Influences on the Solution Process for Large, Numeric-Intensive Automotive Simulations

Myron Ginsberg, Ph.D.

HPC Research and Education
Farmington Hills, Michigan 48335-1222 USA
m.ginsberg@ieee.org

Abstract. This paper focuses on the performance limitations of solving large automotive simulation applications. Some recommendations are given for improving the solution process. The content is based upon the author's experience in the U.S. automotive industry and on his recent investigation of U.S. automotive applications while he was a NASA/ASEE Summer Research Fellow in the Computational AeroSciences Team at NASA Langley Research Center. Most of the applications are in the following areas: computational fluid dynamics (CFD), crash analysis, Noise-Vibration-Harshness (NVH) modeling, manufacturing problems such as metal forming processes, and multidisciplinary optimization problems such as combinations of NVH and crash analysis.

1 Introduction

The global competitiveness of each U.S. automotive company is dependent upon how effectively it can continually reduce lead time between concept and production of new vehicles. By doing so, the individual auto companies can rapidly respond to utilization of new technologies, shifting customer needs and desires as well as competition from both European and Asian rivals. In 1980, the lead time between concept and production was 60 months. In early 2001 it is between 20 and 24 months with an industry target of 18 months or less. To meet such a goal, several changes are required: (1) The amount of physical prototyping must continue to be reduced and replaced by realistic three-dimensional, math-based/physics-based computer modeling; (2) Such computer models must employ very aggressive use of modest and massively parallel hardware, software, and algorithm technologies; (3) To accomplish (1) and (2) above requires extensive multidisciplinary interactions and sharing of resources with industrial/government consortia beyond the automotive companies in which the problems arise; (4) Success requires access to new computer architectural resources as well as hardware, software, and algorithms to experiment with optimal mapping of those resources to produce the fastest, most accurate, and economical solutions.

In the following sections we discuss features of the automotive computational environment (Section 2), properties of the large-scale automotive applications

V.N. Alexandrov et al. (Eds.): ICCS 2001, LNCS 2073, pp. 1189-1198, 2001.

investigated (Section 3), roadblocks to a fast solution (Section 4), approaches to reducing computer solution times (Section 5), recommendations to improve performance (Section 6), summary and conclusions (Section 7). Additional details about the specific problems considered and the nature of the investigation as well as the motivation for the study can be found in [1] and [2], respectively.

2 Automotive Computational Environment

Until the mid 1980s, most automotive companies depended heavily on in-house developed proprietary software for solving their scientific and engineering problems. As the computer technology began to rapidly evolve and the problems became larger and more complex, it became quite apparent that it would be much more cost effective to outsource the development of such software. The net effect was the creation of many commercial independent software vendors (ISVs). At present, approximately a dozen major ISVs serve the growing needs of the automotive industry. These ISVs are generally relatively small companies which primarily focus on supporting one or more application codes that can be used to solve automotive problems on several computer platforms. Most of these codes have been written in FORTRAN and C but in recent years there has been a slow movement to utilize C++ and consider other object-oriented languages such as Java [3].

Global competition has forced the auto companies to drastically reduce their lead times between vehicle concept and production. Lead time reduction has been accomplished by a rapid transition from physical prototyping to math-based computer modeling and simulation. This in turn has required the need for accurate, realistic, and fast computer simulations of automotive problems, often requiring very aggressive use of parallel computing approaches.

The typical large-scale automotive problems are in four categories: (1) crashworthiness modeling; (2) noise, vibration, and harshness (NVH) applications; (3) computational fluid dynamics which includes both exterior aerodynamics and combustion modeling; (4) manufacturing processes such as metal forming applications. More details about all four categories can be found in [4], [5], [6], [7].

The computers for solving automotive scientific / engineering applications in the 1970s and early 1980s were usually the same traditional mainframes used for large data processing problems. Then in early 1984 General Motors Research acquired the first in-house Cray supercomputer in the world automotive industry [7]; the trend during the next seven years was for most of the world automotive companies to acquire their own Cray vector supercomputers as well as some of the Japanese supercomputers from NEC, Fujitsu, and Hitachi. From the mid 1990s until the present, the trend has been to acquire scalar concurrent parallel machines such as the IBM SP, SGI O2000 and O3000 series, HP9000 series, and Sun Enterprise 10000 systems. Many of the European and Japanese auto companies continue to use vector supercomputers especially those from the aforementioned Japanese vendors. In the U.S. most recently, only Ford Motor Company has continued to use several vector supercomputers [8].

