IFS (RAPS) on AWS – Successes and Challenges

23 September 2021

Brian Etherton – Numerical Weather Prediction Scientist, Maxar

With contributions from:
Stefan Cecelski, Christopher Cassidy, Travis Hartman, and Mehzabeen Hoosein – Maxar
Ioan Hadade, Iain Miller, and Umberto Modigliani – ECMWF
Sami Saarinen – ECMWF (Retired)

The opinions expressed in this presentation and on the following slides are solely those of the presenter and not necessarily those of Maxar.

Overall Scope of Project

- Maxar and ECMWF collaborated on defining project goals with objective metrics and lessons to be learned for running the modeling systems contained in the Real Applications on Parallel Systems (RAPS) on Amazon Web Services (AWS) cloud computing resources.

- Those systems are:
  - The 'high-resolution forecast' configuration of the Integrated Forecast System (IFS, CY45R1+)
  - The Nucleus for European Modelling of the Ocean (NEMO)
  - The wave model (WAM)

- The goals of the work are the following:
  - Phase 1 – can IFS compile and run in the cloud at all? What is needed to achieve this?
  - Phase 2 – can IFS, coupled with NEMO, run in the cloud? How does it scale?
  - Phase 3 – when increasing the resolution some, and enabling I/O, how well does the cloud perform? Does it scale well?
  - Phase 4 – getting as close to an operational single run configuration as possible, how does the cloud perform? Does it scale well? How is the strong scaling?

Overall Scope of Project

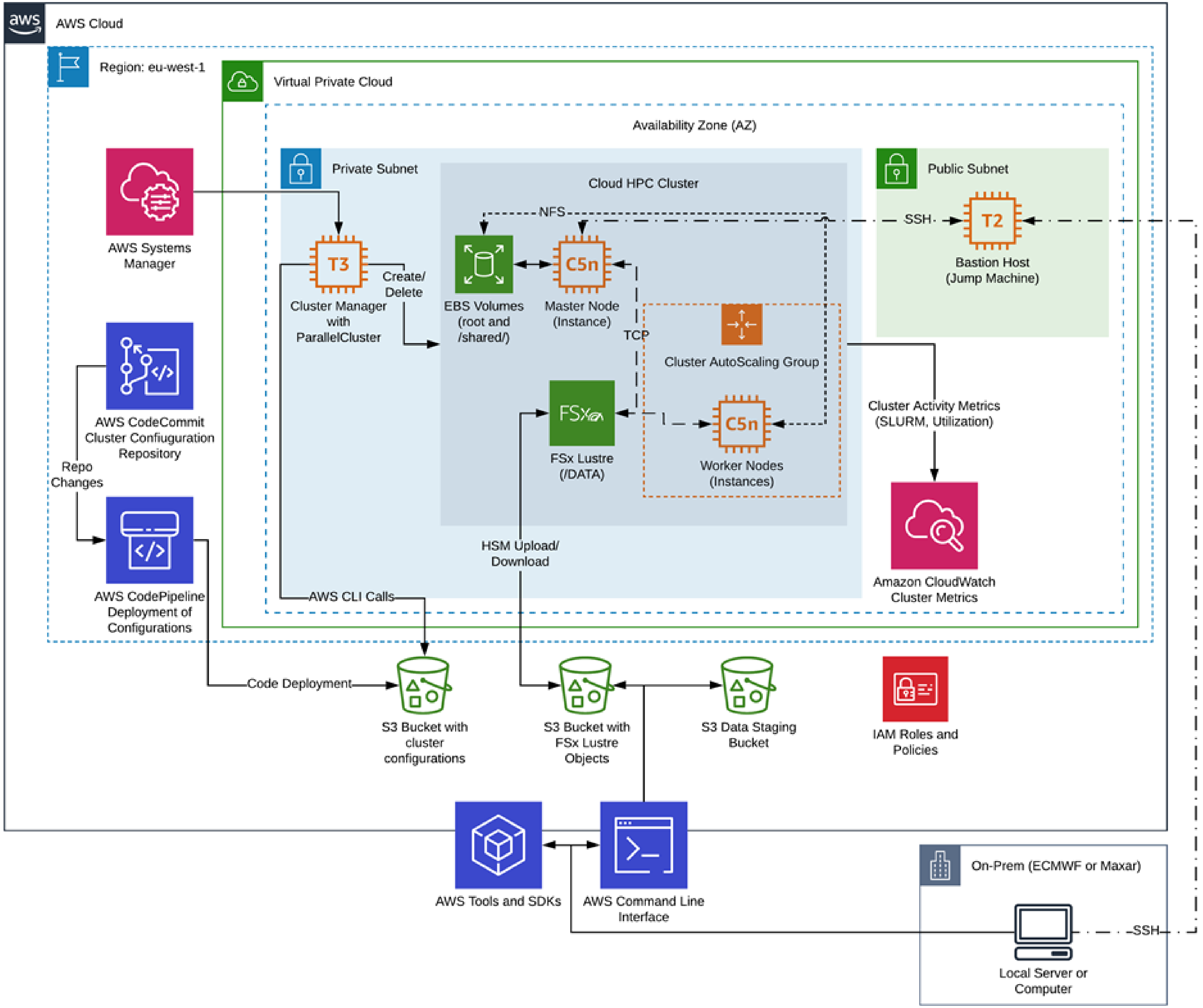
PHASE	RESOLUTION	NEMO	I/O(FDB)	COMPUTE NODES	I/O SERVER NODES	OMP	FORECAST LENGTH
1	TCo399L137	No	No	2		1,2,3,6,9,18	1-day and 5-day
2	TCo399L137	Yes	No	2,6		1,2,3,6,9,18	1-day and 5-day
3	TCo639L137	Yes	Yes	18,26	1, 2	6	5-day
4	TCo1279L137	Yes	Yes	28,33,38	1, 2, 3, 4	6	3-day

- In all of the above, we'll compare the results from AWS with twin/triplet runs on the Cray XC40 Broadwell system and the new ATOS BullSequana XH2000 system (knowing that this new system is still undergoing testing)

- In all our AWS simulations, we never use hyperthreading. As such, we never have overcommitted resources/threads. This is consistent with our prior work on AWS.

