The Computational Complexity, Parallel Scalability, and Performance of Atmospheric Data Assimilation Algorithms

P.M. Lyster*, J. Guo[‡], T. Clune[†], and J.W. Larson**

NASA Data Assimilation Office (DAO), Goddard Laboratory for Atmospheres

lys@dao.gsfc.nasa.gov

Additional affiliations:

**Earth System Science Interdisciplinary Center and Department of Meteorology, University of Maryland, College Park, MD 20742*

[‡] *Science Applications International Corporation/General Sciences Operation, 4600 Powder Mill Road, Beltsville, MD 20705*

[†] *Silicon Graphics Inc., and NASA Center for Computational Sciences*

** *Mathematics and Computer Science Division, Argonne National Laboratory*

September 17, 2001

Submitted to Computers in Science and Engineering

Abstract

The computational complexity of algorithms for Four Dimensional Data Assimilation (4DDA) at NASA's Data Assimilation Office (DAO) is discussed. In 4DDA, observations are assimilated with the output of a dynamical model to generate best-estimates of the states of the system. It is thus a mapping problem, whereby scattered observations are converted into regular accurate maps of wind, temperature, moisture and other variables. The DAO is developing and using 4DDA algorithms that provide these datasets, or analyses, in support of Earth System Science research. Two large-scale algorithms are discussed. The first approach, the Goddard Earth Observing System Data Assimilation System (GEOS DAS), uses an atmospheric general circulation model (GCM) and an observation-space based analysis system, the Physical-space Statistical Analysis System (PSAS). GEOS DAS is very similar to global meteorological weather forecasting data assimilation systems, but is used at NASA for climate research. Systems of this size typically run at between 1 and 20 gigaflop/s. The second approach, the Kalman filter, uses a more consistent algorithm to determine the forecast error covariance matrix than does GEOS DAS. For atmospheric assimilation, the gridded dynamical fields typically have more than 10^6 variables, therefore the full error covariance matrix may be in excess of a teraword. For the Kalman filter this problem can easily scale to petaflop/s proportions. We discuss the computational complexity of GEOS DAS and our implementation of the Kalman filter. We also discuss and quantify some of the technical issues and limitations in developing efficient, in terms of wall clock time, and scalable parallel implementations of the algorithms.

1 Four Dimensional Data Assimilation

Four Dimensional Data Assimilation (4DDA) is the process of combining observations with a dynamical model to generate a gridded best estimate, or analysis, of the state of the system (Daley 1991). It is thus a mapping problem, whereby scattered observations are converted into accurate maps of wind, temperature, moisture and other variables. This is shown schematically in Figure 1. The model propagates in time the estimate of the state, e.g., for the global atmosphere we use a general circulation model (GCM). The analysis is a statistics-based algorithm for combining the model output, or forecast, with observations to produce the best estimate state (the expression "analysis" is used in a context dependent manner to refer both to the algorithm for data assimilation and the resulting dataset). This is a cycled algorithm whereby the analysis state is used to reinitialize the model, and so on. 4DDA is used in weather forecasting to initialize model forecasts, for example, at the National Centers for Environmental Prediction (NCEP) (Parrish and Derber 1992, Parrish et al. 1997), and at the European Center for Medium-Range Weather Forecasts (ECMWF) (Courtier et al. 1998, Rabier et al. 1998, Andersson et al. 1998). 4DDA is also used to perform reanalyses of past datasets to obtain consistent, gridded, best estimates of the state variables of the atmosphere (e.g., wind, temperature, moisture ...), for example, at NASA's Data Assimilation Office (DAO) (Schubert et al. 1993, 1995), at NCEP (Kalnay et al. 1996, Kanamitsu et al. 1999, Kistler et al. 2000), and at ECMWF (Gibson et al 1997). These gridded reanalysis datasets are a valuable resource for the Earth Science research community (DAO 2000). The DAO develops and uses software for scientific research on methodologies for data assimilation; to run as production 4DDA systems and provide background gridded fields in near real-time support of satellite and aircraft missions; and to run as 4DDA scientific reanalysis systems in support of Earth Science research. Input atmospheric observations may be a combination of wind, height, moisture and other constituent gas variables from conventional surface and balloon instruments, plus processed observations, or retrievals, from satellite-borne instruments. The observations and the system that manages them daily by the World Weather Watch under the World Meteorological Organization are described in the review article by Atlas (1997), and some parameters of the observation datasets and analysis fields are described in Section 2.3.

This paper discusses the computational complexity of two important 4DDA algorithms in use at the DAO. The first is Goddard Earth Observing System Data Assimilation System (GEOS DAS) which uses a grid-point based atmospheric general circulation model (GCM) and an observation-space based analysis system, the Physical-space Statistical Analysis System (PSAS). GEOS DAS is very similar to global weather forecasting algorithms, where the analysis fields are used to initialize the GCM for a model forecast. In order to obtain an accurate analysis it is important to determine the appropriate balance between forecast and observations. This is achieved through the use of forecast and observation error covariance matrices P^f and R respectively. The PSAS uses modeled error covariance matrices whose parameters are determined from prior statistics with appropriate simplifying assumptions such as stationarity (Daley 1991). Global 4DDA systems such as GEOS DAS with model grids of the order 100 km have about 10^6 variables. Whether they are used for real-time weather forecasting or to create archive analysis files for research they typically run between 1 and 20 gigaflops/s on parallel computers. The software for the developmental and operational GEOS DAS is constantly changing. In this paper we will provide detailed timings for the baseline shared-memory multitasking parallel GEOS-2 DAS, along with discussion of the scalability for higher-resolution and otherwise modified software. The second algorithm is the Kalman filter, which offers the promise of more accurate analyses because it evolves P^f in a dynamically consistent manner. However P^f is of dimension the square of the number of model variables, so the algorithm could easily scale to petaflop/s proportions. A two-dimensional (latitude-longitude) Kalman filter for the assimilation of constituent gas mixing ratio in the stratosphere was developed by our group as a prototype and research tool (Lyster et al. 1997, Ménard et al. 2000a,b). Some of the results of this work are used to extrapolate to the complexity of a full Kalman filter with three-dimensional meteorological fields.

Where appropriate, estimates of actual floating point counts are calculated; however where this is too difficult or vague we simply specify the order $\mathcal{O}$ scaling. The computational complexity of different algorithms cannot be compared without careful specification of the spatio-temporal problem domains. In this paper we will state when we use two- or three- spatial dimensions. We use the notation $[0, T]$ to specify a fixed time interval. Beyond these, the computational complexity depends on a combination of numerical and physical parameters, including the number of gridpoints in the model (n), the number of observations in an assimilation cycle (p), as well as numerical parameters defined in the text. The GCM and PSAS have tightly-coupled core algorithms with computational and communication-intensive parallel implementations; these are hydrodynamic transport (GCM) and non-sparse large matrix-vector multiplications (PSAS). Technical issues and limitations in developing efficient, in terms of wall clock time, and scalable distributed-memory parallel implementations of the GCM and PSAS, and by extension GEOS DAS, will be discussed in Section 4.

This paper does not address the issue of software complexity. This is emerging as a key issue because of the need to build extensible, maintainable, and reusable code, and because of the difficulties in managing large software projects – the core GEOS DAS algorithm is in excess of 150,000 lines of code and is being used and modified by about 100 staff members. See documentation on the DAO Office-Note pages (DAO 2000), in particular to papers by Guo et al. 1998, and Larson et al. 1998. We will also not discuss in depth the end-to-end distributed heterogeneous computing system (in excess of 100,000 lines of code) that pre- and post-processes observational data and gridded data. See the DAO web pages and the DAO Algorithm Theoretical Basis Document (DAO 2000). Finally, centers such as NCEP and ECMWF have three and four-dimensional data assimilation systems (3DVAR and 4DVAR) in their software suites. The computational complexity of GEOS DAS is similar to that of 3DVAR; however, the complexity of 4DVAR is very different and is not discussed here.

2 Goddard Earth Observing System Data Assimilation System (GEOS DAS)

Derivations of analysis algorithms abound (Daley 1991). We will motivate briefly and derive the analysis equations for GEOS DAS based on a statistical least squares approach. Cohn (1997) places this discusion in the context of general filtering methods. The optimal estimate of the state is the value of the control variable w that minimizes the cost function J:

$$J(w) = \frac{1}{2}[(w^f - w)^T (P^f)^{-1} (w^f - w) + (w^o - Hw)^T R^{-1} (w^o - Hw)] \qquad (1)$$

where

- w is the control vector of state variables ($\in I\!R^n$, i.e., an n vector).

- w^f is the state forecast ($\in \mathbb{R}^n$, i.e., an n vector).
- w^o is a vector of observations ($\in \mathbb{R}^p$, i.e., a p vector).
- P^f is the $(n \times n)$ known forecast error covariance matrix.
- R is the $(p \times p)$ known observation error covariance matrix.
- H is the (here linearized) forward operator that models the observations by acting on the state vector (e.g., if the observations come from direct measurements of the state then H can be implemented by interpolation from the state grid to the observation locations).

