Conferences

What will it take to sell a massive number of massively parallel machines?

Howard Jay Siegel, Purdue University
Daniel W. Watson, Utah State University
John K. Antonio, Texas Tech University

IPPS '96:
10th International Parallel Processing Symposium
Apr. 15–19, 1996
Honolulu, Hawaii

Many parallel machine manufacturers have had to "reorganize." The marketplace seems to be leaning toward small-scale parallelism in the form of symmetric multiprocessors (SMPs), where typically two to 16 processors are interconnected with a common memory over a shared bus. A panel at the Computer Society-sponsored 10th International Parallel Processing Symposium (IPPS '96) examined the market for massively parallel machines—that is, systems with thousands of processors.

Hard questions

There are probably at most 100 or 200 massively parallel machines. This panel addressed the type of machine whose processors are tightly coupled through an interconnection network that allows all processors to insert messages simultaneously over some network topology (such as a mesh, hypercube, multistage interconnection network, or fat tree). The whole machine should be able to work on a common task.

The panelists were asked to address these questions:

- What (if any) application areas can make possible the sale of a massive number (thousands) of massively parallel machines in five to 10 years?
- If no application can do this in five to 10 years, then when will there be such applications and what will they be?
- Which users (customers) need massively parallel machines to perform these applications?
- What R&D should the parallel processing academic and industrial communities do to facilitate the use of parallel machines for these applications and users?

The panel's title put all of this together: "For a Massive Number of Massively Parallel Machines, What are the Target Applications, Who are the Target Users, and What New R&D is Needed to Hit the Target?"

The panel consisted of Howard Jay Siegel (panel chair), Mark Furtney, Paul Messina, Lionel M. Ni, Charles L. Seitz, Marc Snir, Robert Blumofe, and Steve Larson (see the sidebar for the panelists' biographies). Each panelist had approximately 10 minutes to present his views. The panelists and audience then held a general discussion.

Howard Jay Siegel

The panel chair introduced the topic and questions. He related the problems involved in the massively parallel processing field to the old saying, "When you are up to your butt in alligators, it is difficult to remind yourself that your initial objective was to drain the swamp." In this case, draining the swamp is analogous to making massively parallel machines mainstream computing devices. Many types of alligators (obstacles) attack parallel system builders and users, and this panel focused on the alligators that prevent the massive sales of such machines. H.J. indicated that to get companies to use massively parallel machines as standard computational tools, potential customers must be motivated to overcome financial and mental barriers (by mental barriers he meant ease of initial and continued use). He asked the panelists to focus on the applications that can provide this motivation.

Howard Jay Siegel is a professor and the coordinator of the Parallel Processing Laboratory in the School of Electrical and Computer Engineering at Purdue University. His research interests include heterogeneous computing, parallel algorithms, interconnection networks, and the PASM (Partitionable SIMD/MIMD) reconfigurable parallel computer system. He has coauthored over 200 technical papers, coedited six volumes, and written *Interconnection Networks for Large-Scale Parallel Processing*. He is a Fellow of the IEEE, was a coeditor-in-chief of the *Journal of Parallel and Distributed Computing*, and is on the editorial boards of the *IEEE Transactions on Parallel and Distributed Systems* and the *IEEE Transactions on Computers*. He received a BS degree in electrical engineering and in management from MIT, and his MA, MSE, and PhD in electrical engineering from Princeton University. He can be contacted at the Parallel Processing Laboratory, School of Electrical and Computer Engineering, Purdue Univ., West Lafayette, IN 47907-1285; hj@purdue.edu.

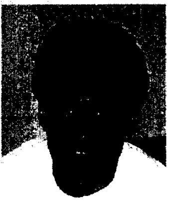
Mark Furtney is the senior director of technology for Cray Research. He has developed the multiprocessing software for the Cray-2, led the team that developed Cray's Autotasking software, led Cray's MPP software development, and led the software division. Mark has spoken extensively around the world on technology and the impact of supercomputing. He earned his Master's degree in nuclear engineering from MIT in 1970, and his PhD in computer science from the University of Virginia in 1983. He can be contacted at mf@cray.com.

