NASA/CR-97- 206341

IN-45-CR

301T

096651

Final Report:

Parallelization and Visual Analysis of Multidimensional Fields: Application to Ozone Production, Destruction, and Transport in Three Dimensions

Grant No. NAGW-3886

Karsten Schwan

October, 1997

College of Computing
Georgia Institute of Technology
Atlanta, Georgia 30332
email: schwan@cc.gatech.edu
fax: 404-853-9378

List of investigators:
Karsten Schwan, College of Computing
Fred Alyea, Earth and Atmospheric Sciences
M. William Ribarsky, College of Computing
Mary Trauner, Information Technology
Jeremy Heiner, PhD student, College of Computing
Milind Nanal, PhD student, College of Computing
Beth Schroeder, PhD student, College of Computing
Jeff Waldrop, MS student, College of Computing
Ray Wang, postdoc, Earth and Atmospheric Sciences
Song Zou, PhD student, College of Computing

List of associated investigators:
Greg Eisenhauer, Research Scientist II, College of Computing
Vernard Martin, PhD student, College of Computing
Jeffrey Vetter, PhD student, College of Computing
Derek Cunnold, Senior Research Scientist, Earth and Atmospheric Sciences

Undergraduate Students:
Onome Okuma, CS, Robert Sumner, CS
Davis King, CS, Roman Khramez, CS

1 Introduction

This final report has four sections. We first describe the actual scientific results attained by our research team, followed by a description of the high performance computing research enhancing those results and prompted by the scientific tasks being undertaken. Next, we describe our research in data and program visualization motivated by the scientific research and also enabling it. Last, we comment on the indirect effects this research effort has had on our work, in terms of follow up or additional funding, student training, etc.

2 Scientific Application and Analysis

2.1 Research Contributions to Atmospheric Sciences

As part of this research, we have constructed a parallel, three-dimensional, spectral dynamical/chemical transport model. This model's ability to run on most modern parallel machines and also across distributed compute engines has led to several advances in atmospheric science research. These advances are due in part to the improved computational performance offered by this model compared to its previous sequential version and in part to the online monitoring, steering, and visualization capabilities included with its parallel realization. Last, most recently, the transport model has been enlarged to simultaneously incorporate a number of additional atmospheric chemical constituents.

During the past 12 months, R. Wang and Derek Cunnold were the major contributors to this effort, following the previous work of T. Kindler, Derek Cunnold and Fred Alyea who, along with G. P. Lou, developed the original dynamical/chemical model and its sequential implementation. Due to their efforts, for the first time it is possible for the full capabilities of the real-time visual rendering of the model variables to be utilized to study the interactions of different chemical constituents in the atmosphere and to simulate both local and global changes in one or more of the constituents. Such changes might be due to bomb bursts, volcanos or other fairly rapid processes. This is accomplished through the "program steering" features built into the visual package.

Initially, to test the model calculations, two constituents which do not chemically interact were selected, and their time dependent global solutions were compared with single constituent model runs. The single constituent version of the model is a scientifically somewhat limited but computationally very efficient dynamical/chemical transport model and has been used for investigations into the stratospheric properties of N2O, Carbon 14 and associated dynamical flow fields [8, 6, 5]. Interesting research results attained with this model and the chemical constituents concern insights concerning atmospheric transport reported in [8] and comparative evaluations of NASA vs. UKMO observational data sets [7].

During the past year, our efforts (which continue today), involve the parallelization and substantial modification of a complex chemical model obtained from M. Pirre of the Laboratoire de Physique et Chimie de l'Environment, CNRS, Orleans, France [10], and the integration of this code with the parallel/dynamical transport model. The resulting, complex dynamical/chemical model of the earth's atmosphere incorporates 31 chemical compounds involving 60 gas phase reactions and 15 photochemical reactions, all of which are evaluated at each time step on a global basis, either on the model's three-dimension grid mesh or in the spectral domain. Preliminary computations with this new tool have been made for model evaluation purposes. Due to the complexity of the physical and chemical problems, it is expected that more such evaluations will be required.

The new 3-D chemistry transport model is designed for realistic stratospheric ozone simulations but problems dealing with other constituents, methane for example, can also be studied. At present, we are conducting a case study of the ozone Quasi-Biennial Oscillation (QBO) during both the easterly and westerly phases. As part of this work, a relationship between the ozone and methane in the stratosphere that may be looked at with the 3-D chemistry transport model is the link between the observed asymmetric ozone depletion in the upper stratosphere and hemispheric differences in methane distributions.

