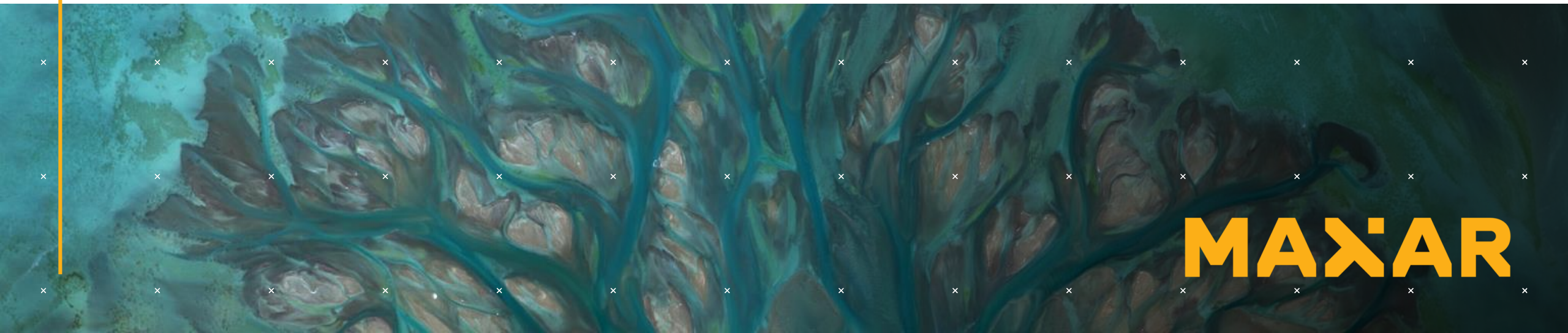
IFS (RAPS) on AWS – Successes and Challenges

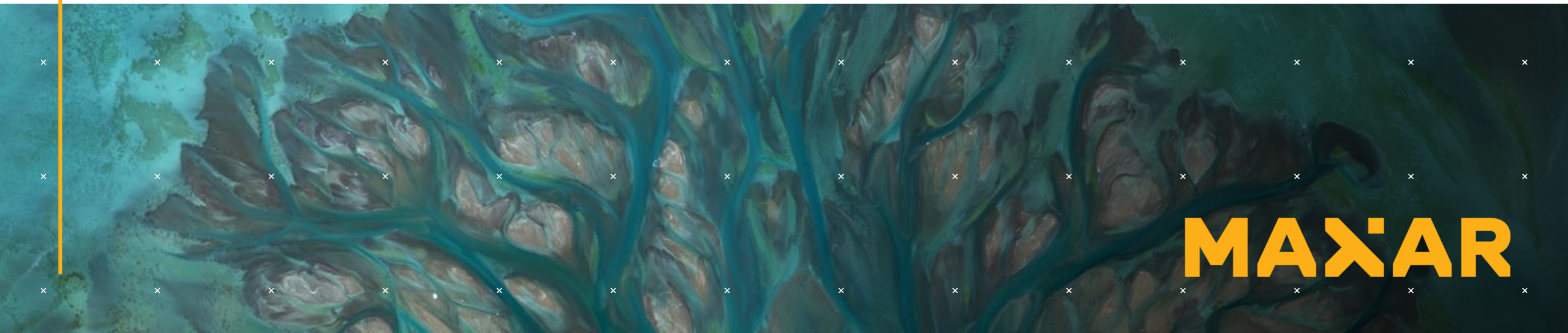
9 February 2021

Brian Etherton – Numerical Weather Prediction Scientist, Maxar

With contributions from:
Stefan Cecelski, Christopher Cassidy, and Mehzabeen Hoosein - Maxar
Sami Saarinen, Iain Miller, and Umberto Modigliani - ECMWF



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



HPC in the cloud – personal car versus a cab

- Cloud computing resources are not the same as on premises resources.
- An on-premises cluster is akin to owning your own car: something to care for (change the oil, tyres, etc), continually provide for (petrol, etc.), have a place for it (garage), and you have for a lifetime.
- Cloud computing is akin to a cab (or perhaps rental car): you go get them when you need them, and when you are done with them, you move on.
- When thinking of HPC in the cloud – how would you use a cab versus a personal car?