3 Properties of the Large-Scale Automotive Problems

The author collected large-scale automotive applications submitted from General Motors, Ford Motor Company, and DaimlerChrysler. Details about this investigation as well as descriptions of all eight problems can be found in a forthcoming paper [1]. The investigation focused on identifying and documenting a diverse collection of large-scale, very important, compute-intensive automotive industry problems. At present, most of these problems cannot readily be solved in a commercial industrial environment because of computer limitations and/or memory or I/O requirements. The intent of this study was to follow up with many of these applications to determine if these problems could be economically solved by utilizing leading-edge hardware, software, and/or algorithm technologies that are not readily accessible within the auto companies. Below is a list of the eight problems submitted; see [1] for details about each of these applications.

- Transient analysis of a V6 exhaust manifold using a coupled 1D/3D Model
- Solving normal modes for models with 100,000 to 500,000 grids, approximately 600,000 to 3,000,000 Degrees of Freedom (DOF)
- Simulation of vehicle impact in real-world crashes
- Metal forming problem involving solution of 500,000 linear simultaneous equations, 1000 times in the entire analysis
- Establish statistical characteristics of vehicle road noise performance using the Monte Carlo simulation method
- Full vehicle external flow simulation
- Stochastic multidisciplinary shape optimization of crash and NVH
- Mid frequency vibration problems

There was no attempt to solve these problems due to the limited time available during the summer initial investigation; however, observations were made about the nature of the bottlenecks that prevented efficient solutions. Those will be discussed in the next section of this paper.

There were several observations that can be made about these problems. Although these applications were submitted in isolation from one another by three different auto companies, there was a remarkable similarity about the size and nature of all problems submitted. The increasing pressure on the auto companies to reduce lead time is forcing them not only to drastically increase the amount of computer simulation modeling but also to do so at an increasingly faster pace. The ultimate goal of automakers is to be able to create computer models of all vehicle attributes including realistic modeling of the possible interactions amongst subsystems. With such a capability, modifications can be easily accomplished in early design phases and much more economically than at times closer to actual production. To be successful requires fast turnaround of computer simulations of any subsystem and this usually means at least one complete detailed simulation per day and/or via overnight turnaround. None of the eight problems listed above are being solved that quickly at present. Although all these problems are now being solved in a parallel computing environment, much more aggressive and innovative approaches are needed to significantly speed up the solution process within the auto companies.

4 Roadblocks to a Fast Solution

The comments made in this section are not intended to place blame on any entity for impeding progress in solving these large problems efficaciously. Often times it has been too risky financially to make some of the needed changes and / or to do so would be too time-consuming in an industrial environment.

Several of the problems listed in Section 3 could indeed most likely be solved today much faster than they are currently being solved within the auto companies. Let's look at some of the impediments. As indicated in Section 2 the auto companies have become almost totally dependent on the ISV-based software that they all use to solve their large applications. This is both an economic blessing and a curse. It saves the auto companies the money they would have to spend in-house to develop the application software but at the same time it excludes immediate use of existing software that could solve the problem much faster than at present. All of the automotive companies are very reluctant to use a computer solution external to their ISV-based software even if the solution is faster and/or more accurate. Since these ISV-based companies are relatively small compared to the auto companies they serve, some do not have the people and computer resources nor the time to conduct extensive research of new and better algorithms or to perform extensive code rewrites of their existing software. The ISVs often need to prioritize in order to respond to the demands of a diverse customer base. Furthermore, when new computer hardware technology becomes available, the ISVs are understandably reticent to port to the new platforms because of the economics of doing so and not readily knowing how rapidly their customers will move to the new architecture. The net result is delays in using new algorithms and architectures to attack these large problems.

There is also a reluctance mostly on the part of the auto companies to port to new operating systems. Thus, for example, although several ISV-based software packages [9] have been ported to a Linux environment, this has not happened at auto companies.

Another impediment to computational improvement is the slow creation and integration of new algorithms into the ISV-based software. Because the ISVs do not have the manpower nor the time to research and create new algorithms and extensively test them, they must of necessity rely on other sources such as universities and/or government labs. Some ISVs have very strong links with academia but there still persist problems with rapid implementation of new academic algorithms in production versions of ISV-based codes. Traditionally, only industry pressure (and sometimes dollars) hastens the process.

