Benchmarking of Medium Range Weather Forecasting Model on PARAM - A parallel machine

Akshara Kaginalkar and Sharad Purohit
Centre for Development of Advanced Computing (C-DAC)
Pune University Campus, Pune 4100 007, India.

Abstract

This paper describes the parallel processing efforts for medium range weather forecasting being pursued at the Centre for Development of Advanced Computing (C-DAC), India. The global spectral T80L18 model has been parallelised using latitude wise data decomposition on distributed memory parallel machine PARAM. The performance of parallel T80 for one day forecast on various platforms is discussed. The compute/communication optimisation issues are studied.

1 Introduction

Numerical weather prediction has emerged as one of the important discipline requiring increasing computing power. To have accurate timely forecasts, state-of-art computers are used all over the world. Climate and weather modelers were among the first users of parallel computers. In parallel processing different parts of application code are computed at the same time by different processors working in parallel. There are World wide efforts to port weather forecasting codes on parallel computers. Leading operational centres like ECMWF(European centre of Medium Range Weather Forecasting, UK) and NCEP(National Centre for Environmental Prediction, USA) are experimenting with parallel processing in numerical weather Prediction. This includes parallelisation of the IFS (Integrated Forecasting System) of ECMWF in collaboration with GMD(German National centre for computer Science)[1] , NMC's parallel spectral model[2] and CHAMMP(Computer hardware, advanced mathematics and model physics) program[3] sponsored by DOE (Department of Energy, USA) on various supercomputers like CRAY T3D, Intel Paragon, IBM SP2, Meiko etc. In order to contribute in this new research endeavour in high performance computing, Department of Science and Technology, Government of India initiated the experiment of cost-effective computing solution to the weather forecasting models. A major step was taken to quantify the induction of parallel processing in the Indian context. Thus a project to port and parallelise the global prediction model T80L18 of NCMRWF (National Centre for Medium Range Weather Forecasting, New Delhi, India) was started with Centre for Development of Advanced Computing, Pune, India[4].

In this paper we describe the performance of medium range weather forecast code T80 on a distributed memory parallel machine - PARAM. The prime objective is to explore the parallelism for the spectral method in T80 code and to analyse its performance. The T80L18 model has triangular truncation and a spectral resolution of 80 waves. The parallel strategy involves data decomposition across the latitudes of the earth where independent computation for FFT, Legendre transform and grid calculations are done across the latitudes distributed over various processors.

In this paper, reproducibility of results, efficiency, compute/communication optimisation issues in parallelisation are discussed. It is demonstrated that the performance of parallel T80 for one day forecast outperforms the elapsed time for the same on CRAY XMP/216 at NCMRWF. Upgradation of Indian forecasting model to finer resolution codes like T213L31 or T126L28 on the C-DAC's parallel machine with the complete solution consisting of decoder, analysis, forecast and post-processing is proposed.

2 Global Spectral Model

Governing equations for global spectral weather model are derived from the conservation laws of mass, momentum, and energy. Vorticity, divergence, temperature, surface pressure and moisture equations are the main constituents of it[5]. Expansion of the global field is done using spherical harmonics. Finite difference operators are used to approximate the derivatives in the vertical direction and a semi-implicit time integration scheme is applied to the coupled equations of divergence, temperature and surface pressure. This is accomplished by applying a central difference scheme to the time derivative. For the non-linear terms calculation a transform method is used to convert the spectral quantities to the grid-point values in the physical space. They are transformed back to the spectral space after the computation[6,7].

The computations for spectral algorithms are performed in three discrete functional spaces the grid point domain, the Fourier domain, and the spectral domain. Linear terms like horizontal gradient are calculated in spectral space and non-linear terms are evaluated in the physical space. Physics calculation has radiation, surface physics gravity wave drag, kuo convection. Most of the computation is either in spectral or grid space and the variables are transformed between them. The time integration and the vertical integration take place in the spectral domain. Coefficients of expansion called the spectral coefficients are evaluated from the known values of the function in the physical space by Fourier and Legendre transforms.