Origin story: Oldest known picture with Etherton and Saarinen (2016)

In a twitter ‘conversation’, it was realized by ECMWF personnel that Maxar had the capability to run a numerical weather prediction model on cloud computing resources

Overall Scope of Project

- After some back and forth, there was a signed contract between ECMWF and Maxar to run some of 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 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 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	3	5-day
4 (TBC)	TCo1279L137	Yes	Yes	28,33,38	1, 2, 3, 4	3	3-day

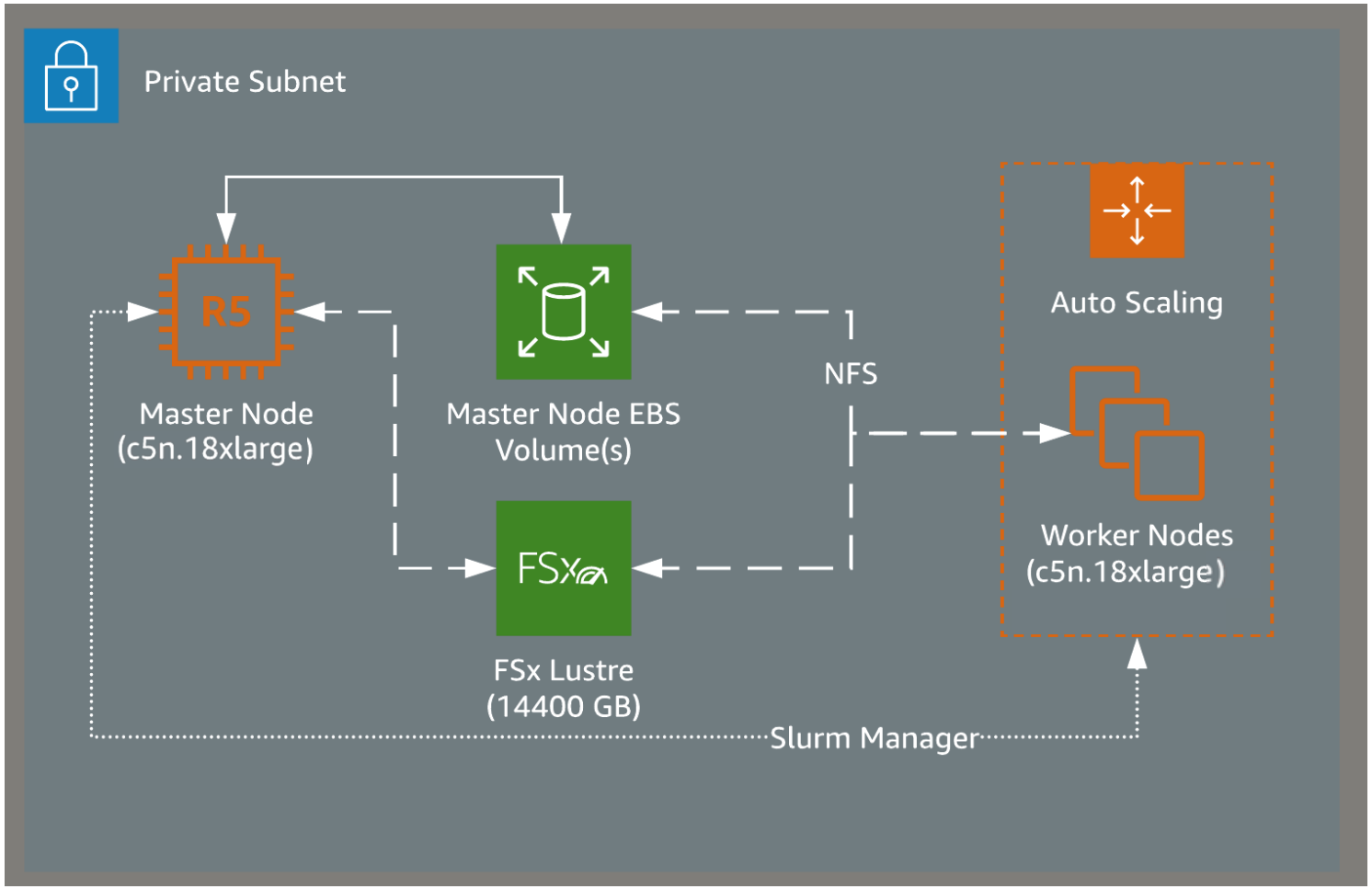
- In all of the above, we'll compare the results from AWS with twin runs on the Cray XC40 Broadwell system.

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
System names	Eu-west-1	cca & ccb
Compute nodes	42	~ 3500 x 2
CPU-model	Intel Skylake	Intel Broadwell
	Xeon Platinum 8124M	Xeon E5-2695 v4
Clock (GHz)	3.0	2.1
Memory/node (GiB)	192	128
Number of sockets	2	2
Number of NUMA-regions	2	2
Number of cores/node	36	36
Hyperthreads per core	1	2
Parallel filesystem	Lustre	Lustre
Interconnect	EFA	Cray Aries
Linux O/S	Amazon Linux 2	CLE (Suse 11)
Batch system	SLURM 19.05.5	PBSPro 13.0.412
Compiler	Intel 2020u2	Cray CCE 8.7.7
BLAS+FFTW	Intel MKL 2019u5	Intel MKL 2019u5
Message passing	Intel MPI 2019u8	Cray MPICH
OpenMP	4.5	4.5

