UNIVERSITY OF BIRMINGHAM

BEAR
BIRMINGHAM ENVIRONMENT FOR ACADEMIC RESEARCH

RESEARCH SOFTWARE GROUP

BEAR - Advanced Research Computing

bear-software@contacts.bham.ac.uk | @uob_bear
www.birmingham.ac.uk/bear-software

2019 Report

Contents

Introduction

The Research Software Group (RSG) is part of Advanced Research Computing at the University of Birmingham. The RSG was formed in November 2017, starting with two people - Simon Branford and Andrew Edmondson. We designed the Advice, Coding, and Coaching model, and began working with researchers with the aim of improving the research software written and used by the researchers. BEAR Software's mission is summed up in the words of the Software Sustainability Institute: 'better software, better research'.[1]

Over the next year the group grew rapidly and by the end of 2019 the RSG had eight team members. A large part of our work is supporting users of BEAR's high performance computing services BlueBEAR, BEAR Cloud and CaStLeS.

The Research Software Group exists to:

- Enable the University of Birmingham's research community to get the best from their research software
- Provide specialist software engineering advice and support to researchers and research software engineers (RSEs)
- Help to enhance the University's reputation for high quality research
- Help researchers get the most from BEAR services, maximising the return on the University's investment in BEAR

In 2019 we began to design our "RSG 2.0" model, where researchers are able to pay, e.g. from grant funding, for RSE time from the team, and we began the process of forming a pool of RSEs who are available for this work. See the section "RSG 2.0" later for details of this.

[1] https://www.software.ac.uk/resources/publications/better-software-better-research

Advanced Research Computing

Advanced Research Computing (ARC) is dedicated to the provision of high powered computational and related services, collectively known as BEAR, designed to support research at the University of Birmingham. The Team has grown rapidly in recent years following the 2015 Vice Chancellor's Review and in direct response to the demand and specific drivers from our academic community:

- the recognition of the benefits advanced computational techniques can bring to a whole raft of disciplines
- the tsunami of data just waiting to be analysed
- the skills gap and the need to develop capability to enable the exploitation of compute power
- the need for experts to support researchers
- the demands and expectations of funding bodies
- the demand for 'more' from established users of HPC
- the need to provide fit for purpose technology for competitive advantage

The twenty members of ARC are organised in three Groups including the Research Software Group with its focus on fostering good practice, embedding skills and supporting the use of software. Alongside sit our Research Engagement and Architecture Infrastructure and Systems Groups, the former making sure our services are known across campus and listening to the needs of researchers with the latter designing and building award-winning compute and storage platforms.

With over a thousand projects and thousands of researchers currently using one or more BEAR services, adoption continues to grow and keeps the team very busy with diverse challenges, reflecting the breadth of the University's research. This report provides an introduction to the work of the Research Software Group and gives brief case studies to illustrate the benefits partnership with ARC can deliver.

Research Software Engineers

The term “Research Software Engineer” (RSE) was coined in March 2012 at the Collaborations Workshop. For a history of RSEs please see the State of the Nation report from 2017: https://doi.org/10.5281/zenodo.495360

A major recent UKRI report includes a paragraph that clearly highlights the importance and value that RSEs can bring:

> *"Software lies at the heart of many research and innovation activities. It is needed to control the instrumentation to record data. It is used for the complex modelling required to understand cell function, fusion reactions and the climate. It is the enabling technology behind major advances such as decoding the human genome and the discovery of the Higgs boson and it lies at the heart of strategically important technologies such as AI. Software can be a few lines of code written by a single researcher or a major framework developed over decades by dedicated teams of researchers and software engineers. In academia, 92% of UK researchers use research software and 69% report that it is fundamental to their research with software development a research activity in its own right. The near-ubiquity of software means that it is not possible to disentangle the quality of the software from the quality of the research. Unreliable and untested software leads to unreliable results that cannot be trusted."*
>
> [UKRI, "The UK’s research and innovation infrastructure: opportunities to grow our capability", 2019, p125, https://www.ukri.org/files/infrastructure/the-uks-research-and-innovation-infrastructure-opportunities-to-grow-our-capacity-final-low-res/]

In particular, from that quote, the phrase "*Unreliable and untested software leads to unreliable results that cannot be trusted*" is the negative of the famous quote from the Software Sustainability Insttitute (SSI):

> "Better software, better research"
>
> [Carole Goble, 2014, https://www.software.ac.uk/resources/publications/better-software-better-research]

The Team

Dr Andrew Edmondson (Ed)

Group Leader

Known as "Ed", he started his career as a software engineer and team leader at QinetiQ, after completing an MMath at the University of Oxford. He left QinetiQ to complete a BA in Theology at Birmingham Christian College after which he worked part-time as a senior developer at ApplianSys.

With the rest of his time he began a part-time PhD in New Testament Textual Criticism in the Institute for Textual Scholarship and Electronic Editing (ITSEE) at the University of Birmingham supervised by Professor David Parker. The title of his PhD is "An analysis of the coherence-based genealogical method using phylogenetics". He is currently an Honorary Fellow of ITSEE.

In 2016 Ed joined Advanced Research Computing at the University of Birmingham and founded the Research Software Group. He was the Programme Chair of the 2019 UK RSE Conference and is the chair of the Power AI User Group.

Dr Simon Branford

Deputy Leader and Senior RSE

Simon's experience is in computational research and he has provided high performance computing and mathematical expertise on several research projects. Prior to moving to Birmingham, he was a postdoc at the University of Reading in two different areas: evolutionary biology and meteorology.

His postdoc research built on the skills he developed during his degrees. Simon started at the University of Oxford, where he obtained an MMath. After this, he moved to the University of Reading for an MSc in Network Centred Computing with a specialisation in high performance computing; and then a PhD, where he researched the use of hybrid Monte Carlo algorithms for linear algebra problems.

Simon is a certified Carpentries Instructor and teaches on several of the workshops provided through BEAR Training.

Laurence Hurst

Senior RSE

Laurence has 10 years experience as a high performance computing (HPC) system administrator, at both Birmingham and Loughborough universities where he spent a substantial amount of time helping users with their HPC jobs and troubleshooting HPC software problems. He was heavily involved in the procurement of the latest iteration of BlueBEAR, as well as two clusters at Loughborough. Prior to this he was a software developer writing stock management systems for car and tractor dealerships. He has a BSc in Computer Science from the University of Warwick.

Cerys Lewis

RSE (Web)

Cerys has over 18 years of full stack web development experience, using a variety of languages. Prior to becoming a Web RSE for University of Birmingham, she worked for the University of Warwick on their internal student administration application software, "Tabula". Previously to that she had a background in Retail Software developing web applications and websites. She is a graduate from Royal Holloway, University of London with a BA(Hons) in English Literature.

Mike Allaway

Senior RSE (College of Arts and Law)

Mike has over 7 years of experience working in IT within the HE sector. He started in a user support role at Cardiff Metropolitan University in 2012 and moved to Birmingham in 2015 to transition into a software development role at the Birmingham Clinical Trials Unit. He holds a BSc with First-class honours in Business Management and Information Systems from Swansea University, as well as an MSc with Distinction in Computing from Cardiff University. He developed a software product as his MSc dissertation for designing medical forms and sold this to the University of Birmingham in 2016.

Dr James Carpenter

Research Applications Specialist

James studied and worked in the Department of Music at the University of Birmingham for 18 years before moving to the Research Software Group in autumn 2018. He completed a PhD in electroacoustic composition in 2013 and has variously worked as a teacher of composition, computer music and studio-related disciplines in the music departments at the University of Birmingham and University of Bristol. Immediately prior to joining the RSG he was Studio Technician in the Birmingham Dept of Music where part of his role involved the administration of the Mac systems, as well as the development of bespoke solutions for the technical management of large-scale and idiosyncratic multichannel loudspeaker concert systems.

Dr Simon Hartley

Senior RSE

(Centre for Computational Biology)

Simon has a degree in theoretical physics and a PhD in Physics and AI. He joined the Research Software Group in October 2018. His previous position was working at Clinical School at Cambridge University, producing tools for breast and ovarian cancer risk prediction.

