PERFORMANCE OF A SPECTRAL ELEMENT ATMOSPHERIC MODEL (SEAM) ON THE HP EXEMPLAR SPP2000.

Mark Taylor, Richard Loft and Joseph Tribbia

National Center for Atmospheric Research
P.O. Box 3000, Boulder CO, 80303

Abstract. We describe the performance of a Spectral Element Atmospheric Model (SEAM) on the HP Exemplar SPP2000. SEAM uses spectral elements in the horizontal directions and sigma coordinates in the vertical. The model is both spectrally accurate, as demonstrated by a variety of test cases, and is well suited for modern distributed-shared memory computers, as demonstrated by the fact that we achieve 24 GFlops on a 240 processor HP Exemplar. More important to climate modelers, a 64 processor Exemplar can integrate a 20 level, 160 km resolution SEAM dynamical core at a rate of 3.6 wall clock hours per model year. SEAM is portable and is implemented in FORTRAN 77 with message passing calls.

1. Introduction

We have recently completed the development a Spectral Element Atmospheric Model (SEAM). SEAM is a dry, three-dimensional general circulation model (GCM) dynamical core. It uses a spectral element discretization of the surface of the sphere taken from the shallow water model described in [13]. The spectral element method is a finite element method in which a high degree spectral method is used within each element. The method provides spectral accuracy while retaining both parallel efficiency and the geometric flexibility of unstructured finite elements grids. In the vertical direction, SEAM makes use of a sigma coordinate finite difference discretization strategy taken from NCAR's Community Climate Model Version 3 (CCM3) [9].

Spectral elements have several advantages for global climate modeling. First of all, handling the spherical geometry presents no problem since the sphere can be tiled with quadrilateral elements of approximately the same size, thus avoiding clustering points at the poles. Secondly, by using a local coordinate system within each element, the singularities associated with spherical coordinates can also be avoided. Additionally, the method is spectrally accurate. This has been shown even for difficult nonlinear problems in spherical geometry [13]. From a computational point of view, the method enjoys several advantages on parallel computers based on RISC microprocessors, for example distributed-shared memory (DSM) computers

We would like to thank Greg Astfalk and Mahesh Rajan of Hewlett Packard for help with the performance results. We would also like to thank Aimé Fournier for helpful comments. This work was supported in part by the U.S. Department of Energy; Office of Energy Research; Special Research Grant: CHAMMP, grant number DE–FG05–91–ER61219.

like the HP Exemplar. The spectral transforms within elements are localized in memory, so they fit well into microprocessor cache, and consist of numerically intensive matrix multiply operations. The ratio of communication between the element's surfaces to the amount of computations within the element's volume is small for the spectral method degree typically used (8-16). These properties of the spectral element method allow SEAM to sustain over 100 MFlops per processor for 320 km, 160 km, and 80 km resolutions on up to 240 processors of the HP Exemplar SPP2000.

The performance of SEAM on a DSM computer will be the focus of this paper, but we will first describe the spectral element method and go into detail about why it should perform well on DSMs. Then we give some details of the SEAM code, along with a brief description of the Held-Suarez benchmark. This benchmark is used to judge both the accuracy and performance of SEAM as a possible GCM dynamical core.

2. The Spectral Element Method

In this section we give a brief summary of the spectral element discretization. For a more thorough review of this material, see [10]. The spectral element discretization used for this work is the one described in [7]. The only differences arise from the complications of spherical geometry and in the treatment of the diffusion term. In the spectral element discretization, the computational domain is decomposed into rectangular regions called elements. Within each of these elements all variables are approximated by polynomial expansions. The discrete equations are derived by using a Galerkin or integral form of the equations to be solved in conjunction with a suitable set of test functions and quadrature formula.

The spectral element discretization is particularly simple because of a clever combination of an integral form of the equations, the family of test functions used, and the quadrature formula chosen to approximate the integrals. This leads to two significant advantages over more conventional finite element methods: first, for time marching problems such as the shallow water and primitive equations, the mass matrix is diagonal. Thus if an explicit time stepping routine is used, the complete calculation is also fully explicit! The second advantage is that spectral elements allow for arbitrarily high degree spectral expansions to be used within each element, without creating any additional coupling at the element boundaries. This is demonstrated in [13], where we show spectral convergence up to degree 56 on a now standard set of shallow water test cases on the sphere [14]. These test cases include both problems with analytical solutions and difficult fully nonlinear problems where extremely high resolution runs are used for the “exact” solution. In practice one must strike a balance between accuracy and computational cost by adjusting the polynomial degree and the number of elements. For the test cases in [14] the most efficient configuration is to fix the polynomial degree between degree 8 and 16 and then determine the number of element to obtain the desired resolution [13].

The procedure to derive the spectral element discretization is as follows. First an integral form of the equations is chosen. To do this, both sides of the equations are multiplied by a test function and integrated over the entire computational domain (in our case, the surface of the sphere). The sphere is then tiled with rectangular elements and the integrals are written as the sum of integrals over each element.

The element integrals are then approximated by Gauss-Lobatto quadrature [3]. Within each element we represent all dependent variables as tensor products of polynomials. Although the actual basis functions used for this representation turn out to be irrelevant, it is convenient to use the Gauss-Lobatto Cardinal functions [7, 2] as basis functions for this polynomial space. These basis functions are uniquely determined by the requirement that each function has the value 1 at one Gauss-Lobatto quadrature point and 0 at the remaining quadrature points.

Finally, the global test functions must be chosen. These are also constructed out of the Gauss-Lobatto Cardinal functions. We choose one test function for each grid point. For grid points in the interior of an element, we set the global test function to be equal to the Cardinal function associated with that grid point inside the element. This function is then extended continuously to a global function by taking it to be zero in all other elements. For grid points on element boundaries, which are shared by several elements, the global test function is built out of several Cardinal functions. Within each element containing the grid point, we set the global test function equal to the Cardinal function associated with that grid point. For all other elements, we set the global test function to zero. This process also generates a continuous test function.

