Architecting for the Cloud: Best Practices and Cost Analysis

Science Data Systems

Wednesday, January 11, 2017
ESIP Federation 2017 Winter Meeting
Bethesda, MD

HySDS Team: Hook Hua, Gerald Manipon, Michael Starch, Lan Dang, Justin Linick
Phillip Southam, Michael Cayanan, Namrata Malarout
NASA Jet Propulsion Laboratory / California Institute of Technology

Copyright 2017, by the California Institute of Technology. ALL RIGHTS RESERVED. United States Government Sponsorship acknowledged. Any commercial use must be negotiated with the Office of Technology Transfer at the California Institute of Technology
Next Generation Missions

Estimated Daily Volume (TB)

The volume of data produced is larger than previous missions.

Data storage, processing, movement, and costs are the biggest challenges.
Total Cost of Ownership (TCO) implications of Earth Science Data Systems in the Cloud:

- Compute
- Storage
- Network
- Deployment Topology

- Discussion in the context of AWS.
- This is not an endorsement of any one specific cloud vendor.
NISAR Science Data System (SDS)

- Average Input Data Volume to SDS: 26 Tbits/day (3.25 TB per day)
- Average Daily Production Volume: 86TB per day (“keep up”)
- Total Mission Data Volume: 100+ PBs

SDS challenges include both internal and external data production rates

JPL GDS

Raw: 3.25 TB/day

JPL SDS

Forward Processing

Raw: 13 TB/day

Bulk Processing

During Reprocessing: L0B: 3.25 TB/day
Products: Nominal = 86TB/day
During Reprocessing 430TB/day (peak)

ASF

ISRO SDS

Raw: 34 TB/day

Network

Project teams for diagnostics

- ADT / Scientists for research, analysis, & Aux data improvements
- Scientists & public: products access

2017-01-11

Architecting for the Cloud: Best Practices and Cost Analysis
A CLOUD-BASED APPROACH
Basic Premise of Cloud-based Science Data Processing

- Science data product into AWS S3 **object storage**
- **Scale up** compute nodes to run in AWS EC2
- **Internal** SDS data throughput needs are scalable via **cloud architecture**
  - Object storage can scale up **data volume** and **aggregate data throughput** by compute instances
- Architectural components can be **collocated**
Cloud Economics