Dr Keith Evans

Senior RSE (HPC)

50% University of Birmingham

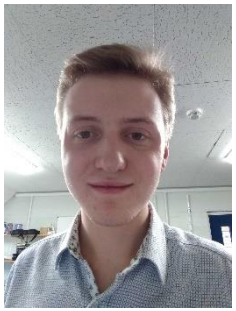
50% Aston University

Keith started his career as an Astrophysicist at the University of Glasgow in 2009 following a Masters in Astrophysics at the University of York. Keith's PhD, entitled "Semi Empirical Modelling of Amorphous mirror coatings for Advanced Gravitational wave detectors", investigated the cause of Brownian and thermoelastic noise in interferometric Gravitational Wave detectors such as Advanced LIGO and GEO600. During his PhD and subsequent Post Doc at the University of Warwick, Keith became more focused on scientific software development for HPC environments. Keith later became formally employed as a Research Software Engineer at the University of Nottingham and aided in the creation of the Nottingham Digital Research Service before taking up the role as "The Fortran Guy" as part of ARC at Birmingham.

James Birch

Research Software Assistant

James is a final year undergraduate at the University of Birmingham, studying towards an MSci in Computer Science. His Masters thesis will be a formal exploration of calculi that support intensional computations. Also, as part of his Masters, James had a year in industry placement at CGI. He first joined ARC in 2016, shortly after completing his first year of study, and has been helping the team wherever possible since. His responsibilities range across: web development, ServiceNow platform development, automated testing, applications builds and producing technical documentation. Notably, he was involved in transitioning the application build process from a completely manual process to using Easybuild.



Jake Foster

Advanced Research Computing Assistant

Jake is an undergraduate at the University of Birmingham, currently completing his industrial placement at ARC. Outside work, he is regularly attending Hackathons, Capture the Flags and coding challenges (ie. Google Hashcode). Since he started at ARC in July he has worked with the Research Software Group, where he was tasked with scraping the application data from the apps documentation pages and building a new website in Django that is updated automatically on application installs. Jake recently moved over to the Architecture, Infrastructure, and Systems group where he is learning about the changes introduced by Redhat 8 so he has the required knowledge when BlueBEAR is upgraded.

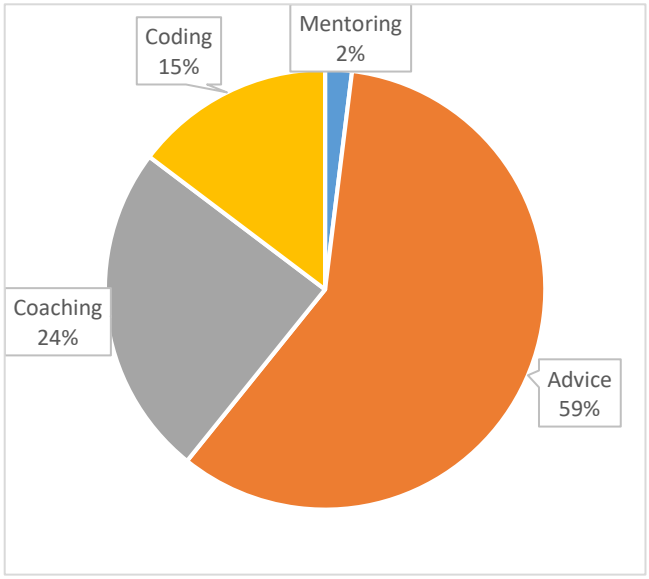
BEAR Software

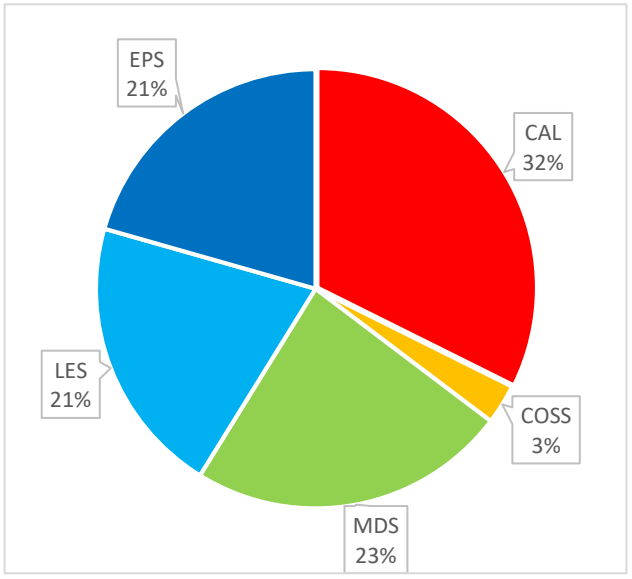
Advice, Coaching, Coding and Mentoring

The RSG offers advice sessions, and short coaching or coding engagements to all researchers in the University, free of charge, under the banner of "BEAR Software". We also offer free mentoring for RSEs across campus. These two charts show the types of session, and how our sessions were distributed between the five colleges in 2019.

The colleges are described at https://www.birmingham.ac.uk/university/colleges, but in brief are:

- Arts and Law (CAL)
- Engineering and Physical Sciences (EPS)
- Life and Environmental Sciences (LES)
- Medical and Dental Sciences (MDS)
- Social Sciences (COSS)





BEAR Software Advice

If a researcher either has need for new research software or wishes to improve their existing software, then BEAR Software RSEs can be engaged to talk through the needs of the researcher and offer advice on how best to achieve those needs.

BEAR Software RSEs are able to offer advice on the whole life-cycle of research software. Example topics include:

- Specifying the requirements for a new piece of software or for additions to an existing piece of software.
- Designing and architecting the software.
- Writing the software and suggestions on which programming languages and tool-kits are best suited for the software being written.
- Documenting code effectively so that when a researcher returns to it at a later date it is easy to understand, and so that it could be made available for others to edit if desired (e.g. Open Source).
- Writing user documentation so that installing, and using (including expected input and output of the software) the software is an easier process.
- Producing or using a testing framework or infrastructure for the software, so that a researcher can easily spot if changes or additions to part of the software, or any third party software it relies on, breaks the existing functionality of the software.
- Porting the software to a new platform or system and how to approach optimisation of the software.
- Upgrading the software, and any third party software that may be in use, and how to ensure that the upgrades have not broken the functionality of the software.
- How to manage the release of the software, so that it is available, in an effective and useful way, for use by others in the research community and beyond.
- Integrating a researcher's software with third party software and libraries.
- Using version control to allow a researcher to manage, and track, changes to their software, data, papers, talks, etc.
- Help with specifying the research software requirements of grant applications and specific assistance with the Technical Appendix.
- Provide recommendations about software licenses and licensing issues.
- Advise on available training courses - those available locally, online, and further afield. We can also discuss tailoring existing training courses to meet the specific needs of a research group.
- General discussion about research software or research computing.

By following the BEAR Software Advice a researcher should be able to improve the reliability and maintainability of the research software they write/use and this will help the researcher with the reproducibility and robustness of their research.

Advice Case Study – Single Cell RNA Sequencing

Dr Rebecca Drummond, Institute of Immunology and Immunotherapy

We were contacted by Birmingham Fellow Dr Rebecca Drummond. Rebecca had just started and was trying to analyse RNA data from a single cell sample. Sequencing from a single cell uses optimized next generation sequencing (NGS) technologies, which provides a higher resolution of cellular differences and a better understanding of the function of an individual cell in the context of its microenvironment. These single cell sequencing techniques require a pipeline of software tools to perform their analysis. Through our help Rebecca was able to learn to use and adapt the sequencing pipeline for their research.

"Simon was very patient and explained Linux commands and accessing BlueBEAR in a way that I could easily understand (I previously had no experience of either of these things) - he was also very available for help and questions whenever I had them. As a result, I got my data in a couple of weeks and I now feel confident using Linux and submitting jobs to BlueBEAR."

