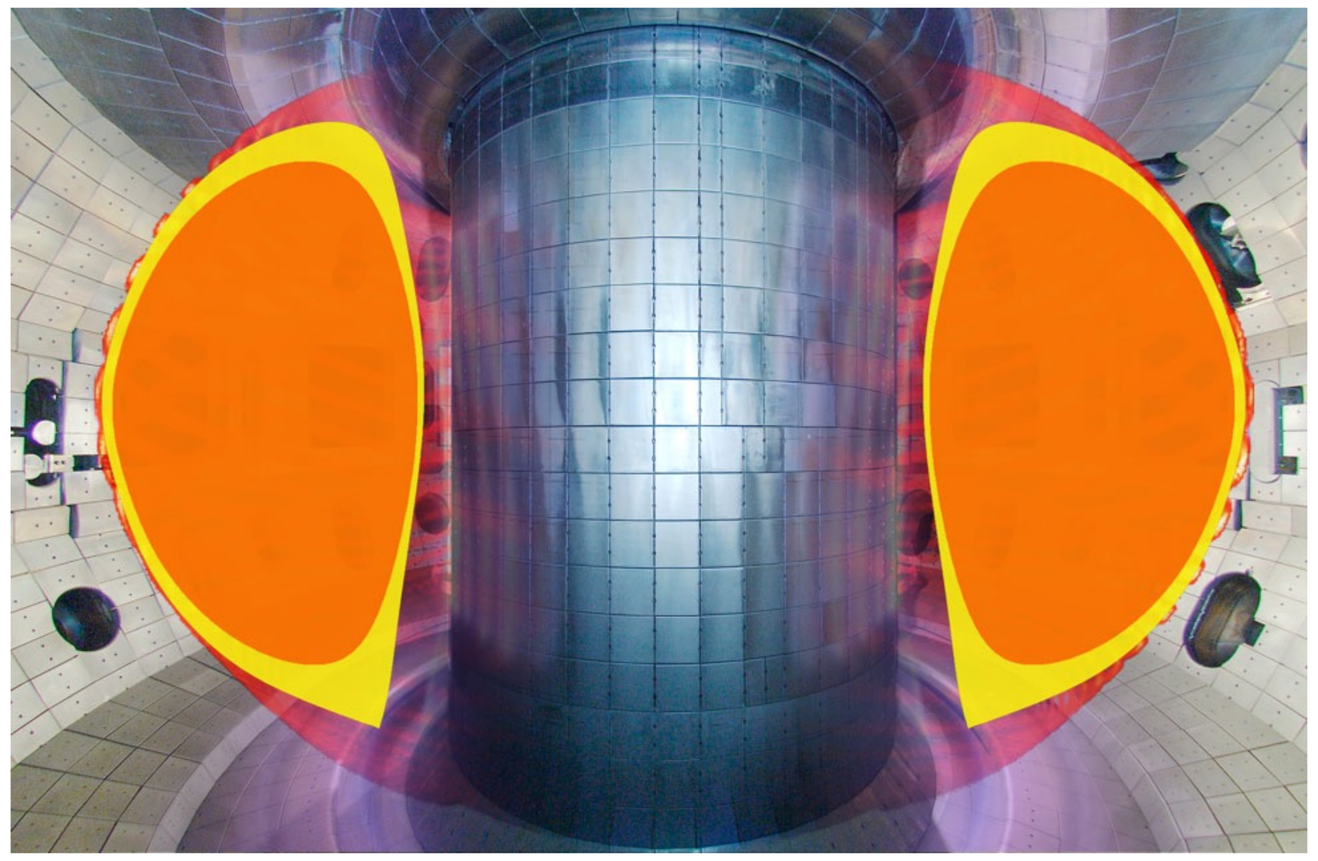
Fusion research using Azure A100 HPC instances