**COMPUTE**
### AWS EC2 Instance Variations

#### Table: EC2 Instance Types

<table>
<thead>
<tr>
<th>Name</th>
<th>API Name</th>
<th>Memory</th>
<th>Compute Units (ECU)</th>
<th>vCPUs</th>
<th>Storage</th>
<th>Arch</th>
<th>Network Performance</th>
<th>EBS Optimized: Max Bandwidth</th>
<th>VPC Only</th>
<th>Linux On Demand cost</th>
<th>Linux Rese</th>
</tr>
</thead>
<tbody>
<tr>
<td>C3 High-CPU Eight Extra Large</td>
<td>c3.8xlarge</td>
<td>60.0 GB</td>
<td>108 units</td>
<td>32 vCPUs</td>
<td>640.0 GB (2 * 320.0 GB SSD)</td>
<td>64-bit</td>
<td>10 Gigabit</td>
<td>N/A</td>
<td>N/A</td>
<td>$1.680 hourly</td>
<td>$1.168 hour</td>
</tr>
<tr>
<td>C3 High-CPU Quadruple Extra Large</td>
<td>c3.4xlarge</td>
<td>30.0 GB</td>
<td>55 units</td>
<td>16 vCPUs</td>
<td>320.0 GB (2 * 160.0 GB SSD)</td>
<td>64-bit</td>
<td>10 Gigabit</td>
<td>High</td>
<td>Yes</td>
<td>$0.840 hourly</td>
<td>$0.584 hour</td>
</tr>
<tr>
<td>C4 High-CPU Eight Extra Large</td>
<td>c4.8xlarge</td>
<td>60.0 GB</td>
<td>132 units</td>
<td>36 vCPUs</td>
<td>0 GB (EBS only)</td>
<td>64-bit</td>
<td>10 Gigabit</td>
<td>10 Gigabit</td>
<td>Yes</td>
<td>$1.591 hourly</td>
<td>$1.130 hour</td>
</tr>
<tr>
<td>C4 High-CPU Quadruple Extra Large</td>
<td>c4.4xlarge</td>
<td>30.0 GB</td>
<td>62 units</td>
<td>16 vCPUs</td>
<td>0 GB (EBS only)</td>
<td>64-bit</td>
<td>10 Gigabit</td>
<td>2000.0 Mbps</td>
<td>Yes</td>
<td>$0.796 hourly</td>
<td>$0.565 hour</td>
</tr>
<tr>
<td>Cluster Compute Eight Extra Large</td>
<td>cc2.8xlarge</td>
<td>60.5 GB</td>
<td>88 units</td>
<td>32 vCPUs</td>
<td>3360.0 GB (4 * 840.0 GB SSD)</td>
<td>64-bit</td>
<td>10 Gigabit</td>
<td>N/A</td>
<td>N/A</td>
<td>$2.000 hourly</td>
<td>$1.090 hour</td>
</tr>
<tr>
<td>Cluster GPU Quadruple Extra Large</td>
<td>cg1.4xlarge</td>
<td>22.5 GB</td>
<td>33.5 units</td>
<td>16 vCPUs</td>
<td>1680.0 GB (2 * 840.0 GB SSD)</td>
<td>64-bit</td>
<td>10 Gigabit</td>
<td>N/A</td>
<td>N/A</td>
<td>$2.100 hourly</td>
<td>unavailable</td>
</tr>
<tr>
<td>32 Double Extra Large</td>
<td>d2.xlarge</td>
<td>61.0 GB</td>
<td>28 units</td>
<td>8 vCPUs</td>
<td>12000.0 GB (8 * 2000.0 GB SSD)</td>
<td>64-bit</td>
<td>10 Gigabit</td>
<td>High</td>
<td>Yes</td>
<td>$1.300 hourly</td>
<td>$0.804 hour</td>
</tr>
<tr>
<td>32 Eight Extra Large</td>
<td>d2.8xlarge</td>
<td>244.0 GB</td>
<td>116 units</td>
<td>36 vCPUs</td>
<td>48000.0 GB (24 * 2000.0 GB SSD)</td>
<td>64-bit</td>
<td>10 Gigabit</td>
<td>High</td>
<td>Yes</td>
<td>$5.520 hourly</td>
<td>$3.216 hour</td>
</tr>
<tr>
<td>32 Quadruple Extra Large</td>
<td>d2.4xlarge</td>
<td>122.0 GB</td>
<td>56 units</td>
<td>16 vCPUs</td>
<td>24000.0 GB (12 * 2000.0 GB SSD)</td>
<td>64-bit</td>
<td>10 Gigabit</td>
<td>High</td>
<td>Yes</td>
<td>$2.760 hourly</td>
<td>$1.608 hour</td>
</tr>
<tr>
<td>32 Eight Extra Large</td>
<td>d2.xlarge</td>
<td>60.0 GB</td>
<td>104 units</td>
<td>32 vCPUs</td>
<td>240.0 GB (2 * 120.0 GB SSD)</td>
<td>64-bit</td>
<td>10 Gigabit</td>
<td>N/A</td>
<td>N/A</td>
