David Bryan Carpenter: Resume

Northeast Parallel Architectures Center
111 College Place
Syracuse University
Syracuse, NY 13244-4100
USA

Phone: 315 443-5068
Email: dbc@npac.syr.edu

December 5, 1999

Personal Details

Born: 15th January, 1958, Birmingham, UK

Nationality: British

Skills

Parallel computing, computational science, parallel languages, parallel compilers and library support, runtime libraries for distributed arrays, message passing libraries (MPI), object-oriented programming (C++ and Java), scientific programming (Fortran), functional programming and logic programming, formal methods, UNIX system programming, theoretical physics.

Education

BSc., Physics, from Westfield College, London. First Class Honours. Degree awarded 1979.

PhD., Theoretical High Energy Physics, also from Westfield. Supervisor: Prof. E. Leader. Thesis: *Lattice and Large N Studies of Two-dimensional QCD*, submitted 1983. Degree awarded 1984.

Positions Held

1996-present. Research Scientist, Northeast Parallel Architectures Center, Syracuse, USA.

1994-95. Programmer, High Performance Computing Centre, Southampton, UK.

1989-93. Research Fellow in Department of Electronics and Computer Science (jointly with Department of Physics through 1991-92), Southampton, UK.

1989. Employed at "Transputer Technology Solutions", Southampton, UK.

1985-1988. Research Fellow in Computational Physics Group, Southampton, UK.

1985. Held Royal Society Overseas Fellowship at DESY, Hamburg, Germany.

1983-1984. Research Fellow in Theoretical Physics Department, Edinburgh University.

Career Summary

Physics

My doctoral research in theoretical physics involved applying newly-developed computational techniques for handling lattice fermions to a low-dimensional field theory [1]. For the first two years of my postdoctoral career I worked as a physicist, engaged in more or less theoretical work on lattice gauge theories and field theory [2, 3, 4]. At Edinburgh I worked in David Wallace's group and was exposed to one of the early massively parallel computers—the ICL DAP.

In 1985 I moved to Southampton, a recruit to Tony Hey's newly formed computational physics group. Initially working as an application programmer, I spent several years implementing physics codes on parallel computers—especially Transputer systems. In close collaboration with *inmos* we wrote some of the first scientific programs to run on Transputers. Between 1989 and 1992 I collaborated with Prof Chris Sachrajda and a group of European physicists on first-principles calculations of the parameters of heavy quark systems [12, 13, 14]. Earlier collaborations with physicists involved simulation of solid state [5, 9] and pure gauge theories [10]. During this time, however, my principal interests moved increasingly towards computational issues.

The Supernode Project (1985-88)

I have been particularly interested in design and development of communication libraries to support data-parallel computation (and the associated language issues). This interest started during the ESPRIT *Supernode* project, which developed a reconfigurable multi-Transputer supercomputer. I worked on programming paradigms for exploiting Transputer networks in simulation of physical systems [6]. The deliverables for that project included *occam* libraries and harnesses to facilitate distribution of grid-oriented problems [7].

The PUMA Project (1989-91)

During the subsequent *PUMA* (Parallel Universal Message-passing Architectures) project I designed and prototyped a more advanced collective communication library, *ACL* [15]. The PUMA project was strongly influenced by Valiant's ideas on Bulk Synchronous Parallel (BSP) computation, which were popular at that time. The choice of some of the collective communication primitives in ACL was influenced by the BSP model. Other features of the ACL model were high-level primitives for dynamically spawning multi-dimensional process arrays, and the use of abstract data types to describe process arrays and distributed data. The subsequent work on Adlib was influenced by this latter feature of ACL. ACL was initially implemented in *occam*.

In the PUMA project John Merlin worked on the *ADAPT* system for semi-automatic parallelisation of Fortran programs with distributed data. ADAPT was one of many precursors to High Performance Fortran (HPF), and John was later involved in the HPF Forum. During PUMA he proposed a collective communication library called ADLIB which the ADAPT translation tool would target. The ADAPT system was not completed during the PUMA project, and ADLIB as specified at that time remained unimplemented.

Computer Science (1992)

After the PUMA project finished I returned to physics collaboration for a year. Also at that time I became interested in theoretical (CSP-based) models of the dynamic channel communication operations used in the implemention of ACL.

