eScience in the Cloud: A MODIS Satellite Data Reprojection and Reduction Pipeline in the Windows Azure Platform

Jie Li1, Deb Agarwal2, Marty Humphrey1, Keith Jackson2,
Catharine van Ingen3, Youngryel Ryu4

University of Virginia eScience Group1
Lawrence Berkeley National Lab2
Microsoft Research3
University of California, Berkeley4

IPDPS - April 20, 2010

Outline

Background

AzureMODIS Framework Overview

Dynamic Scalability & Fault Tolerance

Evaluation

Conclusions & Future Work

Data-intensive eScience: Opportunities

Increasing data availability for science discoveries

Growing data size from large scientific instruments

Emerging large-scale inexpensive ground-based sensors

Computational models with increasing complexities and precisions

MODIS Basics

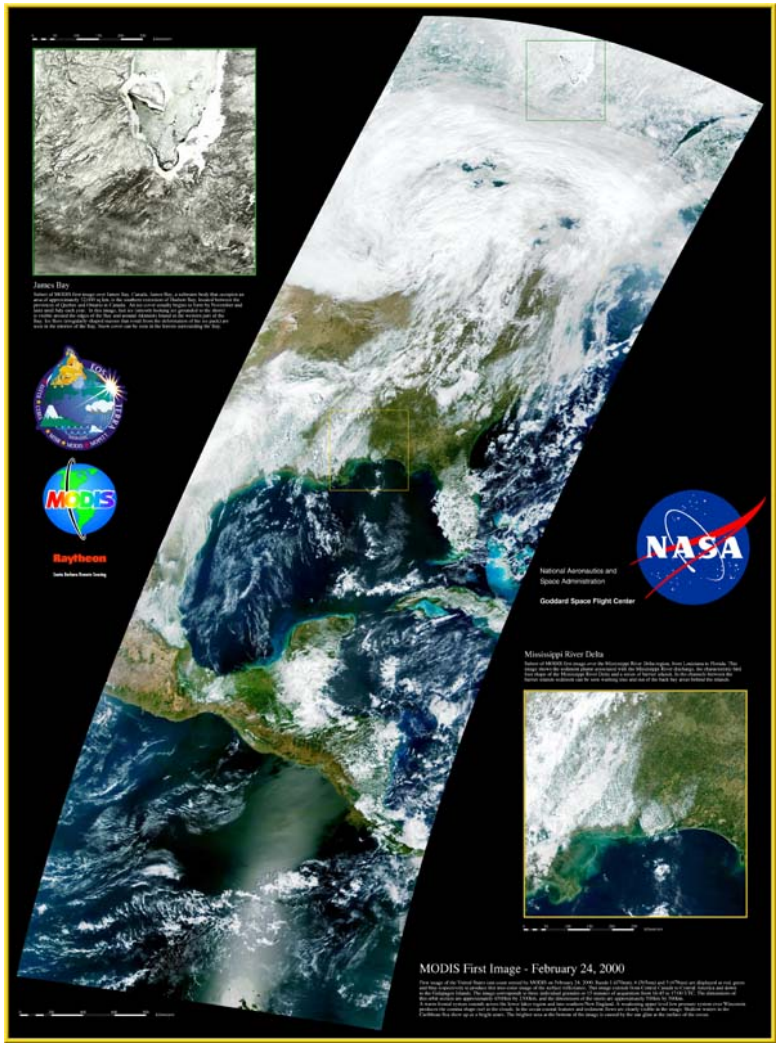
Moderate Resolution Imaging Spectroradiometer Satellites:

Viewing the entire Earth's surface every 1 to 2 days

Acquiring data in 36 spectral bands

Multiple data products (Atmosphere, Land, Ocean etc.)