<td>$2.600 hourly</td>
<td>$1.896 hour</td>
</tr>
<tr>
<td>General Purpose GPU 16xlarge</td>
<td>p2.16xlarge</td>
<td>732.0 GB</td>
<td>188 units</td>
<td>64 vCPUs</td>
<td>0 GB (EBS only)</td>
<td>64-bit</td>
<td>20 Gigabit</td>
<td>20 Gigabit</td>
<td>Yes</td>
<td>$14.400 hourly</td>
<td>$10.951 hour</td>
</tr>
<tr>
<td>General Purpose GPU Eight Extra Large</td>
<td>p2.8xlarge</td>
<td>488.0 GB</td>
<td>94 units</td>
<td>32 vCPUs</td>
<td>0 GB (EBS only)</td>
<td>64-bit</td>
<td>20 Gigabit</td>
<td>N/A</td>
<td>N/A</td>
<td>$7.200 hourly</td>
<td>$5.476 hour</td>
</tr>
<tr>
<td>H1.1 High I/O Quadruple Extra Large</td>
<td>hi1.4xlarge</td>
<td>60.5 GB</td>
<td>35 units</td>
<td>16 vCPUs</td>
<td>2048.0 GB (2 * 1024.0 GB SSD)</td>
<td>64-bit</td>
<td>10 Gigabit</td>
<td>High</td>
<td>Yes</td>
<td>$3.100 hourly</td>
<td>$1.998 hour</td>
</tr>
<tr>
<td>High Memory Cluster Eight Extra Large</td>
<td>r1.8xlarge</td>
<td>244.0 GB</td>
<td>88 units</td>
<td>32 vCPUs</td>
<td>240.0 GB (2 * 120.0 GB SSD)</td>
<td>64-bit</td>
<td>10 Gigabit</td>
<td>High</td>
<td>Yes</td>
<td>$3.500 hourly</td>
<td>$2.989 hour</td>
</tr>
<tr>
<td>High Storage Eight Extra Large</td>
<td>hs1.8xlarge</td>
<td>117.0 GB</td>
<td>35 units</td>
<td>16 vCPUs</td>
<td>48000.0 GB (24 * 2000.0 GB SSD)</td>
<td>64-bit</td>
<td>10 Gigabit</td>
<td>N/A</td>
<td>N/A</td>
<td>$4.600 hourly</td>
<td>$2.574 hour</td>
</tr>
<tr>
<td>2 Double Extra Large</td>
<td>t2.2xlarge</td>
<td>61.0 GB</td>
<td>27 units</td>
<td>8 vCPUs</td>
<td>1600.0 GB (2 * 800.0 GB SSD)</td>
<td>64-bit</td>
<td>10 Gigabit</td>
<td>High</td>
<td>Yes</td>
<td>$1.705 hourly</td>
<td>$0.848 hour</td>
</tr>
<tr>
<td>2 Eight Extra Large</td>
<td>t2.8xlarge</td>
<td>244.0 GB</td>
<td>104 units</td>
<td>16 vCPUs</td>
<td>6400.0 GB (2 * 3200.0 GB SSD)</td>
<td>64-bit</td>
<td>10 Gigabit</td>
<td>High</td>
<td>Yes</td>
<td>$8.620 hourly</td>
<td>$3.392 hour</td>
</tr>
<tr>
<td>2 Quadruple Extra Large</td>
<td>t2.4xlarge</td>
<td>122.0 GB</td>
<td>53 units</td>
<td>32 vCPUs</td>
<td>3200.0 GB (4 * 800.0 GB SSD)</td>
<td>64-bit</td>
<td>10 Gigabit</td>
<td>High</td>
<td>Yes</td>
<td>$3.410 hourly</td>
<td>$1.966 hour</td>
</tr>
<tr>
<td>M2 High Memory Quadriple Extra Large</td>
<td>m2.4xlarge</td>
<td>68.0 GB</td>
<td>26 units</td>
<td>16 vCPUs</td>
<td>1680.0 GB (2 * 840.0 GB SSD)</td>
<td>64-bit</td>
<td>10 Gigabit</td>
<td>High</td>
<td>Yes</td>
<td>$0.980 hourly</td>
<td>$0.444 hour</td>
</tr>
<tr>
<td>M3 General Purpose Double Extra Large</td>
<td>m3.2xlarge</td>
<td>30.0 GB</td>
<td>26 units</td>
<td>32 vCPUs</td>
<td>160.0 GB (2 * 80.0 GB SSD)</td>
<td>64-bit</td>
<td>10 Gigabit</td>
<td>High</td>
<td>Yes</td>
<td>$0.532 hourly</td>
<td>$0.380 hour</td>
</tr>
<tr>
<td>M4 16xlarge</td>
<td>m4.16xlarge</td>
<td>256.0 GB</td>
<td>188 units</td>
<td>128 vCPUs</td>
<td>0 GB (EBS only)</td>
<td>64-bit</td>
<td>20 Gigabit</td>
<td>20 Gigabit</td>
<td>Yes</td>
<td>$3.447 hourly</td>
<td>$2.699 hour</td>
</tr>
</tbody>
</table>

