The Development and Operation of Edinburgh Parallel Computing Centre's Summer Scholarship Programme

G. V. Wilson N. B. MacDonald C. Thornborrow C. M. Brough
Edinburgh Parallel Computing Centre
The University of Edinburgh
Edinburgh EH9 3JZ
United Kingdom
epccssp@epcc.ed.ac.uk

Abstract

Between 1987 and 1994, more than 100 students in a broad range of disciplines worked as summer scholars at Edinburgh Parallel Computing Centre. Many of these students have since taken their parallel computing skills into graduate work and industry, and over a quarter of EPCC's technical staff are alumni of the Programme. This report describes the evolution and present operation of the Summer Scholarship Programme, and its costs and benefits.

1 Introduction

Edinburgh Parallel Computing Centre (EPCC) is the focus of the University of Edinburgh's interests in high-performance computing. Its main aim is to accelerate the exploitation of high-performance parallel computing systems in academia, industry and commerce.

EPCC has employed summer students since 1987, and has run a Summer Scholarship Programme since 1989. This Programme has been very successful in tackling four problems facing the Centre:

User recruitment and training: EPCC provides a national service to UK academic institutions on a number of parallel systems. Potential users must be made aware of these services, and shown their potential.

Programmer training: As with any new technology, the uptake of parallel computing by industry will largely be determined by the rate at which people can be trained to use parallel computers effectively. EPCC has a responsibility to the whole community to enlarge the parallel computing skill base.

Programmer recruitment: EPCC needs to recruit staff with parallel computing skills and experience. As will be discussed below, the Summer Scholarship Programme has proved to be a rich source of talented, motivated people.

Staff development: Most of EPCC's work involves direct collaboration with industry on sizeable projects. The Summer Scholarship Programme helps prepare the Centre's technical staff for project management by giving them supervisory experience.

Many current users of the Centre's parallel machines initially tested the waters with the help of summer scholars. The Programme has given experience of parallel computing to over 100 students (predominantly senior undergraduates) during the ten-week summer vacation period, and many former summer scholars are now using their experience as they pursue careers in academia and industry. Over a quarter of the 35 staff directly involved in technical work on parallel systems at the Centre are former summer scholars, and many of the staff who are now responsible for managing industrial collaborations gained their first experience of management as supervisors in the Summer Scholarship Programme.

We begin this paper with a review of the evolution of the Programme (Section 2), before describing its current operation (Section 3), and the projects in this year's Programme (Section 4).

1063-9535/94 $4.00 © 1994 IEEE

Year of Study	'87	'88	'89	'90	'91	'92	'93	'94
2	1	2	6	8	6	3	2	2
3	1	1	4	5	18	7	4	8
4		1	2	1	2	10	12	5
graduate						5		1
total	2	4	12	14	26	25	18	16
returning		1		3	2	3	1	

Table 1: Student Numbers

2 Evolution

In 1987, an undergraduate in the university's Department of Physics decided that he would like to learn about parallel computing during his summer job. At the same time, an electronic engineering student at a local college wanted to spend a work term doing something with computers at the University. The first author was made responsible for finding projects for them, and for supervising them.

By the end of the summer, the physicist had implemented a diffusion-limited aggregation program, which was used as a demonstrator for the next two years, while the college student had ported a graph-plotting package called EasyGraph to graphics boards embedded in a Meiko Computing Surface. Based on these successes, the Department of Physics provided money the following summer (1988) to employ two more of its undergraduates. The first author was also made responsible for supervising two visiting American undergraduates.

The *ad hoc* arrangements of 1987 and 1988 were formalized in 1989 as the Summer Scholarship Programme and twelve students were hired. Table 1 shows how far through their degrees these students were when they were recruited, in this and other years. Each student was required to give two half-hour presentations on his or her work to their peers during the course of the summer, and to prepare a final report on it in September. While the first author was responsible for day-to-day supervision, several of their projects were proposed, and co-supervised, by members of academic staff from Physics, Computer Science, Chemical Engineering, and the Department of Mechanical Engineering at Heriot-Watt University.

