

High performance computing and computational aerodynamics in the UK

D. R. Emerson, A. J. Sunderland and M. Ashworth

Computational Science and Engineering Department
CCLRC Daresbury Laboratory
Warrington, UK

K. J. Badcock

CFD Laboratory, FST Group
Department of Engineering
University of Liverpool
Liverpool, UK

ABSTRACT

The establishment of the UK Applied Aerodynamics Consortium in 2004 brought together many of the UK's leading research groups to tackle challenging aerodynamic problems on the national computing facility, HPCx. This paper provides a brief history of some early pioneers of numerical simulation and highlights some key contributions to development in parallel processing that laid the foundations for today's researchers. The transition from vector to massively parallel processing is discussed from a UK viewpoint along with technological barriers that could have a significant impact on future systems. Solutions to these barriers are already being sought and the paper discussed some of the novel technologies that may be deployed in the future. In its short history, the consortium has made substantial progress and this is briefly discussed with several highlights that illustrate the scientific output. Although a number of challenges are identified, particularly with respect to developing a comprehensive visualisation capability, the consortium is well placed to build upon its initial success.

1.0 INTRODUCTION

Computers now play a key role in all scientific disciplines, from gaining fundamental understanding through to highly applied research. Consequently, computational science and engineering

(CSE) is now widely accepted as an independent discipline that complements theory and experiment. The stimulus for the creation of CSE lies in the rapid advancement of computer performance, as predicted in 1965 by Intel co-founder, Gordon Moore⁽¹⁾. This prophetic article led to the establishment of Moore's law, which is often interpreted as a doubling in computer performance every 18 months. However, in his original article, Moore was asked to predict future developments in integrated electronics and the semiconductor industry, subjects that were still quite young and gaining acceptance, and stated that the number of components (e.g. a transistor, resistor, diode or capacitor) would double every 12 months. In 1975, Moore⁽²⁾ reassessed the available data and predicted a slowing down of this trend, largely due to an expected drop in packaging efficiency, and projected a doubling every two years from 1980 onwards.

Regardless of the exact exponential scaling for computer performance, a major beneficiary from these rapid developments has been computational science. In this paper, the focus is on computational aerodynamics and particularly the application of computational fluid dynamics (CFD) to applied aerodynamic problems that require extensive use of the high-end computing (HEC). Parallel computing now plays a significant role in delivering increased understanding of complex flow phenomena associated with the simulation of geometrically realistic aircraft. The fact that computers would play an increasingly important role was recognised by Chapman⁽³⁾, who gave

an estimate of the computer power needed to perform various types of calculations. In this important outlook on computational aerodynamics, Chapman projected that the late 1990s could see turbulent simulations using around 10^9 or 10^{10} grid points. This anticipated growth in capability relied on the existing trends in memory and computer power continuing alongside developments in numerical methods. Similar papers^(4,5) also concluded that computational aerodynamics was going to have a major impact on research and development and would eventually become an integral part of the design process. There is now a clear recognition that CFD has created a paradigm shift in vehicle design, as stated by Johnson *et al*⁽⁶⁾, where the last 30 years has seen Boeing increase the number of CFD runs by two orders of magnitude to around 20,000.

This paper will describe some notable contributions to developments recognising that numerical computation was going to be a major factor in scientific advances. Although there have been many contributions to the field of parallel processing, a few key examples will be highlighted that have made a significant impact. The paper will then present an overview of the UK Applied Aerodynamics Consortium (UKAAC), which currently involves 11 UK institutions and four commercial partners, and the aerodynamic challenges currently being performed on HPCx[†], the UK's national flagship supercomputing facility. Finally, the UK is preparing for a new national computing facility that is scheduled to be available in 2007. However, as we move into an era when thousands of processors will be required to tackle the next generation of scientific problems, we discuss some of the fundamental issues involved in being able to scale CFD codes to this size of system and the challenge of extracting good floating-point performance.

2.0 COMPUTING: PAST AND PRESENT

One of the early pioneers of using numerical techniques to solve scientific problems was Lewis Fry Richardson. In a fascinating 1911 paper⁽⁷⁾, he developed a finite difference method to determine stresses in a masonry dam. The reason for this work was simple: analytical methods were quite limited in application, particularly when the partial differential equations were applied to complex shapes. Although computers were not available to Richardson, the paper clearly highlights that 'parallel processing' was employed through the use of boys to perform the calculations. One of the typical calculations was for Laplace's equation

$$\nabla^2\phi = \frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} = 0 \quad \dots (1)$$

The finite difference approximation to Equation (1) on a uniform mesh was

$$\frac{\phi_{i+1,j} + \phi_{i-1,j} + \phi_{i,j+1} + \phi_{i,j-1} - 4\phi_{i,j}}{h^2} = 0 \quad \dots (2)$$

The boys were paid 'piece rates' for each correct answer they produced. The rate paid was $n/18$ 'old pence' per coordinate point, where n is the number of significant digits which was typically 3. One of the quickest boys averaged 2,000 correct answers per week. This would have earned the boy around £1.40 for his week's work.