2.2 Papers

T. P. Kindler, The development of supercomputing tools in a global atmospheric chemistry model and its application on selected problems in global atmospheric chemistry modeling. *PhD Thesis, School of Earth and Geophysical Sciences, Georgia Institute of Technology, Atlanta*, March, 1996.

G. P. Lou, F. N. Alyea, D. M. Cunnold, and T. P. Kindler, N2O transport in a 3D model driven by UKMO winds. To appear in *Journal of Geophysical Research*, 1997.

T.P. Kindler, D. M. Cunnold, F. N. Alyea, W. L. Chameides, and G. P. Lou, A comparison of CLAES N2O simulations using 3D transport models driven by UKMO and GSFC assimilated winds. Submitted to *Journal of Atmospheric Sciences.*

Thomas Kindler, Karsten Schwan, Dilma Silva, Mary Trauner, and Fred Alyea, A Parallel Spectral Model for Atmospheric Transport Processes, *Concurrency: Practice and Experience*, vol. 8, no. 9, November, 1996.

3 High Performance Computing

Current communication tools and libraries for high performance computing are designed for platforms and applications that exhibit relatively stable computational and communication characteristics. In contrast, additional complexities exist in terms of dynamic behaviors for (1) mixed environments in which high performance applications interact with multiple end users, visualizations, storage engines, and I/O engines - termed 'distributed laboratories' in our research - and (2) high performance collaborative computing applications in general. The atmospheric modeling application and its use by earth and atmospheric scientists here at Georgia Tech have given rise to exciting new research streams in high performance research: (1) a focus on complete experimentation environments, or laboratories, rather than the broader HPC community's previous focus on individual, large scale parallel and distributed applications, and (2) a focus on the online interaction with computational tools in these environments. The first has led to the creation of a broader, multi-year and multi-faculty research effort, called *Distributed Laboratories*. This effort's highly visible research has involved the creation of a National Advisory Board to set its direction (comprised of both industry and academic researchers), it has led to the creation of a state of the art computing facility at Georgia Tech (with funding from the National Science Foundation and from industry), and it has resulted in strong multi-faculty research efforts addressing topics ranging from high performance distributed object/agent applications to high performance communications. Most recently, it has resulted in the inclusion of Georgia Tech in the NCSA center's renewal effort as a member of Team C as well as in Intel's high performance computing initiative (involving the donation of approximately $4,000,000 of Intel equipment over three years). The second research stream has led to strong research in program steering, including research on novel visual interfaces for program steering, the use of immersive interfaces for steering complex scientific applications, and research on the middleware infrastructure required for online monitoring and steering. All of these efforts are ongoing and are currently funded by the National Science Foundation, DARPA, other branches of the DoD, and by industry support.

In the remainder of this section, we focus on specific software artifacts and research papers attributed to this NASA-funded effort.

3.1 Research Contributions

Distributed laboratories are environments where scientists and engineers working in geographically separated locations share access to interactive visualization tools and large-scale simulation computations, share information generated by such instruments, and collaborate across time and space to evaluate and discuss their results. The intent of our research is to permit scientists, engineers, and managers at geographically distinct locations (including individuals telecommuting from home) to combine their expertise in solving shared problems by allowing them to simultaneously view, interact with, and steer sophisticated computation instruments executing on high performance distributed platforms. Research efforts being undertaken at Georgia Tech that address the topic of distributed laboratories are:

- Development of steering and monitoring tools and infrastructure used in the online observation and manipulation of scientific computations developed jointly with end users.
- Creation of middleware to transport the on-line monitoring and steering events, and exploration of its runtime adaptation to adjust event streams to current system loads and monitoring/steering needs.
- Understanding and constructing visualization support that permits the definition of appropriate visual abstractions and their efficient representation on 2D and 3D graphical displays.
- Creating abstractions and infrastructure that jointly enable multiple users to collaborate across distributed underlying machines and via the same or different computational tools.

The recent outcomes of the first two items, the steering and monitoring tools and infrastructure and the middleware, are summarized next. Visualization and collaboration work is described in Section 4.

