RESEARCH SOFTWARE GROUP

BEAR - Advanced Research Computing

bear-software@contacts.bham.ac.uk

www.birmingham.ac.uk/bear-software

2020 Report

UNIVERSITY OF BIRMINGHAM | RESEARCH SOFTWARE GROUP

BEAR

BIRMINGHAM ENVIRONMENT FOR ACADEMIC RESEARCH

Contents

Introduction 2
Research Software Engineers 4
The Team 5
Coronavirus 9
BEAR Software 10
BEAR Software Advice 11
BEAR Software Coaching 12
Coaching: Code Review 15
BEAR Software Coding 16
BEAR Support and Applications 22
RSEs Dedicated to Academic Areas 25
RSG 2.0 28
Baskerville 30
Internal BEAR Systems 31
Accessibility 32
Training 33
Community, conferences and meetings 34
Contact Us 35

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 2020 the RSG had seven team members, and two vacancies being interviewed for in January 2021. The RSG work is split into two, roughly equal, halves:

1. Behind the scenes: Supporting users of BEAR's high performance computing services BlueBEAR, BEAR Cloud and CaStLeS
2. Working with researchers: providing advice or coaching to researchers or coding for their research projects

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

2020 saw our first major funded coding engagement, where a member of the RSG worked for 6 months (at 50%) on a research project. 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 the University and listening to the needs of researchers with the latter designing and building award-winning compute and storage platforms.

With over two 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 Institute (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 a 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.

Cerys Lewis

RSE(Web)

Cerys has over 20 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 8 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 (HPC) – Formerly RSE (Centre for Computational Biology)

Simon has degrees in Theoretical Physics, Computer Graphics and Virtual Environments, and a PhD in Physics and Artificial Intelligence. He joined the Research Software Group in October 2018 for the Centre for Computational Biology and has recently undertaken the position of RSE for High Performance Computing. Simon has written software in C++, MATLAB, R, Python, HTML5, JavaScript, MySQL for research projects in Artificial Intelligence, Finite Element Analysis for Computational Fluid Dynamics in Physics, Medical Applications, Interactive Systems and point cloud processing and visualization. He has written software for Medical Devices using rigorous software standards such as ISO 13485 and ISO 14971.

Simon is a certified NVIDIA Deep Learning Institute Instructor as well as a Carpentries Instructor.

Dr Mohammad Afraz Ahmed

Research Applications Specialist

Afraz holds an MSci in theoretical physics and a PhD in computational materials science, both from the University of Birmingham. After his PhD, he worked for several years as a research fellow in the PRISM2 research group in the School of Metallurgy & Materials, developing research software for multiscale materials modelling. He joined the Research Software Group in March 2020.

The following team members left in 2020. We include them here to recognise the great work they did this year in the RSG.

Laurence Hurst – *Senior RSE*

Laurence has over 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 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.

James Birch – *Research Software Assistant*

James graduated from the University of Birmingham in 2020, with an MSci in Computer Science. 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 helped the team until the end of his course. His responsibilities included: 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.

Dr Keith Evans – *Senior RSE (HPC)*

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 an RSE at the University of Nottingham and aided in the creation of the Nottingham Digital Research Service.

Jake Foster – *Advanced Research Computing Assistant*

Jake is an undergraduate at the University of Birmingham and undertook an industrial placement in ARC during the 2019-2020 academic session. Outside work, he regularly attends Hackathons, Capture the Flags and coding challenges (i.e. Google Hashcode). In ARC he initially 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 then moved over to the Architecture, Infrastructure, and Systems team where he learnt about the changes scheduled to be introduced by the RedHat 8 operating system upgrade.

Coronavirus

As might be expected, the Coronavirus pandemic of 2020 affected the RSG significantly. The wider Advanced Research Computing (ARC) team was heavily involved in a number of COVID-19-related projects. The RSG was specifically involved in the following:

In the early days of the first lockdown ARC launched a pilot "Open OnDemand" service, https://portal.bear.bham.ac.uk. This allows researchers to use BlueBEAR, our University supercomputer, via a web browser from their remote locations. This service has proved very popular.

Simon Hartley, along with others from ARC, was instrumental in helping to develop and deliver the UK Coronavirus Cancer Monitoring Project: https://ukcoronaviruscancermonitoring.bham.ac.uk/ (see p17)

One notable effect of the lockdown and subsequent phases of researchers doing significantly more home-working than before, is that demand for our face-to-face (or by Zoom etc.) advice sessions reduced dramatically - with people choosing to communicate and receive advice much more by email than previously. Conversely, demand for research application installations continued to increase and demand for our coaching and coding engagements has continued as expected.

BEAR Software

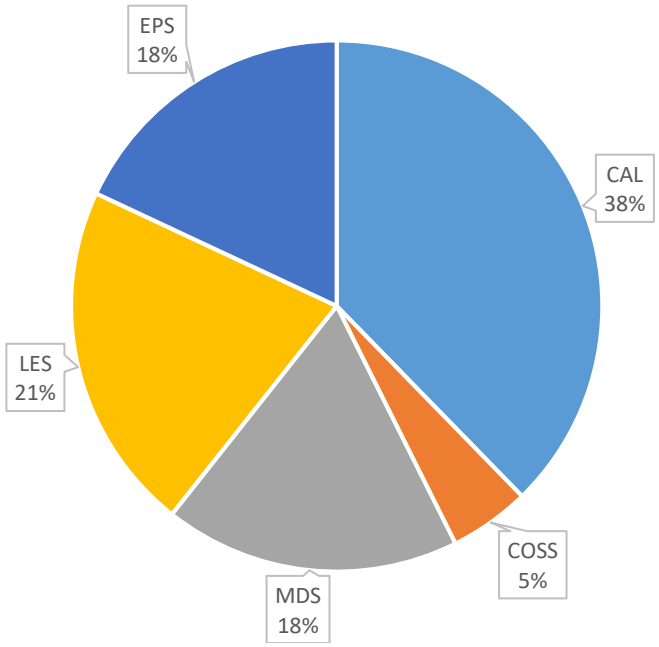
Advice, Coaching, Coding and Mentoring

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. The chart shows how our sessions were distributed between the five colleges in 2020.

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)

The large figure for CAL engagements reflects the fact that CAL fund a dedicated RSE, and implies that there is probably considerable demand in other colleges that we are not yet able to meet.



Advice, Coaching, Coding and Mentoring Engagements, by College

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.

To request Advice please fill in the form at https://intranet.birmingham.ac.uk/bear/sd/bear-software-advice

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 you to complete specific agreed objectives. The RSG can also provide mentoring to a 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 agreed. Example tasks might include:

- Migrating a project to use version control (e.g. GitLab or GitHub).
- Reviewing code.
- 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 with you to complete these tasks, for example doing pair programming. Our intention is that after the sessions you will be able to continue the work in your own team and have learnt technical, research software skills. Of course, later follow-on engagements are available, demand permitting.

To request Coaching please fill in the form at https://intranet.birmingham.ac.uk/bear/sd/bear-software-coaching-coding

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 session per week or fortnight. 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 could identify further tasks for the follow on engagements.

