FINAL PROJECT REPORT

NAS PARALLEL BENCHMARKS

MOHIT PALLIYIL SATHYASEELAN, VISHNU V

Introduction

In this project, we have implemented the 8 different kernel benchmarks in the NAS parallel benchmarks. The Numerical Aerodynamic Simulation (NAS) Program, which is based at NASA Ames Research Center is a research group that focuses on advancing the state of computational aerodynamics. We have studied in great detail about these different benchmarks and how they work.

Kernel Benchmarks

- Integer Sort (IS)
- Embarrassingly parallel (EP)
- Multigrid (MG)
- Conjugate gradient (CG)
- 3-D FFT PDE (FT)
- LU solver (LU)
- Pentadiagonal solver (SP)
- Block tridiagonal solver (BT)

S.no	Benchmark	Parameter	Class S	Class W	Class A	Class B	Class C	Class D
1	**CG**	no. of rows	1400	7000	14000	75000	150000	1500000
		no. of nonzeros	7	8	11	13	15	21
		no. of iterations	15	15	15	75	75	100
		eigenvalue shift	10	12	20	60	110	500
2	**EP**	no. of random-number pairs (pow(2,x))	24	25	28	30	32	36
3	**FP**	grid size	64 x 64 x 64	128 x 128 x 32	256 x 256 x 128	512 x 256 x 256	512 x 512 x 512	2048x1024 x1024
		no. of iterations	6	6	6	20	20	25
4	**IS**	no. of keys (pow(2,x))	16	20	23	25	27	31
		key max. Value (pow(2,x))	11	16	19	21	23	27
5	**MG**	grid size	32 x 32 x 32	128 x 128 x 128	256 x 256 x 256	256 x 256 x 256	512 x 512 x 512	1024 x 1024 x 1024

		no. of iterations	4	4	4	20	20	50
6	**BT**	grid size	12 x 12 x 12	24 x 24 x 24	64 x 64 x 64	102 x 102 x 102	162 x 162 x 162	408 x 408 x 408
		no. of iterations	60	200	200	200	200	250
		time step	0.01	0.0008	0.0008	0.0003	0.0001	0.00002
7	**LU**	grid size	12 x 12 x 12	33 x 33 x 33	64 x 64 x 64	102 x 102 x 102	162 x 162 x 162	408 x 408 x 408
		no. of iterations	50	300	250	250	250	300
		time step	.5	.0015	2	2	2	1
8	**SP**	grid size	12 x 12 x 12	36 x 36 x 36	64 x 64 x 64	102 x 102 x 102	162 x 162 x 162	408 x 408 x 408
		no. of iterations	100	400	400	400	400	500
		time step	.015	.0015	.0015	.001	.00067	.0003

NAS Parallel Benchmarks Problem Size

<u>Integer Sort</u>

This is a large integer sort kernel.It performs an integer sort among a sparse set of numbers, which can be compared with particle-in-cell applications. By default, the sorting method is based on the bucket sorting approach. Accordingly, the number of keys for each bucket is determined, and the total count is distributed among each bucket. When completed, each bucket receives sorted numbers and points to the final accumulated sizes. Finally, the keys within each bucket are sorted, and a partial test is performed to verify the results.

Computer System	**Date Received**	**Number Processor**	**Class A**	**Class B**
Intel Paragon (SunMos)	Jan 95	32 64 128	5.48 3.77 2.29	NA 11.98 7.22
Thinking Machines CM-500	Sept 95	64 128 256	3.16 1.67 1.16	16.1 8.2 4.3
Fujitsu VPP700	Nov 96	1 2 4 8 16 32	2.3968 1.8038 1.2519 1.1249 1.0204 0.9839	9.1964 6.0875 4.1363 3.1959 2.7231 2.7211

HiPerGator AI: NVIDIA DGX A100	**Number Processor**	**Class S**	**Class A**	**Class B**
OMP	1 2 4 8 16 32	0.00 0.00 0.00 0.00 0.01 0.01	0.45 0.44 0.39 0.38 0.34 0.32	1.74 1.74 1.35 1.27 1.21 1.20
CUDA		0	0	0
MPI (ScoreP)	1 2 4 8 16 32	0.21 0.17 0.13 0.12 0.11 0.11	4.66 NA NA NA NA NA	17.97 NA NA NA NA NA

<u>Embarassingly Parallel</u>
This kernel provides an estimate of the higher limits that can be achieved for a floating point performance. It generate pairs of Gaussian random variables according to a particular method and finds out the number of consecutive square annuli. After they are calculated, they are verified by testing. It is typical of many Monte Carlo simulations.

