Zacros: a Kinetic Monte Carlo Package for Surface Catalysis

I. Christidi[1], R. Benson[2], R. Guichard[3], J. Nielsen[3], S. Ravipati[2], R. Reocreux[2], G. Savva[2], D. Stansby[1], M. Stamatakis[2]

[1]UCL Advanced Research Computing Centre; [2]Thomas Young Centre and Department of Chemical Engineering, UCL; [3]Research Software Development Group, Research IT Services, UCL

Background

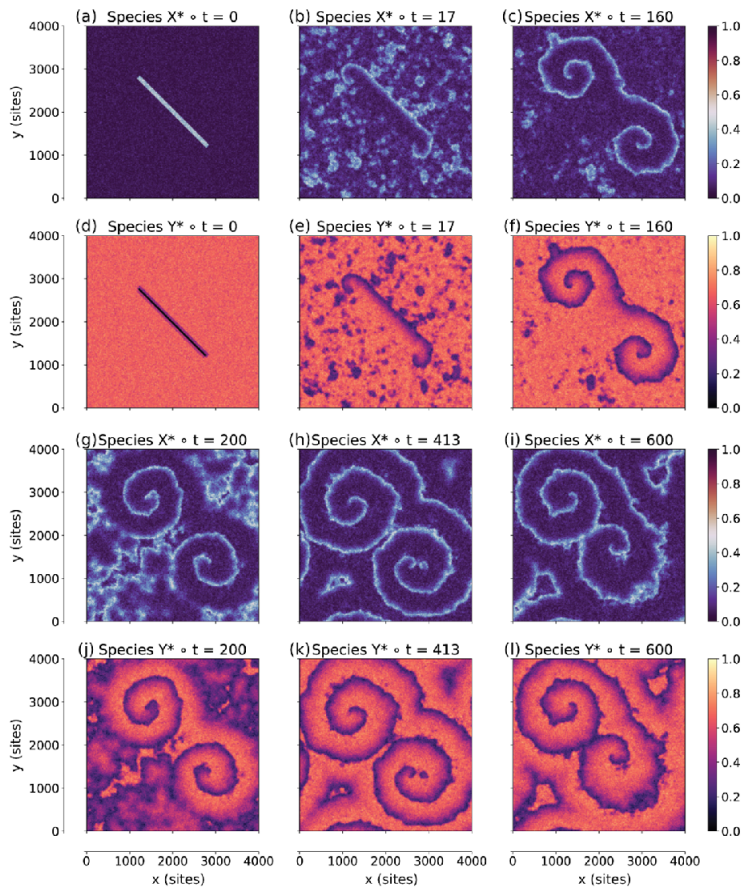
Kinetic Monte-Carlo (KMC) simulations have been instrumental in multiscale catalysis studies, enabling the elucidation of the complex dynamics of heterogeneous catalysts and the prediction of macroscopic performance metrics, such as activity and selectivity. However, the accessible length- and time-scales have been a limiting factor in such simulations. For instance, handling lattices containing millions of sites with "traditional" sequential KMC implementations is prohibitive due to large memory requirements and long simulation times.

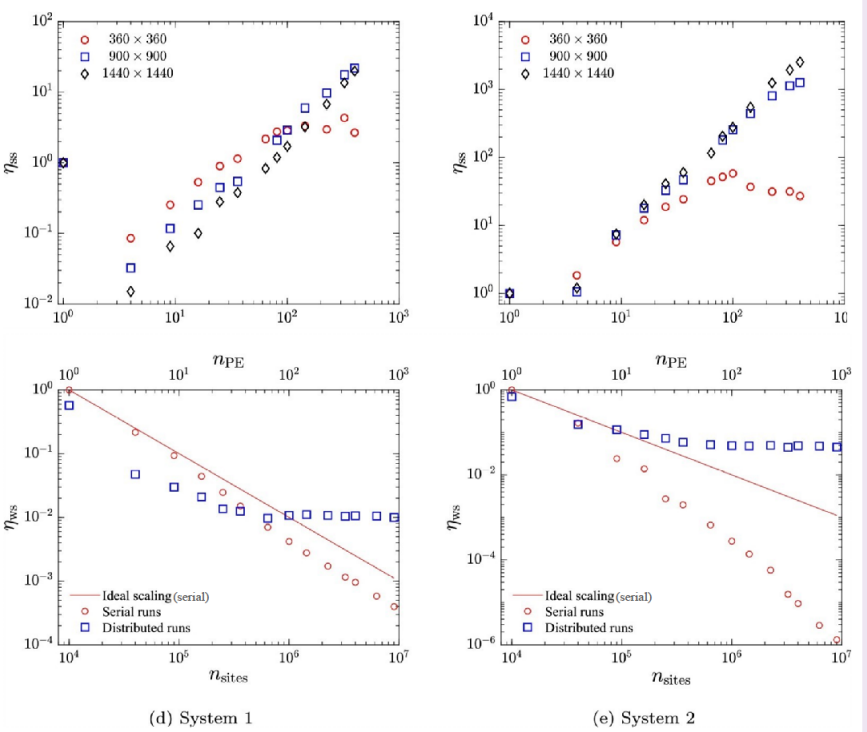
Zacros is a KMC software package for simulating molecular phenomena on catalytic surfaces. It is written in modern Fortran and incorporates an MPI implementation of the Time-Warp[1] algorithm for distributed, exact KMC simulations, making it possible to harness the computational power of massively parallel computing architectures.

Results

ARC Research Software Engineers have incorporated into *Zacros* the Time-Warp algorithm, an optimistic approach to Parallel Discrete Event Simulation. It has resulted **in speed-ups of 1 to 4 orders of magnitude[2,3]**, depending on the chemical system simulated and the number of MPI processes used.

The power of the implementation is demonstrated by the simulation of a variant of the Brusselator system[4], a **prototype chemical oscillator** that can form spiral wave patterns, which would be computationally impossible to capture with sequential KMC. The results of a 38-day long simulation run on 625 MPI processes, representing a **speed-up of a factor of 16**, are shown for different snapshots of KMC time, as the fractional coverage of the lattice surface by element X* (blue panels) and Y* (orange panels).





(d) System 1 (e) System 2

Strong (top) and weak (bottom) scaling behavior of the distributed-memory parallel implementation of *Zacros* (Time-Warp) for two different simulated systems.

The strong scaling efficiency n_{ss} is the ratio of simulation time progression in a run distributed over n_{PE} MPI processes, over in a serial run, as a function of n_{PE}. It is shown for three different size lattices.

The weak scaling efficiency n_{ws} is the ratio of simulation time progression in a run with n_{sites} lattice sites, over in a run with the minimum n_{sites}, as a function of n_{sites} . It is shown for both serial and parallel runs, where in the latter case the number of MPI processes increases with n_{sites}.

Optimal performance is obtained for large enough lattices distributed over an appropriate number of MPI processes that keeps the local computational workload per process large enough relative to communication: weak scaling flat, and strong scaling away from the high-n_{PE} plateau. Those conditions vary significantly for different chemical systems.

[1]D.R. Jefferson (1985). *Virtual Time*. ACM Transactions on Programming Languages and Systems. **7**, 404-25. (doi: 10.1145/3916.3988)

[2]S. Ravipati, et al. (2022). *Coupling the Time-Warp algorithm with the Graph-Theoretical Kinetic Monte Carlo framework for distributed simulations of heterogeneous catalysts*. Comput. Phys. Commun. **270**, 108148. (doi: 10.1016/j.cpc.2021.108148)

[3]G. Savva, et al (2022). *Large-scale benchmarks of the Time-Warp/Graph-Theoretical Kinetic Monte Carlo approach for distributed on-lattice simulations of catalytic kinetics*. Under review.

[4]G. Savva, et al. (2022). *Exact Distributed Kinetic Monte Carlo Simulations for On-Lattice Chemical Kinetics: Lessons Learnt from Medium- and Large-Scale Benchmarks*. Under review.