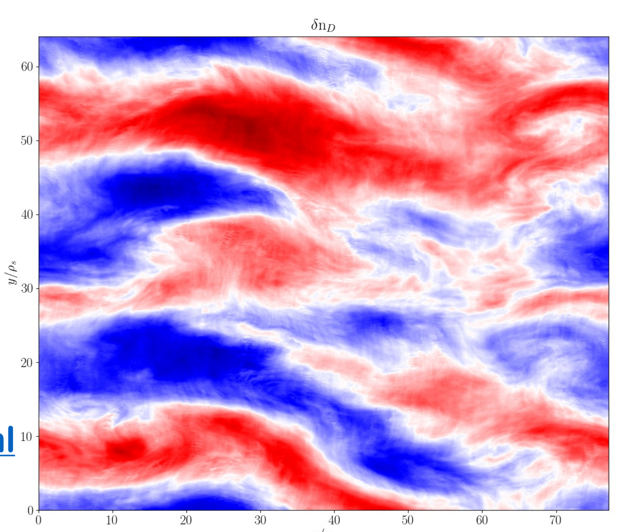
Igor Sfiligoi – University of California San Diego   Jeff Candy – General Atomics   Devarajan Subramanian – Drizti Inc

CGYRO is a premier tool for multi-scale plasma turbulence simulation and has been in use by the fusion community for several years.

Several aspects of fusion energy physics are still not well understood. The complexity of the turbulence in toroidal plasmas makes it difficult to accurately predict fusion reactor performance