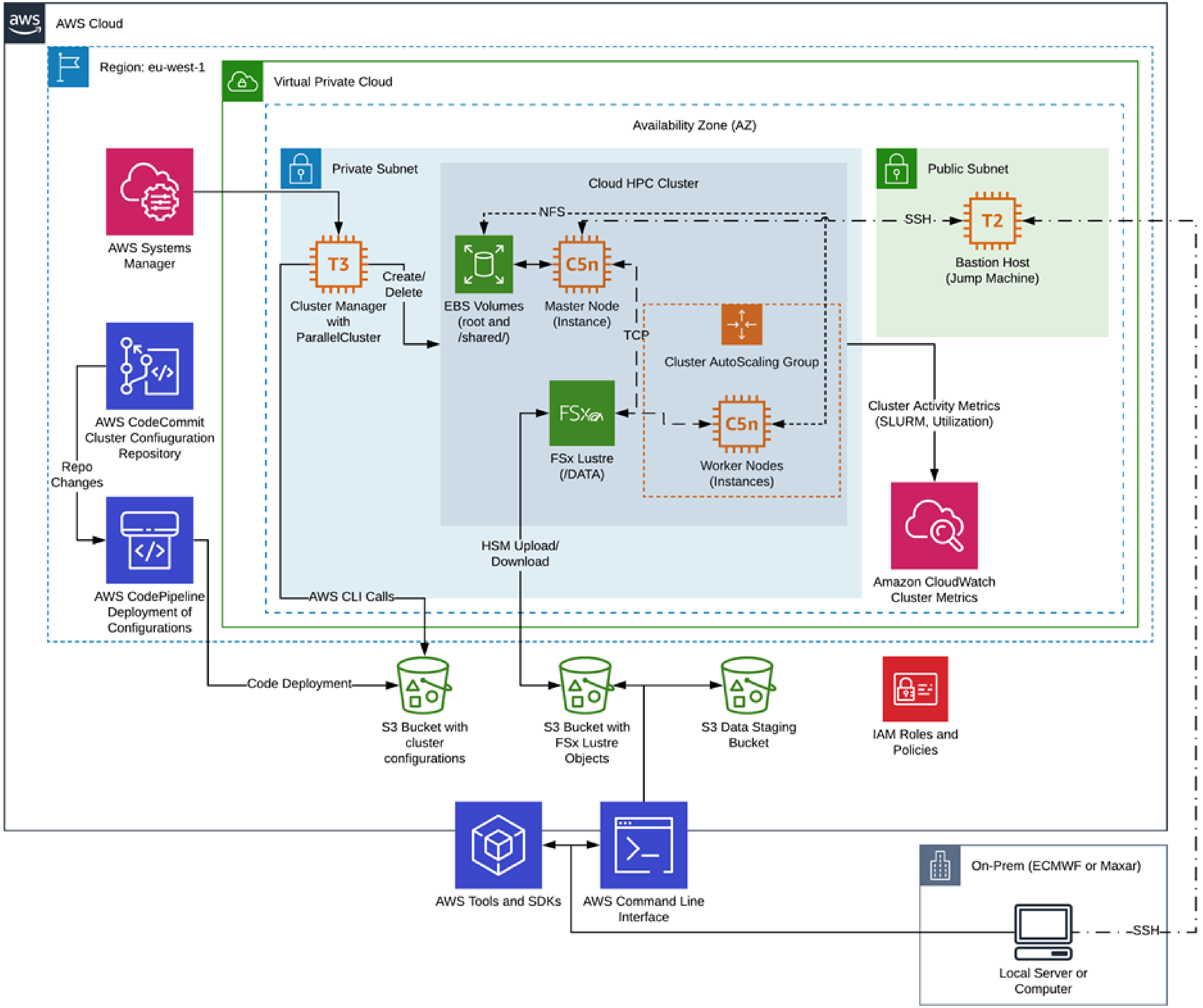
Cloud Environment

- Cloud HPC is a process in whicl one starts from nothing and mus build a cluster: obtain all the pieces, put them together, and fi the software stack.
- AWS ParallelCluster is a cloud HPC management tool created by AWS to configure, build, and manage cloud HPC environments within the AWS cloud ecosystem.
- There are several AWS-related configurations that need to mad before installing ParallelCluster and using it to spin up a cloud HPC cluster.



Cloud Environment

- Ensure AWS account limitations are set properly.
- AWS virtual private cloud (VPC) with subnets large enough to allocate all IPs associated with the cluster.
- IAM privileges (roles, policies, etc.) are properly set so that all AWS sources used by ParallelCluster.
- An EC2 instance (or local machine) that has Python3 (preferred) with the ability to access AWS resources.

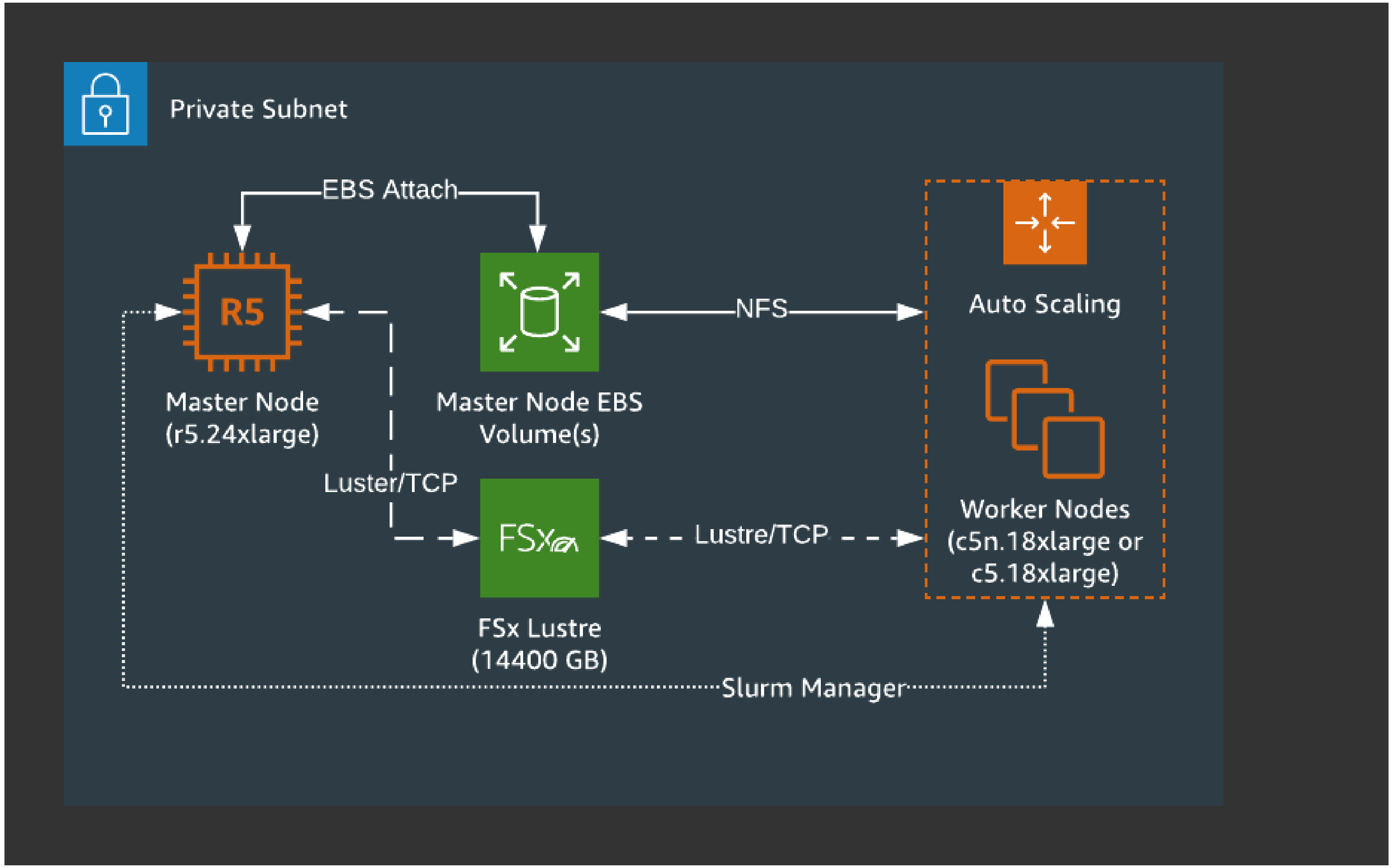


A Cloud HPC Environment for RAPS

- AWS ParallelCluster software orchestrates the spin-up/spin-down of cluster resources.