- Hyperthreading and vCPUs
  - m2.4xlarge instances previously had hyperthreading disabled
- **Multi-tenancy and contention**
  - m2.4xlarge instance previously occupied the entire physical node, which had no multi-tenancy contention.
- Memory to CPU ratio
- **AWS trend to move away from ephemeral(instance) storage to SSD to EBS**
- Network bandwidth
- EBS bandwidth
### Benchmarks for OCO-2 Level 2 Full Physics processors

<table>
<thead>
<tr>
<th>instance</th>
<th>vCPU</th>
<th>physical hardware</th>
<th>cores</th>
<th>threads</th>
<th>memory</th>
<th>memory-cpu ratio</th>
<th>disks</th>
<th>soundings per job</th>
<th>concurrent processors</th>
<th># test runs</th>
<th>mean runtime (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>m2.4xlarge</td>
<td>8</td>
<td>Intel Xeon E5-2665 @ 2.40 GHz 1 processor, 8 threads</td>
<td>8</td>
<td>8</td>
<td>68.4</td>
<td>8.55</td>
<td>2 x 840</td>
<td>32</td>
<td>42</td>
<td>13.69</td>
<td></td>
</tr>
<tr>
<td>cc2.8xlarge</td>
<td>32</td>
<td>Intel Xeon E5-2670 @ 2.30 GHz 2 processors, 16 cores, 32 threads</td>
<td>16</td>
<td>32</td>
<td>60.5</td>
<td>1.89</td>
<td>4 x 840</td>
<td>32</td>
<td>32</td>
<td>143.65</td>
<td></td>
</tr>
<tr>
<td>m3.2xlarge</td>
<td>8</td>
<td>Intel Xeon E5-2670 @ 2.60 GHz 1 processor, 8 threads</td>
<td>4</td>
<td>8</td>
<td>30</td>
<td>3.75</td>
<td>SSD 2 x 80</td>
<td>32</td>
<td>8</td>
<td>131</td>
<td></td>
</tr>
<tr>
<td>c3.8xlarge</td>
<td>32</td>
<td>Intel Xeon E5-2680 v2 @ 2.80 GHz 2 processors, 16 cores, 32 threads</td>
<td>16</td>
<td>32</td>
<td>60</td>
<td>1.88</td>
<td>SSD 2 x 320</td>
<td>32</td>
<td>32</td>
<td>461</td>
<td></td>
</tr>
<tr>
<td>r3.8xlarge</td>
<td>32</td>
<td>Intel Xeon E5-2670 v2 @ 2.49 GHz 2 processors, 16 cores, 32 threads</td>
<td>16</td>
<td>32</td>
<td>244</td>
<td>7.63</td>
<td>SSD 2 x 320</td>
<td>32</td>
<td>32</td>
<td>489</td>
<td></td>
</tr>
</tbody>
</table>

- **m2.4xlarge** has best performance for OCO-2 L2_fp processing
  - m2.4xlarge instance *may* occupy the entire physical node. No multi-tenancy contention
  - 32 concurrent processors on c3 and r3 instances may have more memory and disk contention
  - More memory on r3.8xlarge not as large affect on performance
  - m2.4xlarge does not have SSD disks. L2_fp may not be as heavy on disk I/O bound.