Phase 1 – demonstrate that IFS (uncoupled) can run on AWS

- Access AWS resources:
  - Account is setup with a 2-tier network environment (public/private) to provide increased security
- Get the data onto AWS resources
  - aws s3 cp --recursive tarballs -- at **average speed over 60MBytes/s**
- A fair number of early tests were done on AWS, in advance of the formal first runs. Those included:
  - Baseline runs to verify reproducibility
  - Use of huge pages
  - Sample DrHook outputs
  - Verifying IO mode
  - Optimizing NPROMA
  - Checking on TBB-malloc
  - Sample ecProf outputs
- The first formal runs consisted of 2-Nodes, TCo399L137, no nemo, 5-days, small pages, NPROMA=32. A comparison of this run to one using huge pages:

FCdays	EstTime	IFStime	JOBtime	NodeHrs	PerTSTEP	MPIStart	Pages	VMPEAK	TOTMEM	MaxError
210	4118	2075	2088	1.16	4.286	4.748	Small	646G	106G	2.64362%
219	3948	1988	2000	1.11	4.113	4.098	Huge	657G	116G	2.64362%

- Results of this first pair of runs show that bit reproducibility was achieved, as the value of Maximum Error was the same for both runs. Bit reproducibility was the case for all simulations in phases 1-3 on AWS resources.

Phase 1 – demonstrate that IFS (uncoupled) can run on AWS: comparison to other hardware

- Early results comparing the 3 sets of hardware show differences in performance:

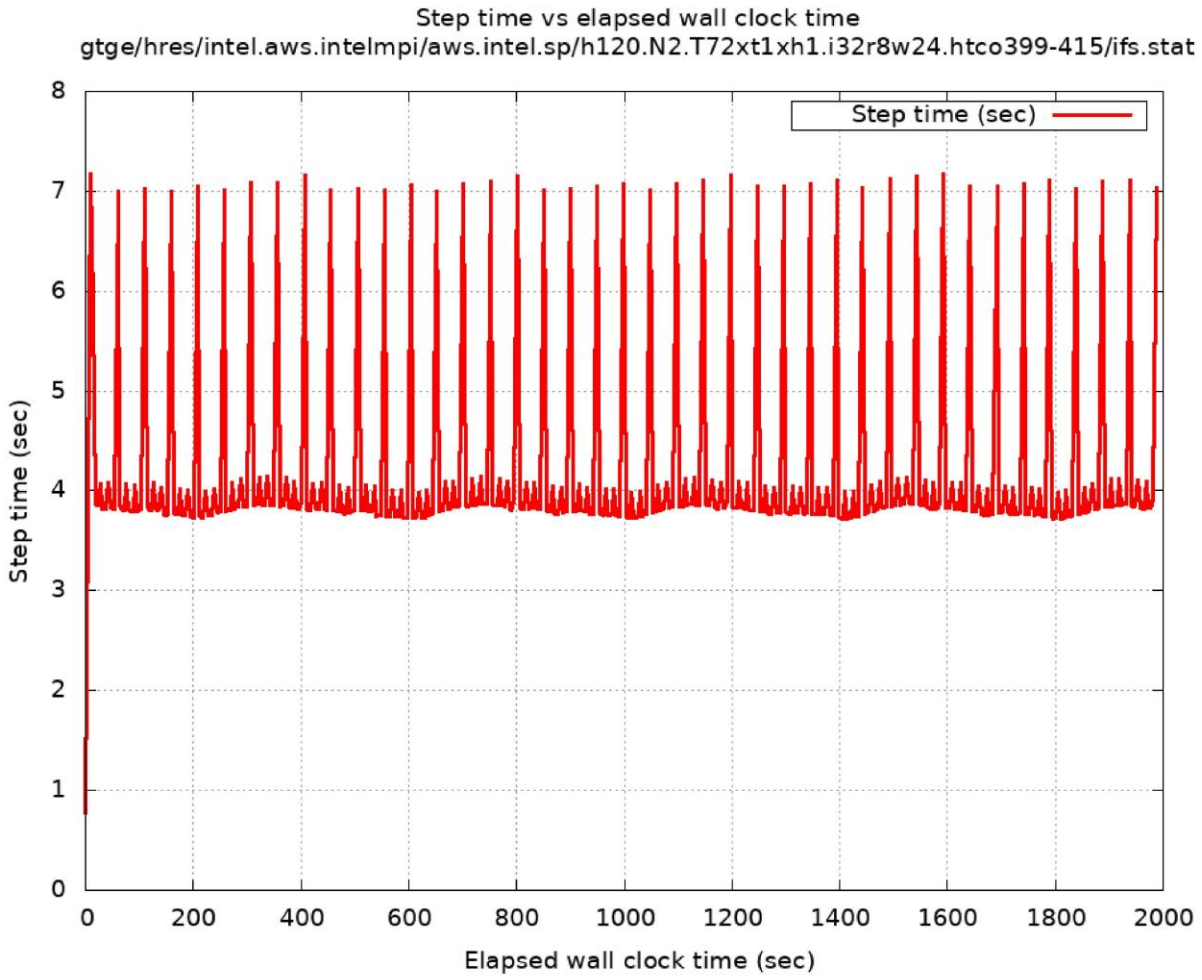
PROCESSOR/CORES PER NODE	SPEED	TIME	(RADTIME)	COMMENTS
AWS Skylake 36	3.0GHz	3.8s	(7.0s)	
Cray XC40 Broadwell 36	2.1GHz turbo	5.7s	(10.1s)	– 1.5X slower than AWS Skylake

- The run times are approximately as expected, the faster clock speeds of the Skylake resulting in less time spent per time step than the CRAY Broadwell.

