RESEARCH SOFTWARE ENGINEERING

Overview and value to your research group

Presenters:
Simon Yin and David Gunawan
Outline

1. Overview: The origins of research software engineering (presented by Simon Yin)
2. Value: The motivation for a 15 week engagement to provide research software engineering support to UNSW Business School – School of Economics (presented by David Gunawan)
3. Value: Learning outcomes to be applied in research activities in the upcoming year (presented by David Gunawan)
4. Reporting and expected timeline of a typical research software engineering support engagement (presented by Simon Yin)
Research Software Engineering

All tasks related to implementing software for the current purposes of a researcher, and for subsequent reuse by future researchers who need to reproduce current results.
Research Software Engineering

IT DevOps

Research Software Engineer

Researcher
Algorithms: can derive proofs and publish in peer reviewed publications…
Programming knowledge…

Interpreted eg. Python, Bash

Compiled eg. C++, Julia

Technical IDE eg. Matlab, R-Studio
Maths: code for Linear Algebra, Statistics...
Performance: can parallelise codes and optimise systems/networks...
Performance: can trade-off accuracy for improved performance...
Agile, CI and UX best practices...
Case: UNSW School of Economics

■ How did you initiate the engagement of UNSW ResTech?
■ My research focus is on computationally expensive Bayesian simulation methods.
■ Some of the Bayesian methods:
  - Particle Markov chain Monte Carlo (PMCMC)
  - Annealing/Density Tempered Sequential Monte Carlo
  - Particle Filter
  - Variational Bayes, etc….
■ All these methods are computationally very expensive.
  - ‘For’ loop that you cannot avoid (sequential algorithm).
  - Large Matrix computations.
■ Initially, I ran all these algorithms on Katana (UNSW HPC managed by Martin Thompson and Joachim Mai).
■ ResTech sent Simon Yin to help optimise our codes.
Case: UNSW School of Economics

- What inputs did you bring to start the engagement?
- Submitted papers:
  - Gunawan, D., Kohn, R., Carter, C., and Tran, M.N. Flexible Density Tempered for State Space Model. (Matlab codes)
  - Gunawan, D., Carter, C., and Kohn, R. Efficiently combining pseudo Marginal and particle Gibbs sampling. (Matlab code from David and Julia code from Yue (NCI)).
Case: UNSW School of Economics

■ What learning outcomes will you apply to your research in the upcoming year?
■ From this engagement, I have learnt about:
  - Automatically generating Mex files in Matlab
  - Automatic Differentiation in Matlab using Casadi package
  - GPU offloading
  - High Dimensional Tensor in Matlab
■ Found that Matlab is better than Julia if we do not carefully write the Julia code.
Expected timeline for 15 week engagement

- **Week 1**
  - Scoping and Introduction
  - Systems to support the engagement

- **Week 2**
  - Read previous written work
  - Install previous codes

- **Week 3**
  - Delivery target: early speedup from simple changes (e.g., function substitution)
Expected timeline for 15 week engagement

- **Week 4**: Start path with deepest code impact but highest expected payoff
- **Week 5**: Implement and test
- **Week 6**: Evaluate: report speedup and choose next path
Expected timeline for 15 week engagement

- **Week 7**: Start 2nd path with large code impact and high expected payoff
- **Week 8**: Implement and test
- **Week 9**: Evaluate: report speedup and choose next path
Expected timeline for 15 week engagement

- **Week 10**: Augment/Offload: low code impact and high payoff
- **Week 11**: Implement and test
- **Week 12**: Evaluate: report speedup
Expected timeline for 15 week engagement

- **Week 13**: Revisit promising paths in more detail
- **Week 14**: Report writing
- **Week 15**: Presentation
Summary of this engagement

“...A 15 week engagement with Professor Robert Kohn’s research group (Economics - UNSW Business School) has improved the efficiency of their numerical simulations by 20% (and as much as 50%).

In each allocation period, this amounts to freeing up 17kSU (out of a typical 50kSU usage based on historical trends).

This research group has now been ‘seeded’ with up-to-date implementations of their routines in Matlab and Julia.

They are independently able to use these on personal computing systems, as well as systems available through UNSW Research Technology Group (eg. Katana, Raijin)…”
Examples of Intractable Likelihood Problems

- In big data, the log-likelihood is given by:

\[ \log p(y|\theta) = \sum_{i=1}^{n} \log p(y_i|\theta) \]

With big \( n \), it is too expensive to compute \( \log p(y|\theta) \).

- In random effect panel data models, the likelihood is given by:

\[ p(y|\theta) = \prod_{i=1}^{n} \int p(y_i|\alpha_i, \theta)p(\alpha_i|\theta) d\alpha_i \]

The likelihood is intractable.
Examples of Intractable Likelihood Problems

- In a time series state space model, the likelihood is given by

\[ p(y_{1:T} | \theta) = \prod_{t=1}^{T} \int p(y_{1:T}, x_{1:T} | \theta) dx_{1:T} \]

- The likelihood is typically intractable because it is an integral over the latent state variables \( x_{1:T} \).
Bayesian Methods

- Some of the Bayesian methods:
  - Particle Markov chain Monte Carlo (PMCMC)
  - Annealing/Density Tempered Sequential Monte Carlo
  - Particle Filter
  - Variational Bayes, etc.

- All these methods are computationally very expensive.
  - ’For’ loop that you cannot avoid (sequential algorithm).
  - Large Matrix computations.

- Initially, I ran all these algorithms on Katana (UNSW HPC managed by Martin Thompson and Joachim Mai).

- ResTech sent Simon Yin to help optimise our codes.