- **Optimize job sizes per instance type**
  - m2.4xlarge: use 80 soundings per 1-hour job
  - c3.8xlarge: use 256 soundings per 1-hour job
**“Cost Per Unit” Estimates**

<table>
<thead>
<tr>
<th>instance</th>
<th>soundings per job</th>
<th>concurrent processors</th>
<th># test runs</th>
<th>mean runtime (s)</th>
<th>mean runtime per sounding per processor (m)</th>
<th># soundings (evenly distributed processors)</th>
<th>mean costs per job</th>
<th>$ per sounding</th>
<th>mean costs per job</th>
<th>$ per sounding</th>
<th>mean costs per job</th>
<th>$ per sounding</th>
<th>mean costs per job</th>
<th>$ per sounding</th>
</tr>
</thead>
<tbody>
<tr>
<td>m2.4xlarge</td>
<td>32</td>
<td>8</td>
<td>42</td>
<td>1384</td>
<td>5.3</td>
<td>69</td>
<td>$0.4223</td>
<td>$0.01320</td>
<td>$0.0250</td>
<td>$0.00078</td>
<td>$0.4138</td>
<td>$0.01293</td>
<td>$0.0384</td>
<td>$0.00120</td>
</tr>
<tr>
<td>cc2.8xlarge</td>
<td>32</td>
<td>32</td>
<td>14365</td>
<td>613</td>
<td>10.3</td>
<td>156</td>
<td>$0.3417</td>
<td>$0.01068</td>
<td>$0.0462</td>
<td>$0.00144</td>
<td>$0.3417</td>
<td>$0.01068</td>
<td>$0.0462</td>
<td>$0.00144</td>
</tr>
<tr>
<td>m3.2xlarge</td>
<td>32</td>
<td>32</td>
<td>126</td>
<td>461</td>
<td>7.3</td>
<td>208</td>
<td>$0.2448</td>
<td>$0.00765</td>
<td>$0.2151</td>
<td>$0.00672</td>
<td>$0.2151</td>
<td>$0.00672</td>
<td>$0.3073</td>
<td>$0.00960</td>
</tr>
<tr>
<td>r3.8xlarge</td>
<td>32</td>
<td>32</td>
<td>131</td>
<td>489</td>
<td>8.2</td>
<td>196</td>
<td>$0.4238</td>
<td>$0.01324</td>
<td>$0.0543</td>
<td>$0.00170</td>
<td>$0.3803</td>
<td>$0.01189</td>
<td>$0.3803</td>
<td>$0.01189</td>
</tr>
</tbody>
</table>

- **OCO-2’s “$ per sounding” yielded non-obvious results**
  - *More* expensive compute instance was *cheaper* per science unit output
## Spot Market