In 1990, the Programme expanded again. Fourteen undergraduates were employed, while four graduate students funded from other sources worked beside them. Several of that year's projects were continuations of projects done in the previous year, while two

System	'89	'90	'91	'92	'93	'94
Occam	4	1	2			
C/CS Tools	4	6	8		1	
Fortran/CS Tools	3	3	3	1		
Fortran-Plus		4	6			
C-Linda			1			
Prolog-Linda			2			
C/Unix	1	1	6	1		1
C/CHIMP				12	12	3
RPL/RPL-2				2	1	2
Concurrent ML				1		
CM-FORTRAN				5	3	2
C*				1		
C/MPI						8
FORTRAN/MPI						4
none	1			2		

(These numbers may not sum correctly, as some students used several systems.)

Table 2: Software Used

Category	'89	'90	'91	'92	'93	'94
Computer Science /Visualisation	7	3	11	7	7	5
Engineering	1	3		3		1
Business/Marketing	1			1	1	
Mathematics /Optimization			2	4	2	1
Physical Sciences	2	5	5	8	4	7
Teaching and Training			4			
Utilities/Tools	1	3	4	2	1	1

Table 3: Project Types

of the students employed were alumni of 1989.

The 1990 Programme was less successful than its predecessor for several reasons. The first was tardiness: projects were not solicited, and students were not recruited, early enough. As a result, many of the students taken on had finished only the second year of their undergraduate degree, rather than the third. In addition, several of those taken from the physical sciences and mathematics had little or no previous programming experience.

More effort was put into organising the 1991 Programme. Twenty-six undergraduate students were employed that year to do a wide variety of projects using SIMD and MIMD platforms and a network of workstations. These students were supervised by more than a dozen people drawn from EPCC technical staff, the University's academic staff, and the graduate stu-

Degree Programme	'89	'90	'91	'92	'93	'94
Computer Science	2	5	9	7	9	8
& Artificial Intelligence	4	2	6	5	2	1
& Electrical Engineering	1		1	1	2	2
& Management Science			1			
& Mathematics				2		
Electrical Engineering	1			1		
Civil Engineering						1
Mechanical Engineering		3				
Mathematics		1	2	1	1	1
Physics	4	3	7	4	2	2
Other Physical Sciences				2	2	1
Business				1		
Geography				1		

Table 4: Student Backgrounds

dent community.

When the second author took over the Programme in 1992, recruitment for it was internationalized. Almost 200 applications were received from students at dozens of institutions around the world. This necessitated a more formal selection process: interviews were carried out face-to-face with students from Edinburgh, and by telephone with students from further afield. The telephone interviews were very demanding, for both the applicant and the interviewer, but we believe that they were crucial to recruiting a strong group of students. Eventually, 25 students from 11 institutions in 6 countries came to Edinburgh for the summer. A formal review process for project proposals was also established for the first time in 1992, and several projects were tackled by students working in pairs. Typically, a computer science student worked with one whose background was in the physical sciences, so that strong programming skills and detailed knowledge of a particular application domain could be combined in a balanced team. Our concerns about the difficulties of this approach proved to be almost entirely unfounded, as these joint projects were particularly successful. This summer saw collaboration with an increased range of academics, including the Departments of Business Studies and Geography.

In 1993, responsibility for the Programme changed hands again, to a team of three led by the third author. The format of the Programme was largely unaltered, with world-wide recruitment and external project supervision being maintained. The number of students was reduced that year, as fewer members of EPCC staff were available for supervision due to an increased number of external contracts. 180 applicants were screened, 70 interviewed, and 18 accepted from five countries. The greatest change in the 1993 Programme was an increased reliance on multiple supervisors. This proved very beneficial during the projects: every project was watched more carefully than it would have been by an individual, and projects that encountered difficulties, for whatever reason, were spotted and dealt with much more quickly than in previous years.

This year, the Programme is managed by the fourth author. 16 students from 6 countries around the world are spending their summer at EPCC learning about parallel computing as they collaborate with University researchers and members of technical staff.

3 Recent operation

The Programme has evolved over the years from *ad hoc* supervision of two students to an organized effort to recruit, train, and assess a large pool of students. In its present form, the Programme is organized into six stages: recruiting students, selecting projects and assigning students and supervisors to them, making sure students and supervisors know what they are supposed to do and when, running the Programme on a day-to-day basis once students are in place, collecting project results, and assessing students' performance. The sections below deal with these in turn.

