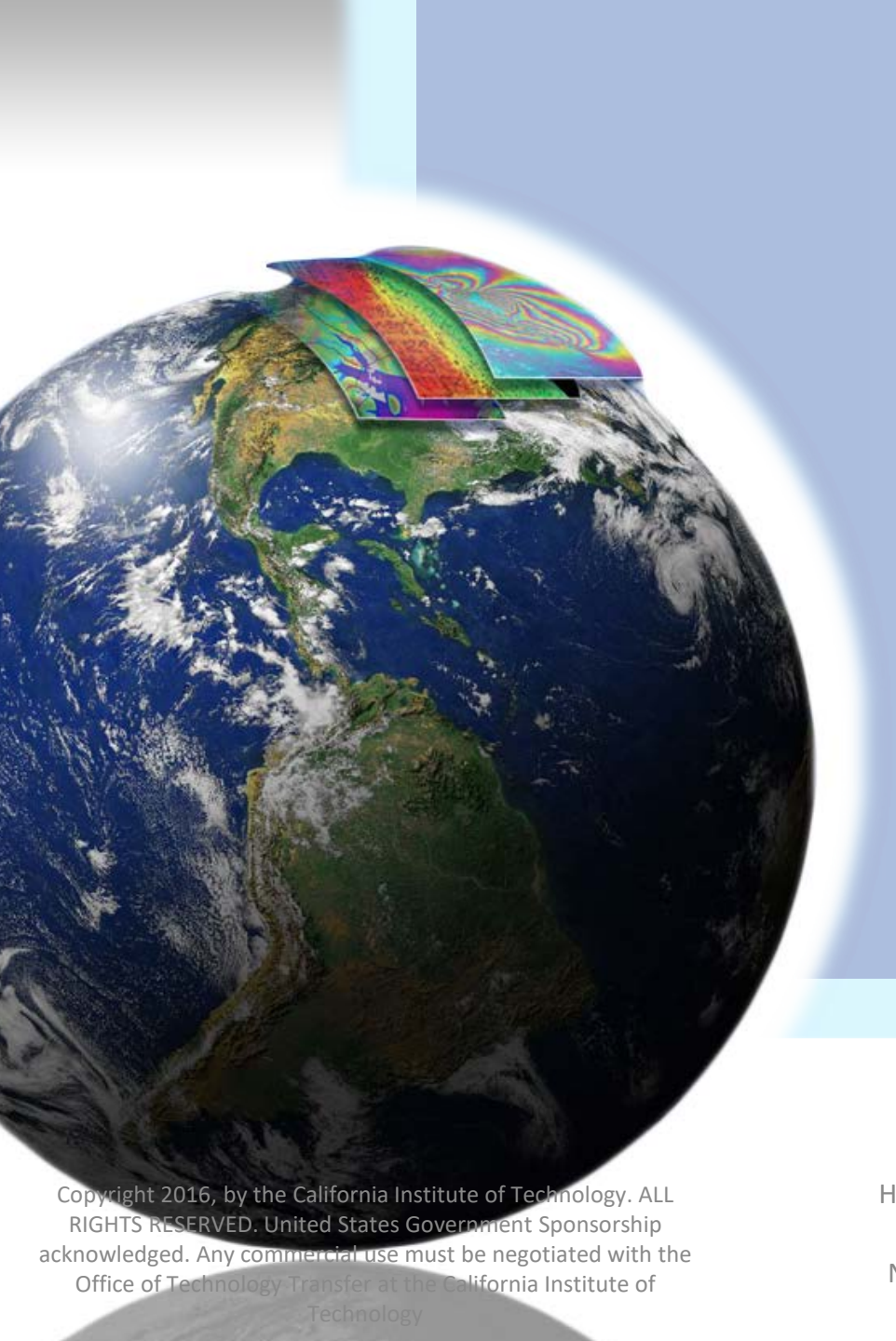


Jet Propulsion Laboratory
California Institute of Technology

Lessons Learned on Optimization Approaches for High-Scalability and High-Resiliency in AWS

Emerging Big Data Technologies for Geoscience
ESIP 2016 Winter Meeting
Wednesday, January 6, 2016
Washington, DC

Copyright 2016, 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

Hook Hua, Gerald Manipon, Brian Wilson, Michael Starch, Lan Dang, Phillip Southam, Paul Ramirez
NASA Jet Propulsion Laboratory / California Institute of Technology

Science Data Systems for Big Data

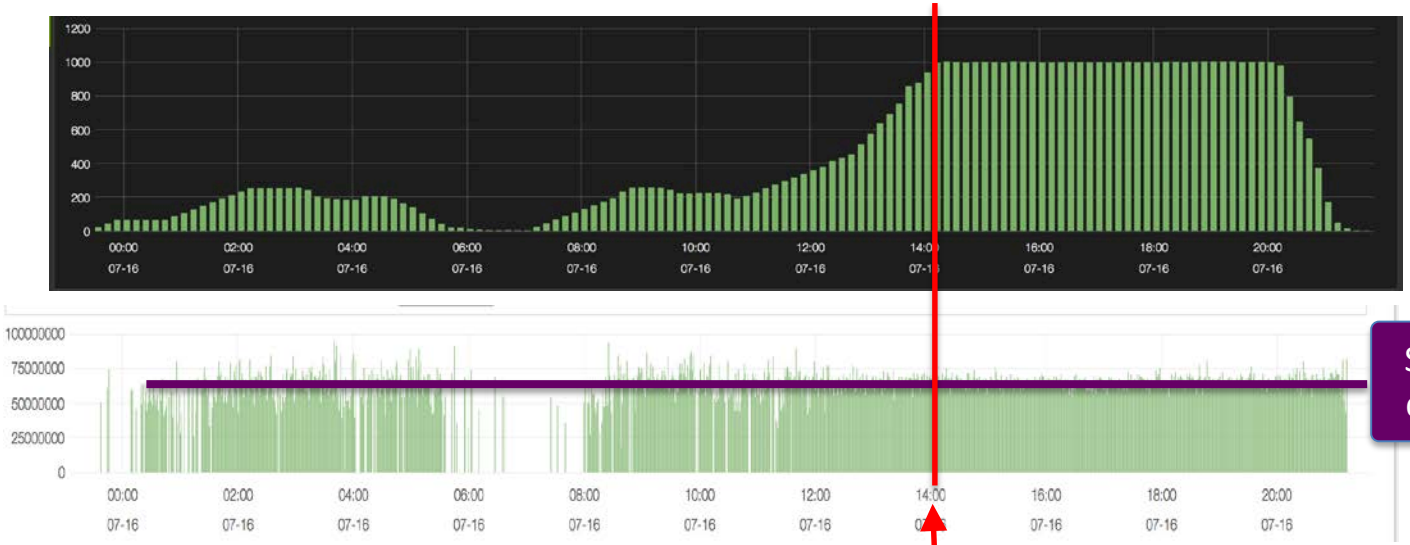
- More than just moving existing code to cloud resources
- **Hybrid cloud** computing give us flexibility
- At large scales, need to deal with **scaling issues**.
- Running on **spot market** for cost savings—but need **resiliency**
- **Fault tolerant** science data systems can scale better in cloud computing environments