Important for understanding global environment and earth system models



http://aqua.nasa.gov/doc/viz/media/aqua_orbit_sm.mpg

Barriers for Using MODIS Data

Data Collection

Multiple FTP sites for MODIS source data

Metadata maintained separately

Data Heterogeneity

Different time granularities and imaging resolutions

Two different project types: “*Swath*” and “*Sinusoidal*”

Data Management

Current use case: 10 years of data covering US continent

5 TB source data (~600,000 files)

2 TB timeframe- and space-aligned harmonized data

~50000 CPU hours of parallel computation

AzureMODIS: A Client+Cloud Solution

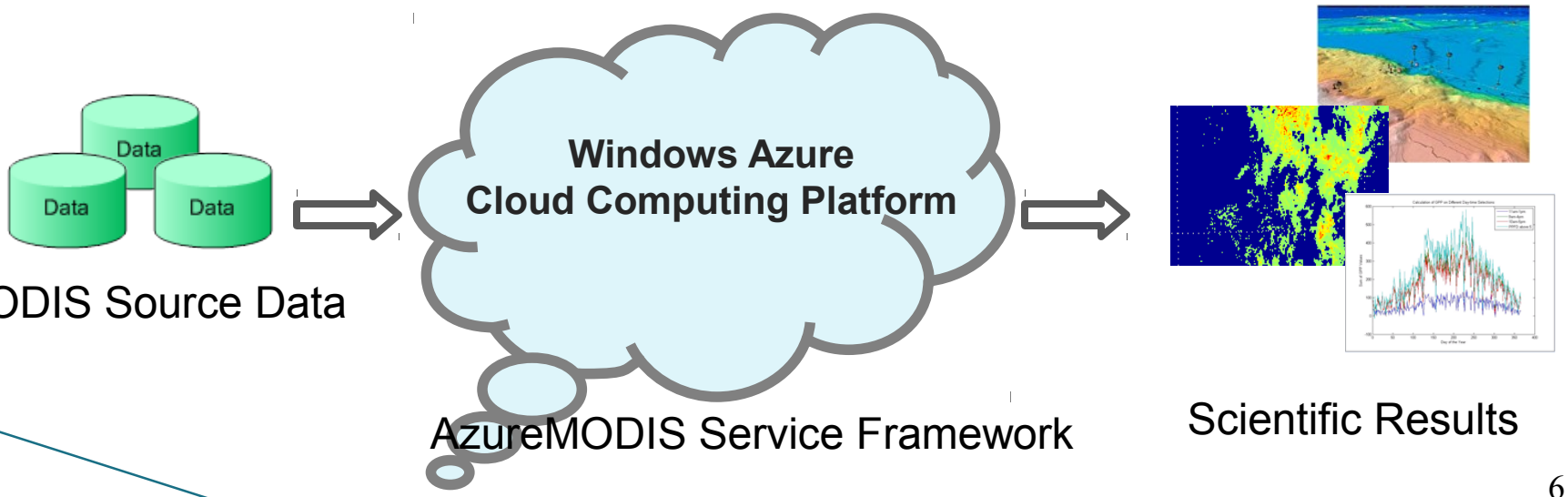
A MODIS Data Processing Framework in Microsoft Windows Azure cloud computing platform

Leverage scalability of cloud infrastructure and services

Dynamic, on-demand resource provisioning

Automate data processing tasks to eliminate barriers

A generic *Reduction Service* to run arbitrary analysis executables



Outline

Background

AzureMODIS Framework Overview

Dynamic Scalability & Fault Tolerance

Evaluation

Conclusions & Future Work

Windows Azure Platform Basics

Hosted Services

Web Role: Host web applications via an HTTP and/or an HTTPS endpoint

Worker Role: Host user-customized code/applications

Storage Services

Blob service: Storage for entities in the form of binary bits

Queue Service: A reliable, persistent queue model for message-based communication between instances