<table>
<thead>
<tr>
<th>Instance</th>
<th>vCPU</th>
<th>Memory</th>
<th>Memory-CPU Ratio</th>
<th>Disks</th>
<th>On-Demand ($/hr)</th>
<th>Reserved 1-yr Upfront ($/hr)</th>
<th>Reserved 3-yr Upfront ($/hr)</th>
<th>Spot Linux ($/hr)</th>
<th>On-Demand ($/cpu/hr)</th>
<th>Reserved 1-yr Upfront ($/cpu/hr)</th>
<th>Reserved 3-yr Upfront ($/cpu/hr)</th>
<th>Spot Linux ($/cpu/hr)</th>
</tr>
</thead>
<tbody>
<tr>
<td>m2.4xlarge</td>
<td>8</td>
<td>68.4</td>
<td>8.55</td>
<td>2 x 840</td>
<td>$1.0780</td>
<td>$0.4087</td>
<td>$0.2440</td>
<td>$0.1000</td>
<td>$0.1348</td>
<td>$0.0511</td>
<td>$0.0300</td>
<td>$0.0125</td>
</tr>
<tr>
<td>cc2.8xlarge</td>
<td>32</td>
<td>60.9</td>
<td>1.89</td>
<td>4 x 840</td>
<td>$2.0000</td>
<td>$0.9131</td>
<td>$0.613</td>
<td>$0.2705</td>
<td>$0.0625</td>
<td>$0.0285</td>
<td>$0.0192</td>
<td>$0.0085</td>
</tr>
<tr>
<td>m3.2xlarge</td>
<td>8</td>
<td>30.0</td>
<td>3.75</td>
<td>SSD 2 x 80</td>
<td>$0.6160</td>
<td>$0.3750</td>
<td>$0.2300</td>
<td>$0.0700</td>
<td>$0.0770</td>
<td>$0.0468</td>
<td>$0.0280</td>
<td>$0.0088</td>
</tr>
<tr>
<td>c3.8xlarge</td>
<td>32</td>
<td>60.0</td>
<td>1.88</td>
<td>SSD 2 x 320</td>
<td>$1.6800</td>
<td>$0.9920</td>
<td>$0.6280</td>
<td>$2.4001</td>
<td>$0.0525</td>
<td>$0.0310</td>
<td>$0.0190</td>
<td>$0.0075</td>
</tr>
<tr>
<td>r3.8xlarge</td>
<td>32</td>
<td>244.0</td>
<td>7.63</td>
<td>SSD 2 x 320</td>
<td>$2.8000</td>
<td>$1.4860</td>
<td>$0.9820</td>
<td>$2.8000</td>
<td>$0.0875</td>
<td>$0.0464</td>
<td>$0.0300</td>
<td>$0.00875</td>
</tr>
<tr>
<td>c3.xlarge</td>
<td>4</td>
<td>7.5</td>
<td>1.88</td>
<td>SSD 2 x 40</td>
<td>$0.2310</td>
<td>$0.1370</td>
<td>$0.0870</td>
<td>$0.0353</td>
<td>$0.0578</td>
<td>$0.0343</td>
<td>$0.0210</td>
<td>$0.0088</td>
</tr>
</tbody>
</table>

- **Major cost savings (75%-90% savings over on-demand)...if can use spot instances**
- **On spot market, AWS will terminate compute instances if market prices exceed bid threshold**
- **Need to self-recover from spot instance terminations.**
“Market Maker”

This OCO-2 production run of 1000 x 32vCPUs affected the market prices

Strategy:
- Mitigate impact on spot market
- Diversification of resources
- “Spot fleet”
This OCO-2 production run affected the market prices
Cloud Economics

STORAGE
Forward and Bulk Reprocessing

Example Bulk Reprocessing Campaigns

2017-01-11

Architecting for the Cloud: Best Practices and Cost Analysis
AWS Storage: Monthly Tiered Example

• Storage types
  – EBS
  – S3
  – S3 Infrequent Access
  – Glacier

• Storage cost tiers

**Monthly S3 Storage Costs (rack rate)**

<table>
<thead>
<tr>
<th>Monthly cost</th>
<th>Price/GB</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Tier 1: First 1 TB</td>
<td>1.00 TB</td>
<td>0.0300 =</td>
</tr>
<tr>
<td>Tier 2: Next 49 TB</td>
<td>49.00 TB</td>
<td>0.0295 =</td>
</tr>
<tr>
<td>Tier 3: Next 450 TB</td>
<td>36.00 TB</td>
<td>0.0290 =</td>
</tr>
<tr>
<td>Tier 4: Next 500 TB</td>
<td>0.00 TB</td>
<td>0.0285 =</td>
</tr>
<tr>
<td>Tier 5: Next 4000 TB</td>
<td>0.00 TB</td>
<td>0.0280 =</td>
</tr>
<tr>
<td>Tier 6: Above 5000 TB</td>
<td>0.00 TB</td>
<td>0.0275 =</td>
</tr>
<tr>
<td><strong>Total Monthly</strong></td>
<td>86.00 TB</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2,519.50 $</td>
</tr>
</tbody>
</table>