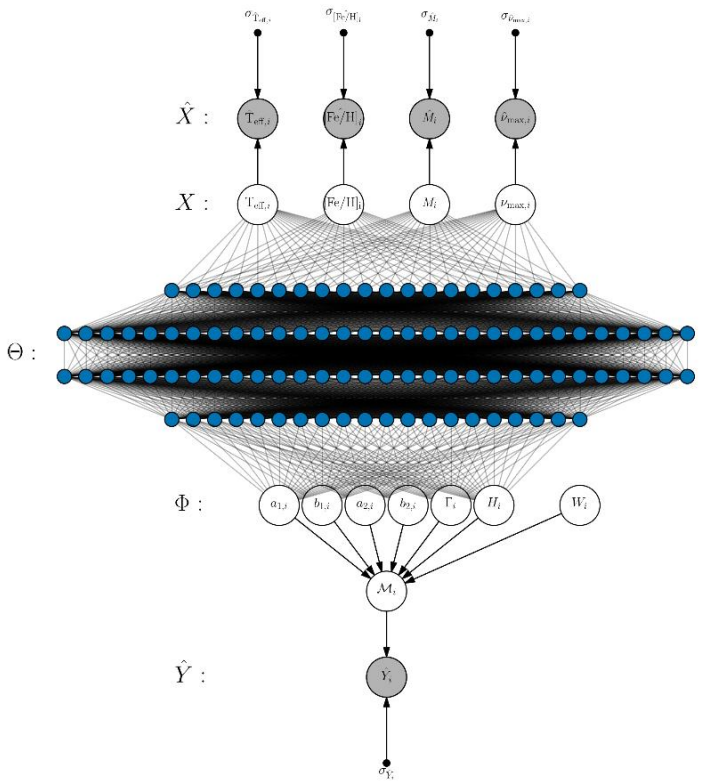
Advice Case Study - Understanding our place in the Galaxy

Alexandra Dixon and Dr Guy Davies, Physics and Astronomy

The group's research focuses on understanding our place in the Galaxy. By using stars as tracers of galactic evolution and planet formation the group works to understand our solar system, the occurrence rate of other planetary systems, and the Milky Way itself. Stellar properties can be inferred by using Bayesian neural networks built on top of rigorous statistical methods. The group used BEAR resources to train a Bayesian Neural Network to learn and predict stellar properties using photometric data from NASA's Kepler Space Telescope.

The Research Software Group assisted Guy Davies and his doctoral student Alexandra Dixon in migrating their machine-learning workflows to using the BlueBEAR POWER9 nodes by improving the POWER9 applications offering and providing general assistance around running codes on a non-Intel based architecture.

Technologies: IBM POWER9 Compute Nodes, Python 3, Theano, Virtual Environments



"We are incredibly grateful to the BlueBEAR team for allowing us to use the POWER9 nodes. They always respond promptly to our queries and have been a great help installing the extras that we need for our work and general advice for the system. Thank you for working with us."

Advice Case Study - Good Practice

Researcher in the College of Engineering and Physical Sciences

We were contacted by a PhD student in the College of Engineering and Physical Sciences who is writing a Matlab-based software package for analysis of DNA sampling data. Although the software was working well, he was looking for advice on good practice to improve his software. We met with the student to review his software.

In general, he showed good practice already; the code was written with plain English variable and function names. For the most part the code was properly functionalised and reasonably commented. There was however no use of version control at any level, nor was there any documentation for the code. Since he was using Matlab to develop code, it was obvious to suggest that he write standard Matlab help pages. These are similar to a standard documentation tool like Doxygen uses to make properly formatted comments within functions. Matlab can then auto-generate help pages and auto-complete calls.

Matlab also has Git functionality with the correct toolboxes installed, however his version did not have these toolboxes. Fortunately his PC had Git for windows and Git gui pre-installed so we gave him a quick tutorial on how to use Git, when to use it and why it improves software development. There was a follow up meeting a week later to discuss any difficulties he had using Git where we gave him a Git cheat sheet, demonstrated a simple pull, checkout, commit, push cycle and highlighted the most important Git functions for day to day use.

Providing this kind of advice on good practice is important and high-impact. We have found that teaching one student in a research group good software engineering practices can lead to them teaching the rest of the group. One group has recently come back to us, some months later, saying they are all using Git and would now like some advanced lessons in integrating it with continuous integration, unit testing etc.

BEAR Software Coaching

If a researcher, or research group, has the need for specific research software expertise, then an RSE can be requested, free of charge, for multiple half-day coaching sessions (max. 10). During these sessions the RSE would be embedded in the research group (physically or virtually) to work with the group to complete specific agreed objectives. The RSG can also provide mentoring to an RSE, to enable the RSE to develop technical and non-technical skills.

This work could follow on from an initial advice meeting where the further need would be identified and clearly defined objectives would be specified and agrees. Example tasks might include:

- Migrating a project to use version control (e.g. gitlab or github).
- Designing and implementing a testing framework.
- Writing user documentation, licensing info etc.
- Testing, debugging, and fixing issues.
- Implementing a new feature.
- Porting the software to a new operating system.
- Designing and implementing a release process.

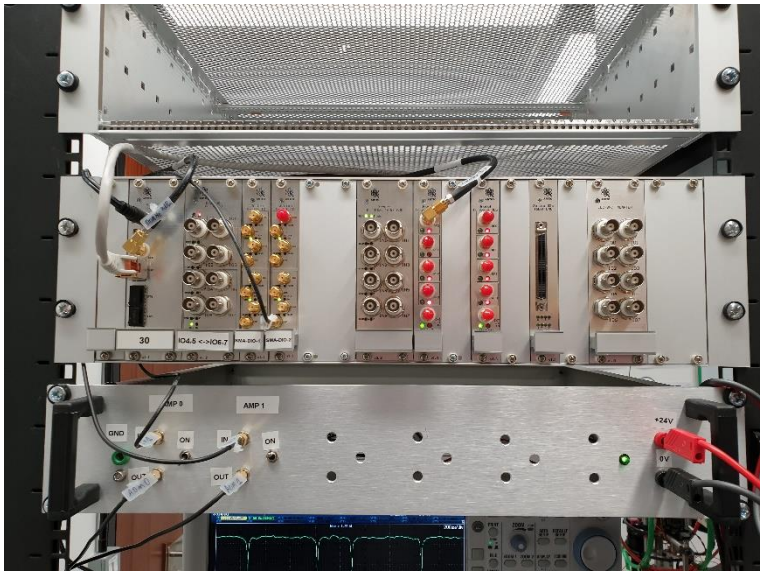
The BEAR RSE will work the researcher(s) to complete these tasks, for example doing pair programming. Our intention is that after the sessions the researcher(s) would be able to continue the work in their own team and have learnt technical, research software skills. Of course, later follow-on engagements are available, demand permitting.

Example of a typical coaching engagement

The BEAR Software RSE would meet with the researcher for a number of half-day Coaching sessions, usually one or two sessions per week. During each of these sessions there would be a task, or tasks, to complete during the session. Between sessions the researcher would work further on the project and this would identify further tasks for the follow on engagements.

Coaching Case Study – Atom Interferometry for Gravity Sensing

Rustin Nourshargh – School of Physics and Astronomy



When cooled to temperatures near absolute zero, atoms unveil their wave-like nature and quantum mechanical laws replace those of classical mechanics. By using finely controlled lasers and magnetic fields, scientists are able to cool small ensembles of atoms down to the lowest temperatures in the universe – just a few millionths of a degree above absolute zero – and thus to manipulate the quantum state of the atoms.

We coached Rustin through the use of Git, continuous integration (CI), and testing in Python. This enabled him to develop a system for controlling the FPGA experimental controllers that are used to run the cold atom experiments. The system has allowed the group to operate their experiments in a far more controllable and reproducible manner. This has improved reliability and reduced the time taken to solve problems with the experiment when they arise.

"Working through the problems together (for example putting a CI pipeline in GitLab) means that not only is the solution in place for me to use, but that in future projects I will have the experience to draw on. Simon took the time to explain things properly. Having prearranged meetings forced me to prepare for the meeting and continue to progress when otherwise it may have stalled."

Coaching Case Study – Scripts, Version Control and Multiple Servers

Researcher in the College of Medical and Dental Sciences

The researcher originally attended a Software Carpentry course delivered by Laurence from the RSG. Part of this course curriculum covers revision control using the Git tool, and afterwards she asked if this tool could be used to help her manage and consolidate tools she uses to provide a pipeline to researchers in China across four different servers.

