Combining Open MP and MPI within GLOMAP Mode; Keeping Pace with Hardware Developments.

Mark Richardson, *Numerical Algorithms Group*
Graham Mann, *School of Earth and Environment, University of Leeds*

ABSTRACT: *The MPI version of GLOMAP MODE is used in production runs for research into atmospheric science. The memory requirement prohibits use of high resolution scenarios so 32 MPI tasks is the usual decomposition. One way to attempt higher resolution simulations is to under-populate the nodes, leaving more memory available per MPI task. Although this is wasteful of resource, it does provide a shorter time per existing simulation. The NAG Ltd DCSE service has examined the code and introduced Open MP so that the otherwise "idle" cores can contribute to the MPI task. It was demonstrated, on the XT4 and XT6, that this improves the performance so that the additional cost of a simulation is reduced.*

KEYWORDS: XT-4h, XT-6, mixed-mode, hybrid, MPI, Open MP, atmosphere, aerosol processes, GLOMAP, TOMCAT, parallel

1. Introduction

GLOMAP MODE MPI

GLOMAP mode MPI is a FORTRAN computer program used to simulate global aerosol processes. It is a combination of a chemical transport model TOMCAT which has algorithms for integrating the gas phase chemistry/deposition and advection of the trace gas and aerosol species around the globe and the main part of GLOMAP which is a size-resolved aerosol microphysics module to simulate processes such as nucleation, coagulation, condensation and cloud-processing.

The simulation is the lower to mid-atmosphere (up to 10hPa) for the earth, the data set particular to this project has "moderate" resolution with 31 vertical levels (in hybrid σ-p co-ordinates) and T42 spectral resolution in the horizontal (2.8 x 2.8^{o} latitude/longitude) . The volume of fluid that forms the atmosphere is mapped onto a three-dimensional Cartesian rectangular block of computational space. The numbers of cells in each direction are as follows:

- Number of latitudes: 128 (the I loop index within the code)
- Number of longitudes: 64 (the K loop index within the code)
- Number of vertical layers: 31 (the L loop index within the code)

GLOMAP uses "offline meteorology" to drive the transport and wet removal, reading in data produced from other numerical meteorological simulation software (6-hourly ECMWF analyses). There are two main versions of GLOMAP – Mode and Bin, using modal and sectional aerosol dynamics approaches respectively, with versions of each in Open MP and the subject of this research is GLOMAP Mode MPI. The Open MP version of the software has been used on SMP class machines (for instance the previous national HPC systems CSAR and HPCx) and thus has a soft limit maximum of number of threads set by the outermost loop (over latitude). At the resolution of the test case this is set to 64. The MPI GLOMAP versions have only been developed recently (2008) and differ from the Open MP versions mostly in the TOMCAT sections of the code, with MPI being used for communication between parallel tasks.

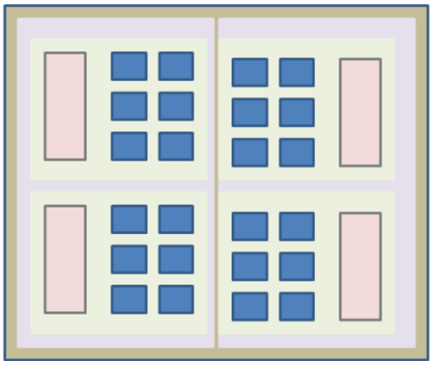
HECToR Phase 2a

The XT4h comprises 1416 compute blades, each of which has 4 quad-core processor sockets. This amounts to a total of 22,656 cores, each of which acts as a single CPU. The processor is an AMD 2.3 GHz Opteron. Each quad-core socket shares 8 GB of memory, giving a total of 45.3 TB over the whole XT4 system. The theoretical peak performance of the system is 208 Tflops. In addition there is a small X2 system

HECToR Phase 2b

Phase 2b of the HECToR service will provide a Cray XT6 and access was granted by Cray ahead of the delivery to exercise several applications, incuding GLOMAP. Each node of the XT 6 has 24 cores that are two physical processors known as Magny-Cours. The cores are arranged in groups of 6, because there are effectively two hex-core dies per socket (AMD Istanbul), for this reason the PGI compiler option for targeting processor (-tp) is set to Istanbul-64.

Figure 1 is a schematic of a node of the XT6. The total memory on a node is 32GB.