- Running on cloud resources, which requires some overhead (like hypervisor) did not result in a significant performance degradation.

Phase 1 – issues brought up via diagnostics

- A plot of step time versus elapsed wall clock time shows a ‘jitter’ – the expectation was that the non-radiation time steps would all have the exact same duration
- This ‘jitter’ was not expected, it is not seen on other systems
- One hypothesis: the jitter is an artifact of the lack of hyperthreading – something that often absorbs and would smooth out these kind of small variations
- This jitter is less notable at higher node/core counts (phases 2 and 3)



Phase 2 – Demonstrate that IFS can run coupled to NEMO, and give first assessment of scalability

- The runs in which NEMO was coupled with IFS were successful

- We then took a look at scalability, and at the value of NPROMA. Values below:

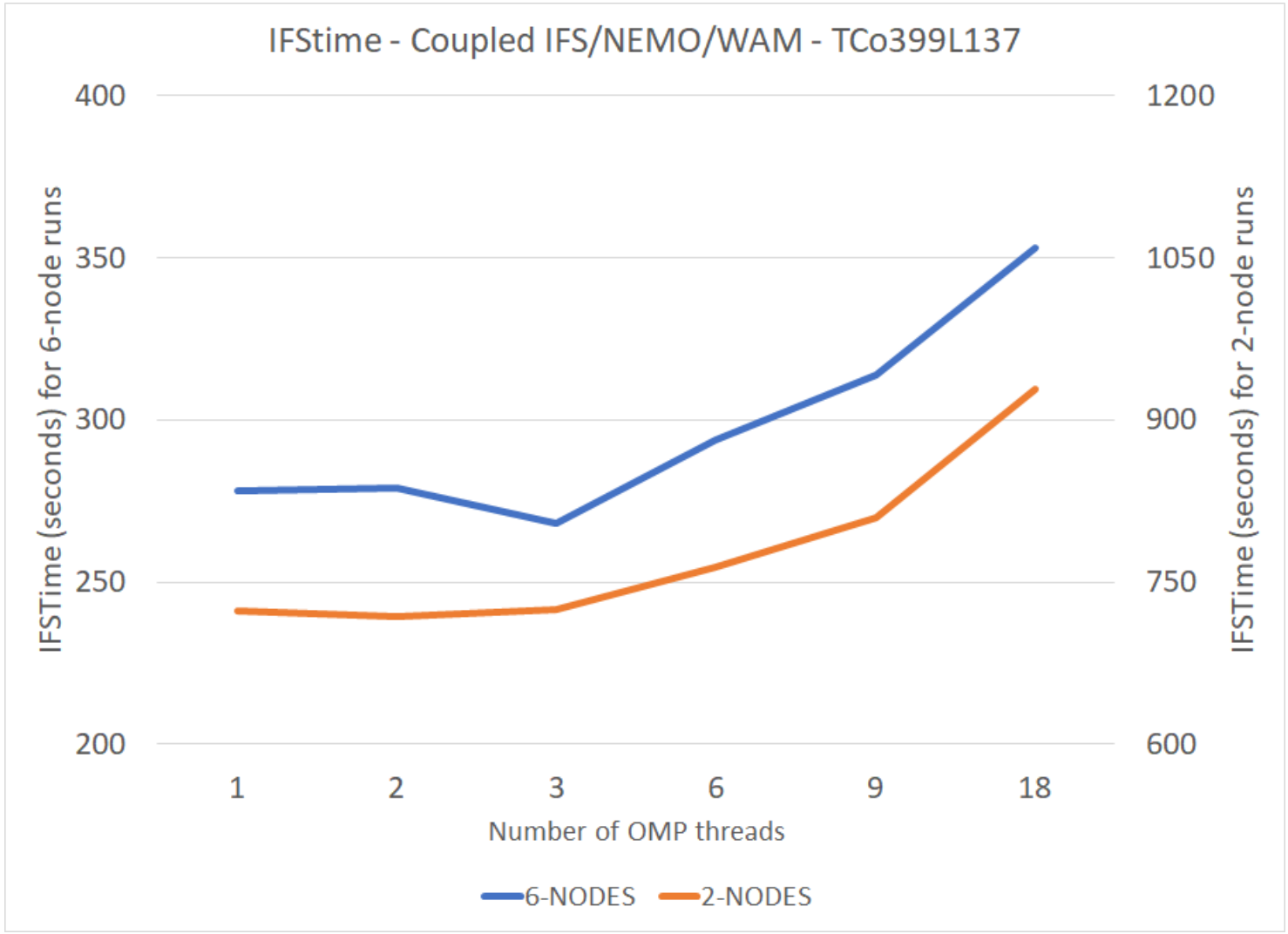
Nodes	FCdays	MPI	OMP	EstTime	IFStime	JOBtime	NodeHours	NPROMA	VMPEAK	TOTMEM	STACK	MaxError
6	375	216	1	2319	1179	1192	1.99	**24**	2160G	529G	168	1.39403%
6	378	216	1	2321	1186	1202	2.00	32	2160G	528G	168	1.39403%
2	133	72	1	6508	3292	3305	1.84	32	792G	266G	168	1.39403%