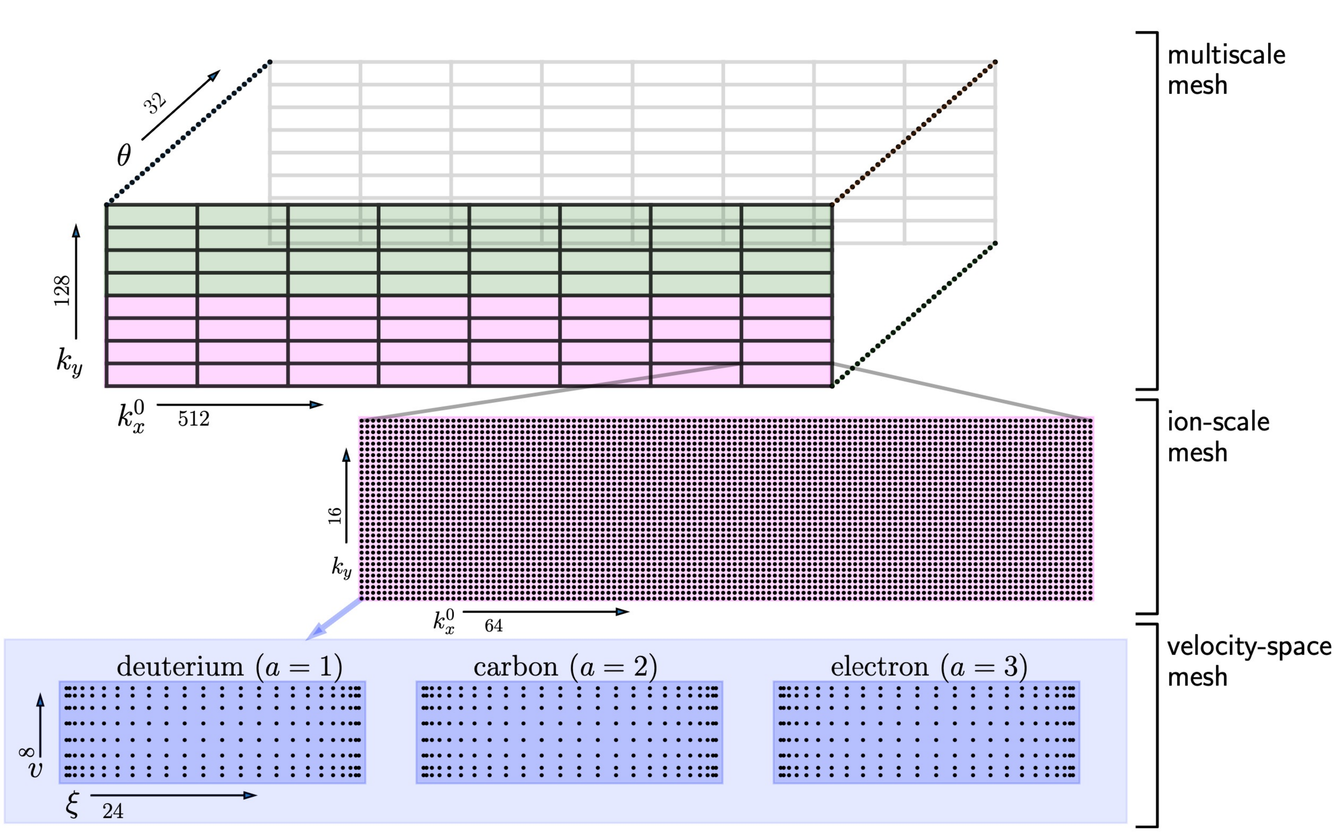


Experimental methods are essential for gathering new operational modes. But simulations are used to validate basic theory, plan experiments, interpret results on present devices, and ultimately to design future devices.



https://gafusion.github.io/doc/cgyro.html

CGYRO is an Eulerian gyrokinetic solver. The gyrokinetic equations specify a 5D distribution function for each species, with species coupled through both the Maxwell equations and collision operator.



The benchmarked nl03 5D mesh is shown above. Its resolution is roughly typical of a multi-scale simulation. The relation of the multiscale wavenumber domain (512 × 128) to an ion-scale domain (64 × 16) is also illustrated.

The high parallelism is great for GPU compute, but memory needs can be a limiting factor. The benchmarked nl03 requires about 560 GB when using 16 GPUs (16 x 35GB).

https://github.com/gafusion/gacode/tree/master/cgyro/tools/input/nl03

In the past CGYRO simulations were only run on leadership-class HPC systems, since one would require hundreds, or even thousands of nodes to get enough compute power and memory for multi-scale fusion simulations.
Recent advances in hardware capabilities now allow for just a couple of nodes to deliver the necessary compute power and memory.

Microsoft Azure

While leadership-class HPC centers provide for a very pleasant and effective work environment, they are heavily sought after and typically over-subscribed. Cloud computing has been recently identified as a viable alternative.

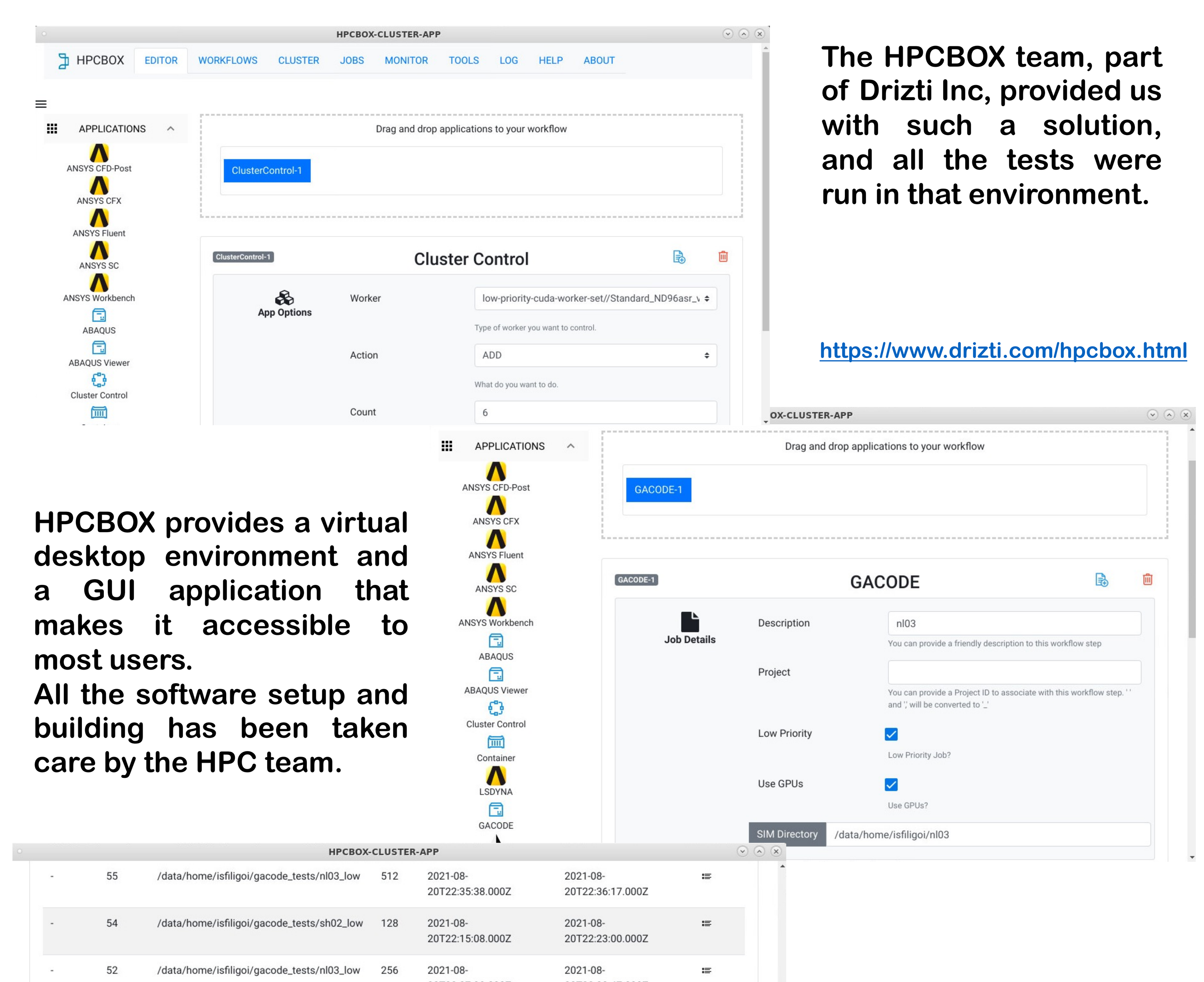
Microsoft Azure Cloud offers several HPC instance types, several of which have been benchmarked in the past. In this poster we focus on the newly released ND A100 v4 instances, which provide