The solution of Equation (2) involves three additions, one subtraction, two multiplications, and one divide operation. For simplicity, we can assume that this solution equates to approximately ten numerical operations on a typical computer (a divide operation usually involves many more operations than an addition, subtraction, or multiplication which are often counted as a unit operation). The boy would have performed about 20,000 operations and, assuming a 40 hour week, his floating point rate would be 140

mflop/s i.e. 140×10^{-3} floating point operations per second. We can compare this computation to what can be done on HPCx, the UK's national facility, which currently has a peak performance of 6 Gflop/s per processor. If we assume that we can extract 10% of the peak performance, the 20,000 operations would be completed in just 33 μ s. Moreover, today's cost of each HPCx central processing unit (CPU) hour is 83-85p and the total cost for the boy's weekly calculation would be the remarkably small sum of just 0.0000077p.

Limitations in analytical solutions were also a motivating factor for work reported by Kawaguti⁽⁸⁾ who used finite difference techniques to solve the Navier-Stokes equations for flow past a circular cylinder at a Reynolds number of 40. It was already known that solutions derived by Stokes and Oseen could only be applied at low Reynolds numbers. However, experiments had shown that once a critical Reynolds number is exceeded, the flow becomes unstable with the onset of a von Kármán vortex street. Although a number of computers had been built (e.g. EDSAC at Cambridge, ENIAC in USA), they were primarily for military applications like ballistics. For Kawaguti, and many other pioneers of numerical solutions, the calculations involved substantial dedication as the following quote by Kawaguti illustrates: "The numerical integration in this study took about one year and a half with twenty working hours every week, with a considerable amount of labour and endurance." The total drag predicted by Kawaguti was 1.6177, which compares very favourably with more recent work which gives the drag to be around 1.5.

2.1 Overcoming Amdahl's law

In 1967, Gene Amdahl wrote a seminal paper⁽⁹⁾ that was to have a major impact on the perception of parallel computing. In essence, the argument stated that parallel performance will be limited by the serial part of the code, regardless of how many processors are available. This can be shown as follows: if we assume that all operations are carried out at the same computational speed and that a percentage, f , is carried out in parallel on p processors while the remaining work, $(1-f)$, has to be carried out on a single processor, then it can be shown⁽¹⁰⁾ that the speed-up, S_p , is given by:

$$S_p = \frac{p}{f + (1-f)p} \quad \dots (3)$$

Equation (3) is known as Amdahl's law and it is clear that f must be very close to unity to achieve good scalability. If an infinitely large number of processors were available, then Equation (3) is bounded by $(1-f)^{-1}$ i.e. if only 1% of the program remained serial then the speed-up will be limited to 100, regardless of how many processors are available to solve the problem. For many years, the very pessimistic projection offered by Amdahl's law was a considerable obstacle to the acceptance that parallel computing was the way forward. One major difficulty with Amdahl's law is that it implicitly assumes that the proportion of parallelised code is independent of the problem size. Neglecting cache-based effects, if a fixed-size problem is run on an increasing number of processors, then the speed-up bound would take effect. However, in practice, the problem size is usually scaled with the number of processors. In 1988 Gustafson⁽¹¹⁾ proposed that it might be more realistic to consider the run time and not the problem size. As shown in Emerson⁽¹⁰⁾, this allows the speed-up to be written as follows

$$S_p = p - (p-1)(1-f) \quad \dots (4)$$

In the work performed by Gustafson on a 1024 Ncube computer, the speed-up varied from 1016 for a flux-corrected transport code to 1021 for a conjugate gradient solver.

[†] www.hpcx.ac.uk

2.2 Partitioning of unstructured grids

At the start of the 1990s, when parallel computing was making an uneasy transition from esoteric academic research into the mainstream, there was renewed interest in developing efficient methods to partition unstructured grids. Many scientific problems have a data structure that can be described by a 'graph' e.g. structural mechanics. To map this type of problem onto a distributed memory machine was a formidable task.

The graph partitioning problem had been considered previously for designing VLSI networks⁽¹²⁾ but the interest now was on complex geometric shapes. The basic problem essentially consists of balancing the computational load whilst minimising the communication cost between processors i.e. given p processors, divide the graph G into roughly the same number of vertices, V , representing the work load, whilst minimising the number of edges, E , that are cut, which represents the communication. Unfortunately, this type of optimisation falls into a category of problems known as NP-complete⁽¹³⁾ (non-deterministic polynomial-time). Even for the simplest case with uniform weighting, it is unlikely that an efficient algorithm exists and it is therefore necessary to use heuristic methods.