Project PI: Michail Stamatakis: m.stamatakis@ucl.ac.uk
ARC Lead: Ilektra Christidi: ilektra.christidi@ucl.ac.uk
Project Websites: https://zacros.org , https://xip.uclb.com/product/zacros-3
UCL ARC: https://www.ucl.ac.uk/arc

Engineering and Physical Sciences Research Council
archer
archer2
The Leverhulme Trust

Centre for Computational Science: Science for the Exascale

J. W. S. McCullough, A. Rallis, S. Wan, M. Vassaux, P. V. Coveney

Centre for Computational Science, Department of Chemistry, University College London

The Centre for Computational Science is concerned with many aspects of theoretical and computational science, from chemistry and physics to materials, life and biomedical sciences as well as informatics. We explore these domains through high performance, data intensive, supercomputing and distributed (grid/cloud) computing.

Our different computational techniques span time and length-scales from the macro- through the meso- to the nanoscale. We are committed to studying new approaches and techniques that bridge these scales.

A key component of our current work is focused on preparing our tools for use on some of the newest, fastest and largest computing resources becoming available.

QUANTUM COMPUTING

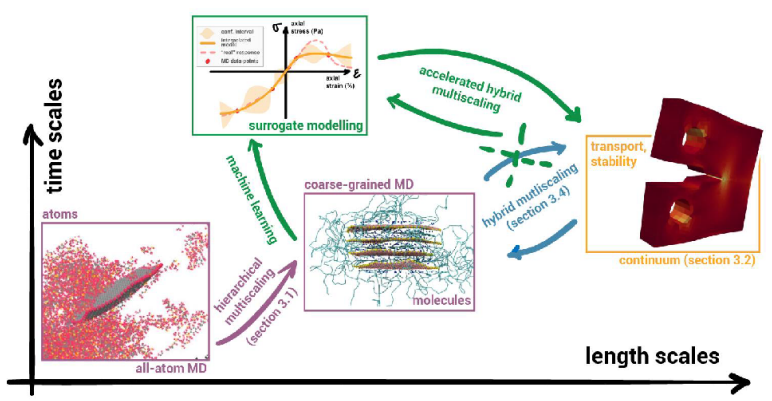
Solving the electronic structure problem is essential in theoretical chemistry, as it allows properties of materials to be modelled without needing direct access to the molecular system. However, numerical representations of systems quickly become too complex, making studying these problems on standard HPC hardware quickly intractable.

Our work with projection-based embedding, qubit tapering and the contextual-subspace variational quantum eigensolver algorithm seek to allow quantum computers to be able to solve this problem. As quantum devices can represent chemical problems more efficiently, they will provide a more tractable way to solve electronic structure problems compared with traditional HPC approaches.

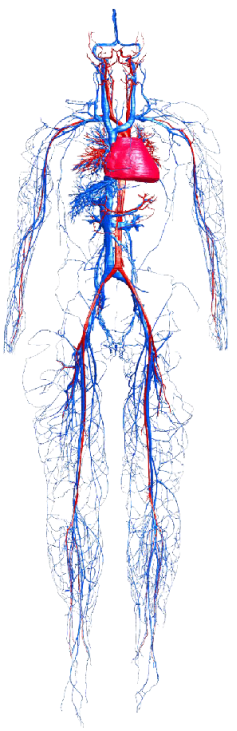
MATERIALS

Material science is a archetypal multiscale problem – responses to loading at the continuum scale is governed by structures and interactions at the atomic scale.

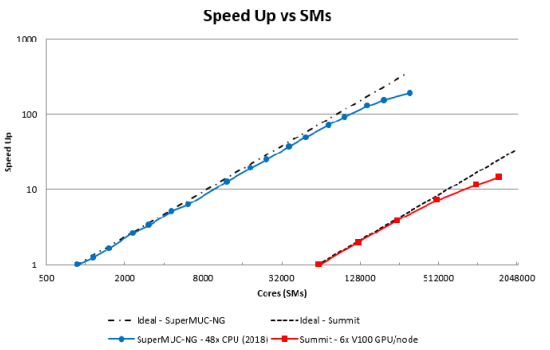
With SCEMa we are able to combine simulations at these two extremes of the length scale. Combined with uncertainty propagation based on molecular dynamics ensembles we can develop more reliable models of material behaviour.



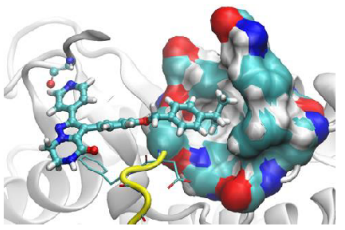
BIOMEDICAL

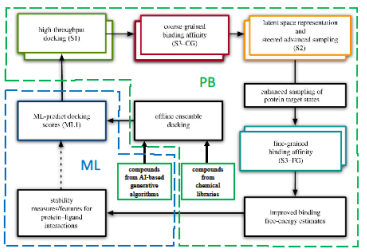


HemeLB is a 3D macroscopic fluid dynamics solver that utilises the lattice Boltzmann method to solve for blood flow in large and complex vascular geometries. CPU and GPU versions are available with both demonstrating strong scaling to very large scales. Porting and preparing the GPU code for new hardware is essential for use on exascale platforms.



The IMPECCABLE pipeline combines physics and machine learning based methods to efficiently and accurately assess drugs candidates from a large sample space.
This approach has sampled over 4.2 billion molecules as possible candidates against SARS-CoV-2.





Co-design with leading hardware manufacturers and HPC centres is further enabling our applications to benchmark new machines and prepare for upcoming hardware innovations.

Professor Peter Coveney - Director of the Centre for Computational Science, Associate Director of the Centre for Advanced Research Computing, UCL

p.v.coveney@ucl.ac.uk

LEXCI

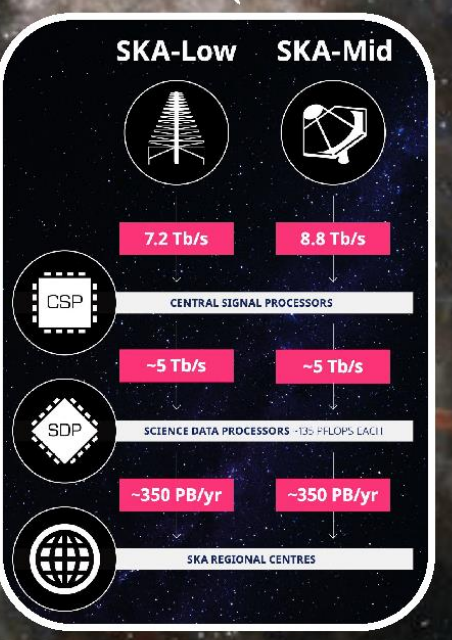
Learned Exascale Computational Imaging

Science Contact: *jason.mcewen@ucl.ac.uk*
ARC Contact: *t.koskela@ucl.ac.uk*

UCL

Abstract

We are developing Learned EXascale Computational Imaging (LEXCI) algorithms that **learn image and instrument models**, significantly enhancing image fidelity and interpretability while dramatically reducing computational cost. Our hybrid techniques will **blend model-based and deep learning approaches**, ensuring effectiveness, efficiency, and generalisability, while facilitating **uncertainty quantification**. Our software implements novel, highly parallelised and distributed algorithms, designed to usher in the era of exascale computational imaging.

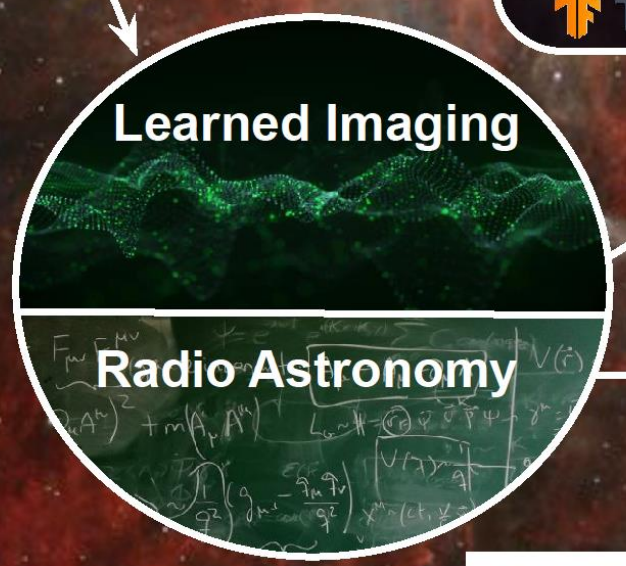


Leverage extensive software development expertise of **Advanced Research Computing (ARC)** based at UCL.