2.1 The Algorithm

The T80 code has spectral resolution of 80 waves and 128 latitudes and 256 longitudes with 18 varying pressure levels in the vertical direction. Associated Legendre polynomials are evaluated at each latitude. Inner sum in the step 2 is calculated in the Legendre transform. Due to the symmetry of spectral transform, latitudes from northern and southern hemispheres are paired. At different stages of the algorithm the data dependencies are in different directions like FFT calculation is latitude dependent, whereas the Legendre transform depend on longitudes and in spectral space the calculation depends on wave number m.

Each time step of the algorithm consists of the following steps

- step 1: Input spectral coefficients $u_n^m(Z_k)$ for all m, n and k.
- step 2: Compute Fourier coefficients using inverse Legendre's transform

  $$u^m(\mu_j, Z_k) = \sum_{n=|m|}^{M} u_n^m(Z_k) P_n^m(\mu_j)$$

  for all m, j and k. P_n^m is the Legendre polynomial of degree n and order m.

  Step 3: Compute Gaussian grid point values $u(\lambda_l, \mu_j, Z_k)$ using the inverse Fourier transform

  $$u(\lambda_l, \mu_j, Z_k) = \sum_{m=-M}^{M} u(\mu_j, Z_k) e^{im\lambda_l}$$

  for all l, j and k.

- step 4: Compute non-linear terms and physics in grid point domain on Gaussian grid. Update $u(\lambda_l, \mu_j, Z_k)$ for all i, j and k.
- step 5: Compute Fourier coefficients $u^m(\mu_j, Z_k)$ using the direct Fourier transform.

  $$u^m(\mu_j, Z_k) = \frac{1}{K_2} \sum_{l=0}^{K_2-1} u(\lambda_l, \mu_j, Z_k) e^{-im\lambda_l}$$

- step 6: Compute spectral coefficients by direct Legendre's transform.

  $$u_n^m(Z_k) = \sum_{j=1}^{K_1} \omega(\mu_j) u^m(\mu_j, Z_k) P_n^m(\mu_j)$$

  for all m, n and k, where $\omega(\mu_j)$ are the gaussian weights.

- step 7: perform calculations in spectral domain.

where

M	- truncation number and equals to 80
K_0	- number of vertical levels and equals to 18
K_1	- number of latitudes and equals to 128
K_2	- number of longitudes and equals to 256
$0 \leq m, n \leq M$	- spectral indices and their ranges
μ	- Gaussian latitude
$1 \leq j \leq K_1$	- latitude index and its range
λ	- longitude
$0 \leq l \leq K_2 - 1$	- longitude index and its range
Z	- vertical level
$1 \leq k \leq K_0$	- level index and its range

3 PARAM - A parallel machine

The parallel T80 code is ported successfully on various platforms of the PARAM series (Table 1), based on distributed memory parallel architecture. The initial sequential porting of the T80 code was done on single i860 processor with 64 Mbyte as main memory of the PARAM8600[8]. In addition to the compute processor i860, PARAM8600 has another microprocessor called the transputer for communication. Each transputer has links of 20 Mbits/sec speed. A 54 Mbyte /sec high speed bus called DRE(data restructuring engine) between a transputer and an i860 processor. For any i860 node the bandwidth to the external world is 320 Mbits/sec. It has 16 i860 processors and 64 transputers. The original CRAY code had asynchronous I/O related operations which were replaced by the PARAS special calls on PARAM8600. Later for faster execution these calls were modified to standard Fortran arrays using local memory.

Then the parallel T80 code was ported on next machine in the PARAM series the PARAM9000SS, the SUN SuperSparc based machine with 64 Mbyte main memory for each processor. The operating system is SOLARIS 2.5. The architecture is built around an interconnection network(CCP network) based on components complying to the IEEE1355 standard. PARAM9000/AA, has DEC ALPHA processor as compute node and networked by FDDI (Fibre distributed data interface) links. OSF is the operating environment on each of the nodes.

And very recently the parallel T80 on PARAM Openframe has been made available. The PARAM Openframe has SUN UltraSparc processor as main compute en-

Table 1: The PARAM series

Machine Name	Processor
PARAM8600	Intel i860
PARAM9000SS	SUN SuperSparc
PARAM9000AA	DEC ALPHA
PARAM Openframe	SUN UltraSparc

gine. The UltraSparc processor is based on SPARCV9 64-bit RISC architecture with SOLARIS 2.5.1. Fast ethernet with link speed of 100 Mbits/sec as well as Myrinet from Myricom(USA) with link speed of 1.2 Gbits/sec are the communication networks for the 8 Dual CPU SMP (Symmetric Multiprocessor) UltraSparcs. PARAM Openframe is a combination of shared memory and distributed memory architecture.