SEE/NCAS

This project was initiated by Dr. Graham Mann at the School of Earth and Environment at the University of Leeds. He is a NCAS-funded researcher. The National Centre for Atmospheric Science (NCAS) provides a national capability in atmospheric science research through its research programmes and a facilities and support infrastructure distributed over UK Universities

NAG DCSE

The distributed computational science and engineering (DCSE) support function is a resource provided by the Research Councils UK. It can be used to enhance the performance of computer programs and improve working practices for researchers using UK National High Performance Computing Facilities (HECToR). It is specifically a person identified to carry out the work in support of improving the use of national computing resources through better software engineering and education of the users of these facilities.

2. Analysis of code

The typical use of this code is to run decade simulations with a 30 minute time step implies 48 steps per day for one decade would take 3650x48=175200 steps for a CPU time of 2 seconds per step per core would result in a total of 97 hours. In the production case this would be done as several 3 hour jobs (33) which is effectively a full week for a research project.

Previous studies of this code have highlighted specific subroutines that account for the time of the simulation. The Performance Analysis Tools supplied by Cray were used to confirm that these were the case for this MPI version.

The first pass was a basic sampling experiment that was used to guide the tracing experiment. The following table indicates the proportion of time spent in six specific sections of the code. There is a lot more data generated during the Cray PAT experiments although table 1 shows data extracted for the the target subroutines for Open MP work, which dominate the calculations.

Table 1. Cray PAT results of tracing experiment

Configuration	M32 % of whole sim	M32 % of GMM only	M64 % of whole sim	M64 % of GMM only	M128 % of whole sim	M128 % of GMM only
ADVX2	4.2	5.7	2.8	5.5	1.3	5.7
ADVY2	11	14.9	7.1	13.9	2.2	9.7
ADVZ2	4.9	6.6	3.2	6.3	1.4	6.2
CONSOM	5.4	7.3	3.6	7.1	1.1	4.8
CHIMIE	40.9	55.3	27.4	53.7	12.4	54.6
MAIN	7.7	10.4	7	13.7	4.4	19.4
TOTAL FOR GMM	74	100	51	100	22.7	100
MPI	13.3	-	28.3	-	47.4	-
MPI_SYNC	12.7	-	20.7	-	29.9	-

The case name M32 indicates that the run is configured for 32 MPI tasks.

3. Analysis of Advection Parts

The four subroutines (ADVX2, ADVY2, ADVZ2 and CONSOM) that were previously identified as candidates for Open MP treatment have been examined in this more recent version of the code. The Cray PAT analysis shows that they are high in the work load. The following descriptions are based on choosing four Open MP threads per node.

SUBROUTINE ADVX2

For ADVX2 the main outer loop is over L=1,NIV. This is significant for XT6 hardware as, in the T42 model, NIV is 31 levels. For the XT4 hardware the core count per node is 4 so the loops will divide into blocks of 8 iterations per thread. Only one thread will do seven iterations. It is indeterminate which thread works on which loop cycle as the (dynamic, 1) schedule is applied.

SUBROUTINE ADVY2

For ADVY2, the main outer loop is over L=1, NIV. This is the same as ADVX2 therefore the threads also get 8 iterations of work. It is a feature of TOMCAT that the north-south fluxes across the poles have to be treated specially. The fluxes across the poles are given special treatment as the face areas on the polar side of the "grid-boxes" tend to zero as the lines of longitude approach the poles. Thus, the 0.91 version of the subroutine is different to version 0.81 and several MPI calls are needed to deal with the Polar Regions. The method that had been implemented (*in a completely separate and long finished project*) in the MPI version uses a rudimentary global sum. It occurs within the ***L*** iterations and thus inhibits any Open MP implementation. This code was re-written to move these global sums outside the loop and four extra arrays were required to achieve that.

SUBROUTINE ADVZ2

For ADVZ2, the main outer loop is over K=1, MYLAT (and therefore dependent on the MPI decomposition of the computational domain). For the case with 32 MPI Tasks, there are 8 latitudes per patch and there are 32 longitudes cells per patch as in Table 2. For the case configured for 64 MPI tasks there are 4 latitudes per computational domain.

Table 2: Open MP loop limits

Configuration	M32	M64	M128
ADVX2	31	31	31
ADVY2	31	31	31
ADVZ2	8	4	2
CONSOM	36	36	36
CHIMIE	8	4	2

SUBROUTINE CONSOM

The outer loop of CONSOM is over K= 1, MYLAT. However, the Open MP implementation in the previous version had been inserted within the K loop and applied to a loop over JV = 1, NTRA. The reason is that early versions of the code read convection coefficients from external files per latitude. The Open MP implementation was retained because NTRA is 36 for this case and decomposition over 4 Open MP threads gives 9 iterations per thread and does not change for any of the decompositions.

