ALGORITHM COMPARISON AND BENCHMARKING USING A PARALLEL SPECTRAL TRANSFORM SHALLOW WATER MODEL*

Pat Worley

Mathematical Sciences Section
Oak Ridge National Laboratory
P.O. Box 2008, Bldg. 6012
Oak Ridge, TN 37831-6367, U.S.A.

and

Ian Foster and Brian Toonen

Mathematics and Computer Science Division
Argonne National Laboratory
Argonne, IL 60439, U.S.A.

1. Introduction

In recent years, a number of computer vendors have produced supercomputers based on a massively parallel processing (MPP) architecture. These computers have been shown to be competitive in performance with conventional vector supercomputers for some applications. As spectral weather and climate models are heavy users of vector supercomputers, it is interesting to determine how these models perform on MPPs, and which MPPs are best suited to the execution of spectral models.

The benchmarking of MPPs is complicated by the fact that different algorithms may be more efficient on different architectures. Hence, a comprehensive benchmarking effort must answer two related questions: which algorithm is most efficient on each computer and how do the most efficient algorithms compare on different computers. In general, these are difficult questions to answer because of the high cost associated with implementing and evaluating a range of different parallel algorithms on each MPP platform.

In a recent study, we developed a testbed code called PSTSWM that incorporated a wide range of parallel spectral transform algorithms (Worley and Foster, 1994). Studies with this testbed confirm that the performance of different algorithms can

*This research was supported by the Atmospheric and Climate Research Division of the Office of Energy Research, U.S. Department of Energy, under Contracts DE-AC05-84OR21400 and W-31-109-Eng-38.

vary significantly from computer to computer, and that there is no single algorithm that is optimal on all platforms (Foster and Worley, 1994). Availability of this testbed makes it feasible to perform a comprehensive and fair benchmarking exercise of MPP platforms for spectral transform codes. We report the results of this exercise in this paper, presenting benchmark results for the Intel Paragon, the Cray T3D, and the IBM SP1.

2. Method

We first outline the structure of the PSTSWM testbed code, the experiments that were performed during the benchmarking exercise, and the computers on which benchmarks were performed.

4.1 *The PSTSWM Testbed*

A number of researchers have investigated parallel algorithms for the spectral transform algorithm used in atmospheric circulation models. A variety of different parallel algorithms have been proposed, but until recently no detailed quantitative comparisons of the different approaches had been attempted. (However, some qualitative comparisons have been reported (Foster and Worley, 1993; Kauranne and Barros, 1993).) To permit a fair comparison of the suitability of the various algorithms, we have incorporated them into a single testbed code called PSTSWM (for parallel spectral transform shallow water model). PSTSWM is a message-passing parallel implementation of the sequential Fortran code STSWM 2.0 (Hack and Jacob, 1992). STSWM uses the spectral transform method to solve the nonlinear shallow water equations on a rotating sphere; its data structures and implementation are based directly on equivalent structures and algorithms in CCM2.

PSTSWM differs from STSWM in one major respect: vertical levels have been added to permit a fair evaluation of parallel algorithms based on matrix transposes. This is necessary because in a one-layer model, a transpose algorithm reduces to a one-dimensional decomposition of each grid and hence can utilize only a small number of processors. The addition of vertical levels also has the advantage of modeling more accurately the granularity of the dynamics computation in atmospheric model. In all other respects we have changed the algorithmic aspects of STSWM as little as possible. In particular, we did not change loop and array index ordering. Although such changes would probably improve performance of some algorithms, our goal was to have a code as similar to a real atmospheric model as possible.

PSTSWM is structured so that a variety of different algorithms can be selected by runtime parameters (Foster and Worley, 1994). Both the FFT and LT can be computed using several different distributed algorithms (in which data structures are decomposed over processors and communication is performed as required by the FFT or LT) and transpose algorithms (in which data is explicitly reorganized prior to an FFT or LT to avoid a need for communication during the transform. PSTSWM also incorporates variants of these basic algorithms, such as a two-block distributed FFT permits that computation/communication overlap. Additional parameters select a range of variants of each of these major algorithms (Foster and Worley, 1994; Worley and Foster, 1994). All parallel algorithms were carefully implemented, eliminating unnecessary buffer copying and exploiting our knowledge of the context in which they are called. At the present time, this allows us to achieve better performance than can be achieved by calling available vendor-supplied routines. Hence, it provides a fairer test of the parallel algorithms.