- Parallelized File System: FSx for Lustre
  - Size: 14,400GB (14.4TB)
  - Number of OSTs: 12
  - Progressive file layout:
    < 32MB = 1 OST, 1MB SIZE;
    < 256MB = 4 OST, 4MB SIZE;
    < 1GB = 8 OST, 8MB SIZE;
    >1GB = ALL OST, 16MB SIZE

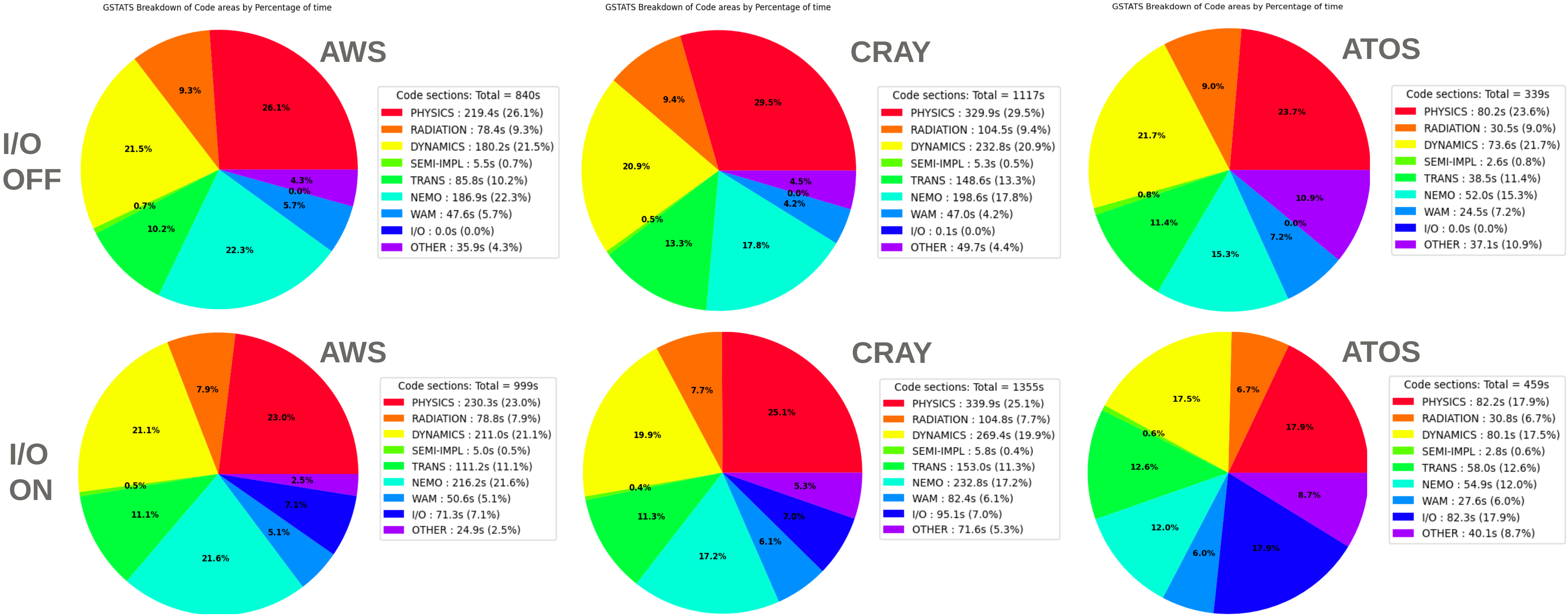
- EFA Network Interconnect
  - Throughput: 100Gbps
  - Integration: Using libfabrics as a part of Intel MPI.



Hardware and Related Configurations

	AWS	Cray XC40	Atos
System names	Eu-west-1	cca & ccb	AA
Compute nodes	42	~ 3500 x 2	1872
CPU-model	Intel Skylake	Intel Broadwell	AMD Epyc
	Xeon Platinum 8124M	Xeon E5-2695 v4	7742 (Rome)
Clock (GHz)	**3.0**	**2.1**	**2.2**
Memory/node (GiB)	192	128	256
Number of sockets	2	2	2
Number of NUMA-regions	2	2	8
Number of cores/node	**36**	**36**	**128**
Hyperthreads per core	1	2	2
Parallel filesystem	Lustre	Lustre	Lustre
Interconnect	EFA	Cray Aries	Mellanox HDR200
Linux O/S	Amazon Linux 2	CLE (Suse 11)	RHEL 8.3
Batch system	SLURM 19.05.5	PBSPro 13.0.412	SLURM 19.05.7-Bull.1.1
Compiler	Intel 2020u2	Cray CCE 8.7.7	Intel 2020u2
BLAS+FFTW	Intel MKL 2019u5	Intel MKL 2019u5	Intel MKL 2019u5
Message passing	Intel MPI 2019u8	Cray MPICH	OpenMPI HPCX 2.8.1
OpenMP	4.5	4.5	4.5

Phase 3 – how enabling I/O impacts time to completion - TCo639L137

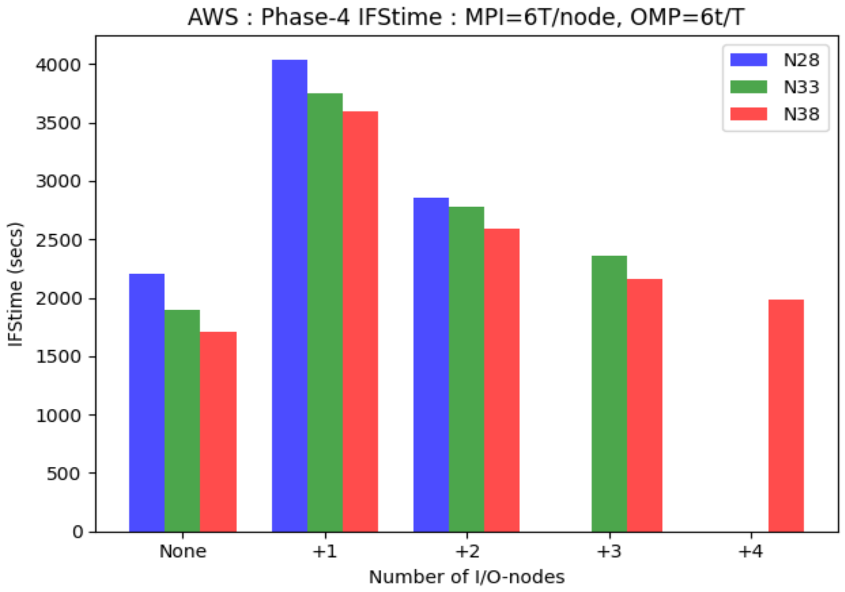


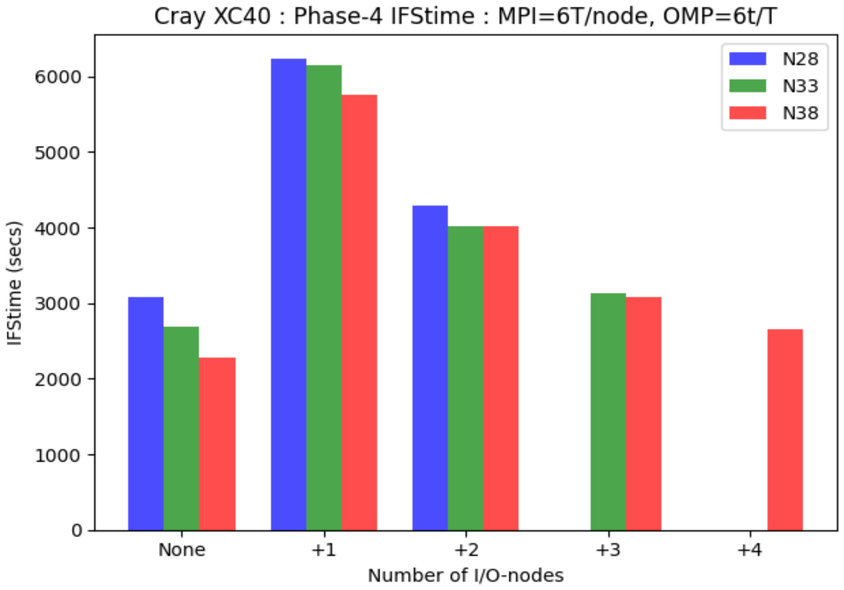
- Enabling I/O (adding 2 I/O nodes on to 28 compute nodes) results in a similar per cent increase in time to completion on AWS and CRAY, whereas ATOS being so fast with the compute, had a relatively larger per cent of time spent on I/O.