- One would expect that the 6-node simulation would take 33.3% (1/3) the time of the 2-node simulation, and the above shows about 38% (IFTime from 3292 seconds to 1186 seconds)

- The change of the NPROMA value did result in a rather slight change in time to completion (7 seconds out of 1186, which is less than 1%)

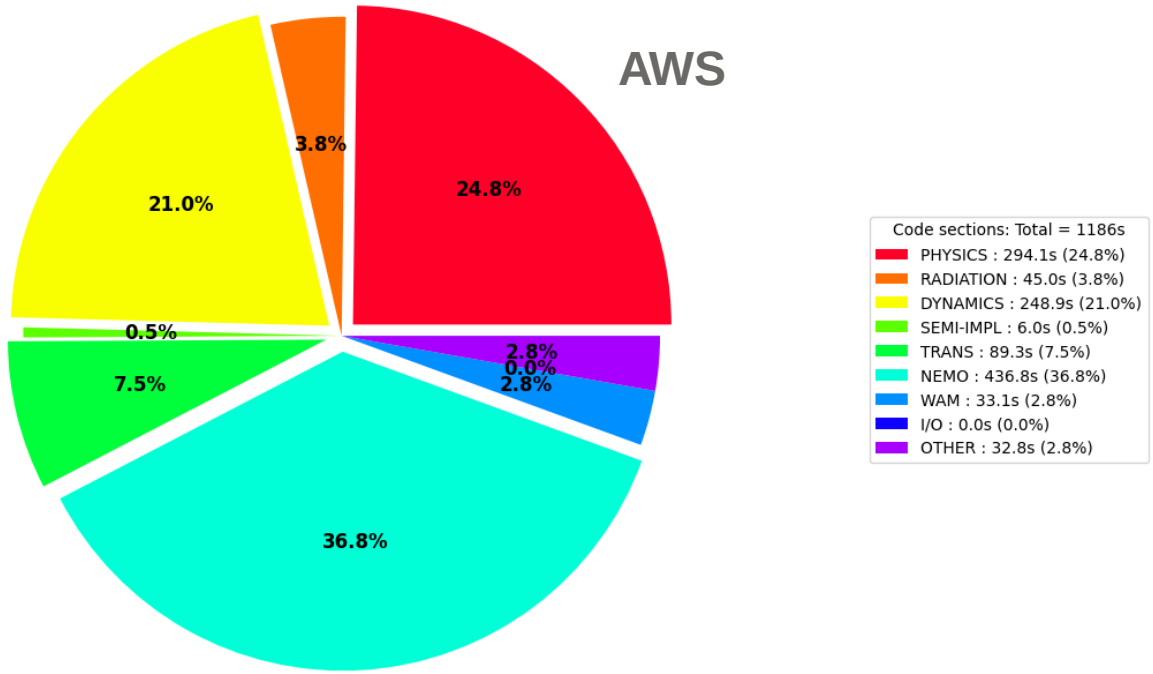
Phase 2 – impacts of changing the number of OMP threads

- In the relatively low-resolution simulations on the relatively low node counts, we did see performance differences resulting from varying the number of OMP threads used.
- In addition, for most of these thread counts, the speed reduction is close to the 1/3 one would expect from tripling the number of compute cores