3.1 Recruiting students

Recruiting good students has always been the most difficult part of the Programme, and a variety of approaches have been adopted over the years. In the beginning, members of academic staff in various departments (particularly Computer Science and Physics) were asked to suggest promising students. These were then approached individually. This was reasonably successful — many students seemed to feel flattered that they had been singled out for attention — but required a great deal of time.

A less selective, but more productive, approach was to go into classes to make a brief presentation and solicit applications. This generated a great many responses, but filtering them proved to be almost as time-consuming as one-to-one recruitment. Notices put in the University Bulletin and EPCC's own newsletter, and messages posted to electronic bulletin boards with a university-wide distribution, produced disappointingly little response.

Prior to 1992, the vast majority of summer scholars were students at the University of Edinburgh. Individual recruitment from this group could therefore be based on both an examination of a student's academic record and a one-to-one interview with him or her. We have found that while a good academic record is usually a sign of ability, it is not necessarily an indication that a student is willing to follow direction. Conversely, a poor academic record is sometimes not a sign of a lack of ability, but rather reflects that the student is more interested in learning about computers than in passing exams. In 1991, approximately one-third of students were taken despite their academic records, and only a few of this group disappointed.

In 1992, recruitment was expanded to include institutions throughout the U.K., Western Europe, and North America. This was motivated primarily by the Programme's role as a recruiting ground for EPCC staff. Most students being recruited locally meant that a significant proportion of the Centre's staff had come from the same institution, gaining the same perspective of their subjects. Concerns over this lack of diversity also arose within the Summer Scholarship Programme itself. Because many students had studied the same subjects and worked together for months or years before joining the Programme, it was becoming increasingly difficult to create a stimulating and challenging atmosphere.

In order to attract a wider range of students, flyers advertising the Programme were sent to the heads of all departments in the University which taught numerate disciplines. Departments such as Economics, Geography, Business Studies, and the Biological Sciences were targeted for the first time, which broadened the base of applicants. Much greater efforts were made to attract outside students by sending the flyer to over 100 other institutions around the world; the mailing list used was drawn up using personal contacts of the Centre's staff and lists maintained by the University's International Office.

These measures have proved to be successful. The Programme now attracts roughly 200 applicants each year from dozens of different institutions. After an initial selection phase, some 40 students are interviewed. For local students, this is done face-to-face by a panel which includes the coordinator of the Programme and two other technical staff (with experience relevant to the student's background wherever possible). Interviews are conducted by telephone with students from outside Edinburgh. Although organizing times and numbers for telephone interviews is onerous, and the interviews themselves are demanding, we believe that they are a much more successful basis for selection than written applications alone.

The increased diversity of participant backgrounds has had a marked effect on the Programme, and has remedied earlier concerns about inbreeding. A number of new difficulties have arisen, however. Firstly, differences in institutions' summer vacation dates have prevented a number of students from taking part in the Programme. While this problem cannot be eliminated, careful scheduling of the Programme can minimize its impact. Secondly, we have found that the administration in many institutions have not distributed our flyers to a range of departments as requested, but instead forwarded it to Computer Science alone. This has led to an unwelcome imbalance in the background of applicants. We now attempt to counter this by stressing the interdisciplinary nature of the Programme in a covering letter.

We have found that it is important to select students as early as possible in the academic year, if strong applicants are not to accept jobs elsewhere. Information about the Programme is typically circulated in the preceding November, with a closing date for applications at the end of January. Interviews are conducted in late February and early March, and final offers are made before Easter.

3.2 Project selection

Finding projects and supervisors for students first became an issue in 1989, when the Programme expanded from four to twelve students. The logical course of action would have been to ask people using EPCC's facilities to suggest projects, to select the most promising of these, and then to try to find students whose skills would be a good match for particular projects. However, it proved difficult to obtain project proposals from academics far enough in advance of the summer for them to be used in recruitment. Moreover, asking established users to propose projects would not achieve our goal of expanding our user community.

Our selection scheme has therefore concentrated on demonstrating the benefits of parallel supercomputing to a particular researcher or research group. In these cases, there is usually a large existing program with which the student must come to grips before any parallel programming can begin.