4.1 *Experimental Technique*

We performed experiments for a range of horizontal and vertical resolutions, as summarized in Table 1. (Horizontal resolution is specified in terms of both spectral truncation and physical grid size.) This range of resolutions was considered necessary firstly because there is little agreement on standard resolutions and secondly because the performance of different parallel algorithms can vary significantly depending on whether (for example) the number of vertical levels is a power of two. The highest vertical resolutions in the T42 and T85 models are intended to be representative of resolutions used in stratospheric models. In the experiments reported in this paper, we restrict horizontal resolutions to be powers of two; in later studies, we will expand the set of experiments to include other common resolutions such as T63 and T213.

All experiments involved a five day simulation using the performance benchmark proposed by Williamson et al. (1992): global steady state nonlinear zonal geostrophic flow. The number of timesteps (Table 1) assumes a Courant number of 0.5. In practice, a timestep twice as large could be used without losing stability, halving the execution time, but at the cost of some degradation in the solution accuracy for the larger model resolutions. We report raw execution times, Gflops rates, and Mflops achieved per processor. The Gflops rates are derived from floating point operation counts obtained using the hardware performance monitor on a Cray Y/MP. Mflops/processor is a useful measure of the scalability of a parallel computer for an application.

Experiments were performed in two stages. In the first stage, a set of tuning experiments were performed to determine the most efficient algorithms and algorithmic variants on each computer and at each resolution. These experiments were very detailed, involving around 400 separate measurements on each computer. Based on the results of these experiments, we selected the algorithms to be used in the benchmarking proper.

In the second stage, we measured execution times on each computer and for each

TABLE 1
Problem Sizes Considered in Empirical Studies, Floating Point Operations per Vertical Level, and Number of Timesteps for a 5-Day Simulation

Horizontal	Grid	Vertical Levels	Flops/level/step	Steps
T42	64×128	L16, L18, L44	4129859	222
T85	128×256	L16, L18, L24, L36, L60	24235477	446
T170	256×512	L18, L24, L32, L36, L48	153014243	891

resolution listed in Table 1. Experiments were performed using both the optimal algorithms identified in the first round of experiments.

4.1 *Target Computers*

We performed experiments on the five parallel computer systems listed in Table 2. These systems have similar architectures and programming models, but vary considerably in their communication and computational capabilities. Our values for message startup time (t_s) and per-byte transfer time (t_b) differ from those reported by most researchers because we measure the time required to swap words between two processors rather than the time to send bytes from a source to a destination. Also, the computational rate (Mflops/sec) is measured by running PSTSWM on a single node at high resolution and so is an achieved rather than a peak rate. The Paragon used the SUNMOS operating system developed at Sandia National Laboratories and the University of New Mexico, which provides higher communication performance than the standard Intel operating system. The SP1 used in these studies is a nonstandard system incorporating the faster SP2 interconnection network. Communications were performed using MPI-f version 1.3.3., an experimental implementation of the MPI message-passing standard developed and made available to us by Hubertus Franke of IBM Yorktown. Experiments were performed using single-precision arithmetic (32-bit floating point values) except on the T3D, where only double precision (64-bit) is supported. In future work, we will evaluate the performance impact of using double precision arithmetic on other machines. On the SP, the computation rate should stay about the same, while communication costs will of course increase; on the Paragon, the computation rate can be expected to decrease.[*][†]

3. Results

The results of the experiments are summarized in Figures 1–9. These figures show results on the different machines at T42, T85, and T170 resolution, expressed

[*]Need to add OS details for T3D in table.
[†]Need to change t_w values to t_b values. Also need to incorporate SP1 COMMTEST values.