Computer System	**Date Received**	**Number Processor**	**Class A**	**Class B**
Intel Paragon (SunMos turbo)	Jan 95	64 128 256 512 1024	7.80 3.93 2.00 1.12 .59	31.15 15.60 7.82 3.98 2.05
Thinking Machines CM-500	Sept 95	64 128 256	6.9 3.6 2.0	27.1 13.7 7.0
Fujitsu VPP700	Nov 96	1 2 4 6 16	30.3312 15.3105 7.3720 3.9217 1.9962	120.9485 57.6907 30.4367 14.6048 7.7563

		32	1.0265	3.7444

HiPerGator AI: NVIDIA DGX A100	**Number Processor**	**Class S**	**Class A**	**Class B**
OMP	1 2 4 8 16 32	1.52 1.66 1.73 1.73 1.86 2.22	29.83 27.62 27.54 24.41 23.90 23.73	119.69 115.32 113.48 112.56 109.54 108.21
CUDA	1	0	0	0

Multigrid

It is a simplied multigrid kernel that requires very structured long distance communication. The function of thai kernel is to test both short distance and long distance communication. To obtain the answer to a discrete Poisson problem, we use four iterations of the multigrid algorithm. As like the Embarassingly Parallel benchmark, here also we use verification testing. The timing should be performed in a correct manner in order to ensure accurate results.

Computer System	**Date Received**	**Number Processor**	**Class A**	**Class B**
Intel Paragon (SunMos turbo)	Jan 95	64 128 256 512	5.85 3.17 1.87 1.46	27.9 15.2 9.05 7.01
Thinking Machines CM-500	Sept 95	64 128 256	2.2 1.41 0.91	10.6 6.2 3.9
Fujitsu VPP700	Nov 96	4 8 16 32	1.38 0.75 0.41 0.25	6.58 3.59 1.95 1.30

HiPerGator AI: NVIDIA DGX A100	**Number Processor**	**Class S**	**Class A**	**Class B**
OMP	1 2 4 8 16 32	0.00 0.00 0.00 0.01 0.02 0.03	0.82 0.80 0.80 0.79 0.78 0.78	4.29 4.29 4.32 4.18 4.11 3.97
CUDA		0	0	0

Conjugate gradient

This kernel is used to calculate an approximate value to the eigenvalue of lowest value of a large sparse symmetric matrix that is positive definite. It tests long distance communication that is not regular using matrix vector multiplication that is not structured. To find an estimate of the eigenvalue of highest value of a positive definite matrix that is symmetric and sparse, it utilizes the inverse power method. This confirms that the least amount of time is involved in calculating the results.

Computer System	**Date Received**	**Number Processor**	**Class A**	**Class B**
Intel Paragon (SunMos)	Jan 95	64 128 256 512	3.59 2.76 2.44 NA	NA 125.4 63.6 40.5
Thinking Machines CM-500	Sept 95	64 128 256	5.4 3.9 3.4	149 91 62
Fujitsu VPP700	Nov 96	1 2 4 8 15 16 30 35	5.8703 3.1802 1.8185 1.0656 NA 0.7514 NA 0.6726	NA 104.1408 57.8277 32.0605 19.7696 NA 13.4585 NA

HiPerGator AI: NVIDIA DGX A100	Number Processor	Class S	Class A	Class B
OMP	1 2 4 8 16 32	0.23 0.23 0.25 0.25 0.29 0.35	0.98 0.96 0.94 0.83 0.82 0.78	44.52 44.31 43.05 42.46 42.17 41.69
CUDA		0.01	0.01	0.05

<u>3-D FFT PDE</u>

It uses FFTs to find a 3D partial differential solution. It performs the main part of many codes that are spectral. Like Multigrid and Conjugate gradient kernel benchmarks, this kernel also tests the performance of long distance communication ina rigorous manner. This kernel involves a lot of mathematical operations as it performs FFTs. They are an integral part of computational fluid dynamics applications, mainly large simulations involving edy turbulence. They need a lot of communications to perform basic operations such lie the transposition of arrays.