For distributed memory machines, developing an efficient partition depends on both the problem and the target architecture. A number of different approaches have been developed e.g. simulated annealing, greedy algorithm⁽¹⁴⁾, Kernigan and Lin algorithm⁽¹⁵⁾, recursive coordinate bisection, recursive graph bisection. For complex geometries, such as a multi-element airfoil, the partitioning could lead to disconnected domains⁽¹⁶⁾, which is very undesirable on a distributed memory machine, or very long thin sections that would incur a significant communication overhead. However, a breakthrough came with the development of the recursive spectral bisection approach⁽¹⁷⁾ which produced good quality partitions but at extra computational cost. This approach stimulated research into making the algorithm more computationally efficient and led to the development of a multi-level technique⁽¹⁸⁾. Progress was being made very rapidly and by the end of 1993, the algorithms were becoming efficient and robust and the first practical package, CHACO⁽¹⁹⁾, was made available to the general user community. Other packages were to follow, such as JOSTLE⁽²⁰⁾, but the de facto standard used today is METIS⁽²¹⁾ or its latter variant, ParMETIS. At the start of the 1990s, the typical partitioning problem consisted of 30,000 vertices and 45,000 edges. In 2005, Mavriplis *et al.*⁽²²⁾ used 72m grid points (315m cells) to model an aircraft wing and nacelle configuration on 2008 processors of NASA's Columbia supercomputer. The partitioning was performed with METIS.

3.0 HIGH-END COMPUTING IN THE UK: A BRIEF HISTORICAL PERSPECTIVE (1980-2006)

In the 1980s, high-end computing provision in the UK was dominated by vector systems. The aging CDC 7600s at the University of Manchester's Regional Computing Centre (UMRCC) were replaced in 1982 by a CDC Cyber 205 to supplement the existing Cray X-MP/8 at the University of London Computer Centre (ULCC). The next national system was a Cray Y-MP/8 installed at Rutherford Appleton Laboratory with a peak performance of 2.7Gflop/s. The Cyber 205 at UMRCC was replaced in 1988 by an Amdahl VPI200 vector processor and ULCC's Cray was succeeded in 1991 by a Convex C3860. Following a review that considered future scientific research and the potential of parallel computing, the break with vector systems was made in 1994 and a 256-processor Cray T3D was installed at Edinburgh Parallel Computing Centre (EPCC). This was when most user groups in the UK made the transition from serial code with vectorisable loops, possibly with some parallelism through Cray microtasking, to explicit parallel coding using the newly adopted message passing standard MPI. To assist this process the funding for

the hardware was supplemented by specialist support through the High Performance Computing Initiative, with centres of expertise established at Daresbury Laboratory, EPCC and Southampton University. In 1997 the high-end capacity was supplemented by extending an existing Cray T3E-900 at EPCC for national users and in 1998, in a new tranche of funding, a Cray T3E-1200 was installed at the University of Manchester.

This signalled the start of a strategy for HEC provision in the UK in which procurements would be started every three years with the aim of awarding a six-year contract to a consortium who would provide hardware, accommodation and management, and CSE support. There would thus always be two concurrently active services, which would be both complementary and competing. The six-year contract attempts to give some continuity of tenure to experienced staff, whereas the rapid rate of development of high-end systems is allowed for by requiring three deliveries of hardware, at years zero, two and four, with the aim of doubling performance at each stage (not quite keeping pace with Moore's law).

The 1988 service at the University of Manchester was the first of these six-year contracts with systems operated by Computing Services for Academic Research (CSAR). As Cray Inc. was taken over by SGI, the T3E was eventually supplemented by an SGI Origin 3800 system, and finally replaced in 2003 by a 256-processor SGI Altix 3700 which, with further upgrades, reached 512 processors in 2004.

The next contract was awarded in 2002 to HPCx, a consortium comprising CCLRC Daresbury Laboratory, EPCC and IBM. The start of service in November 2002 featured an IBM p690 cluster with the SP Switch, which at 3.2Teraflop/s on the Linpack benchmark, was the first single system available to UK academics to have a sustained performance in excess of 1 Teraflop/s. The upgrade in 2004 increased the clock speed from 1.3GHz to 1.7GHz with p690+ frames but more importantly introduced IBM's High Performance Switch, and increased the sustained performance to 6.2 Teraflop/s. A switch to POWER5 technology in 2005 leads the way to a further doubling of performance to around 12 Linpack Teraflop/s in late 2006.

The High End Computing Group at EPSRC, as the managing agent for high-end computing in the UK, is leading the next supercomputing procurement. From April 2007, the High End Computing Technology Resource (HECToR) will be the UK's next generation high-end service. As with HPCx, the service will be upgraded in three phases during the lifetime of the service, with performance doubling from an initial 50-100 Teraflop/s at the start of service to 100-200Teraflop/s and finally to 200-400Teraflop/s. This time there are separate procurements for the three different components of the ca. £100m provision i.e.