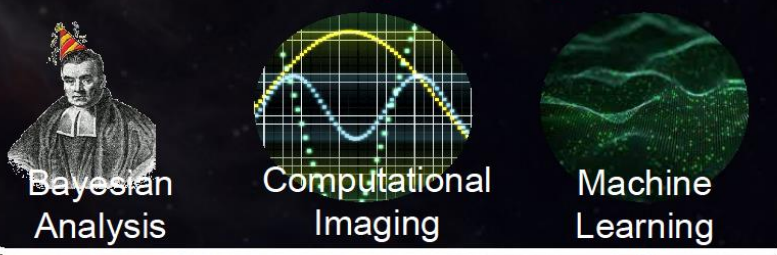
LEXCI will develop **parallelised models**, which can run on highly **distributed algorithms**, and support **machine learning priors** with principled **uncertainty quantification**.

OpenCL OpenMP TensorFlow

Develop and integrate **next-generation learned imaging** techniques to unlock the potential of **big data** and **exascale computing**.



Methodological Impact



Scientific Impact

Team Members

UCL + HERIOT WATT UNIVERSITY

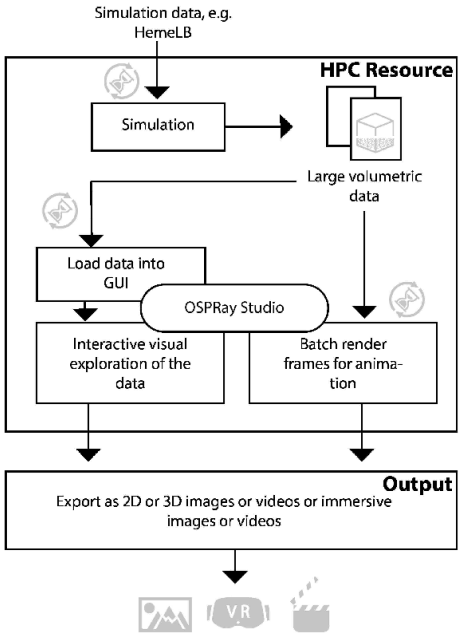
SOPT PURIFY

Immersive 3D blood flow visualization with Intel OSPRay Studio

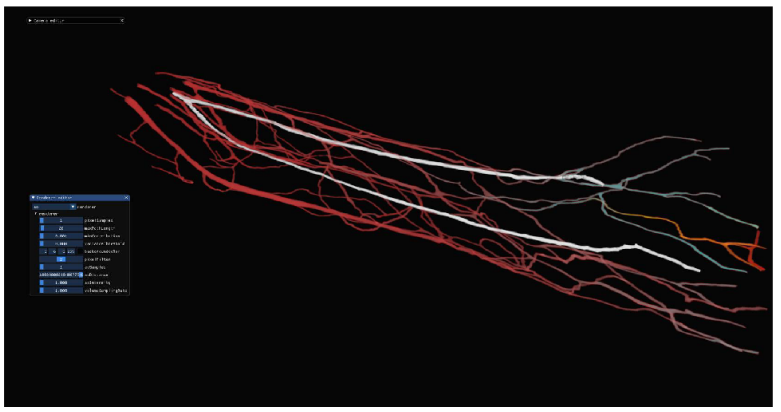
J.W.S. McCullough[1], E. Mayer[2], S. Cielo[2], J. Günther[3] and P.V. Coveney[1]

1. University College London; 2. Leibniz Supercomputing Centre; 3 Intel Corporation (Germany)

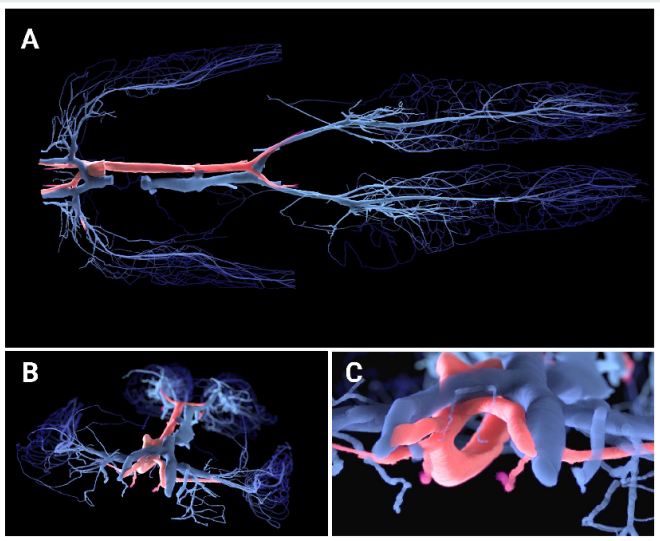
One challenge yet to be fully overcome when using high-performance computing in biomedical research is how to effectively communicate the simulated data to those from a clinical background. Recent advances in the availability and capability of virtual reality make this technology ever more convenient in assisting the visualization of complex data sets. In this work we use the HemeLB solver to simulate high resolution, 3D blood flow through selected personalized domains of human-scale vasculatures. We then use Intel OSPRay Studio to generate immersive visualizations of these large data sets where the viewer is situated within the 3D vascular structure and able to examine flow throughout the whole domain. Our approach can help bridge the knowledge gap between the computational and clinical domains and provide a pathway for practical and efficient utilization of personalized digital twins of the human body.



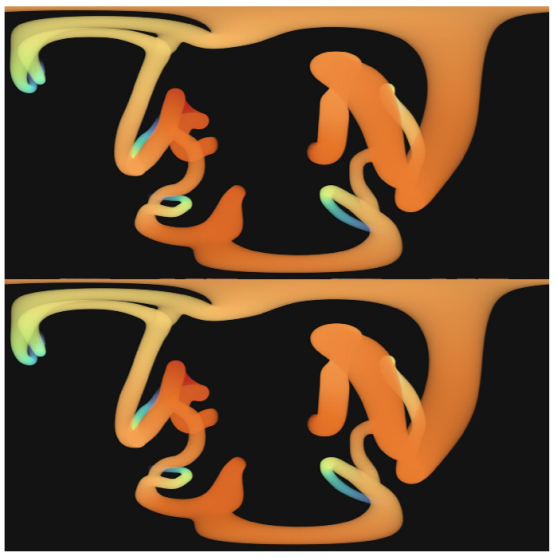
Our workflow revolves around OSPRay Studio, part of the Intel oneAPI Rendering Toolkit. OSPRay Studio offers advanced cameras and stereo rendering capabilities useful for VR applications, besides a GUI we use to easily access and explore the data in real-time. For larger productions, we can also launch batch jobs distributed among multiple nodes using various parallelization strategies. This is currently in use on SuperMUC-NG at the Leibniz Supercomputing Centre.



With the GUI of OSPRay Studio the user is free to browse the ray-traced data set and examine features of interest. The figure also shows the renderer menu panel, in which the users select their preferred renderer (e.g., ray casting and shadows with 'SciVis', or 'Path Tracer') and viewing parameters (sampling, background colors, ambient occlusion distance and more).



Detailed perspectives of simulated, full human arterial (warm/red tones) and venous (cold/blue tones) domains colored by pressure. (A) Full domain. (B) looking down on the data set from the nek region (image center, in focus) (C) High-detail close-up on the neck region, indicative of our high-resolution..



Stereoscopic 360° visualization of the blood pressure in the Circle of Willis arterial structure. Located at the base of the brain, the Circle of Willis is responsible for regulating the blood flow in and out from it

Stereoscopic renderings are meant for immersive and interactive user experiences through virtual reality headsets.