In late 1992 I worked briefly with Hugh Glaser on the FAST (Functional programming on Arrays of Transputers) project. We implemented a few scientific codes in the lazy functional language Haskell, with a view, ultimately, to automatic parallelisation [16].

HPF and Adlib (1993-95)

In 1993 HPF became fashionable. John Merlin and I started a collaboration to implement a prototype HPF translation system, recycling components of the ADAPT system to build an HPF-to-Fortran translator, with technology from ACL to implement the run-time library. Some time into this project we obtained formal funding to continue from the UK JISC and EPSRC councils.

In the early stages of project I hoped we could formally specify our translation scheme for HPF. This proved impractical, but parts of a translation scheme were prototyped in the logic language Prolog.

The initial specification of the new run-time library, *Adlib*, was completed in 1994. Adlib was defined as a C++ class library. Unlike ACL, which was designed primarily to support an *ad hoc* set of common applications and algorithms, Adlib was designed specifically for translation of

HPF. It supported the full HPF distributed data model, providing high-level gather/scatter and other data remapping operations on HPF arrays. It also provided a set of arithmetic and reduction operations suitable for implementing the Fortran 90 array intrinsics on distributed data.

An initial implementation of Adlib, built on MPI, was completed in 1995 [17]. Adlib was implemented in C++. A restricted Fortran interface, suitable as a target for the modified ADAPT translator, was also provided.

The associated subset HPF translation system, *shpf* [18], was installed at several beta test sites during 1995. It is now maintained by John Merlin, who has moved to the European Center for Parallel Computing in Vienna.

PCRC (1996-1998)

In 1996 I moved to the Northeast Parallel Architectures Center (NPAC) in Syracuse, New York. I was employed in the ongoing Parallel Compiler Runtime Consortium project (PCRC). This was a DARPA-funded project involving several leading research groups working on parallel runtime systems and compilers. NPAC was prime contractor.

Through the influence of Xiaoming Li, who was visiting NPAC at that time, the NPAC group was collaborating with universities in China to develop a new HPF translator. I prototyped a runtime system for the emerging compiler on top of Adlib. When this proved successful [19], it was decided to that a version of Adlib would be adopted as one of NPAC's contributions to the runtime software to be delivered in PCRC. The library was redesigned and rewritten as a standalone piece of software: the NPAC PCRC Runtime Kernel [20].

Under direction of Geoffrey Fox, the PCRC group was managed by Don Leskiw and later by Xiaoming Li. In mid 1997, Xiaoming left and I took over as a local project leader. At that time the NPAC PCRC group included myself, two postdoctoral researchers from China, and two graduate students. I was responsible for producing reports to DARPA for the PCRC project as a whole, as well as leading group meetings and directing the other researchers in the NPAC group.

The PCRC project was extended by six months, and formally finished in July 1998. The collected works of PCRC are available through my Web pages at http://www.npac.syr.edu/projects/pcrc/.

HPJava

An unforseen influence on the development of PCRC was the emergence of Java. In 1996 the participants of the project issued a draft white paper on the implications of Java for High Performance Computing. Subsequently Syracuse organized a series of workshops on the theme of Java for Computational Science and Engineering. This led to the formation of the Java Grande Forum—a group aiming to promote Java standards for communication and compute intensive applications.

The acronym "HPJava" had been coined in the PCRC white paper. NPAC adopted the name for its Java environment for SPMD computing. This environment is envisaged to contain Java bindings for distributed-array based libraries, general MIMD programming libraries, and a preprocessor for an extended dialect of Java.

The first useable component of the HPJava environment is our recently released *mpiJava* binding of MPI [30]. At the instigation of Geoffrey Fox we had been experimenting with Java interfaces to native MPI since 1996 [21, 22]. I specified a full Java API for MPI in late '97. It was implemented by graduate students under my supervision during 1998. This API influenced the draft API specification produced for Java Grande, in collaboration with several groups outside NPAC, later that year [29].

HPspmd (1998-present)

By 1997 those of us remaining in the PCRC group at NPAC were doubtful about the prospects for producing a full HPF compiler with useful performance. On the other hand we had a runtime library.