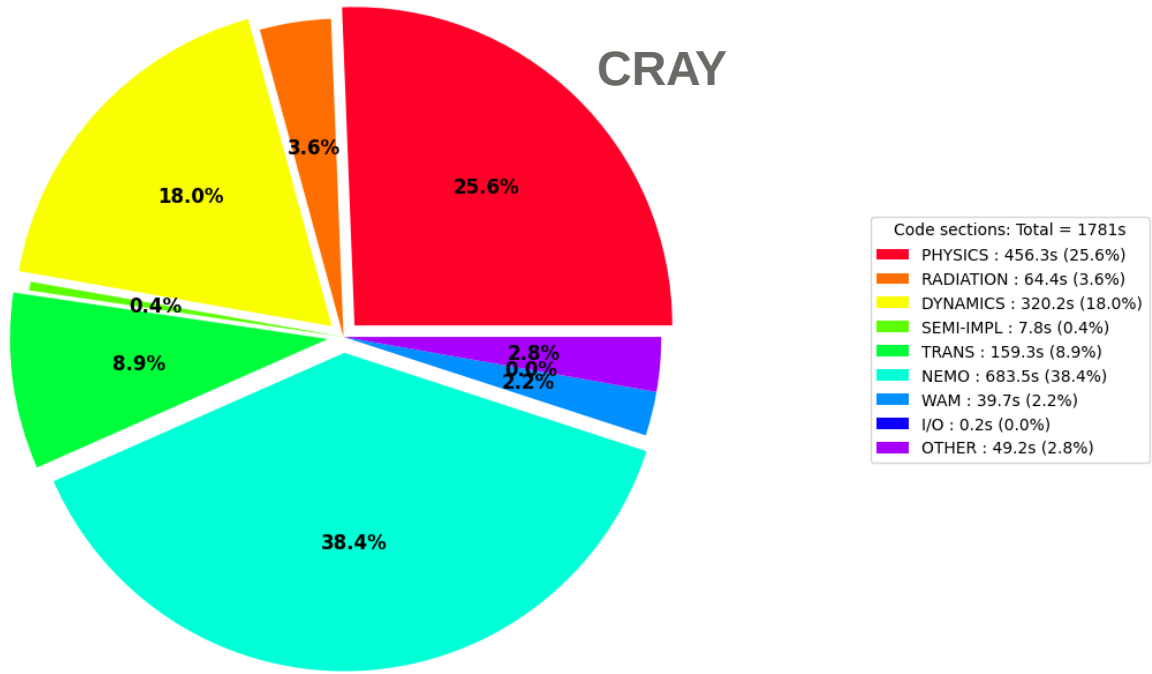


Phase 2 – insights from profiling

PHASE2/AWS [NPROMA=32]: h120.N6.T216xt1xh1.i32r8w24.ORCA025_Z75.htco399-423



2/Cray-XC40 [NPROMA=32]: h120.N6.T72xt3xh1.i32r8w24.ORCA025_Z75.htco399-6148305.bitrep