1. Hardware Technology (currently a shortlist of vendors)
2. Computational Science and Engineering Support (tender issued April 2006)
3. Accommodation and Management (tender issued April 2006)

The size of this supercomputer will see researchers having to scale their codes to efficiently use 1,000s of processors. This represents a major challenge for code developers, performance specialists, and developing appropriate capabilities for post-processing the significant amounts of data that will be generated using HECToR.

4.0 THE APPLIED AERODYNAMICS CONSORTIUM

The UK Applied Aerodynamics Consortium was initiated for three years in June, 2004 by a successful application to the Engineering and Physical Sciences Research Council (EPSRC) for time on HPCx. The consortium currently has 11 academic members (Liverpool (lead), Bristol, Cambridge, Loughborough, Surrey, Imperial, Sheffield, Manchester, Cranfield, Swansea and Daresbury)

Table 1
Organisation of themes

Theme	Lead	Partners
1. High fidelity simulation of helicopter interactions	Liverpool	Bristol, Cranfield, Manchester, Sheffield
2. Simulation of a free flying flexible aircraft	Liverpool	Bristol
3. Simulation of vertical landing aircraft	Loughborough	Cambridge, Surrey
4. Aeroelasticity studies for aero-engine core-compressors	Imperial College	
5. Simulation of internal air system	Surrey	
6. Simulation of store separation	Swansea	

and four industrial partners (BAE Systems, Rolls-Royce, Westlands and Fluent). The consortium was allocated 1.7m processor hours. In addition, a bi-annual series of meetings aims to promote the exchange of experience between consortium partners and a two-day conference was organised by the consortium in April, 2006 which had 73 delegates from ten industrial and 17 academic organisations.

The consortium's technical programme is organised into six themes, four of which were prompted by the rotor aeromechanics DARP[‡] and the Partnership for Unsteady Methods in Aerodynamics (PUMA) DARP. The themes are summarised in Table 1. The actual computer usage has exceeded original plans and at least nine of the academic partners have been regular users and have presented significant results at the regular consortium meetings. The codes used by the consortium have proved mature and robust, contributing to the successful exploitation of the supercomputer.

The unifying theme of the consortium is aerodynamics associated with realistic aircraft systems. This poses a number of challenges. First, the geometries which are being tackled are generally complex. Examples include the simulation of vortical flow on the complete F-16XL geometry in theme 2, the hot gas ingestion on a Harrier test case in theme 3, and the internal air systems of an engine in theme 5. Secondly, the Reynolds numbers considered are relevant to flight conditions. Thirdly, in several cases the aero-structural interaction and rigid body dynamics needs to be considered as part of the simulation. Finally, the majority of cases simulated in the consortium involve unsteady flow.

A number of landmark simulations have been completed. A small number of examples are included here to illustrate the current status of large scale CFD for applied aerodynamics.

A classical problem in helicopter aerodynamics is the behaviour and influence of the tip vortices. Simulations⁽²³⁾ have investigated the mesh requirements for resolving these vortices, which tend to diffuse in grid based simulations. Results obtained on up to 32m points have shown that the influence of the tip vortices on the integrated loads is perhaps not as important as previously imagined. The level of mesh density required to simulate interactions has also been demonstrated. Future work being prepared is the simulation of a complete helicopter and this is being approached with confidence due to this sort of building block study.

Cavity flows⁽²⁴⁾ are important due to the renewed interest in internal store carriage for stealthy aircraft. The high Reynolds number, high speed flow over a deep cavity is one of the most challenging aerodynamic flows due to the highly energetic unsteady flow which is established. Detached eddy simulations (DES) requiring 50,000 time steps on a computational grid with 4m points were carried out and showed that DES can consistently and robustly provide good agreement with measurements for the empty cavity case. This has paved the way for studies involving the incorporation of active and passive flow control techniques, and their influence on cavity stores.

An unsteady Reynolds Averaged Navier-Stokes (URANS) study of the Harrier in vertical descent⁽²⁵⁾ has been carried out. The computational grid, a mixture of tetrahedral and hexahedral cells, had 22m

points and was required to deform as the Harrier descended. The impingement and interaction of the hot jets, and their re-ingestion, was simulated and validated with experimental measurements. A partner study⁽²⁶⁾ is simulating the hovering version of this problem using large eddy simulation (LES) to improve the prediction of the behaviour of the jets.

