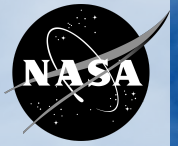
# How to Cloud for Earth Scientists: An Introduction

Chris Lynnes, NASA EOSDIS* System Architect

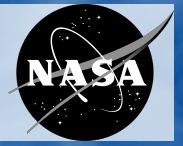
*Earth Observing System Data and Information System

# Outline



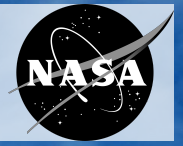

- Cloud Basics
- What good is cloud computing to an Earth Scientist?
- What's the catch?
- Getting Started...

# What Cloud Computing Is




- “Someone else’s computer”...
- ...but also someone else’s problem
- Rent instead of own, like:
  - A box truck (Bigger than your Sport Utility Vehicle)
  - A seat on an airplane (Faster than your sports car)
- Computing a la carte
- Service-based computing

# What Cloud Computing Isn't




1. It's not the solution for everything
2. It's not the solution for everyone

# So Why Should We Care?

More and bigger data are coming!

200+ PB to EOSDIS

# Cloud Fundamentals - Elasticity

- Elastic = scaling up, down or sideways instantly
  - Compute:
    - more or less
    - optimized for compute, memory, or input/output
  - Storage:
    - more or less
    - faster or slower
- Elastic = pay for only what you use
  - (Remember to turn off when not using!)

# “Undifferentiated Heavy Lifting”

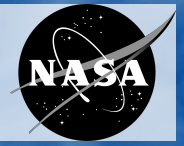
| *Stuff for which you need remote sensing expertise* |
| --- |
| Radiative transfer modeling |
| Atmospheric correction |
| Geophysical parameter retrievals |

# “Undifferentiated Heavy Lifting”

| *Stuff for which you need Earth science expertise* | *Stuff for which you **DON’T** need Earth science expertise* |
|---|---|
| Radiative transfer modeling | Finding available floor space for computers |
| Atmospheric correction | Installing and patching operating systems |
| Geophysical parameter retrievals | Calculating power and cooling requirements |

# What Good Is Cloud Computing to an Earth Scientist???

# Go Faster




- Commercial cloud CPUs are faster than ours...
- ...*And* you can use as many as you want
- Uses
  - Near-real-time processing
  - Massive reprocessing
  - Compute-intensive analysis
  - Deep learning

# Pop Quiz!

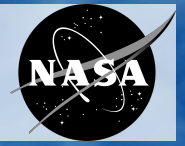
If a compute-optimized CPU with 16 cores costs 80 ¢ / hr...

And you need 1000 CPU-hours to compute your calculation…

Which of these is cheaper?

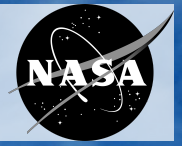
1. 1 CPU running for 1000 hours
2. 1000 CPUs running for 1 hour