TABLE 2

Parallel Computers Used in Empirical Studies, Characterized by Operating System Version, Microprocessor, Interconnection Network, Maximum Machine Size Used in Experiments, Message Startup Cost (t_s), Per-Byte Transfer Cost (t_b), and Per-Processor Mflops/Sec

Name	OS	Processor	Network	P
Paragon	SUNMOS 1.6.1	i860SP	16×100 mesh	1600
SP1	AIX + MPI-f	Power 1	multistage crossbar	128
T3D	??	Alpha	$16 \times 4 \times 4$ mesh	256

Name	t_s (μsec)	t_b (μsec)	MB/sec	MFlops/sec
Paragon	72	0.03	141	11.6
SP1	49.5	0.13	30.5	25.6
T3D	18	0.05	84	18.2

in terms of both execution time for a 5-day forecast, execution rate in Gflops/sec, and megaflops achieved per processor, all as a function of processor count. Notice the use of a log scale in the X axis. In the figure keys, P represents Paragon, T represents T3D, and S1 represents SP1. On the Paragon, data were obtained on 128, 256, 512, 1024, and 1600 processors. On the T3D, data were obtained on 64, 128, and 256 processors. On the SP, data were obtained only on 128 processors. Some problem sizes did not fit on a small number of processors on the T3D, and hence are missing. SP1 data were obtained only at T42 and T85 resolution.

4. Discussion

Figures 1–9 present a large amount of data from which one can derive many interesting conclusions about both the spectral transform method and the performance of the T3D, Paragon, and SP. For example:

- In many cases, performance is significantly better when the number of vertical levels is a power of 2. This phenomonen is to some extent an artifact of the fact that the number of processors and the physical grid size are both powers of 2.

- It appears that the T3D is roughly 75 percent faster than Paragon with the same number of processors across all problem sizes. This result is not surprising, given that T3D serial times are one half Paragon serial times.

- Paragon performance is roughly comparable to T3D performance on one-half as many processors, across a wide range of problem sizes. As PSTSWM's

