# Towards Ultra-High Resolution Models of Climate and Weather

Michael Wehner, Leonid Oliker, John Shalf
*CRD/NERSC, Lawrence Berkeley National Laboratory, Berkeley, CA 94720*

**Abstract**

We present a speculative extrapolation of the performance aspects of an atmospheric general circulation model to ultra-high resolution and describe alternative technological paths to realize integration of such a model in the relatively near future. Due to a superlinear scaling of the computational burden dictated by stability criterion, the solution of the equations of motion dominate the calculation at ultra-high resolutions. From this extrapolation, it is estimated that a credible kilometer scale atmospheric model would require at least a sustained ten petaflop computer to provide scientifically useful climate simulations. Our design study portends an alternate strategy for practical power-efficient implementations of petaflop scale systems. Embedded processor technology could be exploited to tailor a custom machine designed to ultra-high climate model specifications at relatively affordable cost and power considerations. The major conceptual changes required by a kilometer scale climate model are certain to be difficult to implement. Although the hardware, software, and algorithms are all equally critical in conducting ultra-high climate resolution studies, it is likely that the necessary petaflop computing technology will be available in advance of a credible kilometer scale climate model

## 1 Introduction

Accurately modeling climate change and weather prediction is one of the most important challenges facing computational scientists today, with economic ramifications in the trillions of dollars. Effectively performing these calculations is predicated on our ability to run high resolution models — the ultimate target being simulations that are capable of resolving climate processes down to the kilometer scale. The case to develop models at such ultra-high resolutions has many convincing arguments. Certainly the need for greater fidelity in both short term weather prediction and long term climate change estimates will be better met with higher horizontal resolution. However, an even more compelling case can be made when one considers that at kilometer-scale modeling, important processes that are currently parameterized may instead be resolved. Principal among these are the deep convective processes responsible for the transport of moisture from near the surface to higher altitudes [18]. Uncertainty in both short and long term forecasts can be traced to an inability to adequately represent this transport. For instance, seasonal predictability depends crucially on the correct distribution of water vapor in the atmosphere while predictions of anthropogenic climate change are highly dependent on cloud-radiation interactions.

In this paper, we will not address how to bridge the gap between cumulus convection parameterizations and cloud system resolving submodels. Rather, we will quantify the computational scaling behavior of an existing atmospheric model at existing horizontal resolutions and extrapolate to the kilometer scale to demonstrate that such calculations are not unreasonable in the next few years. This is not meant to suggest that such a model would be appropriate at these scales without major modifications. However, this methodology gives an estimate as to the computational burden required for these ultra-high resolutions and provides guidance to the design of hardware and software to achieve such goals. Design and construction of a kilometer scale global atmospheric general circulation model would indeed require careful consideration of all processes represented, not just cloud processes. Nonetheless, a general strategy of integrating the equations of motion in the form of the Navier-Stokes equations plus source terms representing moist, radiative, turbulent and boundary processes remains sound.

The model used to aid in this speculation is the finite volume version of the Community Atmospheric Model (fvCAM). Developed at the National Center for Atmospheric Research, a great deal of effort has gone into the CAM software system to ensure portability and robustness across a wide variety of high performance computing platforms. The fvCAM uses a finite volume approximation to the atmospheric equations of motion as developed by Lin and Rood [14]. It is representative in both scientific sophistication as well as computational cost to any of the leading

atmospheric general circulation models in current use. The public release of version 3.1 was used without modification in this study.

The computational costs of three different horizontal resolutions were measured and analyzed. Designated for convenience as the *B*, *C* and *D* grids, the details of the measured and extrapolated resolutions are summarized in Table 1. As with many other production climate and weather prediction models, the spheroid of the Earth is discretized by equally spaced (in solid angle) intersecting lines of latitude and longitude. Note that the longitudinal lines intersect at the poles, causing a convergence in the distance spacing in that region. The vertical dimension was held fixed in the default configuration of 26 layers.

| Mesh Name | Resolution | Mesh Dimension | Number Horizontal Points | Horizontal Resolution Near Equator |
|---|---|---|---|---|
| *B* | 2°x2.5° | 144x91 | 13,104 | 200km |
| *C* | 1°x1.25° | 288x181 | 52,128 | 100km |
| *D* | 0.5°x0.625° | 576x361 | 207,936 | 50km |
| *E* | 0.25°x0.3125° | 1152x721 | 830,592 | 25km |
| *F* | 0.125°x0.156° | 2304x1441 | 3,320,064 | 12.5km |
| *G* | 0.068°x0.078° | 4608x2881 | 13,275,648 | 6.25km |
| *H* | 0.038°x0.039° | 9216x5761 | 53,093,376 | 3.13km |
| *I* | 0.016°x0.020° | 18432x11521 | 212,355,072 | 1.57km |

Table 1: A summary of mesh characteristics considered in this paper. Only the properties of the *B*, *C* and *D* meshes are directly measured.

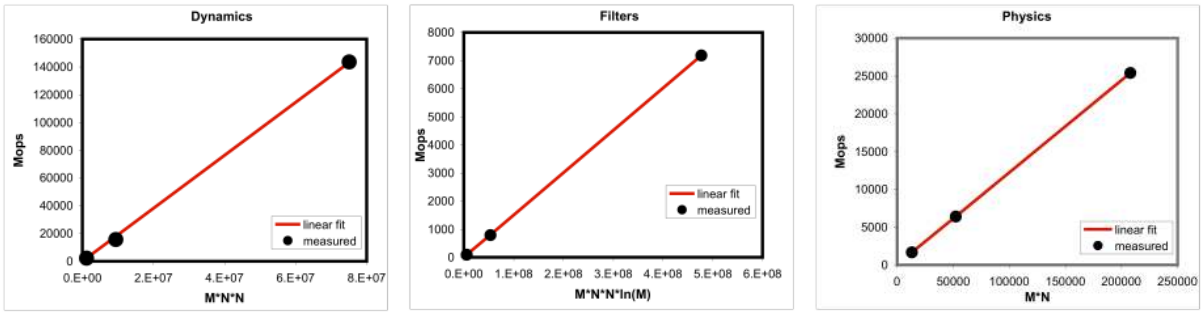
# 2 Measured Computational Cost of fvCAM

Atmospheric general circulation models (AGCM) consist of two distinct parts solving different aspects of the problem. Each has distinct scaling properties of the computational costs. The first part, often referred to as dynamics, solves the atmospheric equations of motion expressed by the Navier-Stokes equations of fluid dynamics. There are many different approaches to approximating a numerical solution to these equations. In CAM, a finite volume method [14] is one of three options. In all options for CAM, the atmosphere is approximated to be in hydrostatic equilibrium. The highly stratified nature of the atmosphere at large scales makes this a very good approximation at the relatively coarse resolutions of fvCAM in current use. It will break down at the resolutions we will extrapolate to in the next section and is important to consider in that discussion. The second part of AGCMs, often referred to as physics, approximate the unresolved or external processes relevant to the state of the atmosphere. These processes are included in the solution as source terms to the Navier-Stokes equations but must be calculated externally to the dynamics portion of the code. Relevant processes include but are not restricted to radiative transport of energy, convective transport of moisture, boundary layer effects, gravity wave drag, sub-grid scale turbulence and surface conditions.

The stability of explicit solutions to the Navier-Stokes equations is governed by the Courant-Friedrichs-Levy (CFL) condition. This restriction on the allowable time step depends linearly on the maximum wind speed and the grid spacing. As the horizontal resolution of an explicit model is increased, the number of time steps to solution of a fixed duration must also be increased. Hence, the number of operations of the dynamics portion of the code should scale with resolution as the number of horizontal mesh points ($M \times N$) times another factor of $N$ where $M$ is the number of longitudinal lines and N in the number of latitudinal lines. However, in latitude-longitude mesh codes like fvCAM, the CFL condition is overly restrictive due to the convergence of longitude lines at the poles. These high aspect cells near the pole do not offer any functional increase in local resolution in a long term solution. Any wave that might be resolved in this region will eventually move to regions of coarser resolution negating the effect of the high polar longitudinal resolution. As the CFL condition is applied globally, the time step for polar stability is overly restrictive everywhere else and provides no tangible benefits. To counter this, the dynamics time step may be chosen by a mid-latitude stability criterion. This will result in conditionally unstable solutions poleward of these latitudes [1]. However, this instability may be successfully damped out by careful application of appropriate latitudinally dependent Fourier filter functions without impact on the components of the solution that are stable to the CFL condition. The scaling of

the computational costs of such a filter are different than the rest of the dynamics portion of the code as a sum over all longitudinal points is necessary. This cost may be reduced by application of Fast Fourier Transforms (FFT) resulting in a scaling with resolution of $M \ln(M) N^2$, where the factor $M \ln(M)$ is characteristic of FFTs and the factor of $N^2$ is similar to the scaling in an unfiltered dynamics algorithm. The scaling behavior of the dynamics portion of code excluding the filter remains $M \times N^2$ although the constant multiplier is much reduced when this increased time step strategy is employed.