Program Monitoring and Steering

Monitoring. Initially, our research focussed on the efficient, online monitoring of threads-based parallel programs, resulting in insights concerning the definition of alternative monitoring constructs embedded in application code, the construction of effective online monitoring support infrastructures co-resident with applications on shared memory parallel machines, and the creation of monitoring abstractions that give rise to flexible methods for online monitoring. Recent results include the evaluation of the performance effects of using alternative monitoring abstractions (attained in part with additional NASA student support and with support from the DOE via an extended student internship at Los Alamos Laboratories) [14]. Results also include the definition of quality of service constraints for online monitoring and the application of these constraints to distributed computational instruments operating in conjunction with the atmospheric model [9]. Next steps will include the creation of a *real-time monitoring infrastructure* so that specific monitoring rates may be guaranteed across distributed and parallel systems alike. This is particularly important when monitoring is combined with online program steering[14] or with online program adaptation[12].

Steering. We focus on the use of steering for program exploration rather than data exploration. Namely, we assume that scientists wish to use steering to explore different alternatives when running their large-scale simulations. Sample exploratory actions we have studied include alternative settings to selected simulation parameters, the use of alternative mathematical methods for computing certain simulation characteristics, and the online comparison of observational data with simulation output to drive the simulation and to detect model inaccuracies. Our approach to such steering actions is distinct from past work on program adaptation in its emphasis on user-driven steering actions. Namely, by providing users with rich graphical interfaces with which they may study simulation and observational data and the comparison of both, we can then enhance these same interfaces with graphical (and textual) constructs through which steering may be performed. Such 'steering by data manipulation' provides a natural way with which steering and data analysis/evaluation actions may be integrated. Our next steps in this work focus on the access to and use of very large data sets and complex computations, where a single graphical action by an end user may trigger significant changes in computations or may result in accesses to large amounts of data. Such user-triggered actions must imply changes outside the graphical environment itself or they will remain limited in scope and size to the single machine on which the user interface resides. The configurable middleware work described in the next section addresses this topic. First, however, we describe the monitoring and steering infrastructure(s) developed as part of our research.
The Falcon [4] Monitoring and Steering System is the basis on which all of our group's online program monitoring and steering is performed. Namely, Falcon provides the basic infrastructure for the online capture, buffering, and transport of events from a parallel or distributed application program and 'into' it from monitoring and steering servers associated with it. Prior to program execution, a program is instrumented

with application-specific constructs for monitoring or steering, typically by the end user. At runtime, Falcon's threaded servers associated with the program (and located in the program's address space) capture, buffer, and forward monitoring events and also inject appropriate steering events into the application. In earlier work, Falcon's utility was evaluated on shared memory platforms with performance monitoring tasks, using a parallel molecular dynamics application. In our current work, Falcon's server threads or processes monitor and steer the atmospheric modeling code developed in this project. It is also the basis on which runtime program adaptation for embedded applications is being explored [12].

Using Falcon as a basis, two different prototypes of steering toolkits have been developed and evaluated. Both prototypes share the same monitoring infrastructure, but differ in the steering and monitoring abstractions they implement. More importantly, they each explore different research goals.

The Progress Steering Toolkit [13] has been created to understand in detail the runtime overheads of steering and monitoring with alternative functionality and implementations ascribed to each and on shared memory computing platforms. When using Progress end users instrument their applications with library calls and then steer parallel applications with the Progress runtime system. Progress provides monitoring abstractions like sensors and probes and steering abstractions called *actuators*. Once created, the instances of such abstractions associated with the application being steered are known to and manipulated by Progress' two runtime components: (1) a steering server executing in the same address space as the target program and capable of inspecting and manipulating program state, and (2) a potentially remote client providing command and graphical interfaces. The application is steered via explicit user actions at the remote client, via algorithms executed at the client or server, or both. Also present in Progress is a 'steering language' with which sets of steering actions may be stated and even optimized with respect to the perturbation associated with monitoring and steeering or with respect to the latencies of such actions.

The Eagle toolkit is exploring a more 'natural' model of steering than the earlier models offered by Falcon and Progress. Specifically, Eagle offers true object abstractions for monitoring and steering that may be created dynamically, in servers or clients, and associated with program instrumentation at runtime. The intent is to provide within the monitoring and steering infrastructure 'mirrors' of application-level objects so that it appears to developers that they are actually adjusting application objects when they are extracting state from or manipulating the target application. The technical contributions of Eagle include the formulation of its object model, the implementation of its event infrastructure implementing mirror objects and their efficient association with target applications, and the runtime optimization of events and event transport to deal with heterogeneous and dynamic target execution platforms.