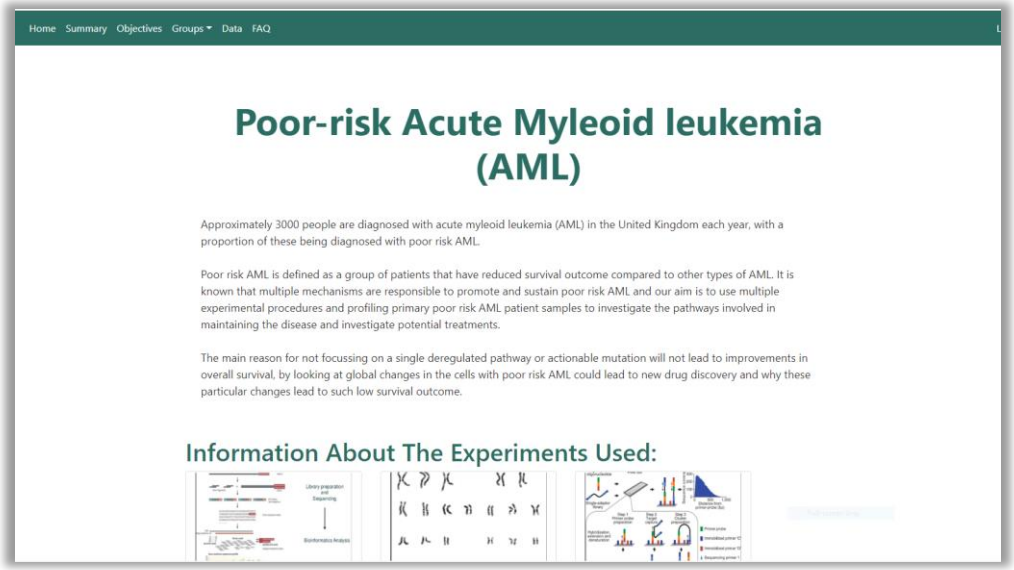
Following this chance encounter regarding a problem unrelated to the content of the course, we arranged a coaching agreement with the researcher to help her explore this and successfully managed to set-up Git to manage her scripts. In the process we merged the slightly different versions of the scripts into a single canonical version, that incorporated all of the bug fixes and corrections that, until that point, were spread out with different servers having different fixes for the same script. We used Ansible to automate deployment and update of these scripts as well as a couple of other management tasks and set-up shared folders, utilising new Unix groups and folder sticky-bits to enforce group ownership and access rights regardless of which researcher created the files.

"The coaching by Laurence has really streamlined the delivery of a complex project for us. It was particularly valuable that he showed me how to consolidate very similar scripts into a single script across all project servers which is configurable based on differences in the projects, really reducing my workload on this task, but also ensuring improved reproducibility and security of the scripts as they are all version-controlled with git!"

Coaching Case Study - Poor Risk AML Website

Dr Christopher Middleton, Institute of Cancer and Genomic Sciences

Approximately 3000 people are diagnosed with acute myleoid leukemia (AML) in the United Kingdom each year, with a proportion of these being diagnosed with poor risk AML. Poor risk AML is defined as a group of patients that have reduced survival outcome compared to other types of AML. It is known that multiple mechanisms are responsible to promote and sustain poor risk AML and our aim is to use multiple experimental procedures and profiling primary poor risk AML patient samples to investigate the pathways involved in maintaining the disease and investigate potential treatments. Christopher Middleton is a research fellow in the Poor Risk AML project. This is a collaboration with numerous sites across the UK including UCL, the Crick Institute and QMUL to integrate multi-omics datasets on a very special set of patients.



Technologies: Python 3, DJANGO, HTML5, Javascript, Git, Data Visualisation

Chris has attended our Software Carpentries course as an introduction to software engineering and to introduce him to the Python programming language. With this introduction he has learnt about best practices, such as version control with Git, as well as the essentials of Python so that he can process and visualise his data. The coaching sessions built upon this to enable Chris to use the Django Python web framework. To date, we have held coaching sessions in the Centre for Computational Biology between May and December 2019, enabling him to show external collaborators the prototype website running on our CaStLeS VMs.

> *"Simon has helped lots during the project, in particular with using the python Django framework to build a prototype website to enable researchers to access project information. He has helped me with the coding logic, which involves incorporating multiple languages (such as python and JavaScript), building the databases and getting the VM set up to run the server."*

Coaching Case Study – Code Review

Dr Ralf Weber, School of Biosciences

This coaching began as a request for a code-review of a particular website's code before deployment on a live virtual server. When we met with Ralf to discuss it further, he expressed a desire for a more general review of a few coding projects out of the dozens his group have produced so they can take that feedback and use it to develop their own, internal, code review processes.

We reviewed one of their Python projects, providing feedback using a "traffic light" system of "Urgent issues", "matters of good practice" and "stylistic advice" - the latter of which may be somewhat subjective. We actively invited Ralf to engage in a discussion with us over anything he disagreed with or wanted clarification as to why we had highlighted an item. We will be reviewing one of his R projects next under this agreement.

"The feedback we received was very constructive and useful. It helped us to further improve our best practices in coding and internal code reviewing processes."

BEAR Software Coding

If a researcher, or research group, has the need for specific research software expertise, then an RSE can be requested, free of charge, for up to ten days of coding time (normally spread over a few weeks).

This work could follow on from an initial advice meeting where the further need would be identified and clearly defined objectives would be specified and agrees. Example tasks might include:

- Implementing a new feature.
- Developing a research website.
- Producing a mobile application.
- Re-implementing an algorithm in a different language, e.g. to run faster or be more portable.

Of course, later follow-on engagements are available, demand permitting.

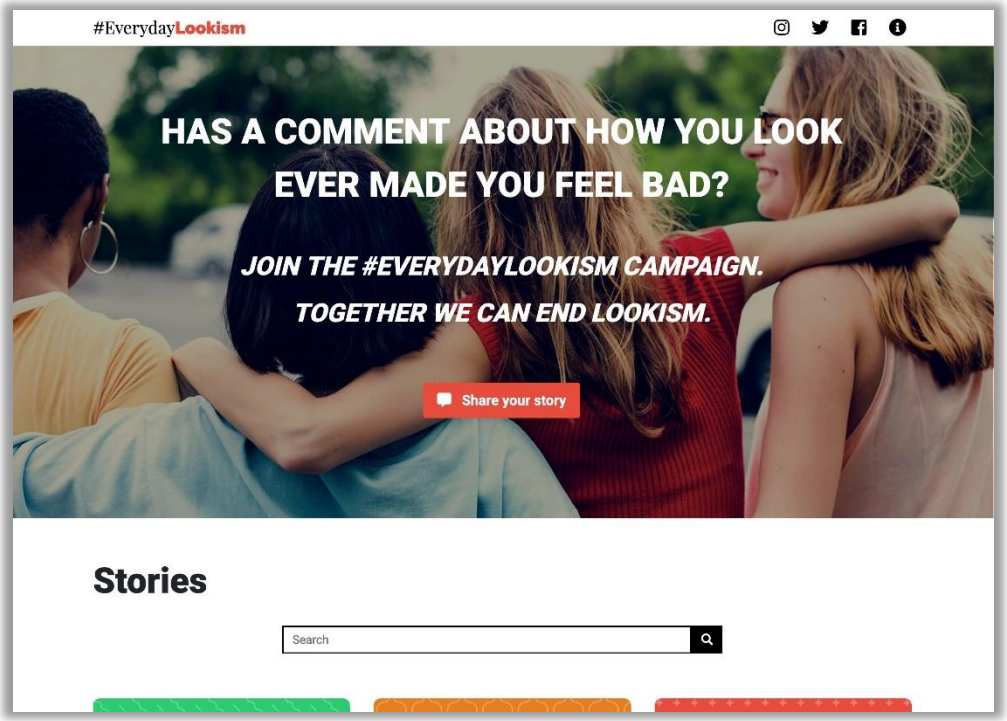
See the “RSG 2.0” section later in this document for news about our new pool of RSEs available for longer, funded coding projects.

Coding Case Study – #EverydayLookism

Professor Heather Widdows, College of Arts and Law

The #EverydayLookism project is collecting anonymous stories from people who have experienced negative comments about how they look. The project had a way for users to submit their stories through the main University of Birmingham website, but felt that having a bespoke website designed specifically to appeal to the their target audience would increase the amount of stories that they would be able to collect. Furthermore, they wanted to be able to automate some of the manual admin processes to save the project time, such as converting submitted stories from plain text to an image to look more attractive and engaging.

Technologies: Django 2, Python 3, HTML5, CSS3, JavaScript, jQuery

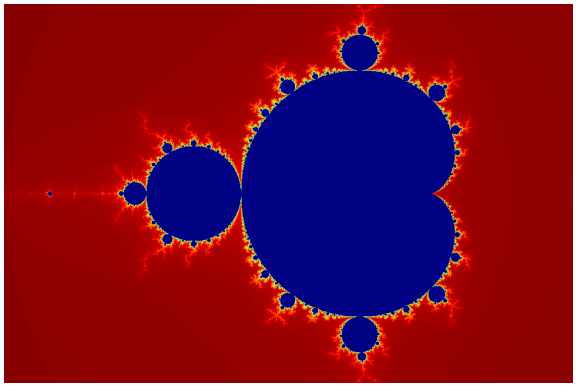


The RSG designed a new, custom website for the #EverydayLookism campaign in just 3 weeks to boost the volume of research data they'll collect throughout the remainder of the project.