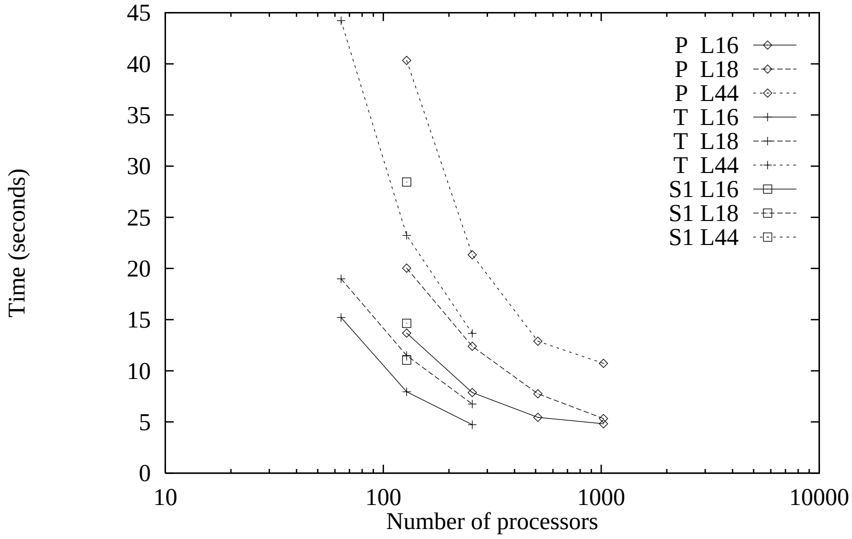

FIG. 1. *Execution times for 5-day forecast at T42 resolution.*

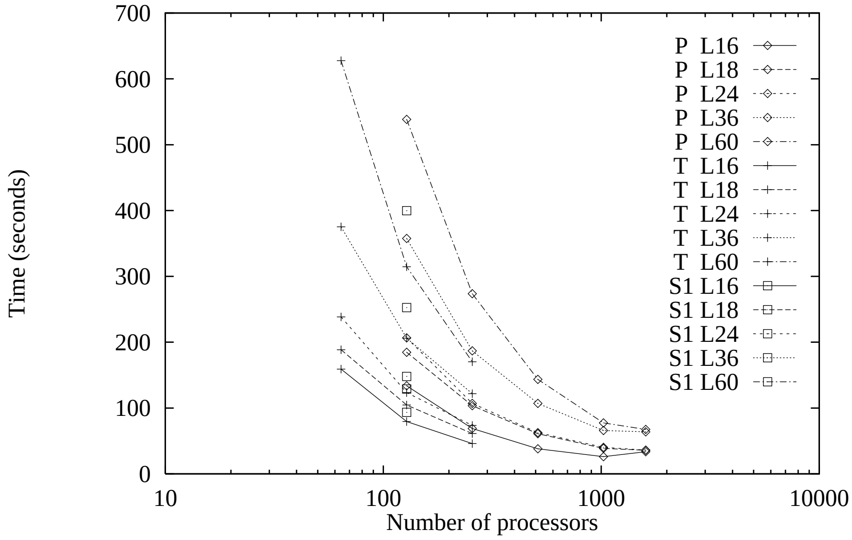

FIG. 2. *Execution times for 5-day forecast at T85 resolution.*

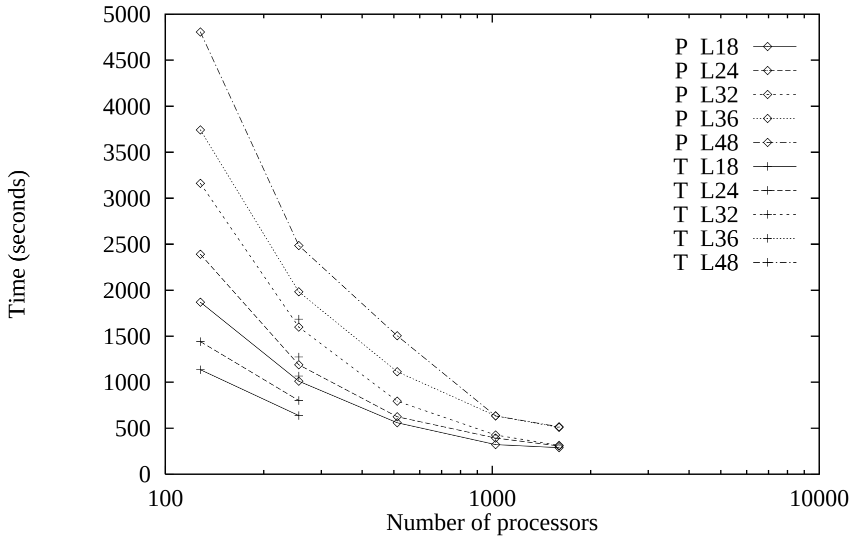

FIG. 3. *Execution times for 5-day forecast at T170 resolution.*

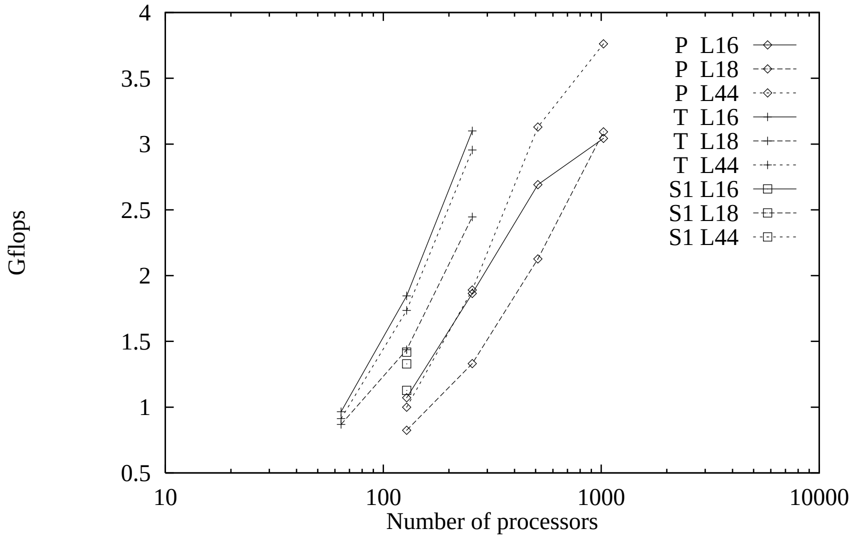

