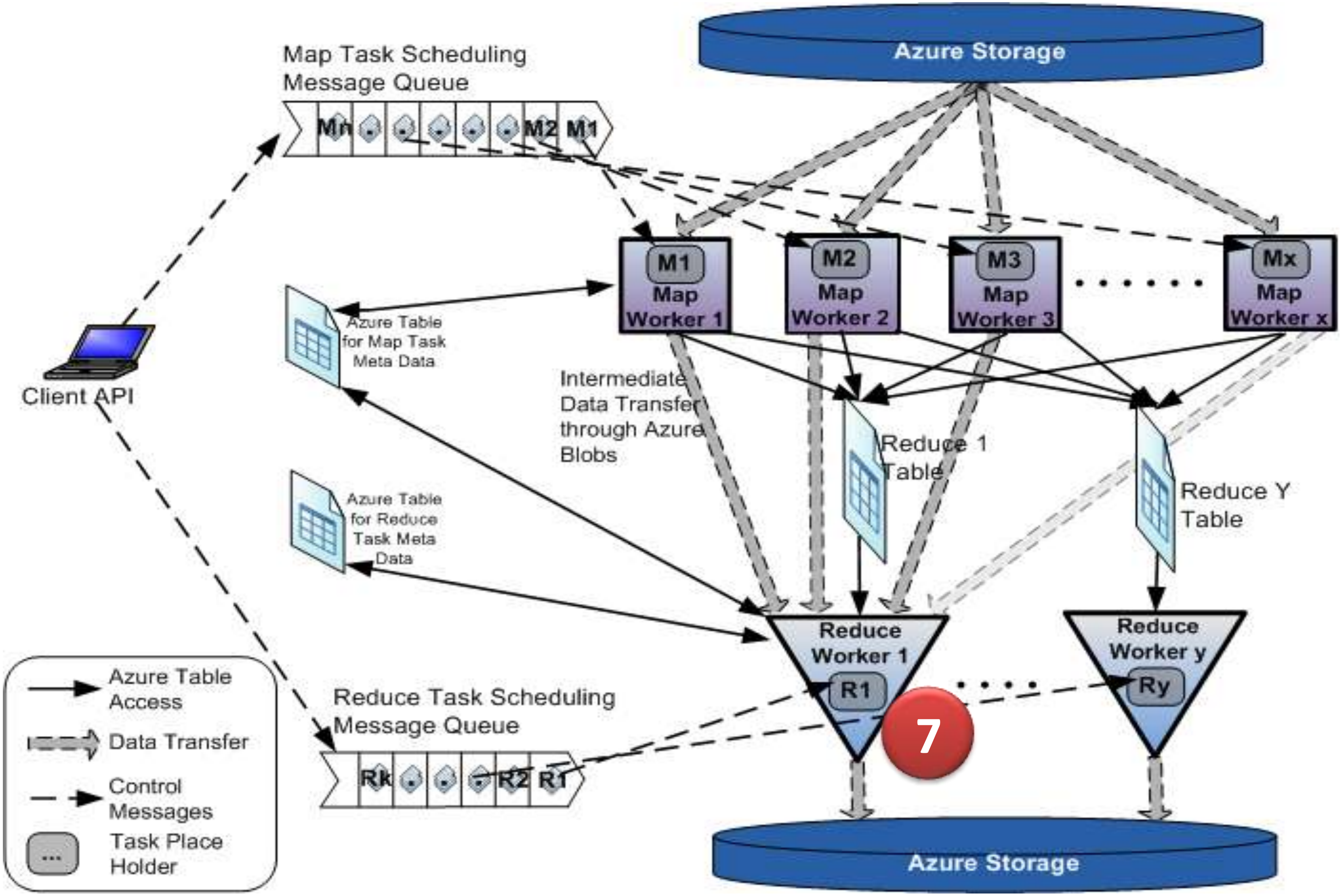
4. Reduce workers pick the tasks from the queue and start monitoring the reduce task tables



5. Finished map tasks upload the results to Blob storage. Add entries to the respective reduce task tables.



6. Reduce tasks download the intermediate data products



7. Start reducing when all the map tasks are finished and when a reduce task is finished downloading the intermediate data products