4. Analysis of Aerosol Process Model

SUBROUTINE CHIMIE and the GLOMAP model

The TOMCAT program has a chemistry processing section that is invoked through the subroutine "CHIMIE". It is the interfacing routine between the advection calculations and aerosol chemistry. These processes are dealt with in a per-grid-box, per-latitude manner. Within the loop over latitudes, data from the three-dimensional advection arrays are copied into planes of longitude and altitude. These are stored as one-dimensional vectors of length "NBOX" and the chemistry is applied to individual grid-boxes.

The Open MP treatment is applied to "loop 6" that is the loop over planes of latitude. In the hybrid version of the software each Open MP thread will process a plane as the loop count increases, for example with four threads the first four latitudes are processed by different Open MP threads. The schedule is (dynamic, 1) and thus each thread works on a plane for as long as is needed without delaying the processing of other planes. When a thread has finished work it looks to the loop counter to determine which iteration of the loop it must process next. This helps balance out any workload variation between latitudes.

The GLOMAP functionality is provided through a call to UKCA_AERO_STEP from within the latitude loop of CHIMIE. Thus the work done by GLOMAP is proportional to the number of planes that are processed. If the planes are shared out to the threads then it is anticipated that all the "UKCA" functions will benefit.

ASAD functionality

This code originates from a separate research group and is sourced from a separate repository (UNIASAD0.2X2). The Open MP that existed in the library is not been changed from what has been implemented before (it is possible to use UNIASAD0.2 with UNICAT0.81). The whole of this function is processed below the CHIMIE loop 6 and operates on individual grid-boxes. Additional care was needed to ensure that data in certain COMMON blocks is retained per thread using the THREADPRIVATE directive.

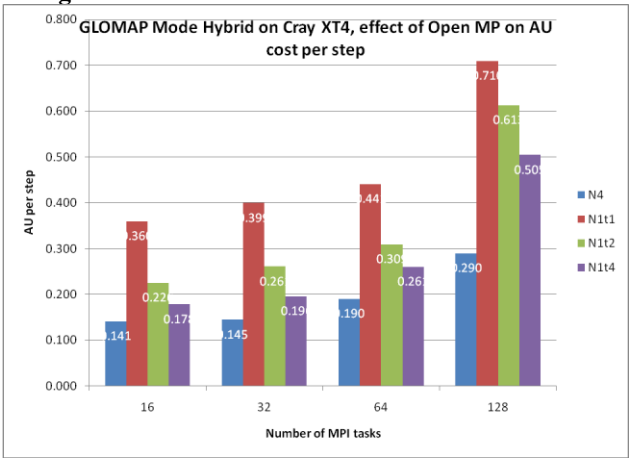
5. Results

Cray XT4h results

HECToR phase 1b was the only system available for nearly the entire duration of the project. The XT6 was available briefly at the end of the project. The nodal arrangement of a single quad-core processor allows three variations in Open MP configuration: N1t4, N1t2, and N2t2, where N is the number of MPI tasks per node and t is the number of threads per MPI task. Figures 2 and 3 present the effect of activating Open MP. The cases are restricted to one MPI task per node to show this effect. Thay are aslo compared to the normal mode of operation of the pure MPI code i.e. 4 MPI tasks per node.

In Figure 2, each group the first column is the pure MPI fully packed on the node. The three remaining columns are for the hybrid code with 1, 2 and 4 threads.

Figure 2. Effect of Open MP on the cost of different configurations on the XT4



As expected the cost per time step is higher when the MPI tasks are spread out to one task per node. However, it is countered to some extent by the fact that the time per step is shorter, as shown in Figure 3.