**Monthly cost**

<table>
<thead>
<tr>
<th>Monthly cost</th>
<th>Price/GB</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Tier 1: First 1 TB</td>
<td>1.00 TB</td>
<td>0.0300 =</td>
</tr>
<tr>
<td>Tier 2: Next 49 TB</td>
<td>49.00 TB</td>
<td>0.0295 =</td>
</tr>
<tr>
<td>Tier 3: Next 450 TB</td>
<td>450.00 TB</td>
<td>0.0290 =</td>
</tr>
<tr>
<td>Tier 4: Next 500 TB</td>
<td>500.00 TB</td>
<td>0.0285 =</td>
</tr>
<tr>
<td>Tier 5: Next 4000 TB</td>
<td>1580.00 TB</td>
<td>0.0280 =</td>
</tr>
<tr>
<td>Tier 6: Above 5000 TB</td>
<td>0.00 TB</td>
<td>0.0275 =</td>
</tr>
<tr>
<td><strong>Total Monthly</strong></td>
<td>2580.00 TB</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>73,015.50 $</td>
</tr>
</tbody>
</table>

**Monthly cost**

<table>
<thead>
<tr>
<th>Monthly cost</th>
<th>Price/GB</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Tier 1: First 1 TB</td>
<td>1.00 TB</td>
<td>0.0300 =</td>
</tr>
<tr>
<td>Tier 2: Next 49 TB</td>
<td>49.00 TB</td>
<td>0.0295 =</td>
</tr>
<tr>
<td>Tier 3: Next 450 TB</td>
<td>450.00 TB</td>
<td>0.0290 =</td>
</tr>
<tr>
<td>Tier 4: Next 500 TB</td>
<td>500.00 TB</td>
<td>0.0285 =</td>
</tr>
<tr>
<td>Tier 5: Next 4000 TB</td>
<td>4000.00 TB</td>
<td>0.0280 =</td>
</tr>
<tr>
<td>Tier 6: Above 5000 TB</td>
<td>26390.00 TB</td>
<td>0.0275 =</td>
</tr>
<tr>
<td><strong>Total Monthly</strong></td>
<td>31390.00 TB</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>866,500.50 $</td>
</tr>
</tbody>
</table>

**..Not cost-effective to store everything in S3**
Data Tiering

• Optimizing cost to storage use
Cloud Economics

NETWORK / DATA MOVEMENT
Understanding Egress and Costs

Storage Pricing (varies by region)

<table>
<thead>
<tr>
<th>Region: US West (Oregon)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Standard Storage</strong></td>
</tr>
<tr>
<td>First 50 TB / month</td>
</tr>
<tr>
<td>Next 450 TB / month</td>
</tr>
<tr>
<td>Over 500 TB / month</td>
</tr>
<tr>
<td><strong>Standard - Infrequent Access Storage</strong></td>
</tr>
<tr>
<td>First 50 TB / month</td>
</tr>
<tr>
<td>Next 450 TB / month</td>
</tr>
<tr>
<td>Over 500 TB / month</td>
</tr>
<tr>
<td><strong>Glacier Storage</strong></td>
</tr>
</tbody>
</table>

- **S3 bucket with Data**
  - $19.3K for 300TB egress per month
- **Amazon CloudFront**
  - No cost
  - $16.4K for 300TB egress per month, to U.S.

2017-01-11

2017-01-11

Architecting for the Cloud: Best Practices and Cost Analysis
Rate and Cost Shaping

- Concern of “unbounded costs”
- Can limit bandwidth or data volume to bracket cost to bound costs
- *Note additional infrastructure and workforce costs for TCO*

Us-west-2 to Coresite One Wilshire, Los Angeles, CA
$220 per month for 1Gbps
(~300TB/month at max)

AWS Direct Connect

Internet

corporate data center

users

S3 bucket with Data

AWS Import/Export Snowball

S3 export via USB3 disk
300TB once per month.
$3K per month

corporate data center

users

S3 bucket with Data
Cloud Economics

DEPLOYMENT TOPOLOGY
• Science data product generation at SDS
• Science data products moved to DAAC facilities
  – (copying large data volumes)
• End users access from DAAC
• Bottlenecks and cost impact of high network data stream
• **Shared data as interface** between SDS and DAAC
• No egress nor external network limitations between SDS and DAAC
• DAAC still incurs end-user egress costs.
TOTAL COST OF OWNERSHIP