With progress in HPC and digital twins of the human body, the need to effectively visualize large and complex datasets is only going to grow. Platforms such as OSPRay Studio, that can achieve this on same high-performance computers where simulations are conducted provide a promising path forward.

Discover more in our virtual room – https://hubs.mozilla.com/2oVrcCv or scan the QR code →

Corresponding author: Peter Coveney - p.v.coveney@ucl.ac.uk

Julia: a high-level language for exascale

Mosè Giordano[1], Milan Klöwer[2], and Valentin Churavy[3] **Contact**: m.giordano@ucl.ac.uk

[1]Advanced Research Computing, University College London, United Kingdom; [2]Atmospheric, Oceanic and Planetary Physics, University of Oxford, United Kingdom; [3]CSAIL, EECS, Massachusetts Institute of Technology, Cambridge, United States of America;

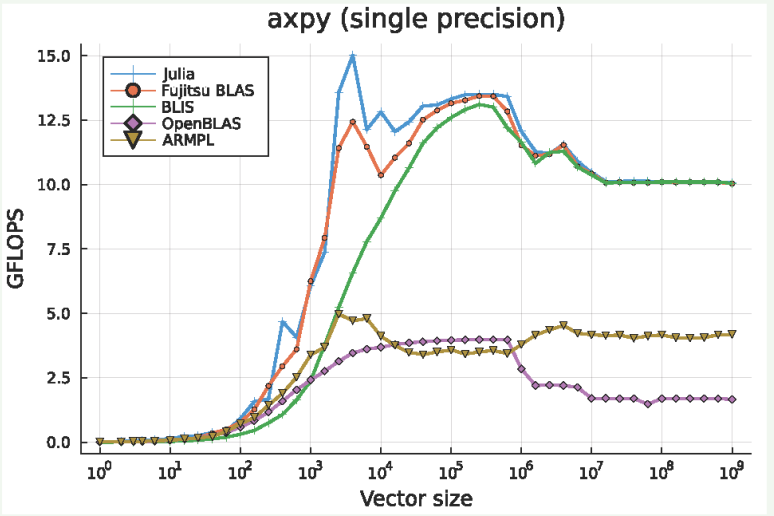
What are Julia and A64FX?

Julia is a modern **high-level**, **dynamic**, and **high-performance** programming language, particularly well suited for numerical computing. By combining speed and ease-of-use, Julia aims to **bridge the gap between domain experts and performance engineers**. It has support for multi-threading and distributed computing, and it comes with a built-in package manager focused on **reproducibility**. It also features advanced **code introspection**, for performance fine-tuning, and easy **interoperability with other languages**, to work with existing code. **A64FX** is a highly-specialised CPU for HPC, powering, among others, **Fugaku** (second fastest supercomputer in the world as of June 2022) in Japan and **Isambard 2** in the UK.

Standing on the Shoulders of the Giants

A simple and generic implementation of BLAS `axpy` can **outperform vendor libraries**.

```
function axpy!(a::T, x::Vector{T}, y::Vector{T}) where {T<:Number}
    @simd for i in eachindex(x, y)
        @inbounds y[i] = muladd(a, x[i], y[i])
    end
    return y
end
```

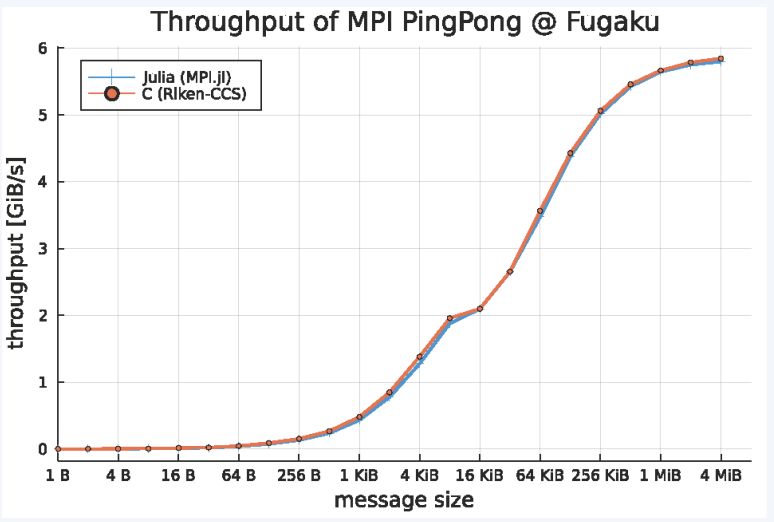


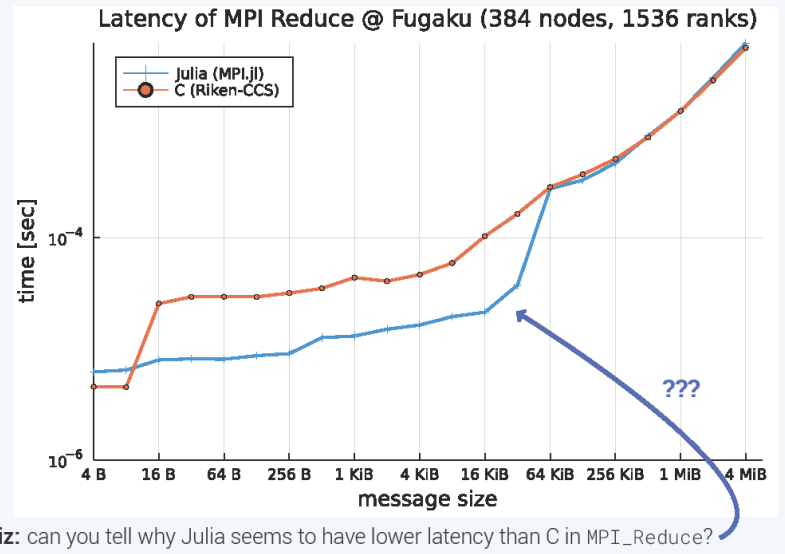
Performance comparison of single-thread `axpy` implementations in Julia versus Fujitsu BLAS, BLIS, OpenBLAS, and ARMPL on **Fugaku**.

There is no magic: Julia is **built on top of LLVM** and it benefits from its optimisations.

Ready for HPC

Julia can use the **MPI protocol** through the `MPI.jl` package and scale up computation to multiple nodes with little overhead compared to low-level languages.

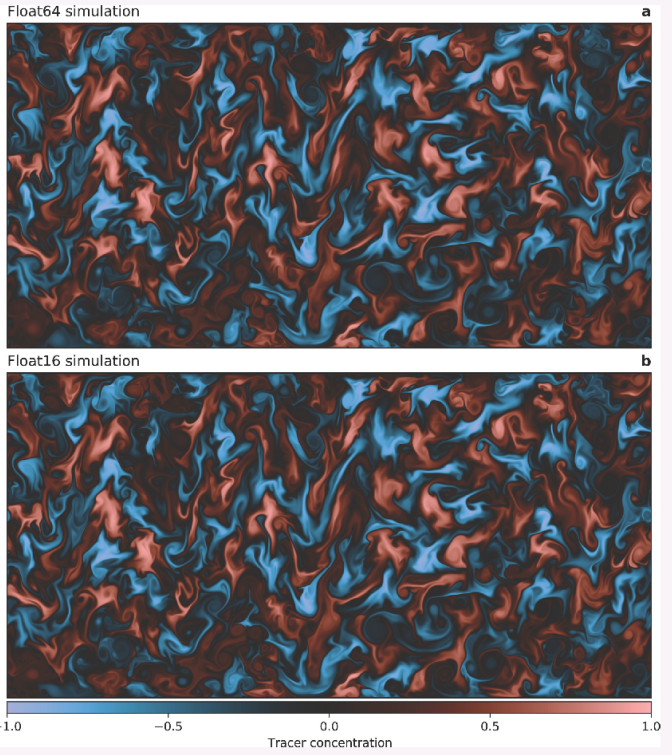




Quiz: can you tell why Julia seems to have lower latency than C in `MPI_Reduce`?