Hybrid Cloud Science Data Systems

- OCO-2 L2 Full Physics processing operational in AWS
  - Processing of L2 full physics data products in Amazon cloud across multiple regions
  - Scaled up **thousands** of compute nodes
  - Demonstrated capability of higher internal data throughput rates than NISAR needs

compute nodes over time

Per node transfer rate over time



Scalable internal data throughput

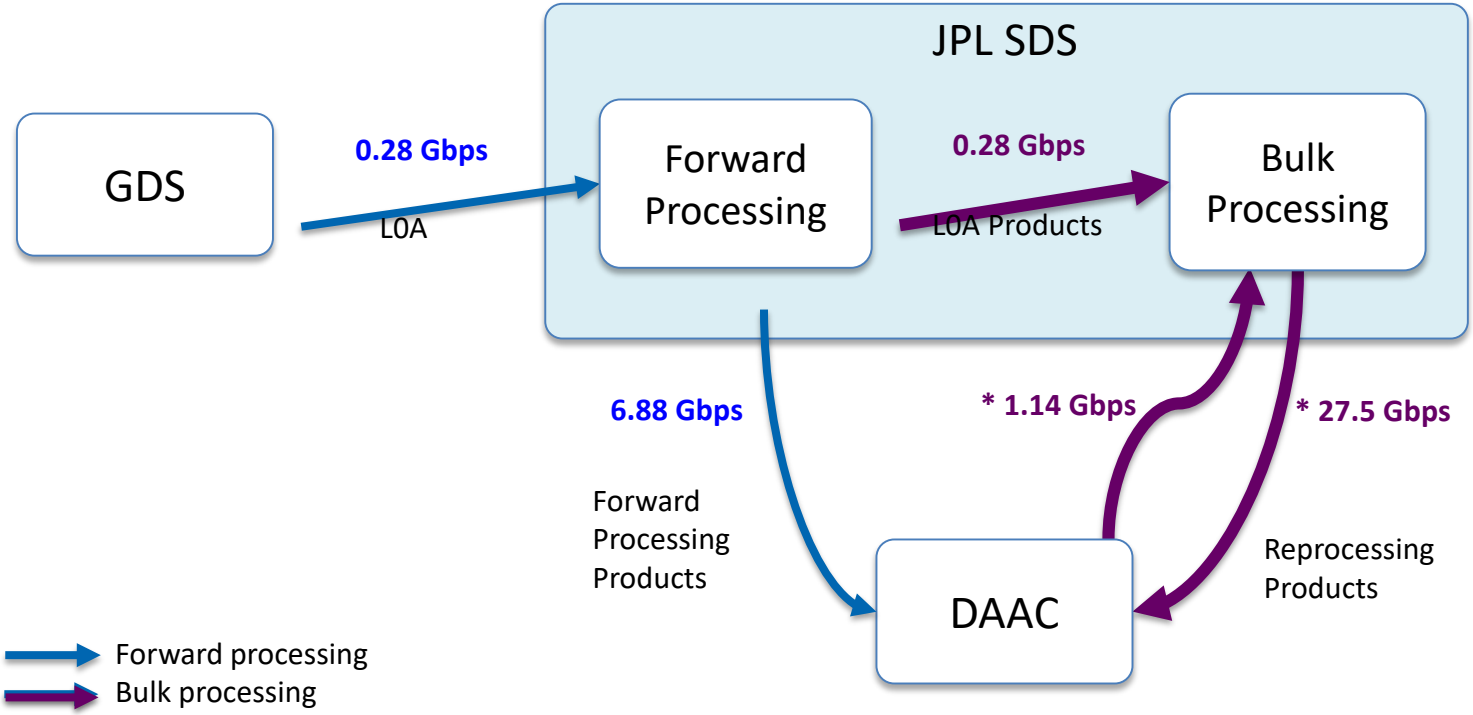
@ 32,000 PGEs on 1,000 nodes

Hybrid Cloud Science Data Systems

- Advanced Rapid Imaging & Analysis for Monitoring Hazards (ARIA-MH)
  - Continuous NRT processing of COSMO-SkyMed (CSK) for 3-years
- MEaSUREs WVCC A-Train data fusion
  - Continuous data production
- Big Data Analytics & Machine Learning for Quality Assessment
  - Assessing quality measures for existing interferograms
  - Improving generation of interferograms
- SAR Science Data Processing Foundry
  - Starting with UAVSAR, AirMOSS, other airborne and space borne sensors
  - Expanding science areas e.g. biomass
- ESA Sentinel-1A processing support
  - Facilitate mature TRL9 SDS for NISAR and SWOT readiness in 2017
  - Hazard response: currently only open operational SAR platform
- NISAR SDS architecture
- SWOT SDS architecture

NISAR SDS

- Average Input Data Volume to SDS: 24 Tbits/day (3 TB per day)
- Average Daily Production Volume: 84 TB per day
- Total Mission Data Volume: 89 PB over 3-years



General Cloud Strategy

- Leverage cloud-scaling for **storage** and **compute**
- **Stream** input data into AWS S3 object storage
- **Scale up** compute nodes to run in AWS EC2
  - Compute nodes localize input data from AWS S3 object storage (highly parallel)
  - Compute nodes publish processed soundings back onto AWS S3 object storage (highly parallel)
- Asynchronously **move** results from AWS S3 back to JPL *(or on-premise cloud)*
  - Can push results to JPL continuously
- Avoid vendor **lock-in**
  - Approach must be able to run independently from AWS
  - Cannot use AWS-only services
  - Leverage OpenStack compatibility

Science Data Production in AWS

- More than just moving existing code to cloud resources
- Assess capacity and demand of different regions
  - E.g. US-west-1 (Northern California), US-west-2 (Oregon), US-east-1 (Virginia)
  - US GovCloud (Oregon) **for ITAR and Export Control**
- Benchmark performance on various EC2 instance types
  - Non-obvious results
- Compute cost model types
  - On-demand
  - Reserved instance
  - Spot market
    - Compute node terminations when out-bid!
- Auto-scaling of compute nodes
- At large scales, need to deal with **scaling issues**.