Paul Messina is assistant vice president for scientific computing at Caltech, a faculty associate in scientific computing, and the director of Caltech's Center for Advanced Computing Research, and serves on the executive committee of the Center for Research on Parallel Computation. He also has a joint appointment at the Jet Propulsion Laboratory as the manager of high-performance computing. His interests focus on advanced computer architectures, especially their application to large-scale computations in science and engineering. He also is interested in high-speed networks and computer performance evaluation. He heads the Scalable I/O Initiative, a multi-institution, multi-agency project aimed at making I/O scalable for high-performance computing environments. He received his BA in mathematics in 1965 from the College of Wooster, and his MS in applied mathematics in 1967 and his PhD in mathematics in 1972, both from the University of Cincinnati. He can be contacted at messina@zephyr.ccsf.caltech.edu.

Lionel M. Ni, a professor at Michigan State University, is currently on leave as program director of the National Science Foundation's Microelectronic System's Architecture Program. He is on the editorial boards of *IEEE Transactions on Parallel and Distributed Systems* and *IEEE Transactions on Computers*. He has published extensively in the areas of parallel architectures, distributed systems, high-speed networks, VLSI design automation, operating systems, fault-tolerant computing, parallel compilers, and benchmarking techniques. He received his PhD in electrical engineering from Purdue University. In 1994, he was elected a Fellow of the IEEE for his contributions to parallel processing and distributed systems. He can be contacted at ni@cps.msu.edu.

Charles L. Seitz is the founder of Myricom, Inc. Before founding Myricom in 1994, he served for 16 years on the Caltech computer science faculty, where his research and teaching areas were computer architecture, concurrency, VLSI, and self-timed systems. He was elected to the National Academy of Engineering in 1992 "for pioneering contributions to the design of asynchronous and concurrent computing systems." He earned his BS, MS, and PhD, all in electrical engineering, from MIT. He can be contacted at charles@myri.com.

Marc Snir is a senior manager at the IBM T.J. Watson Research Center, where he leads research on scalable parallel systems. His group has been involved in the initial development of many of the technologies now used in the IBM SP2. Marc has published numerous papers on parallel algorithms, parallel architectures, and software for parallel computing. He recently participated in the HPF Forum and made significant contributions to the design of MPI. Marc is a Fellow of the IEEE and a member of the IBM Academy of Technology. He can be contacted at snir@watson.ibm.com.



Robert Blumofe is an assistant professor at the University of Texas at Austin. He started his research career working on computer graphics with Andy van Dam at Brown, and did his PhD work on algorithms and systems for parallel multithreaded computing with Charles Leiserson at MIT. As part of this dissertation work, he developed an algorithmic theory of multithreaded computation, and he designed and implemented Cilk, a multithreaded language and runtime system based on that theory. He has also developed an adaptive and fault-tolerant version of Cilk, called Cilk-NOW, which runs on networks of workstations. He is continuing his work on Cilk and Cilk-NOW. He received his BS in computer science from Brown University in 1988 and his PhD in computer science from MIT in 1995. He can be contacted at rdb@cs.utexas.edu.

Steve Larson is the director of business development for board-level products at Cambridge Parallel Processing. His interests include heterogeneous computing and hardware design, specializing in both low-cost and dense packaging. He received his MS in computer science from the University of Dayton. He can be contacted at larson@cppus.com.

H.J. concluded by asking the panel to refrain from using the expression "embarrassingly parallel." He asked the audience if we should really be embarrassed by tasks that map well onto parallel machines. He stated that such tasks did not embarrass him, and he requested that the panelists instead use the expression "pleasantly parallel."

Mark Furtney

Mark treated the audience to a unique perspective on parallel computing's past, present, and future. His insight is not surprising, given his broad background, which includes 12 years' experience in the nuclear power industry, experience leading several software projects at Cray, and his present position as Cray's Senior Director of Technology.