Coaching Case Study – GPU and CUDA Coaching

Anonymous

A PhD student in EPS asked for RSE support to solve memory access errors in their GPU (CUDA) code. The code used managed memory to transparently access memory from both the GPU kernels and host memory. As we started during the initial pandemic lockdown, the coaching all happened remotely via Skype. During testing of the code, we found that it failed with segmentation faults with certain parameter values. The code did not check the results of CUDA calls for error states. Adding these showed there was an illegal memory access reported.

After the student had added checks to all their CUDA calls, we investigated further to identify the underlying memory error. This required system-level changes (including a reboot) to the HPC GPU nodes in order to be able to use the NVIDIA profiler, something which, due to security concerns, is not generally available to our users but our close working relationship with the BlueBEAR sysadmins meant we could enable it for a short amount of time. Debugging the problem was frustrated by CUDA not flushing device-side printf statements, so they are not displayed, if the CUDA kernel fails.

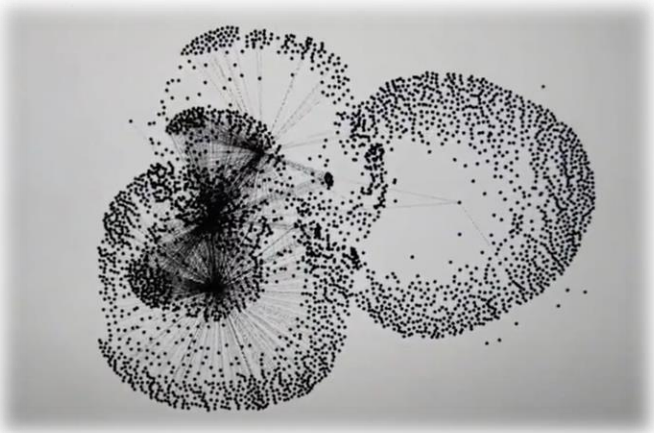
In the end, in order to identify which specific threads were failing, we added a new integer array the size of the number of threads and updated the value at the corresponding thread index to indicate progress through the code, in order to identify which blocks failed and which got all the way to the end. This helped track down the problem to being a mathematical issue in one of two methods which calculated values later used as array indexes, causing the memory access error. The RSE handed the problem back to the researcher at this point to work on the complex and specialised mathematics required.

The student wanted further help to migrate part of the code to use shared memory, to improve performance, which led to a follow-up coaching agreement being arranged.

"Laurence was invaluable in helping to develop the CUDA code I utilize for my research. His ability to debug the code at a deeper level, than I am able on my workstations, was key in identifying nuances in the algorithm that required consideration. Thanks to Laurence's efforts, by reordering the calculations, we were able to reduce the number of calculations per step, thus increasing the speed of the simulation. Ever iterating closer to faster than real-time simulation"

Coaching Case Study – Quakers and the History of Anthropology

Kelvin Beer-Jones (PhD Student), College of Arts and Law



The project aims to design, develop and apply integrated digital analytics and data visualisation tools (a historical data digital toolkit, HDDT) to explore and analyse the ordered data contained in the many catalogues and indexes of national and international collections of 19c manuscripts. The HDDT is used to interrogate biographical and social connectivity data for 3,000 persons who from 1830 to 1870 supported the development of the discipline of anthropology in Britain, and the 700 Quakers amongst them.

Kelvin, the PhD candidate completing this research, approached the Research Software Group to help support the more technical aspects of this project. Prior to this coaching engagement, Kelvin primarily managed his data using Excel spreadsheets. During the course of this engagement, we've worked with Kelvin to transform the way he stores and works with his data, including:

- Storing his data in a relational database, instead of a spreadsheet
- Using Python to automate the import of new data sources from spreadsheets into the relational database
- Using SQL to script his management of the data (e.g. selecting data, inserting new records, updating existing records)
- Analysing the data using Python, including popular packages like Pandas and Matplotlib
- Exporting his data into Gephi, an open-sourced tool for visualising and working with network data
- Using Git and GitHub to manage the version history of his work

The impact of this engagement has led to significant improvements in the quality of the research data, the depth of the research output, and the technical skills and knowledge of the researcher, which will serve him throughout his future research career.

"To begin with my digital skills were very poor, without RSG committed support I would not have undertaken this project. Tailored, regular and intensive support was spread over two years of my research and my skill set, confidence and ability to perform the research I wanted to do has been delivered. Mike became a critical friend who provided encouragement, understanding and advice. He made sure that I acquired skills at an appropriate level, when I needed them and with little fuss. His commitment and interest in my work resulted in highly efficient skills learning and development throughout." - Kelvin Beer-Jones

Coaching: Code Review

One of the goals of the RSG is "Provide specialist software engineering advice and support to researchers and RSEs". An important service we provide to fulfil this goal is to offer code reviews to RSEs throughout the University and researchers who are writing their own code. By doing this, we share and develop good coding practices and in so doing we enhance the University's reputation for high quality research (another goal of the RSG).

We have performed code review for researchers and RSEs across the University, covering a wide range of research topics and programming languages. In 2020 we have code reviewed: Python, R, MATLAB, C++, HTML, CSS, and JavaScript. The group has also discussed how we do code reviews with research groups with the aim of allowing the groups to build up their own culture of reviewing code.

The aim of a code review is to help the researcher refine their code, identify bugs, and to provide clear methods in which the code can be improved. These improvements may be methods to enhance the performance, sustainability, or testing in the code. When a code review is done well, it can improve both the code being reviewed and the future code that is written.

"The "Out Of Our Minds" team [https://outofourminds.bham.ac.uk/] has developed a Python package for deep text modelling [https://github.com/ooominds/Deep_text_modelling/], to make the process of training language learning models, using large corpora, easier. It offers tools for preparing the textual data (e.g., tokenisation, training/validation split), and running biologically inspired and deep learning models (e.g., Widrow-Hoff, Feedforward NNs, LSTM). In the final phase of development, the team has relied on the support from the BEAR Research Software Engineering and Coaching. This support resulted with crucial improvements in robustness, consistency and clarity of the code. Our experience in collaborating with the BEAR team is nothing but excellent. It is a testimony of both the importance of RSE, in general, and the quality of the BEAR support, more specifically. We are looking forward to our future collaboration on our new projects." – Dr Petar Milin, Modern Languages

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 several 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 agreed. Example tasks might include:

- Implementing a new feature.
- Developing a research website.
- Re-implementing an algorithm in a different language.
- Adding parallelisation or CUDA (GPU) support to an application.

