Software, Theology, High Performance Computing, AI and RSEConUK 2019

Andrew Edmondson
University of Birmingham
What I do... and where I do it...
BEAR Software is a range of services provided to researchers, and research groups, 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’.
Part-time PhD

“Institute for Textual Scholarship and Electronic Editing

“An Analysis of the Coherence-Based Genealogical Method using Phylogenetics”
Links

- **BEAR:**
  - https://www.birmingham.ac.uk/bear
  - https://www.birmingham.ac.uk/bear-software

- **RSEConUK 2019:**
  - @RSEConUK
  - https://rse.ac.uk/conf2019

- **Me:**
  - @ed_mondson
  - https://bham.academia.edu/AndrewEdmondson
Introduction
University of Birmingham

RSEConUK 2019

Manuscripts and Phylogenetics
University of Birmingham

- Royal charter in 1900 (history back to 1828)
- Member of Russell Group
- 34835 students (2017/18) – 4th largest in UK
- 11 staff and alumni are Nobel prize winners
- £134.2M research income (2017/18)
- £3.5bn economic impact
- Campuses in Birmingham and Dubai
Questions?
BEAR Software

IBM POWER9 AI Cluster

Index
“The Fourth RSE Conference will open on Tuesday 17th September and run for three full days, closing on Thursday 19th September 2019. We are not only expanding from two days to three but also opening up more delegate places to a maximum of 420 tickets.”
## Key Dates:

<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>7th May</td>
<td>Call for Participation closes</td>
</tr>
<tr>
<td></td>
<td>Call for Volunteers opens</td>
</tr>
<tr>
<td>23-30 May</td>
<td>Notification of successful submissions</td>
</tr>
<tr>
<td>31st May</td>
<td>Call for Volunteers closes</td>
</tr>
<tr>
<td>8th June</td>
<td>Detailed programme published on the website</td>
</tr>
<tr>
<td></td>
<td>Successful volunteers notified</td>
</tr>
<tr>
<td>17th June</td>
<td>Registration opens</td>
</tr>
<tr>
<td>13th August</td>
<td>Registration closes</td>
</tr>
</tbody>
</table>
“An Analysis of the Coherence-Based Genealogical Method using Phylogenetics”
“Non-traditional use of HPC”
An Analysis of the Coherence-Based Genealogical Method using Phylogenetics

New Testament Greek Manuscripts
The Coherence-Based Genealogical Method (CBGM)
(What’s it for?)
The search for the original text of the Bible (Traditional method)

A few manuscripts
The search for the original text of the Bible (20th Century Problem)

Far too many

Over 1,500 known manuscripts of John’s Gospel alone
Over 6,500 places of variation in John’s Gospel alone
Then they were willing to take him into the boat, and immediately the boat reached the shore where they were heading.
The search for the original text of the Bible (21st Century Method)
The search for the original text of the Bible (21st Century Method)
CBGM Process

1. Collate manuscripts
2. Calculate pre-genealogical coherence
3. Create initial local stemmata
   - Revisit local stemmata
   - Create optimal substemmata
   - Generate textual flow diagrams
4. Calculate genealogical coherence
5. Create global stemma
7. Study history of development of the text
Implementing the CBGM

MPI for Python provides an object-oriented interface to standard MPI-2 C++ bindings. The library's design and semantics of standard MPI-2 bindings should be able to use this.

Communicating Python Objects and Array Data
Communicators
Point-to-Point Communications
Collective Communications
Dynamic Process Management
One-Sided Communications

The Python standard library supports pickling and marshaling data.
John 1:8/16: μαρτυρηση vs. μαρτυρησει
Global stemma
Global stemma visualisation
Phylogenetics for the Bible
(In three slides)
Phylogenetics for the Bible
John 18, MrBayes
Nodes = manuscripts
Family 13 - Phylogenetics
PhD Conclusions?

- Phylogenetics works very well with this kind of data
- Phylogenetic and CBGM experiments effectively agree – and agree with traditional scholarship
- => Both methods basically sound
- Lots of recommendations for small improvements to the CBGM
Manuscripts and Phylogenetics

Questions?
BEAR Software – GOALS

- Enable the University of Birmingham's research community to get the best from their research software: "Better software, better research"
- Provide specialist software engineering advice and support to researchers and 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.
Advice

- How best to use HPC to analyse data from Cyclotron experiments
- Transitioning to using git and GitLab
- Parallel MATLAB use
- How to structure Django applications
- Use of HPC VM for health sciences
Coding

- Up to 10 days of free coding. Examples:
  - Replace outdated web application
  - Migrate MPI library to C++
  - Create an application to track the recycling process of batteries
  - Parallelise R script to run on HPC
Coding

10 RSEs

Max 20 projects / month
@ 10 days / project
Coaching

give a man a fish and you feed him for a day; teach a man to fish and you feed him for a lifetime
Coaching

10 RSEs

Impacting 100 projects / month
@ 1 session / week
Coaching

- Up to 20 half-day sessions of free coaching
- Examples:
  - Developing a web application
  - Coaching a department RSE
  - Developing a mobile app
  - Geospatial analysis in R
  - Experiment with C++ library

Fortran

UNIVERSITY OF BIRMINGHAM | RESEARCH SOFTWARE GROUP
Training

- Software Carpentries
  - Python, R, Git, Bash

- NVIDIA Deep Learning

- Intro to HPC
Annual report

RESEARCH SOFTWARE GROUP
BEAR - Advanced Research Computing

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

BEAR Software

☐ Questions?
BEAR AI

Delivering an Artificial Intelligence Platform

bearinfo@contacts.bham.ac.uk | @uob_rescomp
intranet.birmingham.ac.uk/bear-ai
“Researchers at the University of Birmingham are set to benefit from the largest IBM® POWER9™ Artificial Intelligence (AI) computer cluster in the UK.”