The value of w that minimizes J is:

$$w^a = w^f + K(w^o - Hw^f), \tag{2}$$

where the Kalman gain is

$$K = P^f H^T (HP^f H^T + R)^{-1}. \tag{3}$$

GEOS DAS uses a six-hour window $[0, 6hr]$ for the cycle that is shown schematically in Figure 1. Starting from a prior analysis, the GCM generates a forecast by iterating a timestepping algorithm:

$$w^f_{k+1} = \mathcal{M}_k w^f_k, \tag{4}$$

where k is a time index and $\mathcal{M}_k$ is the model operator. By convention (e.g., Daley 1991, DAO 2000), the forecast for each data assimilation cycle ends at (0, 6, 12, 18) hours GMT. GEOS DAS evaluates Eqn. (2) for each of these six-hourly forecasts using data that are accumulated +/- 3 hours (i.e., evenly) about the forecast time. Operational algorithms at weather centers and laboratories (Daley 1991) have more constraints and attributes than the simple form of Eqns. (1), (2), and (3). Indeed GEOS DAS uses a slight modifiction of the cycling method just described, which involves nine hours of model iteration for six-hourly each data assimilation cycle (Bloom et al. 1996). However, these caveats do not substantially modify the evaluation of computational complexity in the present work.

2.1 The Computational Algorithm for GEOS DAS

We describe the complexity and timing profile for a baseline version GEOS-2 of the GEOS DAS. The GEOS-2 GCM (Takacs et al. 1994) comprises a spatial fourth-order-accurate finite-difference dynamical core to model hydrodynamical processes, plus physics components for moist convection, turbulence, and shortwave and longwave radiation. The state, or prognostic, variables are horizontal winds, potential temperature, specific humidity, and surface pressure. There is also capability to model an arbitrary number of passive tracers. A high-latitude spectral filter and a global Shapiro filter and polar rotation algorithm provide smoothing and numerical stability. GEOS-2 GCM used a baseline model resolution of 2^o longitude, 2.5^o latitude, and 70 vertical levels. This corresponds to three-dimensional fields with horizontal resolution 91 gridpoints in latitude and 144 gridpoints in longitude. GEOS-2 GCM uses a multiple time scale computational technique (Brackbill and Cohen 1985). The dynamical core has the smallest timestep of 3 minutes at baseline resolution. The physics components generate time tendencies at longer intervals: moist convection 10 minutes, turbulence 30 minutes, shortwave radiation 1 hour, and longwave radiation

3 hours. These tendencies are applied to the state variables incrementally at the shortest timescale (3 minutes). Fuller details are described in Takacs et al. (1994), and the next Sections will discuss the complexity and timing profile of the GCM in the context of the whole data assimilation system. The number of state variables at the baseline resolution is approximately $n \approx 3 \times 91 \times 144 \times 70 + 91 \times 144 \approx 2.6 \times 10^6$, corresponding to the 3 upper-air (i.e., three-dimensional) field arrays and 1 surface (i.e., two-dimensional) field array, although in practice up to 14 upper-air field arrays are carried by the algorithm.

Currently, the GCM is run with $1^o \times 1^o \times 48$ levels, and developmental versions achieve even higher resolution. An extensive land-surface model with associated prognostic variables has also been implemented in the GCM, but we will not include that in the baseline numbers. The actual resolution is not critical to this paper, which discusses scaling properties starting from the baseline resolution of the GEOS-2 DAS. Note also that this is not the same model as the finite-volume fvGCM that is being developed for the new generation data assimilation system at the DAO. Between these two GCMs some general quantities, such as asymptotic scalability, may be similar but specific values of quantities like the model timestep or wall-clock time of runs are different.

The algorithm for solving Eqn. (2), i.e., the analysis in Figure 1, is the Physical-space Statistical Analysis System (PSAS) (Cohn et al. 1998). This solves:

$$(HP^fH^T + R)x = w^o - Hw^f, \tag{5}$$

and

$$w^a - w^f = P^fH^Tx. \tag{6}$$

The time subscript k will be dropped where it is not important to the discussion. The right hand side of Eqn. (5) is sometimes called the "observed minus forecast residual" or the "innovation", and $HP^fH^T + R$ is called the "innovation matrix". To generate the analysis fields at the end of each six-hourly cycle, GEOS-2 DAS adds the "analysis increment" $w^a - w^f$ incrementally to the state variables in a similar way as the physics tendencies are applied as described above (Takacs et al. 1994, Bloom et al. 1996). The error covariance matrices P^f and R are implemented using models for variances and correlations whose parameters are obtained from prior statistics and simplifying assumptions such as stationarity (Daley 1991, DAO 2000). Sophisticated multivariate formulations are used to improve the quality of the analysis (Guo et al. 1998). Although this has significant impact on the software complexity (Larson et al. 1998) it has only a secondary impact on the computational complexity and will not be considered here. The resulting matrices $HP^fH^T + R$ and P^fH^T are in principle dense, however correlation models with compact support (Gaspari and Cohn 1999) are used, which reduces the computational complexity by setting the correlation to zero beyond a fixed length. As described above, Eqns. (5) and (6) are solved for data that are aggregated over six-hourly intervals. This interval will be shortened to make better use of asynoptic observations (e.g., retrievals from satellites) and accommodate shorter temporal and spatial scales of high-resolution GCMs, but the numbers in this paper refer to baseline GEOS-2 DAS with a six-hour analysis interval. For atmospheric assimilation there are typically $p \sim 10^5$ observations that are made world wide in this interval. The PSAS consists of solving one $p \times p$ linear system (Eqn. 5) for the intermediate vector x using a parallel nested-preconditioned conjugate gradient solver (Cohn et al. 1998, Golub and van Loan 1989, PSAS 1998). Machine-precision solutions for x are not required because the analysis increment $w^a - w^f$ is a first order error statistic. From experience, we find that $\mathcal{N}_i \approx 10$ iterations of the outer loop of the solver provides a satisfactory solution; this reduces the residual of the solver by about an order of magnitude.

2.2 The Computational Complexity of GEOS DAS

We will not calculate the actual floating point operations of GEOS-2 GCM, but rather note here the properties of scaling with respect to spatial and temporal resolution (Takacs 1997). In Section 2.3 we will tabulate the timing profile of components of the GCM in the context of the whole data assimilation system, and then in Section 4 make some general comments about the parallel scalability of distributed-memory parallel implementations of grid-point GCMs, the PSAS, and GEOS DAS. First, we specify separately the number of gridpoints in the longitude, latitude, and vertical coordinates as n_x, n_y, and n_z respectively (i.e., $n = n_x n_y n_z$). The complexity of all four of the dynamics, moist convection, turbulence, and radiation components scale as $\mathcal{O}(n_x n_y)$. In any fixed interval $[0, T]$ the complexity of the dynamics has an additional dependence on the number of timesteps. Generally the number of timesteps of the dynamics, i.e., the temporal resolution, increases in proportion to the horizontal resolution, n_x. As the update interval of the physics components is shortened there will be an additional impact on complexity (Takacs 1997). The complexity of the dynamics, moist convection, and turbulence components scale as n_z, while the radiation scales as n_z^2. Asymptotically, for a fixed time interval the complexity of the dynamics scales as $\mathcal{O}(n^{4/3})$. Thus, if the resolution of the GCM is doubled in all three dimensions the complexity of the dynamics increases sixteen fold. The memory requirement for the GCM scales as n; thus the memory requirement in general scales less rapidly than the computational complexity. These asymptotic calculations help specify the size of computing requirements in a ten year or longer timeframe, however they can be misleading when applied to real developmental or production software in use today where, for example, there may be parameter regimes where the timestep does not need to be reduced in proportion to the horizontal resolution. In this case, it is important to instrument and generate timing profiles of the algorithms (Takacs 1997). The next section will present the timing profile for the GEOS DAS and its components.

For the PSAS, the solver, Eqn. (5), has complexity $f\mathcal{N}_i s p^2$, where $s \approx 0.40$ is the density (fraction of non-zero elements) of the innovation matrix resulting from the use of a correlation function with compact support of 6000 km. The factor f equals two plus the number of floating-point operations required to form each element of the matrix. The baseline version of the PSAS in GEOS-2 re-calculates the matrix elements using pre-calculated lookup tables during the conjugate gradient iteration cycle thus reducing the overall memory requirement and allowing for scalability to larger numbers of observations beyond the current values (Guo et al. 1998, Larson et al. 1998). Therefore f may be as high as 10, but for cache-based computers (i.e., the majority of modern parallel computers) the exact value depends on the optimization of the access to the tables (Lyster et al. 2000b). The complexity of the preconditioners are neglected here. Eqn. (6) evaluates the analysis increment, and this has complexity $fsnp$. The analysis increment is evaluated on a $2.5^o \times 2^o \times 14$ level grid and these fields are interpolated to the model GCM grid. For the baseline GEOS-2 DAS this means that the vertical coordinate systems are interpolated from 14 to 70 levels. Note that because the GCM and PSAS use different resolution grids the values of n are context-dependent in the complexity formulae.