Computer System	Date Received	Number Processor	Class A	Class B
Intel Paragon (SunMos)	Jan 95	16 32 64 128 256 512	20.46 10.54 5.47 3.13 2.82 NA	NA NA 60.9 32.15 18.66 16.17
Thinking Machines CM-500	Sept 95	64 128 256	3.5 1.96 1.33	41.9 22.5 13.2
Fujitsu VPP700	Nov 96	1 2 4 8 16 32	6.3272 4.3788 2.2557 1.1491 0.6005 0.3345	NA 49.6885 24.5128 12.3758 6.3013 3.2577

HiPerGator AI: NVIDIA DGX A100	Number Processor	Class S	Class A	Class B
OMP	1 2 4 8 16 32	0.09 0.09 0.10 0.10 0.10 0.10	2.80 2.78 2.77 2.75 2.74 2.74	NA NA NA NA NA NA
CUDA		0	0	0

<u>LU solver</u>

LU solver kernel is a solution for system for regular block fo five lower and upper triangles that are sparse.The kernel stands for the different calculations involved in a fixed operator of computational fluid dynamics algorithms that are relatively newer. When compared to the pentadiagonal solver and block tridiagonal solver kernels, this kernel uses lesser parallelism. However this does not affect the performance of this benchmark on a large scale.

Computer System	Date Received	Number Processor	Class A	Class B
Thinking Machines CM-500	Sept 95	64 128 256	90 61 43	336 247 149
Fujitsu VPP700	Nov 96	1 2 3 4 6 8 16 17 32	77.732 50.880 NA 30.500 NA 18.444 12.553 NA 10.063	297.94 190.00 138.34 NA 78.026 NA NA 36.736 NA

HiPerGator AI: NVIDIA DGX A100	Number Processor	Class S	Class A	Class B
OMP	1	0.02	22.60	94.10

	2 4 8 16 32	25.40 44.07 104.22 133.70 135.48	NA NA NA NA NA	NA NA NA NA NA
CUDA		0	0	0

Pentadiagonal solver

This kernel is the answer to numerous independent systems of pentadiagonal equations that are scalar and dominant non-diagonally. This kernel is very similar to the block tridiagonal solver kernel but the main difference is that the ratio between the communication and computation in both the kernels is significantly different because they are meant to test out different applications. This helps us identify the solutions in a computationally intensive way while compromising the various issues involved with communication.

Computer System	**Date Received**	**Number Processor**	**Class A**	**Class B**
Thinking Machines CM-500	Sept 95	64 128 256	85 51 34	405 236 140

HiPerGator AI: NVIDIA DGX A100	**Number Processor**	**Class S**	**Class A**	**Class B**
OMP	1 2 4 8 16 32	0.01 0.02 0.03 0.05 0.09 0.18	12.55 12.43 12.29 11.72 11.70 11.35	64.26 63.53 62.84 61.97 60.99 60.02
CUDA		0	0	0

Block tridiagonal solver

It is a solution to the various systems of block of tridiagonal equations that are non-diagonally dominant with a block size of 5. As mentioned earlier, this kernel is very similar to the pentadiagonal solver kernel where the main difference between the two comes down to the difference in the ratio between computation and communication. It is very effective in solving non-linear partial differential equations.

Computer System	**Date Received**	**Number Processor**	**Class A**	**Class B**
Intel Paragon (SunMos)	Jan 95	64 102 128 204 306	224.0 NA 113.0 NA NA	NA 598.0 NA 324.0 215.0
Thinking Machines CM-500	Sept 95	64 128 256	75 43 27	370 209 114
Fujitsu VPP700	Nov 96	1 2 3 4 6 8 16 17 32 34	129.40 68.510 NA 35.518 NA 18.088 9.2841 NA 4.9286 NA	487.29 258.20 175.09 NA 88.541 NA NA 31.634 NA 16.824

HiPerGator AI: NVIDIA DGX A100	**Number Processor**	**Class S**	**Class A**	**Class B**
OMP	1 2 4 8 16 32	0.05 0.05 0.05 0.06 0.08 0.12	35.45 34.84 34.51` 34.16 34.05 34.04	143.67 142.93 142.42 142.28 141.89 141.36
CUDA		0	0	0

<u>Conclusion</u>

In this project we have studied about the NAS parallel benchmarks. The 8 different kernel benchmarks were studied in great detail and the code was implemented using OpenMP. The time taken for each class (S,A and B) for different number of processors was calculated and compared with that of various other systems like Intel Paragon, Thinking Machines CM-500 and Fujitsu VPP700. Observations were noted and its effects were analyzed.The performance variations are observed to be compared with theoretical expectations.

References

.https://www.nas.nasa.gov/software/npb.html

.https://www.nas.nasa.gov/assets/pdf/techreports/1994/rnr-94-007.pdf

.https://en.wikipedia.org/wiki/NAS_Parallel_Benchmarks#:~:text=NAS%20Parallel%20Benchmarks%20(NPB)%20are,the%20NASA%20Ames%20Research%20Center.

.https://digitalcommons.unl.edu/cgi/viewcontent.cgi?article=1149&context=nasapub