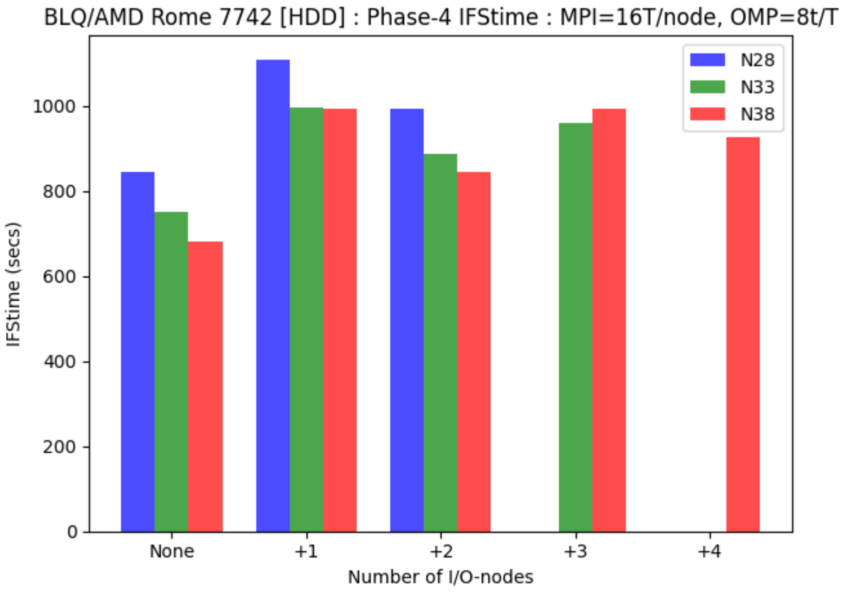
Phase 4 – Full resolution (TCo1279L137), deeper insights on scaling

- On AWS resources, as well as CRAY:
  - Increased numbers of compute cores decreased run time, as expected
  - Increased numbers of I/O (writer) nodes decreased run time.

- On ATOS
  - Times much faster overall, but addition of writer nodes not as impactful as seen on AWS and CRAY

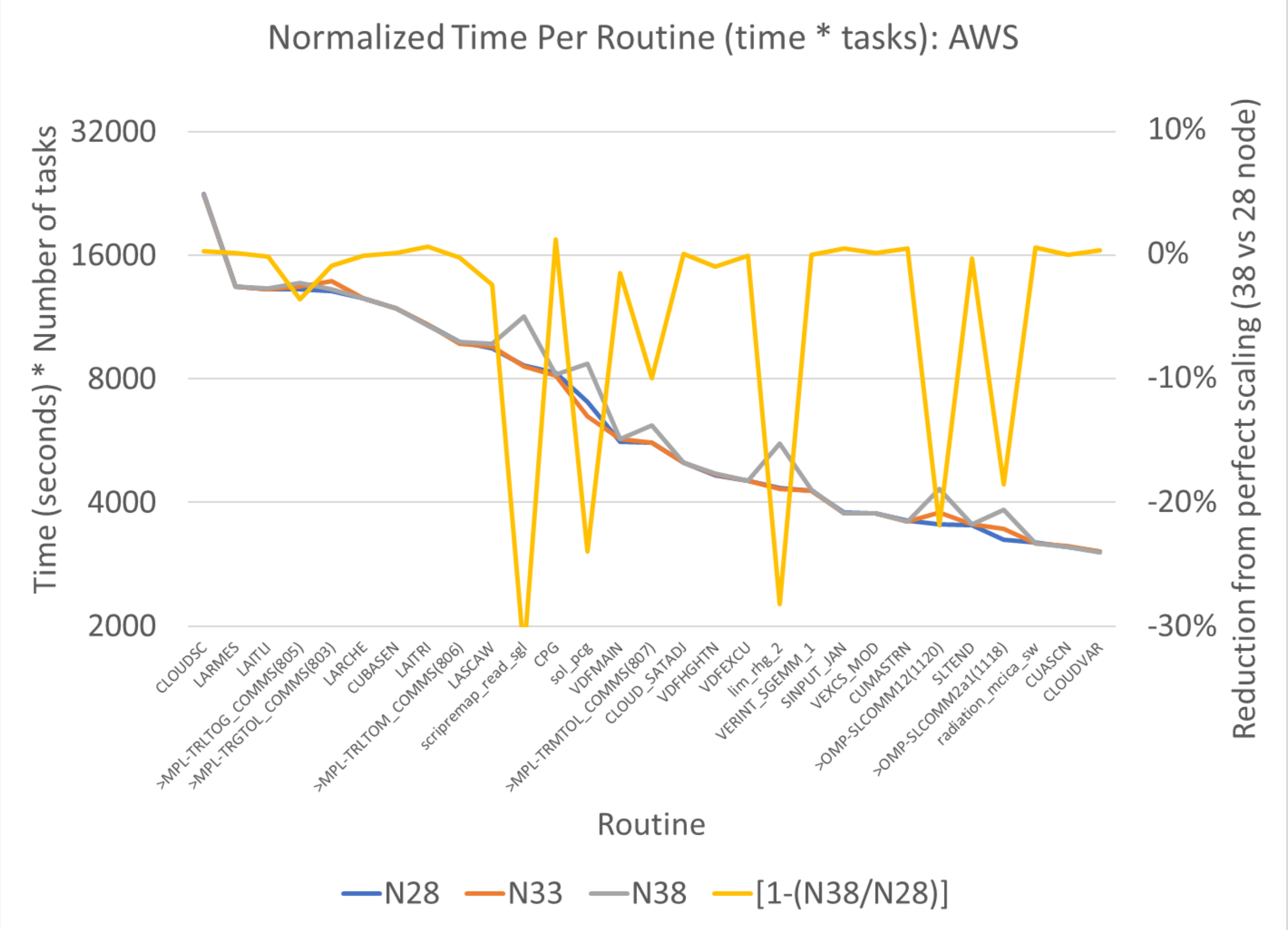






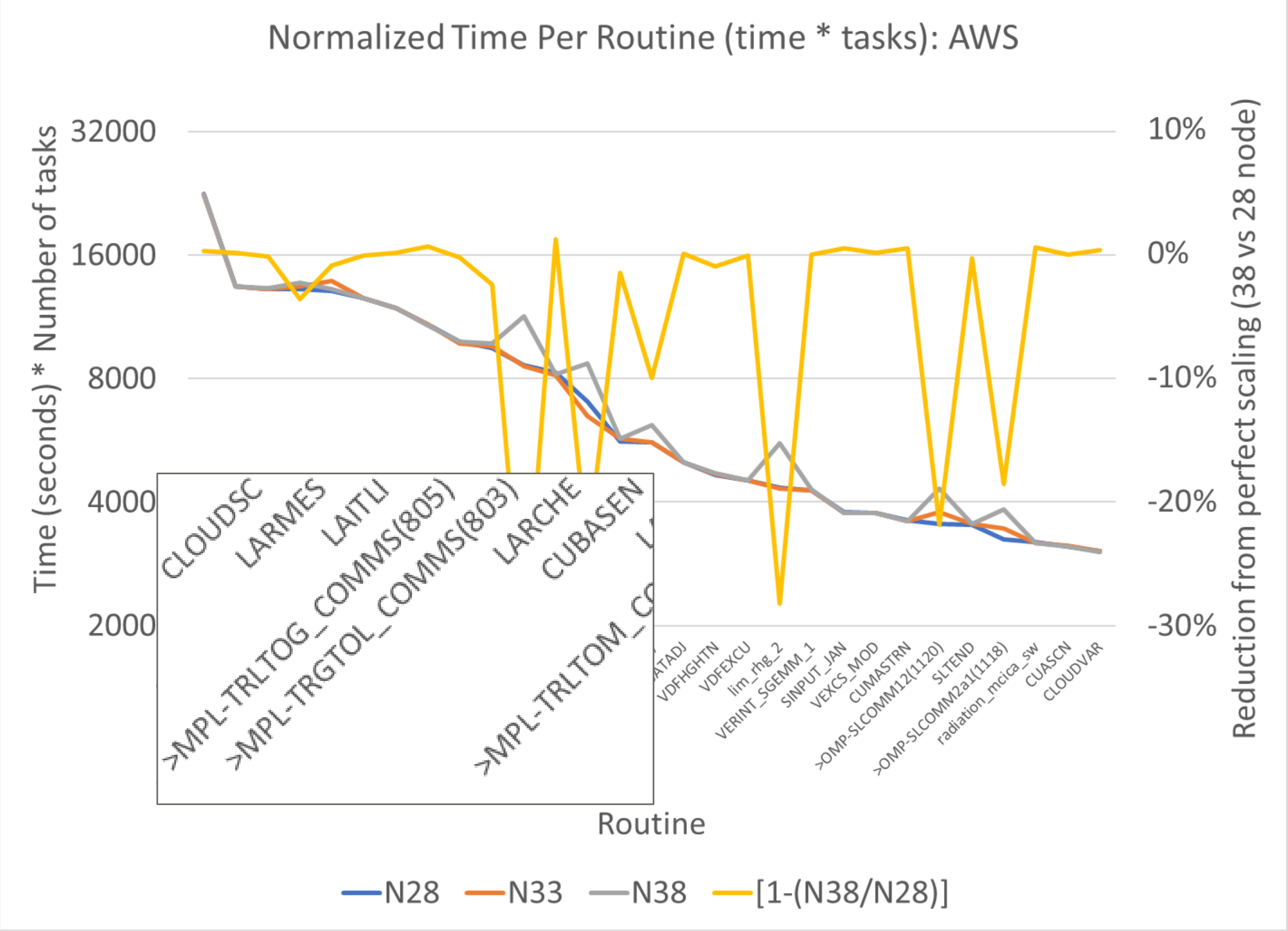
Phase 4 – Full resolution (TCo1279L137), deeper insights on scaling