Azure MapReduce Architecture

- Client API and driver
- Map tasks
- Reduce tasks
- Intermediate data transfer
- Monitoring
- Configurations

Fault tolerance

- Use the visibility timeout of the queues
  - Currently maximum is 3 hours
  - Delete the message from the queue only after everything is successful
  - Execution, upload, update status
- Tasks will rerun when timeout happens
  - Ensures eventual completion
  - Intermediate data are persisted in blob storage
  - Retry up to 3 times
- Many retries in service invocations

	Apache Hadoop [24] /(Google MR)	Microsoft Dryad [25]	Twister [19]	Azure Map Reduce/Twister
Programming Model	MapReduce	DAG execution, Extensible to MapReduce and other patterns	Iterative MapReduce	MapReduce-- will extend to Iterative MapReduce
Data Handling	HDFS (Hadoop Distributed File System)	Shared Directories & local disks	Local disks and data management tools	Azure Blob Storage
Scheduling	Data Locality; Rack aware, Dynamic task scheduling through global queue	Data locality; Network topology based run time graph optimizations; Static task partitions	Data Locality; Static task partitions	Dynamic task scheduling through global queue
Failure Handling	Re-execution of failed tasks; Duplicate execution of slow tasks	Re-execution of failed tasks; Duplicate execution of slow tasks	Re-execution of Iterations	Re-execution of failed tasks; Duplicate execution of slow tasks
Environment	Linux Clusters, Amazon Elastic Map Reduce on EC2	Windows HPCS cluster	Linux Cluster EC2	Window Azure Compute, Windows Azure Local Development Fabric
Intermediate data transfer	File, Http	File, TCP pipes, shared-memory FIFOs	Publish/Subscribe messaging	Files, TCP

Why Azure Services

- Virtually unlimited scalable distributed services
- No need to install software stacks
  - In fact you can't ☺
  - Eg: NaradaBrokering, HDFS, Database
- Zero maintenance
  - Let the platform take care of you
- Availability guarantees

API

- ProcessMapRed(jobid, container, params, numReduceTasks, storageAccount, mapQName, reduceQName,List mapTasks)
- Map(key, value, programArgs, Dictionary outputCollector)
- Reduce(key, List values, programArgs, Dictionary outputCollector)

Develop applications using Azure MapReduce

- Local debugging using Azure development fabric
- DistributedCache
  - Bundle with Azure Package
- Compile in release mode before creating the package.
- Deploy using Azure web interface
- Errors logged to a Azure Table