The baseline GEOS-2 DAS used a six-hour analysis cycle (Figure 1), with $p \approx 10^5$ observations accumulated evenly about the analysis time, as described above. The analysis cycle can be made shorter, potentially leading to a more accurate algorithm, and this is an area of ongoing research. In Section 3 this will be discussed in the context of the Kalman filter. For now, note that as the analysis cycle time is reduced the computational complexity of the analysis Eqn. (6) for the interval $[0, 6hr]$

remains fixed at $fsnp$. However, for this fixed interval the complexity of the solver, Eqn. (5), will be reduced to approximately $\mathcal{N}_t f \mathcal{N}_i s (p/\mathcal{N}_t)^2 = f \mathcal{N}_i s p^2 / \mathcal{N}_t$, where $\mathcal{N}_t$ is the number of analysis cycles in $[0, 6hr]$. Thus, if the analysis cycle time were reduced to the three minute timestep of the model dynamics for baseline GEOS-2 GCM, the complexity of the analysis solver would be reduced by a factor of $\mathcal{N}_t = 120$. In the following section we show that for the baseline GEOS-2 DAS, the implementations of Eqns. (5) and (6) contributes to the computational complexity of the PSAS in the ratio 35:62. Thus reducing the analysis cycle time can reduce the overall complexity significantly, but the steady, inevitable increase in the number of available observations will counteract this.

2.3 The Timing Profile of GEOS DAS

The baseline GEOS-2 DAS uses shared-memory multitasking parallelism on Cray J series and SGI Origin computers. The technical issues and limitations in developing efficient, in terms of wall clock time, and scalable distributed-memory parallel implementations of the GCM and PSAS, and by extension GEOS DAS, will be discussed in Section 4. In this section we discuss the timing profile of shared-memory parallel GEOS-2 DAS.

Table 1 shows the percentage of time taken by the top-level components of the baseline GEOS-2 DAS run on 8 processors of an SGI Origin 2000. Note that the time taken for the Diagnostics involves the CPU time to accumulate and process three-dimensional arrays and the time to write data to disk. The Interface time accounts for the input and inital processing of the ($p \approx 10^5$) observations, plus the Quality Control component, which culls *a priori* unreliable observations (e.g., those observations whose locations or values are in gross error). The GCM, PSAS, Diagnostics, and Interface software make substantial use of shared-memory multitasking parallelism. Overall, 0.6% of the serial time cost of GEOS-2 DAS (i.e., as timed on a single processor) arises from code that is is not parallelized; of this, about half is in the initialization and data processing components of the PSAS and half is in the Interface. As a check, given that the Interface has 0.3% of the non-parallelized component of GEOS-2 DAS, this should correspond to $100 \times 0.003/(0.003 + 0.997/8) = 2.4\%$ of the time cost of GEOS-2 DAS on 8 processors, and this is in line with the figure in Table 1.

GEOS-2 DAS Component	Percentage of Wall Clock time 8 Processors of an SGI Origin 2000
GCM	45.
PSAS	39.
Diagnostics	13.5
Interface	2.5

Table 1: The percentage of time taken by the components of shared-memory multitasking parallel baseline GEOS-2 DAS. Runs were performed on 8 processors of an SGI Origin 2000.

Table 2 shows the percentage of time taken by the top-level components of the baseline GEOS-2 GCM. The GCM is run in "assimilation mode" using the Matsuno timestepping scheme. The times for the dynamics, the Shapiro filter spatial smoother, the polar rotation, and other grid transformations are bundled into a single component designated Dynamical Core (Takacs et al. 1994).

GCM Component	Percentage of Wall Clock time 10 Processors of Cray J90
Dynamical Core	43.
Moist Convection	16.
Turbulence	10.
Radiation	32.

Table 2: The percentage of time taken by the top-level components of GEOS-2 GCM (vc6.5, Takacs 1997). Although these numbers are for 10 processors of the Cray J90, they do not differ significantly from the baseline 8 processors on the SGI Origin 2000.

The percentage of time taken by the top-level components of the baseline GEOS-2 PSAS is shown in Table 3. The solver (Eqn. 5) with complexity $f\mathcal{N}_i sp^2$ takes about 35% of the time while the analysis (Eqn. 6) with complexity $fsnp$ takes 62% of the time. These expressions for complexity can be checked approximately by taking the nominal values, $f = 10$, $p = 10^5$, $n = 10^6$, $\mathcal{N}_i = 10$, and $s = 0.4$. Each of the expressions equals 4×10^{11}, i.e., the estimated count of floating-point operations for the PSAS is 8×10^{11} per analysis. This compares with 5×10^{11} floating point multiplications and 4.5×10^{11} floating point additions for the total complexity of GEOS-2 DAS (including the GCM, PSAS, Diagnostics, and Interface) per analysis obtained from the Cray J916 Hardware Performance Monitors. The estimate for the PSAS is high but a good order of magnitude; Table 1 indicates that $39/100 \times 9.5 \times 10^{11}$ $\approx 3.7 \times 10^{11}$ is more like the actual number of flops per analysis for the baseline GEOS-2 PSAS.

PSAS Component	Percentage of Wall Clock time 8 Processors of an SGI Origin 2000
Solver (Eqn. 5)	35.
Analysis (Eqn. 6)	62.
Utilities	3.

Table 3: The percentage of time taken by the top-level components of the baseline GEOS-2 PSAS.

The GEOS DAS is run in a number of production modes (Stobie 1996). These may be generally categorized as real-time, near real-time, and reanalysis modes. Real-time requires model forecast and analyses to take place sufficiently in excess of one day of assimilation per wall-clock day so that the results may be studied and disseminated to customers such as satellite instrument teams with real-time needs. Reanalyses are multi-year studies designed to provide long-term datasets from a frozen scientific software configuration. For example, the DAO has completed a reanalysis for the years 1979 to 1995 using the GEOS-1 version of GEOS DAS (Schubert et al. 1993, 1995). Appendix A summarizes the baseline GEOS-2 DAS system performance and throughput. GEOS-2 DAS uses shared-memory multitasking parallelism and runs on Cray J90/C90 and SGI Origin 2000 computers. For GEOS DAS, the DAO now uses distributed-memory parallelism with the Message-Passing Interface (MPI) and shmem libraries (Lyster 2000a).

The data acquisition and storage system for 4DDA involves a worldwide instrumentation, telecommunication, databasing, computational and administrative effort (Atlas 1997). We remark here only on the attributes and numbers that are relevant to the present work. In the last 60 years about 2 billion observations that are appropriate for input to atmospheric data assimilation systems have been accumulated. The volume of these data does not present the greatest computational complexity, and operational centers are more concerned with the accuracy of these data. Considerable energy is devoted to finding and validating old observations, i.e., "data rehabilitation". In the coming years, diverse new data types will be made available for data assimilation, and the volume and complexity of the data handling system will increase considerably. For example, satellite sea-surface wind observations have been shown to be useful in increasing forecast accuracy of weather analyses (Atlas et al. 1996). The DAO will also assimilate increasing amount of non-meteorological data, such as trace gase concentration in the atmosphere. During the late 1990s, when GEOS-2 DAS was the main operational data assimilation algorithm at the DAO, about 10^5 observations were produced daily under the World Weather Watch and transmitted to worldwide weather centers and the DAO via the Global Telecommunications System, which is under the supervision of the World Meteorological Organization (Atlas 1997). More than 70% of these were obtained from satellites measurements, mostly as temperature retrievals; the remaining were from in situ balloon-borne and land and sea surface instruments. At baseline resolution for the GEOS-2 GCM ($2^o \times 2.5^o \times 70$ levels), a day of assimilation produced in excess of 1 gigabyte of data. Hence data assimilation at real time (one day of assimilation per wall-clock day) did not stretch the local disk capacity or bandwidth of most modern computer systems. However, extended runs at higher throughput than real time increases the burden on storage and data processing. The most severe challenge is for reanalysis projects where multi-year datasets are analyzed by a fixed Data Assimilation System and the products are made available to the scientific community. The standard benchmark is a rate of 30 days of assimilation per day of wall-clock time (i.e., a fifteen year reanalysis on order half a year). At this rate the GEOS-2 DAS produced about 10 terabytes of data per year.

3 The Kalman Filter

The Kalman filter (Jazwinski 1970, Cohn 1977) assimilates observations sequentially with the model at the corresponding time (t_k) when they are taken. In this regard, it is like the PSAS with a shortened analysis update cycle:

$$w_k^a = w_k^f + K_k(w_k^o - H_k w_k^f) , \tag{7}$$

where the Kalman gain is

$$K_k = P_k^f H_k^T (H_k P_k^f H_k^T + R_k)^{-1} , \tag{8}$$

where s_k observations are assimilated at time t_k. For the Kalman filter analysis the cycle also involves both a model forecast

$$w_{k+1}^f = \mathcal{M}_k w_k^a, \tag{9}$$

and a dynamically consistent forecast of the state error covariance matrix

$$P_{k+1}^f = M_k P_k^a M_k^T + Q_k, \tag{10}$$

where M_k is the tangent-linear model operator, and Q_k is the model (or system) error covariance matrix. The analysis error covariance matrix at the new time t_{k+1} is

$$P_{k+1}^a = (I - K_{k+1} H_{k+1}) P_{k+1}^f, \tag{11}$$

where I is the identity matrix. The filter then proceeds sequentially in time through repeated iterations of Eqns. (7)-(11).