Use Case

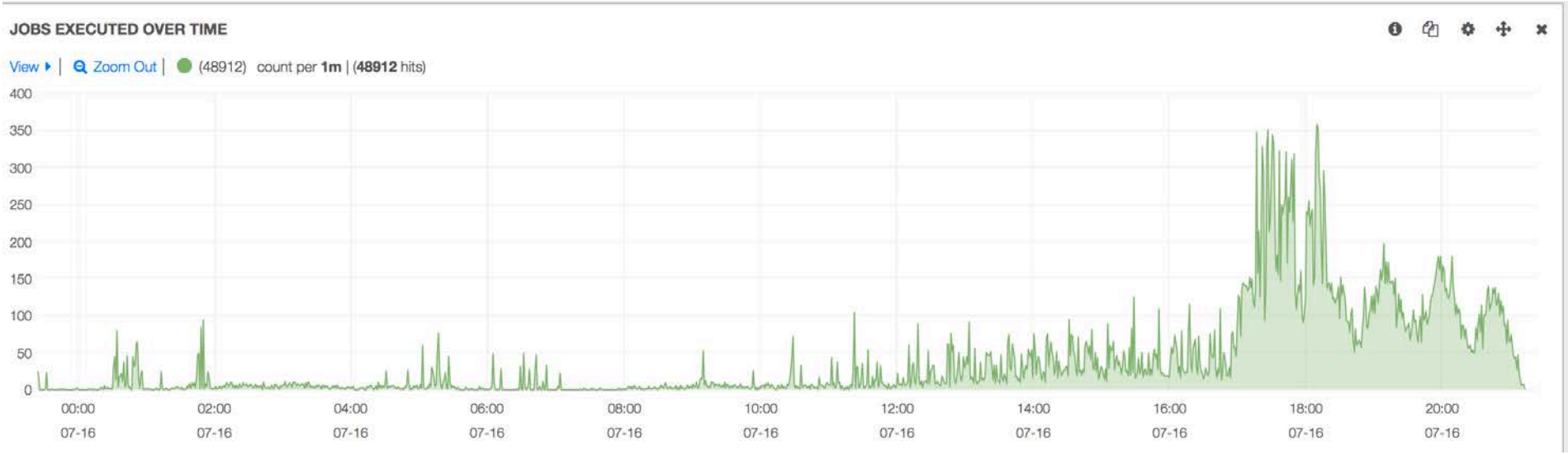
OCO-2 L2 PRODUCTION USING HYSDS IN AWS

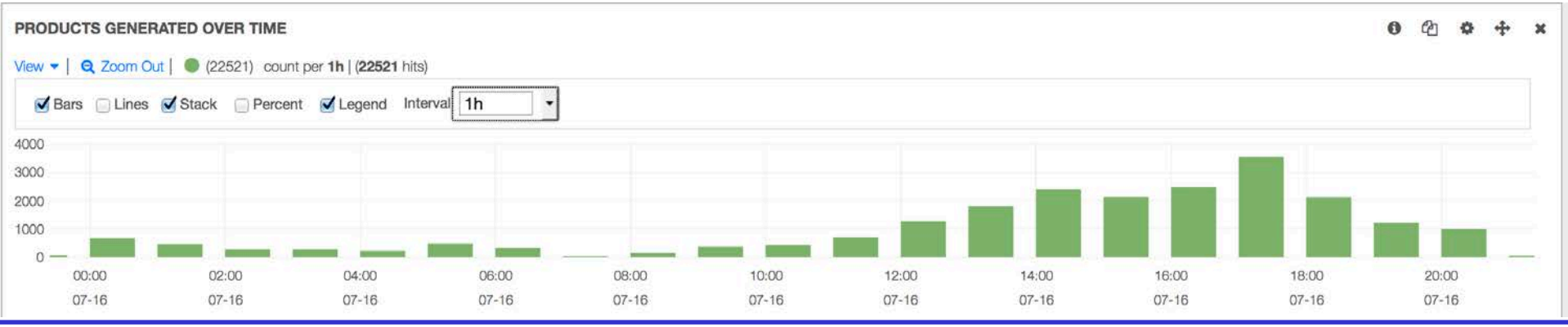
Example Production Run

- **40X real-time processing**
  - 465 granules for October 2014—*processed in under a day*
- Data volumes
  - 1.6TB data products generated
  - 12.5TB data fed into PGEs
- Science Data System
  - EC2 in US-West-2 (Oregon)
- Storage
  - S3 in US-West-1 (Northern California)
- Compute
  - EC2 in US-US-West-2 (Oregon)
- Auto-scaling
  - 1000 x cc2.8xlarge / US-West-2 (Oregon)
  - **32,000 x l2_fp simultaneous processors**

Job and Data Products Generation Profile

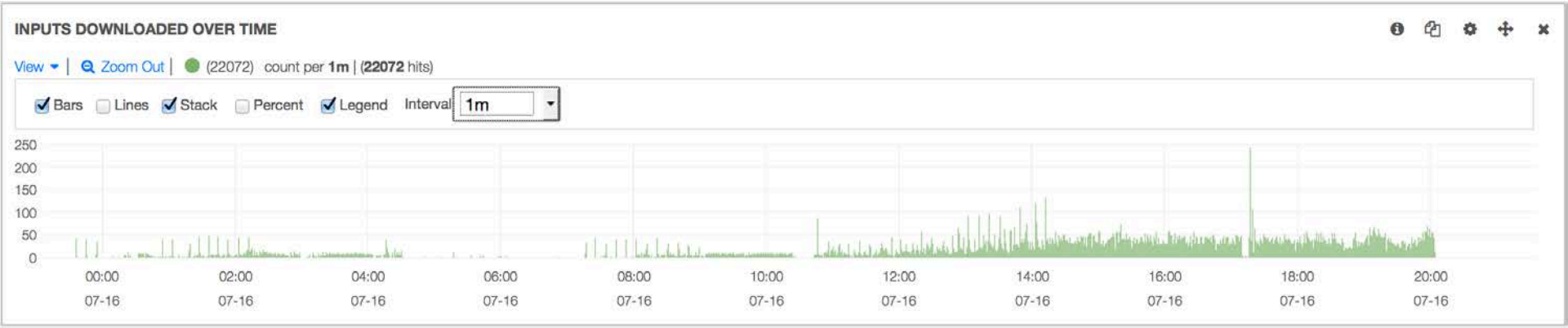
- Peaked 21,000 jobs per hour
- Peaked 3,500 data products generated per hour

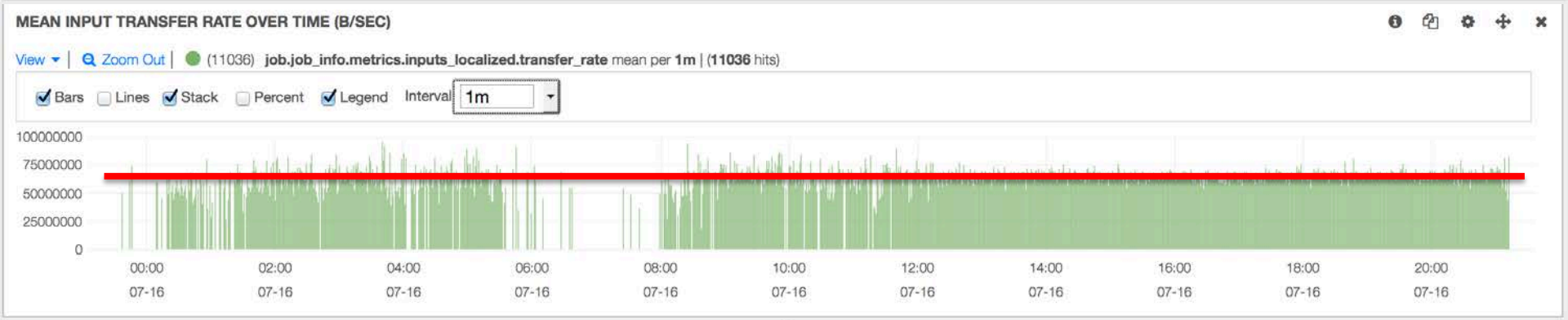




Data Inputs into l2_fp Compute Instances

- US-West-2/EC2 and US-West-1/S3 Data Throughput





INPUT TRANSFER TIME

122200.998 seconds (total)

Query	min	max	mean	total	variance	std_deviation	sum_of_squares
●	0.210 seconds	196.344 seconds	5.536 seconds	122200.998 seconds	40.187 seconds	6.339 seconds	1563560.101 seconds