We can test these theoretical scaling relationships in the actual implementation of fvCAM by measuring the number of operations on an IBM Power3 system using the Hardware Performance Monitor (HPM) tool on the three mesh resolutions currently in use. In Figure 1(a), we show the number of operations in the dynamics portion of code excluding the filters to simulate 30 minutes of real model time as a function of resolution for the *B*, *C*, and *D* meshes. The abscissa axis is $M \times N^2$, the theoretical scaling factor for this section of code. The data plotted was obtained by subtracting a measurement for 30 simulated minutes from 60 simulated minutes to remove operations involved in initialization and termination. As is clear, a linear fit well approximates the measured data in Figure 1(a). In Figure 1(b), we show the number of operations in the filtering section of the dynamics portion of code obtained in the same manner. In this figure, the abscissa axis is $M \ln(M) N^2$ and is also very well described by a linear fit.




(a) (b) (c)

Figure 1: The number of operations required to integrate the (a) dynamics, (b) filters, and (c) physics portions of fvCAM 30 minutes of simulated model time as a function of resolution.

The physics portion of an AGCM is generally not subject to the CFL condition. Rather it is dictated by the physics of the parameterizations and is generally resolution independent over the range of validity of these parameterizations. Therefore, the scaling of this section of the code with resolution depends only on the number of horizontal mesh points, $M \times N$. Over the resolutions that we measured, the physics time step may be set independently of horizontal resolution. This may not hold at the extrapolated resolutions of the next section and will be discussed there. In Figure 1(c), we show the number of operations in the physics portion of code to simulate 30 minutes of model time as a function of the number of horizontal mesh points, $M \times N$. As with the previous two plots, a linear fit quite well represents the measured data. Note that we will return to the physics time step issue at ultra-high resolution in the discussion section of the paper.

We conclude that the scaling behavior of the fvCAM operations count is well represented by three separate relationships lending some confidence to an extrapolation to finer scales.

# 3 Envisioning Kilometer Scale Atmospheric Simulation

## 3.1 Analysis of Operations Count

Climate modeling and weather prediction are among the few HPC applications with well-defined time-critical performance requirements that motivate high computational throughput. Two key applications present real-time constraints on the completion time: simulations of anthropogenic climate change and regional scale weather forecasting.

For climate change analyses, two types of simulations are often employed. The longer of the two are control runs that simulate statistically stationary climate over periods of multiple millennia to adequately characterize internal climate variability for climate change detection studies. A shorter duration category involves transiently forced

simulations of the past or current centuries. Hence, a reasonable metric is to require that the model simulate time 1000 times faster than real time. Under this constraint, the control run integration can be completed in one year and transient runs can be completed several months later depending on how many realizations and scenarios are required. We note that the runs made for the Fourth Assessment Report of the Intergovernmental Panel on Climate Change of the current NCAR Community Climate System Model, CCSM3.0, ran about 1600 times faster than real time on IBM Power4/p690 machines.

The major medium range weather prediction centers generally perform at least a one high resolution ten-day forecast each day in addition to ensembles of forecasts at lower resolutions. The associated throughput metric is significantly relaxed in this application as the model must simulate time only 10 times faster than real time. Seasonal forecasting requires an equivalent computational throughput as the integrations are for a period of about ten times longer but the turnover rate is not as constrained as for the daily forecast.

As shown in Figures 1(a-c), the total number of floating point operations to integrate fvCAM for a specified duration as a function of horizontal resolution is well described by the combination of three linear functions depending on the mesh dimensions. We can use this information to estimate a sustained throughput rate that must be obtained given a constraint on the wall clock time to completion of the integration under the two scenarios — 10 times faster than realtime for weather forecasting and 1000 times faster than realtime for climate change prediction.

In Table 2, we show the minimum *sustained* computational performance in Teraflops required to integrate fvCAM 1000 times faster than real time at the horizontal resolutions detailed in Table 1. The performance required for medium range weather forecasting would be about 100 times less. We note that the relative distribution of operations changes dramatically with resolution. In the 2°x2.5° *B* mesh case, the dynamics and physics respectively consume 57% and 41% of the operations, or roughly equivalent amounts. In the 0.5°x0.675° *D* mesh case, the percentages have shifted to 81% and 14%, a dramatic change. In the kilometer scale *I* mesh, the percentage of total operations required by the physics is almost negligible at just over a half of a percent. Perhaps most surprising is that the filters never dominate the calculation despite the unfavorable scaling relationship described in Figure 1(b). This suggests that the need for alternatives to the simple latitude-longitude discretization of the sphere may not be as critical at high resolution as one might suppose; at least from computational considerations. We note however that some of the alternative discretizations possess other appealing properties, most notably a more uniform distribution of individual cell volumes (see Section 5 for further discussion). These will ultimately prove to be far more favorable strategies to the design of ultra-high resolution global atmospheric models. Additionally, the alternative discretization methodologies do not require polar filtering, thus obviating the need for global communication.

| Mesh Name | Horizontal Resolution Near Equator | Performance 1000x spdup (Tflop/s) | % total time in dynamics | % total time in filters | % total time in physics |
|---|---|---|---|---|---|
| *B* | 200km | 0.002 | 57% | 2% | 41% |
| *C* | 100km | 0.014 | 71% | 3% | 25% |
| *D* | 50km | 0.098 | 81% | 4% | 14% |
| *E* | 25km | 0.727 | 87% | 5% | 8% |
| *F* | 12.5km | 5.608 | 90% | 6% | 4% |
| *G* | 6.25km | 44.15 | 92% | 6% | 2% |
| *H* | 3.13km | 351.3 | 92% | 7% | 1% |
| *I* | 1.57km | 2809.7 | 92% | 7% | 1% |

Table 2: Minimum sustained computational performance (in Tflop/s) required to integrate fvCAM 1000 times faster than real time and the relative percentages of the total operations in three main segments of the code.

## 3.2 Domain Decomposition

Obviously to achieve the sustained computational rates shown in Table 2, large numbers of processors must be effectively used by the code. Design of a highly parallel code such as fvCAM is a series of compromises. Version 3.1 is a highly engineered code with three types of parallelism. One of these parallel aspects is critical to scaling an AGCM to the kilometer scale resolution. Decomposition of the global domain into smaller subdomains allows the assignment of a portion of the globe to each processor. In a mixed parallel mode program such as fvCAM, groups of processors are

assigned to each subdomain. There are two choices for uniform domain decomposition in the horizontal dimension of a latitude-longitude mesh. In fvCAM, essentially the simpler of the two is used, a one dimensional domain decomposition where each subdomain contains all longitude lines but only a subset of latitude lines. This is partly a carryover of legacy code from the spectral dynamics version of CAM but also can be justified as computationally effective on the 200km scale *B* mesh — the only resolution supported for public usage — on the production machines available to the climate science community. Another factor, discussed in more detail below, is that this one dimensional decomposition greatly simplifies the semi-Lagrangian advection transport algorithm in the polar regions. However, this decomposition strategy places rather severe limits on the maximum number of horizontal subdomains and hence processors that may be exploited. The communication pattern necessitated by such a one dimensional domain decomposition strategy is simple, subdomains must only communicate border cell information to their nearest neighbor. Messages are sent and received as pairs; one to the north and one to the south. Analysis of the subdomain perimeter to area ratio, leads to the following formula for the theoretical speedup obtained for this strategy [7],