A two-dimensional (latitude-longitude) Kalman filter for the assimilation of stratospheric chemical constituents was developed by Lyster et al. (1997), and is being used for scientific study of stratospheric constituent gases (Ménard et al. 2000a,b). The dynamical model uses advective transport with a grid-point based flux-conserving algorithm (Lin and Rood 1996). The transport is driven by prescribed winds from GEOS DAS. For example, at a $2^o \times 2.5^o$ resolution the number of grid points is $n = 91 \times 144 = 13104$ and the model timestep is 15 minutes. This was used for the assimilation of retrieved methane from the Cryogenic Limb Array Etalon Spectrometer (CLAES) instrument aboard NASA's Upper Atmosphere Research Satellite (UARS). For CLAES, there were typically $p_k \approx 15$ observations, per layer, per timestep. The Kalman filter achieved 150 days of assimilation per wall-clock day, or 4.1 sustained gigaflop/s, on 128 processors of the Cray T3E-600 at NASA Goddard Space Flight Center. Figure 2 shows a still from a video (NASA ESS 1997) using gridded output from the Kalman filter and produced for a study of a tropical atmospheric wave-braking event in the stratosphere. The visualization employed Vis5D to render isosurfaces of constant mixing ratio of methane to depict the three-dimensional structure and evolution of the stratosphere. The assimilated observations from CLAES covered the interval September 6-14, 1992. The experiments had 18 vertical levels in the atmosphere, and the horizontal resolution was $5^o \times 4^o$. The vertical grid was generated by the assembly of 18 layers, each representing a two-dimensional assimilation experiment. The time taken for the 8 day runs was 20 hours of wall-clock time on 128 processors of the GSFC Cray T3D.

For the grid-point based horizontal transport that is used for the two-dimensional Kalman filter, the complexity of a single timestep of the model, Eqn. (9), is hn, where $h \approx 10 - 100$ takes into account the size of the finite-difference template. The complexity of Eqn. (10) is $(2h+1)n^2$ per analysis cycle. The Kalman gain, Eqn. (8), may be evaluated using a direct solver using $\mathcal{O}(p_k^3)$ operations. Alternatively, Eqns. (5) and (6) may be employed; their computational complexity was discussed in Section 2.2; however, this method does not generate the Kalman gain K_k explicitly. The complexity of Eqn. (11) is approximately $(p_k+1)n^2$. For the GEOS-2 DAS, observations are aggregated over a six-hourly interval. As described above, the value of p_k for the Kalman filter is smaller than for the GEOS-2 DAS by the number of model timesteps in 6 hours. At baseline resolution for the GCM ($2^o \times 2.5^o \times 70$ layers) the timestep of the dynamics is 3 minutes, so p_k is 120 times smaller than for the PSAS. Only small experiments (e.g., $p_k < 10^3$) could afford to evaluate K_k directly. A Kalman filter or an approximate Kalman filter for a large-scale multivariate meteorological system would have to use an iterative solver, such as the PSAS. The matrices $P_k^{f,a}$ are of size n^2, and $H_k P_k^f H_k^T + R_k$ is of size p_k^2.

A Kalman filter based on a tangent-linear three-dimensional GCM would require considerably more resources than the filter described above for stratospheric analyses. The memory to store the error covariance matrices, $P^{f,a}$, would be approximately $n^2 \approx 6.8 \times 10^{12}$ words at the baseline resolution of $2^o \times 2.5^o \times 70$ levels, and the algorithm, based on the complexity of just Eqn. (11), would scale to approximately $n \times 250$ megaflop/s = 0.25 petaflop/s (the value 250 megaflop/s is taken from the baseline GEOS-2 DAS in Appendix A). This is clearly beyond the reach of current resources. GEOS DAS, with an analysis based on PSAS, is an approximate Kalman filter. Efforts are under way worldwide and at the DAO to develop computationally feasible improvements to 4DDA algorithms, such as reducing the analysis cycle time for GEOS DAS, and developing more physically-based error covariance models (Riishøjgaard 1998).

4 The Scalable Distributed-Memory Parallel GEOS DAS

The GCM and PSAS have tightly-coupled core algorithms with computational and communication intensive parallel implementations; these are hydrodynamic transport (GCM) and non-sparse large matrix-vector multiplications (PSAS). The baseline GEOS-2 DAS uses shared-memory multitasking parallelism on Cray J series and SGI Origin computers. A distributed-memory parallel implementation of the GCM was designed (Lyster et al. 1997) and prototyped (Sawyer and Wang 1999) using the Message-Passing Interface (MPI) and shmem libraries. Distributed-memory parallel PSAS was prototyped (Ding and Ferraro 1995), and an MPI PSAS kernel was developed (Guo et al. 1998, Larson et al. 1998). During the year 2001 development and validation of distributed-memory parallel GEOS DAS was completed. The following sections discuss technical issues and limitations related to scalable distributed-memory parallel implementations of the GCM and PSAS. We will focus on the tightly-coupled core hydrodynamic transport and non-sparse large matrix-vector multiply algorithms. In addition, we will discuss the development of scalable end-to-end large-scale applications such as GEOS DAS.

4.1 Asymptotic Scalability of Distributed-Memory Parallel Gridpoint General Circulation Models

We calculate the limit on the number of processors that can be usefully employed to reduce the wall-clock time of a distributed-memory parallel grid-point based transport algorithm. In the parallel decomposition, compact domains of grid points and their associated floating-point operations are distributed across processors. The limit on the number of processors is the result of the surface-to-volume effect (e.g., Foster 1994 Sec. 2.4), whereby the impact of communication of domain surface data becomes comparable to the time to perform the floating-point operations of the algorithm. This is an approximation of the scalability in the sense that it does not account for a number of the typical complications that often occur in General Circulation Models (GCM), viz:

- We are neglecting the algorithms for parameterized physics processes, which include moist convection, turbulence, and radiation; the grid transformations; the diagnostics; and the I/O.
- We are not assessing the impact of load imbalance.
- We cannot simply account for indeterminacy in communications, such as in semi-Lagrangian methods.

The embarrassingly parallel parts of the GCM (e.g, some algorithms for parameterized physics processes) tend to improve the overall scaling with respect to the present calculation, while load imbalance will tend to make the scaling worse. Other components (e.g, the parallel rotation grid transformation) need a separate analysis (Lyster 2000a and articles therein). The communication of domain surface data enables algorithmic consistency across the boundary between processor domains. The present calculation is very similar to the estimate of parallel scalability of particle-in-cell methods by Lyster et al. (1995), except that case involved communication of mobile particles, which represented plasma ions and electrons, across gridpoint domain boundaries. We assume that the communication time can be approximated in terms of the number of bytes communicated per processor and the bandwidth of the communication channel (i.e., latency effects make the scalability worse, so this approximation is still good in terms of evaluating an upper bound on scalability). With this, the following calculation provides a good approximation for the scalability of the distributed-memory parallel dynamical core.

Define the following symbols:

N_p = Total number of processors employed
N_g = Total number of grid points in the computational domain
d = Dimension of the physical problem
D = Dimension of the parallel decomposition
M = single processor speed in megaflop/sec
B = interprocessor communication bandwidth in megabytes/sec
F = Number of flops/gridpoint/timestep for the relevant transport algorithm
G = The number of "layers" of guard cells in each dimension of the parallel decomposition (e.g., $G = 2$ for fourth order finite difference)
P = The precision of the calculation in bytes per word (i.e., $P = 4$ or 8)

Typically $D = 1, 2$, or 3, and $d = 2$ or 3, while $d \geq D$. For instance, the physical world is three dimensional (i.e, $d = 3$), however, we sometimes discuss scalability in terms of the number of gridpoints in the latitude-longitude domain (i.e, horizontal transport), for which $d = 2$. In this case the most efficient parallel decomposition uses compact two-dimensional blocks of gridpoints, sometimes called the "checkerboard" decomposition ($D = 2$). The number of gridpoints around the border of each domain is then $2DN_g^{(d-1)/d}/N_p^{(D-1)/D} \equiv 2DN_g^{\frac{1}{2}}/N_p^{\frac{1}{2}}$.