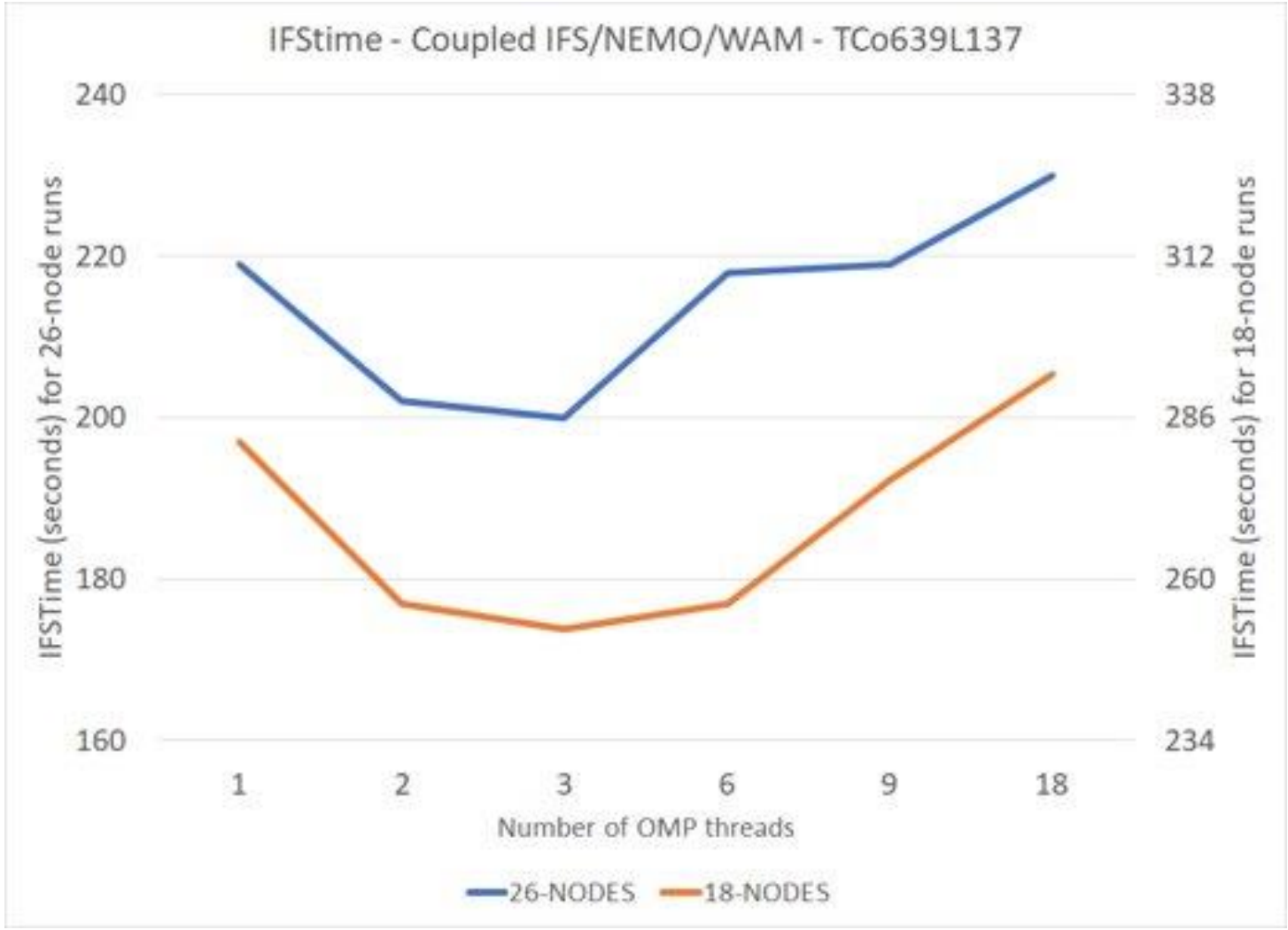


- A comparison of the 6-node, NPROMA=32, 120-hour simulations of the coupled IFS(TCo639L137)/NEMO/WAM simulations shows slight differences between the systems
  - AWS takes longer on dynamics than CRAY
  - CRAY takes longer on dynamics and NEMO

Phase 3 – run at higher resolution, and with full output enabled. Explore scalability further.

- When running the TCo639L137 resolution IFS, coupled to NEMO and WAM, the impacts of varying the number of OMP threads show a more notable improvement for using 3 OMP threads than was the case for the lower resolution runs of phase 2

- The improvement from going from 18 to 26 nodes, a 44% increase, results in about a 30% reduction in time to completion



Phase 3 – speed of I/O

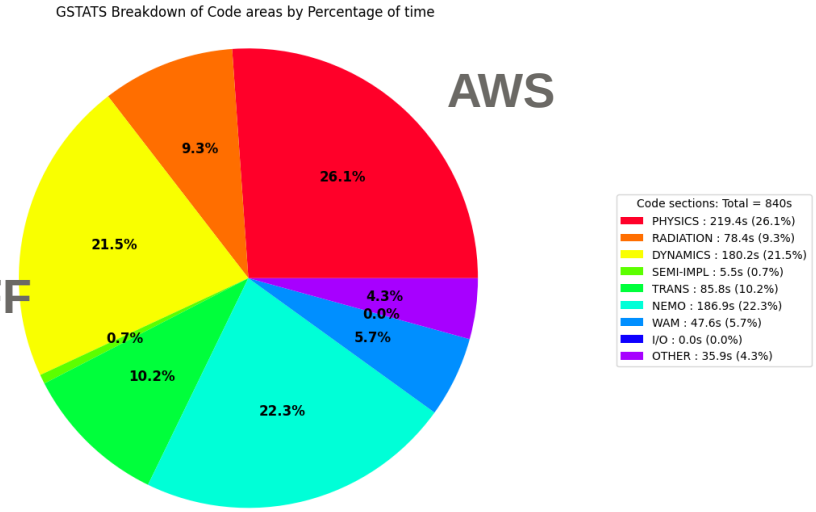
NODES	I/O Enabled	VOLUME (Gbytes)	AVERAGE RATE (Gbytes/second)	I/O TIME (seconds)	IFSTime (seconds)
18+1	Yes	345.4	1.1	346.5	2288
18+0	Yes	345.4	3.7	93.1	2377
18	No	N/A	N/A	N/A	2152
26+2	Yes	345.4	1.1	334.9	1665
26+0	Yes	345.4	3.5	98.3	1885
26	No	N/A	N/A	N/A	1625

- The use of the writer nodes resulted in a slower rate of write than when 2-cores-per compute node were used – but the penalty of stopping the computation to do the write in that manner significantly increases the time to completion

