The Roots of Beowulf

James R. Fischer
NASA Goddard Space Flight Center
Code 606
Greenbelt, MD 20771 USA
1-301-286-3465
james.r.fischer@nasa.gov

ABSTRACT

The first Beowulf Linux commodity cluster was constructed at NASA's Goddard Space Flight Center in 1994 and its origins are a part of the folklore of high-end computing. In fact, the conditions within Goddard that brought the idea into being were shaped by rich historical roots, strategic pressures brought on by the ramp up of the Federal High-Performance Computing and Communications Program, growth of the open software movement, microprocessor performance trends, and the vision of key technologists. This multifaceted story is told here for the first time from the point of view of NASA project management.

Categories and Subject Descriptors

C.1.4 [Processor Architectures]: Parallel Architectures --- distributed architectures; C.5.0 [Computer System Implementation]: General; K.2 [Computing Milieux]: History of Computing

General Terms

Management

Keywords

Beowulf, cluster, HPCC, NASA, Goddard, Sterling, Becker, Dorband, Nickolls, Massively Parallel Processor, MPP, MasPar, SIMD, Blitzen

1. INTRODUCTION

Looking back to the origins of the Beowulf cluster computing movement in 1993, it is well known that the driving force was NASA's stated need for a gigaflops workstation costing less than $50,000. That is true, but the creative conversations that brought the necessary ideas together were precipitated by a more basic need—to share software.

2. THE PRE-BEOWULF COMPUTING WORLD

A flashback to the pre-Beowulf computing world paints a picture of limitations. The perspective is NASA centric, Goddard Space

Flight Center specifically, but the experience was universal. It is only 20 years ago, but the impediments facing those who needed high-end computing are somewhat incomprehensible today if you were not there and may be best forgotten if you were.

2.1 Proprietary Stove Piped Systems

Every system that we could buy ran proprietary system software on proprietary hardware.

Competing vendors' operating environments were, in many cases, extremely incompatible—changing to a different vendor could be traumatic.

A facility's only recourse for software enhancement and problem resolution was through the original vendor.

2.2 Poor Price/Performance

In 1990, "a high-end workstation can be purchased for the equivalent of a full-time employee." [1]

In 1992, some facilities used clusters of workstations to offload supercomputers and harvest wasted cycles; at Goddard we salivated at this idea but had few high-end workstations to use.

The very high and rising cost of each new generation of supercomputer development forced vendors to pass along those costs to customers. The vendors could inflate their prices because they were only competing with other vendors doing the same thing.

2.3 Numerous Performance Choke Points

In 1991, the Intel Touchstone Delta at Caltech was the top machine in the world, but compilation had to be done on Sun workstations using proprietary system software that only ran on Suns.

In 1993, Connection Machine Fortran compile and link performance averaged about 10 lines per second on the host; performance was similar for the MasPar used at Goddard. All development had to be done on the host machine; vendors were not really solving this problem (maybe they could sell you two host machines).

2.4 Instability

In 1993, operational metrics recorded by NASA Ames Research Center for their Intel Paragon reported "Reboots Weekly Average" (typically 15–30) and "Mean Time to Incidents" (typically 4–10 hours). Each reboot forced all running jobs to be restarted, and the reboot for some systems might take 30 minutes. Since the bigger MIMD machines were usually one-off's, the OS developers had to take the entire system away from users into

stand-alone mode to debug the system software (i.e., to increase its stability and reduce the reboots).

My notes from a meeting in early 1993 record a manager's statement that putting a second user on their KSR-1 (Kendall Square Research) system caused crashes, and then another manager immediately states the same situation on their Intel Paragon—this situation was not unusual.

2.5 Diversity of Architectures and Programming Methods

In the early 1980s the Japanese government began pursuit of its 5th Generation Computing Program, which spooked those in the U.S. who saw the strategic importance of high-end computing dominance, resulting in money pouring into computer architecture research from the National Science Foundation (NSF), Department of Defense, NASA, and other U.S. agencies. By the late 1980s every computer science department in the U.S. seemed to be building a novel machine along with a novel language to program it. Some of these approaches were commercialized.