Middleware for Interactive High Performance Applications.

Current middleware for high performance applications tends to focus on the efficient representation and execution of the application itself, not on the execution of this application in the context of additional tools it uses or of interfaces with which it is used. To address these shortcomings, we have developed the DataExchange library [2], a communication infrastructure supporting high performance interactive and collaborative applications. DataExchange addresses the interactive and dynamic nature of high performance applications by providing facilities for naming and locating data sources, for dynamically connecting to those data sources and for automatic and configurable redirection of data flow. Further, to support flexible data processing, analysis and reduction, DataExchange also offers an active-message-style message processing facility that serves to refine and extend DataExchange's data flow management functionality by making content-based message forwarding decisions. Recent enhancements to DataExchange include C++ and Java interfaces to the library, support for connectionless protocols (e.g., UDP), and implementation of a thread-safe model.

High performance interactive and collaborative applications are clearly heterogeneous and may be more strongly described as *diverse*. This means that, in addition to operating in an environment where machine-level representations of elementary types may differ, the various components of the application may be developed in a much less tightly coupled manner than a traditional parallel or distributed program. One implication of this diversity is that it is not generally practical for all of the components operating in the distributed laboratory to agree at compile-time on the formats of all data records to be exchanged during execution. To meet this need and others, the DataExchange library is based on Portable Binary I/O (PBIO) [1], a binary I/O package which provides a variety of mechanisms for handling data in a diverse environment. In addition to features such as named data types and the ability to perform type conversions between matching fields of different basic types, PBIO supports more elaborate and flexible typematching than either PVM or MPI. Recently, PBIO has been extended with dynamic code generation for decoding binary files at the receiver. Dynamic code generation is implemented using the Vcode package [3] developed at MIT.

Our next steps in this research are building on the high performance 'interactivity infrastructure' provided by DataExchange to provide interfaces that are compliant with emerging industry standards and to enable additional functionality not otherwise accessible to the high performance computing community. In particular, layered above DataExchange, we are completing the implementation of a high performance object substrate offering IDL interfaces so that CORBA-compliant objects as well as Java objects may interact with this infrastructure. Technically interesting about our approach is our ability to associate attributes with such IDL descriptions that permit the online configuration of objects. In addition, we are constructing an event infrastructure that supports the efficient transport and analysis of monitoring and steering events across distributed systems, including the specification of quality of service constraints for such events.

3.2 Software Artifacts

Falcon. Falcon is a system for application level on-line monitoring and steering large-scale parallel programs. In addition, it provides a graphical interface for visualizing thread-level information extracted through the monitoring mechanism, and a reorder filter that imposes a partial order on a stream of thread-level events.

Progress. Progress [13] is a toolkit for developing steerable applications. The toolkit provides sensors, probes, actuators, function hooks, complex actions, and synchronization points. Progress' server uses Pthreads or a Mach-compatible Cthreads libraries in its runtime support. It also offers language constructs for formulating and optimizing sets of steering actions.

Eagle. Eagle offers notions of mirror objects and event channels for remote association of monitoring and steering abstractions with a target application. Event channels offer runtime configuration for high performance, for heterogeneous and dynamically changing configurations of target applications.

DataExchange. The DataExchange library is a communication infrastructure supporting high performance interactive and collaborative applications.

PBIO. PBIO is a binary I/O package that supports named data types, the ability to perform type conversions between matching fields of different basic types, and flexible type matching such that receivers are not required to specify every field in the types to be exchanged.

3.3 Published Research

Greg Eisenhauer, Weiming Gu, Eileen Kraemer, Karsten Schwan and John Stasko, Online Displays of Parallel Programs: Problems and Solutions, *accepted for publication in Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA '97)*, 1997.

Greg Eisenhauer, Beth Schroeder, and Karsten Schwan, DataExchange: High Performance Communications in Distributed Laboratories, to appear in *Journal of Parallel Computing*, 1997.

Greg Eisenhauer, Beth Schroeder, Karsten Schwan, Vernard Martin, and Jeff Vetter, DataExchange: High Performance Communications in Distributed Laboratories, *9th Int'l Conference on Parallel and Distributed Computing and Systems*, October, 1997.

Beth Schroeder, Greg Eisenhauer, Karsten Schwan, Jeremy Heiner, Vernard Martin, and Jeffrey Vetter, From Interactive Applications to Distributed Laboratories, to appear in *IEEE Concurrency*, 1997.