Combined with a time stepping scheme, the procedure above generates a set of integral equations which completely specify how the equations are to be solved. Remarkably, this leads us to a simple Legendre spectral transform method within each element. Derivatives are calculated with Legendre transforms on the Gauss-Lobatto grid. Because of this grid and the associated Cardinal functions, the only communication between elements occurs at the element boundaries, where neighboring elements share common points. At these points the terms appearing in the equations are multiple valued and the required area weighted averaging is specified by the discretized form of the equations. This averaging is independent of the flow, and thus does not result in any type of up-winding.

For time stepping, we use the standard second-order leapfrog scheme with a $2\Delta t$ Robert filter of strength .05 [12].

3. Spectral Elements on the Sphere

The first step in applying the spectral element method to spherical geometry is to tile the sphere with rectangular elements, or regions that can be easily mapped to rectangles. This tiling is most readily accomplished by inscribing a regular polyhedron with rectangular faces inside the sphere, and then using the gnomonic projection (project from the center of the sphere) to map the surface of the polyhedron to the surface of the sphere. The most elementary such polyhedron is the cube, which creates 6 large elements. One can then further divide each of these elements into smaller elements, and the projection onto the sphere of such a configuration is shown in Figure 1. Each of the 6 faces of the cube has been divided into an 8×8 array of elements.

The only other complication coming from spherical geometry is the choice of coordinate system. Spherical coordinates are a particularly poor choice since the coordinates are discontinuous at the poles. When vector fields are represented in spherical coordinates, their components are discontinuous and contain large variations at the poles. This will create large errors in the derivative calculations. To avoid this problem we again make use of the cube and the gnomonic projection:

each element is mapped back to the surface of the inscribed cube, where we can make use of the natural Cartesian coordinate system on each face of the cube. One minor drawback of this projection is that it is not orthogonal. This makes differential operators slightly more complicated. The formulas for the divergence, curl and gradient operations in this cube coordinate system are given in [13]. These three differential operators are all that are needed to solve either the shallow water equations or primitive equations, and all the coordinate mappings are completely isolated to these three routines.

4. Spectral Elements for DSMs

Spectral element methods are well suited to modern RISC based parallel computers for two reasons. First, the basic data structure in the method, the spectral element, is naturally cache blocked. Secondly, due to the $O(N^3)$ cost of the spectral transforms, the method has a very low ratio of communication to computation.

4.1 Cache Blocking.

The elements in the spectral element method provide a natural way to cache block the SEAM code. The 64 bit data storage requirement for a typical 8×8 element with 20 vertical levels requires only 10 Kbytes of cache. One hundred such 3-d variables will fit into the 1 Mbyte data cache of the HP PA8000. If the data is stored in memory so as to avoid cache conflicts, then all the computations performed within an element can be done entirely in cache. This blocking is independent of resolution since we can increase the resolution by simply using more elements. This situation is unlike spherical harmonic spectral methods where the natural block size and data access patterns grow with resolution. Thus a spectral element model maintains good performance even at high resolutions where one can expect global spectral models start to experience cache "thrashing".

4.2 The time stepping bottleneck.

The performance bottle neck of the spectral element method on the HP Exemplar is the time stepping routine. In this routine one must do a simple mult-add over the entire data set, with no chance of any type of data reuse. When the working set size on a processor is much larger than the processor's cache the time stepping loop derives almost no benefit from the cache and the performance of this loop is limited by the bandwidth between the processor and DRAM.

4.3 Domain decomposition.

The most natural way to parallelize the spectral element method is to simply assign several elements to each processor. Each element only needs information from adjacent elements, so the domain decomposition reduces to a standard graph partitioning problem. To solve this problem, we use the METIS software package written by G. Karypis and V. Kumar [8]. Once the domain decomposition is computed, we must schedule in which order the domains will send and receive messages to neighboring domains. This scheduling is computed by a simple greedy edge coloring algorithm.

The resulting communication patterns are similar to finite difference/finite element methods which parallelize with the same type of domain decomposition. Thus the parallel efficiency of this method would be similar to these other methods, except for the fact that the spectral element method involves computing high order derivatives using spectral transforms instead of low order stencils. These computationally intensive transforms are performed within each element and are localized to each processor thus requiring no communication.

The resulting communication costs as a percentage of the total wall clock time are minimal. For a wide range of problem sizes, number of processors, and computer architecture, the communication is never more than 10%. In extreme cases, when the number of elements per processor is less than 3 or for very large numbers of processors, the communication costs can rise to 20%. (These results for the HP Exemplar will be presented in Section 6.) Because of these small percentages, we have not yet spent time optimizing our domain decomposition or message scheduling algorithms.

5. SEAM: A Spectral Element Atmospheric Model

The ideas outlined in the previous sections have been combined into a Spectral Element Atmospheric Model, or SEAM. SEAM is closely related to SEOM, the Spectral Element Ocean Model being developed at Rutgers [5]. SEAM and SEOM are both based on the 2D shallow water ocean model described in [7], and both models now solve the full three dimensional primitive equations, although with equations of state and vertical coordinates appropriate for either the atmosphere or the ocean. Another difference between the two models is how they difference the vertical coordinate. SEOM uses a fully three dimensional spectral element discretization, where the elements are topologically cubes rather than squares. SEAM uses spectral elements only for the horizontal directions, and uses the finite difference formulation from [9] in the vertical direction. Thus each element contains many vertical levels and finite differences are used to compute vertical derivatives.

SEAM is written in standard FORTRAN 77 and is very portable. It can be run on a single processor or on multiple processors using either PVM or MPI. SEAM can solve either the 2D shallow water equations [14] or the primitive equations given in [9].

5.1 Shallow water version of SEAM.

We have made extensive use of the shallow water test cases [14] to establish several points about SEAM (running in shallow water mode). As already mentioned, these test cases verify that the spectral element method does converge exponentially fast to the exact solution and that the method has no problems with spherical geometry. The test cases prescribe several forms of error measures so that any method can be compared objectively to several other methods by simply consulting the literature. The results of this comparison were as expected: on a per grid point basis, the method achieves error levels similar to that of the computationally more expensive spherical harmonic spectral methods, and significantly more accurate than finite difference methods. These results are described in [13, 5].