The new website https://everydaylookism.bham.ac.uk/ benefits from:

- A simple user experience (UX) that allows users to easily submit their stories
- A modern and fresh interface to display existing stories in a vibrant and clear way
- Integration with social media platforms, to help users share the website and promote the project
- Stories are automatically and dynamically styled by the website, so the project admins no longer have to manually style each story

Coding Case Study – Parallel use of GPUs on POWER9



One of our RSEs, Keith, was given the challenge of experimenting with our new POWER9 nodes, each of which has four NVIDIA V100 GPUs. This is a brief summary of his investigation into accelerating a simple mathematical calculation using multiple GPUs in Fortran using MPI and CUDA.

Keith chose a Mandelbrot set because it is simple to implement, inherently visual and non-trivial to load balance. The algorithm was implemented in a standard fashion, a simple nested double loop over pixels in the *x* and *y* directions. First calculating the value of the seed complex number, then entering a while loop over successive iterations until either the magnitude of the resulting complex number exceeded 2 or a set maximum number of iterations was reached, typically set to 256.

Simple MPI parallelism was achieved by dividing the calculation up by assigning every ***p***th column to the ***n***th process where *n* and *p* are the ranks of each process and the total number of processes respectfully. This method avoids unnecessary communication but leaves the final image spread across multiple processes in a striped fashion. These stripes can then be written out in parallel by all processes using MPI IO routines provided the total number of rows and columns is known and the bit depth is fixed.

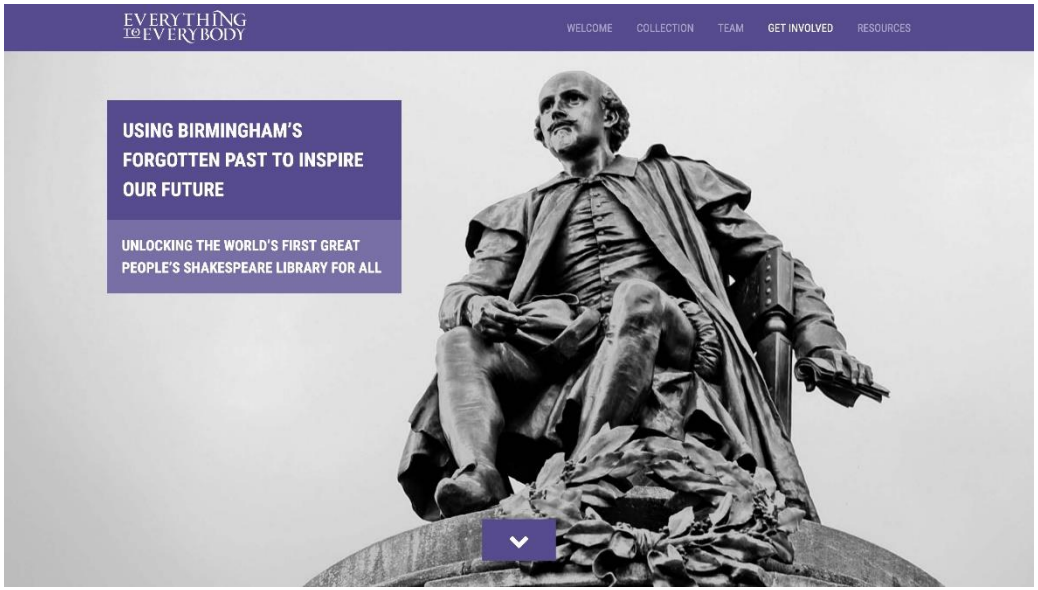
The PGI compiler allows for direct CUDA function within Fortran code, but we decided not to use PGI and explore other options. The solution is write the CUDA kernel as an external C subroutine, compile this subroutine(s) using the Nvidia CUDA Compiler (NVCC) then call this routine from Fortran. There is a bit of ambiguity regarding passing arguments between Fortran and C as there is no difference in syntax between passing by reference or value in Fortran, but this is not a big problem.

With a suitable kernel Keith achieved single process, single GPU acceleration with the expected levels of speed up. This was expanded to multiple GPUs (Multi node, single GPU per node) without modification. Multiple GPU, single node was more difficult as each controlling process needed to choose a different GPU to execute its corresponding kernel. This issue was solved by running one MPI process per GPU, with each process then submitting its kernel to a GPU with the same rank. This was trivially expanded to use 16 GPUs across four nodes. Further additions were made to allow images too large to fit within the memory of a GPU to still be calculated by splitting them up into smaller "sub images" and submitting multiple kernels to calculate each one in turn. Furthermore, each GPU was then assigned multiple processes in order to accelerate the MPI IO. Altogether this was a successful experimentation with Fortran CUDA.

A final note on IO: Due to the use of MPI IO, switching to a single process per GPU resulted in a loss of IO performance because there were fewer processes from which to write in parallel. For comparison, OpenMP was also implemented alongside MPI, while pure MPI vs pure OpenMP performed equally well for computation, OpenMP fell short of MPI IO performance as there doesn't seem to be an analogous parallel IO method for OpenMP.

Coding Case Study – Everything to Everybody

Professor Ewan Fernie, College of Arts and Law



The Everything to Everybody project is a joint project between The University of Birmingham and Birmingham City Council who are collaborating on a £1 million plan to revive the city's almost-forgotten Birmingham Shakespeare Memorial Library, housed in the iconic Library of Birmingham - and the first, oldest and largest Shakespeare collection in any public library in the world. As this project is a collaboration between UoB and BCC, it wasn't desirable to be included on either organisation's website and therefore required a custom designed website of its own.

Technologies: Django 2, Python 3, HTML5, CSS3, JavaScript, jQuery

The RSG designed a new, custom website for the Everything to Everybody project in just 2 weeks.

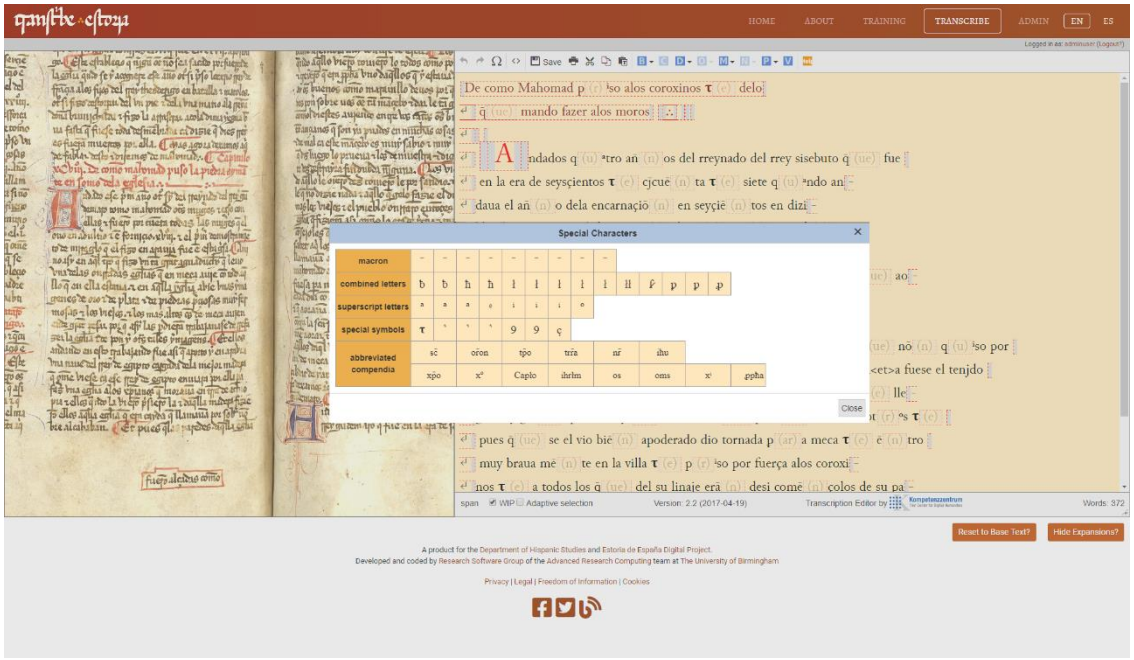
The new website https://everythingtoeverybody.bham.ac.uk benefits from:

- A simple user experience (UX) that allows users to easily find information
- An assortment of images and videos to engage users in the project
- Information about how users can get involved with the project, as well as useful online resources, to help maximise the project's impact

Coding Case Study - Transcribe Estoria

Professor Aengus Ward, College of Arts and Law

The Estoria de Espanna Digital project focused on the construction of the first major digital edition of a sustained text in medieval Spanish prose and provides a digital edition of Alfonso el Sabio's most renowned chronicle. The web application sought to build on this research and its primary aim was to crowdsource participants from all over the world (including outside academia) to be able to contribute to the transcription of these important manuscripts.