INPUT TX RATE

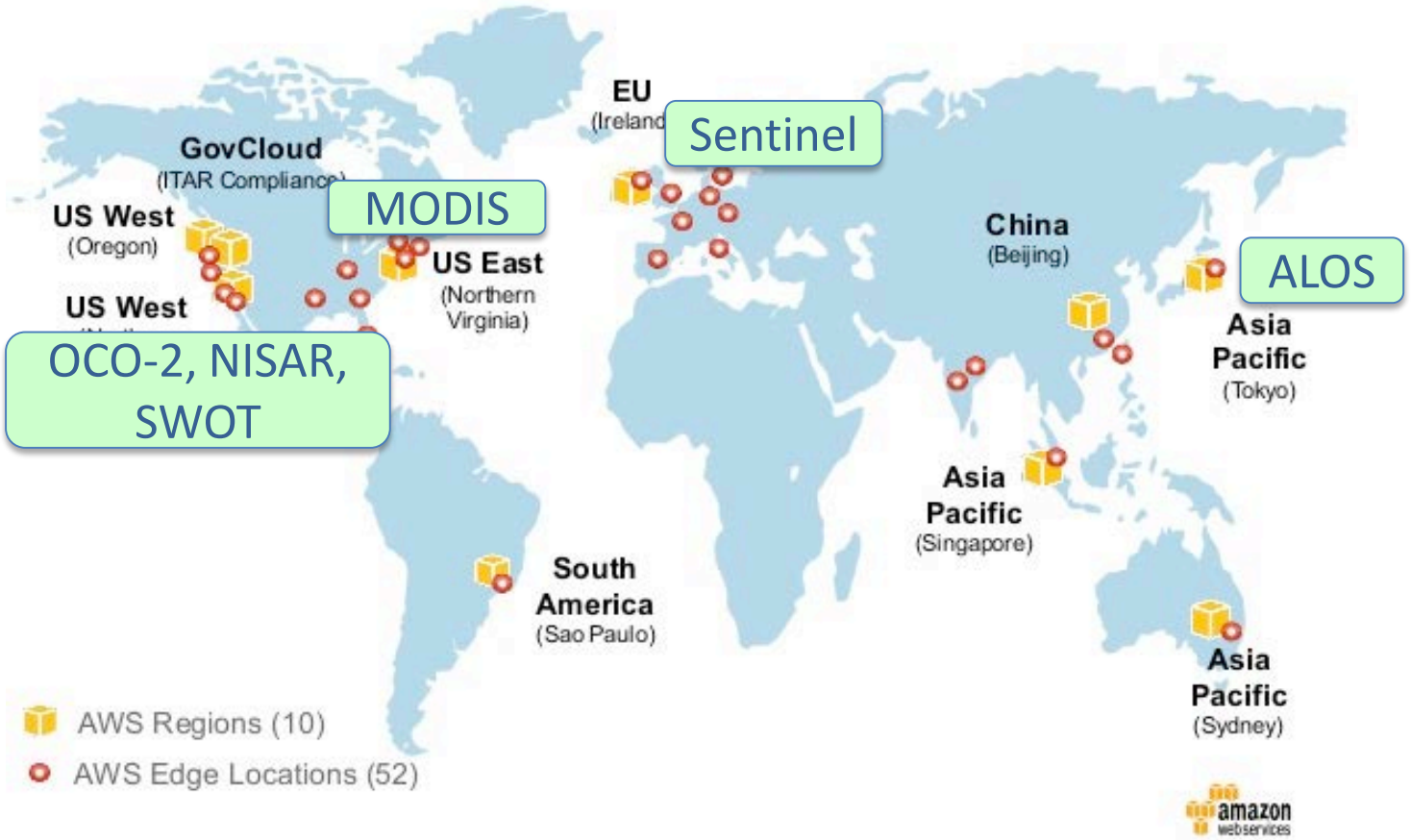
62.13MB per second (mean)

Query	min	max	mean	total	variance	std_deviation	sum_of_squares
●	82.38B per second	205.30MB per second	62.13MB per second	1.31TB per second	4.27PB per second	66.09MB per second	173.20EB per second