5.2 Primitive equation version of SEAM.

The success of SEAM with the shallow water equations prompted us to extend the model to the three dimensional primitive equations. As stated above, we did this by retaining the spectral element discretization over the surface of the sphere and allowing for an arbitrary number of vertical levels within each element. We use the sigma coordinate finite difference discretization from NCAR's CCM3 [9]. CCM3 allows for a sigma-pressure hybrid coordinate system. However, at present SEAM only allows for the pure sigma version of this coordinate system.

When the vertical coordinate is treated as described above, it is relatively simple to adapt a shallow water model to a full three dimensional primitive equation model. In addition to the horizontal components of velocity, we add a third prognostic variable (temperature) and replace the surface height variable by surface pressure. There are additional terms coming from the diagnosed vertical velocity and geopotential, and we must also add thermodynamics. All of these equations are taken from [9]. The result is a complete dry dynamical core of an atmospheric general circulation model. To validate this model and to compare it with other GCM dynamical cores, we have made use of the Held-Suarez dynamical core test case, which we describe in Section 6.

At present we have only implemented an explicit time stepping routine in SEAM. This results in a time step which is eight times smaller than that typically used in a model at a comparable resolution which compute derivatives using global spherical harmonic transforms. However, for the reasons sited in Section 4.1, we expect SEAM to achieve computational rates several times larger than spherical harmonic models on DSMs. Further, the inferior computational scaling properties of spherical harmonic transforms lead us to believe that an explicit spectral element method will be more efficient than a semi-implicit spectral method at resolutions at or above 160 km. This crossover point is an area that needs further investigation.

6. The Held-Suarez Benchmark

The shallow water equations have many of the difficulties associated with the dynamical aspects of climate modeling and any potential climate model should perform well on the shallow water test cases described above. However, these tests are mainly useful for comparing the accuracy of numerical methods and their ability to handle spherical geometry. Since the longest test case is only 15 days, the tests measure forecast accuracy rather than the ability to accurately generate long term climate statistics. A more ambitious climate comparison project is AMIP, the Atmospheric Model Intercomparison Project [4, 1]. This project compares the climate simulations of many atmospheric GCMs using realistic forcing and boundary conditions. However, due to the complexity of these models, it is very difficult to attribute differences in the results to specific differences in the models.

The Held-Suarez Benchmark [6] test case falls in the middle of these two approaches. It is designed to test the dry dynamical core of a GCM. It assumes an ideal gas atmosphere over a rotating sphere with no topography. The flow need not be hydrostatic, but as mentioned above, we use the hydrostatic primitive equations to model this flow. For forcing and dissipation, Held and Suarez use "simple Newtonian relaxation of the temperature field to a zonally symmetric state and Rayleigh damping of the lower level wind to present boundary layer friction." Unlike the shallow water tests, this benchmark tests a complete component that could appear unmodified within a GCM. And unlike the AMIP, just one component of a GCM is tested, making it easier diagnose the causes of differences in model results.

The Held-Suarez forcing has the form:

$$\frac{\partial \mathbf{v}}{\partial t} = \cdots - k_{\mathbf{v}}(\phi, \sigma)\mathbf{v}$$

$$\frac{\partial T}{\partial t} = \cdots - k_T(\phi, \sigma)[T - T_{\mathrm{eq}}(\phi, \sigma)]$$

Here $\mathbf{v}$ is the horizontal velocity, T the temperature, ϕ is latitude and $\sigma = p/p_{\mathrm{s}}$, where p is pressure and p_{s} is surface pressure. The ellipses represent unspecified expressions that model an ideal gas over a rotating sphere. The temperature is relaxed to a prescribed "radiative equilibrium," T_{eq}, with relaxation rate k_T. There is a simple linear damping of the velocities given by $k_{\mathbf{v}}$.

Results from the Held-Suarez benchmark are shown in Figures 2, 3 and 4. These results are all final 1000 day time averages from 1200 day runs. We present these diagnostics mainly to validate SEAM by showing it produces results almost identical to those of a spherical harmonic spectral mode. This is not surprising since the models use a similar vertical discretization, and the horizontal discretization have been shown to agree to several digits in 13.

Figure 2 shows the eddy kinetic energy as a function of zonal resolution for our spectral element model (SEAM), a grid point model, and a spherical harmonic spectral model. For the last two models, the results were taken from [6], which also contains a description of those two models. As the figure shows, the two spectral models have very similar performance.

Figure 3 shows the zonal mean wind from SEAM, and Figure 4 shows the eddy variance of the temperature. Both of these plots are shown for the grid point model and spectral model in [6]. Consulting that paper shows that all the models produce similar results. The results again show the two spectral models are almost identical, and they differ slightly from the results of the finite difference model.

7. SEAM on the HP Exemplar

We now discuss some performance results for SEAM running the Held-Suarez benchmark on an HP Exemplar SPP2000. These results were obtained from the 64 processor Exemplar at NCAR and a similar 256 processor Exemplar system at the California Institute of Technology. The SPP-2000 is built from 180 MHz PA-8000 CPUs, each with a single, direct mapped 1 MByte cache. The processors are combined into 16 processor "hypernodes", which share up to 4 Gbytes of Memory. Larger systems are formed from a 2-d torus of hypernodes interconnected using HP's Coherent Toroidal Interconnect (CTI).

One big advantage of the Held-Suarez benchmark over the shallow water test cases is that it represents a complete component of a GCM. It allows us to compute meaningful performance measures since the efficiency of a model on this benchmark directly impacts how efficient it will be as a GCM. Our results are presented for the Held-Suarez Benchmark using 20 vertical levels (denoted by L20) and three different horizontal resolutions: 96 elements, 384 elements, and 1536 elements. All cases use an 8×8 grid within each element, corresponding to spectral representation up to degree 7. This results in an average grid spacing at the equator of 320 km, 160 km and 80 km respectively.

In Figure 5, we present the total GFlops obtained for these resolutions on up to 240 processors. At 80 km resolution, SEAM achieves 24 GFlops on 240 processors (7 elements per processor). At 160 km resolution, SEAM achieves 14 GFlops on 128 processors (3 elements per processor). At 320 km resolution, SEAM achieves 6.2 GFlops on 48 processors (2 elements per processor).