The foregoing examples all required the computing power afforded by HPCx. The flagship results have served a number of purposes. First, the ability of numerical methods has been stretched beyond current practice. For example, LES has been attempted on unstructured grids required for aircraft geometries with some valuable insight into the accuracy requirements. Secondly, demonstration of simulations on recognisable aircraft geometries has helped to raise the interest of industry. Thirdly, the issue of grid and time dependency can be investigated by one or more highly resolved simulations. Finally, a very expensive calculation can help to show how more routine calculations should be approached. An example of this was supplied by the cavity case where turbulence simulation was shown to be successful where turbulence modelling had previously failed.

The examples provided are not exhaustive. Results have been obtained for various other problems including: drag from helicopter fuselages; design optimisation of a blended wing body; vortical flow on F-16XL aircraft; cavity flow in the internal air system of an aero-engine; rotating stall simulation in an aero-engine; store separation on unstructured meshes.

One area of weakness exposed by these projects has been in the area of visualisation. Very large datasets have been generated which have exceeded the ability of current commercial visualisation tools. Work at Cambridge has started to investigate the use of virtual reality but this effort is at an early stage. A support effort has been made at Daresbury Laboratory to develop some identification methods for vortices which can be used in an automatic fashion. Much remains to be done in this area, particularly for unsteady flows. However, future success will also depend on access to adequate supercomputing resources but, as illustrated in the next section, key technical challenges are also facing the computer industry.

5.0 TRENDS IN HIGH-END COMPUTING

5.1 Hitting the memory wall

For over a decade it has been apparent that the increase in micro-processor clock rates exceeds the rate of increase in the speed of dynamic random access memory (DRAM) components. Both speeds are increasing exponentially, but the exponent for microprocessors is substantially larger than that for DRAMs⁽²⁷⁾. The consequence of this increasing gap is that any program with even a modest cache miss rate will eventually become completely memory bound. This issue is known as the memory wall.

Main memory is now nearly 1,000 clock cycles from the

‡ Defence and Aerospace Defence Partnerships

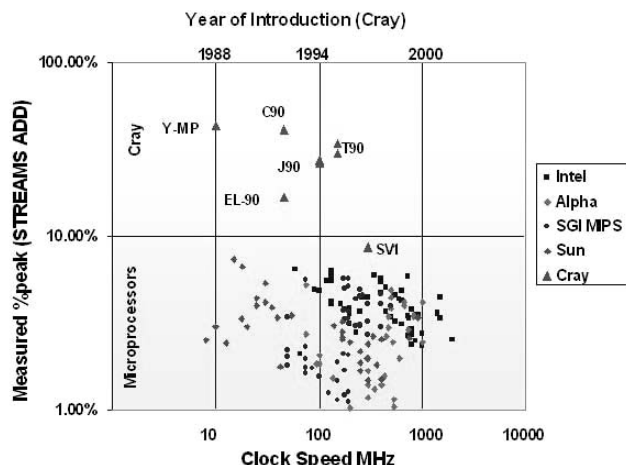


Figure 1. The percentage of peak achieved by the STREAMS ADD benchmark as a function of processor clock speed and year of introduction.

processor. The situation is worse with shared memory nodes and multi-core chips where caches and memory pathways are shared between processors. Designers have attempted to compensate for memory penalties by the introduction of complex memory hierarchies involving up to three levels of cache between the processor's registers and main memory. However, such architectures are expensive, can be inefficient, and introduce special problems for programmers and compiler writers in controlling cache content and managing the memory hierarchy. Attempts at hiding the memory latency by speculative loading of data mean that increases in memory bandwidth bring diminishing returns.

Many researchers have noted the decline in the percentage of peak performance attainable by real application codes⁽²⁸⁾. On vector systems, many codes were capable of achieving around 50% of the peak performance, whereas on current microprocessor systems only 10% or less is commonly achieved. This goes a long way to negating the cost advantage of the commodity-based systems. Figure 1 shows the percentage of peak achieved by the STREAMS ADD benchmark as a function of processor clock speed and year of introduction. As the clock speed increases within a particular microprocessor generation the performance degrades to 1%-3%. New microprocessor generations 'reset' this performance to at most 6% of peak. Consequently, applications with limited memory reuse perform particularly inefficiently on today's systems and it is becoming increasingly important for code developers to understand the detailed issues of data management and cache reuse.