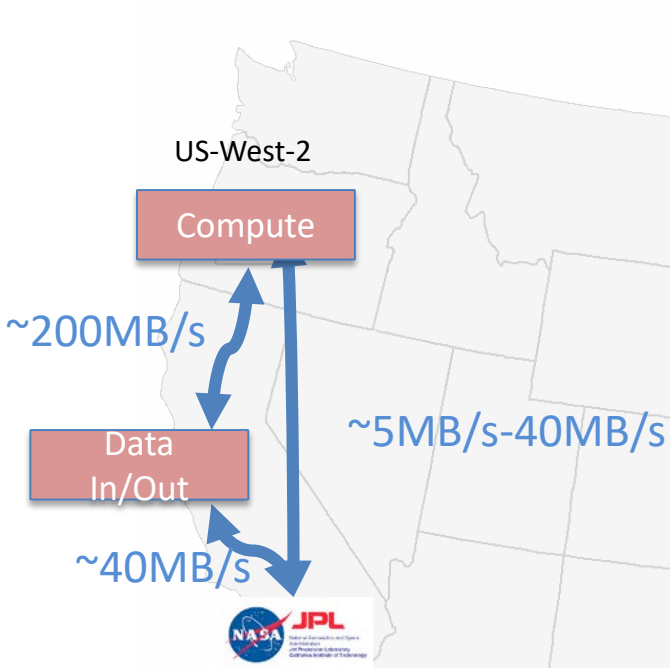
LARGE-SCALE CONSIDERATIONS

Global-scale Processing Regions

- Factoring in data locality
- Move compute closer to data



Optimize for Transport and Compute

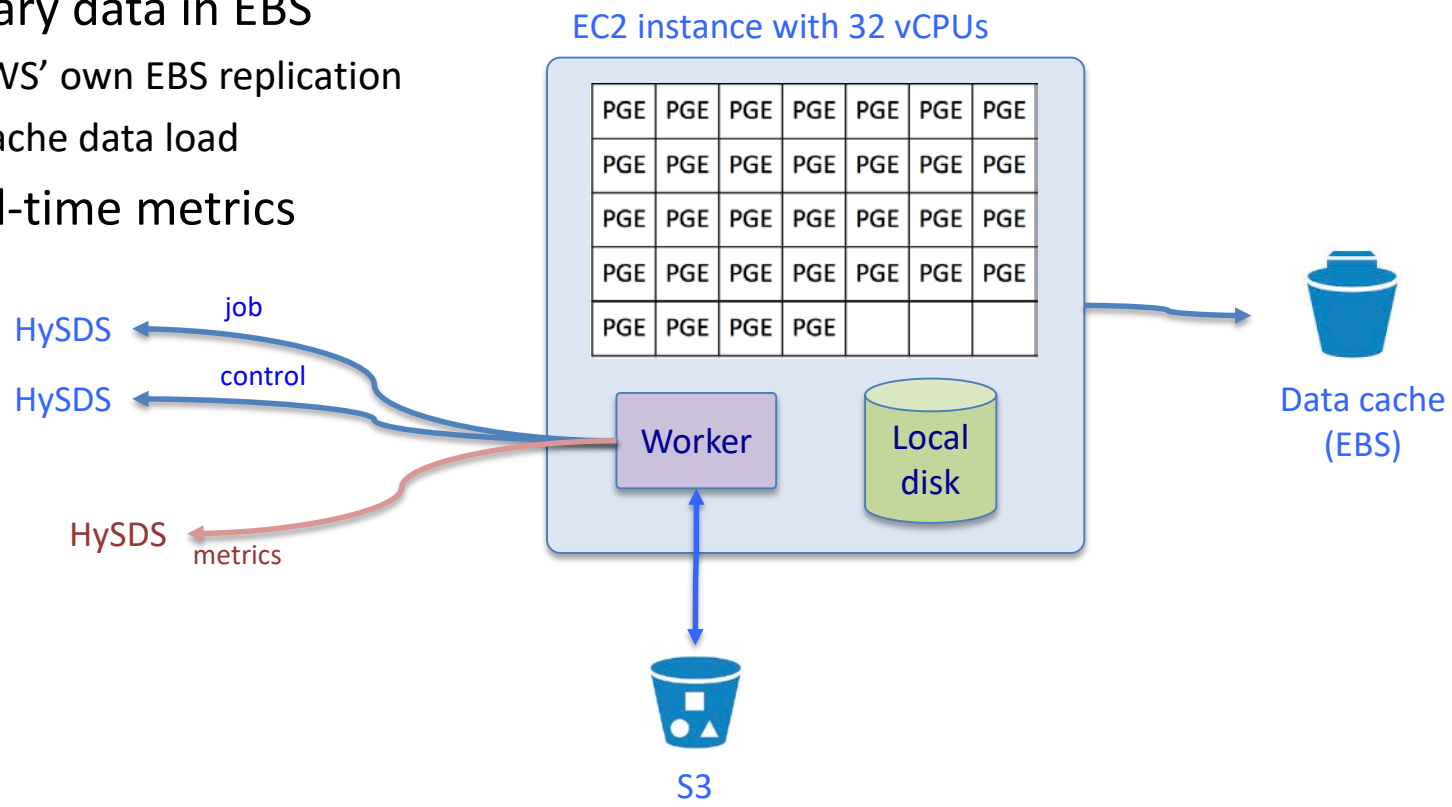


- Entry points into AWS are different
  - JPL to US-West-1 on 10Gbps
  - JPL to US-West-2 over internet
- Transport approach
  - *Stream* data from JPL to S3/US-West-1
  - Maximize JPL-AWS network speeds
- Compute in other AWS regions
  - E.g. EC2/US-West-2
  - Move data from S3/US-West-1
  - Results moved back to S3/US-West-1
- Asynchronously localize results back to JPL from S3/US-West-1

- Raw performance over standard HTTP.
- Data accelerators yield high throughput.

Maximize Per Compute Instance

1. Horizontally scaling up compute workers
   - Fine-grain parallelization (gnu parallel)
   - Coarse-grain parallelization (AWS EC2)
2. Cache ancillary data in EBS
   - Leverage AWS' own EBS replication
   - Scaling of cache data load
3. Monitor real-time metrics



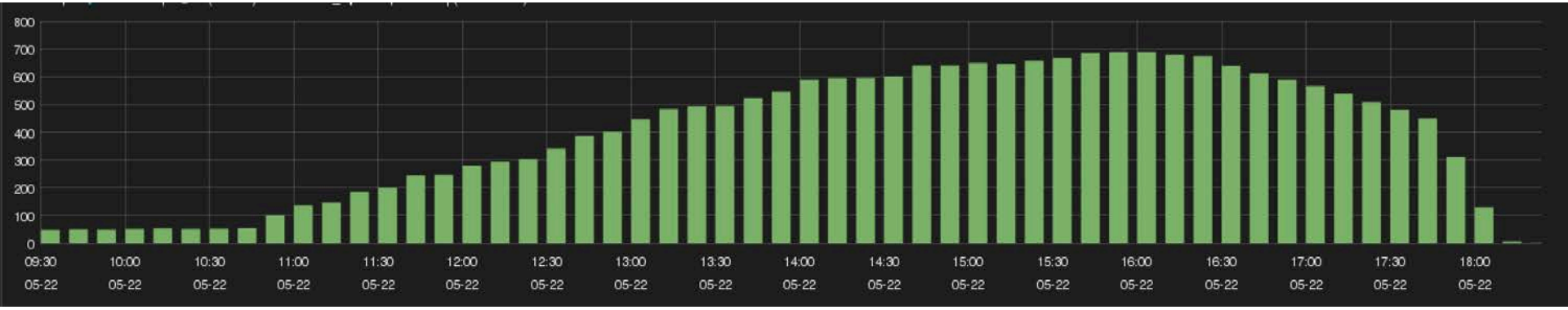
Auto-Scaling Science Data System

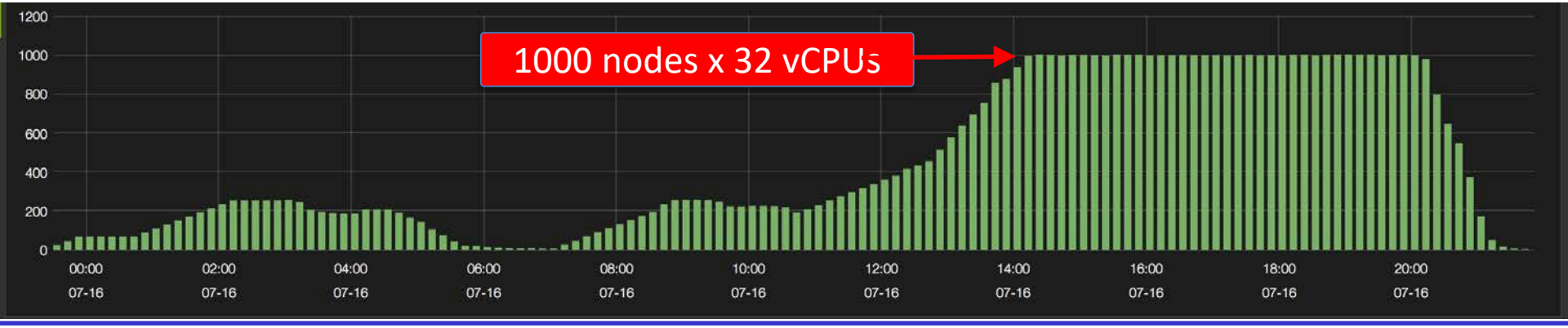
- ***The size of the science data system compute nodes can automatically grow/shrink based on processing demand***

- Auto-scaling group policies
  - Auto-scaling size = 21
  - Metric alarm of queue size > 20
  - Auto-scaling rest period of 60-seconds