Another related difficulty is that there are some inherent problems in algorithm creation that have yet to be resolved before accurate and totally realistic automotive simulations can be achieved. For example, it is very difficult to implement accurate error propagation analysis and do it reasonably fast. Secondly, in some cases the physics of the physical component interactions is not yet completely understood, thus potentially producing deceptive results in the computer simulation. Also the need to deal with multiphysics phenomena complicates the needed algorithm but must be dealt with successfully to promote realistic simulations.

An additional barrier is limited access to a wide variety of computer platforms that might provide faster execution on a specific automotive application than achieved by an auto company on its in-house computers. This situation is further compounded by the fact that a potentially faster machine might be inaccessible to an auto company or not yet even have a port of the ISV-based software that the automaker utilizes to solve such problems. In addition, it may be very difficult and/or time consuming to determine if indeed the external architecture would produce a better and faster results than on the existing in-house machine. This latter difficulty is exacerbated by the lack of any relevant existing hardware performance comparison data.

All the above-mentioned dilemmas are further aggravated by the lack of any significant software, algorithm exploration or hardware investigations by the in-house automotive people. They are rightfully focused on the vehicle engineering problems, not the software to provide accurate and fast simulations. Thus the burden generally falls unfairly on the ISVs but they too lack the resources to do extensive hardware performance comparisons of their software, especially as new platforms become available. Many of the issues described above fall into the cracks amongst the hardware vendors, the ISVs, the academic algorithm creators, and the auto companies. There obviously needs to be more interaction addressing these problems and day-to-day communications amongst these entities.

5 Some Current Activities to Improve Computer Solution

Despite the difficulties described in the previous section, there have been several recent activities which have significantly reduced computation time for these large-scale automotive problems. We present a few examples here.

Dr. Sobieszczanski-Sobieski (Leader of the Computational AeroSciences Team) of the NASA Langley Research Center was able to substantially reduce total wall clock solution time for an earlier problem from Ford Motor Company; details are given in [10]. Essentially this was an optimization of a car body for crashworthiness, NVH, and weight; this problem was formulated as a stochastic multidisciplinary shape optimization of crash and NVH using two automotive ISV-based codes (MSC's NASTRAN and Mecalog's RADIOSS). The goal is to obtain peak performance while design parameters (geometry shape, spot weld pattern, thickness, and material properties) are modified and at the same time reduce the variation of that performance; this is necessary because the design process based on nominal values is not adequate due to physical variation in manufacturing and material which can significantly impact the actual products. The problem attributes are: NVH model is a BIW-trimmed body structure without powertrain and suspension subsystems and using

MSC NASTRAN finite element model of 350,000 + Degrees of Freedom, normal modes, static stress and design sensitivity analysis using Solution Sequence 200 (in version 70.5); there are 29 design variables (sizing, spring stiffness); total computer time was dominated by the crash model (run using RADIOSS, version 4.1b); both NVH and crash models were executed on an SGI O2000.

Fine grain parallelism helped reduce optimization procedure total elapsed time of 292 hours to 24 hours for a single analysis using 12 processors on an SGI O2000. Coarse grain parallel computing for this problem also provided significant reduction in total elapsed time. The total net effect for this problem was a reduction from 257 days of elapsed computing time to 1 day. This was achieved by utilizing the MDO approach in a parallel computing environment without altering any of the NASTRAN or RADIOSS source code. This approach would also be effective for use on one or more of the problems referred to in Section 3.

There are several additional improvements that could be beneficial to the automotive simulation process but space limitation here prevents further discussion. The interested reader should look at [11], [12], and/or the references cited in the rest of this section. Many recent developments are reported in HPCWIRE (http://www.hpcwire.com) and references to specific articles are listed below.