PARAS is the in-house developed message passing communication software which has a micro kernel running on all the processors of PARAM8600. The communication software for PARAM9000/SS, PARAM9000/AA, and PARAM Openframe is standard message passing interface MPI/PVM.

4 Parallelisation Strategy

The main task before deciding the parallelisation strategy of the code was to identify the nature of parallelism involved in the code and subsequent compute intensive parts and their data dependencies. There are many independent variables namely latitudes, longitudes, vertical levels and spectral indices for data decomposition. The most obvious and clear data independent compute intensive work is across the latitudes.

In this strategy of latitude wise data decomposition, data for latitudes is available with each processor. A pair of latitudes is placed on one processor. In spectral models major computation is done in Fourier and Legendre transform. The transforms are performed twice for physics and dynamics part of the code. In order to minimise the communication traffic, these transforms should be performed on each processor sequentially. With latitude distribution, FFT can be performed independently without any communication on each processor, but this leads to parallel Legendre transform (the most time consuming) where partial sums are performed on each processor. By this method the original algorithmic component of the sequential code remains the same.

Initially, an equal number of latitudes are distributed over the processors and accordingly data decomposition takes place. Each processor computes Fourier coefficients using the inverse Legendre transform and the gaussian grid point values using the inverse Fourier transform. Non-linear terms and physics in grid point domain calculations are done simultaneously on each processor. Then each processor computes Fourier coefficients only for those latitudes which are assigned to it. Partial sum of the Legendre transform is performed on each processor for given latitudes. Then these sums are circulated among all the processors to have global sum on each of the processor.

4.1 The Implementation

This parallelisation strategy was implemented for the T80L18 model. The original sequential code has many CRAY dependent routines written in CRAY assembly language (CAL). The CAL FFT was replaced by the Temperton's[9] Fortran algorithm. The important components of this code are the I/O, the forecast loop(GLOOPA), the radiation loop (GLOOPR) which is called once in twelve hours and the physics loop (GLOOPB), FFT and Legendre transform. GLOOPA and GLOOPB are called once in each time step of 15 minutes for the 24 hours forecast. The data distribution of parallel T80 is along the latitudes distributed over the processors. This is a master - workers model (Figure 1). The initial sigma and surface data is distributed along the latitudes to all the workers by the master. Since the original flow of the code has remained unchanged, similar computation is performed for given latitudes on all the processors. Each worker independently works for the radiation calculation and at the end of it master collects the radiation output from all the workers. Then master interpolates this data from radiation grid to a standard grid and is distributed back to all the workers. The dynamics and physics calculations are done simultaneously on all the workers. Global sum of the partial sums in Legendre transform for both dynamics and physics is the major communication for each time step for one day forecast. At the end the final surface and sigma forecast results are collected by the master from all the workers for zonal diagnostics and for the final output.

In this implementation the algorithmic component of the forecast code remains intact, only the loop indices for latitude computation are changed. Another feature of this implementation is that separate communication library for data movements. Depending on the architecture and the underlying communication software, appropriate communication primitives are to be attached. On PARAM8600, the communication library is PARAS and on PARAM9000/SS, PARAM9000/AA and PARAM Openframe, the communication library is written in generalised portable communication software MPI. Global reduction calls of MPI are used for the global sum calculation in the Legendre transform. Typical communication pattern is shown in (Figure 2). This full Fortran parallel T80 MPI version can be ported across any platform with distributed memory architecture.