Mark pointed out that, "Given that computing is only about 50 years old, great strides have certainly been made." He went on to pose the question, "What other technological field can boast a trillion-times improvement in speed over a 50-year period?" Having established that computing is a young and rapidly changing technology, he then pointed out that "MPP [massively parallel processing] is the youngest part of an already young technology." He likened the popular use of serial computing today to the popular use of propeller planes several decades ago. "Back when propeller planes dominated the market, there were only a relatively few holdouts that believed in the great potential of using jet engine technology. Thus, during the time just before jet engines became the technology of choice, it was very difficult to predict that jets would ever actually take the place of propeller planes." He believes that this "pre-paradigm-shift" period experienced by the aircraft industry might be analogous to where we are now regarding the use of serial versus MPP machines.

To further support the argument that it might be only a matter of time before MPPs begin to really find their place in the market, Mark described the rapid advances in engineering and scientific computing that have accelerated demand for more computing power and ultimately for MPP systems. He showed a table that characterized Cray machines over the past 20 years. The table listed nine machines, from the Cray-1 in 1976 to the T3E in 1996, and each machine's clock speed and number of CPUs supported. Clock speeds have begun to stabilize in recent years, but the number of CPUs has steadily risen (from one CPU for the Cray-1 to 2,048 CPUs for the T3E). The table posed the question, "Can you spot a trend?"

The necessary software advances include programming environments that make it easy to add or remove software tasks from an ensemble while keeping the production system operational.

Mark believes that what we need now in the MPP field is one or two "eurekas." In this vein, he quoted Albert Einstein: "Imagination is more important than knowledge."

Paul Messina

Paul stated that "Sensor data analysis and storage—for example, for process monitoring and control—is a class of applications that will eventually result in massive sales of massively parallel machines, realistically within the next 10 years." He mentioned several organizations that have applications he believes will scale to require MPPs.

Electric power utilities are one target for parallel processing research, especially for monitoring generation and distribution equipment and optimizing this equipment's operation. According to Paul, "Collecting sensor data, comparing its dynamic behavior with simulations continually, and performing real-time control are good examples of functions that will be needed. Chemical processing plants will also need MPPs, for reasons similar to those given for the electric power utilities."

Medical centers typically use a complex and varied set of systems. Each system may have a different function, and might often need to interact with other systems. For example, Paul noted that processing the data from a patient's annual physical exam could require numerous processors. For instance, dozens of processors might be dedicated to MRI and CAT scan data processing; hundreds to maintaining databases of the processed data for patients; dozens to fusing data from several tests for a single patient and displaying it for physicians to examine; and a few to detect changes in X rays, blood tests, and so on. A typical large, urban medical center would easily need thousands of processors. While this application has a large distributed-computing flavor, interaction would occur between many of the heterogeneous clusters, particularly for the extensive databases of patient records including all the test data (for example, digital X rays and MRIs).

Paul's reasoning for the use of thousands of processors in these applications was that "First, the applications naturally involve thousands of real entities to monitor and simulate; second, using separate processors/nodes will simplify the programming task and maintenance; third, even modest improvements in the operation of the process can yield millions of dollars a day of savings; and fourth, processors will be inexpensive enough to warrant using thousands of them."

He further noted that the necessary software advances include programming environments that make it easy to add or remove software tasks from an ensemble while keeping the production system operational (no downtime required for software maintenance). For hardware, one useful ingredient would be community-standard communication fabrics that support nodes from

