High Level Optimization Techniques on Pratyush HPC

Bipin Kumar[1], Nachiket Manapragada[2] and Neethi Suresh[1]

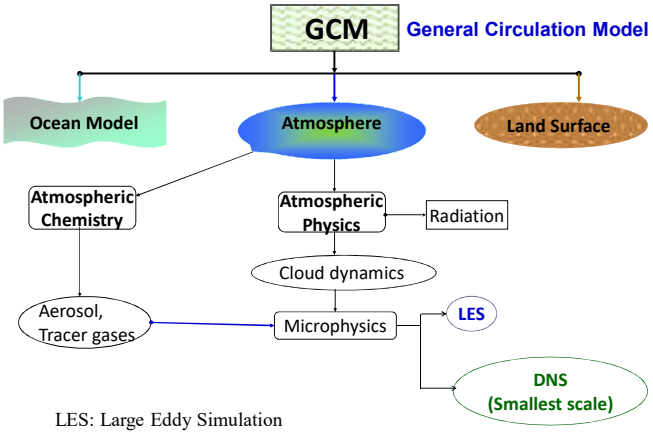
[1]HPCS, Indian Institute of Tropical Meteorology, Pune, Maharashtra, India

[2]Cray Supercomputers (India) Pvt. Ltd, Pune, Maharashtra, India

Contact: bipink@tropmet.res.in

Introduction

Direct Numerical Simulation (DNS) is a three dimensional simulation approach to simulate turbulent flows at smallest scale where the equations are solved explicitly without any approximation. Current study presents an approach to improve the DNS code by overcoming the drawbacks of conventional I/O format as well system level optimizations. A significant reduction in simulation time is achieved.



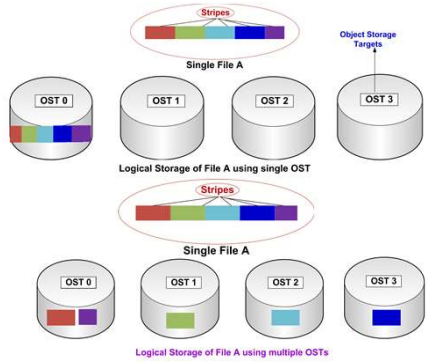
LES: Large Eddy Simulation

DNS: Direct Numerical Simulation

Methodology and Results

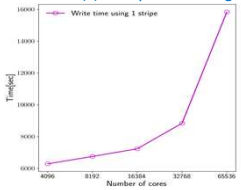
Lustre File system

Lustre file system is actually a set of many small file systems, which are referred to as Object Storage Targets (OSTs). The Lustre software presents the OSTs as a single unified file system. It can support multi-petabytes of storage with a GB/S throughput. The Lustre file systems has ability to stripe data across multiple OSTs in a round-robin fashion. Basically, files can be split up into multiple chunks that will then be stored on different OSTs across the Lustre system.



Striping Optimization

Advantage: (i) Increase in bandwidth utilization and
(ii) ability to store large files taking more space then a single OST



Single file writing time using 1 stripe. The time increases as number of cores increases.

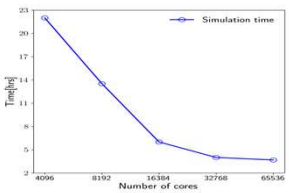
No. of cores	Reading time	Writing time
4K	2250	840
8K	2880	960
16K	3000	960
32K	3720	1080
65K	8640	2280

Average times (seconds) for a single file reading and writing after using file striping optimization. Note that writing time for 4096 core has come down from 6000 to 840 seconds. Similarly it reduces from 16000 seconds to 2280 for 65000 core.

IOBUF Optimization

- Library; enables asynchronous caching and prefetching.
- No source code modification required.
- IOBUF optimization reduced 14% total time.

#4096 Cores	I/O Operations	Total time (sec)
No IOBUF	2 reads + 2 writes	23120
With IOBUF	2 reads + 2 writes	19908 | (14% reduction)



Scaling of total times (in hours) excluding I/O time. The scaling is linear till 16000 cores.