5.2 The power wall

Moore's Law⁽¹⁾ has served the computing industry well for some forty years and feature sizes of 0.1 micron are expected in 2006 leading to 200m transistors per chip. However the trend towards finer and finer lithography leading to increasing numbers of transistors operating at higher and higher clock rates is leading to fundamental design restrictions. Higher densities of transistors result in a higher power density on the chip exacerbated by higher clock speeds. The days of ever increasing microprocessor clock rates are numbered as power and cooling requirements become exorbitant. Cooling systems themselves consume power increasing further the cost of running the service. Table 2 shows the estimated power consumption and electricity costs for some leading high-end systems. It is clear that if multi-100 Teraflop/s systems are built with current technology, the running costs over the lifetime of the machine will exceed the cost of the system itself. This has become known as the 'Power Wall'⁽²⁹⁾. In many cases,

Table 2
Estimated power consumption and electricity costs for some leading high-end systems

System	Peak Teraflop/s (approx)	Power consumption in MW (approx)	Estimated cost/year
HPCx	10	1	
Red Storm	20	2	
Earth Simulator	40	7	\$6M
HECToR	50-100	?	?

the additional number of transistors available on a chip is simply being used to support multi-core chips. Dual-cores are commonplace now and microprocessor roadmaps predict four-way and eight-way multi-core chips within the next few years. Another use for the additional 'real estate' is the provision of more and larger on-core caches, memory controllers, and communications functions.

New power efficient designs will address the power issue directly, looking at ways to reduce transistor power by lowering the operating voltage, and/or reducing the oxide thickness or the channel length or to reduce the switching rate for key operations. Replacing copper or aluminium connections on motherboards and even within processors themselves by optical fibres driven by miniature lasers could be beneficial as optical fibre uses less power and yields better performance. The cost of this technology is high but it may become an essential element of future designs. Some saving can also be made by using asynchronous clocks which reduce the number of simultaneous operations and competing for power.

6.0 NOVEL SOLUTIONS FOR FUTURE TECHNOLOGIES

6.1 Blue Gene

The BlueGene/L Massively Parallel Processing (MPP) architecture from IBM⁽³⁰⁾ uses a reduced processor clock speed to give greatly increased system packing density that reduces floor space, power consumption, and cooling requirements. These properties allow MPP systems to be built comprising of many tens of thousands of processors in a relatively small space with reduced infrastructure overheads. Each Blue Gene CPU contains two modified PowerPC 400s running at 700MHz. A CPU therefore delivers a peak speed of 2.8Gflop/s as the two floating point units (FPUs) can perform fused multiply-add operations. In virtual processor mode both CPUs are used for computation whilst in co-processor mode one processor is used for computation as the other handles communication tasks. By current standards, Blue Gene processors yield only modest performance and, to achieve good system performance, application codes must be able to scale up to many thousands of processors. To this end, BlueGene/L possesses five networks, two of which are of interest for inter-processor communication: a 3D torus network and a tree network. The torus network is used for most general communication patterns and the tree-based network is used for collective communications such as broadcast and reduction operations.

Currently a BlueGene/L machine at the Department of Energy's NNSA/Lawrence Livermore National Laboratory in California⁽³¹⁾, occupies the number one position in the TOP500 supercomputing list⁽³²⁾. This system contains 131,072 processors which yields a sustained performance of 280Tflop/s on the Linpack benchmark. In the UK, a Blue Gene system of 2,048 processors, yielding 4.7Tflop/s Linpack performance, is being evaluated at Edinburgh Parallel Computing Centre⁽³³⁾.

6.2 Field programmable gate arrays (FPGAs)

One approach to address the power, efficiency and scalability problems outlined above is to create processors that are specially designed for specific tasks. Application-specific integrated circuits (ASICs) have algorithms implemented in the design of the silicon and are therefore inflexible in that they require an expensive and time consuming redesign for each application. However, FPGAs⁽³⁴⁾ are general-purpose integrated circuits that are configured by the user rather than the device manufacturer. This is achieved by downloading a configuration program called a bitstream into the memory of the chip. In this way FPGAs can be programmed to carry out exactly the required operations. By designing the on-chip logic to take advantage of parallelism in the target application, highly efficient distributed memory architectures can be produced which may achieve raw memory bandwidths of many hundreds of terabytes per second. As with Blue Gene processors, FPGAs circumvent the power wall by running at a much lower clock speed than contemporary HEC processors (e.g. 100-200MHz), but due to the reduced effects of the memory wall a far higher percentage of peak performance can be achieved than with most other HEC architectures.

The technology is currently being explored by leading vendors, such as SGI⁽³⁵⁾ and Cray⁽³⁶⁾, with a view to using FPGAs as co-processors alongside general purpose CPUs in HEC systems. This would involve offloading compute-intensive parts of the processing onto FPGAs and therefore performance and scalability may be affected by overheads associated with data transfer, scheduling and synchronisation. However, it should be noted that, at present, most FPGAs are not optimised for floating point arithmetic and their efficient use requires developers to program at the hardware level although some high-level tools are now emerging to assist programmers in this task^(37,38).