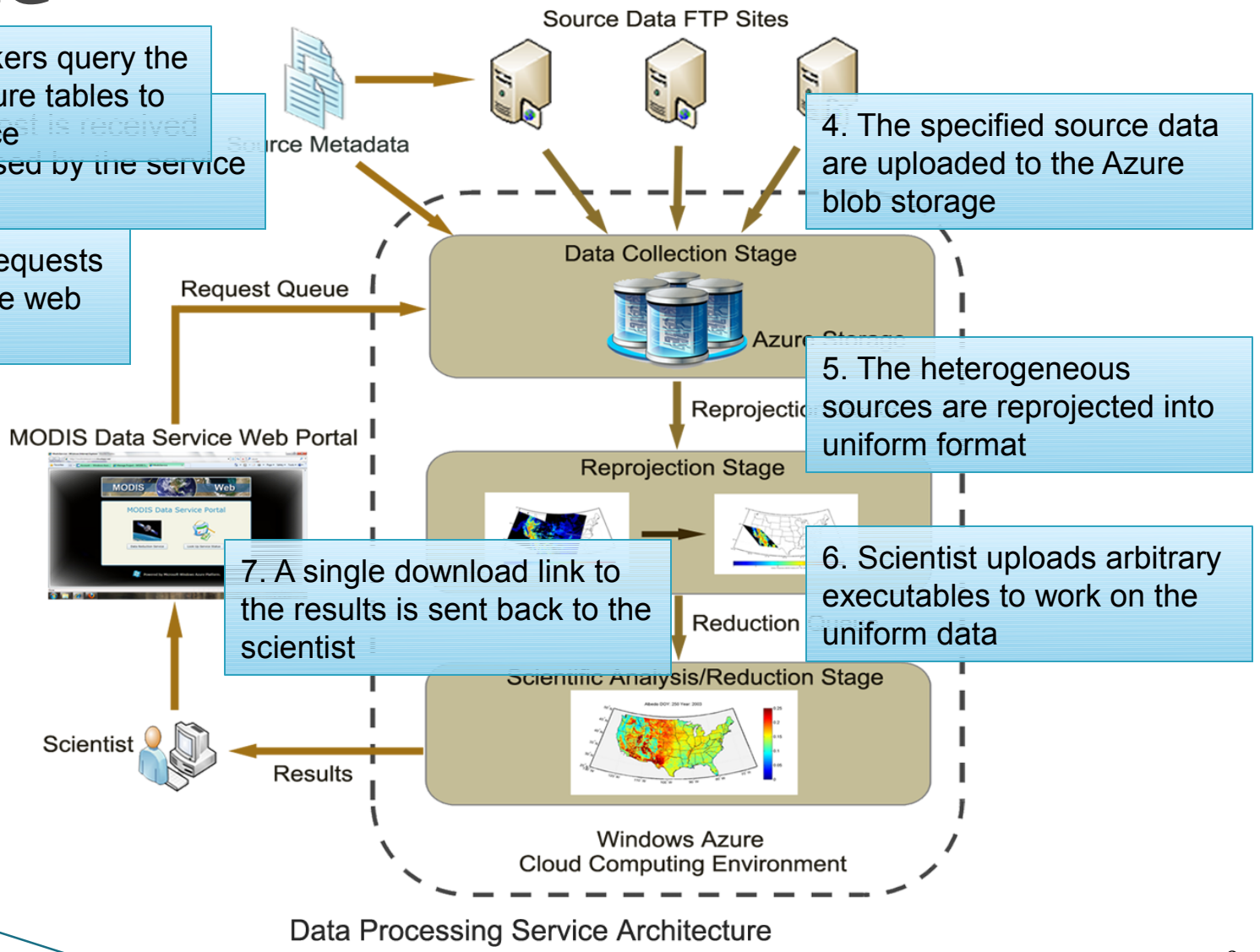
Table Service: Structured storage in the form of tables, with simple query support