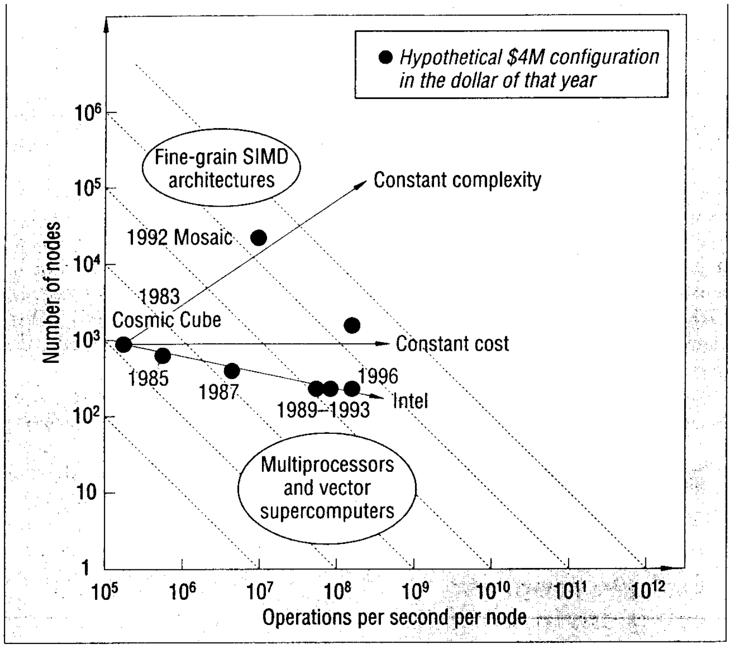


Figure 1. Where are the cost-effective MPPs? Charles Seitz thinks that long-term trends suggest that fine-grained MPPs will provide greater performance per cost.

many different vendors, yet are high-performance. (Some parts of the applications have real-time requirements, so the path from sensors to processor must be low-latency.)

Parallel processing does have a future, claims Paul, but government laboratories will not determine that future. Instead, it will be guided by the commercial interests that have both the need for increased computation and the funds to pay for it.

Lionel M. Ni

Lionel is currently a rotator at NSF for a year, which gives him somewhat of a funding agency's perspective. He reminded us that MPP technology is becoming more stable and mature. For him, the remaining question is, who can make use of it? He summarized his position by suggesting an alternate title for the panel: "Now that we are able to build MPPs, where is the market?"

He provided a brief overview of the current MPP "survivors," and concluded that the number of such systems is far from massive. According to Lionel, virtually all of the few existing MPPs (with processing elements that are 32-bit or 64-bit processors) are housed in research laboratories (for example, Sandia and Oak Ridge).

He then pointed out that the typical parallel system delivered today contains around 100 processors. In his estimation, this trend will continue: "Most machines to be built in the near future will have less than 200 nodes." To support this hypothesis, he noted that although a massive number (approximately 1,000) of IBM SPs have been delivered, the average number of processors on these installations is approximately 10.

Lionel offered some general guidelines regarding the question, "What are the target applications?" He stated that successful applications will employ single program, multiple data (SPMD) programming and require only local communications. Applications of this type exist in the areas of science and engineering, as well as in business.

His final point addressed the question, "Who are the target users?" He divided users into two broad categories: (a) national labs, government agencies, and universities; and (b) the commercial sector. For the first category, he painted a somewhat dim future, consisting of a limited market share (less than 5%) and uncertain funding. He believes the promise is in the second category. In the commercial sector, opportunities exist for generating a greater market, provided that high-performance computing can be viewed as a "revenue-making investment." He concluded that although the commercial sector will be an important part of parallel computing's future, a substantial shift toward that sector will not happen in the next 10 years. He encouraged MPP vendors to ponder, "Why are you building MPPs?"

Charles L. Seitz

Chuck was primarily concerned with performance and cost: "What are the killer applications that could create a market for MPPs with $N > 1{,}000$ (where N is the number of nodes in the MPP)?"

Grouping workstations together is one way to increase performance. Chuck noted, "We are in an era in which the processors in high-performance workstations are as fast as is cost-effective, so that applications beyond their level of performance are best approached by aggregating them." However, he did not propose that grouping workstations together will be the extent of the future of parallel computing. "Applications that require the aggregate performance of 1,000 or more high-end workstations are well-known, but are of limited commercial interest relative to the cost of such machines. If there is a volume market for massively parallel machines, it will be for systems of relatively finer granularity—a few rather than a few hundred chips per node—that provide remarkably high performance per unit cost, power, and size."