$$S \propto \frac{N_p}{1 + c_1 N_p / N_g} \tag{1}$$

where $N_g$ is the number of gridpoints, $N_p$ is the number of subdomains, $c_1$ is a machine and algorithm dependent constant and $S$ is the speedup over using a single domain. The important point of this scaling relationship is that the speedup will saturate to a constant as $N_p$ becomes large. In contrast, a two dimensional horizontal domain decomposition in which both the number of latitude and longitude lines are subsets of the global range possesses significantly better scaling properties. In this case, messages are sent and received in sets of four; north, south, east and west. A similar analysis of subdomain perimeter to area ratios yields the following formula for the speedup,

$$S \propto \frac{N_p}{\sqrt{1 + c_2 N_p / N_g}} \tag{2}$$

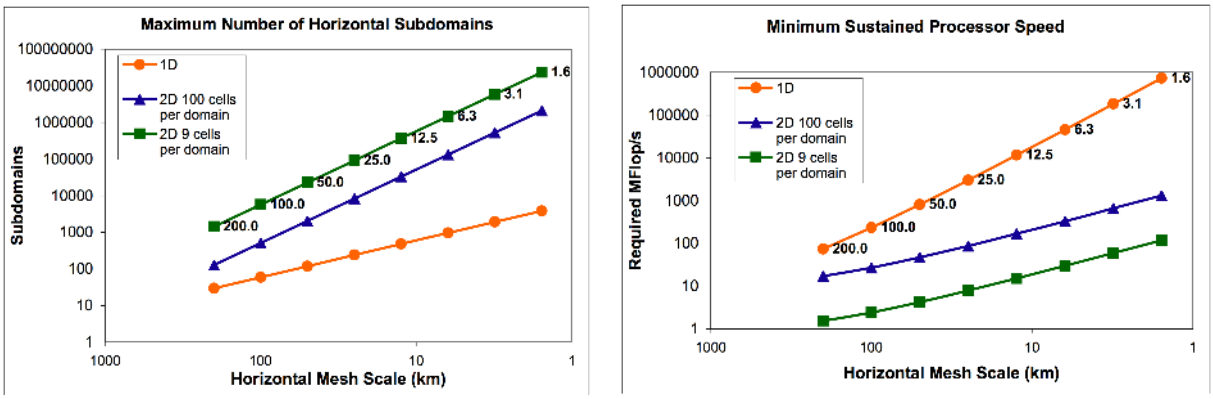
where $c_2$ is a different machine and algorithm dependent constant. Note that in this case, the speedup saturates to a dependence on $\sqrt{N_p}$ for large $N_p$. This can be significantly less than the ideal case which would be linear with $N_p$, but at least it continues to increase at large processor counts. Equation 2 does not include the costs of the non-local communications required to implement the polar filters [23] or semi-Lagrangian advection near the poles.

Measured performance of fvCAM compares well with Equation 1. However, such information provides little quantitative information about extrapolating to the resolutions of Table 1 as it is difficult to predict the dependence of the constants, $c_1$ and $c_2$, on machine parameters. Nonetheless, for large numbers of subdomains, a two dimensional horizontal decomposition allows scaling far beyond that which is possible in the 1D case, regardless of the detailed machined parameters.

A practical limit of the number of horizontal subdomains results from the fact that the number of latitude and longitude lines is not infinite. Obviously, these domain decomposition scaling equations break down when the number of subdomains approaches the mesh dimensions. In the fvCAM one-dimensional domain decomposition case, subdomains are constrained to have not less than three latitude lines. In this case, all data is communicated from a subdomain to its neighbors. In a two-dimensional case, the same limit would be reached if a subdomain were three cells long on each side, for a total of nine cells. In Figure 2(a), we show this practical limit on the number of horizontal subdomains at the range of resolutions in Table 1. We also show a more reasonable case of a two dimensional horizontal decomposition with subdomains constructed to be ten cells long on each side.

As is evident, the one dimensional horizontal decomposition becomes impractical at ultra-high resolutions as it is limited to less than 4000 subdomains on the 1.5 km scale *I* mesh as this scheme depends only on the number of latitude lines, $N$. In order to reach a sustained speedup of 1000 times realtime, each processor in such a system would need to deliver a sustained performance that approaches a Teraflop per subdomain! It would seem unlikely that sustained petaflop performance could be achieved at such an implied low processor count even with additional parallelism coming from a vertical decomposition and/or OpenMP. By contrast, the two dimensional case potentially permits the usage of millions of subdomains as this metric scales with resolution as the number of cells, $M \times N$.

Assuming one processor per horizontal subdomain, we show in Figure 2(b) the minimum sustained individual processor performance required to achieve the total performance showed in Table 2. Note that in both these figures additional parallelism achievable in fvCAM through the vertical decomposition and OpenMP are not included. The curves in Figure 2(b) are calculated assuming the processor configurations of Figure 2(a). At the highest resolutions we are considering, the total computational cost would be dominated by the dynamics (without filters) portion of




(a) (b)

Figure 2: (a) Shows the maximum number of horizontal subdomains for the 1D and 2D horizontal domain decompositions. The number of processors could be higher than these curves if additional parallelism is exploited. (b) The extrapolated minimum sustained processor speed required to integrate fvCAM at the climate model metric of 1000 times faster than real time, using 1D and 2D domain decompositions. For the medium range weather forecasting metric of 10 times faster than real time, the minimum required processor speed is a factor of 100 less than these results. (Horizontal length scales near the equator are shown on the top curve.)

the model. Therefore to maintain the performance metric of simulating time 1000 times faster than real time, the sustained processor speed must increase by a factor of approximately $M \times N$ for the one dimensional horizontal domain decomposition. However, in the two dimensional domain decomposition case, this scaling requirement on processor speed is approximately $N$, a far more favorable situation.

Finally, the extrapolations presented in Figure 2 assume 26 vertical grid layers. A current typical cloud system resolving model [15], uses 64 vertical grid layers. In a full atmospheric model, we consider that at least 100 vertical layers are required to accommodate the cloud system resolving model as well as the lower stratosphere and boundary layer. The additional vertical layers increases the overall computational burden (both CPU cycles and total memory) by a factor of four. However, the increased vertical resolution also opens the possibility of additional domain decomposition in the vertical direction. If the vertical dimension were domain decomposed into 10 discrete subdomains each 10x10x10, the total number of subdomains would be increased from 2million to 20Million and thereby reduce the sustained per-processor performance requirements to 0.5Gflop/s/processor to achieve the target speedup of 1000 times real time. However, some aspects of the vertical solution require very tight coupling which would place significant interprocessor communications requirements between vertical subdomains. Note that in the current model, a hybrid vertical and longitudinal decomposition is implemented within the one dimensional domain decomposition using both OpenMP parallelization and explicit MPI message passing. Purely vertical decompositions would be aided if the computational cores are very tightly coupled, such as chip multiprocessors or a tightly-coupled SMP.

### 3.3 Memory Requirements

The total memory required in the model might be expected to scale as the total number of computational mesh points, $M \times N$. In fact, the measured memory requirements are somewhat below this as can be seen by comparing in Table 3 the measured results (third column) with this upper bound (fourth column). This is likely a consequence of temporary memory requirements which do not cover the entire domain. A linear fit to the measured memory (fifth column) predicts somewhat lower extrapolated requirements. In this calculation, the global arrays required by processor 0 for initialization have been removed as a code scalable to ultra-high resolution must have some sort of parallel I/O. In any event, at the 1.5km scale *I* mesh, the total memory requirement should be of order 25 Terabytes or less.