Researchers proposing a project are often keen to recommend a student. As the number of students from other institutions has increased, we have made a conscious effort to separate the proposal of projects from the proposal of students, preferring to match up students with projects late in, or even after, the selection process. However, encouraging researchers to get involved in parallel computing continues to be a major aim, so a call for project proposals is circulated to all appropriate departments within the university.

Once project proposals start to arrive, filtering them is just as difficult as filtering students. A good project proposal will have a number of built-in contingencies, so that a strong student can be kept occupied, or the project's scope reduced for a weaker student. In addition, a team project proposal must address the problem of breaking down the work into well-defined sub-projects. Some of the proposals received each year would make excellent Ph.D. research topics, but are overly ambitious for a ten-week undergraduate project; their proposers are sometimes reluctant to acknowledge this. On the other hand, some of the ideas received are impossibly vague, rather than impossibly ambitious, or are based on an unrealistic impression of what parallel computers can do. Since 1992, the portfolio of project proposals has been reviewed by a panel of technical staff in EPCC, who are responsible for ensuring that the proposals are viable. Researchers submit *pro forma* proposals, which include details of the skills required in a student, the materials which the researcher will make available, and the intended programme of work. Each year, a number of proposers are asked to reconsider and re-submit their plans. The formality of this process has the advantage of forcing supervisors to think in detail about the project well in advance of it starting. Since this process was put in place, very few of the projects selected have failed because of intrinsic weaknesses.

In contrast with previous years, in which most projects were proposed by members of various departments, roughly three-quarters of the projects selected in 1991 were proposed and supervised by EPCC staff. This reflected both the increase in the number of people employed by EPCC, and their increasing sophistication. However, supervising twenty-six student projects consumed an unacceptable amount of time. We have now returned to the pattern of previous years, with EPCC staff being solely responsible for only a few projects, and assisting in the supervision of the others, as staff numbers and experience have now grown sufficiently to support the (reduced) student load.

3.3 Preparation and training

Once students and supervisors have been selected, the next step is to arrange accommodation and resources.

Pressure on working accommodation within the University has made it impossible for us to provide offices for summer students. As a result, students do most of their work in public computing laboratories, and make (limited) use of lockers which we arrange for secure storage of their belongings.

Providing living accommodation for students has become an issue in recent years as the number of students coming from outside Edinburgh has increased. We have made arrangements for two types of accommodation: study bedrooms in University accommodation; and flats for students accompanied by their partners. On occasion, it has been convenient for us to pay rent directly and deduct appropriate amounts from scholarship payments. Organising accommodation is time-consuming, often involving considerable liaison with other organisations and the students concerned, but we feel it is necessary if we are to attract participants from other institutions and countries.

With these arrangements in place, the initial priority for the programme is to make sure that students and supervisors know what they are expected to do, and to teach them what they need to know in order to do it.

In the early years of the Programme, the first step was to hold a one-hour meeting at the beginning of the third term (i.e. early May), at which students were told as a group how the Programme would be run, what they would have to produce, and how they would be assessed. The most important benefit of this meeting was to ensure that they already knew, as the summer began, who their peers would be. Many of the physical science students taken on were weak programmers, often with little or no previous experience with Unix. Our hope was that if a strong sense of community could be developed, much of the mundane help they would need would be provided by their fellow students. This goal was largely realised.

Since the number of students from other institutions has increased, this early meeting has been impractical. We have instead devoted time during the first week of the program to practical training covering:

- an introduction to Unix, X Windows, and basic utilities
- software engineering tools (debuggers, configuration management, source code control)
- an introduction to parallel computing
- data parallel programming
- message-passing parallel programming
- scientific visualization tools

- document preparation utilities

We have found that this training substantially reduces the amount of one-to-one supervision which students require once their project work begins. Another benefit is that all the students are exposed to a range of tools and programming models. This helps them to relate more easily to the work of other students later in the summer. Finally, by having students work in different pairs each day, this short course allows them to get to know each other, and to recognize each other's strengths and weaknesses. This in turn fosters a stronger and more self-reliant community.