# Answer:




1. 1 CPU * 1000 Hrs * 0.8 = $800
2. 1000 CPU * 1 Hr * 0.8 = $800

# Go Bigger



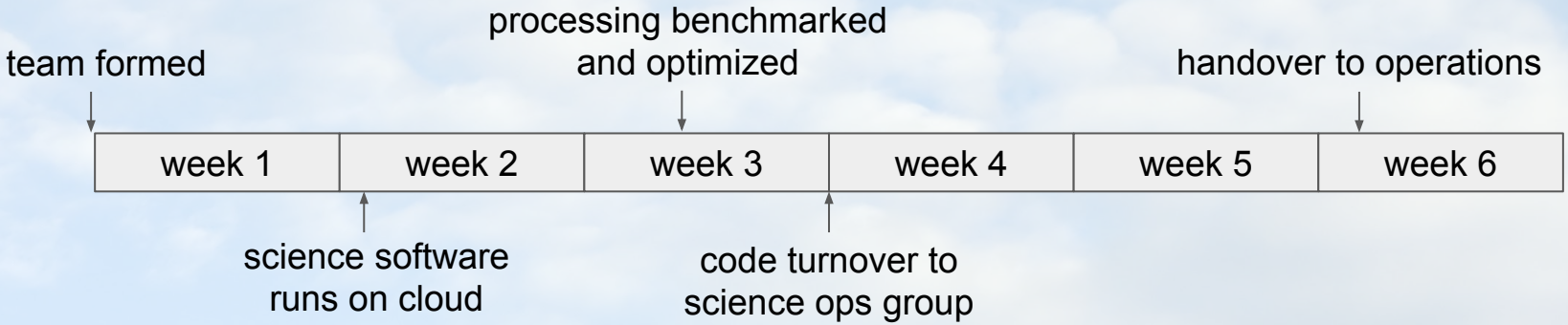

- Many levels of storage
  - Fast but expensive:  $0.30 / GB-month
  - Slow but dirt-cheap:  $0.025 ¢ / GB-month
- You can have as much as you want
- Uses
  - Short-term storage of large interim results
  - Long-term storage of data that you *might* need some day

# Go Cheaper

- Pay only for what you use
  - CPU
  - Storage
- Uses
  - Short bursts of lots of processing
  - Lots of storage needed for a short time

# The OCO-2* Reprocessing Story

- Motivation for reprocessing OCO-2 in the cloud
  - Conflicts with supercomputer down time schedule
  - Increase in computing needs

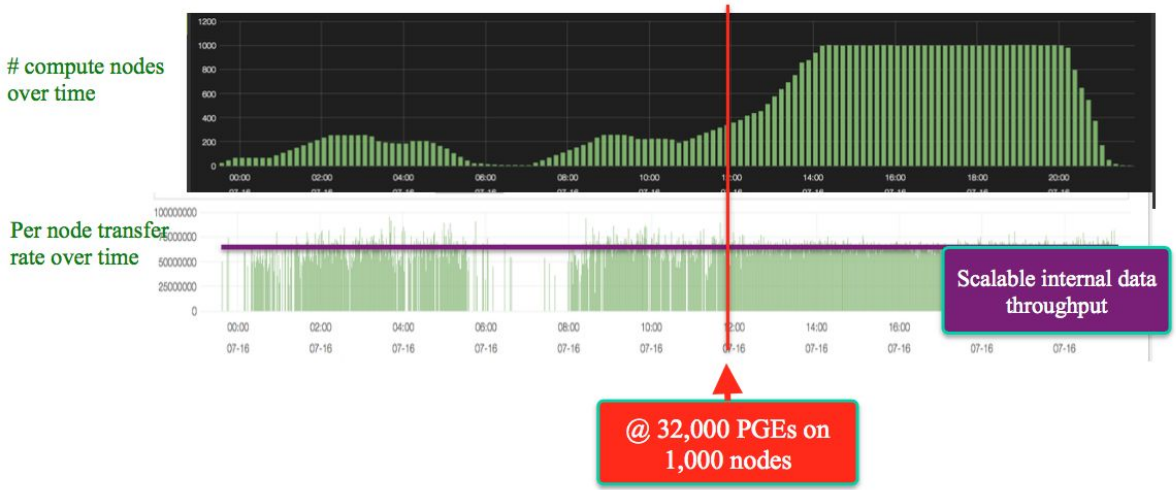



*OCO-2: Orbiting Carbon Observatory #2

# The OCO-2 Reprocessing Story

## Auto-Scaling Science Data System

- The size of the science data system compute nodes can automatically grow/shrink based on processing demand