This scaling behavior of the total memory requirements is independent of processor configuration as it is determined only by the size of the problem and not the geometry of the domain decomposition. Note that this scaling behavior is the same as for the processor count in a two dimension horizontal domain decomposition scheme. Hence, the amount of memory on any individual processor could be held constant with resolution by increasing the number

| Mesh Name | Horizontal Resolution Near Equator | Measured Total Mem (GB) | Max Memory (GB)* | Memory Least Sqrs Fit (GB)† | 1D 3-rows Required Mem/Proc (MB) | 2D 3x3 Required Mem/Proc (MB) | 2D 10x10 Required Mem/Proc (MB) | (Mem Bytes/ /Flop) |
|---|---|---|---|---|---|---|---|---|
| *B* | 200km | 1.5 | 1.5 | 1.3 | 51 | 1.06 | 11.8 | 0.68 |
| *C* | 100km | 4.9 | 6 | 5.3 | 102 | 1.06 | 11.8 | 0.44 |
| *D* | 50km | 21.4 | 24 | 21.3 | 203 | 1.06 | 11.8 | 0.25 |
| *E* | 25km | — | 96 | 85.0 | 406 | 1.06 | 11.8 | 0.13 |
| *F* | 12.5km | — | 384 | 339.9 | 813 | 1.06 | 11.8 | 0.070 |
| *G* | 6.25km | — | 1536 | 1359.4 | 1625 | 1.06 | 11.8 | 0.035 |
| *H* | 3.13km | — | 6144 | 5436.7 | 3251 | 1.06 | 11.8 | 0.018 |
| *I* | 1.56km | — | 24576 | 21745.1 | 6502 | 1.06 | 11.8 | 0.0089 |

Table 3: Estimates of memory requirements for fvCAM as a function of horizontal resolution and decomposition strategies. The last column shows the ratio of single processor minimum memory capacity to minimum sustained performance for any domain decomposition

of processors at the same rate as the number of grid cells. For the maximum processor case of 9 mesh cells per subdomain, the required single processor memory is just over 1 MB. In the 100 mesh cells per subdomain case considered above, this estimate increases to about 10MB. Substantial single processor memory requirements are necessary for the one dimensional decomposition case as the number of processors can only be increased at the same rate as the number of latitude lines. Table 3 (column six) shows the individual processor minimum memory requirements for the one dimension horizontal domain decomposition scheme peaking at over 6GB per processor. The relationship is linear with $M$, the number of longitude lines.

Interestingly, the relationship between single processor performance and single processor memory consumption is independent of domain decomposition strategy. In the last column of Table 3, we see that the ratio of single processor memory footprint to sustained single processor throughput (expressed in bytes per flop) decreases as horizontal resolution increases. These values are the same for the ratio of overall machine memory to sustained floating point operation rate. This decrease is a consequence of the CFL stability criterion which causes the operation count to scale at a rate greater than the number of mesh cells.

If the simulation were expanded to use 100 vertical layers, and employ a 10-way vertical domain decomposition, the memory requirements per subdomain are estimated to be 5MB in the 10x10x10 decomposition discussed in Section 3.2.

## 3.4 Interprocessor Communication Requirements

Three factors cause the sustained machine performance to be less than the advertised machine performance on any platform. The first of these is an inability to realize peak performance on the calculation within a single processor. This is generally a function of chip design as well as compiler effectiveness. The second of these is the communication of data between processors necessary to perform that calculation. Interpretation of the sustained processor speeds shown in Figure 2(b) relative to advertised peak computational rates must take both factors into account. Interprocessor communication in these maximal subdomain configurations is not negligible and may even dominate. A third factor, is the load imbalance caused by uneven distribution of the computational workload dictated by model physics considerations. This factor is highly dependent on machine characteristics and several approaches to balancing the code have been developed [24]. These and other solutions will likely be necessary at ultra-high resolutions but it is difficult to speculate which are most effective in the absence of a code.

Speculating about the scaling behavior of the communications costs is made difficult by the lack of a purely horizontal two-dimensional domain decomposition for fvCAM. However, some information about nearest neighbor communication in this case may be derived by measuring point-to-point communication statistics on the one dimensional decomposition with the Integrated Performance Monitor (IPM) [20] tool. The tool reveals that in the one dimensional

*Upper bound calculated as: Memory$_{n+1}$=4xMemory$_n$.

†Least squares linear fit ($y = Ax$) to the total measured memory.

decomposition the largest messages contain the ghost cells values of three concatenated 3D variables sent once per dynamics time step to the north and south neighboring domains. Approximately ten messages per dynamics time step are sent containing a single 3D variable. A number of considerably smaller messages containing 2D variables are also sent. In the two dimensional domain decomposition, the number of these messages must be doubled to include the east and west neighbors but are reduced in size. For a fixed number of cells per subdomain, message sizes are independent of horizontal resolution from geometrical considerations but are dependent on the number of domains. In contrast the one dimensional decomposition messages contain ghost cell values over the entire range of longitudes. Hence, message sizes are independent of the number of domains but dependent on horizontal resolution in this case.

Table 4 shows an estimate of the messaging requirements for fvCAM as a function of horizontal resolution and decomposition strategies. The second and third columns of Table 4 show the one-dimensional decomposition measured average message size and the largest measured message size in the nearest neighbor communication. Measurements were performed at the 50km *D* mesh and extrapolated to the other resolutions. In the two dimensional decomposition, the nearest neighbor message size is determined by the size and logical shape of the domain. For a fixed number of cells per subdomain, these messages are of a constant size. For the maximum processor case of 9 mesh cells per subdomain, the average nearest neighbor message size would be about 1.5KB and the maximum would be about 5.6KB. In the 100 horizontal cells per subdomain case considered above, the average would be about 5.0KB and the maximum would be about 62.4KB.

| | | | 100% Comm[‡] | | | | | 10% Comm[‡] | |
|---|---|---|---|---|---|---|---|---|---|
| | 1D 3-rows | | 1D 3-rows | | 2D 3x3 | 2D 10x10 | | 2D 10x10 | |
| Mesh Name | Avg Msg Size (KB) | Max Msg Size (KB) | Msgs/ Second/ Domain | Min Msg Bandwdth/ Domain/ (MB/s) | Min Msg Bandwdth/ Domain (MB/s) | Min Msg Bandwdth/ Domain (MB/s) | Max [§] Message Latency (sec) | Min Msg Bandwdth/ Domain (MB/s) | Max[§] Message Latency (sec) |
| *B* | 72.6 | 270 | 23 | 1.6 | 0.07 | 0.21 | $24ms$ | 2.1 | $2.4ms$ |
| *C* | 145 | 540 | 43 | 6.3 | 0.13 | 0.43 | $12ms$ | 4.3 | $1.2ms$ |
| *D* | 290[†] | 1078[†] | 86 | 25 | 0.26 | 0.87 | $5.8ms$ | 8.7 | $580\mu s$ |
| *E* | 581 | 2157 | 172 | 100 | 0.52 | 1.74 | $2.9ms$ | 17.4 | $290\mu s$ |
| *F* | 1161 | 4313 | 344 | 400 | 1.04 | 3.47 | $1.4ms$ | 34.7 | $140\mu s$ |
| *G* | 2323 | 8627 | 689 | 1600 | 2.08 | 6.94 | $720\mu s$ | 69.4 | $72\mu s$ |
| *H* | 4645 | 17252 | 1378 | 6400 | 4.16 | 13.99 | $360\mu s$ | 139.9 | $36\mu s$ |
| *I* | 9290 | 34504 | 2756 | 25600 | 8.33 | 27.78 | $180\mu s$ | 277.8 | $18\mu s$ |

Table 4: Estimate of messaging requirements for fvCAM as a function of horizontal resolution and decompositions. The results for D mesh are measured for one dimensional domain decomposition, all other results are inferred from these results and the simple geometrical considerations.

In the discussion above of the individual processor speed, we performed our calculations without accounting for communication costs, to obtain a lower bound on processor performance requirements. Similarly, we estimate a lower bound on the communication requirements separately from the cost of calculation. Separate treatment of these costs is a reasonable approach given the explicit message-passing communication is bulk-synchronous, where communication and computation occur in distinct phases. The total execution time of the simulation would, of course, be a mixture of these two components. Using the performance metric appropriate to climate simulation (1000 times realtime), the total rate of nearest neighbor messages per processor that must be sent and received each second must exceed the values listed in column four of Table 4 (expressed in messages/second/subdomain) for the one dimensional decomposition. As mentioned before, in the two dimensional decomposition case the number of messages per processor per dynamics time step is twice that of the one dimensional decomposition thus doubling these message rate requirements. The average nearest neighbor send/receive data volume rate per processor of these messages must exceed the values in MB/s listed in fifth column of Table 4 for the one dimensional decomposition. For the two dimensional domain decomposition strategy the average rates are listed for maximum processor case of 9 (3x3) and 100 (10x10) mesh

[†]Average and maximum message size was measured for the D mesh, all other data message sizes were inferred from this data.

[§]Maximum allowable latency to avoid latency-bound messaging.