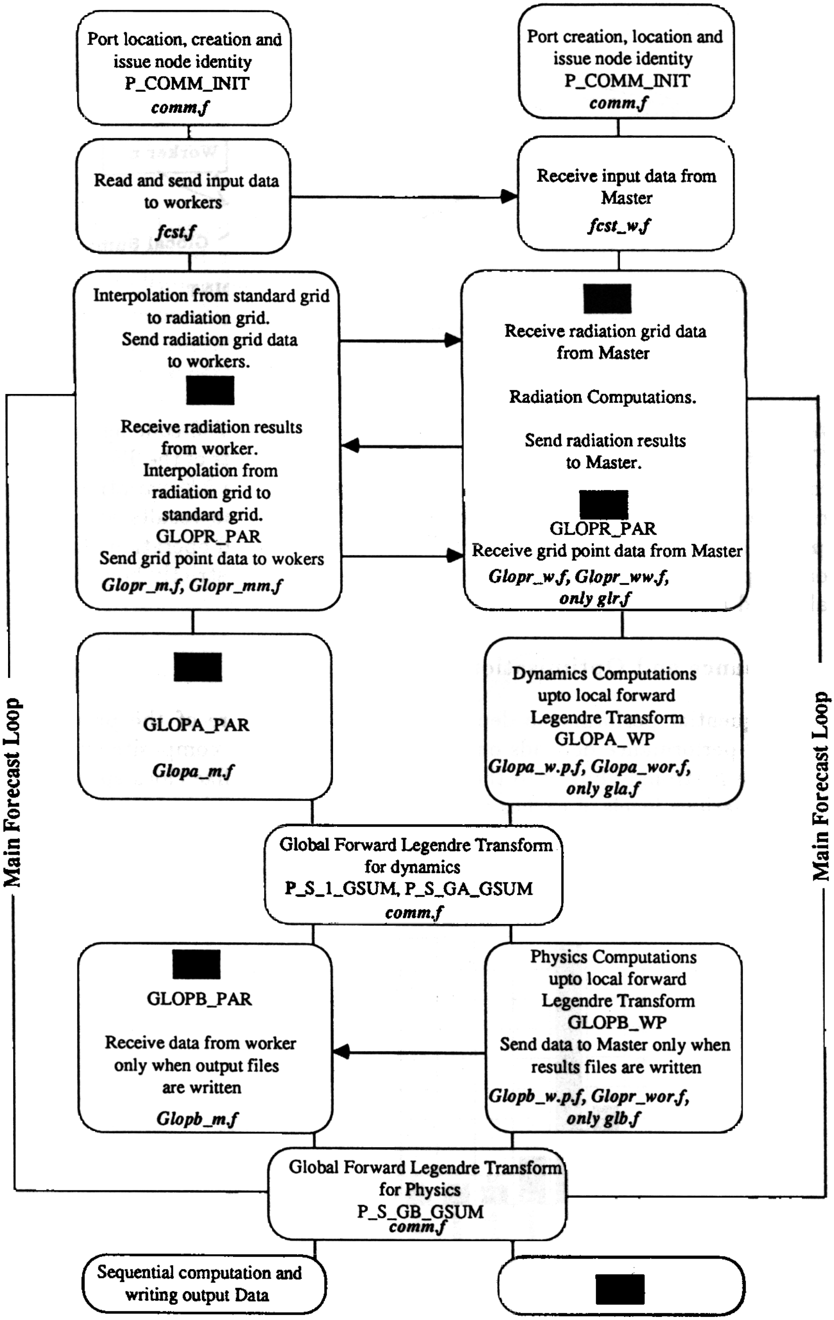
Port location, creation and issue node identity
P_COMM_INIT
comm.f
Port creation, location and issue node identity
P_COMM_INIT
comm.f
Read and send input data to workers
fcst.f
Receive input data from Master
fcst_w.f
Interpolation from standard grid to radiation grid.
Send radiation grid data to workers.
Receive radiation results from worker.
Interpolation from radiation grid to standard grid.
GLOPR_PAR
Send grid point data to wokers
Glopr_m.f, Glopr_mm.f
Receive radiation grid data from Master
Radiation Computations.
Send radiation results to Master.
GLOPR_PAR
Receive grid point data from Master
Glopr_w.f, Glopr_ww.f, only glr.f
Main Forecast Loop
Main Forecast Loop
GLOPA_PAR
Glopa_m.f
Dynamics Computations upto local forward Legendre Transform
GLOPA_WP
Glopa_w.p.f, Glopa_wor.f, only gla.f
Global Forward Legendre Transform for dynamics
P_S_1_GSUM, P_S_GA_GSUM
comm.f
GLOPB_PAR
Receive data from worker only when output files are written
Glopb_m.f
Physics Computations upto local forward Legendre Transform
GLOPB_WP
Send data to Master only when results files are written
Glopb_w.p.f, Glopr_wor.f, only glb.f
Global Forward Legendre Transform for Physics
P_S_GB_GSUM
comm.f
Sequential computation and writing output Data