The communication time per timestep per processor is:

$$T_{comm} = 2DGPB^{-1}N_g^{(d-1)/d}/N_p^{(D-1)/D}. \tag{12}$$

The CPU time per timestep per processor is:

$$T_{cpu} = (F/M)(N_g/N_p). \tag{13}$$

Hence the ratio of communication to CPU time is:

$$\tau := T_{comm}/T_{cpu} = \frac{2DGP}{F}\frac{M}{B}\frac{N_p^{1/D}}{N_g^{1/d}}. \tag{14}$$

The parallel speedup (SU) is defined as the time for the application to run on 1 processor divided by the time to run on N_p processors. With the present assumptions, we have:

$$SU = N_p/(1+\tau). \tag{15}$$

Therefore we may nominally define the maximum speedup, N_{pmax}, as the number of processors for which τ in Eqn. (14) is equal to 1:

$$N_{pmax} = \left[\frac{BFN_g^{1/d}}{2MDGP}\right]^D. \tag{16}$$

Beyond that, the floating point operations in additional processors are effectively wasted.

The terms in τ may be characterized as follows:

- $\frac{2DGP}{F}$: Parameters of the algorithm.
- $\frac{M}{B}$: Parameters of the computer.
- $N_g^{1/d}$: The problem resolution.
- $N_p^{1/D}$: The surface-to-volume effect (i.e., τ gets larger in proportion to the number of processors to some geometry-dependent exponent).

For parameters typical of current global transport algorithms, $N_g = 360 \times 181$ (i.e., 1°× 1°resolution), $d = D = 2$, $M = 100$, $B = 10$, $F = 50$, $G = 2$, and $P = 8$, so Eqn. (16) gives $N_{pmax} = 400$.

4.2 Asymptotic Scalability of Distributed-Memory Parallel Matrix-vector Multiply for the PSAS Solver

We calculate the limit on the number of processors that can be usefully employed to reduce the wall-clock time of a distributed-memory dense matrix-vector multiply. The dominant time cost of the PSAS, Eqs. (5) and (6), are large, dimension $p \approx 10^5$, matrix-vector multiplications. For the present analysis the results do not differ significantly between the symmetric (Eqn. 5) or rectangular (Eqn. 6) cases since the structure of each dimension of the matrix is determined by a compact spatial decomposition of the multi-dimensional data (see Guo et al. 1998). We will therefore only show the scaling analysis for the symmetric case. Parallelism is achieved by assigning subsets of the block matrix-vector multiplications to each processor. The partial vector results are then summed using the `MPI_reduce_scatter()` library call as shown schematically in Figure 3. The cycle of the parallel matrix-vector multiply is then completed using the `MPI_all_gather()` library call (not shown in the figure).

Advanced libraries such as PLAPACK (van De Geijn, 1997) have custom interfaces and decompositions to support dense matrix-vector operations. We chose not to use this because the more general interface of the MPI library is both simple and compatible with the pointer-specified multi-dimensional vectors (Larson et al. 1998). Using a 6,000 kilometer cutoff length for correlation functions, the matrices are semi-dense with density $s \approx 0.4$. For the moment, we focus on the limitations on scalability due to the trade-off between communications in the `MPI_reduce_scatter` and `MPI_all_gather()`, and the time cost of the sub-block matrix-vector multiplications. We ignore the costs of the floating point operations in the reduction. As in Section 4.1, we ignore the cost of latency in the interprocessor communications.

Assuming that the collective `MPI` communication calls described above are implemented using an efficient method such as recursive halving (Foster 1994, Sec. 11.2) the cost of communications is

$$T_{comm} = 2(pP/B)(N_p - 1)/N_p \approx 2pP/B, \tag{17}$$

where we have used the same definitions as Section 4.1, and $p \approx 10^5$ is the size of the vector. The CPU time per processor is:

$$T_{cpu} = fsp^2/(N_pM), \tag{18}$$

where, as in Section 2.2, f equals two plus the number of floating point operations to form each matrix element. The parallel speedup is given by Eqn. (15), and the maximum speedup is defined in the same way as Section 4.1:

$$N_{pmax} = \frac{fspB}{2PM}. \tag{19}$$

For typical values for these parameters as defined in Section 4.1 and above, $N_{pmax} = 625fs$. If the matrix is precalculated, $f = 2$, but it may of order 10 when elements are calculated on the fly. Memory limitations prohibit storing entire matrices, so current implementations enable a combination of pre-stored and on-the-fly calculation of matrix elements. The matrix density $s \approx 0.4$, so its clear that the upper limit of scalability of semi-dense matrix-vector multiplications, and hence the PSAS, is of the order of thousands of processors for current generation machines and current input datasets. The value is larger than the upper limit for a GCM because transport algorithms in the dynamical cores of GCMs are sparse matrix algorithms, which have more stringent scalability limits due to the surface-to-volume effect described in Section 4.1.

The calculation thus far presents an upper limit on scalability. We will discuss in the next section that the non-parallelized code presents a significant limit on the scalability of the end-to-end algorithm through Amdahl's law (1967). We discuss here a number of factors that reduce the scalability of the PSAS below the theoretical limit. First, on large numbers of processors the size of the vector segments are sufficiently small that message latency and synchronization dominate the communication cost of the collective MPI calls. Second, the PSAS has a nested preconditioner which involves successively sparser matrix-vector multiplications (Cohn et al. 1998, Larson et al. 1998). Through Eqn. (19) (i.e., $N_{pmax} \sim s$) these will negatively effect scalability. Third, work load imbalance has a serious impact on parallel scalability. The baseline MPI PSAS Kernel has an upper limit of 57,600 matrix blocks, which should be sufficient to provide a statistically uniform distribution when their work is allocated across one or two thousand processors (Lyster et al. 2000b). However, these blocks are of widely differing size because their dimensions depend on the non-repeatable distribution of observations in geographical areas of the earth. Early versions of the Kernel used a method for load balancing that based the costs of the block matrix-vector multiplications on the dimensions of the blocks. This was later augmented, with only incremental improvement in scalability, by dynamic scheduling and work scheduling based on statistically tuned cost estimates. The lower curve of Figure 4 (from Lyster et al. 2000b) shows the scaling of the baseline MPI PSAS Kernel including the load balancing algorithm for 52,738 observations covering a standard 6 hour analysis cycle. The poorer scaling relative to the above calculation is from a combination of load imbalance and sparse preconditioners; using $s = 0.1$ and $f = 5$ in Eqn. (19) gives $N_{pmax} = 312$ which is in line with Figure 4. The lower curve in Figure 4 corresponds to the case of approximately $57,600s$ blocks. The improved scaling shown in the upper curve of the figure corresponds to the improved load imbalance that resulted from a refinement to $921,600s$ blocks. While the scalability is improved it is clear that the MPI PSAS kernel did not reach the theoretical limit that had been expected from the above calculation. Apart from our work on load balancing algorithms, we have developed and continue to work on collective parallel algorithms using optimized communication procedures.

4.3 Scalability of Distributed-Memory Parallel GEOS DAS

The wall-clock time of an application, which is clearly the bottom line criterion for *performance*, depends on both the scalability of the parallel algorithms and the single processor CPU speed of the algorithms. Indeed, for obvious reasons algorithms must be designed to achieve the maximum performance with the minimum number of processors. We have shown that the highly coupled parallel subcomponents of distributed-memory parallel grid-point GCM and PSAS have upper limits to their scalability in the range $400 - 1000$ processors on SGI Origin 2000 series and similar computers. We have also shown, in Tables 1, 2 and 3, that the main subcomponents of the GEOS DAS (Dynamical Core, Moist Convection, Turbulence, Radiation, PSAS Solver, PSAS Analysis, Diagnostics, and Interfaces) have an approximately flat timing profile. This means that a large fraction of $150,000$ lines of code are candidates for single processor optimization – a significant software effort.

In addition to these issues of single processor optimization and parallel scalability of core algorithms, we have to account for unparallelizable and unparallelized code. This is usually stated in Amdahl's law (1967) which estimates the upper limit of scalability for a fixed problem size in terms of number of processors that can be usefully employed to reduce the wall clock time of an application. This limit is approximately the inverse of the fraction of time taken by the unparallelized code as measured by running the application on a single processor. As a baseline, for GEOS-2 DAS the fraction of unparallelized code is 0.006 (Section 2.3), which is an impressively small number. However the Amdahl's limit on the entire parallel GEOS-2 DAS application is therefore $1./0.006 = 166$ processors. The shared-memory parallel GEOS-2 was not intended to exceed scalability beyond 64 processors. However, even an efficient distributed-memory parallelization of the GCM and PSAS would result in GEOS-2 DAS which does not scale beyond 166 processors. Increasing the resolution of the transport algorithm, and using more observations will improve scalability because there is correspondingly more work to distribute among processors. However it is a fact that, unlike a large portion of modern scientific computing, data assimilation and earth science modeling do not support rapid change in resolution and problem size because these changes require extensive and time consuming testing and scientific validation. From the above discussion, this rather conservative limit can be extended by focused efforts to parallelize more of GEOS DAS, especially the Interface in Table 1; using optimized parallel libraries and enabling overlapping communications and CPU if possible; and single processor optimization to reduce the CPU cost of the unparallelized code. Some of the implications of these kinds of efforts will be discussed in the Summary.