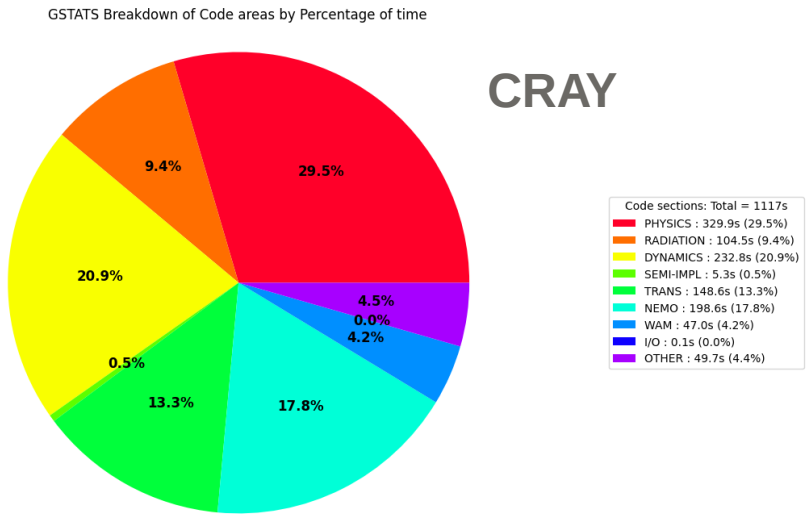
Phase 3 – how enabling I/O impacts time to completion

I/O OFF

AWS

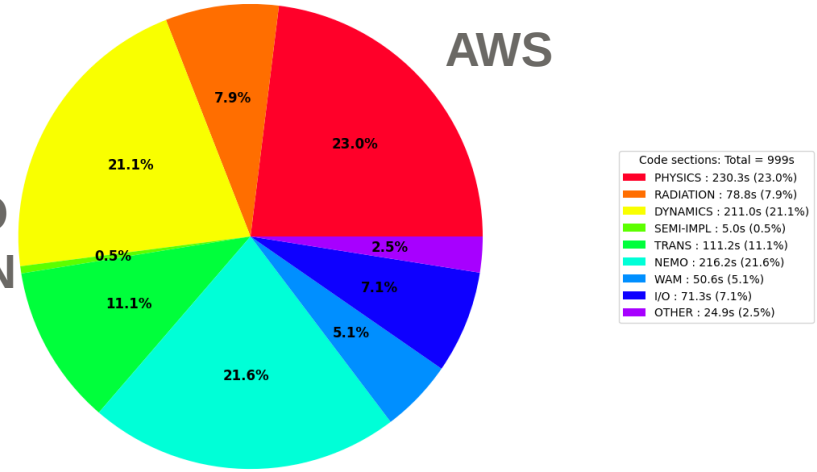


CRAY

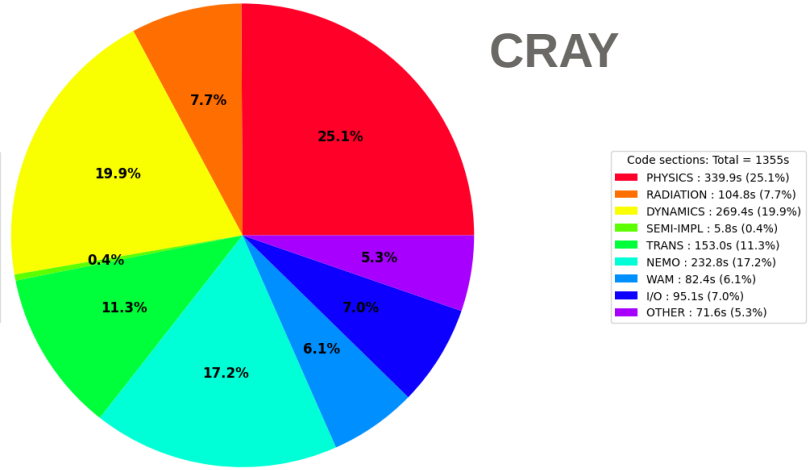


I/O ON

AWS



CRAY



- Enabling I/O, adding in 2 I/O nodes on to 28 compute nodes, results in an expected increase in time to completion on AWS and CRAY.

Lessons Learned

- When thinking of HPC in the cloud – how would you use a cab versus a personal car?

- There are a number of pros and cons to the use of cloud computing for HPC work.
- The use of cloud requires the build of your hardware and software every time you need it
- There is a cost to spin up and down a cluster in the cloud, it takes some time and you pay for that time even though no compute happens. Just leaving it in place incurs costs need to be smart about use
- Also, moving data into and out of non-posix storage (S3) presents some challenges (e. g. sym links)
- Any time sensitive work needs to be planned in advance so that all the pieces are in place when you need the resources to be available.
- Over time, vendor lock can happen – though changing cloud vendors is a months (as opposed to years or weeks) process

Lessons Learned

- When thinking of HPC in the cloud – how would you use a cab versus a personal car?
- There are a number of pros and cons to the use of cloud computing for HPC work.
- The cost of cloud is higher than well allocated on-premises hardware though to make a fair comparison, the cost of electricity and facility should be included in the on-premises cost estimate)
- The benefits of using the cloud are to obtain a 'burst' capacity above what is in continual use.
- The use of cloud allows a many users more rapid access to new hardware.
- Cloud offers easy access to alternate hardware, to find best match for the task (such as GPU)
- Cloud offers the opportunities to have a large amount of resources for a short period of time (like a reanalysis effort). No need to add on additional facility space or electricity to your site.

MAXAR.COM