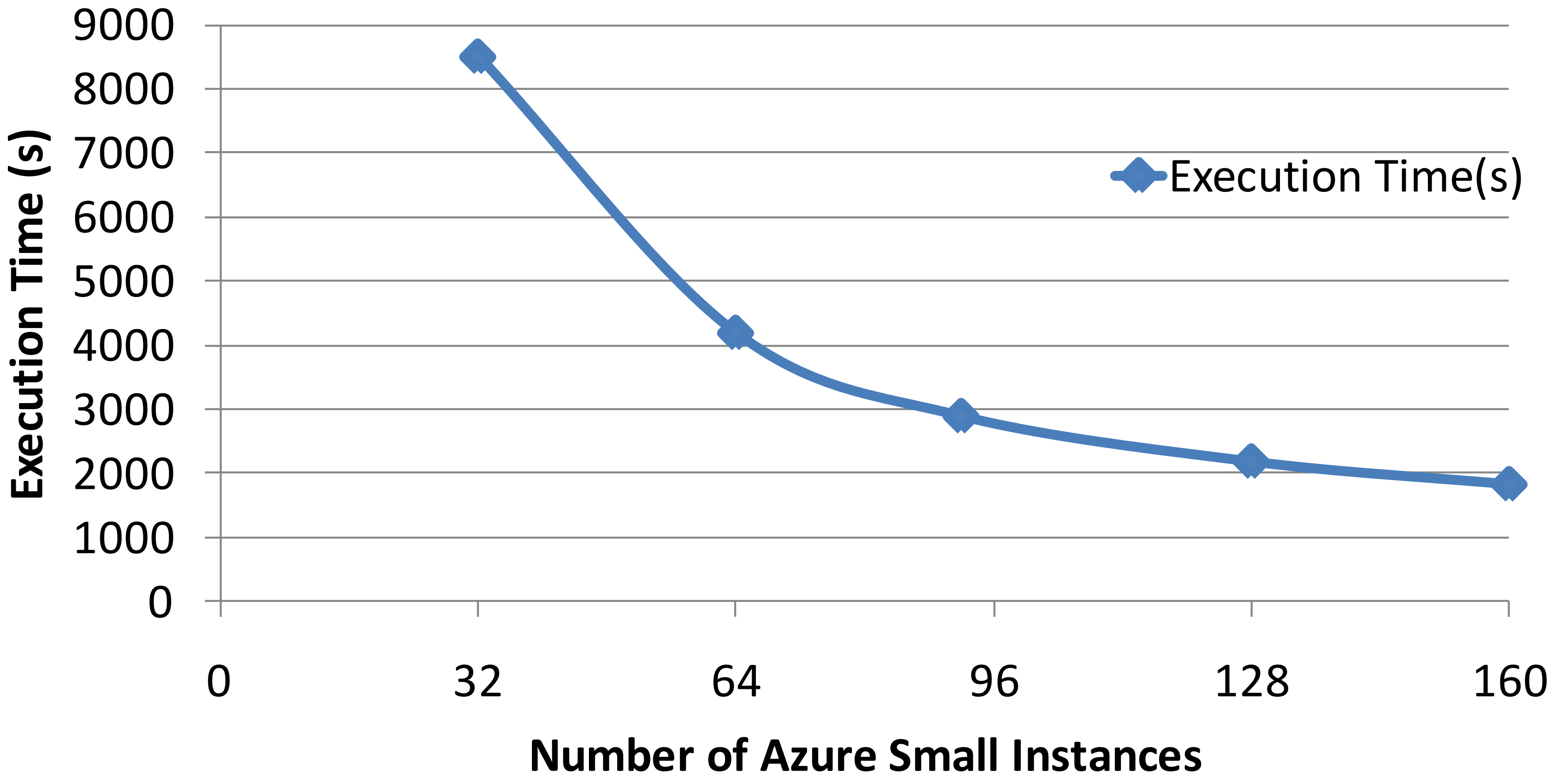
SWG Pairwise Distance Alignment

- SmithWaterman-GOTOH
- Pairwise sequence alignment
  - Align each sequence with all the other sequences