Supervisors, too, need training. Every project is now supervised at least in part by a member of EPCC staff. As the Programme has expanded, it has become increasingly important to ensure consistent supervision, while retaining a good balance between the freedom of individual supervisors and students and the coherence of the whole Programme. All staff are issued with a guide which outlines what is expected of students and supervisors, and explains which member of staff is responsible for dealing with the most common types of problems.

3.4 Operation

The day-to-day running of the Programme now requires far less involvement from any single individual than it did in the initial years. This is due to three factors:

1. The number of individual project supervisors has greatly increased.
2. The training given the students in the first week's lectures forestalls a great deal of individual hand-holding.
3. EPCC's User Support services were restructured in 1991, and are now substantially more effective than they were.

Meetings between supervisors and students are arranged by the parties involved. This has not been uniformly successful — some students find it difficult to get time from their supervisors when they needed it, while some supervisors have difficulty finding their students during normal working hours — but it has worked well in most cases.

The only formal requirements on all students are that each must prepare a weekly "5:15" report and must give two half-hour presentations on his or her progress during the summer. The 5:15 is a one-page report outlining the student's activities during the week, highlighting any problems or issues, and outlining plans and priorities for the future. Its name comes from the fact that it is supposed to take 15 minutes to write, and 5 minutes to read, and is due by 5:15 each Friday. Students submit copies to their supervisor and to the programme coordinator.

Two or three presentations are given over lunch each Tuesday and Thursday, lunchtime being chosen so that supervisors can more easily attend. All students are required to attend all presentations; supervisors are welcome at any time, and are strongly encouraged to be in the audience for their students' talks.

These presentations serve several purposes. First, they force students to talk about their work in front of an audience. For many of them, this is the first time they must prepare slides, speak to an audience, and deal with questions. By the end of the summer, having been given a critique of their first talks, most students are able to present their final results reasonably well.

Second, these presentations give the students the feeling that other people are interested in what they are doing. They also give the students some feeling for how much progress they are making in comparison with their peers, and encourage discussion among students about common problems. While there have been a few notable exceptions, most students put considerable effort into their presentations, and report that they considered them very worthwhile.

From an organizational point of view, the most important benefit of both the weekly talks and the 5:15 reports is that they allow the Programme organizer to track each project without having to chase individual students. If a project is in trouble, or lacking a clear direction, it may be apparent from the 5:15, and it is usually obvious during the student's presentation.

Throughout the summer, students are encouraged to take part in social events with the rest of the Centre's staff. At the end of the first week of the programme, students and staff share a meal and a night on the town. Regular sporting activities are arranged, football and ultimate being the most popular. Other sporting, cultural and social outings take place, organized by students as well as staff, helping to develop a strong sense of community and integrate the students into the Centre.

3.5 Deliverables

Exactly what students are supposed to deliver varies from project to project. While the primary purpose of most of the physical science projects is to generate results, the primary purpose of most of the computer science projects is to create new software. After some unfortunate experiences in 1990, students who were delivering software in 1991 were required to use version management software (RCS) to manage their work, and to provide proper documentation. Several students have reported that working to such standards is the most educational part of the summer for them, and that it has made them understand why their degree instructors place so much emphasis on such things as sensible variable names, modularity, and designing before coding.

All students are required to write and submit a final report at the end of the summer; no student receives his or her final paycheque until this report has been approved by the Programme organizer. These reports have varied in length from four pages to small theses, and in quality from terse summaries to conference-quality papers [2, 1]. All reports from 1993 [4]–[18] and most from previous years are available from EPCC. Getting these reports from students is the most difficult part of the whole Programme, as many of them would rather spend another month programming than a week writing up. More troubling has been the generally low level of students' writing skills. Even simple spelling and grammar often requires a great deal of correction, while significantly fewer than half of the students ever seem to have any idea of how a report should be structured.

In 1987-90, all final reports were proof-read by the organizer at least once. Individual project supervisors now proof-read reports before their submission, and students are encouraged to proof-read one another's work. This has only limited impact on the number of corrections required in the "final" version of each report. Eliminating this bottleneck is a major unsolved problem with the Programme.

Since 1992, each student has been required to prepare a poster on his or her work for display at the Centre's Annual Seminar. The sheer volume of work done means that the Summer Scholarship Programme has a high profile at this event, which is an opportunity for attracting industrial sponsorship. Just as importantly, the students are tangibly contributing to the Centre's showcase event.