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

To request Coding please fill in the form at https://intranet.birmingham.ac.uk/bear/sd/bear-software-coaching-coding

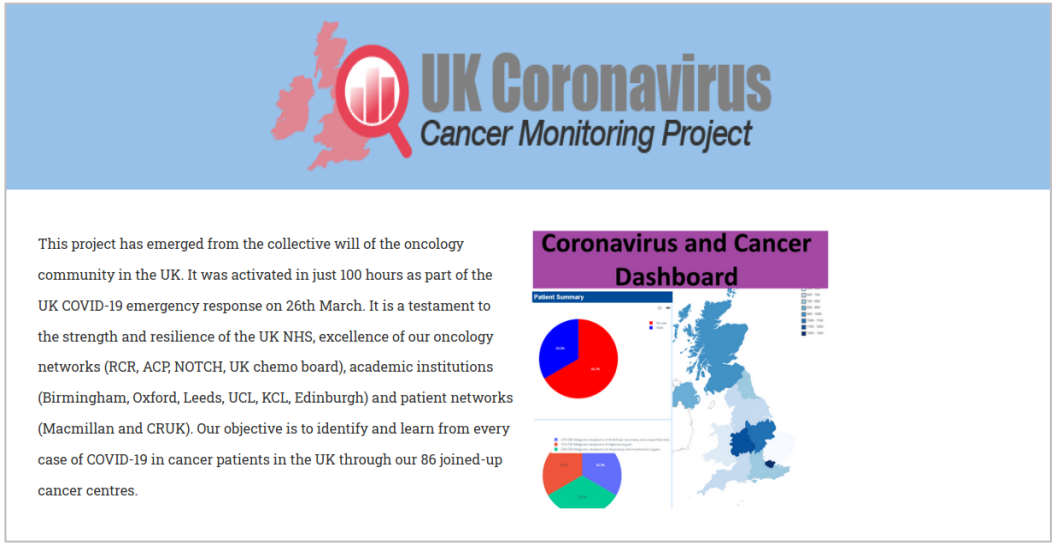
See the “RSG 2.0” section later in this document for news about our pool of RSEs available for longer, funded coding projects.

Coding Case Study – UK Coronavirus Cancer Monitoring Project

Prof Gary Middleton, Institute of Immunology and Immunotherapy and Prof Jean-Baptiste Cazier, Institute of Cancer and Genomic Sciences

The unprecedented Covid-19 pandemic has led to the need to more closely monitor at risk groups, such as those who have cancer. The UKCCM site informs healthcare professionals about the current situation for at risk patients they have in their care.

This project produced a Django website running on a CaStLeS[2] virtual machine that serves a series of Plotly produced JavaScript/HTML graphs from the annotated summary data provided by CCB[3] researchers. The data and visualisations are created and displayed to end users of the website. The website has public information sections and general project information but also has sections for analysed data which are only available to registered healthcare professionals and the project collaboration partners.



The website offers a daily updated, and accessible user interface, as well as fast graphical plots when browsing the analysed data.

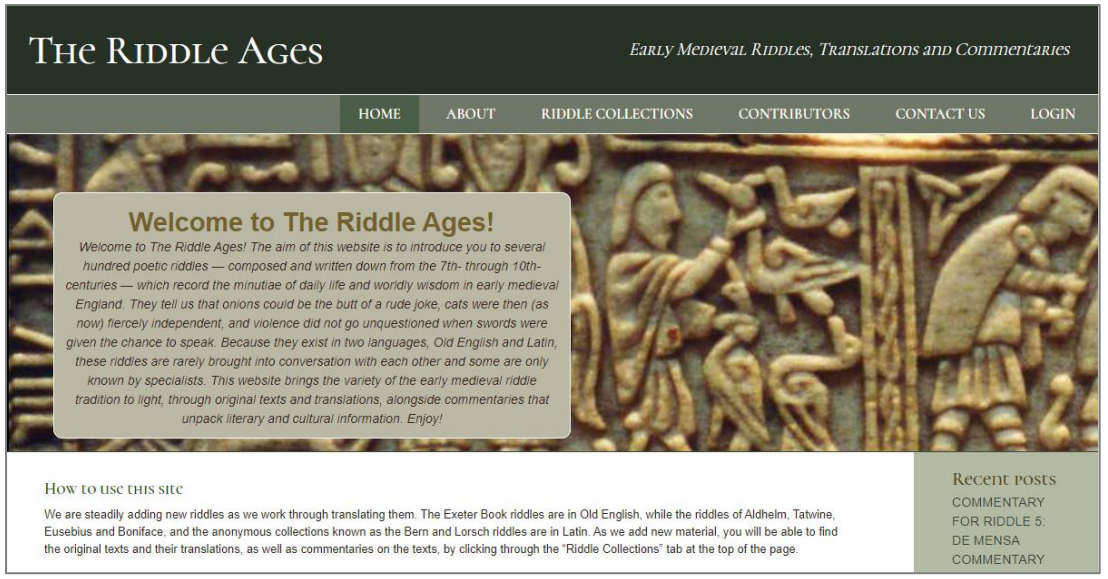
Analysis website: https://ukcoronaviruscancermonitoring.bham.ac.uk/
Redcap data collection website: https://ukcoronaviruscancermonitoring.com/

> *"Running against the progression of the pandemic, Simon, his RSG colleagues and the whole ARC team have supported us enthusiastically and swiftly to create this bespoke interface to near-real time data and analysis from the UKCCMP. The data and results presented interactively now inform oncologists to make the most accurate and up-to-date decision about their patients and their cancer treatments. A definitive tour de force!" – Prof Jean-Baptiste Cazier*

[2] (CaStLeS) Compute and Storage for Life Sciences – https://intranet.birmingham.ac.uk/castles
[3] (CCB) Centre for Computational Biology - https://www.birmingham.ac.uk/ccb

Coding Case Study – The Riddle Ages

Dr Megan Cavell, English Literature



The Riddle Ages project builds on work from the previous WordPress blog, which was active from 2013. The aim of that blog was to provide Modern English translations of all the Old English riddles found in the late 10th-century Exeter Book, along with commentaries on interesting points about each riddle. (Since going live with the new site the researcher and her team have expanded this to include other collections including the Bern riddles which are primary in Latin).

The aim of this project was to redesign and modernise the site allowing it be available on a variety of different devices, better control the content and its display in ways which were meaningful in the context of the transcription and commentary and utilise more modern JQuery interface techniques as well as providing basis for the researcher and her team to be able add future, more interactive, functionality which would allow them to better engage with their visitors.

It was updated to use a fully fluid site design and has been made fully accessible in line with the new guidelines. The site also has a new announcements section to allow the researcher to convey news and highlight areas of interest and an admin area which allows easy addition and editing of posts and riddle translations, uploading of images and updates to individual page text, site titles etc.

You can view the new website at www.theriddleages.com

"Working with the Research Software Group has been invaluable for my AHRC-funded project. Cerys Lewis designed the project website to disseminate hundreds of texts, translations and commentaries, and guided the project team through the use and maintenance of the site. The website launched just last month and the feedback from international colleagues using the website for their own research and teaching has already been incredibly positive" - Dr Megan Cavell

Coding Case Study – Hierarchical DE-MCMC Matlab implementation

Dr Dietmar Heinke, School of Psychology

The aim of this project was to implement a mechanism for creating a user-defined hierarchical Bayesian prior structure for a Differential Evolution Markov Chain Monte Carlo (DE-MCMC) method in MATLAB. Fitting mathematical models to data in cognitive psychology is a challenging goal, especially since the parameters in these models tend to be correlated. DE-MCMC is known for successfully dealing with these correlations. However, as a by-product, DE-MCMC is fairly inefficient. A hierarchical prior structure results in a more efficient parameter estimation (model fitting).