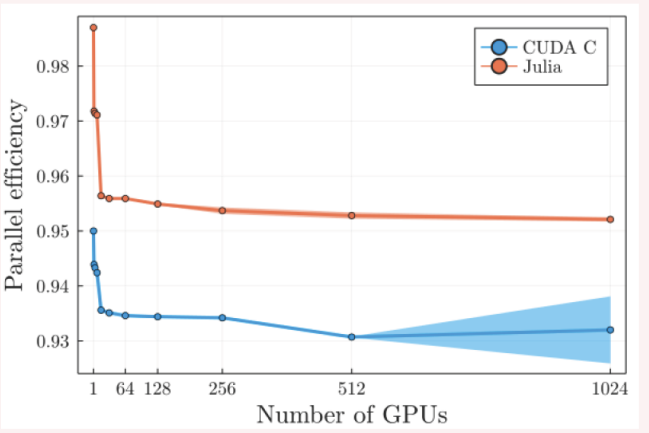
A Case for Type Flexibility: Reduced-Precision Numbers

Thanks to its type system, it is easy to write **generic code** in Julia, which can be used with different data types. For example, a **geophysical turbulence simulation**, powered by the `ShallowWaters.jl` package, can be run on A64FX using half-precision floating point numbers (`Float16`) at **4x the speed** of when using double-precision numbers (`Float64`), while retaining the same accuracy.



Not Only CPUs

Julia can also compile code for accelerators like **GPU** (NVIDIA, AMD, Intel, Apple, etc...). GPU arrays can be used in type-generic code which accepts arbitrary array types and packages like `KernelAbstractions.jl` make it easier to write code **targeting both CPU and GPU**.



Watch the talk

Parallel efficiency of a weak-scaling benchmark using `ParallelStencil.jl` on1 to 1,024 NVIDIA P100 GPUs on the Piz Daint Cray XC50.

Got Curious?

If you got interested, attend the **"Julia for HPC" BoF** on Tue 15th at 12:15PM (@D163) to learn more. There is still a lot more to do for making Julia a first-class citizen in HPC, if you want to contribute **get in touch** with me and join the **monthly JuliaHPC call**: `https://julialang.org/community/`.

UCL ARC

This work, presented at the *IEEE Cluster 2022 Conference*, was the **first evaluation** of the **readiness of the Julia programming language on A64FX**, as an innovative alternative to traditional HPC languages. This promotes the goals of the **UCL Advanced Research Computing Centre**, which is dedicated to the improvement of digital tools, practices and systems that enable science.

Read the paper

Building a Machine-Actionable Pipeline for Research Data

M. Ahmed, K. Buncic, A. Smith, J. Wilson, R. Macneil, J. Hetherington

Centre for Advanced Research Computing (ARC), University College London

Abstract

At UCL, the Centre for Advanced Research Computing (ARC) is building and supporting the vital integrated digital research infrastructure that is critical in allowing researchers to work with data at scale in an efficient and innovative manner.

Alongside its research and academic activities, ARC is providing a range of inter-connected research tools, services and technological initiatives with the aim of amplifying the reusability and impact of research data by making data FAIR-compliant and machine-actionable.

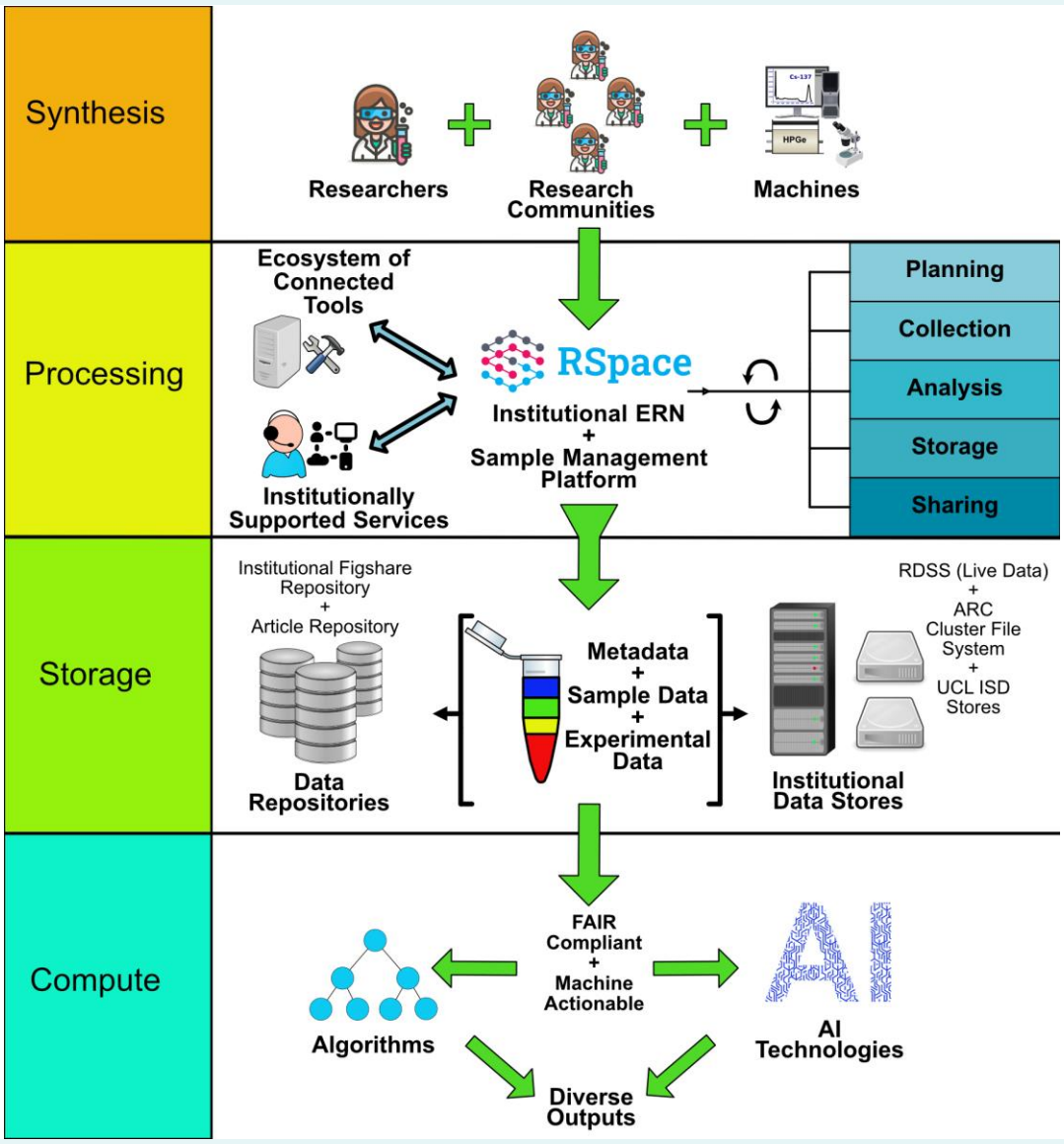
Ultimately, creating an ecosystem of institutionally supported tools and services will enable AI technologies and computing resources to be employed in extracting maximal value from research data produced at UCL across a variety of fields.

Context

Increasingly, the importance of properly managing, preserving and sharing research data is being recognized across the globe.

Currently, the true value of vast amounts of research data is unrealized, in part as a result of information silos, extensive use of paper notebooks and lack of associated metadata.

As global movements to make research data FAIR continue to evolve, it is becoming clear that dedicated support and initiatives must be put in place to enable computation at scale.



At UCL, ARC provides:

Large scale compute facilities including cloud computing, high-throughput and high-performance systems.

Data storage facilities and repositories including the institutional Figshare deployment and an on-premise Research Data Storage Service for live data.

A range of tools, services, software and support to enable computationally empowered science and scholarship.

Research technology professionals- data scientists, informaticians, research software engineers, HPC systems engineers, dev-ops specialists, data engineers and data stewards - who collaborate with researchers.

The Role of Data Stewards

Data stewards at UCL operate in a cross-functional capacity to support, advise and collaborate with researchers. Their work involves translating research data needs into infrastructure and service requirements for deployment at the institutional level.