3.6 Assessment

1991 was the only year in which both students and supervisors were formally assessed. Two questionnaires were prepared, one to ask the supervisor for his or her opinion of the project and the student, the other to ask the student similar questions about his or her supervisor. The answers given were not particularly surprising, as most students, and quite a few supervisors, had not been reticent about making their opinions known previously. In general, students who rated their supervisors highly were highly rated in their turn, while students and supervisors generally agreed on whether the project being done had been a success or not. Most Computer Science students felt that the introductory lectures could have been shorter, while most others felt that they should have been longer; none felt that their projects had been too easy, and many expressed an interest in continuing with their work.

More recently, we have reconsidered the possibility of formal assessment of students. However, we have not implemented testing for a number of reasons. Our primary objection is that the disparate backgrounds and experience of the students make it very difficult to assess their parallel computing skills in a uniform way. We have also resisted the institution of a prize for the top student for the same reason, since we cannot conceive of an objective mechanism for ranking the students. There are also some concerns that assessment will lead the students to treat the Programme as a course, rather than research experience, and that this will be to the detriment of the Programme as a whole.

3.7 Resourcing

The Programme requires a significant commitment of EPCC's resources. Coordinating the programme is approximately a half-time job all year round, with bursts of higher activity during recruitment and shortly before the Programme begins.

During the summer, we expect supervisors to spend around half a day per week with their student, in addition to attending the lunchtime talks. Again, this varies from week to week, and is typically higher at the start and end of the Programme, and lower in the middle. If a supervisor is regularly spending longer than half a day per week supervising a project, it is generally a sign that the project is running into difficulties, or that the supervisor is adopting too close a management style.

Computing resources are an important consideration for the Programme; they must be taken into account when selecting a portfolio of projects, to ensure that systems have sufficient capacity to sustain the workload. Providing access to adequate numbers of workstations for the students is essential. When certain resources, such as high performance graphics workstations, are scarce, we have found it necessary to put a booking system in place to ensure fair access to the equipment.

3.8 Sponsorship

Prior to 1992, EPCC bore the whole cost of the Summer Scholarship Programme directly. Since then, an effort has been made to attract industrial support, and the Programme has been partially supported by Shell U.K. Exploration and Production and the U.K. Meteorological Office. The main problem with sponsorship has been reconciling the needs of the Programme with an industrial perspective on appropriate work for funding. Companies quite reasonably wish to specify the problem to be tackled by a summer student they support, but it is important to ensure that the project undergoes the same review process as those proposed within the University. In practice, we have never rejected a sponsor's project, although extensive discussion has sometimes taken place. Companies have also sometimes proposed a student for the Programme, but once again we have tried to ensure that these students go through the same review procedure as other applicants.

While sponsorship for the Programme is very attractive to the Centre, we have been concerned from the outset that the additional work involved in gaining small amounts of funding and liaising with sponsoring companies would outweigh the funding's value. Our current solution is for sponsors to contribute a fixed amount to the Programme, which in fact supports several students. In return, one student is assigned to work on a project defined in conjunction with the company.

4 The 1994 Programme

This section presents short descriptions of the projects in this year's Programme, and shows the diversity of the students participating in the Programme, the nature of the projects which they are undertaking and the breadth of collaboration with University departments.

Application Engineering Tools for MPI and PUL (Kesavan Shanmugam, University of Oregon, USA and Konstantinos Tourlas, University of Athens, Greece): This project follows on from work in last year's programme which developed a post-mortem visualisation tool for the debugging and tuning of parallel applications written using EPCC's CHIMP message-passing interface and PUL parallel libraries. The aim of this year's work is to adapt the tool to MPI. Both students are computer scientists, and one is concentrating on the graphical user interface while the other is implementing the necessary trace processing.

An Investigation of Parallel Application I/O (Gordon Henderson, Maths and Engineering, Napier University, UK): The aim of this project is to evaluate the performance of parallel applications in terms of their I/O. The intention is to produce a report detailing the kinds of optimisation that may be performed for these applications in terms of configuring a parallel I/O system.