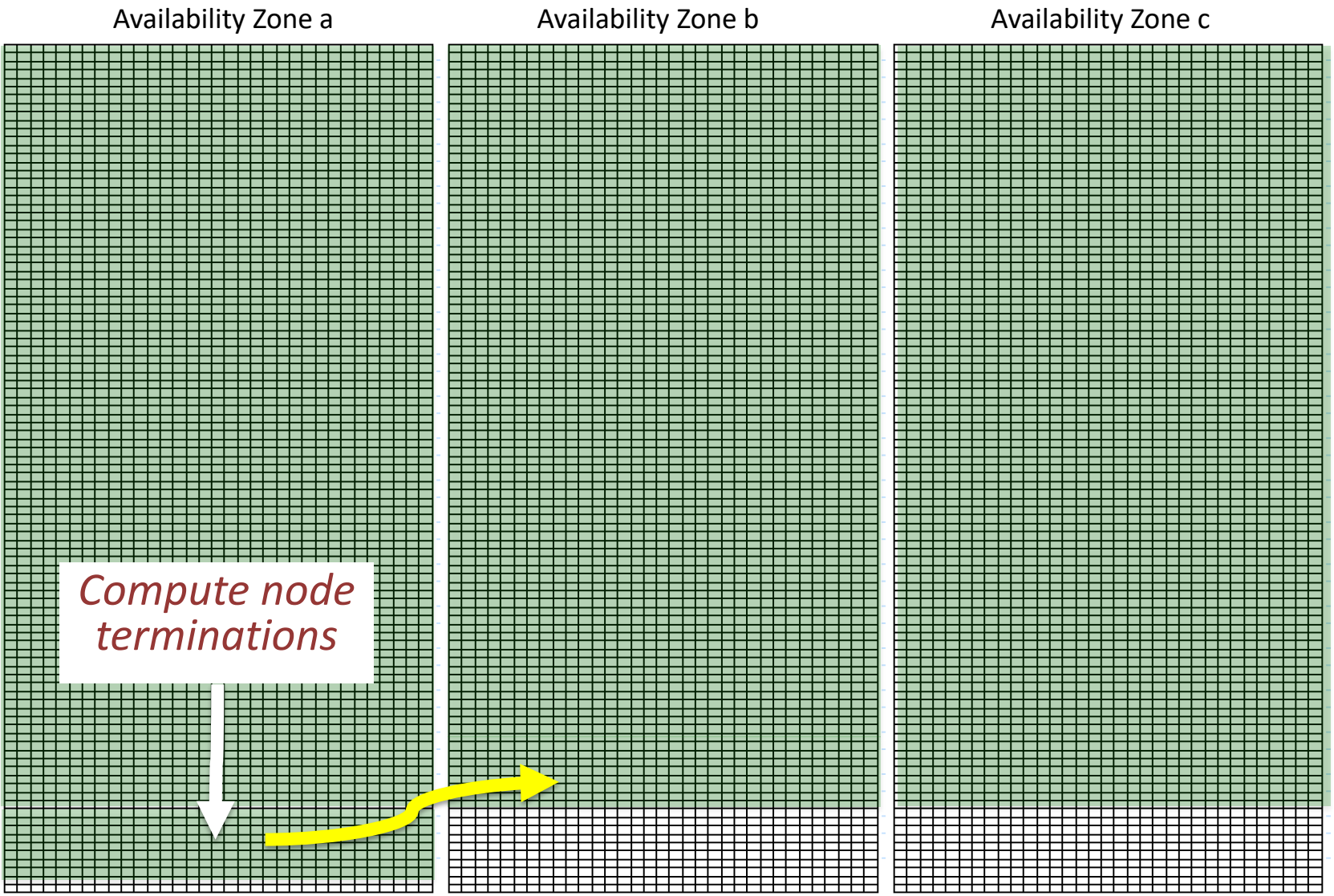
Auto-scaling tests to 3000 compute workers