FIG. 4. *Execution rate for 5-day forecast at T42 resolution.*

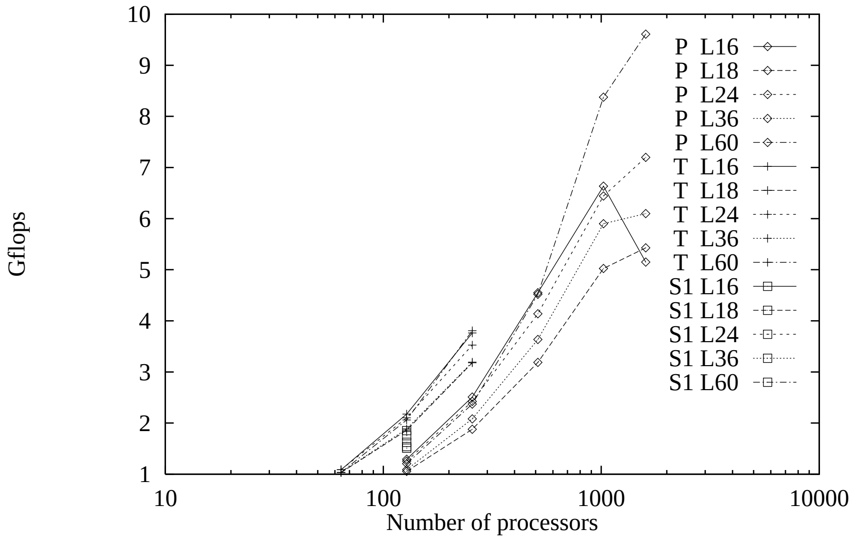

FIG. 5. *Execution rate for 5-day forecast at T85 resolution.*

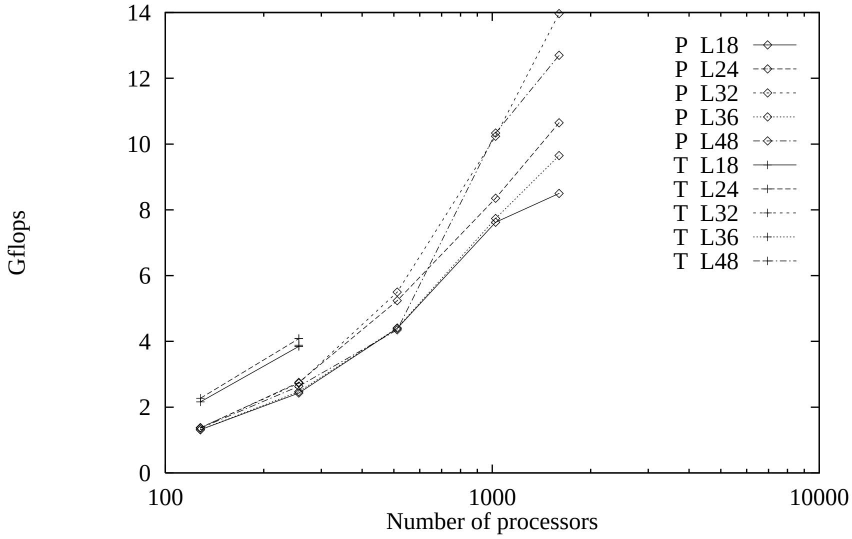

FIG. 6. *Execution rate for 5-day forecast at T170 resolution.*

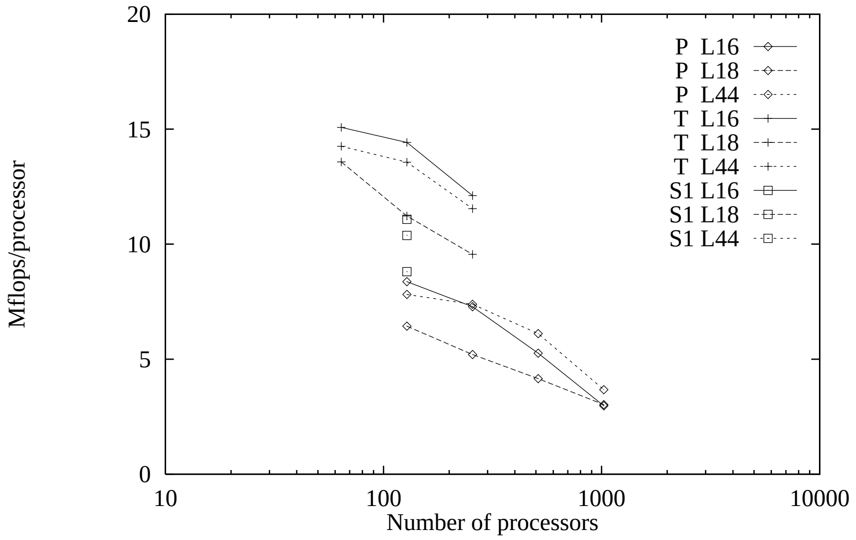

Fig. 7. *Processor performance for 5-day forecast at T42 resolution.*

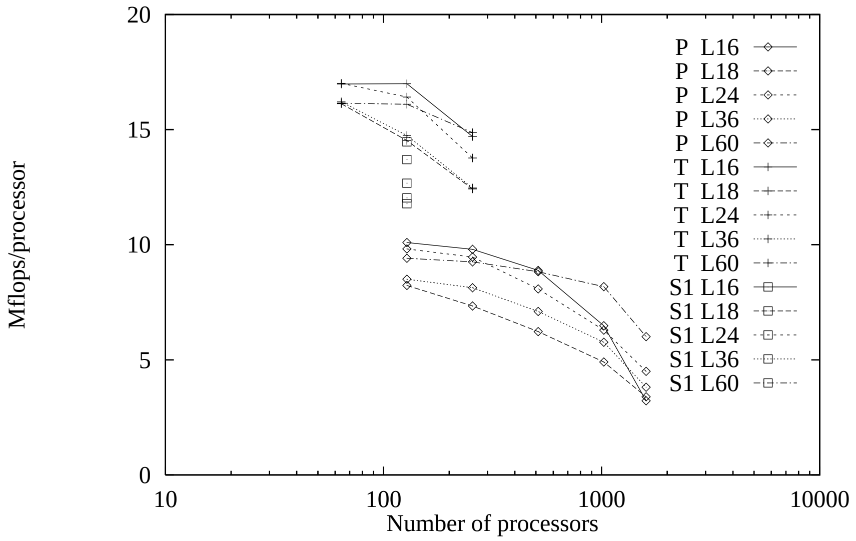