TCO
Total Cost of Ownership (TCO)

- Factoring in compute, storage, network, topology, storage tiering, etc.
- Example monthly rollup for forward stream processing TCO in AWS
  - Does not show other costs e.g. cloud development
  - *(This example uses public “rack rates”)*
LARGE-SCALE CONSIDERATIONS
• Growing volumes of SAR Instrument data remain unprocessed to higher level science products
  – NASA Instruments: UAVSAR, AirMOSS, EcoSAR
  – ASF: ERS/ALOS/RADARSAT/Sentinel-1 archives
  – WInSAR: Envisat archive
  – ESA: Sentinel-1 A/B
  – ASI: COSMO SkyMed archive

• Historically, SAR data products are hand built
  – Dependent upon specialized knowledge by investigator
  – Expensive, laborious and slow to create (days)
  – Unable to scale processing capacity

Sentinel-1A interferogram
April 4 & 20, 2016
Auto-Scaling Science Data System

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

- Auto Scaling group policies
  - scaling size
  - rest period

- Metric alarms
  - E.g. condition: work queue size > 20

Auto scaling enabling runs of over 100,000 vCPUs

ASG max set to 1000 instances x 32 vCPUs
Optimizing for Scaling In/Out Events

Scaling up (scale out)

- Auto scaling group batching and timeout periods
- Scale up in group sizes of **multiples of availability zones (AZ)** to minimize AZ load rebalancing terminations

Scaling down (scale in)

- What policy to set to scale down?
  - E.g. CPU / network utilization
  - Domain knowledge only known within the compute instances
    - Self-terminating instances
    - “harakiri” / “suppuku”
• “The Amazon EC2 Service Level Agreement commitment is 99.95% availability for each Amazon EC2 Region.”
  – Example: expect about 1 failure per 2000 instances
    • E.g. “EC2 has detected degradation of the underlying hardware hosting one or more of your Amazon EC2 instances in the us-west-2 region. Due to this degradation, your instance(s) could already be unreachable.”
    • E.g. disk failures, network issues, etc.

• Plan for failures!
- Running at large-scales (and spot market...) forces the data system to be more resilient to compute failures
- Treat failure as the norm instead of the exception
“Thundering Herd”

• Large fleet of auto-scaled compute instances calling same services at same time
  – “API rate limit exceeded”

• “Jittering” the API calls
  – Introduce *randomizations* to API calls
  – Distributes load on infrastructure
Large-Scaling API Rate Limits

- Example showing rate limits when scaling up to 4000+ EC2 compute instances and scaling down

![Graph showing rate limit exceeded](chart.png)
SUMMARY
Key Points

- Cloud use can be cheaper if can dive deeper in architecture design and TCO impacts

- Cost implications of Earth Science Data Systems in the Cloud:
  - Compute
  - Storage
  - Network
  - Deployment Topology

- Total Cost of Ownership (TCO) analysis

- At large scales, need to deal with scaling issues.

- Running on spot market for cost savings—but need resiliency

- Can achieve large-scale production while saving costs

- Fault tolerant science data systems can scale better in cloud computing environments

- Collocation
  - Data Lakes

- Benchmark at full-scale!
  - Test in full-scale production
  - Assess steady-state at full-scale
  - Monitor real-time metrics
“Data Lake”-extended

- It’s about collocation!
- Minimize data movement
- Maximize user services
- Run on *public cloud provider* or at an *on-premise data center*

Reduce redundancy and foster collocated services

Enabling multi-disciplinary data approach for analysis
“Hot Data Lake”

- Long-term **public cloud storage is expensive** at PB-scales
  - ...Unless we can negotiate deals with cloud vendors
- Use object store data lake for **“hot data”**
  - SDSes generate data into object stores
  - Object stores contain “fresh” / “hot” data as rolling storage
  - Offline moving of older data to DAAC for permanent storage
  - Automate caching of “hot data” back from DAACs