By providing support for uptake and use of the RSpace institutional ERN and contributing to development and integration with existing and future tools and systems at UCL, ARC is supporting the endeavors of researchers while encouraging the research community to take bold strides in making data both FAIR and machine-actionable.

Impact & Outcomes

ARC's hybrid mission focuses on both providing and developing state-of-the-art integrated digital research infrastructure needed to drive computationally empowered science and scholarship at UCL.

ARC aims to computationally enable the research community both at UCL and beyond through the delivery of reliable and secure interconnected tools, services and infrastructure, as well as through continual innovation in the application of advanced computational and data intensive research methods.

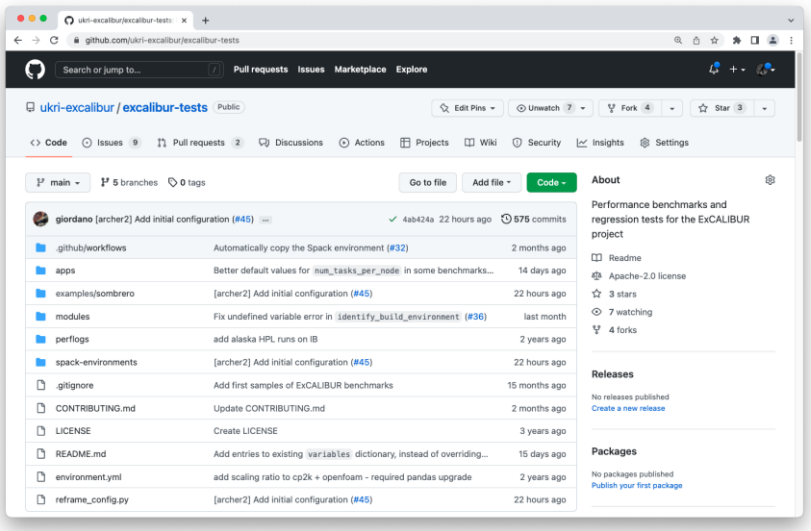
James Hetherington, Director of UCL ARC,
Rory Macneil, CEO of ResearchSpace, rmacneil@researchspace.com

ExCALIBUR Framework for Reproducible Benchmarks

T. Koskela[1], M. Giordano[1], I. Christidi[1], T. Deakin[2], C. Maynard[3,4], J. Quinn[1], D. Case[3]

1. UCL Centre for Advanced Research Computing 2. University of Bristol, … 3. University of Reading, … 4. Met Office, ...

Application benchmarking is a crucial activity in the UK's path to Exascale. It ensures future Exascale systems are understood by the community so that UK science applications, particularly those developed as part of ExCALIBUR, can take advantage of the scientific opportunities at Exascale. **Benchmarking commonly requires manual work and relies on knowledge possessed by a few individuals.** It is vital to rigorously measure the performance of benchmarks in a systematic way to enhance the transparency and enable reproducibility. UCL ARC is leading this collaboration of UK Universities to **provide the tooling to automate collecting and analysing benchmark data.** It provides a suite of benchmarks representative of the UK exascale science and a framework for configuring and running those benchmarks in diverse architectures, and post-processing the results.



How does it work

We use Spack[1] to build benchmarks, where possible, and ReFrame[2] to abstract interactions with the scheduler and the build system.

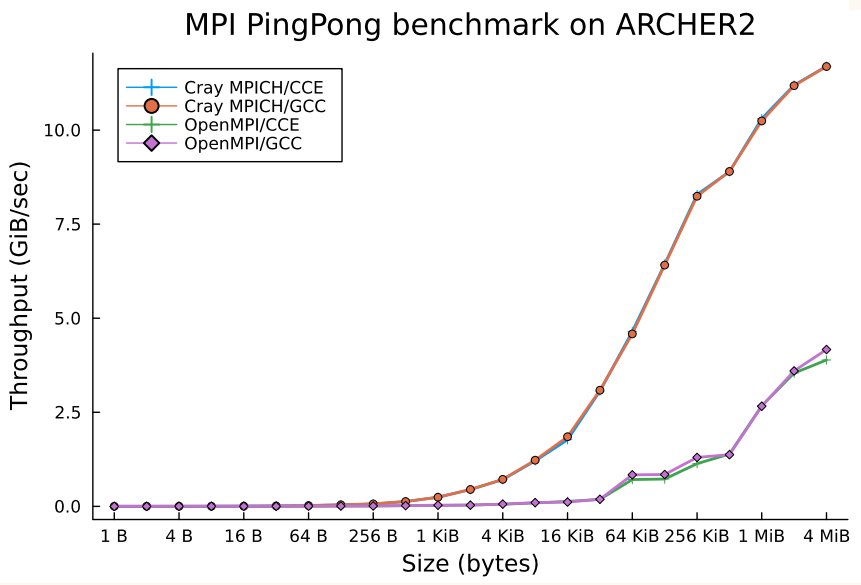
Who is it for

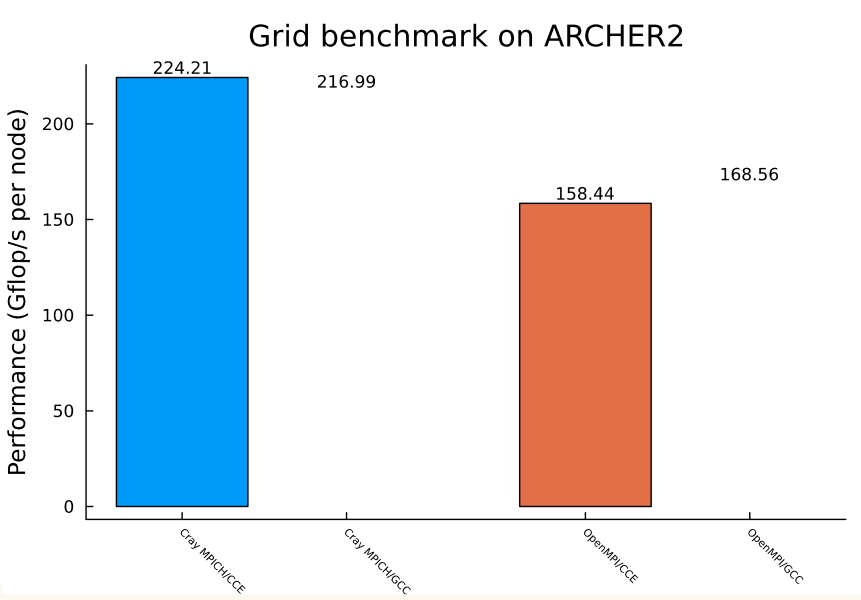
- Application developers, who want to evaluate the performance of their application on different systems
- System maintainers, who want to evaluate the performance of their system for different applications
- People who contribute to procurement decisions
- Vendors

Can I contribute?

Yes! We invite collaborators[3] to add

- Spack environments and ReFrame configs for new systems
- ReFrame benchmarks for scientific applications





For more information about this project, come to the BoF **ExCALIBUR : Exploring Successes, Collaborations and Community Building Opportunities with the UK Exascale Effort** Thursday, 17 November 2022 12:15pm - 1:15pm

[1] https://spack.readthedocs.io/en/latest/
[2] https://reframe-hpc.readthedocs.io/en/stable/
[3] https://github.com/ukri-excalibur/excalibur-tests/blob/main/CONTRIBUTING.md

Project PI: Tuomas Koskela: t.koskela@ucl.ac.uk
Project GitHub: https://github.com/ukri-excalibur/excalibur-tests
The UK Exascale project: https://excalibur.ac.uk/
UCL ARC: https://www.ucl.ac.uk/advanced-research-computing/advanced-research-computing-centre

SAX

Accelerated and differentiable Spherical Transforms in JAX

Contact: m.price.17@ucl.ac.uk

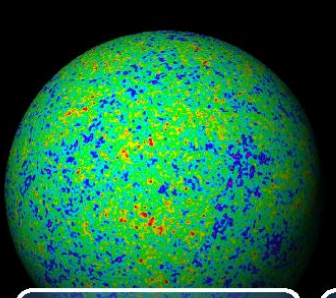
GitHub s2fft | Tests passing | docs passing | codecov 95% | License GPL

UCL

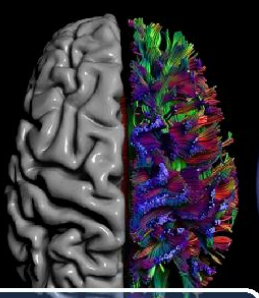
Abstract

We are re-engineering popular spherical image processing packages to provide new JAX implementation that will support (i) accelerators (*e.g.* GPUs/TPUs) and (ii) automatic differentiation. Our resulting SAX software suite will provide foundational tools on which next-generation machine-learning based spherical imaging techniques will be built (*e.g.* imaging, generative models, geometric deep learning).