Fig. 8. *Processor performance for 5-day forecast at T85 resolution.*

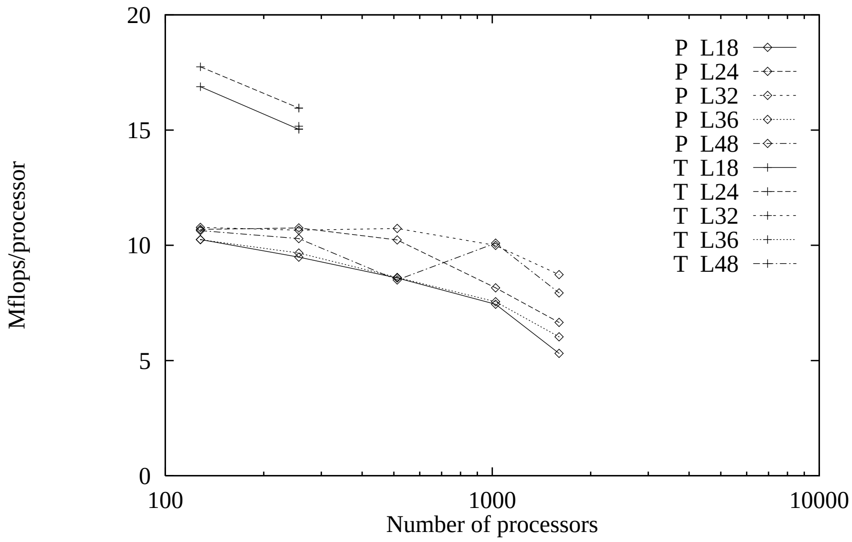

FIG. 9. *Processor performance for 5-day forecast at T170 resolution.*

communication-computation ratio increases with the number of processors, this result indicates that the Paragon interconnect performs significantly better than the T3D interconnect.