To illustrate his point, he introduced the graph shown in Figure 1. This log-log graph plots node performance on the abscissa, and the reciprocal of node cost on the ordinate, by showing how many nodes could be purchased for $4M. The straight, dashed, diagonal lines are hyperbolas of constant aggregate performance whose values can be read on the abscissa. For noncommercial machines, a node's price was estimated to be 3.5 times its production manufacturing cost. The price of commercial nodes varied over the product life, and the most attractive reported price was used to locate the dark circles. But even a factor of two in cost or performance will displace any of the dark circles on this graph only slightly, by about its own diameter.

From Chuck' perspective, fine-grain, $N > 1{,}000$ MPPs are attractive for embedded applications in industrial control, military systems, and communications. Furthermore, these types of systems excel at signal-processing, image-processing, and graphics computations. He observed that, although such machines might be used exclusively for one application, the principal research challenges are in programming and algorithms.

Marc Snir

Marc's group at the IBM T.J. Watson Research Center has been involved in the initial development of many of the technologies now used in the IBM SP2. According to Marc, "The technology exists today to build systems of essentially any size and exploit them on some applications. Suppose one would build now a petaflop machine from commodity components. This would require perhaps two million processors, would consume 200 megawatts, and would cost 10 gigadollars. Heroic, but less hard than building the first atomic bomb." He continued: "The same performance will cost in five years less than one gigadollar, and in 10 years will be achieved by, say, 20,000 processors, at a cost of less than 100 megadollars—that is, it will be affordable within the range of large supercomputers installed now, and within a small factor of the salary cost of the scientists and programmers using it." The problem, therefore, is to identify those problems that require supercomputing performance and that cannot wait five or 10 years.

From his point of view, grand challenge science does not meet these criteria because it can wait five or 10 years. Large engineering applications barely meet these criteria; the inaccuracies of current, approximate physical models limit the advantages accrued from more accurate computing. For example, in computational fluid dynamics, the equations used represent only an approximate model. Most important, Amdahl's law (as applied to the overall system) limits the amount of parallelism that can be effectively used. If it takes one hour before a submitted batch job begins execution, then a parallel execution that reduces compute time from one hour to only seconds is still not interesting. Achieving turnaround times of a few seconds on large VLSI simulations would significantly enhance productivity in VLSI design, but getting there would require much better system support for interactive parallelism.

The amount of compute power needed is not determined by some properties of a physical model; it is determined by the amount of compute power the competition wields.

Commercial applications might meet these criteria better, particularly for decision support. The amount of information on commercial transactions will increase significantly when commerce on the Internet becomes prevalent. In this context, decision-support systems could use data mining and other techniques to guide (on a store-by-store or chainwide basis) inventory control, pricing policies, store staffing levels as a function of time of day, product development, and so on. The amount of compute power needed is not determined by some properties of a physical model; it is determined by the amount of compute power the competition wields. In this area, a small edge in technology over the competition can translate into large profits. But, here too, better support for interactive parallelism is critical.

Marc concluded, "I don't believe that software is a major impediment to massive parallel computing. True, parallel codes are and will remain harder to develop than sequential codes: the system has more degrees of freedom. But the additional cost of porting a sequential application to a parallel system is usually much lower than the initial cost of developing the sequential application. Easing the task of parallel programming is an important goal, but there are no silver bullets there."

Robert Blumofe

Bobby suggested a populist approach. "When everyone has access to massively parallel computers that can be programmed with high-level parallel programming languages, then imaginative programmers will develop new parallel applications that have not yet been dreamed of, and ultimately, everyone who uses computers will use massively parallel computers." Central to his vision is that massively parallel computers will more closely resemble networks of workstations or even internets. "Often referred to as metacomputers, such systems will not be surrounded by queuing software and be available only to an elite few. Rather, everyone will have immediate access to metacomputers, and furthermore, anyone who owns a computer will be able to contribute his or her resources to a metacomputer." He noted that ultimately, metacomputing cannot succeed without automated policies that dynamically manage metacomputer resources with guaranteed efficiency.