The scalability of the method is better illustrated in Figure 6, showing the MFlops per processor. As can be seen in the figure, the 320 km and 160 km resolutions achieve good scalability all the way up to 2-3 elements per processor.

In this range the performance fluctuates between 110 and 130 MFlops per processor at 320 km resolution, and between 130 and 150 MFlops per processor at 160 km resolution. At 80 km resolution, the performance is between 100 and 120 MFlops/processor, all they way up to 240 processors (6-7 elements per processor). It seems likely that this scalability will also be maintained up to 3 elements per processor, which would give us close to 50 GFlops on 512 node machine.

Figure 7 shows the percentage to total wall clock time spent on message passing. As expected, this number grows steadily as we keep the problem size fixed while using more and more processors. The fact that this growth does not impact the scalability must be due to the fact that as we add more processors the problem size per processor decreases leading to better cache utilization.

More important than MFlops is the actual time to solution. These numbers are given in the following table:

		GFlops	Wall clock hours
Method	Resolution	(64 processors)	per model year
SEAM	320km/L20	6.3	1.3
SEAM	160km/L20	9.1	3.6
SEAM	80km/L20	7.3	34

8. Conclusion

The new code SEAM (Spectral Element Atmospheric Model) solves the three dimensional primitive equations in spherical geometry. SEAM uses finite differences in the vertical direction and a spectral element discretization in the horizontal directions. SEAM is based on an extensively tested shallow water model which was extended to a full GCM dynamical core. SEAM has been validated by making use of the Held-Suarez benchmark. Our results are practically identical to the Held-Suarez results from their spherical harmonic spectral model. SEAM also performs well on DSMs. On the HP Exemplar, SEAM achieves between 100 and 150 MFlops per processor on up to 240 processors and over a wide range of model resolutions.

References

1. G. J. Boer et al., *Some results from an intercomparison of the climates simulated by 14 atmospheric general circulation models,* J. Geophys. Res., **97,** pp. 12771–12786 (1992).

2. J. Boyd, *Chebyshev and Fourier Spectral Methods. Lecture Notes in Engineering,* edited by C.A. Brebbia and S.A. Orszag (Springer-Verlag, New York 1989).

3. C. Canuto, M. Hussaini, A. Quarteroni and T. Zang, *Spectral Methods in Fluid Dynamics,* (Springer-Verlag, Berlin, 1988).

4. W. L. Gates, *AMIP: The atmospheric model intercomparison project,* Bull. Am. Meteorol. Soc., **73,** pp. 1962–1970 (1992).

5. D. L. Haidvogel, E. Curchitser, M. Iskandarani, R. Hughes and M. Taylor, *Global Modeling of the Ocean and Atmosphere Using the Spectral Element Method,* to appear, Atmosphere-Ocean (1997).

6. I. Held and M. Suarez, *A Proposal for the Intercomparison of the Dynamical Cores of Atmospheric General Circulation Models,* Bull. Amer. Met. Soc. **75,** pp. 1825-1830 (1994).

7. M. Iskandarani, D. L. Haidvogel and J. P. Boyd, *A Staggered Spectral Finite Element Model With Application to the Oceanic Shallow Water Equations,* Int. J. Num. Meth. Fl., **20,** pp. 393–414 (1995).

8. G. Karypis and V. Kumar, *A fast and high quality multilevel scheme for partitioning irregular graphs,* Technical Report TR 95-035, (Department of Computer Science, University of Minnesota, 1995).

9. Kiehl et al., *Description of the NCAR community climate model (CCM3),* NCAR Technical Note NCAR/TN-420+STR, (National Center for Atmospheric Research, 1996).

10. Y. Maday and A. T. Patera, “ Spectral element methods for the incompressible navier-stokes equations,” In *State of the Art Surveys in Computational Mechanics*, edited by A. K. Noor (ASME, New York, 1988).

11. A. T. Patera, *A spectral element method for fluid dynamics: Laminar flow in a channel expansion,* J. Comput. Phys., **54,** pp. 468–488 (1984).

12. A. F. Robert, *The integration of a low order spectral form of the primitive meteorological equations,* J. Meteor. Soc. Japan, **44,** pp 237-245 (1966).

13. M. Taylor, J. Tribbia and M. Iskandarani, *The spectral element method for the shallow water equations on the sphere,* J. Comput. Phys., **130,** pp. 92-108 (1997).

14. D. L. Williamson, J.B. Drake, J. J. Hack, R. Jakob and P.N. Swarztrauber, *A standard test set for the numerical approximations to the shallow water equations in spherical geometry,* J. Comput. Phys., **102,** pp. 211-224 (1992).