AzureMODIS Data Processing Service

3. Service Workers query the metadata in Azure tables to download source

monitor

1. Scientist submits requests for computation on the web portal



4. The specified source data are uploaded to the Azure blob storage

5. The heterogeneous sources are reprojected into uniform format

6. Scientist uploads arbitrary executables to work on the uniform data

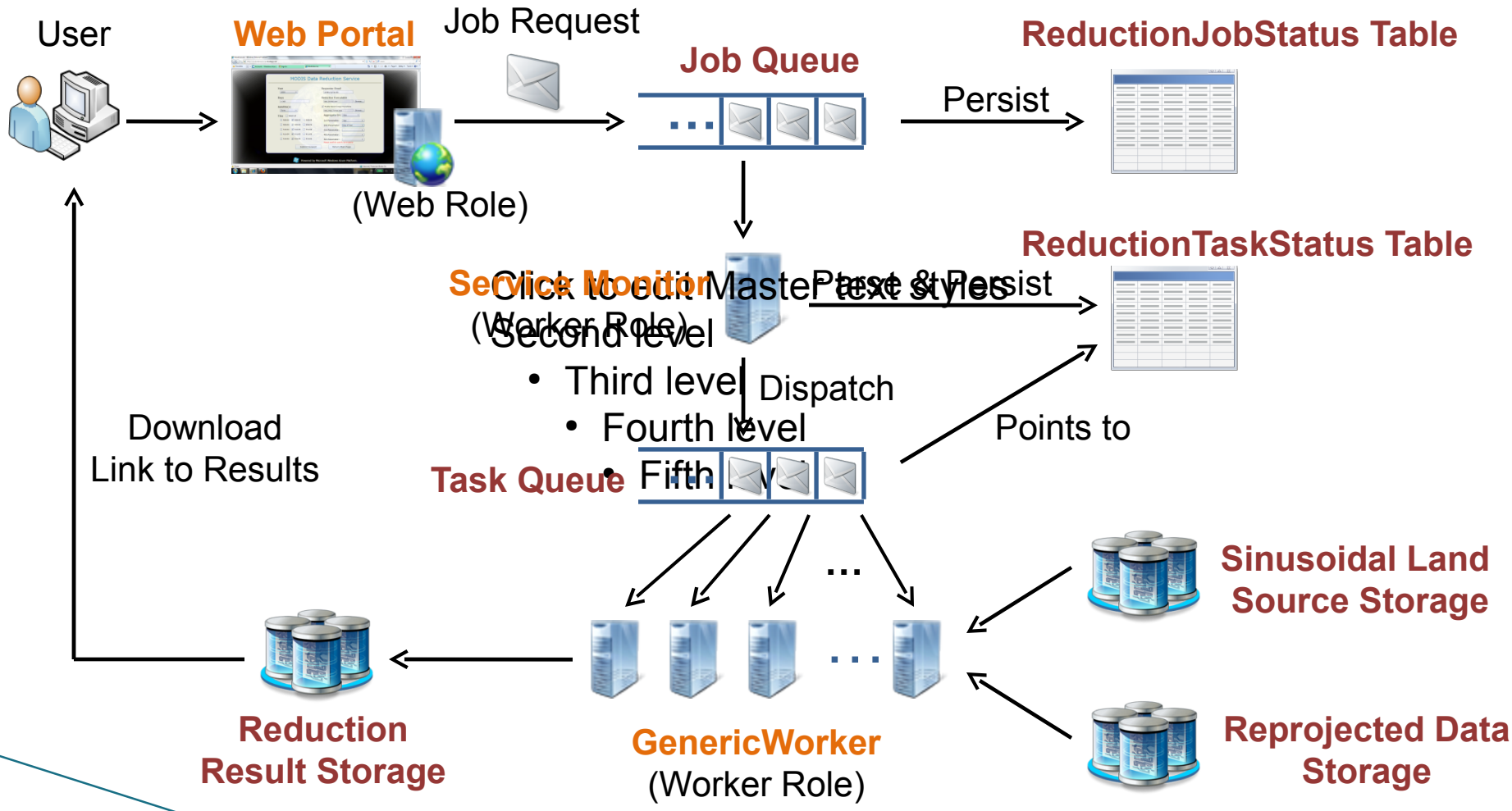
7. A single download link to the results is sent back to the scientist

Data Processing Service Architecture

AzureMODIS Data Service Demo

http://modisazure.cloudapp.net/

Behind the scene…



Data Caching

Blob storage level

Each data file (blob) has a global unique identifier

(Pre-)download and cache all source files in blob storage

(Pre-)compute reprojection results for reuse across computations

Local machine level

Each small size instance has ~250GB local storage

Cache large size data files for reuse

Cost-related Trade offs

Data re-generation cost VS. Blob storage cost

For our case, data re-computation is too expensive

Reduction Service

Scientists upload their analysis binary tools upon request for the reduction service

Benefits

Scientists can easily debug and refine scientific models in their code

Separate system code debugging from science code debugging

A 2nd reduction stage to support more comprehensive computation flows

Outline

Project Background

AzureMODIS Framework Overview

Dynamic Scalability & Fault Tolerance

Evaluation

Conclusions & Future Work

Dynamic Scalability

Use the *Azure Management API* to dynamically scale up/down instances according to work loads