- 8 x NVLINK-connected GPUs,
- 96 x CPU cores and
- 8 x 200Gbps InfiniBand ports.

Size	Physical CPU Cores	Host Memory (GB)	GPUs	Local NVMe Temporary Disk	NVIDIA InfiniBand Network	Azure network
Standard_ND96asr_v4	96	900 GB	8 x 40 GB NVIDIA A100	6,500 GB	8 x 200 Gbps	40 Gbps

DRIZTI

One of the limiting factors of Cloud HPC resources is the lack of native HPC work environments. In order to make CGYRO accessible to a typical Fusion scientist, a turn-key solution is essential.



The HPCBOX team, part of Drizti Inc, provided us with such a solution, and all the tests were run in that environment.

https://www.drizti.com/hpcbox.html

HPCBOX provides a virtual desktop environment and a GUI application that makes it accessible to most users.
All the software setup and building has been taken care by the HPC team.

Challenges using multiple IB interfaces with OpenMPI

The Azure ND A100 v4 instances do not expose the detailed hardware topology behind the hypervisor. Since the topology is not exposed inside the instance, the MPI layer has no way to auto-detect the proper routing.

```
[isfiligoi@a5ca0d6f87-ccn-4 ~]$ nvidia-smi topo -m
        GPU0    GPU1    GPU2    GPU3    GPU4    GPU5    GPU6    GPU7    mlx5_ib mlx5_ib mlx5_ib mlx5_ib mlx5_ib mlx5_ib mlx5_ib mlx5_ib CPU Affinity    NUMA Affinity
GPU0     X      NV12    NV12    NV12    NV12    NV12    NV12    NV12    NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE    0-23    0
GPU1    NV12     X      NV12    NV12    NV12    NV12    NV12    NV12    NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE    0-23    0
GPU2    NV12    NV12     X      NV12    NV12    NV12    NV12    NV12    NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE    0-23    0
GPU3    NV12    NV12    NV12     X      NV12    NV12    NV12    NV12    NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE    0-23    0
GPU4    NV12    NV12    NV12    NV12     X      NV12    NV12    NV12    NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE    0-23    0
GPU5    NV12    NV12    NV12    NV12    NV12     X      NV12    NV12    NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE    0-23    0
GPU6    NV12    NV12    NV12    NV12    NV12    NV12     X      NV12    NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE    0-23    0
GPU7    NV12    NV12    NV12    NV12    NV12    NV12    NV12     X      NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE    0-23    0
mlx5_ib NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE     X      NODE    NODE    NODE    NODE    NODE    NODE    NODE
mlx5_ib NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE     X      NODE    NODE    NODE    NODE    NODE    NODE
mlx5_ib NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE     X      NODE    NODE    NODE    NODE    NODE
mlx5_ib NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE     X      NODE    NODE    NODE    NODE
mlx5_ib NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE     X      NODE    NODE    NODE
mlx5_ib NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE     X      NODE    NODE
mlx5_ib NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE     X      NODE
mlx5_ib NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE    NODE     X

Legend:

  X    = Self
  SYS  = Connection traversing PCIe as well as the SMP interconnect between NUMA nodes (e.g., QPI/UPI)
  NODE = Connection traversing PCIe as well as the interconnect between PCIe Host Bridges within a NUMA node
  PHB  = Connection traversing PCIe as well as a PCIe Host Bridge (typically the CPU)
  PXB  = Connection traversing multiple PCIe bridges (without traversing the PCIe Host Bridge)
  PIX  = Connection traversing at most a single PCIe bridge
  NV#  = Connection traversing a bonded set of # NVLinks
```