In addition, data often contains hierarchical structures that are well-known in advance. In psychology, this structure typically results from individual differences between people's behaviour. For instance, a group of people can respond faster to events compared to another group. Hierarchical priors allow the modeller to define such known structures within data sets.

The prior structure shown in Figure 1 describes the expected structure for an experiment with two conditions. The individual differences between participants are captured via the hierarchical prior.

During the coding project, a tree-like structure, with a series of interconnected levels and nodes, was developed for controlling the hierarchical priors and probability density functions. The code was designed to allow a user to easily modify the prior structure (e.g. Figure 2) without having to modify large sections of the code.

> *"Afraz implemented for us an excellent user-friendly interface for a complex model fitting method. His contribution forms an excellent basis for future research." - Dr Dietmar Heinke*

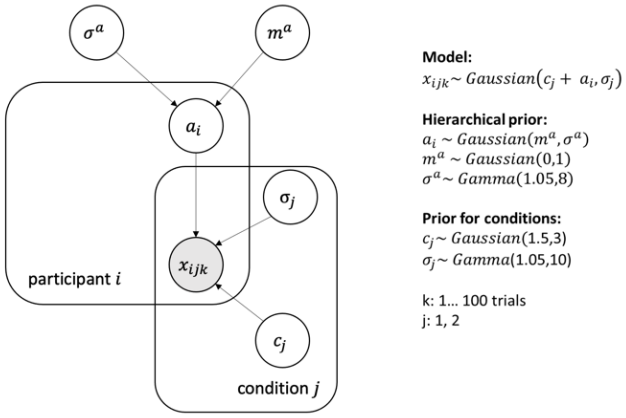


Figure 1: Example prior structure

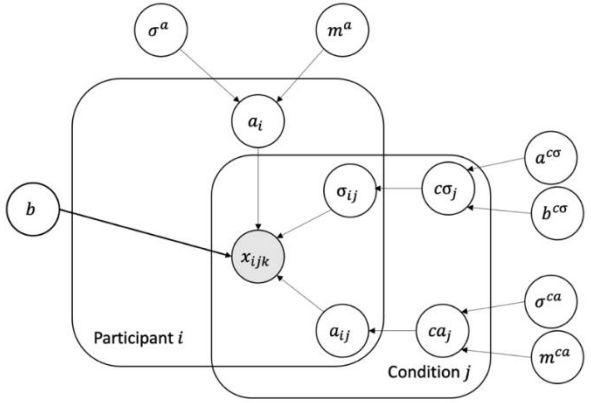
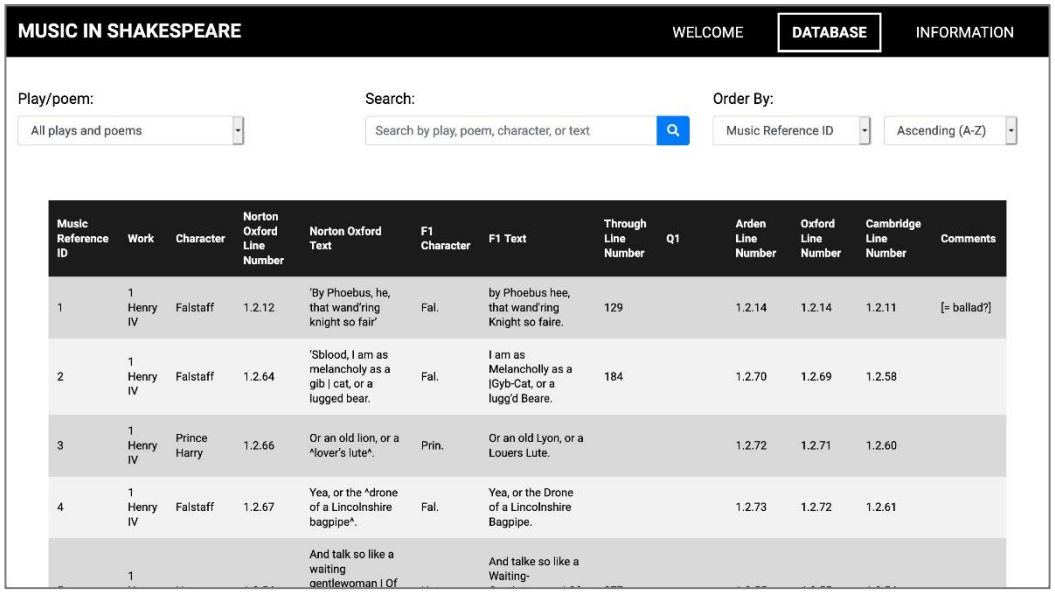


Figure 2: More complicated prior structure

Coding Case Study – Music in Shakespeare

Dr Simon Smith, College of Arts and Law



Music in Shakespeare is a website that attempts to identify every music reference in context in each play and in a number of poems where they occur. Music in Shakespeare is a wide-ranging subject embracing terms and phrases of many differing kinds and definitions. For the purposes of this website, 'music' has been interpreted as relating to sound (instruments, voice and vocalization, noise such as shouting and ordnance, natural phenomenon such as tempests, thunder, wind, etc.), theory and philosophy.

The Music in Shakespeare website was originally developed in 2008 at the University of Hull. The server that was being used at Hull to host this website had to be decommissioned and due to the age of the technologies used to develop the website, it wasn't feasible to port the old website to newer servers. This meant that the website would cease to exist and its research data would no longer be available to the public.

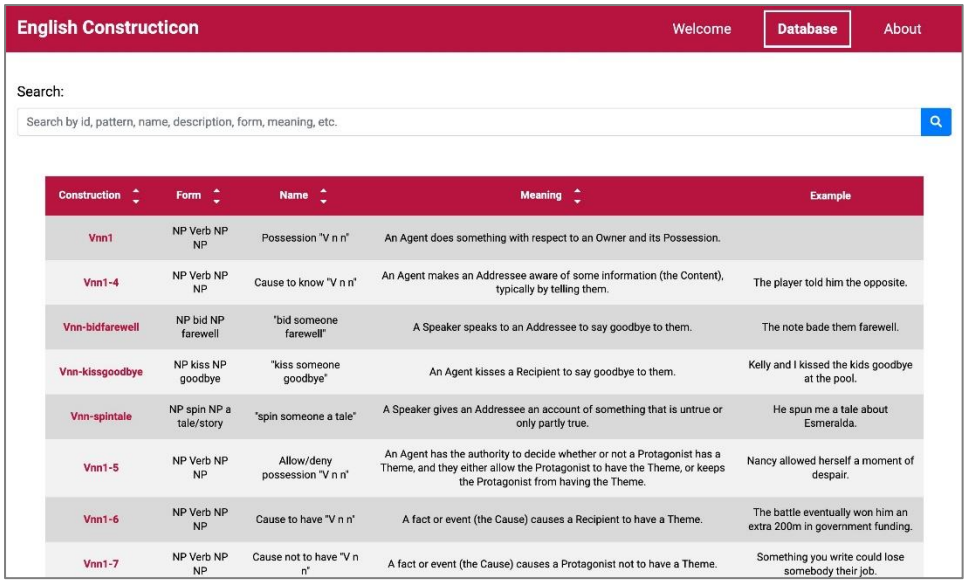
To extend the life of the website and thus the availability of the research data, the RSG redesigned the Music in Shakespeare website from scratch using modern technologies. The new website also benefits from a cleaner and more accessible user interface, as well as a faster user experience when browsing the research data.