Me and Guansong Zhang, with input from Geoffrey Fox and Yuhong Wen, suggested adopting a language model motivated in part by one of the C++ interfaces to Adlib. This interface provided classes to describe distributed arrays and a few "control construct" macros to simplify traversal of those arrays, but required the programmer to explicitly specify calls to collective communication functions (rather than relying on the compiler to insert them automatically). The pure C++ class library implementation of this model was very inefficient, but the suggestion was that by shifting the distributed arrays and distributed control constructs into the syntax of a programming language, these inefficiencies could be removed. Because communication and computation placement are specified explicitly by the programmer, this kind of language would be much easier to implement efficiently than HPF [25, 26].

The ultimate success of this language model has yet to be proven, but in April of 1998 I wrote a proposal with Geoffrey Fox to the National Science Foundation [27], to explore programming models based on advanced language bindings of runtime libraries. The proposal was approved and funded at a level of $115k per annum over three years. The project formally started in December 1998.

The initial implementation of our HPspmd language extensions will be in the HPJava environment [23, 24]. The extensions will be handled by a source to source preprocessor, under development by Guansong Zhang and students. The choice of Java as base language is justified on the basis that it is a simple language with useful support for complicated data types, but we are also considering Fortran bindings of the same model. Xiaoming Li, now back in China, is interested in implementing a translator for the Fortran-based HPspmd system in his group.

The HPspmd language extensions are supposed to be useful for interfaces to any distributed-array based libraries—not only the PCRC runtime kernel. We are collaborating with the developers of the Global Arrays (GA) toolkit [28], and eventually hope to incorporate an interface to GA and other libraries into our environment.

References

[1] D.B. Carpenter, *Two Dimensional Lattice QCD in Quenched Approximation.* Nucl. Phys. B228 (1983) 365.

[2] D.B. Carpenter and C.F. Baillie, *Free Fermion Propagators and Lattice Finite-Size Effects.* Nucl. Phys. B260 (1985) 103.

[3] D.B. Carpenter, *Blocking Transformations for Lattice Fermions.* DESY Preprint 85-064.

[4] S.J. Hands and D.B. Carpenter, *Lattice Sigma Model and Fermion Doubling.* Nucl. Phys. B266 (1986) 285.

[5] C.R. Askew, D.B. Carpenter, J.T. Chalker, A.J.G. Hey, D.A. Nicole and D.J. Pritchard, *Simulation of Statistical Mechanical Systems on Transputer Arrays.* Computer Physics Communications 42 (1986) 21.

[6] C.R. Askew, D.B. Carpenter, J.T. Chalker, A.J.G. Hey, M. Moore, D.A. Nicole and D.J. Pritchard, *Monte Carlo Simulation on Transputer Arrays.* Parallel Computing 6 (1988) 247.

[7] D.B. Carpenter and J.H. Merlin, *Gauge Field Theory Simulation on Transputer Arrays.* ESPRIT project P1085, Workpackage 7, 3rd deliverable report (1989).

[8] J.R. Allwright and D.B. Carpenter, *A Distributed Implementation of Simulated Annealing for the Travelling Salesman Problem.* Parallel Computing 10 (1989) 335.

[9] D.B. Carpenter and J.T. Chalker, *The Phase Diagram of a Generalised XY Model.* J. Phys: Condens. Matter I (1989) 4907.

[10] B. Carpenter, C. Michael and M.J. Teper, *O^+ and 2^+ Glueball Masses from Large Lattices in SU(2) Lattice Gauge Theory.* Phys. Lett. B198 (1987) 511.

[11] C.R. Allton, D.B. Carpenter, V. Lubicz, L. Maiani, G. Martinelli and C.T. Sachrajda, *Decay Constants of Heavy Mesons.* Nuclear Physics B (Proc. Suppl.) 20 (1991) 504.

[12] M. Bochicchio, G. Martinelli, C.R. Allton, C.T. Sachrajda, D.B. Carpenter, *Heavy-quark spectroscopy on the lattice* Nuclear Physics B 372 (1992) 403.

[13] A. Abada, C.R. Allton, Ph. Boucard, D.B. Carpenter, M. Crisafulli, J. Galand, S. Güsken, G. Martinelli, O. Pène, C.T. Sachrajda, R. Sarno, K. Schilling and R. Sommer, *Meson Spectroscopy and Decay Constants with Wilson Fermions at* $\beta = 6.4$ Nucl.Phys. B376 (1992) 172.