*Auto-scaling tests to 3000 compute workers*

***96,000 x l2_fp simultaneous processors***

Courtesy Hook Hua, NASA/Jet Propulsion Laboratory

# What's the Catch?

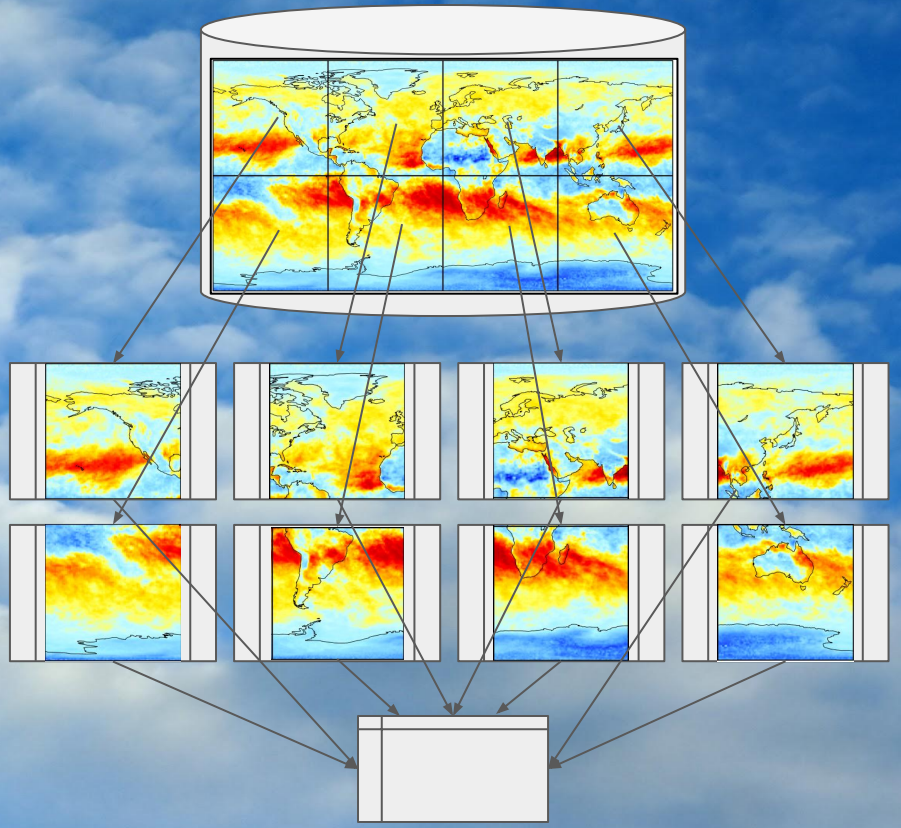
1. New processing paradigm
2. Failures
3. Egress charges

# Catch #1: New Processing Paradigm

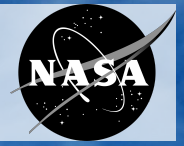
Bad News:

To get the speedup, you must:

1. Spread input data around
2. Go analyze the pieces
3. Reassemble final result



# Catch #1: New Processing Paradigm




**Good News:**

LOTS of packages and frameworks to help with this

1. Distributed Data Stores
   a. Databases (Cassandra, Athena...)
   b. Filesystems (HDFS...)
2. Processing frameworks (MapReduce, Spark)

*Pssst….think seriously about learning Python (just sayin')*

# Catch #2: Failures

**Bad News**:

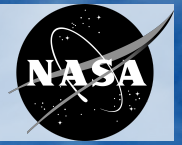
thousands of computers

\+ thousands of “disks”

---

bad stuff happens

# Catch #2:  Failures




**Good News**:

- Many cloud technologies exist to provide resiliency to hardware failures
- BUT our programs need to be able to pick themselves up and/or restart somewhere else
  - Don’t rely on local temporary files for checkpoints

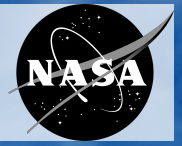
# Catch #3: “Egress” charges

Moving results from cloud to your machine costs money:

| | |
|---|---|
| 1 GB | Free! |
| 10 TB | $900 |
| 50 TB | $4,700 |
| 150 TB | $12,000 |

Analyze as much as you can in the cloud to reduce output size

# Getting Started with Cloud...




- Many vendors offer free tiers for learning
- There is a *lot* of online training
- More “How to Cloud” seminars to come?
  - Short-learning-ramp ways to use cloud?
  - Example science uses?
  - What would YOU like to see?

*Pssst….don’t forget about the Python thing.*