- Plot shows (time per routine) * (number of MPI tasks)
- Perfect scaling – values should be the same.
- For the 3 routines that take the most time, this is the case
  - CLOUDSC
  - LARMES
  - LAITLI
- For some routines, bad scaling:
  - Scripremap_read_sgi
  - Sol_pcg
  - Lim_rhg_2
  - (involve reading of files by 1 node, all-to-allv)



Phase 4 – Full resolution (TCo1279L137), deeper insights on scaling

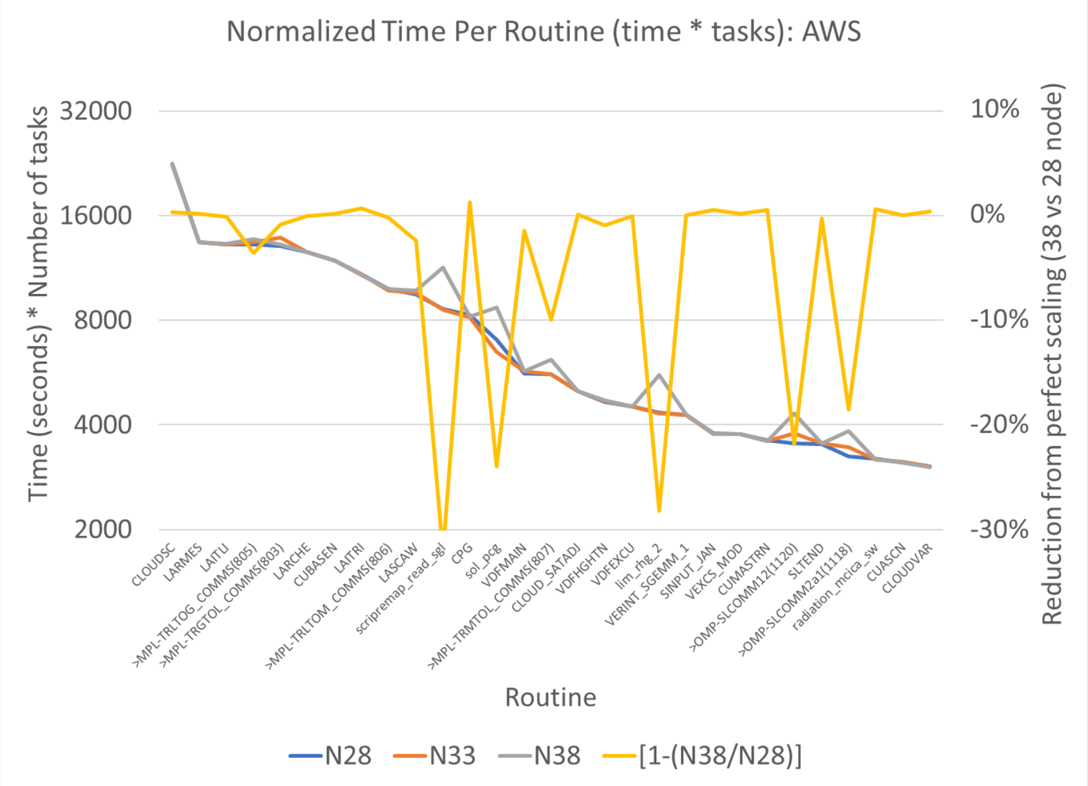
- Plot shows (time per routine) * (number of MPI tasks)
- Perfect scaling – values should be the same.
- For the 3 routines that take the most time, this is the case
  - CLOUDSC
  - LARMES
  - LAITLI
- For some routines, bad scaling:
  - Scripremap_read_sgi
  - Sol_pcg
  - Lim_rhg_2
  - (involve reading of files by 1 node, all-to-allv)