Figure 3. Effect Of Open MP On Time Per Step

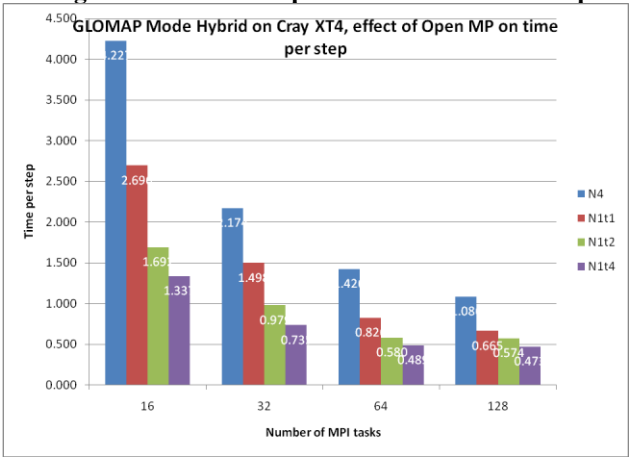
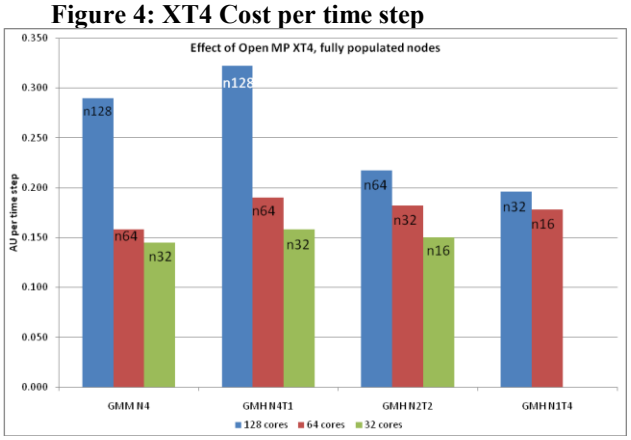


Table 3: XT4h mixed-mode performance on fully populated nodes (equivalent cores) [32 64 128]

XT4h	Cores	N4	N4T1	N2T2	N1T4
AU per step	**128**	0.289	0.322	0.217	0.196
	64	0.158	0.190	0.182	0.178
	32	0.144	0.158	0.149	
Time per step	**128**	1.086	1.209	0.816	0.735
	64	1.426	1.426	1.368	1.337
	32	2.174	2.373	2.249	

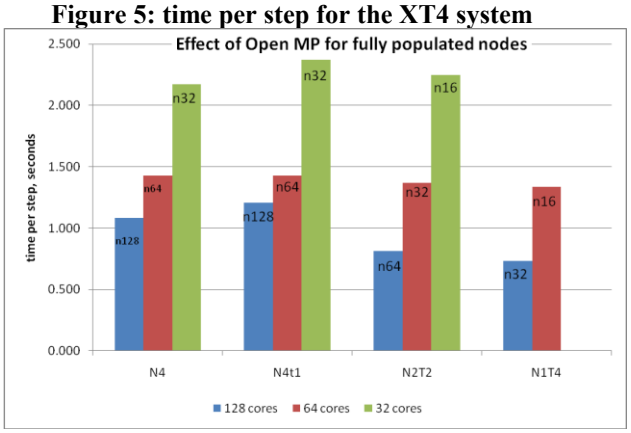
The performance of N4 in table 3 shows that operating with 128 MPI tasks is only providing a speed up of 2. However, using mixed-mode a 32 Mpi task job and 4 threads is achieving a speed up of 3. The comparison of costs for those three points show the purely MPI option to cost twice as much as the 32 MPI tasks. However activatig the Open MP threads with the 32 MPI tasks job only costs 36% more. This makes using 128 cores more realistic.

Figure 4: XT4 Cost per time step



Figures 4 and 5 are associated with Table 3 and show that there is very little difference between the cost of running with solely MPI and running with Open MP enabled, where the same number of cores are working on the problem (e.g. N32T2 compares with N64). However, if the turnaround time is considered then it becomes more attractive. In figure 5 there is a subtle trend that where the node as more MPI tasks the time step is slower.

Figure 5: time per step for the XT4 system



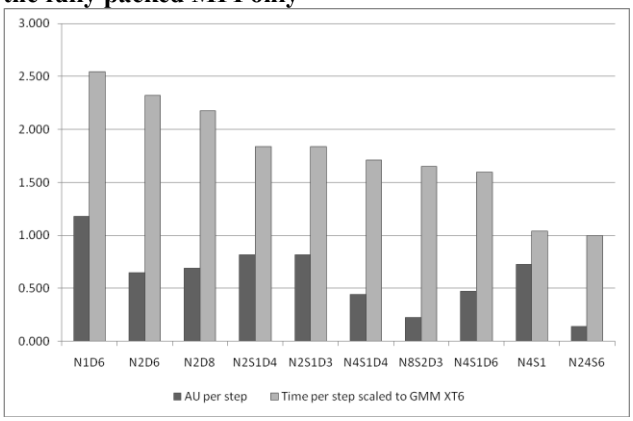
Cray XT6 results

The system has 24 cores per node and 32GB RAM per node. A slight change in notation replaces "*t*" with D to indicate thread depth and additonal parameter "*S*" to indicate the number of MPI taks per hex-core die.