Dynamic instance shutdown could be a problem

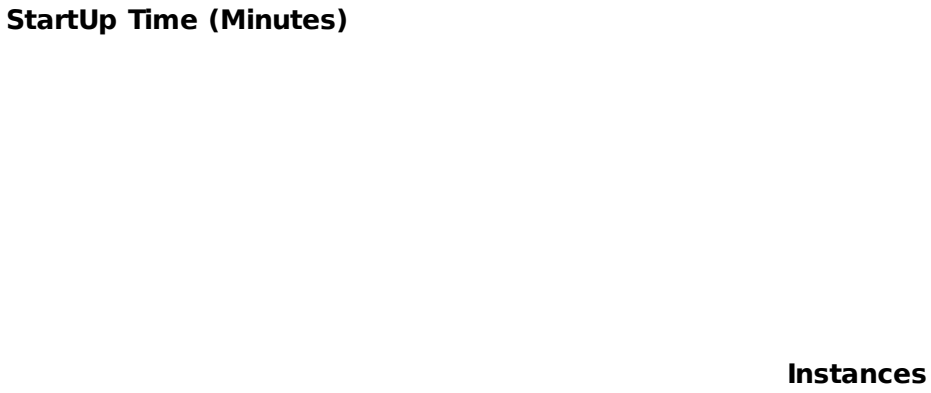
- Azure decides which instance to shutdown
- Instances may be shutdown during task execution

Currently, computing instance usage are charged by hours

- Use CPU hours wisely when applying dynamic scaling strategies

Performance of dynamic instance scaling

Instance Start Up Time (*Test Date: March 31, 2010*)



In contrast, the shutdown time for the instances is small (usually within 3 minutes)

Fault Tolerance

Tasks can fail for many reasons

Broken or missing source data files — Unrecoverable

Reduction tool may crash due to code bug — Unrecoverable

Failures caused by system instability — Recoverable

Customized task retry policies

Task with timeout failures will be resent to the task queue

Task with exceptions caught will be immediately resent

Task canceled after 2 retries (Totally 3 executions)

Why not just use queue message visibility settings for failure recovery?

Service Monitoring & Diagnosing (Demo)

http://modisazure.cloudapp.net/

Outline

Project Background

AzureMODIS Framework Overview

Dynamic Scalability & Fault Tolerance

Evaluation

Conclusions & Future Work

Overall Performance & Scalability

Table 2. Capacity of desktop machine and a single Azure instance

	Desktop	Azure Instance
Capacity	CPU: Intel Core2Duo E6850 @ 3.0GHZ Memory: 4GB Hard Disk: 1TB SATA Network: 1Gbps Ethernet OS: Windows 7 (32-bit)	CPU: 1.6GHZ X64 equivalent processor Memory: 2GB Local Storage: 250GB Network: 100Mbps OS: Windows 2008 Server x64 (64-bit)

Table 3. Processing time for 1500 reprojection tasks (Unit: hours)

	MOD04_L2	MOD06_L2	MYD11_L2.005
150 instances	0.30	0.85	0.44
100 instances	0.40	1.20	0.61
50 instances	0.76	2.25	1.12
Desktop	16.29	72.62	33.45