6.3 IBM cell processor

IBM's cell processor has been produced in collaboration with Sony and Toshiba⁽³⁹⁾. The processor has been described as a 'system on a chip' due to its multi-core heterogeneous hierarchical architecture. It consists of a single 64-bit RISC PowerPC processor and eight 32-bit 'Synergistic Processing Elements' (SPEs) connected by a fast internal bus (EIB). High-speed chip-to-chip communication between multiple cell processors is enabled by a special processor interface and a FlexIO bus. A prototype cell processor ran at 4GHz and is therefore capable of a theoretical 256Gflop/s (single precision). To summarise briefly, the cell processor attempts to optimise performance by assigning all logic control to the PowerPC processor which can therefore start, stop, interrupt and schedule processes to run on the 'workhorse' SPEs. Present day compilers for cell processors may not be sufficiently developed to maximise performance for application codes automatically and therefore input is often required from programmers in order to achieve good performance.

6.4 Clearspeak technologies

The Clearspeak CSX600(40) is an embedded low-power 64-bit parallel co-processor. The principle design strategy is to trade clock speed for efficient parallel performance from the 96 multiple SIMD processing element (PE) cores on board the chip. Although the clock speed is a low 250MHz, many times more work may be achieved during each cycle than with general purpose CPUs. The CSX600's PEs are basic processing units that require a general-purpose host to pass them commands and data. Each PE contains dual pipeline floating-point units each capable of two multiply-add operations, thus the theoretical peak performance of the CSX600 is $96 \times 2 \times 2 \times 0.25 = 96\text{Gflop/s}$. Both AMD and Intel are exploring the use of Clearspeak as mathematical co-processors running specifically optimised computational kernels such as BLAS routines⁽⁴¹⁾.

7.0 CONCLUDING REMARKS

This paper has provided a brief overview of early pioneering work in developing numerical techniques to solve scientific problems and highlighted a few of the achievements by the UK Applied Aerodynamics Consortium. Today's computational research involves the exploitation of massively parallel computer facilities and the paper has provided a brief review of the UK's supercomputing history and its transition to parallel processing in 1994. However, as technology continues to advance at such a rapid pace, barriers to future progress are starting to make a real impact. This effect is already hitting floating point performance where memory access is becoming a limiting factor. As the next generation of supercomputers will need to exploit 1,000s of processors, new approaches such as field programmable gate arrays are already being explored alongside alternative strategies to overcoming these barriers. Other factors, such as the generation, storage, and manipulation of large data sets for efficient post-processing, crucial to the analysis and understanding of complex aerodynamic phenomena, also need to be addressed by the consortium. The UK Applied Aerodynamics Consortium has been very successful in a very short space of time. The consortium has demonstrated that its codes are efficient (several codes have been awarded a gold star on HPCx for their excellent scaling performance), mature and robust and are well placed to make a major scientific impact on HECToR, the UK's next supercomputing facility.

ACKNOWLEDGEMENTS

This work was supported through the Engineering and Physical Sciences Research Council (EPSRC) under grant GR/S91130/01. Additional support was provided by EPSRC under the auspices of Collaborative Computational Project 12 (CCP12).

REFERENCES

1. MOORE, G.E. Cramming more components onto integrated circuits, *Electronics*, April 1965, **38**, (8).
2. MOORE, G.E. Progress in digital integrated electronics, 1975, IEEE International Electron Devices Meeting, pp. 11-13.
3. CHAPMAN, D.R. Computational aerodynamics development and outlook, 1979, AIAA Paper 79-0129.
4. PETERSON, V.L. Impact of computers on aerodynamics research and development, Proc of the IEEE, 1984, **72**, (1), pp 68-79.
5. KORKEGI, R.H. Impact of computational fluid dynamics on development test facilities, *J Aircr*, 1985, **22**, (3), pp 182-187.
6. JOHNSON, T.J., TINOCO, E.N. and YU, N.J. Thirty years of development and application of CFD at Boeing commercial airplanes, Seattle, *Computers and Fluids*, 2005, **34**, (10), pp 1115-1151.
7. RICHARDSON, L.F. The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to stresses in a masonry dam, *Phil Trans Roy Soc London Series A*, 1911, **210**, pp 307-357.
8. KAWAGUTI, M. Numerical solution of the Navier-Stokes equations for the flow around a circular cylinder at Reynolds number 40, *J Physical Society of Japan*, 1953, **8**, (6), pp 747-757.
9. AMDAHL, G. Validity of the single processor approach to achieving large scale computing capabilities, 1967, AFIPS Conference Proceedings, pp 483-485.
10. EMERSON, D.R. Introduction to parallel computers: architecture and algorithms, *High Performance Computing in Fluid Dynamics*, 1996, WESSELING, P. (Ed), Kluwer Academic Publishers, pp 1-42.
11. GUSTAFSON, J.L. Re-evaluating Amdahl's law, *Communications of the ACM*, 1988, **31**, (5), pp 532-533.
12. BHATT, S.N. and LEIGHTON, F.T. A framework for solving VLSI graph layout problems, *J Computer and System Sciences*, 1984, **28**, pp 300-343.
13. GARY, M., JOHNSON, D. and STOCKMEYER, L. Some simplified NP-complete graph problems, *Theoretical Computer Science*, 1976, **1**, pp 237-267.