- Cache performance ...‡

- Some preliminary studies indicate that PSTSWM performance on both the T3D and Paragon is very dependent on loop structure in computationally intensive inner loops. For example, a simple reordering of one loop resulted in a speedup of 20 percent on the T3D; a similar reordering reduced performance by 30 percent on the Paragon. Completely fair performance studies probably require the use of machine-dependent libraries for selected computational routines (such as FFT), just as we currently use machine-dependent communication libraries.

Note that we have not attempted to compare the price-performance (performance per unit of capital investment) of the different machines, due to the difficulty of obtaining accurate price data, and its dependence on nontechnical factors. However, this information must clearly be taken into account when interpreting the results of this study.

‡Complete or delete

5. Conclusions

The results of this study indicate that massively parallel computers such as the Paragon, T3D, and SP are indeed capable of multiGigaflops performance, even on communication-intensive applications such as the spectral transform method. However, performance depends on a wide variety of factors, including parallel algorithm, communication protocols, ...[§]

In future work, we plan to evaluate the IBM SP2 achitecture (which has a processor twice as fast as the SP1 used in these studied) and also to run our benchmarks on larger SP and T3D configurations. We also plan to evaluate performance on other problem sizes, in particular T63 and T213. In a related study, we are evaluating the feasibility of calibrating analytic performance models (Foster and Worley, 1994) and then using these performance models to predict performance on future architectures.

Acknowledgements

We are grateful to members of the CHAMMP Interagency Organization for Numerical Simulation, a collaboration involving Argonne National Laboratory, the National Center for Atmospheric Research, and Oak Ridge National Laboratory, for sharing codes and results; to Hubertus Franke of IBM for providing his MPI-f library on the SP; and to the SUNMOS development team for help in getting PSTSWM running under SUNMOS on the Intel Paragon.

This research was performed using the Intel Paragon system at Oak Ridge National Laboratory, the Intel Paragon system at Sandia National Laboratories, a Cray T3D at Cray Research, and the IBM SP system at Argonne National Laboratory

References

I. T. Foster and P. H. Worley, 1993: Parallelizing the spectral transform method: A comparison of alternative parallel algorithms, in *Parallel Processing for Scientific Computing*, R. F. Sincovec, D. E. Keyes, M. R. Leuze, L. R. Petzold, and D. A. Reed, eds., Society for Industrial and Applied Mathematics, Philadelphia, PA, pp. 100–107.

I. T. Foster and P. H. Worley, 1994: I. T. Foster and P. H. Worley, Parallel algorithms for the spectral transform method, Preprint, Argonne National Laboratory, IL 60439.

J. J. Hack and R. Jakob, 1992: Description of a global shallow water model based on the spectral transform method, NCAR Tech Note NCAR/TN-343+STR,

[§]complete...

National Center for Atmospheric Research, Boulder, CO.

D. L. Williamson, J. B. Drake, J. J. Hack, R. Jakob, and P. N. Swarztrauber, 1992: A standard test set for numerical approximations to the shallow water equations on the sphere, *J. Computational Physics*, 102, 211–224.

T. Kauranne and S. Barros, 1993: Scalability estimates of parallel spectral atmospheric models, in *Parallel Supercomputing in Atmospheric Science: Proceedings of the Fifth ECMWF Workshop on Use of Parallel Processors in Meteorology*, G.-R. Hoffman and T. Kauranne, eds., World Scientific Publishing Co. Pte. Ltd., Singapore, pp. 312–328.