Figure 1 - Parallel T80 flowchart

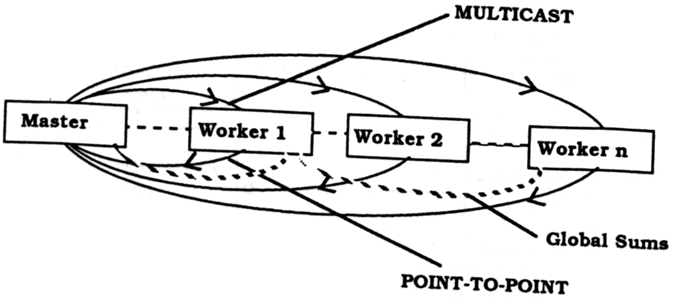

Figure 2 - Typical communication pattern

5 Results

We have evaluated the scientific accuracy of the parallel T80 code against that of the CRAY forecast output. The partial double precision results of the T80 on PARAM approximate the full double precision results of CRAY within 5 % variation. The two main reasons for the variation in the reproducibility of the results are change of computing system and non-associativity[10,11] of mathematical operations (truncation of floating point) on computers. However the results remain strictly bounded for any variable and remain well within the meteorological accuracy.

6 Performance and Optimisation

The sequential performance depends on the architecture of the processor and the parallel performance depends on the architecture of the composite system. The performance is strongly linked to the architectural features like clock speed, number and size of various levels of caches and main memory bandwidth.

In order to optimise the compute portions of the T80 code, we first profiled the code and studied the most time-consuming portion in the latitude loop i.e Legendre, inverse Legendre transform and FFT (Figure 3).

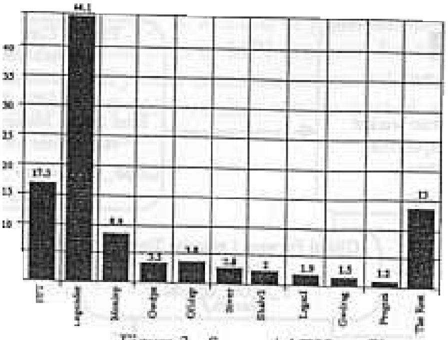

Figure 3 - Sequential T80 profile

Table 2: Sequential Performance of T80 for one day forecast on PARAM8600

Number of nodes	Optimisation	Time(minutes)
1	without optimisation	436
1	with cache utilisation for Legendre transform	280

We tuned the Legendre transform to use on-chip data cache of i860, where the vectors that are reused are retained. We replaced the core time consuming loop i.e matrix vector multiplication of Legendre transform by i860 assembly function 'zxpy'[12], which will use the on-chip cache registers. This is done without changing the Fortran loop structure and considerable time improvement was possible (Table 2).

Another improvement in the overall performance is by optimising the collective operations. Enhancing the conventional tree algorithm used in MPI to hypercube algorithm considering the topology requirement of the application with suitability of the architecture proved to be the best for 8 processor version of parallel T80. As a next level of optimisation specific to the 8 node dual-CPU SMP cluster, a modified hypercube algorithm[13] (Figure 4) is used with local processor on an SMP node communicating using shared memory and remote processes communicating via TCP/IP sockets.

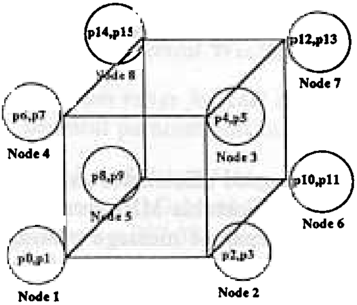

Figure 4 - Extended hypercube topology

We have experimented with different communication networks like fast ethernet, myrinet in the PARAM Openframe for the communication in parallel T80 code. Communication using Myrinet as network is faster than fast ethernet for same number of processors (Table 3).