Scientific Impact



Cosmology



Medical Imaging

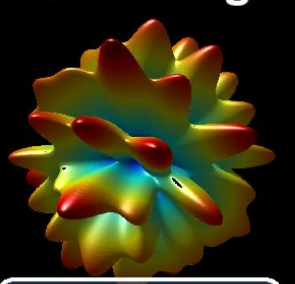
Geophysics and Climate Prediction

Our Development Process

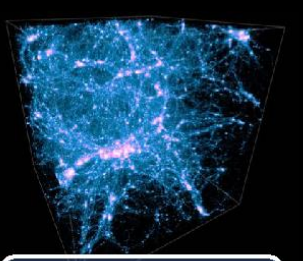
The SAX software is being developed in collaboration with Advanced Research Computing (ARC) situated at University College London (UCL). Their industry leading expertise will ensure SAX is cutting edge and of a professional quality. Additionally, throughout development the SAX suite will be open-sourced to encourage community interaction and adoption.

Differentiable Programming

Spherical Machine Learning



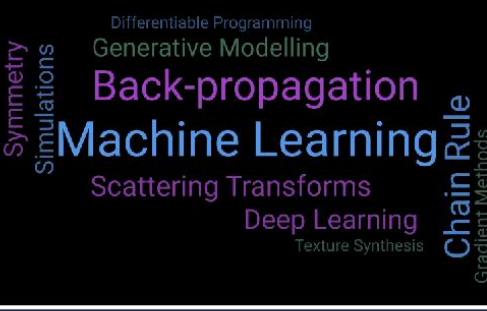
Wavelet Scattering Emulation



Differentiable Physical Simulations

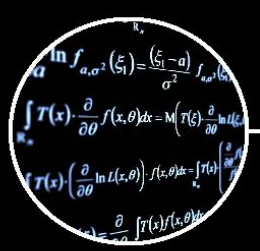
Critical Features:

Automatic Differentiation



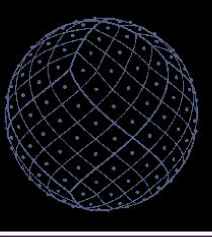
Hardware Acceleration

- Retrofit **existing** spherical packages for speed.
- Execute spherical transforms on GPUs/TPUs **without** writing low-level kernels.



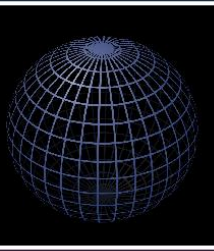
New Spherical Algorithms

Sampling Scheme Agnostic



HEALPix

+



McEwen-Wiaux

SAX:

Core Package

Stretch Goal

Spin-Spherical Harmonic Transforms → Wigner Transforms → Spherical Wavelet Transforms → Spherical Scattering Transforms → Extensions to the 3D ball

Matthew Price

Matthew Graham

Tobias Liaudat

Jason McEwen

JAX is a flexible Python library, developed by Google and adopted in-house by DeepMind, for high-performance scientific computing and machine learning based on numpy.

DYNAMOS & OptoCloud
Optical networks for HPC and Distributed Deep Learning Systems

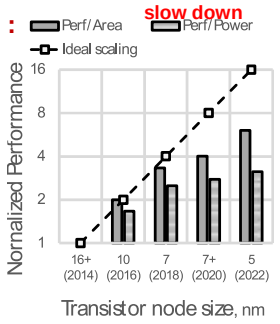
Alessandro Ottino, Joshua Benjamin, Georgios Zervas

Optical Networks Group, Department of Electronic and Electrical Engineering, UCL

alessandro.ottino.16@ucl.ac.uk

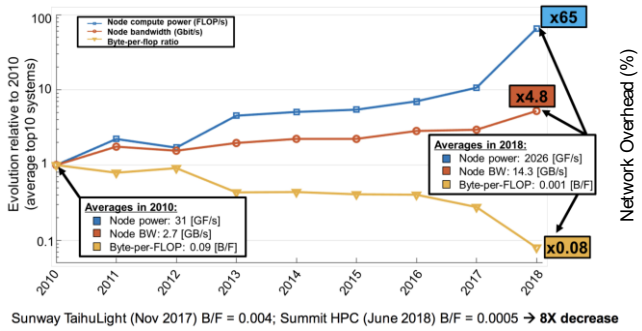
Motivation: Network, the key HPC/DDL system bottleneck

Problems: **Electronic Performance/Power slow down**



[2] Ballani H. et al., "Sirius: A Flat Datacenter Network with Nanosecond Optical Switching", SIGCOMM20

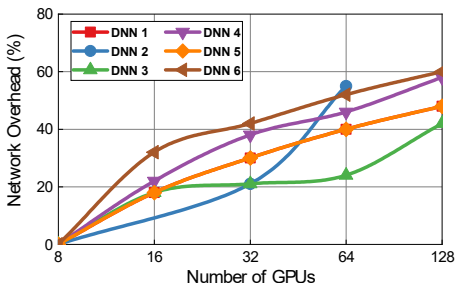
Bytes per FLOP reduction in HPC Systems



Sunway TaihuLight (Nov 2017) B/F = 0.004; Summit HPC (June 2018) B/F = 0.0005 → **8X decrease**

[1] Bergman K, "Empowering Flexible and Scalable High Performance Architectures with Embedded Photonics", IPDPS 2018

High network overhead on distributed ML training time



8] Wang W. et al. "TopoOpt: Optimizing the network topology for distributed dnn training", Arxiv 2022

Solution:

Optical Circuit Switching (OCS) can lead to

1. Exponential increase in capacity [2,3]
2. High performance/power [2]

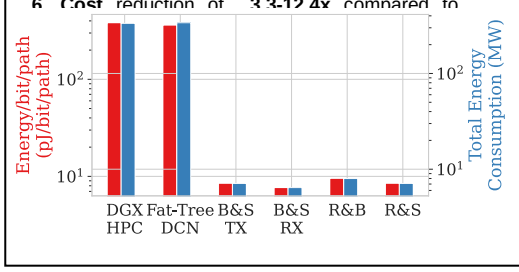
Challenge:

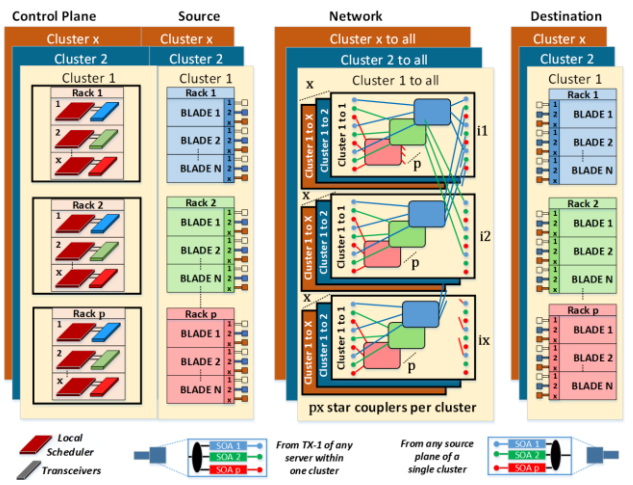
- **Scalable** network architectures
- **Fast** network control and switching
- **energy efficient** technologies
- **Better MPI** collectives