14. FARHAT, C. A simple and efficient automatic FEM domain decomposer, *Comp and Struc*, 1988, **28**, pp 579-602.
15. KERNIGAN, B.W. and LIN, S. An efficient heuristic procedure for partitioning graphs, *The Bell System Technical J*, 1970, **49**, pp 291-307.
16. SIMON, H.D. Partitioning of unstructured problems for parallel processing, *Computer Systems in Engineering*, 1991, **2**, (2/3), pp 135-148.
17. POTHEN, A., SIMON, H. and LIOU, K.-P. Partitioning sparse matrices with Eigenvectors of graphs, *SIAM J. Mat Anal Appl*, 1990, **11**, (3), pp 430-452.
18. BARNARD, S.T. and SIMON, H.D. A fast multilevel implementation of recursive spectral bisection for partitioning unstructured problems, *Concurrency: Practice and Experience*, 1994, **6**, (2), pp 101-117.
19. HENDRICKSON, B. and LELAND, R. The CHACO user's guide version 1.0, November 1993, Sandia Report SAND93-2339.
20. WALSHAW, C., CROSS, M., EVERETT, M.G. and JOHNSON, S. JOSTLE: Partitioning of unstructured meshes for massively parallel machines. *Parallel Computational Fluid Dynamics: New Algorithms and Applications*, 1995, SATOFUKA, N. et al (Eds), pp 273-280.
21. KARYPIS, G. and KUMAR, V. A fast and high quality multilevel scheme for partitioning irregular graphs, 1995, International Conference on Parallel Processing, pp 113-122.
22. MAVRIPLIS, D.J., AFTOSMIS, M.J. and BERGER, M. High resolution aerospace applications using the NASA Columbia supercomputer, 2005, Joint winner Best Paper, SuperComputing 2005.
23. ALLEN, C.B., SUNDERLAND, A.G and JOHNSTONE, R. Application of parallel rotor simulation tools on HPCx, scheduled for publication in *Aeronaut J*, March 2007.
24. NAYYAR, P, BARAKOS, G.N. and BADCOCK, K.J. Numerical study of transonic cavity flows using large-eddy and detached eddy simulation, scheduled for publication in *Aeronaut J*, March 2007.
25. RICHARDSON, G.A., DAWES, W.N. and SAVILL, A.M. An unsteady, moving mesh CFD simulation for Harrier hot-gas ingestion control analysis, submitted to *Aeronaut J*, 2006.
26. LI, Q, PAGE, G.J. and MCGUIRK, J.J. Large eddy simulation of twin impinging jets in crossflow, submitted to *Aeronaut J*, 2006.
27. WULF, W. and MCKEE, S. Hitting the memory wall: implications of the obvious, *Computer Architecture News*, 1995, **23**, (1).
28. ASHWORTH, M. Parallel processing in environmental modelling, *Parallel Supercomputing in Atmospheric Science*, 1993, Proceedings of the Fifth Workshop on the Use of Parallel Processors in Meteorology, HOFFMANN, G-R. and KAURANNE, T. (Eds), pp 1-25, World Scientific.
29. FLYNN, M.J. and HUNG, P. Microprocessor design issues: thoughts on the road ahead, *IEEE Micro*, 2005, **25**, (3), pp 16-31.
30. IBM Research Blue Gene Project Page, <http://www.research.ibm.com/bluegene/>
31. Lawrence Livermore National Laboratory, <http://www.llnl.gov/>
32. Top 500 Supercomputing Sites, <http://www.top500.org/>
33. The University of Edinburgh Blue Gene Server, http://www.epcc.ed.ac.uk/computing/advanced_computing/BlueGene/
34. A brief introduction to FPGAs, Xilinx v8.1 User Guide, http://www.xilinx.com/products/software/sysgen/app_docs/user_guide_Chapter_3_Section_1.htm
35. SGI RASC Technology, <http://www.sgi.com/products/rasc/>
36. Application acceleration with FPGA-based reconfigurable computing, <http://www.cray.com/products/xd1/acceleration.html>
37. The Mitrion Software Development Kit, <http://www.mitrion.com/msdk.shtml>
38. Xilinx ESL Starter Kit, <http://www.celoxica.com/xilinx/eslkit/>
39. The Cell Project at IBM research, <http://www.research.ibm.com/cell/>
40. CSX600 datasheet, <http://www.clearspeed.com/downloads/CSX600Processor.pdf>
41. GUSTAFSON, J.L. and GEER, B.S. A hardware accelerator for the Intel math kernel library, ClearSpeed Whitepaper, <http://www.clearspeed.com/downloads/Intel%20Math%20Kernel%20whitepaper.pdf>