Table 3: Performance of parallel T80 on various platforms of PARAM series

Platform	Nodes	Remarks	Time (minutes)
PARAM8600	1	Legendre transform optimisation	280
PARAM8600	16	Legendre transform optimisation	60
PARAM9000SS	8	CCP network	36
PARAM9000SS	8	Hypercube topology for communication	27
PARAM9000AA	6	FDDI network	20
PARAM Openframe	8	Fast ethernet	11
PARAM Openframe	8	Myrinet	9
PARAM Openframe	8	Myrinet + global communication optimisation	8
PARAM Openframe	16 (8 2-CPU SMP)	Myrinet + extended hypercube communication	7

Thus the performance of parallel T80 has varied from 280 minutes on single i860 for one day forecast to 7 minutes on PARAM Openframe with 8 dual CPU UltraSparc processors. This outperforms the existing performance of 15 minutes for the same on CRAY-XMP/216 at NCMRWF.

7 Conclusion

The parallel T80 performs with good efficiencies on MIMD type distributed memory supercomputer - PARAM. This portable MPI version has been tested on all the platforms of the PARAM series. Separate message passing library for the communication and data distribution is an added advantage for user friendly full Fortran parallel T80, where the original components of the spectral algorithm are intact. This model is scalable over number of processors.

The partial double precision results of parallel T80 code on PARAM scientifically approximate the full double-precision results of T80 on CRAY-XMP/216. Thus the parallel T80 code is easily portable with no specialised architectural dependent functionalities. Also it is easy for the future developments and maintenance from the software engineering point of view.

Further research is on to exploit fast global communication routines to reduce communication overheads as the number of processors added. Exploration of faster intrinsic functions and better utilisation cache of UltraSparc is in progress.

With the confidence gained by this porting, C-DAC along with NCMRWF has initiated to upgrade the existing weather forecasting codes with the better resolution models like T213L31 or T126L28. The activities to have complete parallel system for weather forecast with decoder, analysis and post-processing are in progress.

8 Acknowledgement

We wish to thank Dr.S.K.Mishra, Director General, National Centre for Medium Range Weather Forecasting(NCMRWF) and Dr. B.K.Basu, Director, NCMRWF, for useful suggestions and help during the successful implementation of this project at C-DAC. The interaction with our colleagues Dr. P.S.Narayanan and Dr. T.V. Singh was very fruitful and is hereby acknowledged.

9 References

1. S.R.M. Barros, et. al., *The IFS model : A parallel production weather code*, Parallel Computing, vol.21(1995), No.10, p. 1621.

2. J.G. Sela, *Weather forecasting on parallel architectures*, Parallel Computing, vol.21(1995), No.10, p. 1639.

3. J.Drake and I.Foster, *Design and performance of a scalable parallel community climate model*, Parallel Computing, vol.21(1995), No.10, p. 1571.

4. NCMRWF/DST Project, *Numerical Weather Prediction on PARAM*,1993

5. *Research version of Medium range forecast model (1988)*, Documentation - vol. 1, Hydrodynamics, physical parametrization and user's guide.

6. W. Bourke, et.al., *Global modelling of Atmospheric flow by spectral methods in computational physics*, vol. 7 : General Circulation modes of Atmosphere, ed. J. Chang. Academic Press, p 267-324.

7. S.C Purohit, P.S.Narayanan, T.V.Singh and A.Kaginalkar, *Development and implementation of parallel numerical weather prediction on PARAM*, progress report, (1994).

8. C-DAC technical reports,SSG-TR-1-93,(1993)

9. C.Temperton, *Selfsorting mixed radix Fast Fourier transform*, Journal of computational physics 52, 1-23,(1983).

10. D.H.Bailey, *RISC Microprocessors and Scientific computing*, Conference proceedings.

11. J.Drake, et. al., *PCCM2: A GCM Adapted for scalable parallel computers*, Math and Computer Sci. Division Argonne National Laboratory, (1993).

12. S.C. Purohit, et. al., *Global spectral medium range weather forecasting model on PARAM*, Supercomputer, Vol. XII, No. 3, (1996), p.27

13. C-DAC technical reports,SSG-TR-1-96, (1996).