By 1991, as the High-Performance Computing and Communications (HPCC) Program was ramping up and ready to acquire large parallel testbeds, we needed benchmarks for the vendors to run. The kernel benchmarks of the day were for vector processors, and existing user applications would not run on specific vendor parallel systems until they were properly restructured—so application benchmarking in most cases could not be used.

2.6 Tedious and Time Consuming Acquisition Processes

Within NASA, procurement of prototype parallel architectures for use as HPCC testbeds was subject to the same Federal Acquisition Regulations (FAR) as operational machines—the process would typically take a year and could not select a machine that was not available for benchmarking, making it impossible to bring in experimental machines through standard procurement. Some other agencies were not limited in this way, and the Defense Advanced Research Projects Agency (DARPA) helped many institutions quickly acquire testbeds using their contracts, but they were forced to cease doing this in 1993.

3. MAKING PARALLEL COMPUTING MORE ACCESSIBLE

Goddard began exploring parallel computing in the early 1970s as Earth orbit satellites (e.g., Landsat) were being envisioned as surveying the entire surface of the planet every couple of weeks at 60-meter resolution and producing an immense, continuous stream of data that would easily swamp computing systems of that era. One candidate solution was parallel computing, and NASA invested in a variety of optical approaches starting in the late 1960s, initially with the intent to fly systems in space along side the sensors. By the mid 1970s, the evolution of integrated circuit technology changed the emphasis to digital, electronic, and ground-based systems.

By 1977, prototyping at Goddard produced the specifications for a sixteen thousand processor prototype that was competed full-and-open, resulting in award of a $4.6 million development contract to Goodyear Aerospace Corp. and delivery of the Massively Parallel Processor (MPP) to Goddard in 1983 meeting or exceeding every specification [2]. Beginning in 1985 a nationally selected working group of scientific investigators began use of the MPP and in 1986 reported to NASA Headquarters that the system was appropriate for many of their diverse applications [3]. Its demonstrated performance gained national attention.

The MPP inspired the initial Connection Machine architecture and was commercialized through collaboration between Digital Equipment Corp. and MasPar Computer Corp. Goddard continued its architecture research through research awards to universities and then to the Microelectronics Center of North Carolina, which produced the Blitzen chip in 1989, one of the first million-transistor chips. Blitzen contained 128 processors, each more capable than that of the MPP, and the vision was to package 128 Blitzen chips into a low cost and physically small sixteen thousand processor MPP workstation.

It needs to be pointed out that the MPP architecture ran a single program in a single control unit that broadcast a single instruction to all sixteen thousand processors each machine cycle. This single-instruction-stream-multiple-data-stream (SIMD) architecture has inherent advantages over competing approaches through lower complexity and better power efficiency but requires large problems to keep its many processors busy. SIMD was the favored architecture at Goddard because satellite image data provided just such large problems.

Fifteen years of research with the MPP and its descendants, and the other parallel testbeds that we had access to, had shown us that the right hardware was mighty important but the software environment was equally important, and it was largely missing in our prototypes. I was convinced that the software environment would advance only when parallel systems became cheap enough that they could be purchased in large numbers, thereby drawing the interest of many more software developers.

Goddard's first Project Plan for HPCC/Earth and Space Sciences, written in early 1991 [4], included a task for "development of a prototype scalable workstation and leading a mass buy procurement for scalable workstations for all the HPCC projects. The performance goal for the scalable workstation is one gigaflops (sustained) by FY1995 in the $50,000 to $100,000 price range. The same software development environment will support the workstation and the scalable teraflops system. The same programs will run on the workstation and the scalable teraflops system but only with different rates of execution." It is safe to say that up until the end of 1993 we were putting little effort into this task because we did not know how to achieve its goal.

4. WORKSHOP FINDING: A CLEAR NEED EXISTS FOR BETTER PARALLEL SYSTEM SOFTWARE

When the Federal HPCC initiative began to ramp up in 1991, Goddard and Ames were given prime roles, and I became manager at Goddard of the HPCC Earth and Space Sciences (ESS) Project designed to apply HPCC technologies to the kinds of science that Goddard did. HPCC was a focused technology program that had a 5-year planned lifetime (later extended) allowing us to take a long-term view of the work. Lee Holcomb was the HPCC Program Manager at Headquarters, and Paul Smith was his deputy.