You can view the new website at: https://shakespearemusic.bham.ac.uk/

> "Many thanks, Mike, for your wonderful work on this project. I can confirm that we are both delighted with the outcome, and recognise the effort you have gone to under the current unprecedented working conditions to get this finished for us." - Dr Simon Smith

Coding Case Study – English Constructicon

Dr Florent Perek and Dr Amanda Patten, College of Arts and Law



The English Constructicon project will produce a database of grammatical constructions following the principles of construction grammar, which defines constructions as pairing of forms with meaning. The constructions in the database will arise from descriptive research taking the COBUILD Grammar Patterns, a grammatical resource developed in Birmingham, as its basis, combined with information from the FrameNet lexical database.

The researchers chose to store their research data as XML because of their familiarity with the language. For the development of a Django website a SQL database is the typical approach for storing data, so the RSG worked closely with the researchers to build an XML to SQL import tool. This meant that the researchers could work with the format they were most experienced with when collecting data (XML) and through a simple website interface, in just a few clicks, convert this XML data into a SQL database for use by the website. During this process the RSG also advised the research team about ways to enhance the quality of their data when working with XML, such as using an XML schema to enforce a defined data structure.

The main component of this project was the development of a public website and web API to allow users to browse this research data. The website offers a clean, simple, and accessible user interface, as well as a fast user experience when browsing the database.

Website: https://englishconstructicon.bham.ac.uk/ API: https://englishconstructicon.bham.ac.uk/database/api/

"We're very pleased with our project's website. It looks professional, clear, visually appealing, and will be easy for us to maintain. Mike was very attentive to our needs throughout the development, and also made some very useful suggestions of his own. Of particular interest is the possibility for us to continue developing our data in the same format that we're used to, with the XML to SQL conversion tool that Mike developed with the website. The website and database are key to the future of our project, as they allow us to store and search our research data, and share it with a wide academic as well as non-academic audience." - Dr Florent Perek

BEAR Support and Applications

BEAR Support

In 2020 the Research Software Group handled over 2000 Service Now tickets, covering a broad range of areas including BEAR Project admin requests, application installations (see separate section for further information), licensing queries, HPC-specific queries and other general support tickets.

This year saw a major update to the BEAR Admin website (see separate section for further information), which is used in the processing of common tickets in the RSG (and Advanced Research Computing more broadly) and have helped to minimise human-error in many tasks.

Application Installation Requests

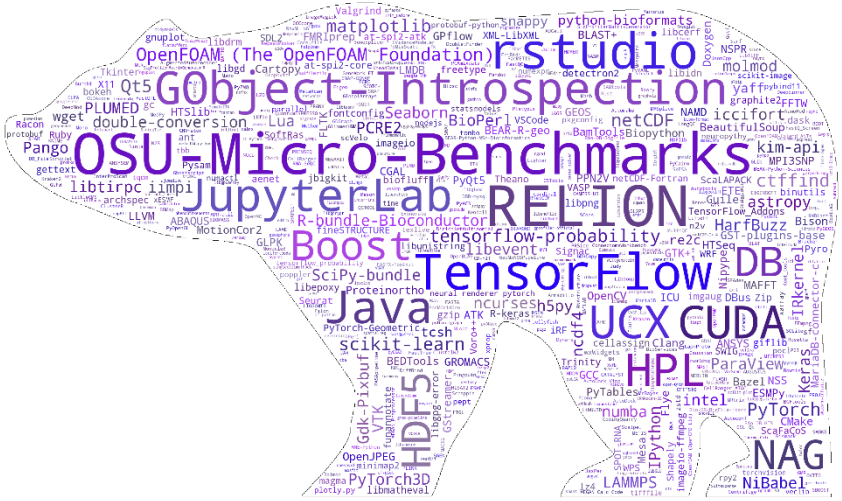
Software application request tickets for BlueBEAR and CaStLeS via Service Now have once again increased significantly on the previous year, jumping by almost 100 to **292**. It should be noted that due to the planned EL8 operating system upgrade (see next page) we actively sought-out users who were running deprecated versions of the BEAR modules and submitted tickets for updated application modules on their behalf. This will have contributed in particular to the increase in tickets during the final 1/3rd of the year.

We built **694** applications (counted by name only), or **1035** counting installed modules, therefore including versions and multiple toolchains.

Whilst the number of applications-by-name built increased significantly from 2019's figure of 609, the overall number of *modules* built decreased from 1238 to 1035. Comparing the lists shows that we generally built across 2 toolchains in 2020 (as opposed to 3 in 2019) but also that in 2019 we built many more modules against Python2, which is something that we actively avoided in more recent toolchains due to Python 2 reaching end-of-life in January 2020.

Applications built

The range of applications is broad – the word cloud shows all of the applications built during 2020, with the size of the font proportional to the number of modules for each given application name.



EasyBuild

In 2020 we continued contributing to the work to allow for easy installation of software on different processor architectures using EasyBuild. This work allows us to install software seamlessly across our heterogeneous cluster. We have also made many other contributions to EasyBuild, and by working together, as a larger community, we are able to install many more applications for use on BlueBEAR and BEAR Cloud. For more information on EasyBuild see https://easybuilders.github.io/

Late in the year, Simon B. was added to the group of people who have merge rights for the EasyBuild GitHub repositories. This allows him to further contribute to the EasyBuild community.

BlueBEAR and BEAR/CaStLeS Cloud Operating System Upgrade

On BlueBEAR we use CentOS and RedHat and on BEAR and CaStLeS Cloud VMs we use a mix of CentOS and Ubuntu. The current versions we use (7 and 16.04 respectively) are nearing the end of life and to continue to keep the systems secure, and in order to provide the latest applications to our users, the operating systems need to be upgraded. The new versions of these operating systems (CentOS 8 and Ubuntu 20.04) include a significant glibc version change, which breaks compatibility for a lot of older software.

In the autumn of 2020, the RSG started the task of installing applications on the new operating systems. As part of the process the oldest of the installed applications were deprecated as the older the installation the harder it is likely to be to re-install it on a new operating system. Where these applications had been used recently, we contacted the users and worked with them to either move to a newer version of the application, migrate to an alternative or, if necessary, install the original version in the new operating system.

Across the various BEAR systems this lead to over six thousand application installs being ready for the BlueBEAR upgrade in December 2020 and for the VM upgrades to follow in the first part of 2021. As we use EasyBuild (see relevant section) this process was largely automated, with the RSG being involved to fix the install problems (a few hundred).