[‡]Percent of overall runtime spent in communication phase.

cells per subdomain, in columns six and seven respectively. The total amount of nearest neighbor data that must communicated depends on the number of processors.

Up to now, our calculations have been based on the overly optimistic assumption that communication can be spread throughout the entire runtime of the calculation (denoted as *100% Comm* on the top row of Table 4). In practice, communication occurs only during a fraction of the overall runtime, thus putting more pressure on the interconnect, as messages have a shorter time period in which they must be transmitted. The ninth column of Table 4 shows the allowable messaging rate (for the 10x10 case) when communication can occur only within a pessimistic 10% of the overall runtime. As expected, the messaging requirements grow by a factor of 10x under these circumstances. Our experience with FVCAM [16] suggests that the actual fraction of time spent in communication would be somewhere between the 100% and 10% duration; we base the remainder of our interconnect analysis on this upper bound of communication requirements (*10% Comm*). Note that if the algorithm and underlying hardware allow for data exchanges which could be overlapped with computation, the interconnect requirements would decrease correspondingly [6].

Finally, it is also important that the latency of the interconnect be balanced with the communication performance, otherwise messages become latency-bound and are not able to fully exploit the available bandwidth of the interconnect. To compute the maximum allowable latency, we use the simple linear relationship (bandwidth $\times$ latency = message_size) provided by Little's Law [3], to determine the latency that can be tolerated before the effective bandwidth of the interface is cut in half (i.e. latency-bound communication). The eighth and tenth columns of Table 4 show the maximum allowable messaging latency based on the *100% Comm* and *10% Comm* metrics, respectively. Analysis shows that even under the aggressive communication requirements, the *I* mesh calculation could tolerate a maximum latency of $18\mu s$ — a readily achievable rate for modern high-performance interconnect technologies [12]. Even higher latencies would be permissible for lower resolution calculations as seen in Table 4.

# 4 Designing an HPC System for Kilometer Scale Climate Modeling

In a previous study [16] we examined the behavior of fvCAM in the 50 km scale *D* mesh version on a variety of leading computing systems in a number of different configurations. In Table 5 we summarize the best performance obtained on these and selected additional systems to begin the discussion of future computing platform requirements. Note that all of these configurations used a mixed 2D domain decomposition that exploits vertical parallelism when possible, and some limited longitudinal parallelism when not. This mixed domain decomposition does not allow the degree of parallelism that a purely horizontal 2D domain decomposition strategy would. Some configurations also permitted the use of OpenMP to further increase the maximum number of permissible processors.

| | | | fvCAM Performance | | | |
|---|---|---|---|---|---|---|
| Architecture | Clock (MHz) | Peak Proc (GF/s) | Speed/ Proc (GF/s) | Sustained % of Peak | Max Procs Used | Speedup (Simulated Time/ Real Time) |
| IBM Power3 | 375 | 0.7 | 0.05 | 3.7% | 1680 | 837 |
| IBM Power5 | 1900 | 7.6 | 0.38 | 5.0% | 672 | 2480 |
| IBM BG/L | 700 | 2.8 | 0.04 | 1.4% | 64 | 708* |
| Intel Itanium2 | 1400 | 5.6 | 0.23 | 4.1% | 644 | 1350 |
| Cray X1E | 1130 | 18.0 | 0.70 | 4.2% | 672 | 4583 |
| Earth Simulator | 1000 | 8.0 | 0.44 | 5.5% | 896 | 3604 |

Table 5: Performance of fvCAM on the *D* mesh for a variety of supercomputing platforms using the largest concurrencies available to date. BG/L results are shown for *C* mesh due to memory constraints related to I/O.

In the next section, we use currently available performance data to predict the resource requirements for future systems and to extrapolate machine performance characteristics necessary to meet those aggressive requirements. The purpose of this highly speculative study is to examine which design paths present a viable approach to achieving the sustained performance requirements dictated by a kilometer scale climate model based on limitations of component costs and power dissipation. First we consider recent changes in the microprocessor design landscape.

*BG/L results are shown for *C* mesh due to memory constraints related to I/O.

## 4.1 Computer Architecture Considerations

Figure 3 shows the improvements in processor performance as measured by the SPEC benchmark [11] over the period from 1978 to present. Since 1986 performance has improved by 52 percent per year with remarkable consistency. During that period, as processor geometries scaled according to Moore's law, the supply voltages were kept constant in order to allow manufacturers to increase clock-speeds. This approach, known as fixed-voltage scaling [13] fed the relentless increases in CPU clock-rates over the past decade and a half. However, below the 90nm scale for silicon lithography, this technique began to hit its limits [5, 9] as power density has become the dominant constraint in the design of processing elements. The result has been a stall in clock frequency that is reflected in the flattening of the performance growth rates starting in 2002. In 2006, individual processor cores are nearly a factor of three slower than if progress had continued at the historical rate of the preceding decade. The mainstream microprocessor industry has responded by halting further improvements in clock frequency and increasing the number of cores on the chip. Patterson and Hennessy [10] estimate the number of cores per chip is likely to double every 18 months henceforth.

The recent trends in the microprocessor industry have important ramifications to the design of an ultra-high resolution atmospheric model. In a one processor per subdomain formalism, the superlinear scaling of the dynamics dictates that processor speed must increase in order to utilize more horizontal domains. If clock speed stagnates, it is imperative that multiple processors be assigned to each horizontal domain in order to maintain scalability to the kilometer length scale. Hence, additional levels of parallelism, such as are already implemented in fvCAM, must be used. The tight coupling between processing cores on the same chip will greatly aid in this task.

Power is proving to be the dominant constraint of present and future generations of processing elements. The embedded processor market relies on architectural customization to meet the demanding cost and power efficiency requirements of embedded applications such as MP3-players, cell phones and PDAs. Whereas one would expect a desktop or server processor vendor to deliver a new CPU core design every 2 years, a typical embedded vendor may generate up to 200 unique chip designs every year. The basic building blocks of BG/L represent the capability IBM Microelectronics has to offer and apply dozens of different designs every year for a wide range of embedded and semicustom applications. In order to keep up with this demanding pace for semi-customized designs, leading embedded design houses such as IBM Microelectronics, Altera, and Tensilica have evolved very sophisticated toolsets to accelerate the design process through semi-automated synthesis of custom processor designs.

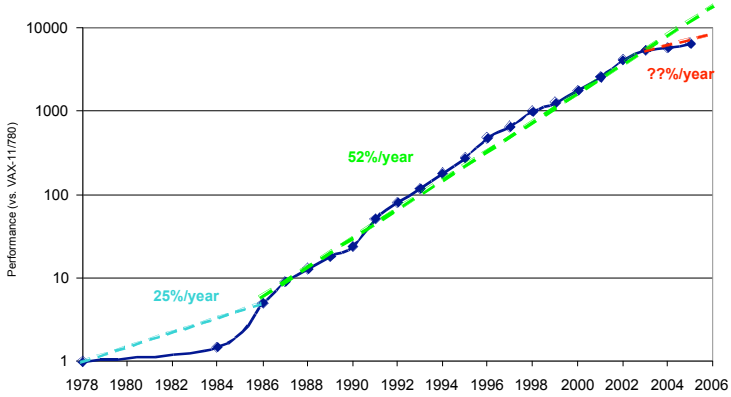



Figure 3: Processor performance improvement between 1978 and 2006 using integer SPEC programs [10]*.

Semicustom solutions to specific HPC applications based on embedded processor technology are not as fanciful as it would appear at first glance. The hardware and software are delivered together as part of an integrated solution. For instance, Tensilica delivers Open64-based compilers that generate code for the custom built processors (one compiler per chip design) as well as profiling and debugging tools that understand the customized CPU architecture. Tensilica's design tools enable semi-automated design of a customized processor core [19, 22], accurate power estimation of the synthesized core and multiprocessor system-on-chip (MPSoC) design [8], as well as system modeling capabilities that enable performance estimation for chips fabricated using existing commodity 65nm process technology. Although Tensilica is not a supercomputing vendor, they examined our design requirements and used their tools to design and

*Figure provided courtesy of David Patterson from the 4th edition of Computer Architecture: A Quantitative Approach, 4 edition, 2006.

simulate a custom hardware implementation for the kilometer scale climate model. The exercise illustrated what power efficiency benefits could be realized by taking the embedded design philosophy to its logical conclusion. The next section examines the ramifications of the embedded approach to system design in more detail.