The approach laid out by the High-Performance Computing Act of 1991 to rapidly mature scalable parallel computing systems was to have them stress tested by sophisticated research teams as

they worked to make progress on their Grand Challenge applications in science and engineering. These Grand Challenge teams were to pioneer parallel computational technology and then share it with the world (or maybe just the U.S. part of the world)

through "software clearinghouses" that "would allow researchers to deposit voluntarily their research software at the clearinghouse where it would be catalogued and made available to others." [5]

NASA's role in the Federal HPCC Program was significant. In addition to conducting Grand Challenge applications development on increasingly capable testbeds it was to:

- coordinate applications and system software development, and
- define and implement the HPCC Software Exchange (the software clearinghouse)

across the entire Federal Program.

One of the earliest national events related to this coordination role took place in May 1993 when nine Federal Agencies jointly sponsored the "Workshop and Conference on Grand Challenges Applications and Software Technology" held in Pittsburgh. It brought together, for the first time, Grand Challenge Investigator Teams from many Federal Agencies. The role of these teams at the workshop was to identify their software technology needs. Paul Smith chaired the organizing committee on behalf of NASA.

The workshop's final report [6] is an impressive snapshot of the issues of the day. The voices of those in the trenches come through clearly in the nine working group reports; the tensions they express were all too familiar to us, often having to do with keeping our talented leading-edge HPCC technology people happy as they tried to move their immature technologies into Grand Challenge application groups who needed to publish to survive. The primary finding of the workshop reads: "A clear need exists for better parallel debugging tools, tools for multidisciplinary applications, performance-monitoring tools, and language support to allow users to write programs at a higher level than currently possible. The cause for the poor software support is the fact that the Grand Challenge grants currently focus on the output of the applications rather than on the software to achieve that output. More effective mechanisms are needed for exchanging information on tool availability and accessibility."

5. DOWNSELECTING ARCHITECTURES

The diversity of architectures and programming methods present in the HPCC community, described earlier, gave strength to the program by exposing our scientists and technologists to valuable out-of-the-box thinking, but the stated goal of the Program was to achieve teraflops sustainable performance on important Grand Challenge codes by 1997, and our Investigator teams' highest performing codes in 1993 were achieving just a few gigaflops. One could say "it's just research" and lower the goal, but that was not how the Federal Program saw it, nor NASA Headquarters, and they had signed us up to very aggressive milestones with clear metrics and success criteria.

We needed to "downselect" from the dozen or so testbed architectures available to us in the form of small systems to one (or two) based on Grand Challenge needs and then supersize that with our generous but finite testbed budget to allow the Grand Challenge Teams to achieve performance milestones for us. Downselection required deep technical insight and reliable intelligence regarding what each vendor's next-generation system would look like, while occasionally listening to the vendors' marketing staff. We also tried to watch vendors' capitalization

because many were venture funded and could go out of business overnight.

From the time I was appointed ESS Project Manager in January 1991, I was looking for someone who could analyze the algorithmic needs of our Grand Challenge investigators and how the various machine architectures supported those needs. It was in June 1992 at a NASA HPCC Working Group meeting at NASA Lewis Research Center in Cleveland (now the Glenn Research Center) that I started talking to Thomas Sterling about this work. Thomas had been providing expert technical analysis to the HPCC Program Office at NASA Headquarters but was feeling what he called "colleague deprivation" and was very happy to come to Goddard and become the Evaluation Coordinator for ESS. This was in September 1992. Thomas was a natural in this role because of his strong background in machine architectures with the National Security Agency's Supercomputing Research Center and before that with Harris Corp. and at MIT. In the summer of 1993 Thomas became NASA's representative to and organizer of the Joint NSF-NASA Initiative on Evaluation (JNNIE), which compared 22 applications on 19 types of computer systems (1993–1995) [7].

6. RESPONDING TO THE PITTSBURGH WORKSHOP FINDINGS

Paul Smith quickly took action to address the Pittsburgh workshop's findings by asking for ideas from within the NASA HPCC Projects and making the Program's reserve money available as an incentive. We were asked to prepare augmentation proposals for verbal presentation on November 9, 1993.