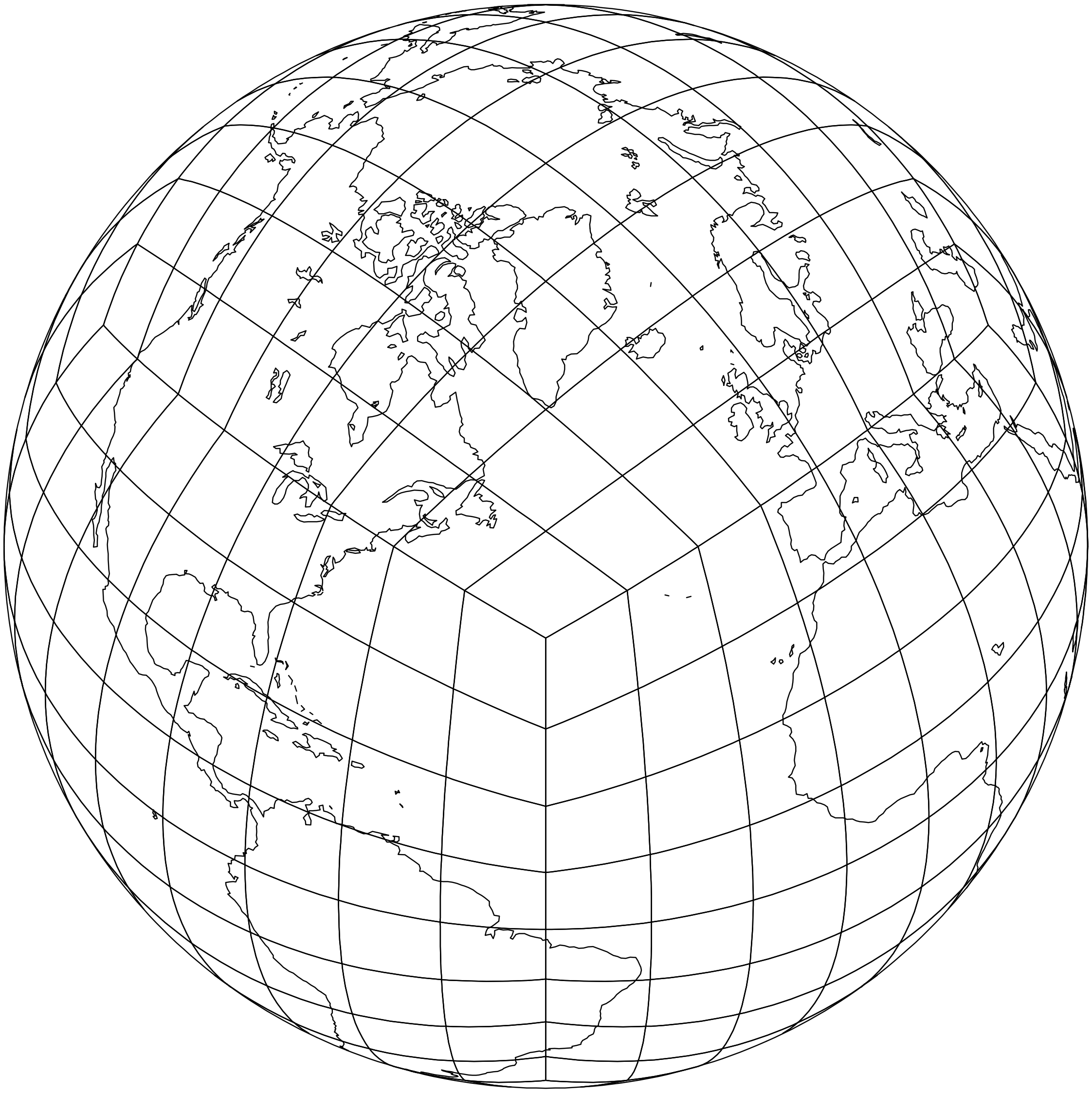

Figure 1. The cube projected onto the sphere. Each of the 6 faces of the cube has been divided into an 8×8 array of elements. Within each element we would typically use an 8×8 Gaussian grid and 20 vertical levels. This results in approximately 160 km horizontal resolution.

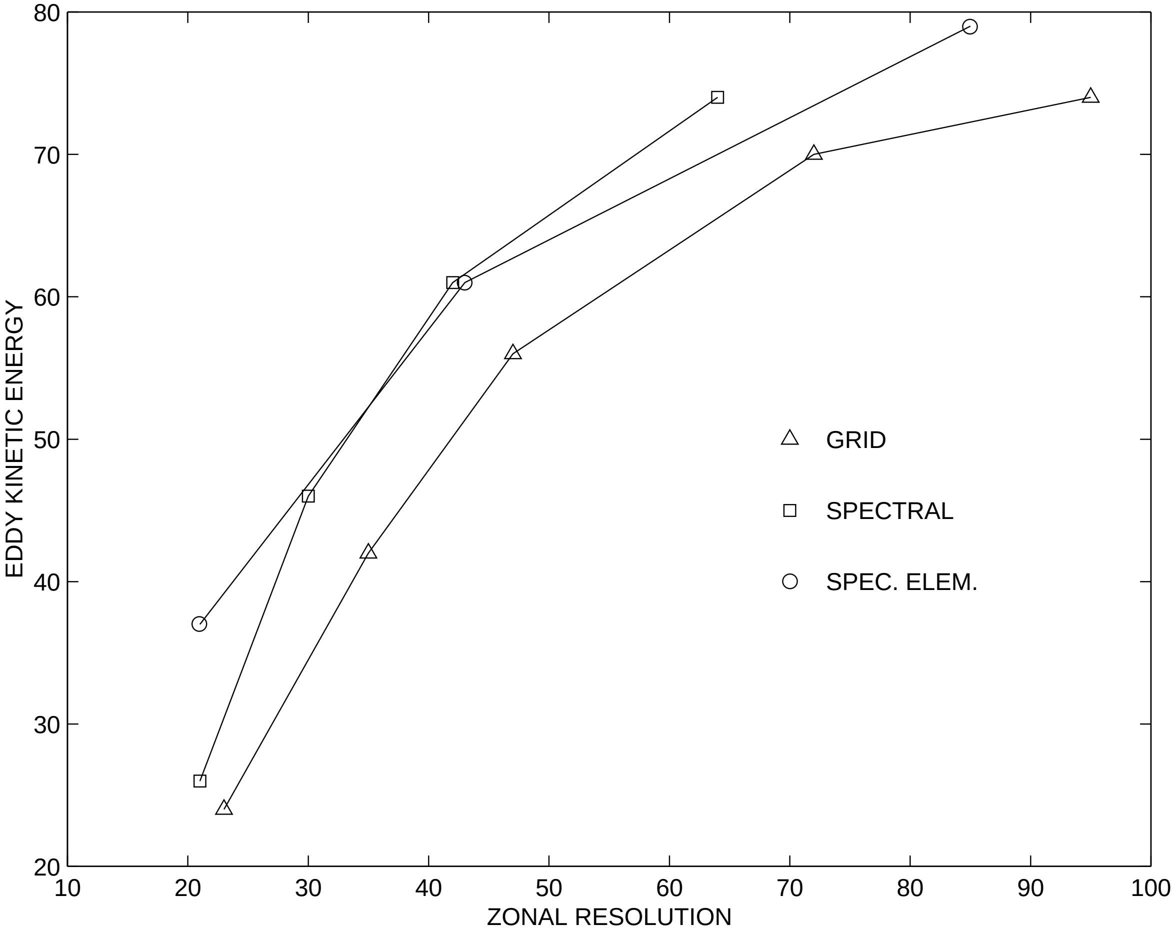

Figure 2. Eddy kinetic energy as a function of zonal resolution. Values for a grid point model (triangles) and a spherical harmonic spectral model (squares) are taken from [6]. Results from SEAM (circles) are in close agreement with the spherical harmonic spectral mode.