Recognising and Avoiding Thrashing in Dynamic Load Balancing (Frank Stangenberg, Computer Science, Universität Paderborn, Germany): This project involves investigating different kinds of thrashing in dynamic load balancing systems, using EPCC's PUL-SM library for irregular mesh applications. A quantitative measure of the level of thrashing within an application will be developed.

Exploiting Trace-Based Parallelism in Geophysical Calculations (Peilin Jia, Computer Science, Yale University, USA): EPCC has developed Euphrates, a tool for performing image-processing style operations on 3D data sets in parallel. One form of seismic data is recorded as time-domain traces from individual receivers. The objective of this work, in collaboration with researchers from the Department of Geology and Geophysics, is to investigate the parallelisation of trace-based operations in Euphrates.

Simulated Evolutionary Stable Strategies (James Hammerton, Computer Science and Artificial Intelligence, The University of Edinburgh, UK):

It is generally accepted in ecological science that individual members of species may adopt different modes of behaviour. Some individuals may act selfishly whereas others may be altruistic. In many cases, the evolutionary stable strategy may be a combination of selfish, altruistic and other behaviours which are co-evolving. This project will use simulation to develop an initial understanding of evolutionary stable strategies.

Genetic Algorithms for Graphics Textures (Andrew Hobden, Computer Science, Napier University, UK): Mapping textures to objects is done in two ways: 2D textures may be used, allowing real textures to be scanned and used, but losing realism once mapped, while 3D textures are trivial to map and control, but are difficult to store and generate. This project aims to use genetic algorithms to help produce functions that generate useful 3D textures.

Invasion Percolation of Fluids into Active Faults (Jianhong Xu, Computer Science and Electrical Engineering, University of Edinburgh, UK): Fluids exert a strong control on the rheology of the earth's curst under natural conditions and can even trigger earthquakes during the filling of dams, hydrocarbon extraction or in the production of geothermal energy. Building on an earlier SSP project and working in collaboration with researchers from the Department of Geology and Geophysics, the aim is to model the effect of fluid channelling on earthquake faults using a cellular automaton.

Development and Investigation of a Parallel Multigrid Algorithm (Mark White, Physics with Computer Applications, Heriot-Watt University, UK): The UK Meteorological Office is researching the application of Multigrid (MG) methods to determine which ones are most appropriate for atmospheric modelling. This project extends a previous SSP project, which implemented a distributed memory version of one multigrid method, concentrating on scalability, performance and efficiency issues.

Host Plant Selection by Cabbage Root Flies (Deborah Bellotti, Computer Science, Universita di Genoa, Italy): By modelling the behaviour of a series of individual cabbage root flies it may be possible to assess how the relative proportion of resistant and susceptible plants affects the damage sustained by a crop during the course of a growing season. This project builds on published data from work at the Scottish Crop Research Institute.

Modelling of Near-Field Properties of Seismic Waves in an Anisotropic Medium (Sergio Damas Arroyo, Computer Science, University of Granada, Spain): Working with researchers from the British Geological Survey this project is focused on adapting an existing finite difference scheme to facilitate more accurate seismic investigation of hydrocarbon reservoirs and allow the results to be visualised.

QCD Visualisation (Helen Ashton, Physics with Computer Applications, Heriot-Watt University): Quantum Chromodynamics (QCD) is the theory of the strong interaction, the force responsible for binding the primordial quarks and gluons to form the protons and neutrons that make up the majority of matter. This project aims to develop approaches for visualising datasets from Monte Carlo simulations of four-dimensional spacetime which are used to explore the properties of QCD.

Estimating Polar Interaction Energies in the Presence of Water (David Grierson, Chemistry, Aberdeen University, UK): Modern Pharmacology and Biochemistry increasingly relies on molecular modelling to explain the interactions between biological macromolecules such as proteins, DNA and RNA. One of the greatest sources of inaccuracy in existing models is the fact that most biological systems use water as their solvent, and water has a large dipole which masks long range polar interactions. This project makes use of high-performance computing to investigate ways of accounting for the masking effect of water without having to account for each water molecule.