Apps Website

Following on from the introduction of our new applications website (https://bear-apps.bham.ac.uk/) in 2019, this year was a time for smaller improvements - many based off the feedback from those who use the website to find out information about the range of applications available on the BEAR systems.

ReFrame – Automated HPC Testing

An important part of our application installation process is testing that these installs work correctly. We use the ReFrame tool (https://reframe-hpc.readthedocs.org/) to enable us to run thousands of tests each night, which check both the functionality and performance of these applications. These same tests have also played an important part in allowing us to check the new installations of applications that we have been doing for EL8 (see above).

EasyBuild and POWER9

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.

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. In 2020 we have continued to contribute work that allows for seamless install of software on different processor architectures using EasyBuild.

We have collaborated with people from Universities across Europe to make various software easily installable on POWER9 via EasyBuild, and have offered advice to others as far afield as the US.

RSEs Dedicated to Academic Areas

College of Arts and Law

The College of Arts and Law (CAL) funds a dedicated Senior RSE, Mike Allaway, in the Research Software Group.

In keeping with the RSG's service offerings, the main services Mike provides are advice sessions, coaching engagements, and coding engagements. In addition, Mike also contributes significantly to the technical aspects of grant funding applications, resolves general enquiries relating to research software, and contributes to technical aspects of the College's strategy, such as how to utilise software and data to make the College's research more open.

Advice	Coaching	Coding
There have been 2 advice sessions for CAL in 2020 in 2 of the College's 5 schools: School of History and Cultures; and Languages, Cultures, Art History and Music. Both of these advice sessions have led to coaching and coding engagements.	There have been 5 active coaching engagements for CAL in 2020 that have contributed to research across 4 of the College's 5 schools: Birmingham Law School; Philosophy, Theology, and Religion; School of History and Cultures; and Languages, Cultures, Art History and Music. These coaching engagements have covered topics such as fundamentals of programming in Python, relational database design and development with SQL, data analysis in Python, version control with Git, and web development with Django. The researchers who have received this coaching range from PhD students to Professors.	There have been 12 coding engagements for CAL in 2020, 10 of which were completed by Mike and 2 by other members of the RSG. These engagements have contributed to research across 3 of the College's 5 schools: Philosophy, Theology, and Religion; English, Drama, and Creative Studies; and Languages, Cultures, Art History and Music The outputs of these coding engagements have all been 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.

"Thanks so much for your hard work, it's a great website" - Dr Florent Perek, Lecturer in the Department of English Language and Linguistics

Centre for Computational Biology

The CCB funds a dedicated RSE, Dr Simon Hartley, in the Research Software Group. In the second half of 2020 Simon reduced his time in CCB to 50% as he prepared to take over ARC's RSE(HPC) role (see next page).

In 2020 Simon aided the Centre for Computational Biology (CCB) in delivering its research and producing cutting edge research in journals such as Nature and The Lancet. Simon also carried out a wide variety of tasks in the CCB, including: induction (for new members and research students to the CaStLeS and BEAR systems), aiding researchers in developing their HPC projects and installing applications. Simon also aided the CCB in its move to online learning.

During 2020 Simon gave the following training to the CCB's wider community, either to groups for up to 20 or personalised for an individual or small group: Introduction to Linux; Introduction to BlueBEAR/CaStLeS; Software Carpentries on bash and git; Python; R; MATLAB; and Fundamentals of CUDA for C++

Like the wider RSG, Simon delivered advice, coaching and coding support to the CCB and wider research community within MDS and LES. The advice is usually personalised on topics such as: Creating pipelines on CaStLeS; Job dependencies; Array jobs; Using Python on CaStLeS; Android Development; or Parallelisation of R.

Coaching sessions are usually bespoke and tailored to the research Groups for up to 5 but most engagements were one on one. They covered topics such as: Python; Django; HTML5 and JavaScript development; Convolutional Neural Networks for digital imaging and communications in Medicine; and version control with "git".

"Over the course of the Bear software coaching, I learnt many things about web development and how the Django framework operates. Some of the technologies I learnt during the coaching sessions were GIT, HTML, CSS and JavaScript. During my time with Simon [Hartley], I was also helped with learning database management and SQL. I would like to thank Simon for his time and helping me understand the Django framework and how it can be used to build complex webpages." - Dr Chris Middleton, Cancer and Genomic Sciences

HPC Midlands Plus dedicated RSE

As part of the University's membership of the HPC (High Performance Computing) Midlands Plus consortium, the University funds 50% of an RSE. Aston University do the same and purchase this time from ARC. The RSE(HPC) role in the RSG provides both universities' contribution. In the second half of 2020 Dr Simon Hartley took over the role from Dr Keith Evans.

During the transition, Dr Simon Branford handled the user and project administrative and much of the user-support workload for HPC Midlands Plus for both universities. This is an example of the strength of the approach of the universities sharing a single role in a larger team, as we could continue to support researchers using the facility without waiting for a recruitment round to complete.

The RSE(HPC) works collaboratively with research groups from across both universities for the definition, documentation, development and satisfactory completion of research software projects. A particular priority has been to engage with researchers at all development stages and disseminate best practices in the development of sustainable research software.

HPC Midland Plus' current HPC system "Athena" will be retired at the end of April 2021. The final resource allocation round for Athena was completed in December 2020.

The replacement system "Sulis" has already been approved by EPSRC, to be hosted at the University of Warwick. "Sulis: An EPSRC platform for ensemble computing delivered by HPC Midlands+" https://gow.epsrc.ukri.org/NGBOViewGrant.aspx?GrantRef=EP/T022108/1 and is expected to be delivered during 2021.

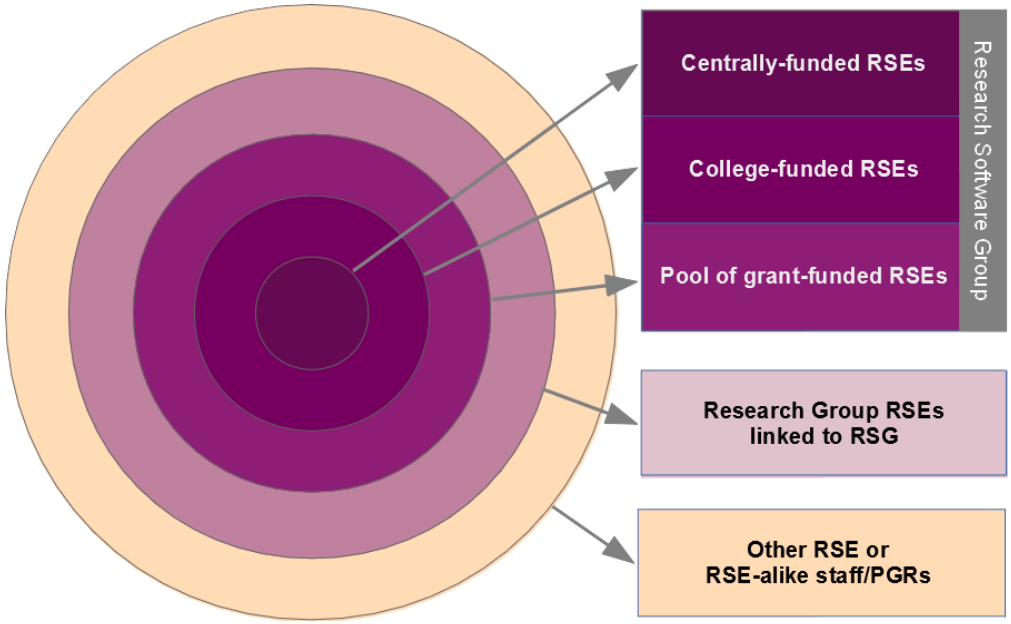
The Sulis proposal contains the following statement, highlighting the importance of this RSE(HPC) role: "A focus on ensemble computing raises challenges to researchers and software engineers. With thousands of simulations, the probability that at least one will fail is substantial. Software must be resilient to this failure. Similarly, managing the input and output of so many calculations can overload traditional data storage subsystems, requiring users to work with database technology rarely encountered by researchers outside of computer science departments. Hence a key feature of the Sulis service will be Research Software Engineering (RSE) support to assist and train users in tackling these problems, future-proofing the competitiveness of UK researchers to the challenges of computing at ever larger scales."