In the visualization area, the following activities could positively stimulate further automotive simulation improvements. Virtual reality is being used to "simulate the entire process of automotive engine constituting from designing, manufacturing including the motion of the engine" [13]; this virtual engine factory will be developed using ESI software on a NEC SX supercomputer for a Japanese auto company. Penn State University and ESI are developing a system that makes it possible to do "virtual" tire testing; i.e., "a facility to roadtest a tire design virtually while the tire is still on the drawing board." [14] Mazda and MTS System Corp. have a vision of integrated simulation - analytical, physical and virtual." [15] Ford and U.K's Defence Evaluation Agency are developing "technology to create full-scale virtual digital models of prototype vehicles which gives Ford the ability to re-design features in real-time and the capability to hold multiple design reviews simultaneously." [16] There are efforts to combine tele-immersion and virtual reality "to create a 3-D visual environment similar to that depicted by the Holodeck on the Star Trek TV show." [17] Some interesting leading-edge work with BMW, SGI, and the University of Erlangen involves the VTCrash virtual reality software written in C++, implemented on SGI Onyx and Octane workstations and which allows computational steering within the model where the average model consists of 200,000 finite elements and the scene visualization "uses one processor of an Infinite Reality Onyx with at least 10 frames per second." [18]

In the HPC architecture area there are several developments which will directly stimulate automotive simulation. Cray just announced plans for a mid-2001 release of an Alpha-based Linux supercluster system in response to customer requests for a "re-architected T3E using leading off-the shelf cluster technologies." [19] "Beowulf class machines using commodity chips are economical machines for some HPC applications but not as capability efficient in throughput for conventional large-scale applications." [20] So far, they have not been utilized in the auto industry. Vector class supercomputers remain widely used in Europe and Japan and at Ford in the U.S. where they are used for "safety and structural analysis."[8] It is interesting to note in Germany that Fujitsu, NEC, Hitachi, and Cray (both T3E and SV1) supercomputers are used but "commercial users do not install the ultra high-end machines" but often share HPC hardware resources with academia; for example, at University of Stuttgart, 50% of HPC resources are shared by U of Karlsruhe and U of Stuttgart, 40% is used by Debis Systemhaus (subsidiary of DaimlerChrysler) and 10% is used by Porsche." [21] A new effort lead by IDC and involving industry and government HPC representatives is attempting to create new effective, realistic HPC performance metrics and benchmarks which could help industry to evaluate HPC architectures [22].

There are several software performance improvement projects which could be beneficial to the auto companies simulation efforts. SGI and ESI using an SGI O3000 with 96 MIPS R12000 processors and ESI PAM CRASH 2000 software achieved a "sustained rate of 12 GFLOPS for a BMW crash model; this was the highest level of performance ever achieved in computer crash simulation." [23] Cray and Mcube are focusing on accurate simulation of aeroacoustics such as "alleviating wind noises from side mirrors and driver's side structures and exhaust system noise." [24] Cray and MSC.Software Corp. have produced a "900% speedup in key calculations for NVH problems which implies massive NVH optimization jobs can now be performed as overnight turnaround rather than requiring several days." [25]

Detailed information about some of the above mentioned performance improvement activities is not readily available. Some views may be overly optimistic, represent targets rather than achievements and/or marketing hyperbole but it is the responsibility of each auto company to separate the wheat from the chaff.

6 Recommendations to Improve Problem Solving

From the remarks in the preceding two sections, it should be apparent that there are four areas in which improvements are needed: (1) increased use of visualization, (2) performance comparisons of HPC architectures, (3) algorithm development and dissemination, and (4) software performance improvements.

The shift from physical prototyping to computer simulation requires tightly integrated use of virtual reality and teleimmersion capabilities with the ISV-based applications software; progress in this area has been very slow. Active collaboration amongst auto companies, the NSF centers, government research labs, and ISVs could greatly enhance this effort. So far activities in this area have been very sporadic. Daily interactions amongst these entities are needed with clear goals to create immediate pragmatic impact on these large-scale problems.

Acquisition of new computer hardware requires thorough scrutiny of existing alternatives; however, the auto companies having to cope with severe time constraints, have often had to make decisions based upon very limited investigations and/or on unsubstantiated vendor claims. A "Consumer's Report" type of ongoing benchmarks should be created to provide objective performance comparisons using ISV-based software on realistic automotive problems across a variety of computer platforms. The results should be updated on a continual basis and available on the Internet, possibly on a subscription basis to interested auto companies. Such performance information would be helpful to the entire automotive community in making hardware selections. New practical and realistic metrics are needed to measure “real” performance; the recent efforts initiated by IDC [22] seem to be in the right direction although widespread automotive industry involvement so far appears to be minimal.