CGYRO has to be compiled with the NVIDIA HPC SDK since it relies on OpenACC pragmas for GPU compute.

- We typically use the NVIDIA-provided OpenMPI, but this version does not allow for manual configuration of network traffic routing. This results in sub-optimal network pairing, i.e., treating all IB ports as equivalent.
- OpenMPI 4.1.1 does allow manual tunning, so we also manually compiled it with the NVIDIA HPC SDK and tied each GPU to its own IB port there. Unfortunately, this version seems to have an upper limit on the number of processes per GPU that it can handle, i.e., 4 processes per GPU and 32 processes per node.

Availability of Azure ND A100 v4 instances was also problematic; this new resource seems to be in high demand, and it was difficult to obtain more than 16 instances at any point in time. It should be noted that getting resources in spot mode was easier than on-demand, due to the difficulty of raising the quota limits.

Measured runtimes for the nl03 benchmark case

ND A100 v4 Azure instance

All tests were run using HPCBOX setup, mostly in spot (low priority) mode.

The numbers represent one typical step during the simulation.

Nodes	MPI version	MPI ranks per node	Total time	Comm. time
2	NVIDIA	64	**200s**	166s
	4.1.1	32	**265s**	224s
4	NVIDIA	64	**150s**	132s
	4.1.1	32	**160s**	137s
8	NVIDIA	64	**117s**	103s
	4.1.1	32	**100s**	88s
16	4.1.1	32	**67s**	60s

NVIDIA MPS used for GPU sharing.

Due to the nature of the compute algorithm, the most MPI-intensive communication is between processes working on the same slice of the top sub-problem, so packing 64 MPI ranks on a single node is highly desirable for nl03.
Unfortunately, that was not possible with OpenMPI 4.1.1, but better network routing still compensates at higher scales.

Comparison with previously published results

Older generation Azure HPC instances

Instance	Nodes	Total time	Comm. time	
NDv2	4	397s	293s	NVIDIA V100 GPUs
NDv2	16	121s	92s	NVIDIA V100 GPUs
HBv2	9	289s	64s	AMD EPYC CPUs
HBv2	36	87s	45s	AMD EPYC CPUs

On-prem instances

	System	Nodes	Total time	Comm. time
	Summit	16	82s	46s
INTEL KNL CPUs	Cori	16	372s	120s
INTEL KNL CPUs	Cori	64	112s	46s

https://sc20.supercomputing.org/proceedings/tech_poster/tech_poster_pages/rpost106.html

The recently added Microsoft Azure ND A100 V4 instances provide an appealing option for running medium-scale CGYRO fusion simulations in the Cloud. At small instance counts, these instances are faster than both OLCF Summit at the same node count and NERSC Cori at significantly higher node count.
Given that the availability of Azure ND A100 V4 instances is currently still very constrained, the leadership-class systems are still needed for any large-scale simulation. But for fusion scientists that have no access to on-prem HPC resources, the Cloud resources now provide a viable path forward for more ordinary simulation needs.

This work was supported by the U.S. Department of Energy under awards DE-FG02-95ER54309 (General Atomics Theory grant), DE-SC0017992 (AToM SciDAC-4 project).

U.S. DEPARTMENT OF ENERGY