For the case where 64 MPI tasks are chosen, activating Open MP threads provides shorter time steps. Figure 6 shows that there is a "sweet-spot" at N8S2D3, part of the reason is that the cost model is one where the researcher is paying per node. The reference configuration uses three nodes and only 64 of 72 cores. The configuration with 8 tasks per node is using 8 nodes. N2D8 was tested as it best matches the Open MP loop

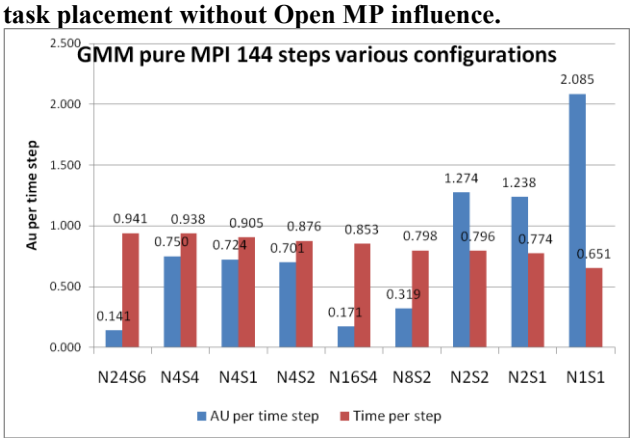
limits but it does not show any significant gain in computation time compared to the N2D6 where all threads are on one hex-core die. The N4S1D4 configuration is most similar to the N1t4 configuration on the XT4h as each MPI task will have good access to 8GB of RAM and spawns three Open MP threads.

Figure 6: XT6 timings of 64 MPI tasks normalised to the fully packed MPI only



Other notable results are shown in Figure 7 for the case using purely an MPI version of the code. These include four configurations that are similar to the XT4 placements. Each node has 4 MPI tasks and each task is either on an individual hex-core die (N4S1) or they are sharing with another MPI task (N4S2) or all four tasks are fully packed onto one hex-core die. This is an unlikely scenario as that would leave 20 unused cores and in the latter case all four are fighting the local memory bus to get to the other 24Gb of RAM. The fourth similar placement is N16S4 as that brings twelve more MPI tasks onto the node but spreads them out evenly among the hex-core dies so each MPI task is sharing 8Gb of local RAM with three other MPI tasks.

Figure 7: MPI only tests to show importance of task placement without Open MP influence.



6. Conclusion

The GLOMAP Mode MPI program has been enhanced with Open MP directives. The code has been successfully tested on XT4h and XT6. The increases in performance have been demonstrated. Mixed-mode processing is more expensive than continuing to operate in pure MPI mode. However if the turnaround time is considered then it becomes more attractive. The simulation performs better when there are fewer MPI tasks per node. However, there is an increasing overhead for when there are more MPI tasks due to the communication characteristics.

A longer term view is for more efficient use of the whole node and a hybrid placement will ensure that all reserved cores are used towards the simulation. The work on the XT6 has shown that for this simulation (with 64 MPI tasks) placing 8 MPI tasks per node is very effective. It allows three Open MP threads per task to be used so that all cores are being used to compute the simulation.

- The cost of using more nodes is recovered almost fully
  - through the increased speed of each MPI task
  - by multi-threaded acceleration
  - by reducing the number of intra-node MPI tasks
- Placement of the tasks on a "many-core" system is critical to performance
- The shorter run-times will allow more research to be done, in particular the standard 15 month simulation may be changed to be 5-year simulations
- Fewer MPI tasks per node will allow higher resolution simulations
- GLOMAP Mode MPI will be ready for better use of XT6

Acknowledgments

The authors wish to thank the following for their support and assistance during this project: HECToR - a Research Councils UK High End Computing Service, HECToR DCSE programme and Professor Martyn Chipperfield for detailed discussion on and changes to TOMCAT.

About the Authors

Mark Richardson is a Technical Consultant with NAG Ltd. He is involved in investigation, support and training for the HECToR service. He can be contacted at the Manchester Offices of NAG by email mark.richardson@nag.co.uk. Graham Mann is a NCAS funded researcher at the University of Leeds. He can be contacted at gmann@env.leeds.ac.uk .