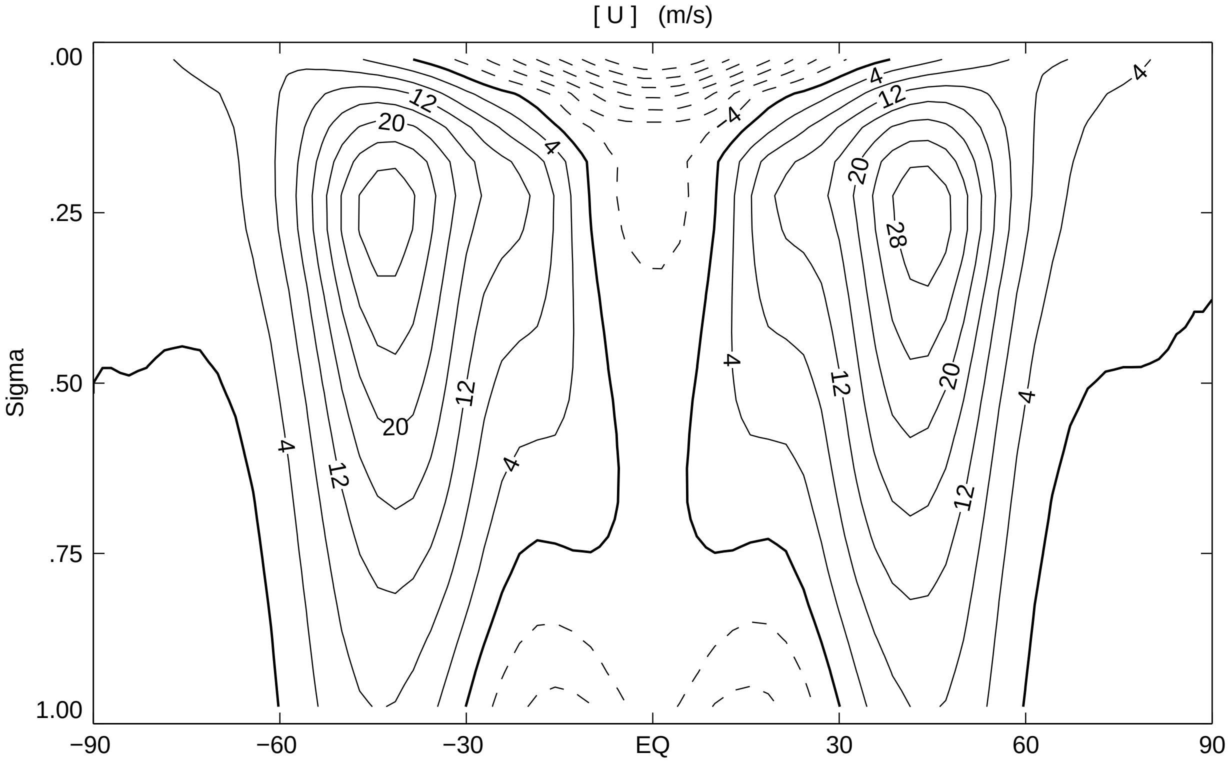

FIGURE 3. The zonal mean wind. The mean is computed from the last 1000 days of a 1200 day run of SEAM at 160 km resolution with 20 levels. The forcing is symmetric about the equator so difference between the hemispheres are indicative of the variability of the mean.

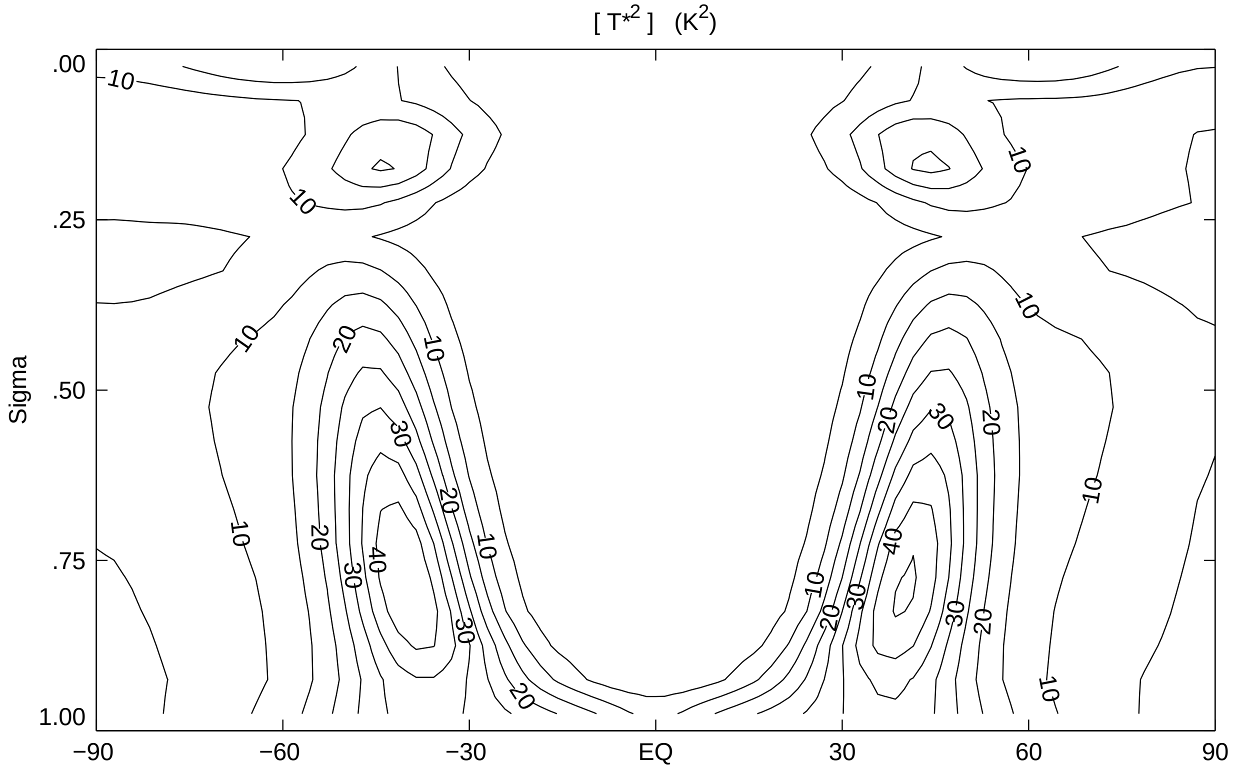

FIGURE 4. The zonal mean of the eddy variance of the temperature. The mean is computed from the last 1000 days of a 1200 day run of SEAM at 160 km resolution with 20 levels.