My handwritten notes from a planning meeting held at Goddard in preparation, probably involving Thomas Sterling and John Dorband, show that we discussed a new idea—I had written "S/W Environments Integration … CAN Software Integrator." We were wrestling with what to do with software that would be produced/submitted from our dozens of teams, say for evaluation, or for system software development, or for software sharing, e.g., Investigator Team reuse. What location would the software come to? How would it be shared? All the vendor-provided systems were transient with a lifetime of 3 years or less. It was going to be impractical to port all this incoming software every 3 years; we needed an architecture and environment that would be around for a long time. This challenge was the precipitator of the Beowulf concept—it was the need for a common/neutral/persistent environment for "software environments integration." The driving force was for software sharing, and the gigaflops workstation embodiment quickly followed.

I remember well the day that Thomas Sterling and John Dorband came to my office and told me about the Linux PC cluster idea that Thomas and Don Becker had conceptualized. Sterling and Becker had been colleagues at the Supercomputing Research Center, and Don still worked there. He was a well-known provider of Ethernet device drivers for Linux, and his drivers were a key part of the plan to couple lots of PCs with network links. John was my deputy project manager for system software, so Thomas had gotten him onboard first. As they described the plan, I could see that the Linux cluster would be amazingly inexpensive. I trusted John's judgment that the proposed demonstration was low risk. I had never heard of Linux before that meeting, it was just 2 years old. When they left my office I was onboard too and had authorized Thomas to recruit Don to come to Goddard.

On November 9, 1993, the NASA HPCC Technical Committee met in the Universities Space Research Association (USRA) Board Room in Washington, DC, chaired by Paul Smith. It was an all-day meeting with a packed agenda. Augmentation requests were not mentioned on the agenda but were embedded under a morning item "Report/discussion of FY 94 systems software tasks at each Center." John Dorband presented for Goddard, and his charts included this augmentation request "Title: Acceleration of parallel operating system development by facilitating extensive external collaboration; Level-2: Prototype public domain operating system for 16-processor workstation under Linux based PVM; Lead: J.Dorband/GSFC; Funding: $100K." In the few minutes given for discussion the proposal was well received. I remember that Paul Messina was very supportive, which might have made the difference in subsequent deliberations. I do not believe that Thomas Sterling was present at that morning discussion but that he arrived in the afternoon.

On November 26, 1993, I sent 14 ESS tasks to Jim Pool at Caltech, who was helping Paul Smith coordinate system software work. Six were marked "Augmentation request," and one of the six is noteworthy: "Title: Extension of the Linux operating system into the distributed domain. Level-3: Prototype public domain operating system for 16-processor workstation under Linux based PVM (FY96). Lead: J.Dorband/GSFC. Funding: $100K. Abstract: Cheap high-performance computing systems are virtually non-existent. This is not due to lack of cheap hardware, but due to the complexity and difficulty of developing a small, tightly coupled, efficient, and reliable operating environment. The most practical and cost-effective way of accomplishing this would be to find the least expensive hardware platform that supports a stable inexpensive operating system that could be easily modified to support tightly coupled copies of the hardware. Contrary to intuition this is not impossible. PC-compatible hardware is cheap and supports the publicly available Unix operating system called Linux. The source code for Linux is also publicly available. This effort will modify Linux so as to support multiple tightly coupled PCs. This will then be the platform for testing highly-distributed, I/O intensive, and GUI applications developed under the architecture-independent programming environment."

I just love the abstract—it is totally objective and completely unassuming. Thomas would select the name Beowulf later. [8]

On December 16, 1993, I received word that the augmentation was granted; it came from Bruce Blaylock at Ames, who also was helping Paul Smith coordinate system software work. This gave us Headquarters visibility and buy in. The augmentation plus gigaflops workstation money already budgeted, allowed us to hire Don, purchase the parts for the first Beowulf cluster, Wiglaf, and assemble a small support team. Don was brought in through Goddard's Center of Excellence in Space Data and Information Sciences (CESDIS).

7. TECHNOLOGY TRENDS

At the same November 9, 1993 meeting, Bruce Blaylock distributed the "Draft HPCCP Software Development Plan," [9] prepared by the NAS (then Numerical Aerodynamic Simulation, now NASA Advanced Supercomputing) Division at Ames.