5 Summary

We have discussed the computational complexity of the GEOS-2 DAS, which is a baseline data assimilation system at NASA's Data Assimilation Office in the late 1990s. The complexity of the General Circulation Model (GCM) generally scales linearly with the number of spatial grid points, n, per iteration of the algorithm (with the exception of the quadratic scaling of the radiation algorithms with respect to the number of vertical levels). The need to reduce the timestep of the dynamics as the spatial resolution is increased results in an asymptotic $\mathcal{O}(n^{4/3})$ scaling for the dynamical core for the simulation of fixed time intervals. The Physical-space Statistical Analysis System (PSAS) has asymptotic scaling $\mathcal{O}(np)$ and $\mathcal{O}(p^2)$. The latter arising from the solver, Eqn. (5), and the former from Eqn. (6) whose fundamental basis is the error correlation between all observations and all grid points in an analysis cycle. The computational complexity of the PSAS is reduced by increasing matrix sparsity, $s \approx 0.4$. Other modifications such as multipole methods and reduction in the analysis cycle time are under research. The computational complexity and the required computer memory of the Kalman filter is quadratic in n. We showed, using a simple estimate based on the performance of current GCMs, that a Kalman filter for atmospheric global data assimilation would scale to petaflop/s proportions. We have developed a reduced spatial dimension version of the Kalman filter suitable for research on stratospheric constituent gas assimilation where the dynamics are two dimensional. We've noted that the development of a full, petaflop/s scale, Kalman filter would be an ambitious and scientific significant exercise, but the main thrust for practical or operational implementations concentrate on approximate Kalman filters with reduced computational complexity.

We developed parameterized formulae that estimate the limit to distributed-memory parallel scalability of the tightly coupled transport and large matrix-vector multiplications which are important components of the CPU time cost of the grid-point GCM and PSAS. For SGI Origin 2000 and similar computers the scalability is limited to $400 - 1000$ processors. We also point out that reducing the wall clock time of the GEOS DAS involves large-scale efforts on single processor optimization of approximately $150,000$ lines of code. In addition, the unparallelizable and unparallelized code poses significant limits on the scalability of the end-to-end algorithms. For example, for GEOS-2 DAS with distributed-memory parallel GCM and PSAS, the CPU cost of the unparallelized code measured as a fraction of the whole algorithm (i.e., as run on a single processor) is only 0.006. This includes I/O and represents by far the bulk of the lines of code of GEOS-2 DAS. However, the Amdahl's limit of this end-to-end algorithm would be $1/0.006 = 166$ processors. Therefore, efforts to improve the scalability of GEOS DAS necessarily involve specialized, optimal, parallelization of a very large number of the $150,000$ lines of code (aside from the core transport and matrix-vector multiply subcomponents), or improving their serial performance. However, there are not just physical limits to the ability to parallelize interfaces, initialization procedures, and I/O, but there are practical issues related to software maintenance in a multi-developer and multi-user scientific environment. The serial code and interfaces undergo rapid simultaneous modification by multiple developers with requirements to improve scientific algorithms and solidify production suites. In this environment, drastic efforts to improve the scalability of code must be coordinated with careful oversight that balances scientific requirements and computational scientific software engineering.

6 Acknowledgments

The authors acknowledges discussions with Steven Cohn, Anthony Colyandro, Chris Ding, Gregory Gaspari, Dan Kokron, William Sawyer, Harper Pryor, Richard Rood, Meta Sienkiewicz, Lawrence Takacs, Ricardo Todling, and Philip Webster. This work was funded under the NASA High Performance Computing and Communications Initiative (HPCC) Earth and Space Science (ESS) Program Cooperative Agreement Number NCCS5-150, the NASA EOS Interdisciplinary Science Program, and the NASA Research and Applications Program.

Appendix A: GEOS-2 DAS System Performance and Throughput

Baseline GEOS-2: $2^o \times 2.5^o \times 70$ level GCM resolution; 400,000 obs/day	
5 days/wallclock day throughput	
This corresponds to the baseline operational code	
Multitasking GEOS-2 DAS run on 8pe Origin 2000	
Main Memory (GB)	2.2 (per image)
Disk (GB)	8.0
Mass Storage (GB)	2300.0 (this is output per year)
Volume of Data (GB)	6.2 (produced per day per image)
Gigaflop/s sustained	0.25 (per image)
Duration of Run	5 days/wallclock day (continuous operation, single image)

6 References

Andersson, E., J. Haseler, P. Undén, P. Courtier, G. Kelly, D. Vasiljević, C. Branković, C. Cardinali, C. Gaffard, A. Hollingsworth, C. Jakob, P. Janssen, E. Klinker, A. Lanzinger, M. Miller, F. Rabier, A. Simmons, B. Strauss, J-N. Thepaut, and P. Viterbo, 1998: The ECMWF implementation of three dimensional variational assimilation (3D-Var). Part III: Experimental Results, *Q. J. R. Meteorol. Soc.*, **124**, 1831-1860.

Atlas, R., R.N. Hoffman, S.C. Bloom, J.C. Jusem, J. Ardizzone, 1996: A Multiyear Global Surface Wind Velocity Dataset Using SSM/I Wind Observations, BAMS, **77(5)**, 869-882.

Atlas, R., 1997: Atmospheric Observations and Experiments to Assess Their Usefulness in Data Assimilation. *J. Meteo. Soc. Japan*, **75** (1B), 111-130.

Amdahl, G.M., 1967: Validity of the single-processor approach to achieving large scale computing capabilities. In AFIPS Conference Proceedings, **30**, (Atlantic City, N.J., Apr. 18-20). AFIPS Press, Reston, Va., 483-485.

Bloom, S.C., L.L. Takacs, A.M. da Silva, and D. Ledvina, 1996: Data Assimilation Using Incremental Analysis Updates, *Mon. Wea. Rev.*, **124**, 1256-1271.

Brackbill, J., and B. Cohen, Eds., 1985: *Multiple Time Scales, Computational Techniques*, Academic Press, Orlando, FL.

Cohn, S.E., 1997: An Introduction to Estimation Theory, *J. Meteo. Soc. Japan*, **75** (1B), 257-288.

Cohn, S.E., A. da Silva, J. Guo, M. Sienkiewicz, and D. Lamich, 1998: Assessing the effects of data selection with the DAO Physical-space Statistical Analysis System. *Mon. Wea. Rev.,* **126**, 2913-2926.

Courtier, P., E. Andersson, W. Heckley, J. Pailleux, D. Vasiljević, M. Hamrud, A. Hollingsworth, F. Rabier, and M. Ficher, 1998: The ECMWF Implementation of three dimensional variational assimilation (3D-Var). Part I: Formulation, *Q. J. R. Meteorol. Soc.*, **124**, 1783-1808.

Daley, R., 1991: *Atmospheric Data Analysis.* Cambridge University Press. New York, 457pp. ISBN 0-521-38215-7.

DAO 2000: Algorithm Theoretical Basis Document Version 2.01, Data Assimilation Office, NASA's Goddard Space Flight Center. Available at: `http://dao.gsfc.nasa.gov/subpages/atbd.html`

Ding, H., and R. Ferraro, 1995: A General Purpose Parallel Sparse Matrix Solver Package, *Proceedings of the 9th International Parallel Processing Symposium*, p. 70, April 1995.

Foster, I., 1994: *Designing and Building Parallel Programs: Concepts and Tools for Parallel Software Engineering*, Addison Wesley.

Gaspari, G. and S.E. Cohn, 1999: Construction of correlation functions in two and three dimensions, *Quart. J. Roy. Met. Soc.,* **125**, 723-757.

van De Geijn, R. A., P. Alpatov, G. Baker, C. Edwards, 1997: *Using Plapack : Parallel Linear Algebra Package (Scientific and Engineering Computation)*, MIT Press, 224pp.

Gibson, J.K., P. Kållberg, S. Uppala, A. Nomura, A. Hernandez, E. Serrano, 1997: ERA Description, ECMWF Re-Analysis Project Report Series, 1.

Golub, G.H. and C.F. van Loan, 1989: *Matrix Computations*, 2nd Edition, The John Hopkins University Press, 642pp.

Guo, J., J.W. Larson, P.M. Lyster, and G. Gaspari, 1998: Documentation of the Physical-space Statistical Analysis System (PSAS) Part II: The Factored-Operator Error Covariance Model Formulation, NASA Data Assimilation Office, Office Note No. 98-04. Available at `http://dao.gsfc.nasa.gov/subpages/office-notes.html`

HPCC ESS, NASA High Performance Computing and Communications Earth and Space Sciences, 1977: *Images of Earth and Space: SC97 Edition*, NASA Scientific Visualization Studio, Goddard Space Flight Center. This was shown at the High Performance Computing and Communications booth at Supercomputing97, San Jose, CA, November, 1997.

Jazwinski, A.H., 1970: *Stochastic Processes and Filtering Theory.* Academic Press, 276pp.

Kalnay, E., et al., 1996: The NMC/NCAR 40-Year reanalysis project, *Bull. Amer. Meteor. Soc.*

Kanamitsu, M., W. Ebisuzaki, J. Woollen, J. Potter, and M. Fiorino, 1999: An overview of the NCEP/DOE reanalysis 2, in *Proceedings of the 2nd International Conference on Reanalysis*, Wokefield Park, Reading, UK.