## 4.2 Design Study

Based on the analyses above, we assume the following configuration to estimate required machine characteristics for ultra-high resolution climate simulation:

- 1.5km model (*I* mesh), 100 vertical levels.
- 1000 times faster than realtime simulation performance. As there are four times more vertical cells than analyzed above, the sustained computational performance to achieve this is approximately ten petaflops.
- *I* mesh (18432x11521x100 cells) discretized into 2 million horizontal subdomains of 10x10x100 cells which may be further subdivided in a hybrid vertical/horizontal manner.

In the previous section, we derived the fvCAM application resource requirements by extrapolating from the measured resource requirements of the climate code running at increasingly higher resolutions. Given these extrapolations, we present the minimum requirements for a system that would meet the application requirements for a 1.5km climate model simulation.

- Each horizontal subdomain must be computed at 5Gflop/s sustained in order to meet the 1000 times realtime requirement. The horizontal domains can in turn be decomposed across multiple processing elements within the socket. (For instance, if 10 processing elements are available within the socket, then each element need only supply 500Mflop/s sustained in order to meet the 1000x requirement.)
- 5GB/s local memory performance per domain in order to remain compute-bound rather than limited by memory bandwidth (1 byte per flop), to ensure sufficient memory bandwidth. (The 1 byte per flop ratio is based on our experience with the systems in Table 5.)
- The memory footprint of each subdomain is 5MB as described in Section 3.3.
- The interprocessor communication requirements are 200MB/sec to each horizontal neighbor in the north,south, east, and west dimensions (100MB/s bidirectional to each neighbor) as described in Section 3.4.
- We assume the subdomains in the vertical dimension are co-located on the same socket or SMP node in order to reflect the extremely tight-coupling of computation and communication in that dimension as described in Section 3.2.
- For inter-socket communication, we consider only the communication requirements between horizontal domains.

Under these assumptions, we compare available technology in the current generation of HPC system building blocks. We will begin with candidate microprocessor building blocks and extrapolate the scale of the system necessary to meet the requirements for the climate model under these stated conditions. We acknowledge that performance for such large-scale systems is far more complicated than simply scaling up the number of processors to meet a peak performance target. Even the most generous interpretation of these extrapolations (theoretical peak performance) merely sets the upper bound for potential system performance. The purpose of the extrapolation is to assess whether systems of the needed concurrency are even remotely feasible based on the most optimistic power, performance and cost requirements.

Table 6 shows three potential HPC design approaches for performing fvCAM simulations at 1.5km resolution using current technology solutions. Our goal is not to promote a particular vendor or solution — but rather to point out the power and cost efficiency opportunities that exist when following the path of these three divergent design philosophies. Through this comparison, we attempt to infer just how close existing technology is to meeting this ambitious goal of 1.5km climate modeling at 1000 times realtime performance.

The AMD Opteron is a popular building block for HPC systems from the commodity clusters to the tightly-integrated XT3. This solution represents the classic "commodity approach" that depends upon a processor architecture that is targeted at a diverse set of server applications that range from technical computing to web serving. The cost-efficiencies of leveraging high-volume commodity technologies is offset by the potentially lower computational efficiency of an architecture that is not as narrowly specialized to scientific application requirements. It is important to point out that the AMD specifications proposed in Table 6 solution do not include the price of the interconnect.

| Name | CPU | Clock Speed (Ghz) | Flops/ Clock/ Core | Peak Core (GF/s) | Cores/ Scket | Power/ Scket (W) | Sub-domains/ Scket | Mem/ Scket (MBs) | Mem/ BW* (GB/s) | Peak Bytes/ Flop | Netwk BW† (GB/s) | Total Scket ($10^6$) | Total Power (MW) | Cost (M$) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AMD | Opteron | 2.8 | 2 | 5.6 | 2 | 95 | 22.4 | 112 | 6.4 | 0.6 | 4.48 | 0.89 | 179 | 1800‡ |
| BG/L | PPC440 | 0.7 | 4 | 2.8 | 2 | 15 | 11.2 | 56 | 5.5 | 0.9 | 2.24 | 1.78 | 27 | 2600§ |
| Tensilica | Custom | 0.65 | 4 | 2.7 | 32 | 22 | 172.8 | 864 | 51.2 | 0.6 | 34.5 | 0.12 | 2.5 | 75¶ |

Table 6: Candidate system characteristics for 1.5km scale climate simulation based on theoretical peak performance of current technology. Under the assumptions of multi-core integration growth [10], an 8–16 fold improvement in power- and cost-effectiveness is expected by around 2011, even in the absence of clock frequency increases. These improvements would be (barely) sufficient to compensate for the low efficiency in terms of delivered sustained performance, and could deliver enough flops to effectively compute at the 1.5km resolution goal. Power calculations account for only processor and memory consumption. Cost calculations are based on publicly available component pricing, and do not include disk, system integration, software infrastructure, power, building, maintenance, etc.

BlueGene/L (BG/L) represents a more power-efficient alternative approach that achieves its performance through higher concurrency than mainstream cluster solutions. The core of BG/L system is a SOC (system on chip) based on the low-power PowerPC 440 embedded core. The PowerPC 440 is a standard in-order CPU core present in many low-power embedded applications, which is optimized for scientific applications by customizing the SOC services that are built around it on the chip. Unlike the AMD solution presented above, the BG/L SOC includes all of the logic for the interconnection network. The BG/L approach offers well-documented improvements to the power efficiency for many scientific applications; albeit the breadth of applications that are well suited for the architecture are narrower than the typical commodity cluster implementations. Note that the computed number of nodes (1.8M sockets) far exceeds the current BG/L design limitations, but may be feasible in future generation implementations.

Whereas the BG/L solution begins with a generic embedded core and customizes the SOC logic that surrounds it to build a power-efficient building block for a system, Tensilica takes the embedded design philosophy one step further: customizing the design of the processor core so that it is custom-tailored to the computational requirements of the application. Table 6 shows the Tensilica solution (calculated by Chris Rowen, Tensilica CEO) for synthesizing a custom processor around the requirements of the fvCAM specifications detailed in the previous sections. The Tensilica performance estimates are based on existing process technology and current-generation Tensilica system design tools [22]. The proposed system design includes power calculations for DC power conversion losses, integrated interconnection fabric, and the comparatively high power requirements (relative to conventional DDR) of XDR DRAM. In these design optimizations, each processor is tailored for its target application in order to optimize the consumption of transistors, and hence, power efficiency. The customized design retains all of the programmability of a conventional processor, but achieves the highest efficiency when applied to the problem for which it is designed. The Tensilica approach represents the most extreme application of architectural specialization in order to improve power efficiency. However, it is important to note that the software infrastructure to utilize a system at this scale (3.7M cores, 116K sockets) remains a daunting challenge to implement. Of course, software will also present enormous challenges for any of these proposed solutions (890K sockets for AMD or 1.78M sockets for the BG/L).

However, the situation is far worse since these calculations are based on the assumption that these systems will sustain peak performance. As shown in Table 5, current platforms achieve around 5% of peak (10% being an overly optimistic figure) — this indicates that the cost and power efficiency are likely to be 10x–20x worse than our calculations suggest. Using Patterson and Hennessy's premise that the number of cores per chip will double every 18 months [10] while holding power, clock-frequency and costs constant, we can thus extrapolate our design parameters to future generation process technologies. Under these assumptions of integration improvements, even in the absence of clock frequency increases, we expect an 8-16 fold improvement in power-efficiency and cost-effectiveness by around 2011. These improvements would be (barely) sufficient to compensate for the low efficiency in terms of sustained

*Minimum required local memory bandwidth per socket.

†Minimum required nearest neighbor interconnect bandwidth per socket (see Table 4).

‡Cost of node and memory only, does not include interconnect is estimated conservatively at $2,000 per node based on current price quotes from Sun and CDW.

§News articles on BG/L cite a minimum cost of $1.5M per rack of 1024 nodes [4].