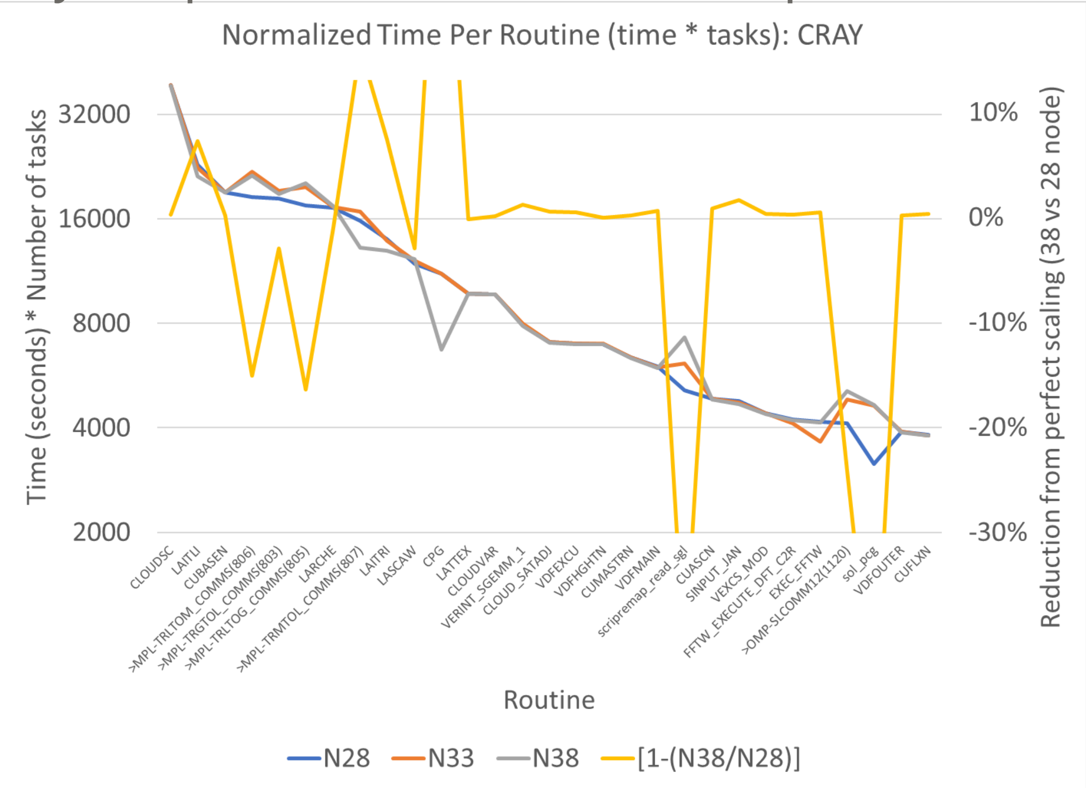


MAXAR

Phase 4 – Full resolution (TCo1279L137), deeper insights on scaling

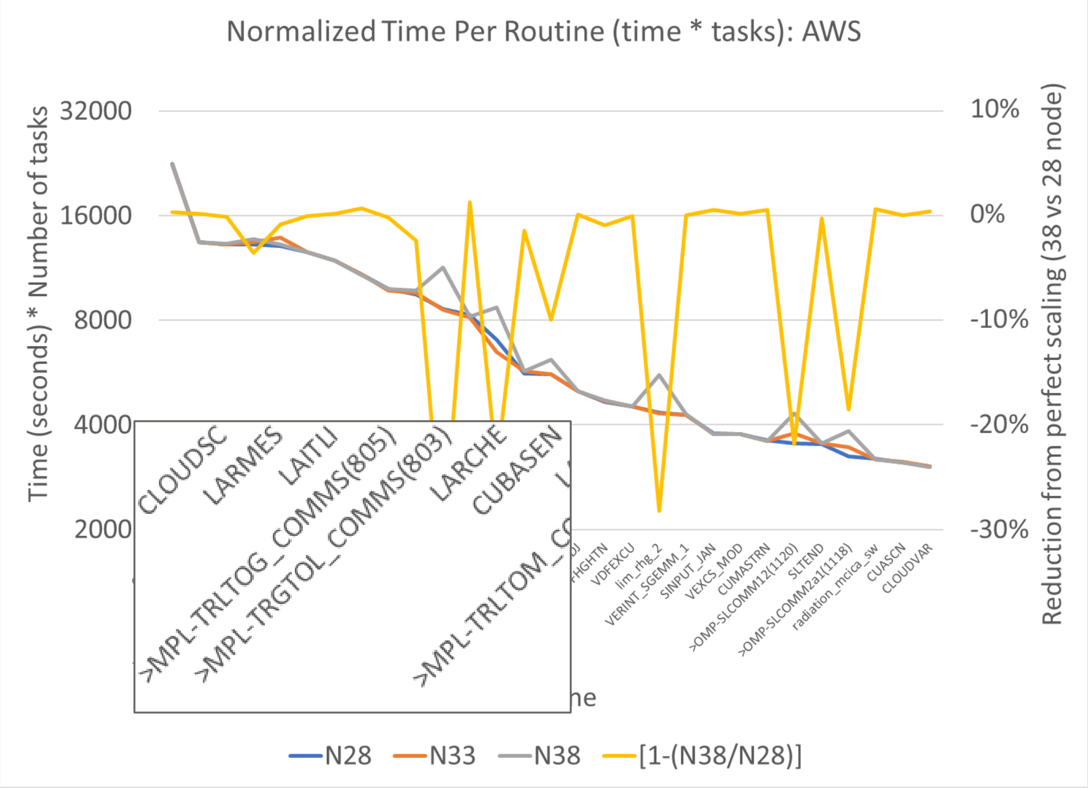
- Comparing AWS results to CRAY:
  - Top routines are the same.
  - Scripremap_read_sgi and sol_pkg do not scale well on either system (expected).
  - MPI related routines seem a little better on AWS than CRAY.
  - CRAY has some super-scaling routines, likely the product of the CRAY compiler.

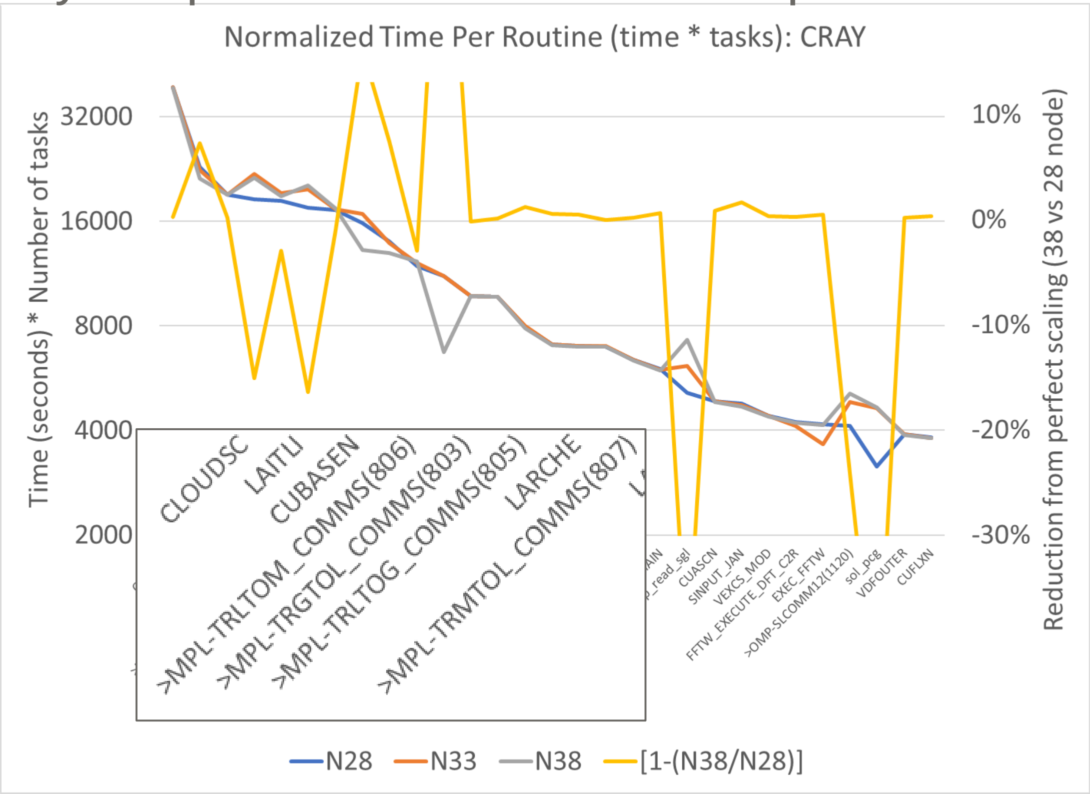




Phase 4 – Full resolution (TCo1279L137), deeper insights on scaling

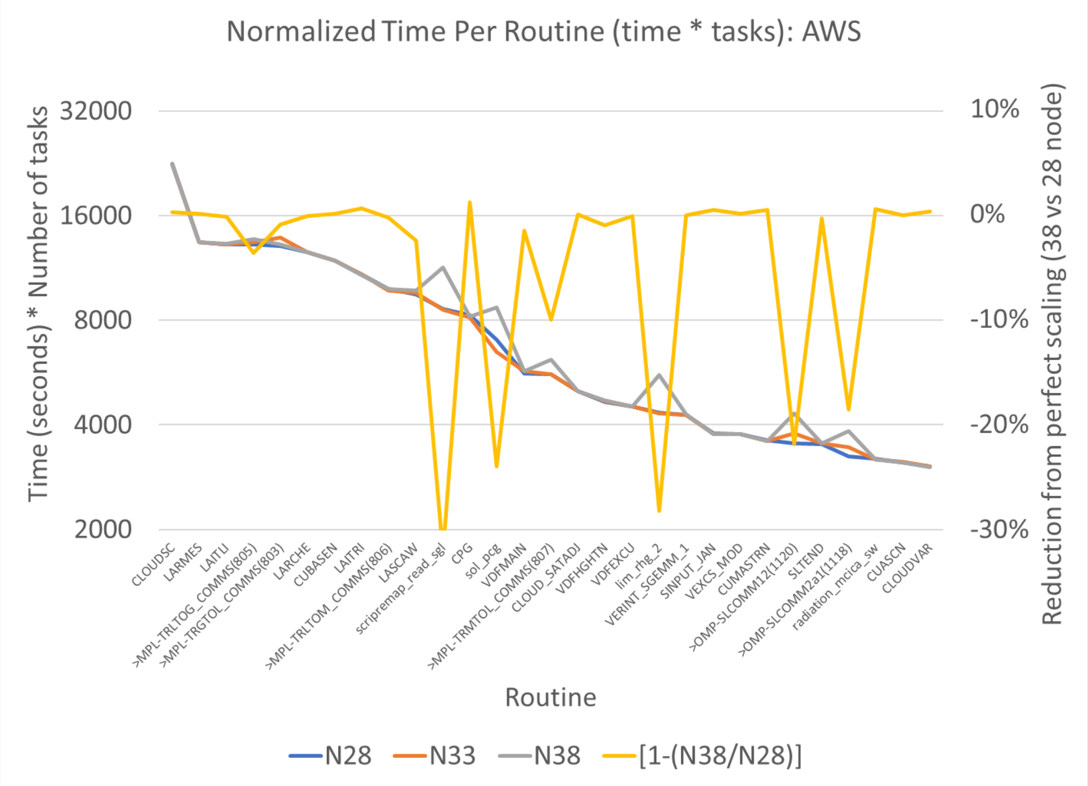
- Comparing AWS results to CRAY:
  - Top routines are the same.
  - Scripremap_read_sgi and sol_pkg do not scale well on either system (expected).
  - MPI related routines seem a little better on AWS than CRAY.
  - CRAY has some super-scaling routines, likely the product of the CRAY compiler.