Echinodome Design Curves (Richard Boyd, Civil Engineering, The University of Edinburgh, UK): The optimal shape for an underwater storage tank turns out to be very similar to that of a sea-urchin, hence "Echinodome". The generation of large suites of curves can be time-consuming, so, working with researchers from the Department of Civil Engineering, this project aims to use high-performance computing to speed this process.

Visualisation of Genetic Algorithms (Graham Clark, Computer Science, The University of Edinburgh, UK): This project aims to investigate techniques and develop tools to aid in visualising the behaviour and execution of genetic algorithms. In particular, visualisation facilities are being added to EPCC's RPL2 language and parallel framework for evolutionary computing.

Network Modelling of Fault Patterns on Parallel Systems (Steven Webster, Computer Science and Electronics, The University of Edinburgh, UK): Modelling the formation and evolution of fault populations using a 2D, thin-plate, numerical simulation of crustal deformation and a network model allows researchers in the Department of Geology and Geophysics to model large scale deformation of a material which behaves elastically at low stress levels, but which can rupture in a pattern which depends on heterogeneous material properties. Since simulation times are measured in days on available workstations, this project aims to use high-performance computing to cut the time each simulation takes.

5 Conclusions

Over the last eight years, the Summer Scholarship Programme has required a large investment of EPCC's scarcest resource: the time of its skilled personnel. The return on this investment has been exceptional: our user base has been broadened; we have given many students skills which they would otherwise not have acquired; our staff have been exposed to a range of applications in different domains; and we have produced a steady stream of potential employees from which to do our own recruiting. Most importantly, the

SSP has fostered and strengthened team spirit within EPCC. While there will always be scope for improvement, those of us who have run the Programme feel it has been one of the most worthwhile projects we have ever been involved in.

References

[1] L. A. Bowman, N. B. MacDonald and M. P. Taylor, "The Exploitation of Parallel Computing in Major UK Companies" in G. R. Joubert, D. Trystram, F. J. Peters and D. J. Evans (editors), *Parallel Computing: Trends and Applications*, pp. 57–63, Elsevier Science B. V., 1994.

[2] U. Dahlèn and N. B. MacDonald, "Scheme-Linda" in *Proceedings of EUROPAL Workshop on High Performance and Parallel Computing in Lisp*, 1990.

[3] J. R. Henderson, C. MacLean, I. G. Main and M. G. Norman, "A fracture-mechanical cellular-automaton model of seismicity", *Pure App. Geophys.*, 1994.

[4] M. Allen, "Parallel Mesh Decomposition and Generation", EPCC-SS93-1, 1993.

[5] G. Anderson, "Coupled Particle Systems and Distributed Visualisation", EPCC-SS93-2, 1993.

[6] J. Blott and J. Fraser, "Implementing and Using a Parallel Tree Searching Utility", EPCC-SS93-3, 1993.

[7] K. Cameron, "Efficient Backward Ray-Tracing by Probabilistic Photon Tracing", EPCC-SS93-4, 1993.

[8] A. Frey, "Parallel Data Mining of Retail Sales Data", EPCC-SS93-5, 1993.

[9] N. Haring, "Implementation of Kohonen and Cascade-Correlation Neural Network Algorithms", EPCC-SS93-6, 1993.

[10] T. Hely, "Computational Micromagnetics", EPCC-SS93-7, 1993.

[11] A. Leonard, "Numerical Simulation of Galaxy Formation", EPCC-SS93-8, 1993.

[12] B. M. M. McInnes, "Parallel Multi-Grid Solutions", EPCC-SS93-9, 1993.

[13] A. Meyster, "Replication Strategies and Update Protocols for the Shared Data-Object Model", EPCC-SS93-10, 1993.

[14] A. Morris, "Reconstruction and Visualisation of Images from Blurred Data", EPCC-SS93-11, 1993.

[15] A. Rambaud "Ant Farming", EPCC-SS93-12, 1993.

[16] V. Reichenberger, "Multidimensional Nonlinear System Modelling and Prediction", EPCC-SS93-13, 1993.

[17] M. D. Westhead, "ChaOs: A Model for Granular Synthesis by means of Cellular Automata", EPCC-SS93-14, 1993.

[18] K.-J. Wierenga and N. Tomov, "Application Engineering Tools for CHIMP and PUL", EPCC-SS93-15, 1993.