¶Cost of custom chips design and fabrication, memory, raw hardware for the system, and interconnect as estimated by Tensilica.

performance, and could deliver enough flops to effectively compute at the 1.5km resolution goal. The percentage of theoretical peak speed achieved in climate models is notoriously low [16]. A semi-custom processor core design might yield better results if the mix of compute loops in the model possesses some common features that could be identified and optimized upon. Given the complexity of climate model algorithms, this is an open question. However, we again highlight, that the software challenges (at all levels) remain a tremendous obstacle to effectively utilizing these levels of concurrency.

Computing the total cost of ownership of these systems is nearly impossible, however we can explore the lower bound of hardware acquisition in today's dollars and technology using publicly available data. Assuming dual-processor, dual-core 1U nodes at an (optimistic) cost of $2,000/node, the AMD system would cost $1.8B without considering for the cost of disks, interconnect, or software infrastructure.* For BG/L we used the minimum per-rack cost described in press releases and articles about the system [4]. The BG/L estimated cost of $2.6B includes the cost of the interconnect since it is integrated on the processor/node SOC. On the other hand, the total cost of the Tensilica solution is estimated be $75M for the hardware including the costs designing the custom chip, fabrication, and raw hardware for the system as well as the interconnect. Of course the published list prices would be significantly discounted for large AMD or BG/L systems configurations, but these extrapolated figures present us with a baseline for first-order cost comparisons.

These figures combined with the enormous power efficiency of custom-tailored embedded solutions (see Table 6) point to a promising potential future direction of ultra-scale HPC system design. Opportunities exist to apply the embedded processor design techniques to tailor a machine to the problem. Although such an approach would be exceedingly ambitious, it offers a compelling opportunity to develop a climate modeling system capable of a 1.5km resolution within a believable cost and power-consumption envelope in the foreseeable future. However, the more aggressive architectural customization design approach offered by vendors such as Tensilica, suffers from being narrower compared to the typical datacenter requirements where systems are procured to serve a broad workload. Thus, this approach may only be appropriate for an HPC center focusing on a comparatively narrow climate modeling or weather prediction workload.

# 5 Implications of Assumptions on Model and Architectural Analyses

The extrapolation of code performance issues in an AGCM to a kilometer scale horizontal resolution requires many assumptions. These include assumptions regarding both model physics as well as code structure. Similarly, assumptions must be made to extrapolate current technologies to the scale necessary to carry out such integrations in a practical fashion. We comment on these assumptions and their impact on our analysis in this section.

Similar to many AGCMs, a hydrostatic approximation is made in the formulation of the equations of motion in fvCAM. Under this approximation, the equation for the vertical velocity may be simplified such that it is diagnosed rather than prognosticated from the solutions of the horizontal velocity components. This approximation is valid when the horizontal scale of the model is large compared to the vertical scale due to the highly stratified nature of the atmosphere. However, these conditions are not true at the resolutions extrapolated to here. Hence, a fully non-hydrostatic approach must be taken. This has the effect of adding another difference equation for the now prognostic variable of vertical wind speed. Vertical dependencies do not complicate the parallelization any more than they do in the physics section of the model. The additional computational burden is not particularly large relative to the remaining parts of the dynamics algorithm so the results presented above would be altered only slightly and the scaling behavior would remain the same.

Of more serious concern to the dynamics portion of the code is the rather extreme cell aspect ratio caused by the convergence of meridians at the poles. Because of the rather poor scaling behavior of the filters, it is surprising that at the kilometer scale the arithmetic cost of these filters does not dominate the entire calculation. This suggests that the familiar latitude-longitude discretization might even be practical at these resolutions. However, other factors complicate the matter. The aspect ratio for the cells nearest the pole approaches $M/2\pi$. A time step chosen to meet stability criterion at the mid-latitudes causes the Courant number in the polar cells to increase linearly with the horizontal resolution. The solution in the explicit portions of the code may not be particularly accurate but at least it may be easily calculated via nearest neighbor interprocessor communication and the aid of the nonlocal filters

*Using Sun's internet store [21], we configured a SunFire x4100 server with one single-core AMD Opteron processor and the minimum selectable quantity of memory (1Gigabyte) and no disks. The price came to $1,999 as of June 7, 2006. A dual-core processor (AMD model 275 for $750) cost more than twice that of a single-core model 248 in our configuration ($280) and the target configuration would require two of them, thereby increasing costs by at least $1,220.

described above. However, the semi-Lagrangian advection of tracers is more problematic in this vicinity. As the Courant number increases far past unity, information from increasing number of cells is required to permit solution.

For the two dimensional horizontal domain decomposition strategy, the implication is that the number of ghost cells in the longitudinal direction must increase as subdomains approach the pole. In very severe aspect ratio cases, the number of ghost cells could be larger than the number of real cells in a subdomain hence requiring interprocessor communication from more distant neighbors if the time step remains fixed. Indeed, one could mitigate this extra communication by reducing the advection time step towards the poles or even eliminate the issue entirely by replacing the semi-Lagrangian transport algorithm with a filtered explicit scheme and its attendant nonlocal communication. In any event, the extra arithmetic as well as load balancing issues would cause the overall performance to be worse than we estimated above. The best solution may be to abandon the latitude-longitude mesh in favor of more uniform discretization strategies such as icosahedral meshes [17] or cubed-sphere approaches [2]. The scaling behavior of the dynamics and physics portions of alternative mesh GCMs would be the same as for the latitude-longitude mesh models as the Courant condition still applies. Hence, the analysis of computational costs presented will not significantly change as implementations of alternative mesh strategies are cost competitive at current resolutions.

In extrapolating the computational cost of the physics portion of the model, it was assumed that there was no change in either the individual parameterizations or the physics time step. Although we have run the model at resolutions ranging from the $B$ mesh (200km) to the $E$ mesh (25km) with minimal modifications and producing reasonable climatologies, neither of these assumptions holds at scales finer than ten kilometers. In fact, the main point to kilometer scale atmospheric modeling is to resolve processes that are currently parameterized, thus necessitating significant modifications to the physics portion of the model and increasing the computational burden.

For instance, the replacement of the deep convective parameterizations with cloud system resolving submodels will add multiple prognostic equations to solve. Also, at these resolutions, wide angle scattering of solar radiation could transport energy horizontally from one cell to a neighbor. Multi-angle radiation transport algorithms will be necessary to address this process. Additional computational burden also arises because the physics time-stepping algorithm must be dramatically altered to represent such microphysical detail. For instance, current cloud system resolving models operate at a ten second time step [15]. For comparison, the mid-latitude Courant limit on the 1.5 km $I$ mesh extrapolates to about three seconds. On the other hand, not all processes must be resolved at this detail. The diurnal cycle remains at 24 hours regardless of resolution. Hence, even significantly more sophisticated radiation transport schemes could probably be substantially subcycled with time steps of multiple minutes. The central conclusion of this paper, that the solution of the dynamics portions of atmospheric models at ultra-high resolution dominates the computational cost, remains unchanged upon consideration of all these factors.

Horizontal decomposition of grid-based atmospheric general circulation models is well established on both regular and irregular grids. The additional parallelism demanded by multi-core processors is less well developed. Decomposing in the vertical dimension is appropriate in much but not all of the dynamics portion of the code. This is less true in the physics portion, especially in the radiation transport and cloud physics routines where a tight coupling of the vertical dimension is required. In the sections of the code where vertical decomposition is inappropriate, further horizontal decomposition is possible. Alternating between two parallelization strategies is only feasible if communication bandwidth is very high. However, this is what we expect for the processors within a socket. Whether the communications would be implemented by message passing, at the loop level or a combination of both is an open question.

Finally, an atmospheric general circulation model is but one component in a fully coupled climate system model. Increases in resolution of the other components should be commensurate with those imposed on the atmosphere. Increased sophistication of the other components is expected both independent of and as a result of such resolution increases. Two areas where computational considerations will be important are the ocean component and atmospheric chemistry. In current ocean models, dynamics is the dominant source of computational burden and is expected to remain so with resolution increases. The arguments we present for the atmospheric model translate directly to the ocean. The only significant difference is that the state of the art in oceanic modeling is significantly closer to 1km. Global atmospheric chemistry models are developing rapidly and present interesting questions in the context of this study. Composed of both potentially stiff chemical rate equations and advective transport equations, the mix of computation load could be a very complex function of resolution. The advection timestep depends on resolution so will exhibit the same superlinear scaling as the general circulation. The rate equations are local and would not be as sensitive to resolution issues. However, there are instances where sharp gradients of chemical constituents are important, complicating the matter. Nonetheless, we fully expect that if millions of processors can be successfully applied to the atmospheric general circulation similar efforts would be successful with other components of a fully coupled climate system model.