Beth Schroeder, Greg Eisenhauer, Karsten Schwan, Fred Alyea, Jeremy Heiner, Vernard Martin, Bill Ribarsky, Song Szou, Mary Trauner, Jeffrey Vetter and Ray Wang, Framework for Collaborative Steering of Scientific Applications, *Science Information Systems Newsletter*, Vol. IV, No. 40, 1997.

Greg Eisenhauer, Weiming Gu, Eileen Kraemer, Karsten Schwan and John Stasko, Online Displays of Parallel Programs: Problems and Solutions, *Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA '97)*, 1997.

Greg Eisenhauer, Beth Schroeder and Karsten Schwan, From Interactive High Performance Programs to Distributed Laboratories: A Research Agenda, *Proceedings SPDP'96 Workshop on Program Visualization and Instrumentation*, October 1996.

Greg Eisenhauer, Weiming Gu, Thomas Kindler, Karsten Schwan, Dilma Silva and Jeffrey Vetter, Opportunities and Tools for Highly Interactive Distributed and Parallel Computing, *Parallel Computer Systems: Performance Instrumentation and Visualization*, Rebecca Koskela and Margaret Simmons, editors, ACM Press, 1996.

Jeffrey Vetter and Karsten Schwan, High performance computational steering of physical simulations, *Proceedings of the 11th International Parallel Processing Symposium (IPPS 97)*, 1997.

Vernard Martin and Karsten Schwan, ILI: An Adaptive Infrastructure for Dynamic Interactive Distributed Systems, submitted to *4th International Conference on Configurable Distributed Systems*, 1997

Daniela Rosu, Karsten Schwan, Sudhakar Yalamanchili, and Rakesh Jha, On Adaptive Resource Allocation for Complex Real-Time Applications, to appear in *Proceedings of the 18th IEEE Real-Time Systems Symposium (RTSS)*, December, 1997

4 Visualization

4.1 Research Contributions

Visual Steering and Analysis of Simulations

We have developed a heterogeneous environment for visual steering of computer simulations [16]. Visual steering implies two-way communication through 2D and 3D graphical data representations to bring about user involvement with the calculations as they occur. One must be able to insert on-the-fly parameter changes, even over 4D (3D + time) regions of the simulation and see results dynamically updated. The environment differs from previous steering systems in that it combines (i) a high level set of tools for instrumenting simulations; (ii) tools for efficient data exchange; and (iii) a visual steering interface for direct interaction with visualizations of the data. The instrumenting tools can be inserted by users into their codes without having to know about networking protocols or machine-dependent binary formats. The visual interface allows visualization of a variety of data from different applications since it employs general graphical objects that users can map to variables of their choice. The steering tools are integrated with the middleware that controls and monitors the distributed simulation and with the visualization/analysis tools so that one can apply steering commands simply by knowing the I/O formats of the instrumentation and without knowledge of the middleware structure.

Employing the instrumentation and I/O libraries, users can reconfigure their application codes to show new variables or accept different steering controls. This information is transmitted automatically to the visual steering environment. Since the steering tools appear in the data visualization space, users can steer based on the detailed 3D structure and time-dependent behavior of the data. The visualization/analysis tools can bring out correlations between variables or can focus on selected regions of the data; these analyses can be used to refine the steering or more closely investigate the results of the steering input. Finally, the same visualizations can function both as data representations and as steering inputs. We demonstrate this with isosurfaces. These may define regions of physical interest but irregular geometry where spatial distributions of selected variables can be inserted.

Recent work includes support for time-dependent steering. Here time is treated on exactly the same basis as the spatial dimensions so there is a 4D environment, three shown spatially and one through animation. The steering interface is built upon a flexible visualization/analysis system. This permits the immediate display of time-dependent results from the dynamic simulations and refined interaction with the results to bring out the character and correlations of multivariate data. The user can then launch new simulations at any stage in this exploration using the visualization to define and focus the simulation parameters, region of interest, and time frame.

Collaboration via Visual Interfaces

We have developed a visual steering approach that supports collaboration between distributed users, allows asynchronous collaboration, supports project histories, and supports different roles [15]. Collaboration is achieved through sharing and direct manipulation of 3D interfaces for steering and analysis. With the collaboration tools a user may steer a simulation and share both the steering input and visual output with one or more geographically distributed colleagues.