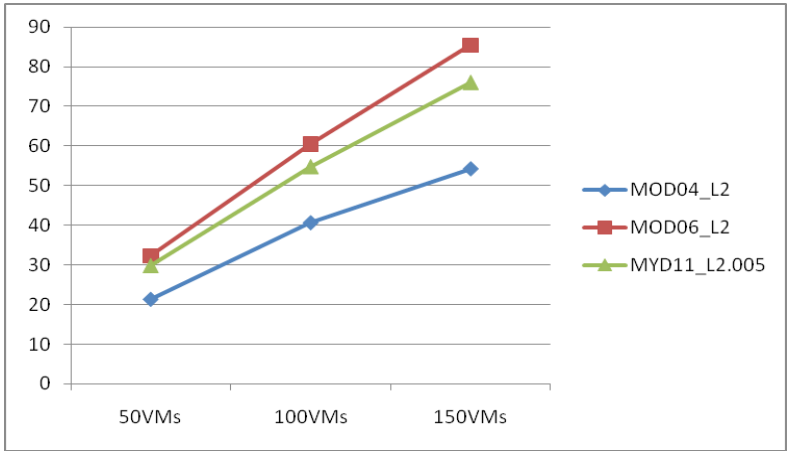
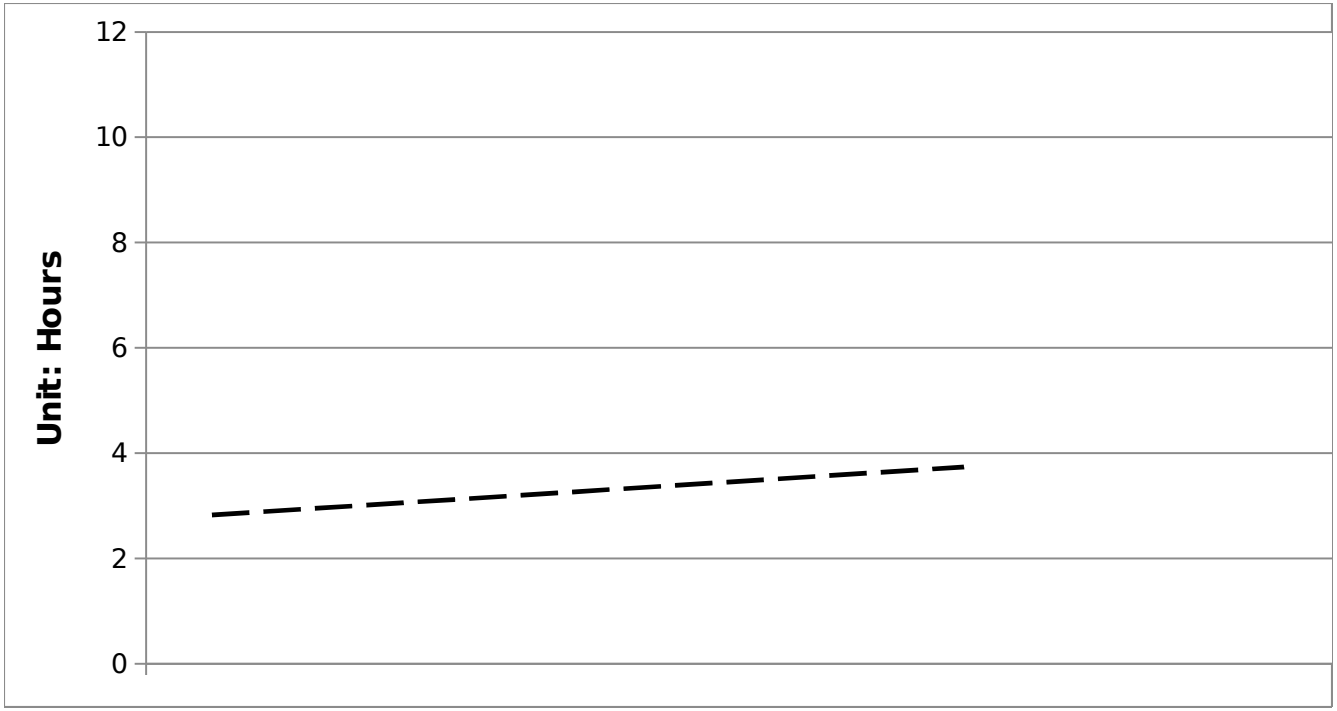


Fig. 1 Performance speedups over a single desktop

Storage Service Scalability



Accumulated time for data transfer from/to Azure blob storage increases as #VM increases

Outline

Project Background

AzureMODIS Framework Overview

Dynamic Scalability & Fault Tolerance

Evaluation

Conclusions & Future Work

Conclusions

Cloud computing provides new capabilities and opportunities for data-intensive eScience research

Dynamic scalability is powerful, but instance start up overhead is not trivial

Built-in fault tolerance & diagnostic features are important in the face of common failures in large-scale cloud applications and systems

Future Work

Scale up computations from US continent to the global scale

Develop and evaluate a generic dynamic scaling mechanism with AzureMODIS

Evaluate the similarities/differences between our framework and other generic parallel computing frameworks such as MapReduce

Thank you!

&

Questions?