Kistler, R.E., and E. Kalnay et al., 2000: The NCEP/NCAR 50-year reanalysis, to appear in *Bull. Am. Meteor. Soc.*

Larson, J.W., J. Guo, P.M. Lyster, 1998: Documentation of the Physical-space Statistical Analysis System (PSAS) Part III: Software Implementation, NASA Data Assimilation Office, Office Note No. 98-05. Available at `http://dao.gsfc.nasa.gov/subpages/office-notes.html`

Lin, S.-J., and R.B. Rood, 1996: Multidimensional Flux-form semi-Lagrangian transport schemes, *Mon. Wea. Rev.*, **124**, 2046-2070.

Lyster, P.M., P.C. Liewer, R.D. Ferraro, and V.K. Decyk, 1995: Implementation and Characterization of Three-Dimensional Particle-In-Cell Codes on Multiple-Instruction-Multiple-Data Parallel Supercomputers. *Comp. Phys.*, **9(4)**, 420-432.

Lyster, P.M., S.E. Cohn, R. Ménard, L.-P. Chang, S.-J. Lin, and R. Olsen, 1997: Parallel Implementation of a Kalman Filter for Constituent Data Assimilation, *Mon. Wea. Rev.*, **125**, 1674-1686.

Lyster, P.M., W. Sawyer, and L. L Takacs, 1997: Design of the Goddard Earth Observing System (GEOS) Parallel General Circulation Model (GCM), NASA Data Assimilation Office, Office Note No. 97-13. Available at `http://dao.gsfc.nasa.gov/subpages/office-notes.html`

Lyster, P.M., 2000a: Final Report on the NASA HPCC PI Project: Four Dimensional Data Assimilation. Available at
`http://dao.gsfc.nasa.gov/DAO_people/lys/hpccfinal`

Lyster, P.M., T. Clune, J. Guo, J.W. Larson, 2000b: Performance Optimization of the Physical-space Statistical Analysis System (PSAS) Available at
`http://dao.gsfc.nasa.gov/DAO_people/lys/psasopt.ps`

Ménard, R., S. E. Cohn, L.-P. Chang, and P. M. Lyster, 2000a: Assimilation of Stratospheric Chemical Tracer Observations Using a Kalman Filter. Part I: Formulation, *Mon. Wea. Rev.*, **128**, 2654-2671.

Ménard, R., and L.-P. Chang, 2000b: Assimilation of Stratospheric Chemical Tracer Observations Using a Kalman Filter. Part II: χ^2-Validated Results and Analysis of Variance and Correlation Dynamics, *Mon. Wea. Rev.*, **128**, 2672-2686.

Parrish, D.F. and J.C. Derber, 1992: The National Meteorological Center's spectral statistical-interpolation analysis system, *Mon. Wea. Rev.*, **120**, 1747-1763.

Parrish, D.F., J.C. Derber, R.J. Purser, W.-S. Wu, and Z.-X. Pu, 1997: The NCEP global analysis system: recent improvements and future plans, *J. Met. Soc. Japan*, **75**, No. 1B, 359-365.

PSAS 1998: the technical documents for the PSAS are da Silva and Guo (1996), Guo et al. (1998), and Larson et al. (1998).

Rabier, F., A. Mc Nally, E. Andersson, P. Courtier, P. Undén, J. Eyre, A. Hollingsworth, F. Bouttier, 1998: The ECMWF implementation of three dimensional variational assimilation (3D-Var). Part II: Structure Functions, *Q. J. R. Meteorol. Soc.*, **124**, 1809-1830.

Riishøjgaard, L.P. 1998: A Direct Way of Specifying Flow-Dependent Background Error Correlations for Meteorological Analysis Systems Tellus, **50A**, 42-57.

Sawyer, W., A. Wang, 1999: Benchmark and Unit Test Results of the Message-Passing GEOS General Circulation Model, NASA Data Assimilation Office, Office Note No. 99-04. Available at
`http://dao.gsfc.nasa.gov/subpages/office-notes.html`

Schubert, S.D., J. Pfaendtner, and R. Rood, 1993: An Assimilated Data Set for Earth Science Applications, *Bull. Am. Met. Soc.*, **74**, 2331-2342.

Schubert, S.D., C.-K. Park, C-Y. Wu, W. Higgins, Y. Kondratyeva, A. Molod, L.L. Takacs, M. Seablom, and R.B. Rood, 1995: A Multiyear Assimilation with GEOS-1 System: Overview and Results. NASA Tech. Memo. 104606, Vol. 7, 201 pp., Available at `http://dao.gsfc.nasa.gov/subpages/tech-reports.html`

da Silva, A., and J. Guo, 1996: Documentation of the Physical-Space Statistical Analysis System (PSAS) Part I: The Conjugate Gradient Solver Version, PSAS-1.00, NASA Data Assimilation Office Note 96-02. Available at
`http://dao.gsfc.nasa.gov/subpages/office-notes.html`

Stobie, J.G., 1996: Data Assimilation Computing and Mass Storage Requirements for 1998, NASA Data Assimilation Office, Office Note No. 96-16. Available at
`http://dao.gsfc.nasa.gov/subpages/office-notes.html`

Takacs, L.L., A. Molod, and T. Wang, 1994: Documentation of the Goddard Earth Observing System (GEOS) General Circulation Model – Version 1. *NASA Technical Memorandum 104606*, Volume **1**, NASA Goddard Space flight Center, Greenbelt, MD 20771. Available at
`http://dao.gsfc.nasa.gov/subpages/tech-reports.html`

Takacs, L.L., 1997: Impact of Resolution on GEOS GCM Model Development, DAO internal report.

Takacs, L.L., 2001: Personal Communication.

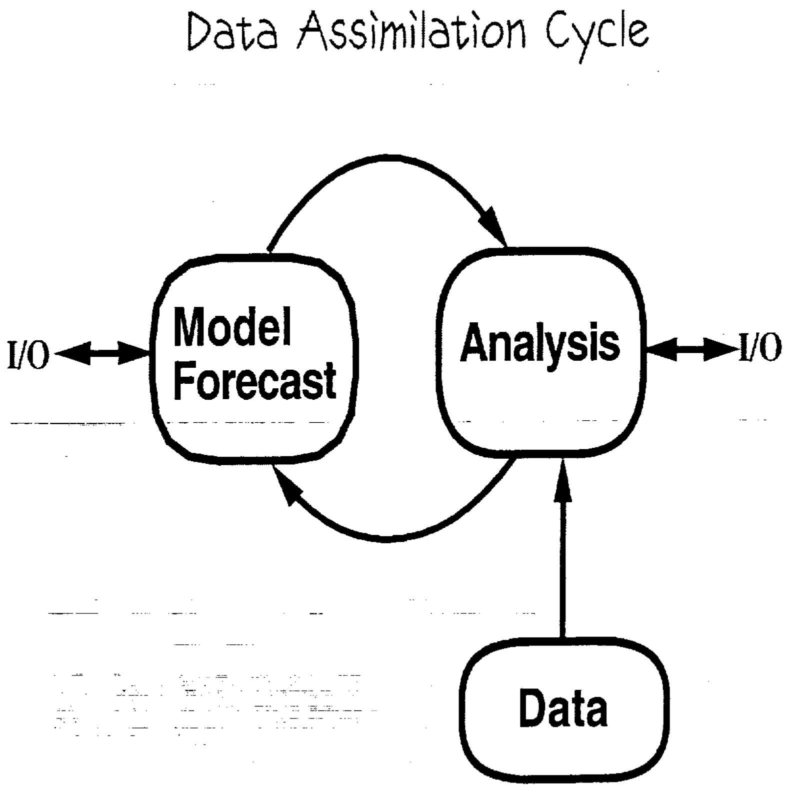

Figure 1: Schematic of cycled four dimensional data assimilation.

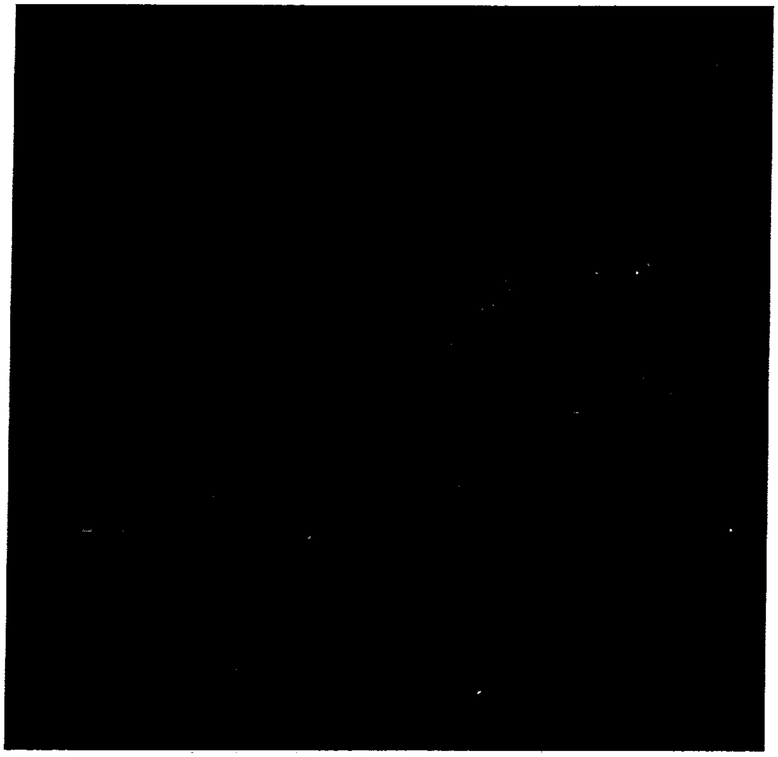

Figure 2: Still from the video (HPCC ESS 1997) showing an isosurface of constant methane mixing ratio in the stratosphere produced by the Kalman filter.