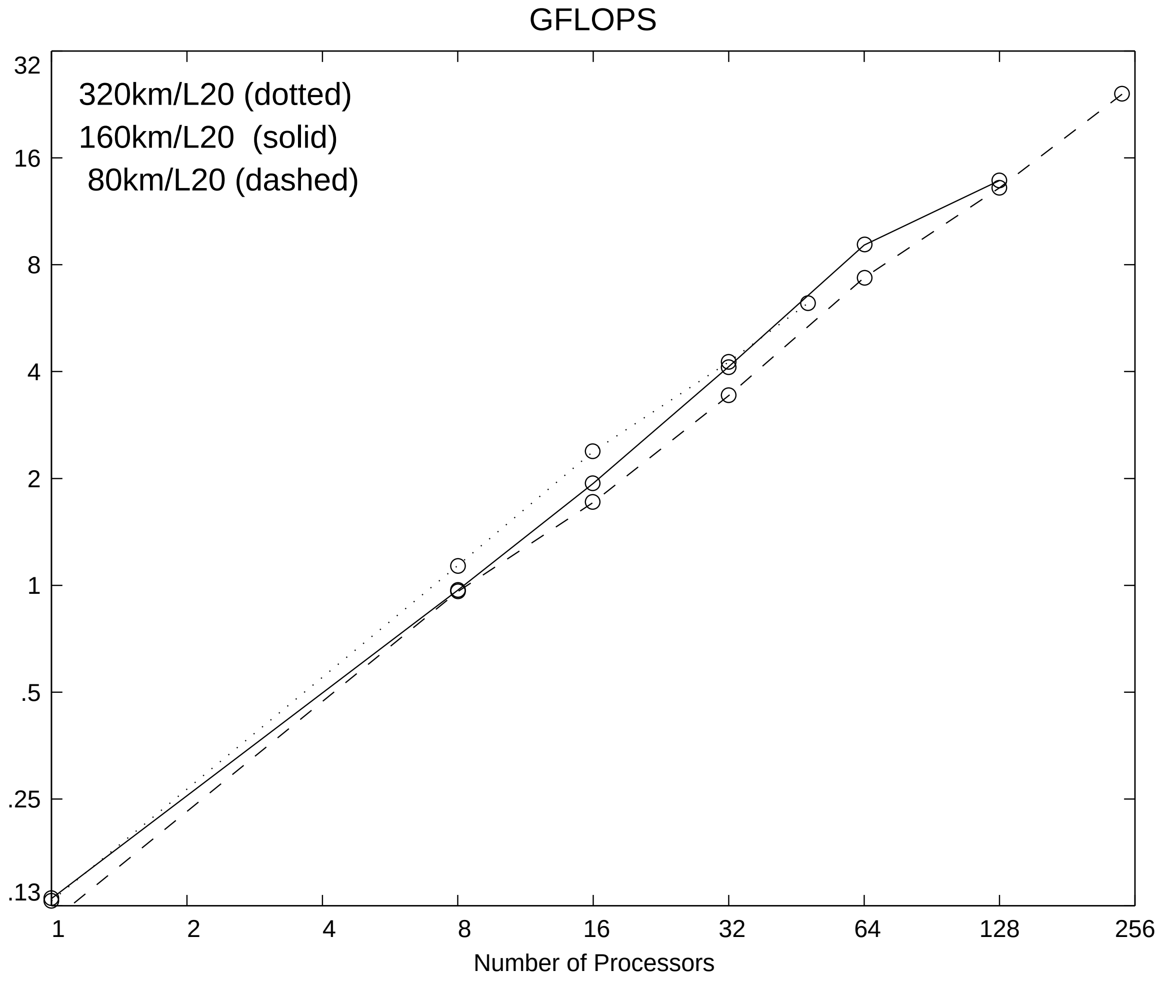

FIGURE 5. GFLOPS achieved by SEAM for three different model resolutions and up to 240 processors.