Our approach, which employs Open Inventor for interaction and rendering, relies on the directed acyclic graph (DAG) abstraction employed by many modern user interface libraries including Open Inventor. Collaboration is built into the Open Inventor graphics library by utilizing the DAG abstraction. Since the data visualizations are contained in one or a few connected Inventor nodes, collaboration is achieved through maintaining consistency between shared nodes in a relevant portion of the DAG structure. This consistency is achieved through the introduction of a *sharer node* into the DAG where the sharer node is responsible for maintaining consistency between shared child nodes, including the nodes used for data visualizations.

We have developed three roles for use in the collaborative tools. In the first role, a "teaching" user offers to "share" nodes that simply enable other users to follow what the teacher is doing, perhaps at different levels of detail. In the second role, two or more users can be aware of each others' actions by seeing where other users are actively exploring data or making changes but not being coupled to those explorations or changes. A third, more complex scenario is illustrated with the atmospheric model. Here users not only share views of each other's actions but also pass control of steering objects between each other and see the results of their steering inputs.

4.2 Software Artifacts

GlyphMaker/Explorer based visualization interface. The original visual analysis tool used Glyphmaker [11] built on top of the SGI Explorer environment to help the user explore relations between spatially complex, time dependent data.

Open Inventor based visualization interface. Described in the preceding section, the Open Inventor based analysis tool allows asynchronous collaboration and 4D steering in addition to visually displaying complex, 3D atmospheric modeling data.

4.3 Published Research

Song Zou, William Ribarsky, Yves Jean, Jeremy Heiner, Karsten Schwan, Bobby Sumner, and Onome Okuma, Collaboration and Visual Steering of Simulations, *Proceedings SPIE Conference on Visual Data Exploration and Analysis IV*, 1997.

Yves Jean, Song Zou, William Ribarsky, Karsten Schwan, Bobby Sumner, and Onome Okuma, A Heterogeneous Environment for Visual Steering of Computer Simulations, submitted to *IEEE Computer Graphics and Applications.*

Song Zou, William Ribarsky, Jeremy Heiner, and Karsten Schwan, Collaboration and Steering, GIT-GVU-97-22.

5 Funding and Indirect Effects

This NASA-sponsored effort has been one of the most interesting tasks undertaken by our research group during the last five years. Specifically, due to its emphasis on cooperation between end users and computer scientists, the Computer Science team was able to use project insights to develop an entirely new research direction in the area of high performance computing. This direction concerns the *interaction* with high performance applications rather than their efficient execution, as addressed by most major national efforts (e.g., the metacomputing efforts, MPI, the parallel scalable I/O initiative, etc.). This enabled our team to generate additional funds and to undertake related research efforts that have already had significant additional impact.

Efforts leveraging this NASA grant have resulted in both an excellent equipment infrastructure now available to our research group and in significant additional research funding. It has also resulted in the exposure of many U.S. graduate and undergraduate students to the scientific applications and high performance computing technologies of interest to NASA (more than 50 PhD students, 20 MSc. students, and 20 undergraduates, including more than 10 women and minority students – detailed data available upon request). For brevity, we do not list these students individually. We also do not report the indirect results in terms of technical output as papers, software products, and technical reports.
New funding includes:

- Karsten Schwan (PI), jointly with R. Das and G. Eisenhauer, "Program Steering: From Interactive Programs to Distributed Laboratories", NSF New Technologies Program, Jan. 1998 - Dec. 2000, $406,040.
- Karsten Schwan (Investigator), David Rosen (PI), "Computational Issues in The Distributed Rapid Prototyping Facility", NSF Engineering Directorate, Oct. 1997 - Sept. 2000, $116,248.
- Karsten Schwan (PI), jointly with Richard Fujimoto, Ellis Johnson, Uzi Landman, David McDowell, Suresh Menon and Lakshmi Sankar, William Ribarsky, Jarek Rossignac, and Sudharkar Yalamanchili, "Proposal to Intel Corporation for Utilization of Advanced Intel Based Platforms in Computationally Demanding Tasks: Interactive Computational Laboratories", $3,700,000 for Intel equipment and including cash grants, Sept. 1997 - Aug. 1999.
- Karsten Schwan (Investigator), NSF New Technologies Program, subcontract from NCSA, Team C (Dan Reed, PI), "Using Immersive Interfaces for Program Steering", Sept. 1997 - Aug. 2002, approx. $500,000.
- Karsten Schwan (PI), jointly with R. Fujimoto and S. Yalamanchili, "The Critical Systems Laboratory – High Performance Distributed Simulation on Low Cost Platforms", Army Research Office, Jagdish Chandra Program Manager, $243,043 from ARO, $14,390 cost sharing, March 1997 - Feb. 1998.
- Karsten Schwan (PI), jointly with M. Ahamad, M. Ammar, R. Fujimoto, and S. Hudson, "Interactive computing on Cluster Computers", NSF CISE equipment grant, $105,000 from NSF, $50,000 cost sharing, July 1995-June 1997.
- Karsten Schwan (PI), jointly with R. Fujimoto, S. Hudson, J. Limb, and M. Ahamad, "Distributed Laboratories", NSF CISE Research Infrastructure Grant, approx. $1,200,000, Aug. 1995 - July 1998.