Bruce was the hard-nosed operations manager at NAS and was usually a bit closer to reality than most others in the room. The plan starts off "The overall objective of this activity is to identify, define, and provide the system software resources that will enable the successful integration of a highly parallel computer into a balanced production environment. The challenge is not to simply install a highly parallel system as a high-speed processor, but to

create a computing environment enabling application migration to span many architectural options." Bruce documented well the specter he was facing as the NAS operations manager, looked to for path-forward-support by users who had productively used generations of Cray vector processors and were now being presented with a changing spectrum of parallel machines based on different programming paradigms and with incomplete stables of feature implementations. In Figure 1 he lists his machine options in the 1993–1995 time frame.
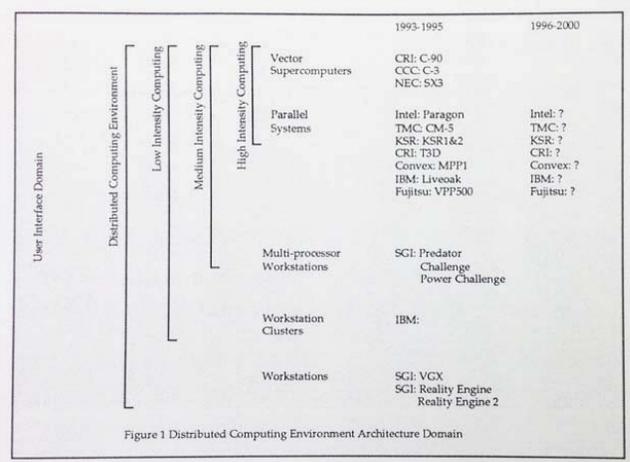

Figure 1

Augmentation proposals were supposed to fit into Bruce's plan, although it contains no mention of Linux. It does say, "While the HPCCP program is not intending to embark upon an independent operating systems development effort, it is essential that the HPCCP create a clear enough vision of what it would develop to meet the needs of the high-performance computing community that the vendors engaged in the effort can be persuaded to produce the needed product. Current vendor efforts already demonstrate that without such guidance the resulting operating systems are very large, very inefficient, and highly unreliable. Operating systems are complex entities whose development should not be undertaken lightly. However, their development can not be assumed to be proceeding rationally just because a computer vendor is involved."

The presence of this analysis at the same meeting where the Beowulf concept is first mentioned provides a striking contrast between the dinosaurs and the specification for the mammal. Figure 1 has all question marks in the 1996–2000 column—in fact, by that time range, most of the listed vendors had either gone out of business or were packaging some version of a Linux cluster.

Figure 2 [10] plots, as of 1992, the performance gains being made by the processor chip vendors compared to a single processor in the Cray product line; the crossover point was approaching in performance and had already occurred in price/performance. In short order these trends became even more pronounced. In 1994,

the first Beowulf cluster, Wiglaf, consisted of 16 100MHz 80486DX4-based PCs. One year later Hrothgar was built from 16 100MHz Pentium Pros and was about three times faster.
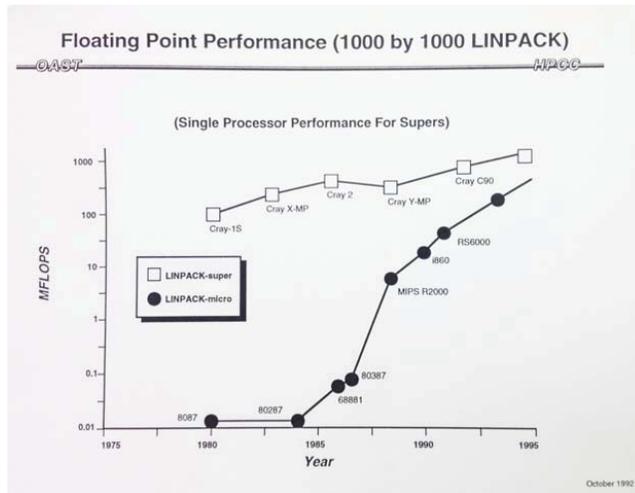

Figure 2

8. IMPACT

After Thomas Sterling, Don Becker, John Dorband, and technician Dan Jacob built the first Beowulf cluster in 1994, I exposed the ESS Grand Challenge Investigators to the Beowulf team's advances at each ESS Science Team meeting. Within a couple years several of those teams were running Beowulfs at their home institutions.