Currently 11 nodes, each with:

- Dual IBM POWER9 CPUs with 18 cores each, which currently present themselves as 144 cores using simultaneous multithreading (SMT4).
- Four NVIDIA Tesla V100 Tensor Core GPUs
- 1 TB system memory
- High speed NVIDIA NVLink interconnect fully meshed between the GPUs and also into the system memory
- 100G InfiniBand interconnect to other nodes and storage systems

https://intranet.birmingham.ac.uk/bear-ai
GPU-accelerated Software

**PyTorch**

Installed version: 1.0.1
*An open source deep learning platform from Facebook.*

**TensorFlow**

Installed version: 1.10.1
*Python-based open source machine learning framework from Google.*

**Amber**

Installed version: 18
*A suite of biomolecular simulation programs with GPU acceleration*

**GROMACS**

Installed version: 2018.4
*HPC molecular dynamics package with GPU acceleration.*
HPC Software

- **Python3**
- **TensorFlow**
- **mappy**
  - interface to minimap2 to align genomic and transcribe nucleotide sequences
- **h5py**

- “optimized for x86-64 CPUs” using SSE2: Intel extensions for “Single Instruction, Multiple Data”
- SSE2 code won’t build on POWER
- POWER equivalent is “Altivec”
- So we need to port the C code…
Porting SSE to Altivec

- [https://github.com/IvantheDugtrio/veclib](https://github.com/IvantheDugtrio/veclib) implements a subset of the SSE2 -> Altivec conversion
- [https://github.com/IvantheDugtrio/veclib/pull/6](https://github.com/IvantheDugtrio/veclib/pull/6) is our PR which has been merged upstream, including our implementation of a missing function and some fixes for others
- Problem: there is no direct Altivec equivalent for SSE2’s `_mm_cvtsi32_si128` function – so we had to write one.
- Now we have installed veclib on our POWER9 boxes – using EasyBuild – for whenever we need it.
MAY 7, 2019 BY ANDREW EDMONDS

Second Power AI User Group Meeting

The second Power AI UG meeting will take place on Monday 8th July in Birmingham from 10am to 4pm. Register here.

https://www.poweraiug.org
BEAR AI

☐ Questions?
“EasyBuild is a software build and installation framework that allows you to manage (scientific) software on High Performance Computing (HPC) systems in an efficient way.”

https://easybuilders.github.io/easybuild/
EasyBuild

- EasyBuild allows us to easily and reproducibly build software for various different platforms in BlueBEAR and BEAR Cloud.
- We have:
  - EL7 sandybridge, haswell, broadwell, skylake
  - Ubuntu 16.04 haswell
  - EL7 POWER9
  - And Cascade Lake has just arrived on site…
EasyBuild

- Wasn’t so “Easy” with POWER9
  - EasyBuild expects Intel
  - Lots of software doesn’t build out of the box via EasyBuild on POWER9
- But… I’ve worked through various of the problems and am contributing it all upstream.

- And now it is “Easy”
EasyBuild

☐ Questions?
Good Coding Practices

IBM POWER9 AI Cluster

Index
Good Coding Practices
Reproducible Research

It’s two years later… and you need to reproduce those results… but your software doesn’t produce the same results today…
Reproducible Research

- VERSION CONTROL
  - Data
  - Software
  - Environment
- Does the same version of the software using the same version of the data running in the same environment produce the same results?
If your software is not under control, then you cannot have confidence in your results. Or hope to reproduce them.

(or hope that anyone else could reproduce them either)

Reproducible Research
Collaboration

Time has passed, you are now a PI. And you’ve got some RFs working with you now…

And they need to work on your code.
Collaboration (1/3)

- Version control (again!)
  - Branching, merging etc.
- Issue tracking
- Testing
  - Unit tests with continuous integration
- Building
  - Straightforward documentation
  - Use standard tools
Collaboration (2/3)

- Comments – not too much, not too little
- Naming conventions
  - Use meaningful names
- Coding standards
  - E.g. PEP8 for Python
  - Static code analysis (while editing if possible)
- Make it readable
Collaboration (3/3)

- **Structure**
  - Functions should do one thing
  - Functions should fit on one page
  - Files should contain related things
  - Folders should contain related things
- **Simple is better than complex**
  - Don’t write code you don’t need

Complexity
It’s 5 years later and you need to edit your own code… but you have no idea how it works.
Editing your own code

- See “Collaboration”

Thinking “I’ll never need to…”
Optimisation

You need to run your code with data 1,000,000 times bigger than before... and it's really slow.
Optimisation

- Profiling
  - Language/system specific
- DRY
- Docstrings
- Architecture documentation
- Then optimise the slow, frequently used bits
  - Optimising usually adds complexity
Good Coding Practices

☐ Questions?