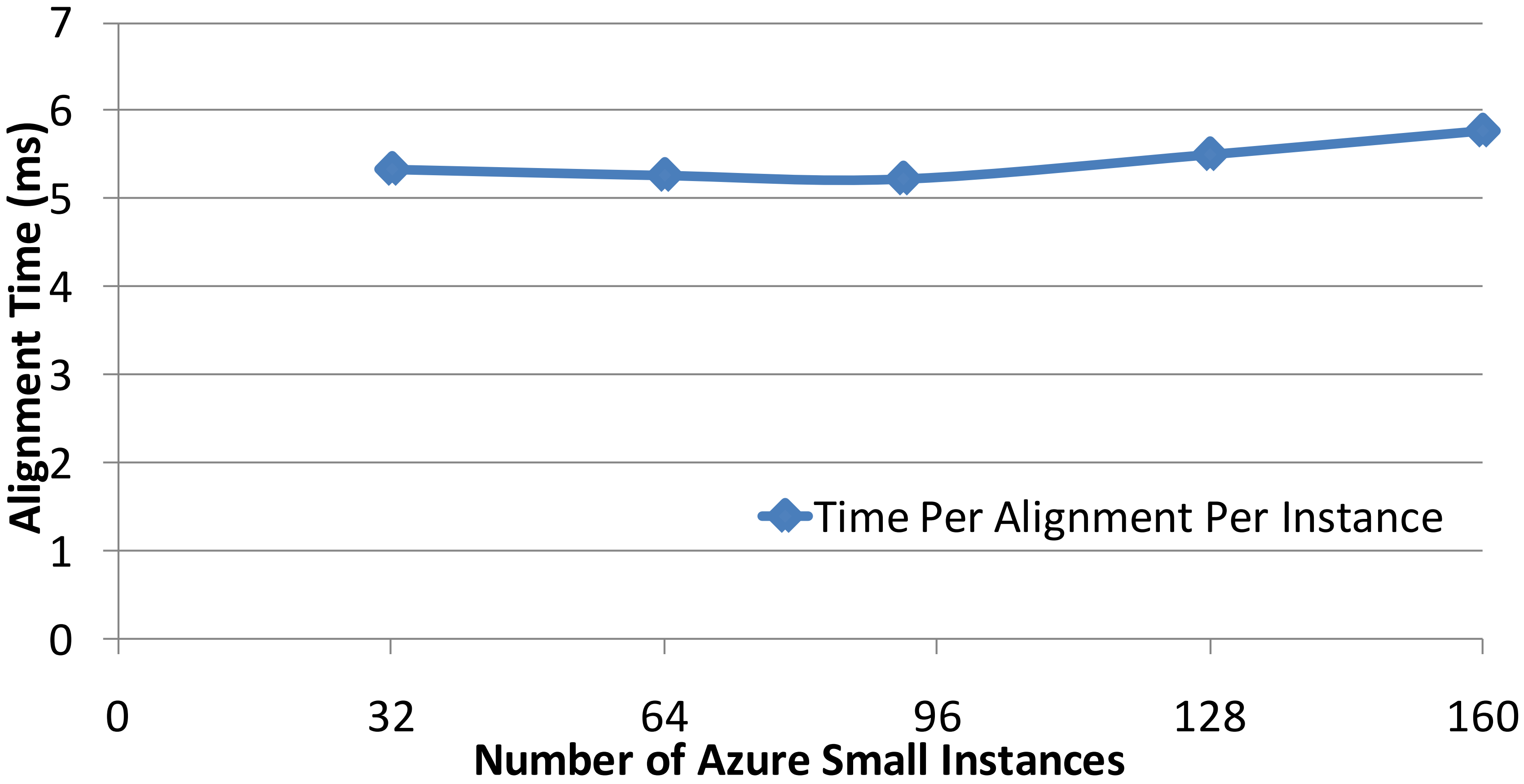
Application architecture
Block decomposition

	1 (1-100)	**2** (101-200)	**3** (201-300)	**4** (301-400)	
1 (1-100)	**M1**	**M2**	*from M6*	**M3**	**Reduce 1**
2 (101-200)	*from M2*	**M4**	**M5**	*from M9*	**Reduce 2**
3 (201-300)	**M6**	*from M5*	**M7**	**M8**	**Reduce 3**
4 (301-400)	*from M3*	**M9**	*from M8*	**M10**	**Reduce 4**