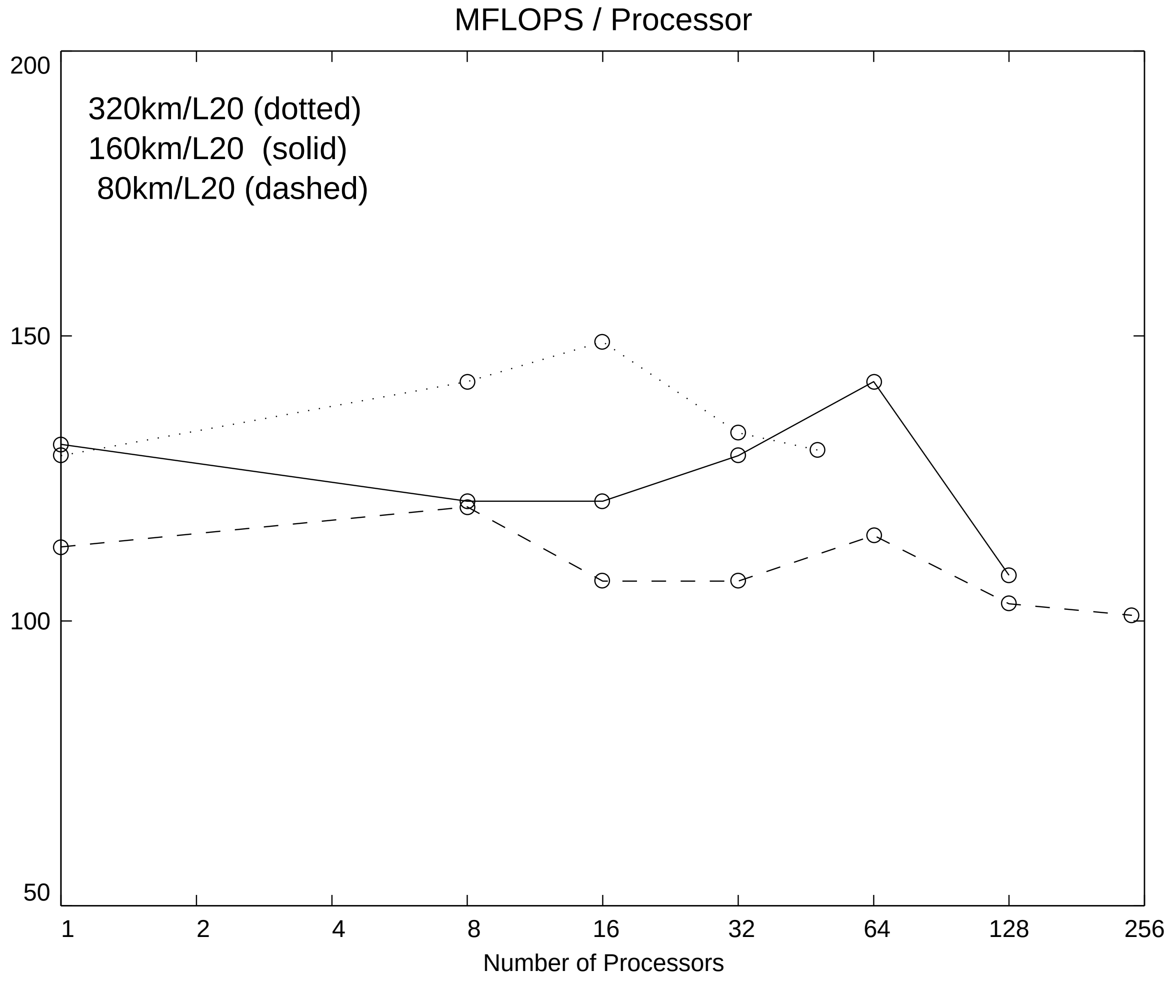

Figure 6. MFLOPS per processor achieved by SEAM at three different model resolutions and up to 240 processors.

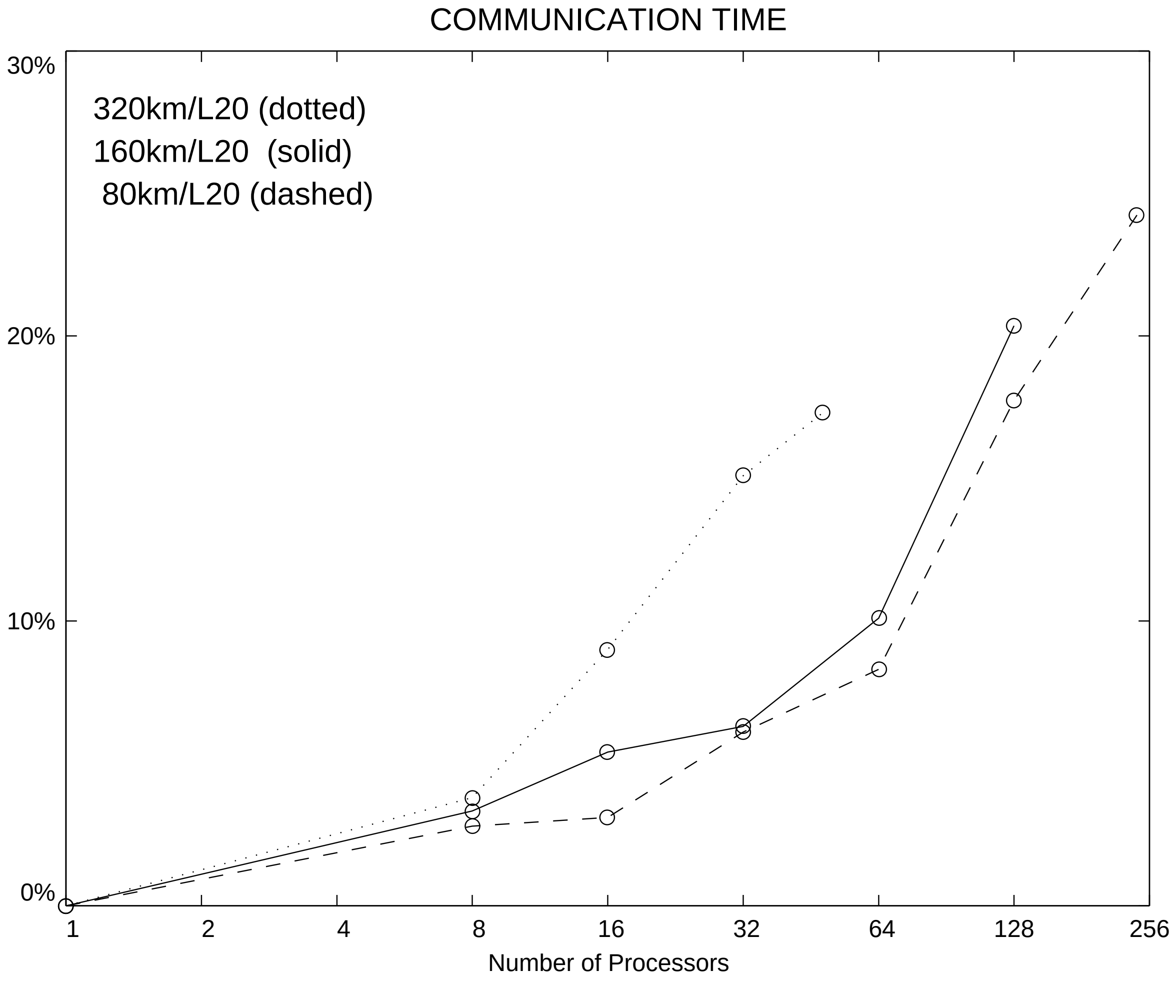

Figure 7. Percentage of the total run time spent message passing.