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.

2020 saw our first major funded coding engagement, and in December we began the process of recruiting our first dedicated Pool RSE.

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.



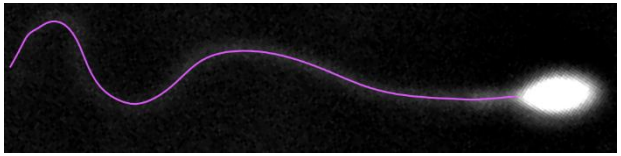
The RSG is comprised of three groups of RSEs. First the centrally funded RSEs who provide support for BEAR's main services and offer the BEAR Software Advice/Coding/Coaching services. Second are the college-funded RSEs who are line-managed in the RSG but work exclusively for the relevant college or academic unit. Third the new pool of grant-funded RSEs is available for including in grant applications, and will grow over time as more RSEs are recruited to it.

Fourth, some research groups will continue to recruit and manage their own RSEs, but we offer the option of a formal "dotted line" to the RSG to assist with managing the RSEs if desired. Finally some RSEs will continue to work independently of the RSG, meeting up informally at events such as the Academic Programmers Special Interest Group.

Please discuss any grant applications with us before including a pool RSE in the proposal.
See https://intranet.birmingham.ac.uk/bear-software/funded for more information on including a BEAR RSE in grant applications.

Funded Coding Case Study – Sperm Analysis

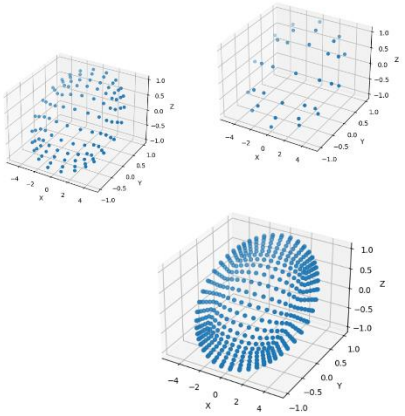
Prof Dave Smith and Dr Meurig Gallagher of School of Mathematics



This project studies Locomotion in Stokes flow for analysing and modelling sperm movement. This is an intensively studied problem because it describes important biological phenomena such as the motility of many species' sperm. The numerical computations involved can be challenging, particularly in three dimensions, due to the presence of moving boundaries and complex geometries. Building upon work which developed the first ever automated, high-throughput, flagellum detection algorithms for microscopy data[4], this project aims to reproduce the work done in MATLAB for the open source community using C++ and utilizing the software design principles of test driven development.

To date the new C++ code has over 60 unit tests proving the efficacy of the library and validity of the computations. Work on further OpenMP optimisations has begun.

$$\begin{pmatrix} A_{11}^S & A_{12}^S & A_{13}^S & A_1^U & A_1^\Omega \\ A_{21}^S & A_{22}^S & A_{23}^S & A_2^U & A_2^\Omega \\ A_{31}^S & A_{32}^S & A_{33}^S & A_3^U & A_3^\Omega \\ A_1^F & A_2^F & A_3^F & & \\ A_1^M & A_2^M & A_3^M & & \end{pmatrix} \begin{pmatrix} F_1[1] \\ \vdots \\ F_1[N] \\ F_2[1] \\ \vdots \\ F_2[N] \\ F_3[1] \\ \vdots \\ F_3[N] \\ \boldsymbol{U} \\ \boldsymbol{\Omega} \end{pmatrix} = \begin{pmatrix} B_{1j}\dot{\xi}_j[1] \\ \vdots \\ B_{1j}\dot{\xi}_j[N] \\ B_{2j}\dot{\xi}_j[1] \\ \vdots \\ B_{2j}\dot{\xi}_j[N] \\ B_{3j}\dot{\xi}_j[1] \\ \vdots \\ B_{3j}\dot{\xi}_j[N] \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix},$$



"The RSE team have been progressing from our initial Matlab code to a professionally-engineered product deployable on any system; we are excited about the potential of this to increase the impact and uptake of our work." - Prof Dave Smith

[4] Meshfree and efficient modelling of swimming cells, Meurig T. Gallagher and David J. Smith, Phys. Rev. Fluids 3, 053101 – Published 31 May 2018

Baskerville

The University has been awarded £4m for a major project to develop a computing system aimed at helping researchers to speed up the scientific discovery process and provide new insights into important research questions. The project is a collaboration between the University of Birmingham, The Rosalind Franklin Institute, The Alan Turing Institute and Diamond Light Source, the UK's national synchrotron. It is being funded by the Engineering and Physical Sciences Research Council (EPSRC), part of UK Research and Innovation and an allocation of this service will be available to EPSRC-funded researchers.

Called "Baskerville" and named after John Baskerville, the enlightenment-era Birmingham industrialist, the Tier 2 accelerated compute facility will provide a state-of-the-art platform for graphics processing unit (GPU)-accelerated computing. It will help researchers to accelerate machine learning algorithms and simulation technology, with wide-ranging applications in computer vision, language processing, molecular modelling, and materials science.

The new resource will be built with 46 Lenovo® Neptune™ liquid cooled servers each featuring twin Intel Xeon next generation CPUs, 512GB system RAM and 980Gb local NVMe storage built to support four of NVIDIA's new A100 Tensor Core GPUs attached to each system via high performance PCIe gen4 connection. In total 184 GPUs across the 46 compute nodes will provide over 2 Petaflops of computing resource. To support large data set processing, Lenovo will also be providing ~5PB (usable) of spinning disk and ~0.5PB (usable) of flash storage offered on the Lenovo DSS-G storage systems running IBM® Spectrum Scale™ RAID. The storage and compute systems will all be inter-connected using NVIDIA® Networking ConnectX® HDR InfiniBand adapters and Mellanox Quantum™ switches, providing 200Gbps links, to support high-throughput and scale-out workloads.