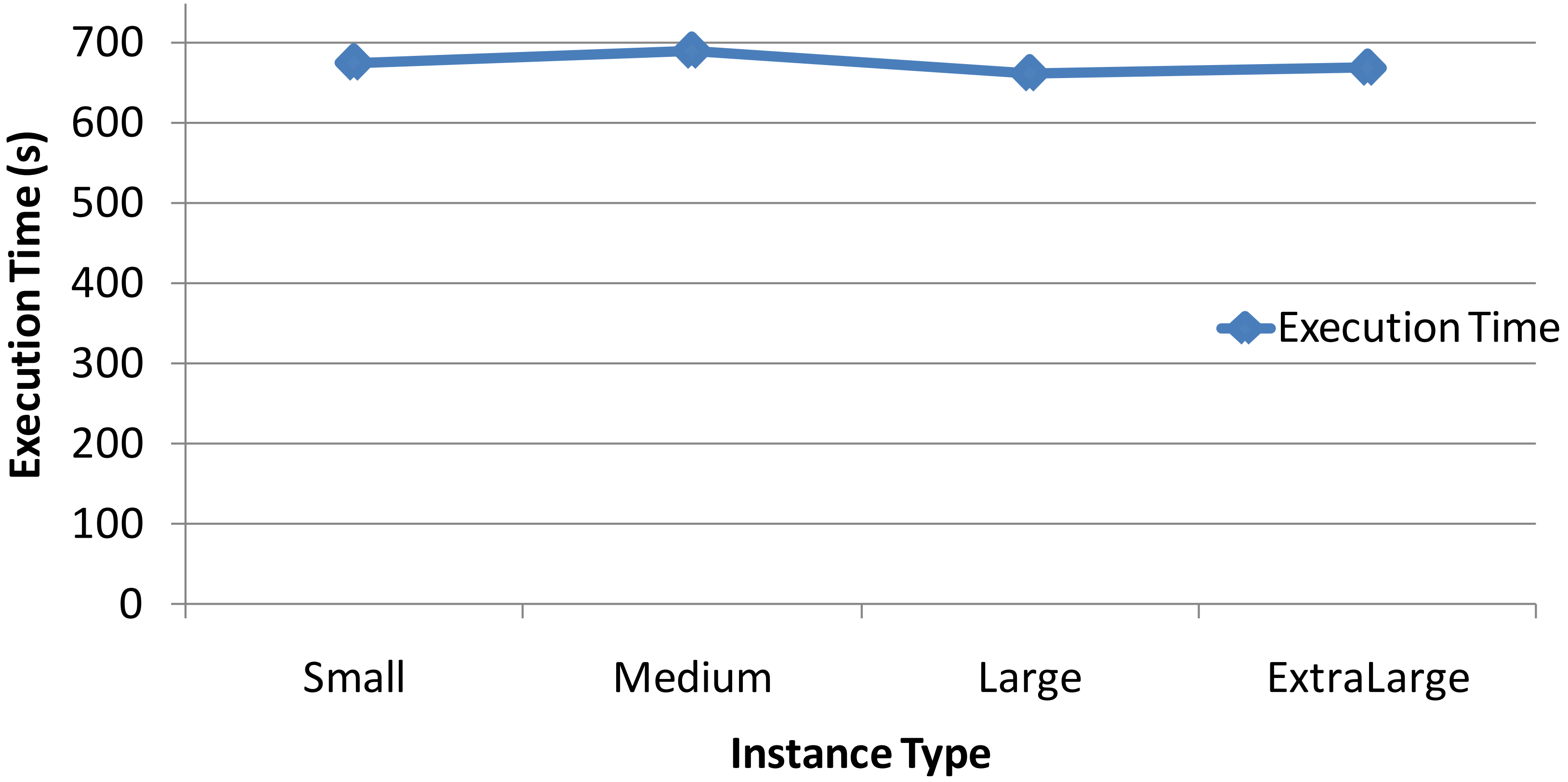
AzureMR SWG Performance 10k Sequences



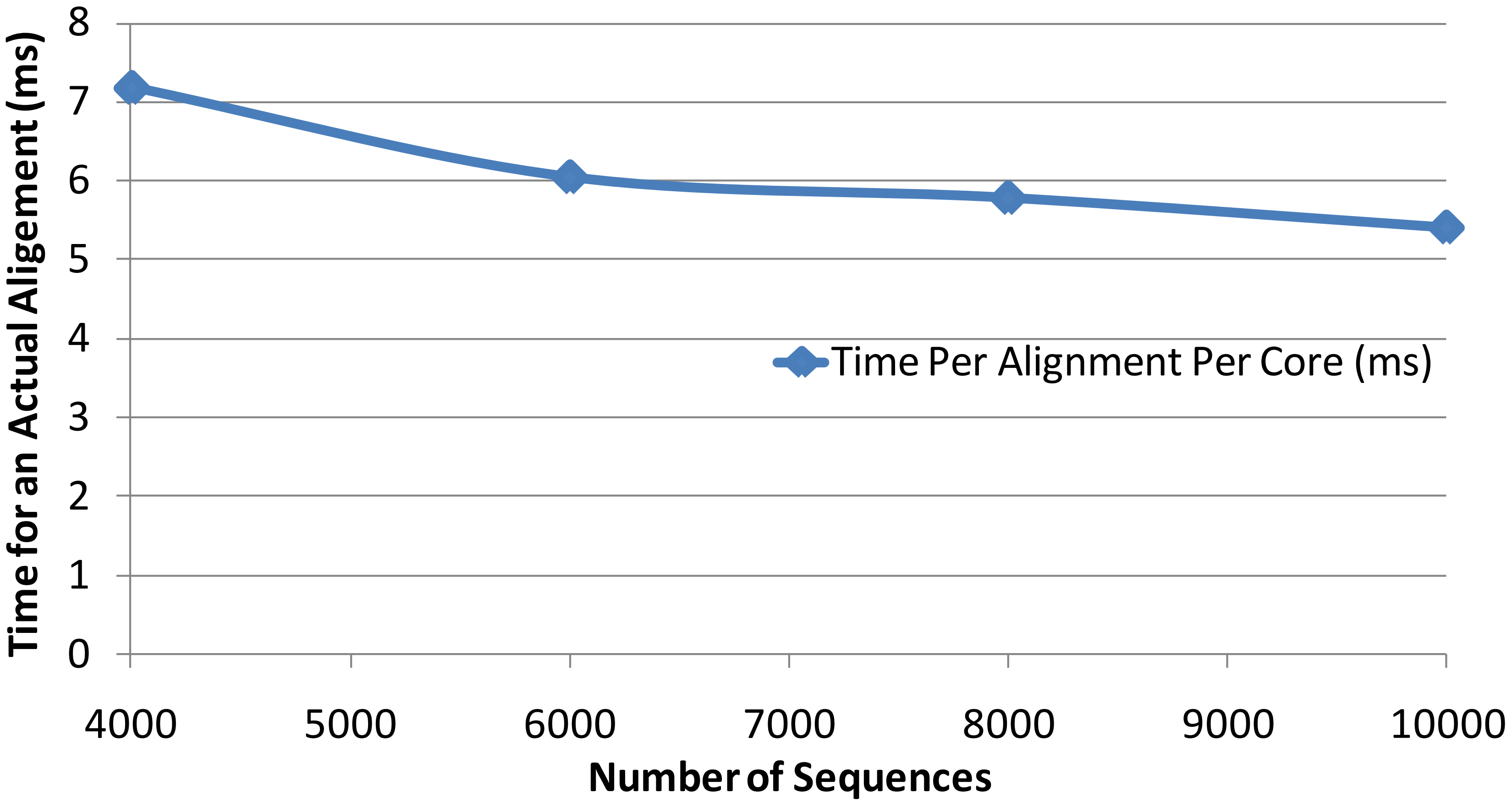
AzureMR SWG Performance 10k Sequences



AzureMR SWG Performance on Different Instance Types



AzureMR SWG Performance on Different Data Sizes



Next Steps

- In the works
  - Monitoring web interface
  - Alternative intermediate data communication mechanisms
  - Public release
- Future plans
  - AzureTwister
    - Iterative MapReduce

Thanks!!

- Questions? ☺

References

- J. Dean, and S. Ghemawat, “MapReduce: simplified data processing on large clusters,” *Commun. ACM,* vol. 51, no. 1, pp. 107-113., 2008.
- J.Ekanayake, H.Li, B.Zhang *et al.*, “Twister: A Runtime for iterative MapReduce,” in Proceedings of the First International Workshop on MapReduce and its Applications of ACM HPDC 2010 conference June 20-25, 2010, Chicago, Illinois, 2010.
- *Cloudmapreduce*, http://sites.google.com/site/huanliu/cloudmapreduce.pdf
- "Apache Hadoop," http://hadoop.apache.org/
- M. Isard, M. Budiu, Y. Yu *et al.*, "Dryad: Distributed data-parallel programs from sequential building blocks." pp. 59-72.

Acknowledgments

- Prof. Geoffrey Fox, Dr. Judy Qiu and the Salsa group
- Dr. Ying Chen and Alex De Luca from IBM Almaden Research Center
- Virtual School Organizers