# 6 Conclusion

We present a speculative extrapolation of the performance aspects of an atmospheric general circulation model at ultra-high resolutions and offer a technological path to realize integration of such a model in the relatively near future. We estimate that a credible kilometer scale atmospheric model would require at least a sustained ten petaflop computer to provide scientifically useful climate simulations. Numerical weather prediction needs at a global kilometer scale could be met at a sustained one hundred Teraflops. Construction of a machine capable of sustaining ten petaflops will be a formidable task stretching both budgets and power infrastructure. We explored the possibility of a customized architecture design that is considerably more favorable in both these aspects as compared to mainstream designs. Software infrastructure issues will present a daunting challenge regardless of architectural path chosen.

Due to the stability condition dictated by an explicit solution of the Navier-Stokes equations, solution of the dynamics portion of the model dominates the computational burden at ultra-high resolutions. The associated superlinear scaling with horizontal resolution of this computational burden has significant implications in the design of machines useful to such climate and weather problems. For instance, atmospheric models can efficiently utilize more processors by increasing horizontal resolution but the time to solution will increase if the processor speed is not increased at the same rate. The stability criterion also dictates differences between the scaling behavior of the total operation count and the total memory requirements that compare favorably with the current hardware design trend of decreasing memory per processor.

While the degree of parallelism at current resolutions is fairly low, systems constructed of millions of processors could be exploited at horizontal scales approaching one kilometer. However, recent technological trends indicate that the rate of processor clock speed changes is slowing. Moore's Law regarding the density of transistors is expected to hold through the next decade [5, 9] but due to power considerations chip designers are opting to increase overall per-socket performance by adding more processing cores rather than increasing clock speed [10]. These architectural trends towards increased parallelism and multi-core chip designs are driven primarily by power-efficiency considerations. Our design study shows that further power-efficiency gains can be realized through customized processor design in line with the embedded-systems design philosophy. This portends an alternative approach to high end computing system design where efficiency can be gained through hardware designs that are customized around the application rather than the other way around. While the costs of custom hardware design may not be cost-effective for the mid-range problems, the approach may prove essential for the handful of applications that are poised to take advantage of future Petaflop scale systems. Without detailed attention to power efficiency concerns, the energy costs for operating such systems is likely to create a hard ceiling for practical ultra-scale computing in the future.

Exploitation of such multi-core processors in an ultra-high resolution atmospheric general circulation model will require not only a two dimensional horizontal decomposition containing millions of subdomains, but also additional levels of parallelism. The tight coupling of multi-core processors to the entire socket's local memory permits a variety of options. We envision a parallelization strategy where a two dimensional horizontal decomposition is distributed among multi-core sockets. Within the horizontal subdomain, additional parallelization would be obtained by alternating between a vertical decomposition and a horizontal decomposition that might be implemented with message passing, at the loop level or a mixture of both techniques.

Experience reveals that new versions of coupled climate models can take a large team several years to develop into useful scientific tools. The major conceptual changes required by a kilometer scale climate model are certain to be difficult to implement. Although the hardware, software, and algorithms are all equally critical in conducting ultra-high climate resolution studies, it is likely that the necessary petaflop computing technology will be available in advance of a credible kilometer scale climate model.

# 7 Acknowledgments

We thank David Parks (NEC Corp.) for the Earth Simulator results and Yu-Heng Tseng (National Taiwan University) for the BG/L benchmark results. We thank Art Mirin (LLNL) and Pat Worley (ORNL) for valuable comments about the manuscript. We also thank Chris Rowen, CEO of Tensilica, for using his company's tools to synthesize and model a system designed around our specifications. All authors from LBNL were supported by the Office of Advanced Scientific Computing Research or the Climate Change Prediction Program under the Office of Biological and Environmental Research in the Department of Energy Office of Science under contract number DE-AC02-05CH11231.

# References


[1] A. Adcroft, J-M Campin, C. Hill, and J. Marshall. Computational design of the basic dynamical process of the ucla general circulation model. *Meth. Comput. Phys.*, 17, Academic Press:173–265, 1977.

[2] A. Adcroft, J-M Campin, C. Hill, and J. Marshall. Implementation of an atmosphere-ocean general circulation model on the expanded spherical cube. *Mon. Wea. Rev.*, 132 (12), 2004.

[3] D. Bailey. Littles law and high performance computing. RNR Technical Report, Sep. 1997.

[4] Cost of bluegene l. `http://news.zdnet.com/2100-9584_22-5442285.html`.

[5] S. Borkar. Design challenges of technology scaling. *IEEE Micro*, 19(4):23–29, Jul-Aug 1999.

[6] P. Hargrove C. Iancu, P. Husbands. HUNTing the Overlap. In *Proc. of 14th International Conference on Parallel Architectures and Compilation Techniques (PACT)*, Saint Louis, Missouri, 2005.

[7] W. Dannevik. Massively parallel computing and large eddy simulation in geophysical fluid dynamics. *Large Eddy Simulation of Complex Engineering and Geophysical Flows*, 1993.

[8] Y. Fei, S. Ravi, A. Raghunathan, and N.K. Jha. A hybrid energy-estimation technique for extensible processors. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 23:652– 664, May 2004.

[9] P. P. Gelsinger. Microprocessors for the new millennium: Challenges, opportunities, and new frontiers. In *International Solid State Circuits Conference, ISSCC*, pages 22–25, 2001.

[10] John L. Hennessy and David A. Patterson. *Computer Architecture : A Quantitative Approach; fourth edition*. Morgan Kaufmann, San Francisco, 2006.

[11] John L. Henning. PSPEC CPU2000: Measuring CPU Performance in the New Millennium. *IEEE Computer*, pages 28–35, 2000. `http://www.spec.org`.

[12] HPC challenge benchmark. `http://icl.cs.utk.edu/hpcc/index.html`.

[13] Borivoje Nikolic Jan M. Rabaey, Anantha Chandrakasan. Integrated circuits, a design perspective. In *Prentice Hall, Second Edition*, 2003.

[14] S.-J. Lin and R. B. Rood. An explicit flux-form semi-Lagrangian shallow-water model on the sphere. *Q. J. R. Meteorological society*, 123, 1997.

[15] M.F.Khairoutdinov and D.A. Randall. Cloud-resolving modeling of the ARM summer 1997 IOP: Model formulation, results, uncertainties and sensitivities. *J. Atmos. Sci.*, 60:607–625, 2003.

[16] L. Oliker, J. Carter, M. Wehner, A. Canning, et al. Leading computational methods on scalar and vector hec platforms. In *Proc. SC2005*, 2005.

[17] D.A. Randall, T. D. Ringler, R. P. Heikes, P. Jones, and J. Baumgardner. Climate modeling with spherical geodesic grids. *Computing in Science and Engr.*, 4:32–41, 2002.

[18] David Randall. Counting the clouds. 2005 SciDAC meeting, San Francisco, CA, June 26-30. `http://www.itjungle.com/tlb/tlb031505-story03.html`.

[19] C. Rowen and S. Leibson. *Engineering the Complex SOC*. Prentice Hall, 2004.

[20] D. Skinner. Integrated Performance Monitoring: A portable profiling infrastructure for parallel applications. In *Proc. ISC2005: International Supercomputing Conference*, Heidelberg, Germany, June 21-24, 2005.

[21] Sun's internet store. `http://store.sun.com`.

[22] Tensilica xtensa design tools. `http://www.tensilica.com/pdf/Tools_white_paper_final-1.pdf`.

[23] M. F. Wehner, J. J. Ambrosiano, J. C. Brown, W. P. Dannevik, P. G. Eltgroth, et al. Toward a high performance distributed memory climate model. In *Proc. 2nd Intern. Sympos. on High Performance Distributed Computing (HPDC2)*, Spokane, WA, July 20–23, 1993.

[24] P. H. Worley and J. B. Drake. Performance portability in the physical parameterizations of the community atmospheric model. *International Journal for High Performance Computer Applications*, 19(3):187–202, 2005.