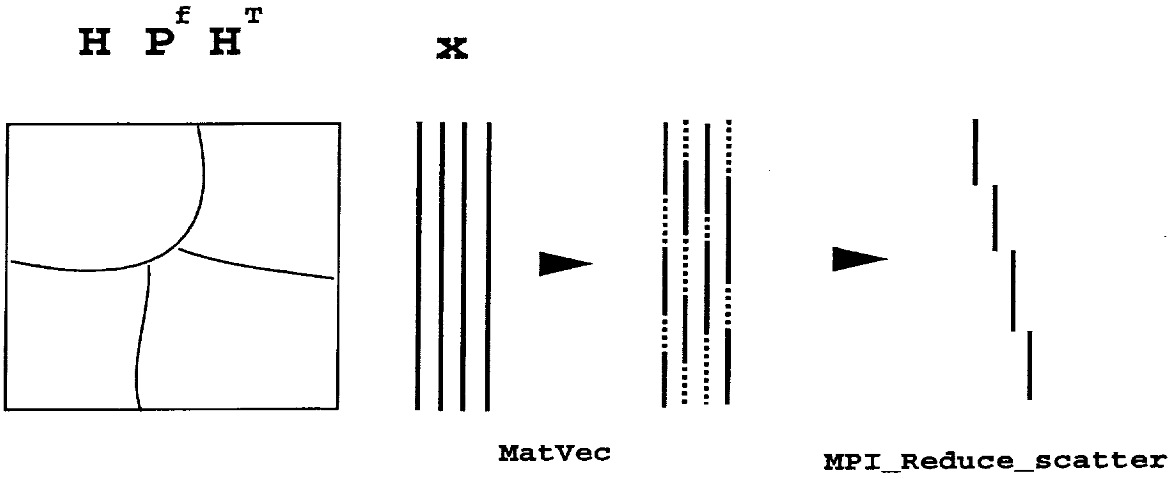

Figure 3: Schematic of the parallel decomposition for dense matrix-vector multiply for 4 processors.

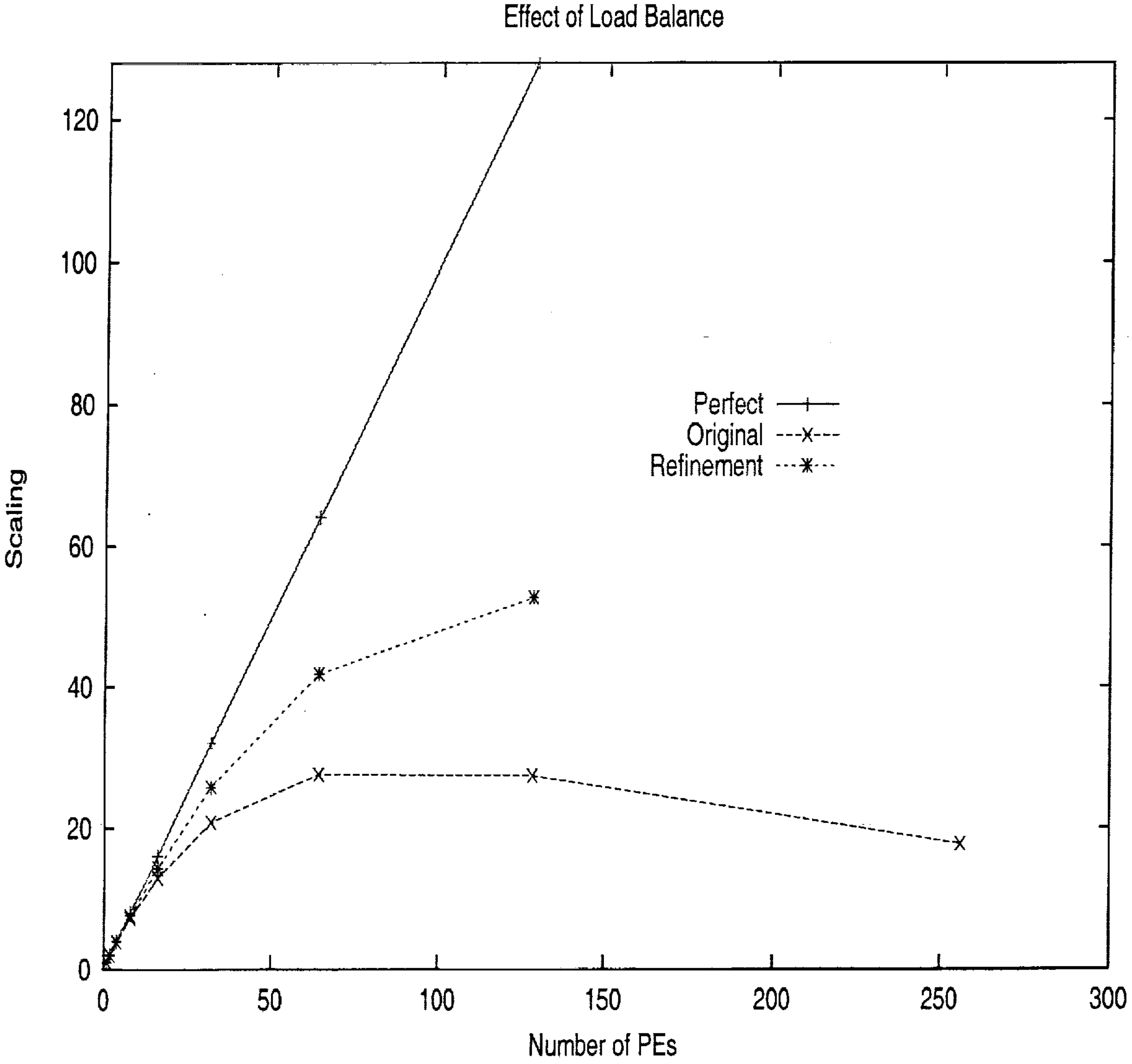

Figure 4: Improvements in scalability of MPI PSAS kernel due to load balancing.

Popular Summary

The Computational Complexity, Parallel Scalability, and Performance of Atmospheric Data Assimilation Algorithms

Authors: P.M. Lyster, J. Guo, T. Clune, and J.W. Larson

Institution: NASA/Data Assimilation Office

Submitted to: Computers in Science and Engineering

Source: ftp://dao.gsfc.nasa.gov/pub/papers/lyster/submitted.pdf

Data Assimilation algorithms and their parallel implementations are of considerable importance to Earth Science computational systems for operational as well as research applications. However, to date there have been few, if any, substantive publications on the complexity of the algorithms used in these systems. This paper addresses the computational complexity and scalability of distributed-memory parallel implementations of a number of important algorithms in use at NASA's Data Assimilation Office. The work has been written for a general scientific audience, but contains detailed analyses that should be of use to mainline software developers in Earth Science. We expect that information presented in this work can be used as a starting point in estimating the requirements for future data assimilation systems, in particular, parallel implementations using the Message-Passing Interface (MPI). We also expect that this paper can be used as a starting point for future work on characterizing complexity of advanced methods and algorithms in data assimilation.

Following a general overview of the algorithms, we presents formulae that may be used to obtain back-of-the envelope estimates of the number of floating point operations of an implementation of the Goddard Earth Observing System Data Assimilation System (GEOS-2 DAS) as well as a Kalman filter that is used to assimilate trace gas measurements. We present the timing profile (measured as a percentage of CPU time on 8 processors of an SGI Origin computer) for GEOS-2 DAS, and show how this profile is distributed over about half a dozen sub-components of the large system. The final thrust of the paper is two calculations that estimate the scalability of distributed-memory parallel implementations of two highly coupled sub-components of GEOS-2 DAS: gridpoint-based transport schemes and large semi-dense matrix-vector multiplications. We also discuss how Amdahl's law may provide a strong limitation on achieving better scalability of parallel computational systems due to both "unparallelizable" and "unparallelized" algorithms. This particularly applies to parallel Earth Science computational systems where even attempts to change the resolution or the size of the input dataset must be coordinated with scientific development and validation.