[14] A. Abada, C.R. Allton, Ph. Boucard, D.B. Carpenter, M. Crisafulli, S. Güsken, P. Hernadez, V. Lubicz, G. Martinelli, O. Pène, C.T. Sachrajda, K. Schilling, G. Siegert and R. Sommer, *Semi-leptonic Decays of Heavy Flavours on a Fine-grained Lattice* Nucl.Phys. B416 (1994) 675.

[15] D.B. Carpenter, *Array Communication Library: Introduction.* ESPRIT project PUMA, Workpackage 5.1, report (1992).

[16] D.B. Carpenter and H. Glaser, *Some Lattice-Based Scientific Problems, expressed in Haskell*, Journal of Functional Programming, 6(3):419-443, May 1996.

[17] D.B. Carpenter, Adlib*: A Distributed Array Library to Support HPF Translation.* Presented at 5th Workshop on Compilers for Parallel Computers, Malaga (1995).

[18] John Merlin, Bryan Carpenter, and Tony Hey. shpf*: a subset High Performance Fortran compilation system.* Fortran Journal, pages 2–6, March 1996.

[19] Guansong Zhang, Bryan Carpenter, Geoffrey Fox, Xiaoming Li, Xinying Li, and Yuhong Wen. *PCRC-based HPF compilation.* In *10th International Workshop on Languages and Compilers for Parallel Computing*, volume 1366 of Lecture Notes in Computer Science. Springer, 1997.

[20] Bryan Carpenter, Guansong Zhang, and Yuhong Wen. *NPAC PCRC runtime kernel definition.* Technical Report CRPC-TR97726, Center for Research on Parallel Computation, 1997.

[21] Bryan Carpenter, Yuh-Jye Chang, Geoffrey Fox, Donald Leskiw, and Xiaoming Li. *Experiments with HPJava.* Concurrency: Practice and Experience, 9(6):633, 1997.

[22] Bryan Carpenter, Yuh-Jye Chang, Geoffrey Fox, and Xiaoming Li. *Java as a language for scientific parallel programming.* In *10th International Workshop on Languages and Compilers for Parallel Computing*, volume 1366 of Lecture Notes in Computer Science. Springer, 1997.

[23] Bryan Carpenter, Guansong Zhang, Geoffrey Fox, Xiaoming Li, Xinying Li, and Yuhong Wen. *Towards a Java environment for SPMD programming.* In David Pritchard and Jeff Reeve, editors, *4th International Europar Conference*, volume 1470 of Lecture Notes in Computer Science. Springer, 1998.

[24] Bryan Carpenter, Guansong Zhang, Geoffrey Fox, Xinying Li, and Yuhong Wen. *HPJava: Data parallel extensions to Java.* To appear in Concurrency: Practice and Experience.

[25] Bryan Carpenter, Geoffrey Fox, Donald Leskiw, Xinying Li, Yuhong Wen, and Guansong Zhang. *Language bindings for a data-parallel runtime.* In Michael Gerndt and Hermann Hellwagner, editors, *3rd International Workshop on High-Level Parallel Programming Models and Supportive Environments.* IEEE Computer Society Press, 1998.

[26] Guansong Zhang, Bryan Carpenter, Geoffrey Fox, Xinying Li, and Yuhong Wen. *The HPspmd model and its Java Binding.* In Rajkumar Buyya, editor, *High Performance Cluster Computing*, Prentice Hall. To appear.

[27] Bryan Carpenter and Geoffrey Fox. *HPspmd: Data Parallel SPMD Programming Models from Fortran to Java.* Proposal to the National Science Foundation, April 1998.

[28] Jarek Nieplocha and Bryan Carpenter. *ARMCI: A Portable Remote Memory Copy Library for Distributed Array Libraries and Compiler Run-time Systems.* To be presented at *3rd Workshop on Runtime Systems for Parallel Programming*, San Juan, Puerto Rico, April 12, 1999.

[29] Bryan Carpenter, Vladimir Getov, Glenn Judd, Tony Skjellum and Geoffrey Fox. *MPI for Java—Position Document and Draft API Specification.* Java Grande Forum Technical Report JGF-TR-03. November 1998.

[30] Mark Baker, Bryan Carpenter, Geoffrey Fox, Sung Hoon Ko, and Xinying Li. *mpiJava: A Java interface to MPI.* In *First UK Workshop on Java for High Performance Network Computing, Europar '98*, September 1998.