In the algorithm development area there needs to be a pragmatic focus on tech transfer from government and academic algorithms to the automotive ISV-based codes. For example, one of the problems listed in Section 3 is very likely to be solved much faster using a code available from ICASE [26] but this is unacceptable at present because the code is not part of the ISV-based software the automaker uses. A cooperative effort should be initiated with government and academic algorithm developers directly working with the ISVs to provide fast ports for important applications.

In the software area there is a need for the ISVs to transition their legacy FORTRAN and C based codes to an object-oriented environment such as C++, Java, or even object-oriented FORTRAN. Such a move would in the long run drastically reduce maintenance costs and time as well as help promote use of grid technologies [27] that are now emerging on the Internet and which would be very amenable to use with cluster machines for large applications.

Another software issue which should be addressed is the gap in the creation of the ISV-based software: a computational science gap, i.e., we have experts examining the physics and engineering issues and another group focusing on writing thousands of lines of code but there is no one in the middle to interface new software developments such as VR, grids on the Internet, and user friendly parallel computing tools; most of the work in those areas is being done in a few government labs and academia but transferring that expertise to the ISVs has been very slow. Hopefully, the new breed of computational scientists that are emerging from about two dozen universities in the U.S. and elsewhere will soon have positive impact on ISV progress in this area.

In the visualization area perhaps the automotive people should be seeking closer collaboration with the visualization people in the motion picture industry. Visualization technology (real-time animation and VR) is the cornerstone for successful comprehensive simulation in the automotive industry. An observation at BMW [18] indicates that ”30 percent of the efforts involved in a typical simulation go into the preprocessing phase and about 10 percent is taken up by the actual computation, while approximately 60 percent goes into the analysis and communication of the results. Clearly there is a strong incentive to reduce the last percentage through the implementation of insightful, intuitive visualization tools which allow effective communication between engineers.” [18] Real-time (or near real-time) computational steering is essential to the total automotive simulation process; faster hardware and improved visualization techniques with large datasets are rapidly making that possible. Such visualization capabilities as demonstrated by activities with BMW [18] need to be rapidly integrated into the automotive ISV-based software.

7 Summary and Conclusions

Conversion costs, performance concerns, time constraints, and desire for a stable computational environment are primary barriers to extensive adoption of new technology for large-scale automotive simulation problems. Improvements in software tools including visualization capabilities, creation of numerical libraries, use of optimizing compilers, and pragmatic cooperative academic/industrial/government interactions can stimulate auto industry movement to use of new technologies.

In some cases as in the Ford example in Section 5, faster results can be achieved on existing in-house machines using existing ISV-based software but utilizing multidisciplinary optimization techniques external to the ISV-based software in a parallel computing environment. Several of the problems listed in Section 3 could benefit from use of such strategies. Also there exist several codes available from government labs that could immediately speed up the solution process of some of the submitted problems and then be phased in to the ISV-based software via a cooperative effort between the ISV and the government lab involved.

8 Acknowledgment

I am very grateful to Dr. Jaroslaw Sobieszczanski-Sobieski (Leader- Computational AeroSciences Team, NASA Langley Research Center) for the opportunity to serve as a NASA/ASEE Summer Research Fellow in his group and to investigate large-scale automotive problems. Thanks to James P. Johnson (General Motors) and Dale Shires (Army Research Lab) for suggestions to improve this paper's readability.

References

1. Sobieszczanski-Sobieski, J., Ginsberg, M.: A Preliminary Investigation of Large-Scale High-Performance Computing Applications from the U.S. Automotive Industry, NASA /TM-2001-xxxxxx Computational AeroSciences Team, NASA Langley Research Center, Hampton, VA, to be published, 2001
2. Biedron, R.T., Mehrotra, P., Nelson, M.L., Preston, F.S., Rehder, J.J., Rogers, J.L., Rudy, D.H., Sobieski, J., Storaasli, O.O.: Compute as Fast as the Engineers Can Think! -Ultrafast Computing Team Final Report, NASA/TM-1999-209715, NASA Langley Research Center, Hampton, VA, September 1999
3. Hauser, J. et al.: A Pure Java Parallel Flow Solver. Proceedings, 37th AIAA Aerospace Science Meeting, Paper AIAA 99-0549 (1999)
4. Ginsberg, M.: An Overview of Supercomputing at General Motors Corporation. In: Ames,, K.R., Brenner, A.G. (eds): Frontiers of Supercomputing II: A National Reassessment. volume in the Los Alamos Series in Basic and Applied Sciences, University of California Press, Berkeley (1994) 359-371
5. Ginsberg, M.: Creating an Automotive Industry Benchmark Suite for Assessing the Effectiveness of High-Performance Computers. SAE Trans (J. Passenger Cars). 104, Part 2 (1995) 2048-2057