Bobby's position was that metacomputer programming models must abstract all metacomputer resources and free the programmer from the arduous task of resource management. The complexity and dynamic nature of metacomputer resources requires that applications be coded using only virtual resources, so that runtime systems can automatically and dynamically map these virtual resources to physical resources. He continued: "For example, in the Cilk parallel multithreaded programming language and runtime system, threads abstract processors, and 'dag-consistent' [directed acyclic graph-consistent.] virtual memory abstracts physical memory. The Cilk runtime system automatically and dynamically schedules the execution of threads on the processors and moves data in virtual memory to the physical memories where it is needed. For thread scheduling, Cilk uses a provably efficient algorithm based on the technique of 'work stealing.'" Efforts are underway to extend Cilk's programming abstractions to parallel I/O.

Bobby stated that metacomputer job-scheduling policies must dynamically respond to the changing availability of resources in the metacomputer and the changing demand for resources in the applications. In addition, the owners will not contribute their resources to a metacomputer unless the job-scheduling policies preserve each owner's sovereignty. One approach being pursued in his ongoing Cilk project uses the rate of work stealing to determine automatically each application's instantaneous resource requirements so that users need not specify their application's resource needs. He pointed out that fault tolerance is critical to metacomputing: "Metacomputer runtime systems must automatically detect and recover from hardware and software faults, while applications themselves remain 'fault oblivious.'"

He summarized: "Who are the users? Massively parallel metacomputers will be available to and used by everyone. What are the applications? Metacomputers will run a set of applications that includes traditional parallel applications in science, engineering, and business, as well as new applications that we have not yet dreamed of. How do we get there? In addition to the traditional research focus on mechanisms, research must focus on the development and automation of policies that will dynamically manage metacomputer resources with guaranteed performance."

Steve Larson

As the director of business development at Cambridge Parallel Processing (CPP), Steve knows firsthand the importance of attracting a wide range of users, including those from the commercial sector, into the MPP fold. Steve gave the audience an idea of how tough the current market environment is by responding to Lionel's concluding question, "Why are you building MPPs?" Steve's tongue-in-cheek response: "I ask myself that question every day!"

Research in parallel processing should focus on reducing the cost of parallel systems, making parallel systems easier to use, or developing applications that can dramatically increase the profits for the end user.

Steve started with a brief overview of CPP, which he stated is the largest current manufacturer of SIMD computers. CPP supports a range of application domains, "from computer room systems to embedded systems," and has delivered over 200 systems.

He then explained why he believes CPP is still in business, given that "the roads to success are littered with the corpses of the parallel vendors that did not make it." He attributes CPP's success primarily to its basic business strategy: "Produce small, low-cost, 1,024-processor distributed-array processors (DAPs); leave the megabuck hardware to the 'Big Boys'." Other factors that have contributed to CPP's success include embracing the open systems concept, moving toward Khoros and C++ and away from proprietary software, and making a commitment to "run the company like a business, not an R&D shop." He believes that MPP vendors must "move away from what is fun and move toward what will keep them in business. This means creating solutions to problems that the computer marketplace will purchase in order to ensure a level of operation that will keep them in business."

Steve then responded to the question, "What are the target applications?" In his view, MPPs (SIMD computers particularly) can provide effective solutions for high-end image processing, text search, data mining, and C^4I (Command, Control, Communications, Computers, and Intelligence) decision support.

Steve also offered some insight into the question, "How do we get to massive sales?" First, he believes that as the price goes down, sales will go up. It is also important to continue to provide portable, open software. He then stated that cheap "off-the-shelf I/O" is important to support heterogeneous and embedded systems. Finally, he reiterated the importance of transforming MPPs into real solutions for the commercial marketplace.