References

[1] Greg Eisenhauer. Portable self-describing binary data streams. Technical Report GIT-CC-94-45, College of Computing, Georgia Institute of Technology, 1994. *(anon. ftp from ftp.cc.gatech.edu)*.

[2] Greg Eisenhauer and Beth Schroeder. The dataexchange library. Technical Report GIT-CC-96-17, College of Computing, Georgia Institute of Technology, 1996. http://www.cc.gatech.edu/tech_reports.

[3] Dawson R. Engler. Vcode: a retargetable, extensible, very fast dynamic code generation system. In *Proceedings of SIGPLAN Conf. on Programming Language Design and Implementation (PLDI'96)*. University of Linz, May 1996.

[4] Weiming Gu, Greg Eisenhauer, Eileen Kraemer, Karsten Schwan, and John Stasko. Falcon: On-line monitoring and steering of large-scale parallel programs. In *Frontiers 95*, 1995.

[5] T. P. Kindler. *The development of supercomputing tools in a global atmospheric chemistry model and its application on selected problems in global atmospheric chemistry modeling*. PhD thesis, School of Earth and Geophysical Sciences, Georgia Institute of Technology, March 1996.

[6] T. P. Kindler, D. M. Cunnold, F. N. Alyea, W. L. Chameides, and G.P. Lou. A comparison of CLAES N2O simulations using 3D transport models driven by UKMO and GSFC assimilated winds. *Journal of Atmospheric Sciences*, 1997.

[7] Thomas Kindler, Karsten Schwan, Dilma Silva, Mary Trauner, and Fred Alyea. A parallel spectral model for atmospheric transport processes. *Concurrency: Practice and Experience*, 8(9):639–666, November 1996.

[8] G. P. Lou, F. N. Alyea, D. M. Cunnold, and T. P. Kindler. N2O transport in a 3D model driven by UKMO winds. *Journal of Geophysical Research*, 1997.

[9] Vernard Martin and Karsten Schwan. Ili: An adaptive infrastructure for dynamic interactive distributed systems. In *Submitted to 4th International Conference on Configurable Distributed Systems*, 1997.

[10] R. Ramaroson, M. Pirre, and D. Cariolle. A box model for on-line computations of diurnal variations in a 1-d model: potential for application in multidimensional cases. *Ann. Geophysicae*, 10:416–428, 1992.

[11] William Ribarsky, Eric Ayers, John Eble, and Sougata Mukherjea. Using Glyphmaker to create customized visualizations of complex data. *IEEE Computer*, July 1994.

[12] Daniela Rosu, Karsten Schwan, Sudhakar Yalamanchili, and Rakesh Jha. On adaptive resource allocation for complex real-time applications. In *To appear in Proceedings of the 18th IEEE Real-Time Systems Symposium (RTSS)*, December 1997.

[13] Jeffrey Vetter and Karsten Schwan. Progress: a toolkit for interactive program steering. In *Proceedings of the International Conference on Parallel Processing*, 1995.

[14] Jeffrey Vetter and Karsten Schwan. High performance computational steering of physical simulations. In *Proceedings of the 11th International Parallel Processing Symposium (IPPS 97)*, 1997.

[15] Song Zou, William Ribarsky, Jeremy Heiner, and Karsten Schwan. Collaboration and steering. Technical Report GIT-GVU-97-22, Georgia Institute of Technology, 1997.

[16] Song Zou, William Ribarsky, Yves Jean, Jeremy Heiner, Karsten Schwan, Bobby Sumner, and Onome Okuma. Collaboration and visual steering of simulations. In *Proceedings of SPIE Conference on Visual Data Exploration and Analysis IV*, 1997.