The system will be supplied by Lenovo® via the University of Birmingham's research computing framework partner OCF Limited who will support the logistics of delivering the system. Integration and operation of the facility will be by the Advanced Research Computing team, with the software support being provided by the Research Software Group.

More details available in the press release: https://www.baskerville.ac.uk/press-release/

Internal BEAR Systems

In 2020 we undertook several important tasks to enhance the internal software that we provide for interfacing with the systems that Advanced Research Computing provide to researchers. These were:

- upgrading to Python 3
- upgrading the Django version
- changing to use the Django ORM

The combined reasons for these changes were: to move away from Python 2 and Django 1, both of which stopped receiving updates in 2020; to vastly simplify the code and to reduce the number of lines of code; and to improve the general understanding of the code, which will mean more of the team can update it more easily. The web components of these systems also underwent an update - by refreshing the style we were able to also make sure the websites comply with accessibility legislation (see accessibility section).

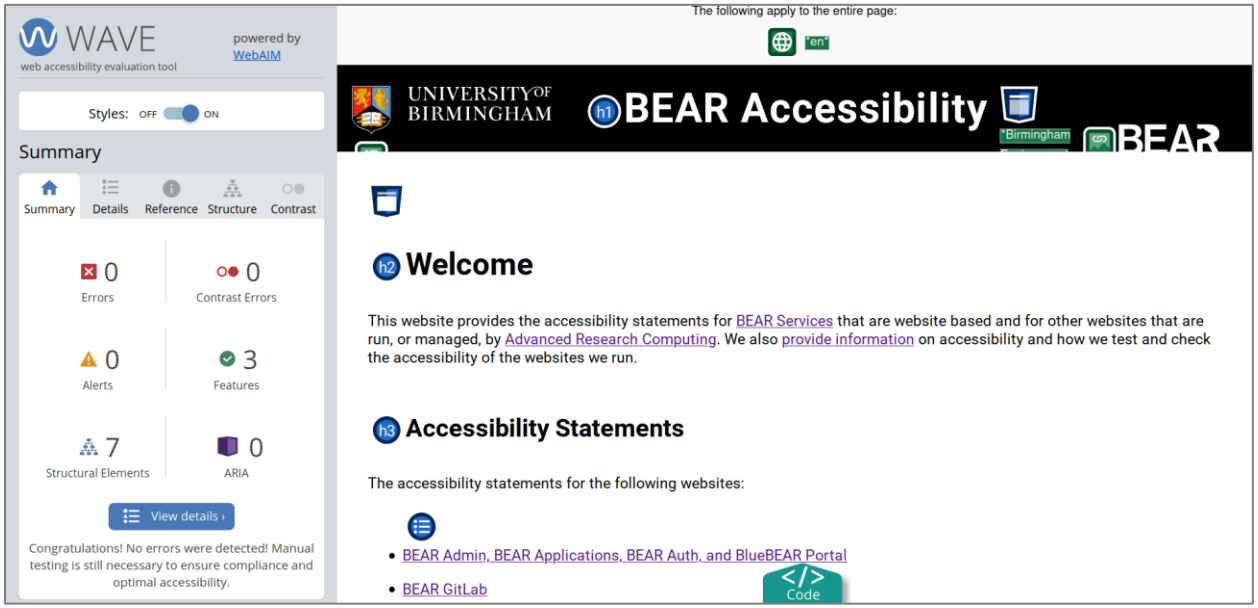
We also further developed our scripts and systems that we use to manage users and projects, and automatically control this information in several systems such as LDAP, Slurm, GitLab, etc. Our ongoing goal in this area is to automate tasks, to reduce the potential for human error, and to provide a fast and reliable service to the research community.

Accessibility

In September 2018 the government introduced regulations concerning the accessibility of websites and mobile apps. These regulations formally require that websites and mobile applications are suitable for those using them. September 2020 was the deadline for making older websites compliant with these new regulations.

In the RSG, we manage a number of websites that allow access to, and information about, the systems that are available to researchers at the University. We have also produced a number of websites for research projects (such as the ones described above in the case studies). In 2020 we spent time ensuring that these website meet the regulations and also that we have processes in place to ensure that these websites will continue to meet the regulations when they are updated. We have also provided support and advice to researchers across the University who are writing their own websites to ensure that these websites are compliant with the regulations.

As part of this work we have produced information about some of the tools we use to test accessibility - see https://accessibility.bear.bham.ac.uk/advice.html.



Training

DEEP LEARNING INSTITUTE

This year has proved a challenge to provide training workshops and courses to researchers at the University. In response to feedback from past attendees, we have, led by the Research Engagement Group in ARC, piloted a modular approach to the training, where many of our workshops are available as 1/2 day blocks. For most of 2020 our training has been delivered online.

BEAR Necessities

The RSG has continued to provide a general introduction to HPC course, titled *BEAR Necessities*, to researchers at the University. Additionally, we provide tailored versions of the course to specific areas of the University, for example addressing a group's or section's specific workflow or compute resources, whilst also providing a general level of information.

Carpentries

The Carpentries is a fiscally sponsored project of Community Initiatives, a registered 501(c)3 non-profit organisation based in California, USA. The Carpentries teaches foundational coding, and data science skills to researchers. At the University we provide workshops covering programming with Python, R, and MATLAB; version control using Git; and the Unix shell.

NVIDIA

We deliver the *NVIDIA Deep Learning Fundamentals for Computer Vision* and the *NVIDIA Fundamentals of Accelerated Computing with CUDA C/C++*.

C++

We provided a six day introduction to the C++ programming language and tools and techniques for software development.

Community, conferences and meetings

In 2020 most conferences and meetings were either cancelled or changed to being online. As a result the following list looks rather different than normal:

- James and Simon B attended the 5th EasyBuild User Meeting in Barcelona in January, with Simon presenting on "EasyBuild and POWER9"
- We contributed lots to upstream EasyBuild, and benefits lots from others' contributions. In particular, Simon B has now been invited to join the group of people who can review and merge contributions into the main EasyBuild repositories on GitHub.
- Ed attended the RUGRIT (Russell Universities Group Research IT sub-committee) meeting in Leeds, then subsequently via Zoom with Carol Sandys, acting head of ARC.
- Simon B presented at HPC Champions about Baskerville
- James attended the February HPC SIG and HPC Champions meetings in Bath
- Ed attended the November HPC SIG on Zoom
- Laurence attended FOSDEM in Brussels
- Ed attended the UK RSE leaders meeting in June (via Zoom).
- Luke Sudbery (AIS) and Ed attended the RUGRIT Slurm virtual workshop
- Simon B, Simon H and Afraz have attended the bi-monthly Zoom HPC Champions

Contact Us

BEAR Software

For any 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 help or information about any BEAR service 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 your route to find answers. Find all Advanced Research Computing items here: https://intranet.birmingham.ac.uk/bear/sd/bear

For all other IT Services items/help or to log faults and any complaints visit: https://www.itservicedesk.bham.ac.uk/