96,000 x I2_fp simultaneous processors



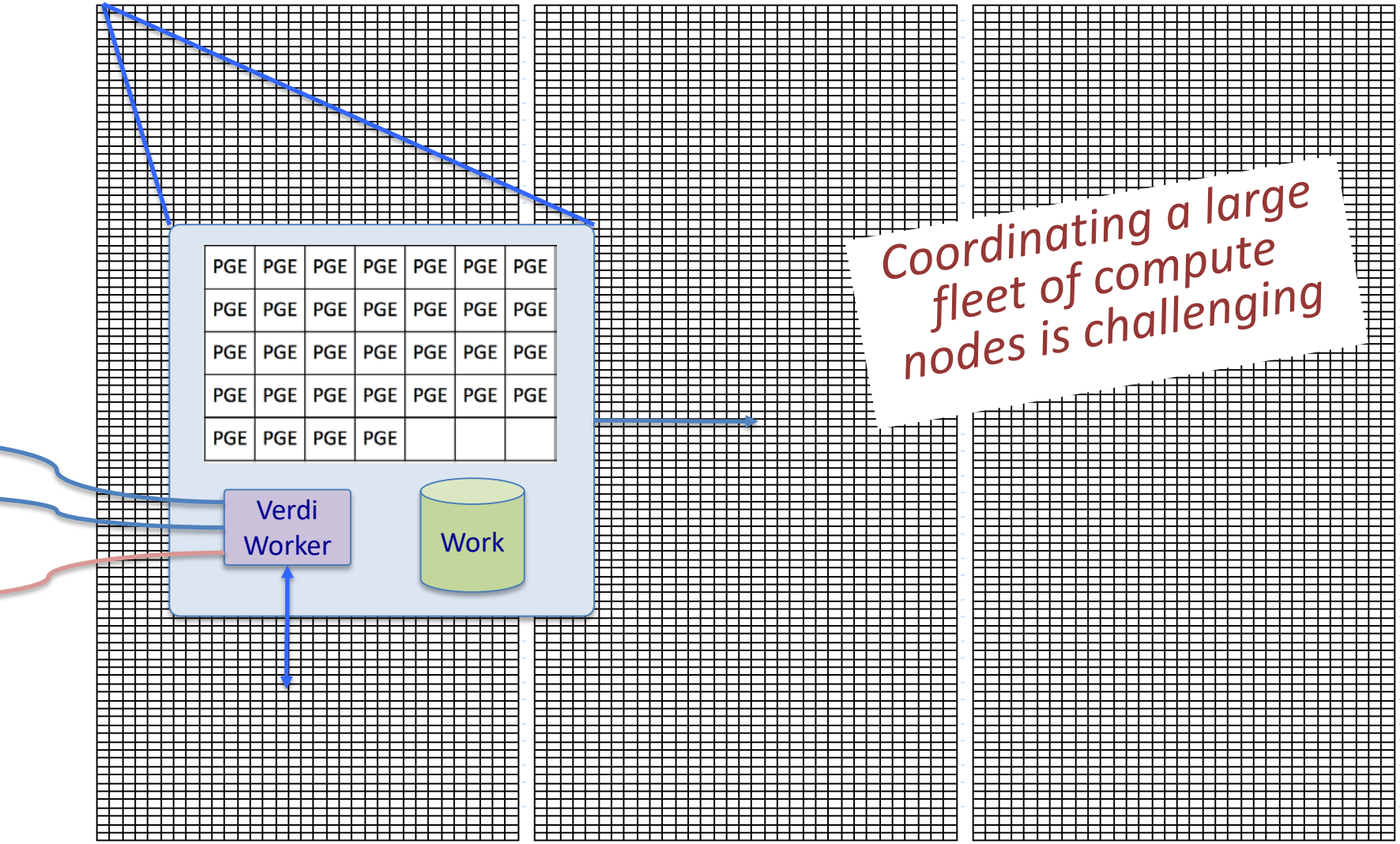


Availability Zone Load Rebalancing



Large Scale Considerations

1000 instances x 32 vCPUs each



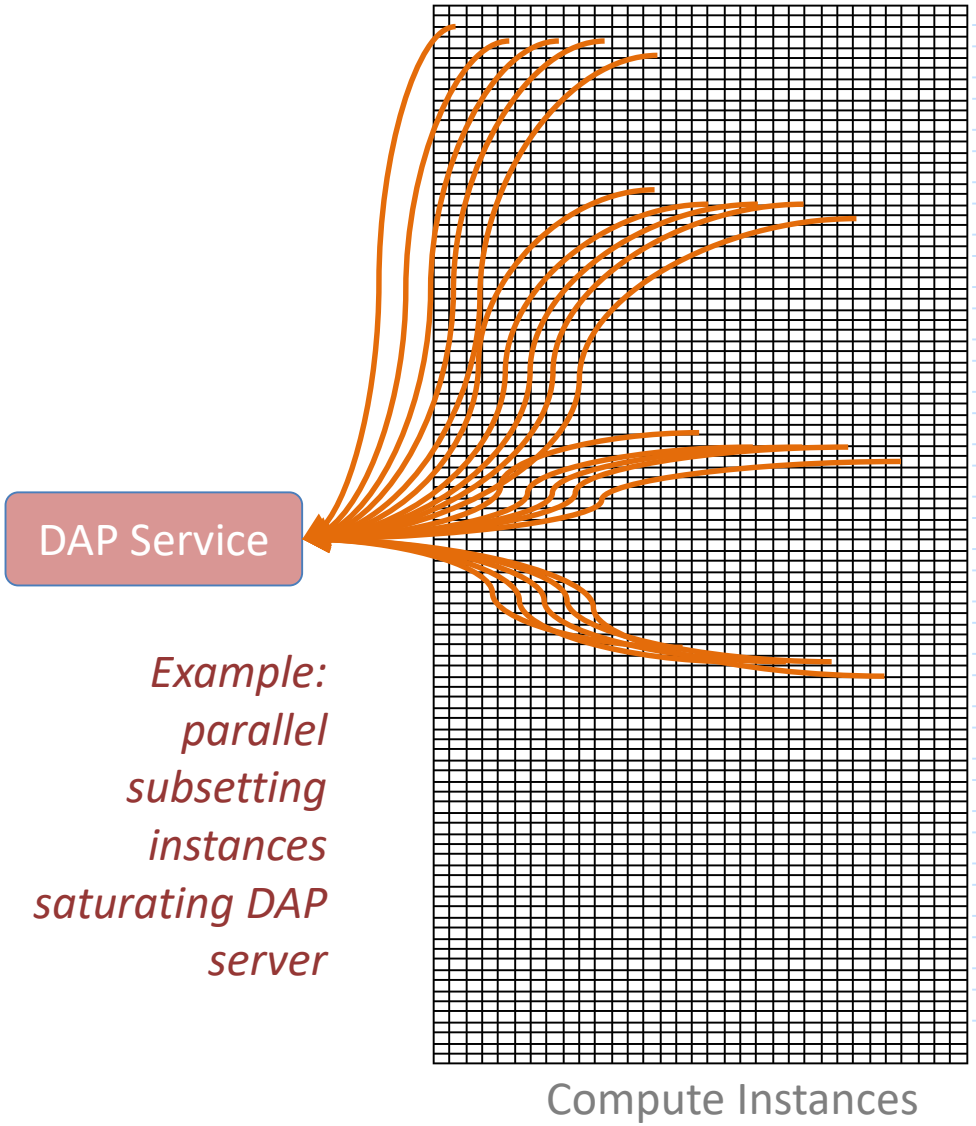
Cloud Reliability

Monthly Uptime Percentage	Service Credit Percentage
Less than 99.95% but equal to or greater than 99.0%	10%
Less than 99.0%	30%

- The Amazon EC2 Service Level Agreement commitment is 99.95% availability for each Amazon EC2 Region.
  - means 1 node failure per 2000 nodes
    - 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

"Thundering Herd"

- Fleet of compute instances hitting same services at same time
  - *"API rate limit exceeded"*
- Mitigation plan
  - Introduce randomizations to API calls
  - Distributes load on infrastructure
- AWS already does some randomizations



Compute Sizing

CLOUD ECONOMICS

AWS EC2 Physical Hardware

instance	vCPU	physical hardware	cores	threads	memory	memory-cpu ratio	disks
m2.4xlarge	8	Intel Xeon E5-2665 @ 2.40 GHz 1 processor, 8 threads	8	8	68.4	8.55	2 x 840
cc2.8xlarge	32	Intel Xeon E5-2670 @ 2.59 GHz 2 processors, 16 cores, 32 threads	16	32	60.5	1.89	4 x 840
m3.2xlarge	8	Intel Xeon E5-2670 @ 2.60 GHz 1 processor, 8 threads	4	8	30.0	3.75	SSD 2 x 80
c3.8xlarge	32	Intel Xeon E5-2680 v2 @ 2.80 GHz 2 processors, 16 cores, 32 threads	16	32	60.0	1.88	SSD 2 x 320
r3.8xlarge	32	Intel Xeon E5-2670 v2 @ 2.49 GHz 2 processors, 16 cores, 32 threads	16	32	244.0	7.63	SSD 2 x 320

- m2.4xlarge instances
  - *Hyperthreading disabled by AWS*
  - Instance occupied the entire physical node, which has no multi-tenancy contention
    - Note: this is no longer the case
  - No SSDs
- c3 are more optimized for compute than r3's memory
  - 2.80GHz for c3
  - 2.49GHz for r3
- cc2 are older but cheaper
- 32 vCPUs != 32 cores
  - Need to account for performance of l2_fp under 32 threads

AWS EC2 Benchmarks for l2_fp

								test runs				
instance	vCPU	physical hardware	cores	threads	memory	memory-cpu ratio	disks	soundings per job	concurrent processors	# test runs	mean runtime (s)	mean runtime per sounding per processor (m)
m2.4xlarge	8	Intel Xeon E5-2665 @ 2.40 GHz 1 processor, 8 threads	8	8	68.4	8.55	2 x 840	32	8	42	1312	5.8
cc2.8xlarge	32	Intel Xeon E5-2670 @ 2.59 GHz 2 processors, 16 cores, 32 threads	16	32	60.5	1.89	4 x 840	32	32	14365	615	10.3
m3.2xlarge	8	Intel Xeon E5-2670 @ 2.60 GHz 1 processor, 8 threads	4	8	30	3.75	SSD 2 x 80					
c3.8xlarge	32	Intel Xeon E5-2680 v2 @ 2.80 GHz 2 processors, 16 cores, 32 threads	16	32	60	1.88	SSD 2 x 320	32	32	126	461	7.7
r3.8xlarge	32	Intel Xeon E5-2670 v2 @ 2.49 GHz 2 processors, 16 cores, 32 threads	16	32	244	7.63	SSD 2 x 320	32	32	131	489	8.2

- Initial comparison benchmarks
  - Run 32 soundings per job
    - Run 8 x l2_fp concurrent processors (via gnu parallel) on m2.4xlarge instance
    - Run 32 x l2_fp concurrent processors (via gnu parallel) on c3.8xlarge and r3.8xlarge instances
  - Mean runtime based on average of soundings in granule set
- M2.4xlarge has best performance for l2_fp
  - m2.4xlarge instance occupies 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.
- Slightly faster Intel Xeon E5-2680 v2 @ 2.80 GHz on c3.8xlarge more useful
- If want to run jobs for 1-hour
  - m2.4xlarge
    - use 80 soundings per job
  - c3.8xlarge
    - use 256 soundings per job