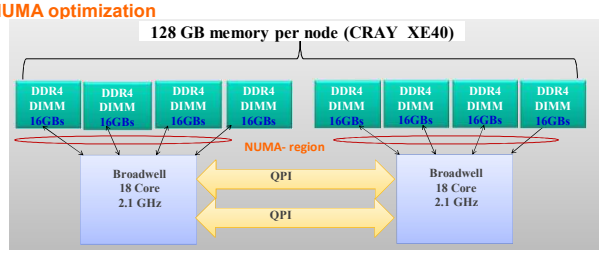
Methodology and Results

Vectorization

- Pratyush HPC has Broadwell processor which supports AVX2 vectorization.
- It provided 5% speedup in total time.

4096 Cores	I/O Operations	Total time (sec)
AVX	1 reads + 1 writes	9005
AVX2	1 reads + 1 writes	8555 | (5% reduction)

NUMA optimization



Pratyush node :

2 Broadwell processors with 64 GB RAM providing total 128 GB RAM which either processor can access.

NUMA: Non Uniform Memory Access

Each of processors, with its corresponding memory, NUMA-region.

- It is generally much faster for a CPU to access its local NUMA node memory. Accessing the remote NUMA-region shall have higher latency and may slow down performance.
- Only 24 MPI cores out of the 36 were used to provide optimal utilization of RAM.
- A restriction of 12 cores per CPU was imposed. Without this restriction, the system may consume all the 18 cores on one CPU creating possible imbalance for proper memory utilization.

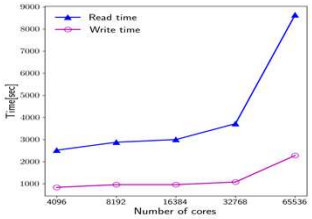
# Cores	With NUMA
4096	6%
8192	15%
16384	17%
32768	38 %

Up to 38% reduction in simulation time has been achieved with NUMA

Conclusion and outlook

A DNS code has been optimized using four different types of techniques. The optimization experiments done here are mainly focused on high level file system and parallel I/O optimization. The computational domains for all experiments have (4096)3 grid points. The whole work can be summarized as follows

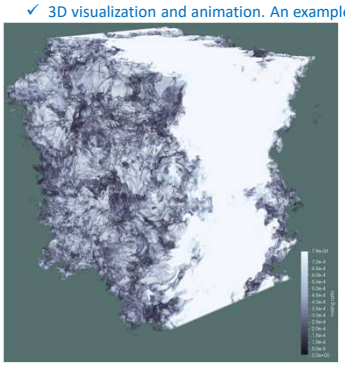
- **Optimized DNS code using parallel I/O optimization.**
- **Four optimization techniques attempted.**
  - ✓Striping in Object Storage Targets (drastically reduced file processing time)
  - ✓IOBUF optimization provided 14% reduction.
  - ✓AVX2 added 5% more time reduction.
  - ✓NUMA optimization gave 38% speedup.
- **Overall scaling has shown linear speedup till 16K cores.**
- **Further experiment**
  - ✓Hyper threading
  - ✓Multithreading
  - ✓Use of advanced MPICH options
  - ✓Increased number of aggregators per OST
  - ✓Experiment on file system with more than 10 OSTs.



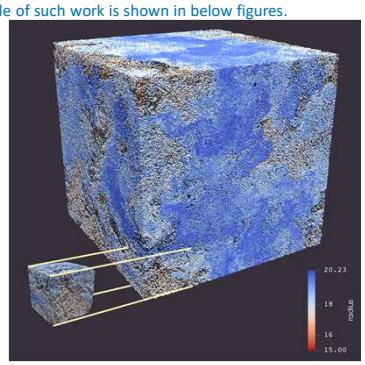
Scaling of parallel I/O time after optimization.

Visualization

✓ 3D visualization and animation. An example of such work is shown in below figures.



Visualization of vapor mixing ration in computational domain. The central cloud slab clearly visible in white color.



Visualization of cloud droplets. The color represents droplet size with blue one is bigger droplets and small are shown by red color.

References

1. Kumar, B., Manapragada, N. and Suresh, N. "High Level file system and parallel I/O optimization of DNS Code", *Second Workshop on Software Challenges to Exascale Computing*, 13-14, Dec., 2018, New Delhi, India.
2. Kumar, B. Rehme, M. and Suresh, N., "Visualization of Droplet Dynamics in Cloud Turbulence", *The International Conference for High Performance Computing, Networking, Storage, and Analysis*, 11-16 Nov, 2018, Dallas, TX, USA.