The ESS scalable gigaflops workstation milestone was met in the fall of 1996:

– In September, Mike Warren/Los Alamos National Laboratory (LANL) assembled a 16-processor Beowulf system from Pentium processors and ran a Salmon/Warren treecode simulation on 10 million particles, achieving a sustained floating-point performance of around 1.1 gigaflops for a cost of under $60,000.

– In October, the same problem was ported to the Caltech/Jet Propulsion Laboratory (JPL) 16-processor Beowulf and achieved a sustained performance of 1.26 gigaflops for a cost of approximately $50,000.

– In November, the LANL and Caltech systems were brought to the Supercomputing '96 (SC96) conference in Pittsburgh, and joined together into a 32-processor Beowulf (worth around $100,000) on the exhibit floor and ran Warren/Salmon tree code problems at around 2.2 gigaflops.

By 1997, Beowulf was getting broad attention, and "How to Build a Beowulf" tutorials were being held. After Mike Warren/LANL and John Salmon/Caltech won the Gordon Bell Prize for price/performance at SC97 on a Beowulf system, we took on the additional role of amazed spectators as the concept spread rapidly into many vendor products. By the time MIT Press published Thomas Sterling and collaborators' book *How to Build a Beowulf* [11] in 1999, we were buying them commercially.

The limitations that I listed at the beginning were largely resolved. The operating system was non-proprietary. The cost of nodes became the lowest possible because they came from the mass PC market. Application software could be developed on cheap deskside clusters. Systems could be customized with more communication links, storage nodes, and host processors as needed. The system people could have their own dedicated platforms to develop on. The Linux kernel was amazingly stable, and clusters might run for months between reboots.

By the year 2000 this movement had brought on what Thomas Sterling terms the "Pax MPI," a period when MPI was "the" programming model and innovative architectures either fit under it or were sidelined. Pax MPI simplified benchmarking enormously.

In recent years, Beowulf-inspired commodity cluster systems have grown to represent greater than 80% of the world's Top 500 supercomputers and are now operated by high-performance computing centers of all sizes at universities, industrial facilities, and government labs around the world.

9. AN ELEGANT SOLUTION

This story is shaped around a rich and totally positive irony—I brought Thomas Sterling into ESS as Evaluation Coordinator to analyze Grand Challenge applications and look for the way forward through the maze of available architectures. He did this and then in his spare time brought about the Beowulf revolution, which removed the architectural maze and provided the way forward—an elegant solution indeed!

10. EPILOGUE

Compared to other highly visible aspects of the HPCC/ESS Project, the Beowulf activity did not cost much. It only lasted around 4 years and involved three to four people at any one time to get it going. Once initiated, the work propagated through peer-to-peer relationships in the open software movement.

Don Becker remained at the leading edge of Beowulf maturation, leaving CESDIS in 1998 to form Scyld Software where he innovated methods that made clusters easier to manage. [12]

Other limitations of the pre-Beowulf era were overcome. For one, the lack of effective benchmarking methods for parallel machines in 1991 led to development of "paper and pencil" methods such as David Bailey's NAS Parallel Benchmarks, initially released in 1992 [13], giving the vendor leeway to code the benchmarks any way they want.

Simplification of 1993's tedious and time-consuming acquisition process required help from other parts of NASA because architecture research could not fix that, but two things did:

1. NASA's invention of the cooperative agreement (neither a grant nor a contract) especially for technology development, allowing significant interaction between the government and awardees and supporting performance-based milestone payments. ESS's Round-2 Grand Challenge Investigations and testbed were selected in 1996 through a cooperative agreement notice.

2. NASA's implementation of a series of SEWP (Scientific and Engineering Workstation Procurement) government-wide acquisition contracts, the first of which was awarded in 1993; the SEWP acronym evolved to Solutions for Enterprise Wide Procurement, and by 2010 the SEWP contract was offering 1.3 million products from 3,000 manufacturers and serving 70 Federal Agencies.