Technologies: Django, Python, HTML, CSS, Javascript / JQuery, Bootstrap

Transcribe Estoria has been through 3 coding engagements and is currently available at https://transcribeestoria.bham.ac.uk with over 300 transcribers using the online tool. It is available in both English and Spanish, allows users to independently register as a transcriber, learn how to use the interface with training videos and pdf guides, practice with a sample text and then transcribe from illuminated manuscripts released weekly. The project was designed to be an intuitive interface with a zoom-able pane to allow the viewer to focus on the specific part of the manuscript they were looking at as they went along and giving them the ability to save their progress and return at a later point, with the transcription being stored behind the scenes in TEI XML. The tool also provided the researcher with an admin facility to upload images and manuscript text and gave them the ability to view and sort the recorded data from the transcribers. Features such as expandable special characters, medieval punctuation, decorative features such as rubrication, coloured or larger initials, the option to mark where text was missing / illegible and other marginalia makes this tool particularly useful to transcribers and researchers.

BEAR Support and Applications

In 2019 the RSG handled over two thousand Service Now INC and RTASK service desk tickets, covering a broad range of areas including BEAR Project admin requests, application installations, licensing queries, HPC-specific queries and other general support tickets.

Continual improvements to the *BEAR Admin* website have streamlined the processing of common tickets in the RSG (and Advanced Research Computing more broadly) and have helped to minimise human-error in many tasks.

BEAR – application installations overview

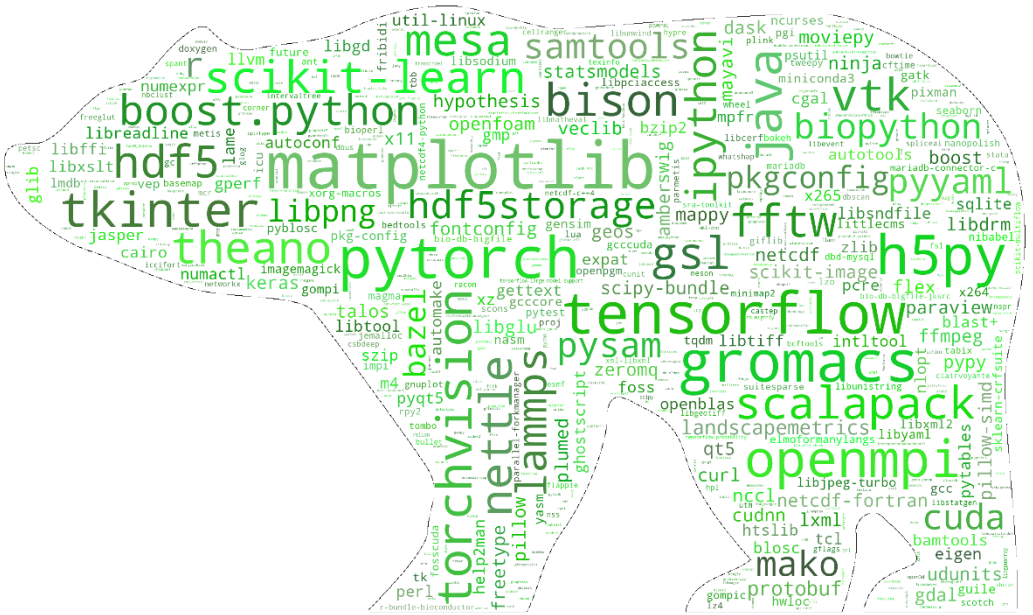
Following previous annual trends, the number of application installation request tickets received via Service Now for BlueBEAR and CaStLeS has continued to increase and totalled **196** (submitted via the approved "ServiceNow" route). These applications frequently require several extra dependencies to be installed.

Headline statistics

- Applications built (by name only): **609**
- Applications built (by module, therefore counting versions and multiple toolchains, e.g. foss, fosscuda etc.): **1238**

Applications built

The range of applications is broad – this word cloud shows all of the applications built during 2019, with the size of the font proportional to the number of modules for each given application name. (The colours are for aesthetic purposes only!)

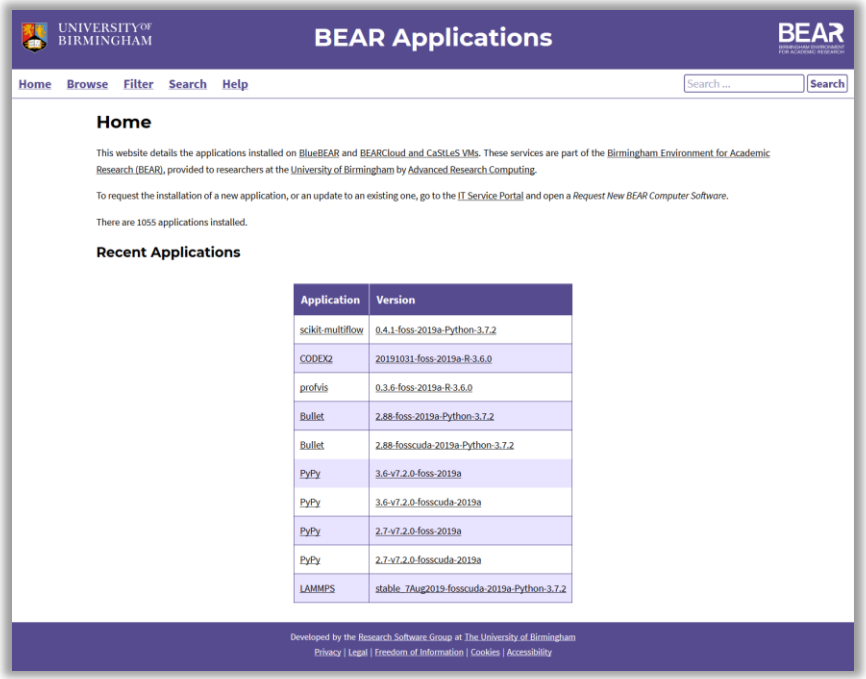


BEAR Applications Website

During 2019 it became clear that we required a new solution for providing the documentation for the applications we have installed on the BEAR system (BlueBEAR, CaStLeS Cloud VMs, and BEAR Cloud VMs). We receive many requests for new and updated applications each week and the time taken to manually generate the documentation was not sustainable or an efficient use of the RSG's time. This made it clear that we required a system that could automatically generate the documentation as part of the application install process.

The new BEAR Applications website was launched in October 2019 and is written in Django, with the development process following the good practices espoused by the RSG. The new website also allows us to easily and flexibly add extra information about the installed applications. Several members of the RSG contributed to the various stages of this project that delivers a website which better meets the needs of those who use the BEAR systems. The previous website contained years of accumulated knowledge and it was important part of the process to combine this information into the automatically generated documentation.

https://bear-apps.bham.ac.uk



EasyBuild and BEAR AI

In late 2018 ARC deployed the largest IBM POWER9 AI cluster in the UK (https://intranet.birmingham.ac.uk/bear-ai) and helped form the Power AI User Group (https://www.poweraiug.org) of which Ed is the current Chair.

The Second Power AI User Group Meeting was hosted at the University in July, with Simon B. presenting on “EmPOWERing Software: Porting Code to Run on our Power AI Cluster” and Ed presenting on “Building scientific software on POWER, what help is available?” The Third Power AI User Group Meeting was co-located with CIUK in December with Ed presenting on "POWER9 at Birmingham – TensorFlow 2 and more".

An important part of our IBM POWER9 AI cluster is the software. We use EasyBuild to script the installation of the research software across the BEAR compute systems. Several members of the RSG use EasyBuild regularly to build the software available on the BEAR compute systems. In 2019 we have led the work to allow for easy installation of software on different processor architectures using EasyBuild, which has allowed us to install software seamlessly across our heterogeneous cluster. We have also contributed a number of “easyconfigs”, the scripts detailing the information required to install a specific version of an application, to the community. The contributed easyconfigs include several versions of PyTorch (1.0.1, 1.2.0, and 1.3.1 for both CPU and GPU systems), OpenBLAS, NCO, a new version of R (3.6.2), and number of Python packages.

The EasyBuild v4.1.0 release notes said this version included “improved support for installing software on Linux/POWER systems, thanks to the contributions by the BEAR team at the University of Birmingham;”

While bug-free software is the ideal, we do find faults with software. These faults can become evident either during installation or through our users making use of the software. We have worked with the developers of several pieces of software to provide comprehensive bug reports enabling the faults to be fixed. Examples include working with the EasyBuild community to debug a SciPy/NumPy/OpenBLAS issue; various faults shown with specific hardware and software setups revealed while running the PyTorch test suite all reported to the PyTorch developers and fixed in more recent versions; and liaising with the CMake developers to debug an issue with different options available in different linker versions.