Steve then suggested the types of R&D that the parallel processing community should be conducting. These suggestions included developing more software environments like Khoros that aim at hiding the underlying parallel hardware, increasing pin counts in VLSI technology, reducing power consumption (important in certain embedded applications), developing standardized parallel libraries, and working on real applications that benefit from heterogeneous computing.

Steve's conclusion expressed a positive future for MPPs. He warned, though, that the research community must step out of the sandbox and into the real world.

Panel summary

If parallel processing as a field is to survive its reorganization phase, researchers and parallel machine manufacturers can no longer ignore "unscientific" customers such as investment, entertainment, and manufacturing industries. Even industries that are immersed in science, such as medical services, must be specifically targeted. Market forces alone are often not enough to motivate these industries to seek parallel processing solutions to their problems. Instead, complete parallel solutions must be provided to them, so that they can see the revenue-generating attractiveness of this technology for their specific application. According to the panel, these industries have the potential to profit from parallel processing:

- Chemical manufacturing companies.
- Investment firms.
- Entertainment industries.
- Virtual reality firms.
- Electric power utilities.
- Retail industries.
- Communications industries.
- Defense industries.
- Medical industries.

To make this a reality will require R&D in many areas. Research in parallel processing should focus on reducing the cost of parallel systems, making parallel systems easier to use, or developing applications that can dramatically increase the profits for the end user. The panel listed these research areas:

- Decision support.
- Sensor monitoring.
- Data mining.
- Software environments that hide underlying hardware.
- Interactive parallel computing environments.
- Dynamic resource-management policies.
- Inexpensive processors customized for fine-grain parallelism.
- Standard interprocessor communications.
- Embedded and heterogeneous computing systems.
- Standard intermachine communications.

For parallel processing researchers and manufacturers, the call to action is simple and clear: if we expect to sell massively parallel machines, we have to prove to the buyers that they can profit from their use. The users and markets are there; it is up to the MPP community to produce complete solutions if we want massive sales.

EPILOGUE

IPPS '97, to be held in Geneva, Switzerland, on April 1-5, 1997, will feature a related panel titled "Building Parallel Machines: Experiences and Lessons Learned." For more details, access the IPPS '97 Web site at http://cuiwww.unige.ch/~ipps97.

ACKNOWLEDGMENTS

We thank the panelists, Viktor Prasanna, and Janet M. Siegel for their comments on this report.

Howard Jay Siegel's biography appears in the sidebar on p. 64.

Daniel W. Watson is an assistant professor in the Department of Computer Science at Utah State University. He has coauthored numerous conference papers, journal articles, and book chapters on mixed-mode parallelism and heterogeneous computing. In heterogeneous computing, his recent work includes the static matching of program segments to machines and dynamic scheduling approaches for data-dependent tasks. Dan is the vice general chair for the 1997 Heterogeneous Computing Workshop (HCW '97). He received his BS in electrical engineering at Tennessee Tech University in 1985 and his MSEE and PhD from the School of Electrical Engineering at Purdue University in 1989 and 1993. He is a member of the IEEE Computer Society; is a member of the Gamma Beta Phi, Tau Beta Pi, and Eta Kappa Nu honorary societies; and has received the Ronald G. Harber Award from HKN's Beta Chapter. He can be contacted at the Dept. of Computer Science, Utah State Univ., Logan, UT 84322-4205; watson@cs.usu.edu.

John K. Antonio is an associate professor of computer science at Texas Tech University. His research interests include the development of automated techniques for the effective use of parallel and distributed processing systems, tools and methodologies for software development for parallel processing systems, and practical applications of control and optimization theory. He has coauthored over 50 publications in these areas. For the past three years, he has organized the industrial track and the commercial exhibits of the IPPS. He received his BS, MS, and PhD, all in electrical engineering, from Texas A&M University, College Station, in 1984, 1986, and 1989. He can be contacted at the Dept. of Computer Science, Texas Tech Univ., Lubbock, TX 79409-3104; antonio@cs.ttu.edu.