Earlier in this paper, I referred to computer architectures in terms of dinosaurs and mammals, but in total humility because these roles can reverse over time. A case in point is the

commercialization of the Goodyear MPP's SIMD architecture in the form of the MasPar MP-1 and MP-2. These products had found good fits in some important markets based on custom designed processor chips that were advancing along a Moore's Law curve. When MasPar suddenly exited the hardware business

in 1996 (Connection Machine manufacturer Thinking Machines having done the same in 1994) that architecture looked to be a dinosaur.

In 2010, I caught up with John Nickolls, who had been MasPar's Vice-President for Engineering. Nickolls was heading the architecture group at NVIDIA, where he had been for several years, and he said, "we are doing here what we were doing at MasPar, but the price point has gone way down and the performance has gone way up." Nearly 15 years later, the SIMD dinosaur had become a mammal. In effect, with NVIDIA GPU chips accelerating millions of laptops, our goal of the Blitzen-accelerated workstation has been achieved in spades and has become part of the infrastructure. [14]

11. ACKNOWLEDGMENTS

12. REFERENCES

[1] K.G. Stevens, 1990. "Budget Rationale" in chart set "Computational Aeroscience Project." Comprehensive HPCI Review, July 12–13, 1990.

[2] Jerry L. Potter and Dennis B. Gannon, editors, 1985. *The Massively Parallel Processor.* MIT Press, Cambridge, MA.

[3] James R. Fischer, Chester Grosch, Michael McAnulty, John O'Donnell, and Owen Storey, 1987. *Report From the MPP Working Group to the NASA Associate Administrator for Space Science and Applications.* NASA Technical Memorandum 87819, November 1987.

[4] *Project Plan for the Earth and Space Science Project of the NASA High-Performance Computing and Communications Program (NASA HPCC/ESS),* Jim Fischer (Project Manager), Version 3.0: May 3, 1991, NASA Goddard Space Flight Center.

[5] *High-Performance Computing Act of 1990*, Report of the Senate Committee on Commerce, Science, and Transportation on S. 1067, July 23 (legislative day, July 10), 1990, Calendar No. 710, page 22.

[6] *Workshop and Conference on Grand Challenges Applications and Software Technology*, May 4–7, 1993, Pittsburgh, PA, GCW-059.

[7] Wayne Pfeiffer, San Diego Supercomputer Center; Steven Hotovy, Cornell Theory Center; Nicholas A. Nystrom, Pittsburgh Supercomputing Center; David Rudy, NASA Langley Research Center; Thomas Sterling, Universities Space Research Association; Mark Straka, National Center for Supercomputing Applications, 1995. *JNNIE: The Joint NSF-NASA Initiative on Evaluation.* San Diego Supercomputer Center, July 1995, GA-A22123.

[8] Thomas Sterling, 2003. "Beowulf Breakthroughs: The Path to Commodity Supercomputing." *Linux Magazine* <http://www.linux-mag.com/id/1378/>, June 15, 2003.

[9] *HPCCP Software Development Plan* – DRAFT, prepared by NAS Systems Division, NASA Ames Research Center, November 6, 1993, Draft 2, pages 3 and 5.

[10] Lee B. Holcomb, 1992. "High-Performance Computing and Communications Program." Presentation to NASA Advisory Council, October 28, 1992.

[11] Thomas L. Sterling, John Salmon, Donald J. Becker, Daniel F. Savarese, 1999. *How to Build a Beowulf: A Guide to the Implementation and Application of PC Clusters.* The MIT Press, Cambridge, MA.

[12] Donald Becker's keynote, LWN Feature Article, LWN.net, March 26, 2001, <http://lwn.net/2001/features/Singapore/becker.php3>.

[13] David Bailey, John Barton, Thomas Lasinski, Horst Simon, editors, 1991. *The NAS Parallel Benchmarks.* Report RNR-91-002 Revision 2, August 22, 1991.

[14] David A. Patterson and John L. Hennessy, 2009. "Graphics and Computing GPUs," Appendix A, pages A-2 to A-77, in *Computer Organization and Design, The Hardware/Software Interface*, Fourth Edition. Morgan Kaufmann Publishers, Burlington, MA.