Some of the software we wish to evaluate has specific calls to SSE instructions[2] to use special features of X86 CPUs. To enable running this software on POWER9, in support of pipelines which make use of the available GPUs, we have contributed to the veclib project (https://github.com/IvantheDugtrio/veclib). This provides a port of many of the SSE functions to equivalent Altivec ones.

[2] SSE is Streaming SIMD Extensions, designed by Intel.

RSEs Dedicated to Academic Areas

Mike Allaway, College of Arts and Law

As well as delivering the standard advice sessions and coaching /coding engagements, Mike contributes significantly to the technical aspects of grant applications, resolves general enquiries relating to research software, and contributes to technical aspects of the College's strategy.

Mike's 15 advice sessions have spanned all 5 of the College's schools. There were 8 coaching engagements covering topics such as fundamentals of programming in Python, development of chat bots, automated emailing in Python, data analysis in Python, version control with Git, and fundamental principles of machine learning. There were 8 coding engagements, all web applications that offer a broad range of functionality, such as collecting research data, presenting research data, promoting the work of research projects, and helping users to engage with research projects.

"An excellent job all round. I am very happy with the solution found to the problem presented by an outdated server" - Professor Francis Lough, Department of Modern Languages

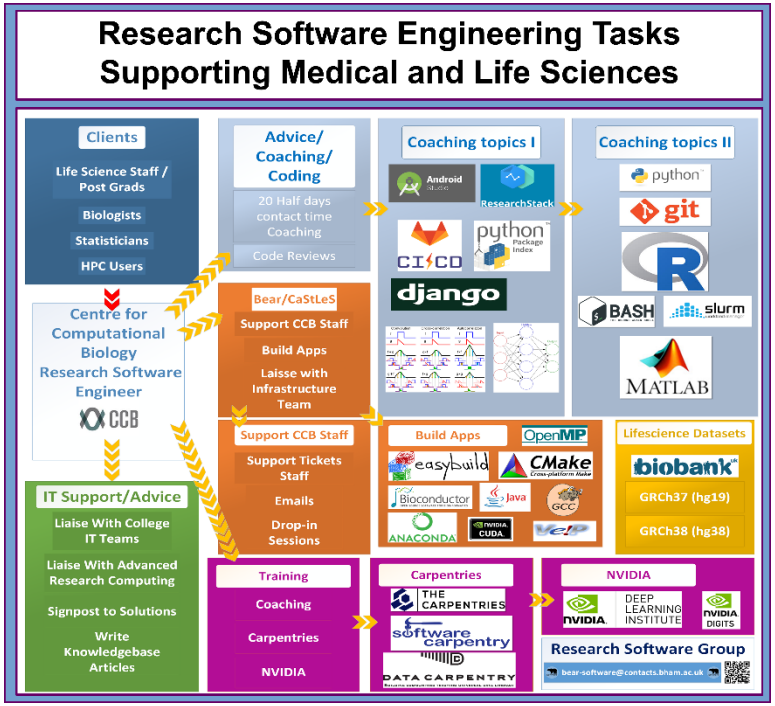
"Excellent product. Mike created an excellent website: beautiful design, easy to use and navigate and within a short time frame." - Ruth Millington, 'Everything to Everybody' Project Manager

"I do communicate a lot with Mike. He is always exceptionally responsive, competent and polite. Mike is amazing support and a great person to work with." - Dr Petar Milin, Department of Modern Languages

"The website looks beautiful and works brilliantly - Mike did an excellent job and in the timescale agreed." - Professor Sara Jones, Department of Modern Languages

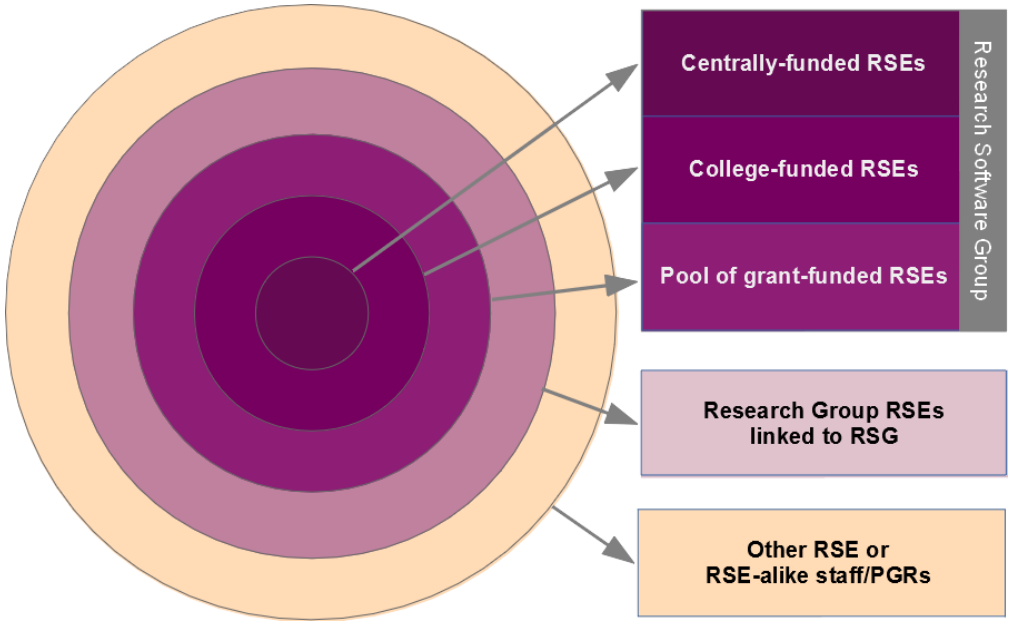
Dr Simon Hartley, Centre for Computational Biology

Few fields have had the explosive change and adoption of big data, AI and HPC systems as much as medical and life science research. The challenges faced by researchers in this field are varied, for example: scripting and programming tasks for file format conversions; data intensive tasks such as annotations of mouse embryo gene-expression databases; computational intensive tasks such as whole genome sequence analysis; and emerging AI fields such as computer vision for MRI scans. This diagram shows the variety of work for an RSE in the life sciences:



RSG 2.0

In 2019 we began the process of expanding our offering to include larger coding projects. The idea was to allow a research group to pay for an RSE for a period of time. This could be a short project (e.g. 0.5 FTE for 3 months) or a long project (e.g. 0.8 FTE for 4 years). To achieve this we envisioned a central pool of RSEs that could satisfy such projects without them leaving the university as the project finished.



The community of RSEs on campus is described by this diagram. Many RSEs will continue to operate in colleges, departments and research groups independently of the RSG. However, there are now three ways for RSEs to become part of the RSG to a greater or lesser extent.

The RSG is comprised of three groups of RSEs. First the centrally funded RSEs who provide support for BEAR's main services and offer the BEAR Software Advice/Coding/Coaching services.

Secondly are the college-funded RSEs who are line-managed in the RSG but work exclusively for the relevant college or academic unit.

Thirdly the new pool of grant-funded RSEs will be available for including in grant applications, and will grow over time as more RSEs are recruited to it. *Please discuss any grant applications with us before including a pool RSE in the proposal.* See below for more information on including a BEAR RSE in grant applications.

Fourthly, some research groups will continue to recruit and manage their own RSEs, but we are now offering the option of a formal "dotted line" to the RSG to assist with managing the RSEs.

Finally some RSEs will continue to work independently of the RSG, meeting up informally at events such as the Academic Programmers Special Interest Group.

Grant Applications

Here we set out three options for including an RSE in a grant application.

Option 1: Independent RSE

Here a research group would include either a named existing member of the research group, or a new anonymous RSE on a grant proposal. If a new anonymous RSE is included then they would then be recruited by the research group on the award of the grant, probably at 1.0 FTE. The new RSE would probably be fixed term (for the life of the grant) and might be academic (governed by the FEC model and thus including the estates charge as well as employment overheads) or academic related (including the employment overheads but without the estates charge).

It is worth noting that advertising an RSE job as an academic post may attract different candidates from an academic-related post.

It can also be valuable to mention the "free" BEAR services in the grant application, such as the BEAR Software services or BlueBEAR, Research Data Store, CaStLeS, etc. These are centrally funded by the university and thus an important university contribution to a research project.