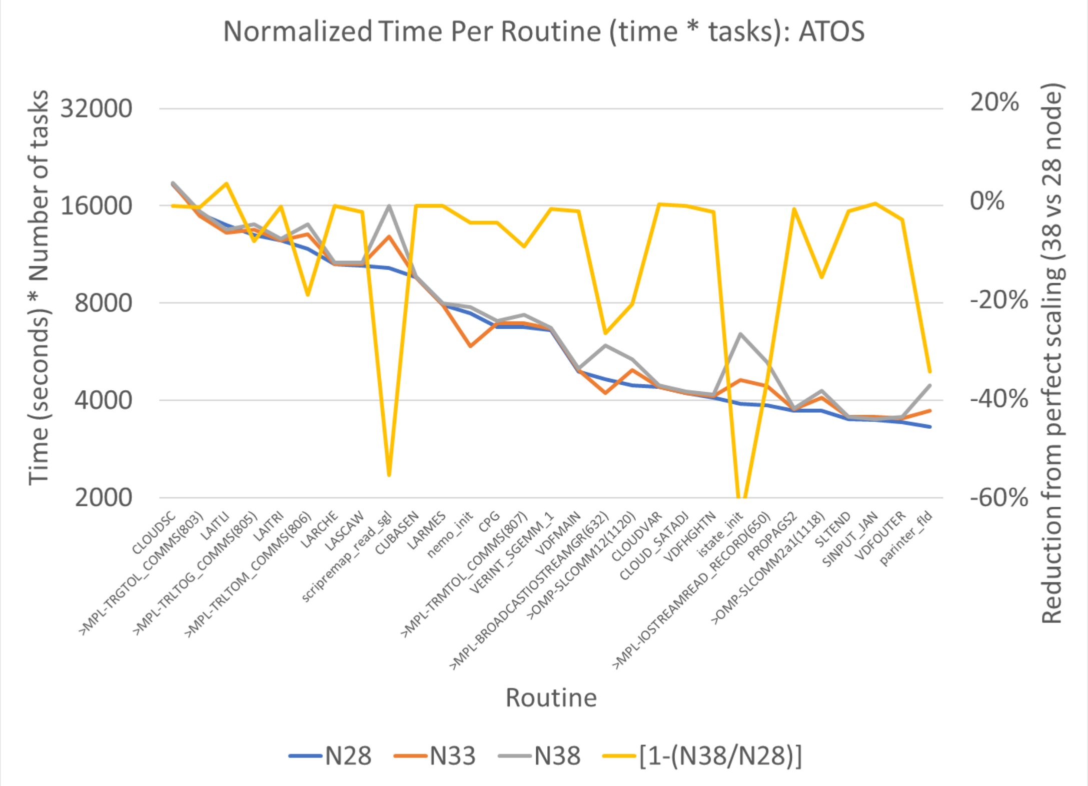




Phase 4 – Full resolution (TCo1279L137), deeper insights on scaling

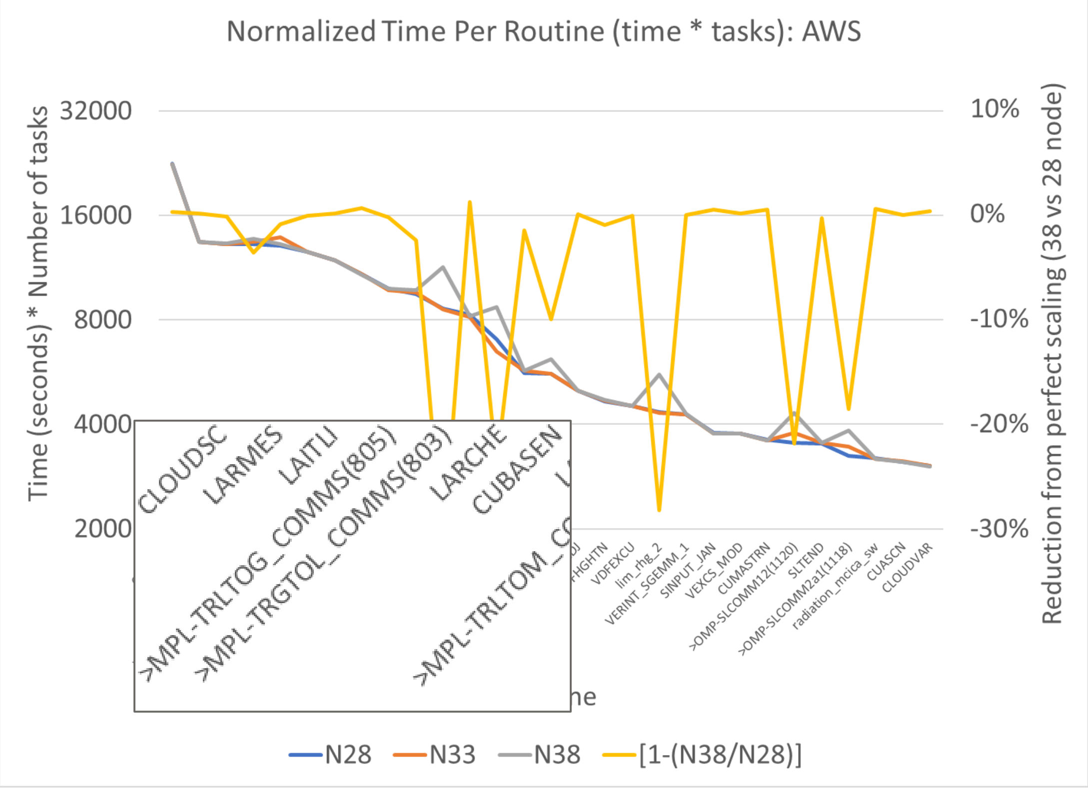
- Comparing AWS results to ATOS:
  - Top routines are not the same – but ATOS still undergoing testing.
  - When the compute is sufficiently fast, other routines (MPI,COMM) stand out.
  - Scripremap_read_sgi and sol_pkg do not scale well on either system.
  - Some routines that were vectorized on one system were not on another.