“Cost Per Sounding” Estimates

prefered minimun job duration (m) :	50

	test runs					prefered job size		on-demand (US-West-1)		spot (US-West-1)		on-demand (US-West-2)		spot (US-West-2)	
instance	soundings per job	concurrent processors	# test runs	mean runtime (s)	mean runtime per sounding per processor (m)	# soundings	# soundings (evenly distributed processors)	mean costs per job	$ per sounding	mean costs per job	$ per sounding	mean costs per job	$ per sounding	mean costs per job	$ per sounding
m2.4xlarge	32	8	42	1382	5.8	69	7	$0.4223	0.01320	$0.0250	0.00078	$0.4138	0.01293	$0.0384	0.00120
cc2.8xlarge	32	32	14365	615	10.3	156	16					$0.3417	0.01068	$0.0462	0.00144
m3.2xlarge															
c3.8xlarge	32	32	126	461	7.7	208	22	$0.2448	0.00765	$0.2151	0.00672	$0.2151	0.00672	$0.3073	0.00960
r3.8xlarge	32	32	131	489	8.2	196	22	$0.4238	0.01324	$0.0543	0.00170	$0.3803	0.01189	$0.3803	0.01189

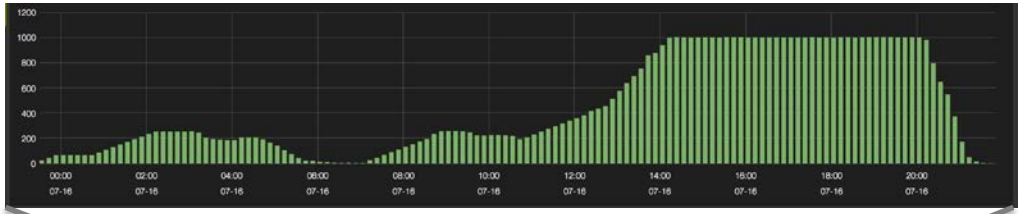
- Raw costs estimates. No overhead nor margin applied.
- Pricing is estimated based on computed “*$ per sounding*” metric estimate
- Compute price estimate from 2015-05-12
- Estimates used to determine best instance type to use for bulk processing
- Spot prices fluctuate based on market demand
  - m2.4xlarge spot price appears to hold steady
  - Have observed c3.8xlarge fluctuate $0.4 to $1.9 per hour
    - Lowest spot prices are not available all-day
- *Long-term cost-effective strategy is to use hybrid-cloud with on-premise baseline processing (“free”) with bursting to AWS reserved instances*

High-Resiliency & Spot Market

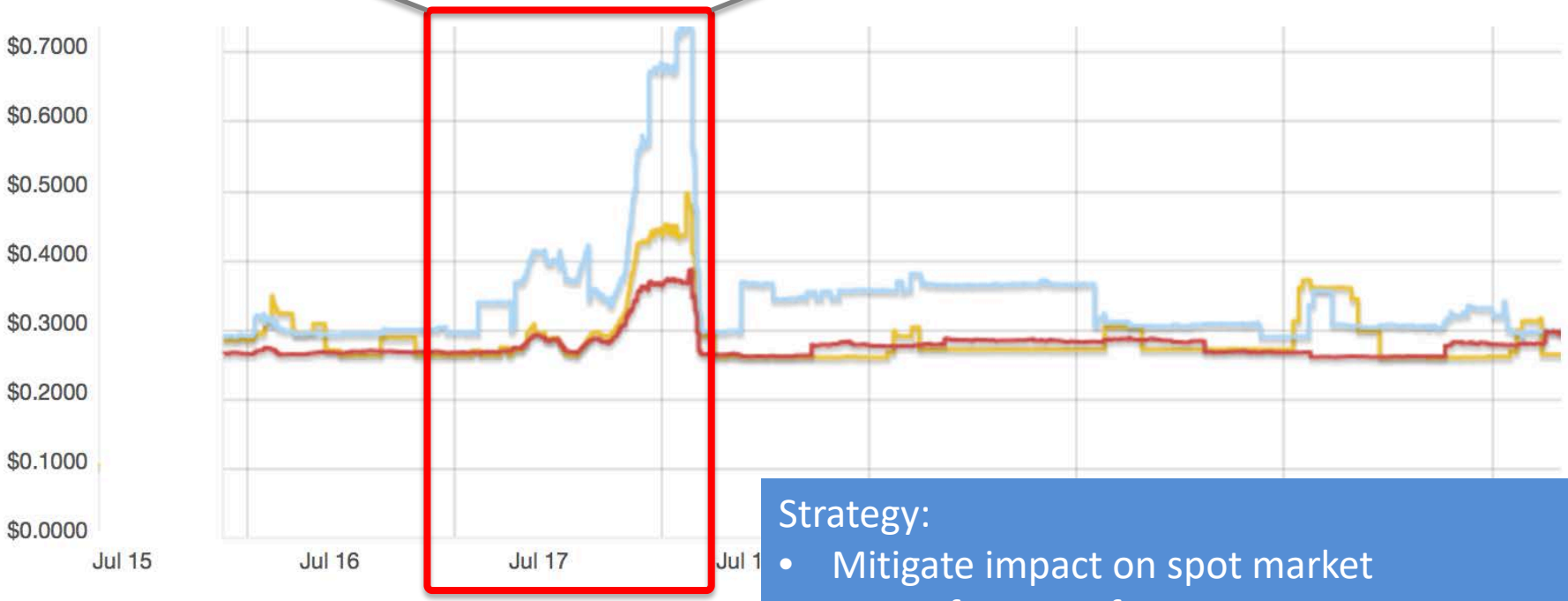
					US-West-2 (Oregon)							
					Hourly Costs				Per vCPU Costs			
instance	**vCPU**	**memory**	**memory-cpu ratio**	**disks**	**on-demand ($/hr)**	**reserved 1-yr upfront ($/hr)**	**reserved 3-yr upfront ($/hr)**	**spot linux ($/hr)**	**on-demand ($/cpu/hr)**	**reserved 1-yr upfront ($/cpu/hr)**	**reserved 3-yr upfront ($/cpu/hr)**	**spot linux ($/cpu/hr)**
m2.4xlarge	8	68.4	8.55	2 x 840	$1.0780	$0.4087	$0.244	$0.1000	$0.1348	$0.0511	$0.0306	$0.0125
cc2.8xlarge	32	60.5	1.89	4 x 840	$2.0000	$0.9131	$0.613	$0.2705	$0.0625	$0.0285	$0.0192	$0.0085
m3.2xlarge	8	30.0	3.75	SSD 2 x 80	$0.6160	$0.3750	$0.230	$0.0700	$0.0770	$0.0469	$0.0288	$0.0088
c3.8xlarge	32	60.0	1.88	SSD 2 x 320	$1.6800	$0.9920	$0.628	$2.4001	$0.0525	$0.0310	$0.0196	$0.0750
r3.8xlarge	32	244.0	7.63	SSD 2 x 320	$2.8000	$1.4860	$0.982	$2.8000	$0.0875	$0.0464	$0.0307	$0.0875
c3.xlarge	4	7.5	1.88	SSD 2 x 40	$0.2310	$0.1370	$0.087	$0.0353	$0.0578	$0.0343	$0.0218	$0.0088

- 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
- HySDS able to **self-recover** from spot instance terminations
- *Running in spot market forces data system to be more **resilient** to compute failures*

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”

SUMMARY

Key Points

- More than just moving existing code to cloud resources
- **Hybrid cloud** computing give us flexibility
- 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
- **Test at full-scale**
  - Test in full-scale production
  - Assess steady-state at full-scale
  - Monitor real-time metrics