Option 2: Dotted line to RSG

This option is similar to Option 1, but in this case the research group and the RSG could work together to recruit the RSE. The RSE would have a "dotted line" to the RSG, providing some formal line management from the RSG and a direct community of colleagues. These RSEs would be able to participate in certain RSG meetings and have direct access to the RSG's RSEs to ask questions etc.

Option 3: RSG pool of RSEs

If a suitable pool RSE is available for inclusion in a grant application then they could be named on the application. Otherwise a new RSE could be hired by the RSG if a grant is awarded.

This model allows for an RSE's inclusion on a grant for any reasonable amount of time, i.e. 0.1 to 1.0 FTE – or even lower such as "six weeks in June/July 2022". The RSG will manage the availability of its RSEs to meet its commitments.

In the situation where a new RSE needs to be hired and the grant is funding less than 1.0 FTE then the RSG would seek to combine two or more grants to create a single post – or failing that may be able to fund the remainder of the post from its own resources. Please note that this is not guaranteed, but a commitment will be made on a case by case basis.

All RSG pool RSEs are academic related, and so do not attract the estates charge. They are line managed by the RSG, while their tasks are overseen (or perhaps specified) by the project principal investigator(s).

Training

RSEs from the RSG regularly instruct on a variety of training courses offered by ARC. This year we have been primarily involved in three categories of training:

NVIDIA Deep Learning

We regularly run NVIDIA Deep Learning Institute Fundamentals of Computer Vision workshops for researchers, taught by Laurence Hurst - our certified NVIDIA Deep Learning instructor and University Ambassador. This is an instructor led hands-on workshop during which attendees learn the basics of deep learning by training and deploying neural networks. Dr Simon Hartley has also recently been accredited to deliver the NVidia Fundamentals of Accelerated Computing with CUDA C/C++ course as well as the Fundamentals of Computer Vision course, and we will be offering both of these courses in 2020. See https://www.nvidia.com/en-us/deep-learning-ai/education/

BEAR Necessities

The RSG has continued to provide a general introduction to HPC course, entitled *BEAR Necessities*, to researchers at the University. Additionally, over the past year we have branched-out into providing tailored versions of the course to specific areas of the University, for example addressing a group's specific workflow whilst also providing a general level of information. For example, this year we have delivered two BEAR Necessities courses that focused on the IBM POWER9 nodes, one for the Centre for Computational Biology (CCB) and one for a group in Computer Science, whose research is centred on AI-based image processing.

Software Carpentries

The Carpentries builds global capacity in essential data and computational skills for conducting efficient, open, and reproducible research. As part of the University's membership of Carpentries, we have trained ten new instructors over the past 2 years - four from the RSG and six from across campus. This year we continued to run R and Python Carpentries courses, and piloted MATLAB Carpentries. See https://carpentries.org/

"The online material is brilliant and very easy to follow. The unix shell and Git had excellent pacing. There was an appropriate amount of time to practice concepts."

"I really liked starting with Unix; as a novice this made MATLAB more understandable. This is a fantastic tutorial which provides a great foundation for programming."

RSEConUK 2019

On the 17th to 19th September, the RSE Community was reunited at the Fourth Conference of Research Software Engineering at the University of Birmingham. As Programme Chair, Ed feels a great deal of ownership of this year's conference, and is proud of what we were able to deliver as a committee. This year three members of ARC were on the committee:

- Andrew Edmondson was the Programme Chair, and co-chair of the conference
- Keith Evans was the Volunteer Chair
- Debbie Carter (Research Engagement Group) was the Publicity Chair

Two members of the RSG presented at the conference.

- Simon B. presented a talk "BEAR Software – Coaching: Teaching Researchers to Fish"
- Simon H. presented a poster "Supporting CaStLeS in the Cloud"

The conference saw three fabulous invited speakers. Our first keynote was given by Prof Andy Stanford-Clark, Chief Technology Officer for IBM in UK and Ireland. Andy is an IBM Distinguished Engineer and Master Inventor with more than 40 patents. Andy spoke about the Internet of Things, AI and Quantum computing. The second keynote was delivered by Dr Ben Goldacre, best-selling author, broadcaster, campaigner, medical doctor and academic. Ben specialises in unpicking the misuse of science and statistics by journalists, politicians, quacks, drug companies, and more. Our after dinner speaker was Dr Maggie Aderin-Pocock. Maggie has spent much of her career making novel, bespoke instrumentation ranging from hand held land mine detectors to an optical subsystem for the James Webb Space Telescope. Among her many TV and radio appearances, she is currently presenting Sky at Night on BBC 4, Mini Stargazing for Cbeebies and is a panellist on Sky One's successful science quiz show Ducks Quack Don't Echo. Her latest book is "The Sky at Night's Book of the Moon".

We also had over a hundred speakers delivering talks, walkthroughs, panels and workshops in four parallel streams over three days. We had a great poster competition, with 18 entries.

The conference was attended by over 360 people, and has received excellent feedback. See https://rseconuk2019.sched.com for the programme and to download slides for all the talks.

Community

University of Birmingham

In conjunction with the Research Engagement Group, we are active participants in:

Special interest groups

Bioinformatics SIG, CFD SIG, MATLAB SIG, Academic Programmers SIG (Ed is co-chair)

Communities and Conferences

The Hacker Within – most of the team attend these events, and we have presented four times in 2019:

- Laurence: Teaching C++ as a first language
- Ed: "Code and the future of the university" (in collaboration with Dr Matthew Brett)
- Keith: “What on Earth is Parallelism” – writing parallel code for High Performance Computing
- James C.: iTerm2 (as part of a session on 'favourite tools')

Laurence spoke at the fourth Digital Research Conversation - on the "Importance of Data for Deep Learning".

BEAR Challenge – Advanced Research Computing ran a three-day workshop for undergraduate students on HPC. James C., Laurence, Keith, and Simon H. set challenges concerning the use of HPC.

External Conferences[3]

- James Carpenter attended the 4th EasyBuild User Meeting at Louvain-la-Neuve in Belgium.
- Ed gave the keynote talk at the IBM "Innovation Technologies Workshop: Blockchain, Cloud, AI and Security" day (jointly with Simon Thompson).
- Cerys and Mike attended SSI's Collaborations Workshop 2019
- Cerys and Mike attended DjangoCon Europe 2019
- Keith attended the 32nd VI-HPS Tuning Workshop Performance Analysis course
- Simon B. attended the Aspiring RSE Leaders workshop
- Ed spoke at the KQ Codes Tech Social event in London: “Software, theology, high performance computing, AI and RSEConUK 2019”
- Keith attended the 2nd Annual Workshop on OpenMP in Edinburgh
- Ed and Simon H. spoke at the OCF AI and Quantum Symposium. Ed on "BEAR AI - the largest IBM POWER9 AI cluster in the UK" and Simon H. on "Research Software Engineering Tasks Supporting Medical and Life Sciences".
- Ed, Keith, James C., and Jake attended CIUK with Jake presenting a poster describing the opportunities for students in ARC, and his work on the new bear-apps website.

[3] RSEConUK 2019 and the Power AI User Group are covered elsewhere in this report.

Feedback

We often request feedback from those we provide BEAR Software Advice, Coaching, or Coding. This feedback is invaluable in shaping the service we provide to researchers at the university.

A consistent message we see in the feedback is that researchers are happy to, and do, recommend our services to others at the University. We also see many asking for further engagements with us, as their first engagement has proved so valuable. The feedback often contains praise for the friendly, professional, and clear way the RSE has worked.

"Service was great - will be recommending to others in my department who need help with their analyses and computing needs." - Rebecca Drummond, Birmingham Fellow

"This is an absolutely fantastic provision which has been tremendously helpful for my research. Will happily preach it's virtues to any audience!" - Jeremy Kidwell, Senior Lecturer in Theological Ethics

"Very knowledgeable coach. The sessions were extremely helpful. I would highly recommend coaching to others." - Academic in the School of Computer Science

"An excellent job all round." - Frank Lough, Dept of Modern Languages

Contact Us

BEAR Software

For help with anything related to research software at the University of Birmingham please email bear-software@contacts.bham.ac.uk

For information about BEAR Software, see https://www.birmingham.ac.uk/bear-software

BEAR / Advanced Research Computing

For general HPC enquires (including BEAR Cloud), contact the BEAR team by email at bearinfo@contacts.bham.ac.uk

Follow the Advanced Research Computing's twitter feed @uob_bear at https://twitter.com/uob_bear

For information about BEAR, see https://www.birmingham.ac.uk/bear

Requests, Faults, Complaints

The IT Service Desk is the route to find answers, request items/help or log faults and any complaints online at https://www.itservicedesk.bham.ac.uk/