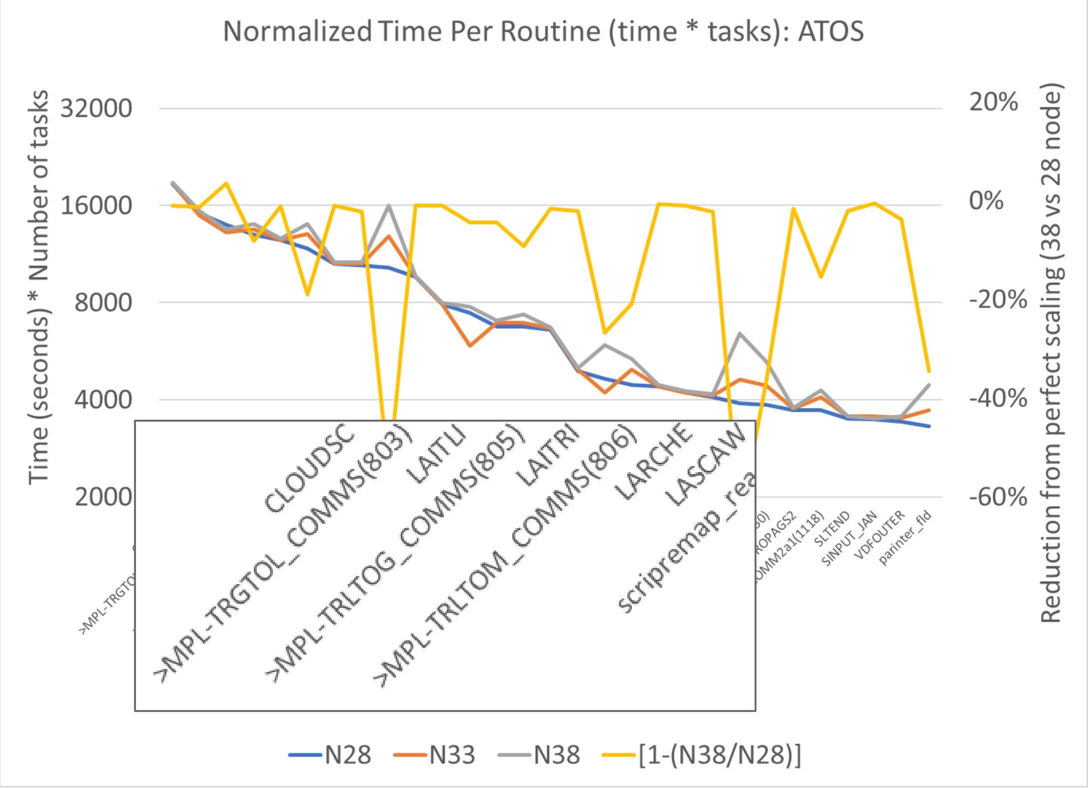




Phase 4 – Full resolution (TCo1279L137), deeper insights on scaling

- Comparing AWS results to ATOS:
  - Top routines are not the same – but ATOS still undergoing testing.
  - When the compute is sufficiently fast, other routines (MPI,COMM) stand out.
  - Scripremap_read_sgi and sol_pkg do not scale well on either system.
  - Some routines that were vectorized on one system were not on another.





The costs of AWS: hourly rates for the instance, the FSx Lustre file system, and the EBS volumes provisioned

AWS Component	Instance Type or Total Storage Used by Service	Hourly Rate in AWS Ireland Region (USD)	Additional Notes
EC2 (Compute):	c5n.18xlarge	On-demand: $4.392 Spot market: variable (% savings off on-demand pricing)	Hourly rate listed here: https://aws.amazon.com/ec2/pricing/on-demand/
FSx (Storage):	14TB	$0.000213 per GB $3.06 per 14TB file system	Hourly rate per GB calculated from monthly rates listed here: https://aws.amazon.com/fsx/lustre/pricing
EBS (Storage):	6.5TB	$0.000153 per GB $0.993056 per 6.5TB of EBS storage used	Hourly rate per GB calculated from monthly rates listed here: https://aws.amazon.com/ebs/pricing/

- Time provisioned is the duration for which a component is ‘on’ - whether or not it is doing anything.

The costs of AWS: a 3-day and a 10-day run of TCo1279L137

	tco1279l137 38+4 IFS Run Time (minutes)	Compute Instance Type	Number of Compute Instances	Compute Cost Per Simulation	Master Instance Type	Master Instance Cost Per Simulation	FSx Storage Cost Per Simulation	EBS Storage Costs Per Simulation	30-minute Spin-up Costs (all resources combined)	Total Estimated Cost Per Simulation without Spin-up	Total Estimated Cost Per Simulation with Spin-up
3-day Forecast Standard On-Demand Pricing (USD)	~35	c5n.18xlarge (36 cores per instance)	42	$107.60	c5n. 18xlarge	$2.56	$1.79	$0.58	$96.45	$112.53	**$208.98**
10-day Forecast Standard On-Demand Pricing (USD)	~95	c5n.18xlarge (36 cores per instance)	42	$292.07	c5n. 18xlarge	$6.95	$4.85	$1.57	$96.45	$305.44	**$401.89**

- In both cases, the compute cost ($107.60 out of $112.53, or $292.07 out of $305.44) is approximately 96% of the total cost of a simulation.
- After completion, the cost shifts from compute to storage.

AWS costs: storing the static RAPS files in S3 between cloud HPC

	Monthly Rate per GB (USD)	Cost to Store 275GB of Static RAPS Files Monthly (USD)	Time for Egress	Retrieval cost per GB(USD)
Standard S3:	$0.023	$6.32	Online/instant	None
Infrequent Access S3:	$0.0125	$3.43	Online/instant	$0.01
Glacier S3:	$0.004	$1.10	1-5 minutes 3-5 minutes 5-12 hours	$0.03 $0.01 $0.0025
Deep Glacier S3:	$0.00099	$0.27	Up to 12 hours Up to 48 hours	$0.02 $0.003

- Ingress to S3 AWS: $0.00/GB Egress from S3 within region: $0.00/GB
  Egress from S3 to other regions: $0.02/GB
  Egress from S3 outside AWS: $0.09/GB
  (as low as $0.05/GB after 150TB/month)
- Additional details at: https://aws.amazon.com/s3/pricing/

MAXAR

Overall Scope of Project

- The goals of the work are the following:

- Phase 1 – can IFS compile and run in the cloud at all? What is needed to achieve this?
  - Yes, IFS, NEMO, WAM, (all of RAPS) compiled and ran in the cloud.

- Phase 2 – can IFS, coupled with NEMO, run in the cloud? How does it scale?
  - The performance, relative to CRAY, was about equal to the difference in clock speeds between the CRAY Broadwell processors and the AWS Skylake processors – no discernable performance drop resulting from interconnect or non-vectorization of the code.

- Phase 3 – when increasing the resolution some, and enabling I/O, how well does the cloud perform? Does it scale well?
  - IFS scales well (results not shown), for the most part increased node counts result in reduced time spent on the significant routines within IFS by an expected amount.

Overall Scope of Project

- The goals of the work are the following:

– Phase 4 – getting as close to an operational configuration as possible, how does the cloud perform? Does it scale well? How is the strong scaling?
  – Overall run time did decrease as expected when the number of compute nodes increased.
  – The I/O also saw improved performance as writer nodes were added, as expected
  – The strong scaling results indicate that the most time-consuming routines scaled well with increase in node count, and that rank of the most time-consuming routines on AWS was a good match to the ranking on the CRAY and ATOS systems

- In summary – the goals of the proposed work were achieved and have shown that cloud computing is an option for all the components of the RAPS system: WAM, NEMO, and IFS.

Full details at: https://confluence.ecmwf.int/display/MAX/Maxar+Home

MAXAR.COM

Overall Scope of Project

- The systems in this work are:
  - The ‘high-resolution forecast’ configuration of the Integrated Forecast System (IFS, CY45R1+)
  - The Nucleus for European Modelling of the Ocean (NEMO)
  - The wave model (WAM)
- Thus, this is what is contained in RAPS – and is not the current operational suite of models.
- While some of the runs above did write output to disk, that output was never read back from disk. There was no product generation. As such, the file system was not taxed.
- Only single model run experiments were performed – we did not attempt to run a forecast ensemble.
- There was no servicing of data beyond the runtime of the run.