6. Ginsberg, M.: AUTOBENCH: A 21st Century Vision for an Automotive Computing Benchmark Suite. In: Sheh, M. (ed.): High Performance Computing in Automotive Design, Engineering, and Manufacturing, Cray Research, Inc., Eagan, MN (1997) 67-76

7. Ginsberg, M.: Supercomputer Help Auto Manufacturers Decrease Lead Time. In: Redelfs, A. (ed.): HPC Contributions to Society. Tabor Griffin Communications, San Diego, (November 1998) 74-79

8. Cray Inc.: Ford Installs Two More Cray SV1 Supercomputers, URL: http://www.cray.com/news/0008/ford.html (Aug. 3, 2000)

9. HPCWIRE: MSC LINUX Makes Supercomputing Accessible. Article 19168, HPCWIRE (January 5, 2001)

10 Sobieszczanski-Sobieski, J. et al.: Optimization of Car Body under Constraints of NVH and Crash. AIAA Paper 2000-1521 (2000)

11 Ginsberg, M.: Influences, Challenges, and Strategies for Automotive HPC Benchmarking and Performance Improvement. Par. Comp J. (1999) 1459-1476

12. Ginsberg, M.: Current and Future Status of HPC in the World Automotive Industry. In: Henderson, M.E., Anderson, C.R., Lyons, S.L. (eds.): Object Oriented Methods for Inter-Operable Scientific and Engineering Computing, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, (1999) 1-10

13. HPCWIRE: NEC & ESI Announce Virtual Engine Design System. Article 18918, HPCWIRE (Nov. 17, 2000)

14. HPCWIRE: Computer Simulation Allows “Virtual” Tire. Article 18715, HPCWIRE (Oct. 13, 2000)

15. HPCWIRE: Mazda Selects MTS for Design Improvement Project Article 18775, HPCWIRE (Oct. 20, 2000)

16. HPCWIRE: Ford Teams with UK’s Defence Evaluation Agency. Article 19064, HPCWIRE (Dec. 8, 2000)

17. HPCWIRE: Tele-Immersion May Change How We Communicate. Article 18761, HPCWIRE (Oct. 20, 2000)

18. Schulz, M., Ertl,, T., Reuding, T.: Crashing in Cyberspace - Evaluating Structural Behaviour of Car Bodies in a Virtual Environment. Proceedings, IEEE VRAIS’98, (1998) 160-166

19. HPCWIRE: Cray Announces Plans for Alpha Linux Supercluster Systems. Article 19347, HPCWIRE (February 2, 2001)

20. HPCWIRE: The Beowulf Factor in High Performance Computing. Article 19022, HPCWIRE (December 8, 2000)

21. HPCWIRE: Scientific Supercomputing in Germany. Article 18650, HPCWIRE (Oct. 6, 2000

22. HPCWIRE: SC User Group Notes Progress on Plan for Better Perf Tests. Article 60176, HPCWIRE (Nov. 7, 2000)

23. HPCWIRE: SGI and ESI Achieve Unprecedented Computing Power. Article 18871, HPCWIRE (Nov. 3, 2000)

24. HPCWIRE: Cray Inc. and Mcube Team Up on Auto Tests. Article 18839, HPCWIRE (Oct. 27, 2000)

25. HPCWIRE: Cray and MSC Software Team Up. Article 18733, HPCWIRE (Oct. 13, 2000)

26. Mavriplis, D.J.: Parallel Performance Investigations of an Unstructured Mesh Navier-Stokes Solver. ICASE Report No. 2000-13, ICASE, Hampton, VA (March 2000)

27. Foster, I., Kesselman, C. (eds.): The Grid Blueprint for a New Computing Infrastructure. Morgan-Kaufmann, San Francisco, CA (July 1998)