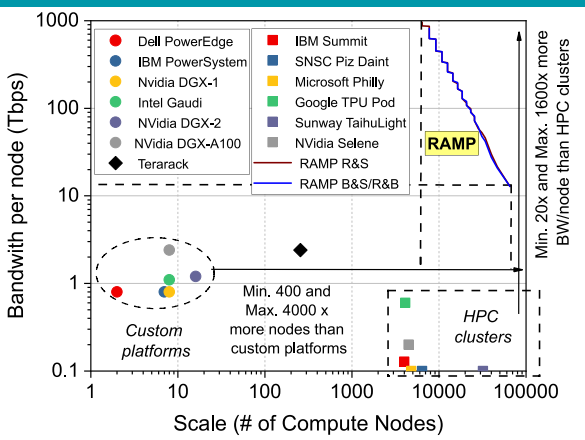
Ultra-fast Optical Circuit Switched Network Architecture

RAMP achieves:

1. **One hop, all-to-all port level connectivity.**
2. **65,536 nodes**
3. **0.84 Eb/s** system BW => **~ 0.1 Byte-per-FLOP** .
4. **<20ns circuit reconfiguration time.**
5. **>12.8Tbps** node pair capacity.
6. **Cost** reduction of **3.3-12.4x** compared to





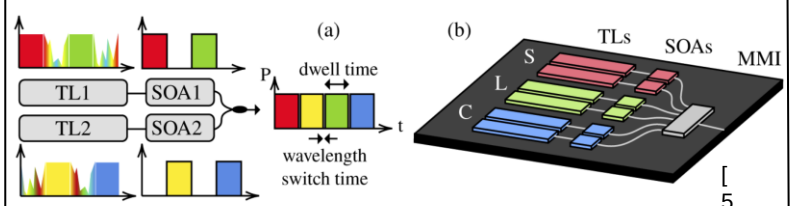


Comparison between RAMP and HPC/DDL systems. Fig. adapted from [3]

[3] Benjamin, J. L. et al., "PULSE: Optical Circuit Switched Data Center Architecture Operating at Nanosecond Timescales," JLT 38, 4906-4921, 2020

[4] Khani M. et al "TeraRack: A Tbps Rack for Machine Learning Training" 2020

Transceiver, Optical Switching and Scheduler



[5]

Ultra-fast transceivers:

1. **Wavelength switching (<1ns)** [5].
2. Tunability up-to **112 wavelengths** [5].
3. Integrated **sub-ns space switching** [6].

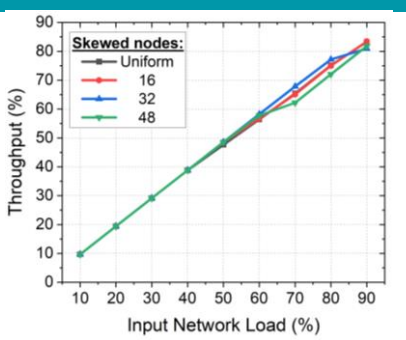
Passive Network Core:

1. **No power**
2. **No control**
3. **No cooling**

[5] Gerard, T. et al., "AI-optimised tuneable sources for bandwidth-scalable, sub-nanosecond wavelength switching," Opt. Express 29, 11221-11242, 2021

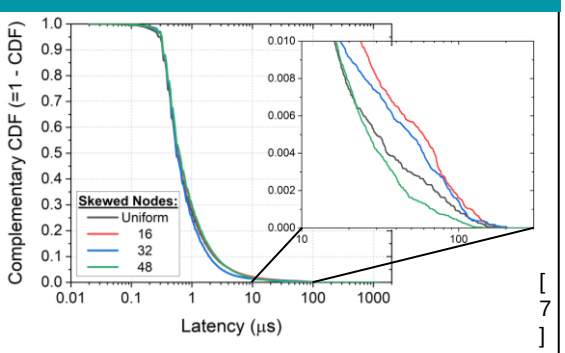
[6] Alkharsan H.et Al, "Optimal and Low Complexity Control of SOA-Based Optical Switching with Particle Swarm Optimisation" ECOC22

[7] Benjamin, J. L. et al., "Traffic Tolerance of Nanosecond Scheduling on Optical Circuit Switched Data Center Network", OFC 2022



Scheduler Achieves:

- **>80% throughput** for traffic @ **90%** network load.
- **Average latency under 2 µs at 90%** network load.



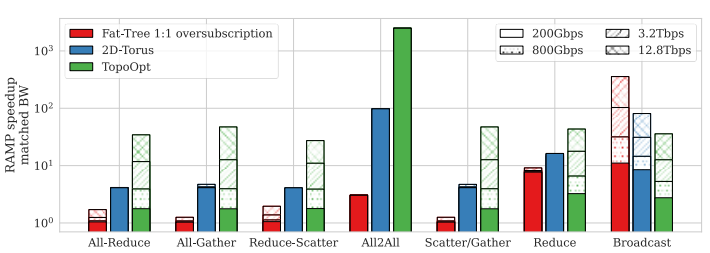
[7]

- **Median latency (at 0.5 CCDF) < 1 µs** in contrast to **47-447 µs** in EPS.
- **Tail latency 16-24 µs** compared to **76-1440 µs** in EPS network solutions.

MPI strategies and Application Speed-up

Novel MPI strategies co-designed with the architecture, which lead to:

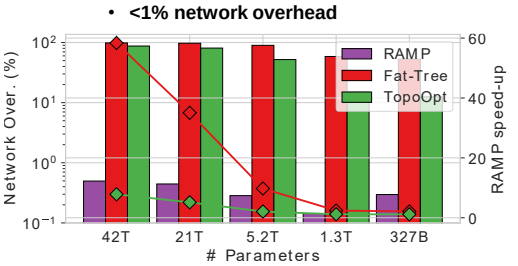
- **7.6-17x speedup** in MPI completion time wrt EPS and TopoOpt [8] on 1GB MSG.
- **1.04-2.24E3x speedup for matched bandwidth networks** at maximum scale.



Distributed Deep Learning application improvement at different scales:

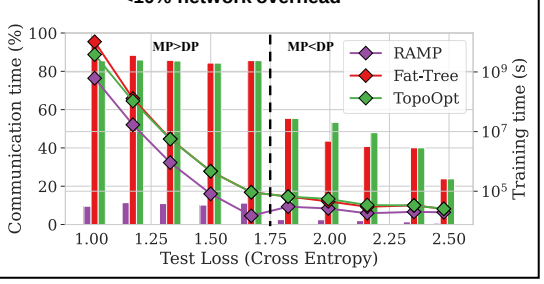
DLRM:

- **7.8-58x iteration time reduction**
- **<1% network overhead**



Megatron:

- **1.3-16.7x training time reduction**
- **<10% network overhead**



Horizon Europe 2021-2027

DYNAMOS

EPSRC Engineering and Physical Sciences Research Council

OptoCloud

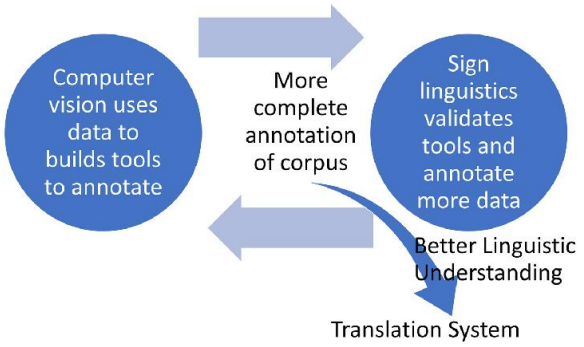
End to End Translation of British Sign Language

Neil Fox, Bencie Woll and Kearsy Cormier

UCL Deafness, Cognition and Language Research Centre

A partnership between sign linguists and computer scientists

- Computer vision is undergoing a revolution in terms of deep learning, breaking new ground
- Linguists want to better understand the language, its structure and use
  - achieved by recording and studying language data
- However, annotation of that data is extremely slow and laborious. Linguists need tools to help them automate the annotation process



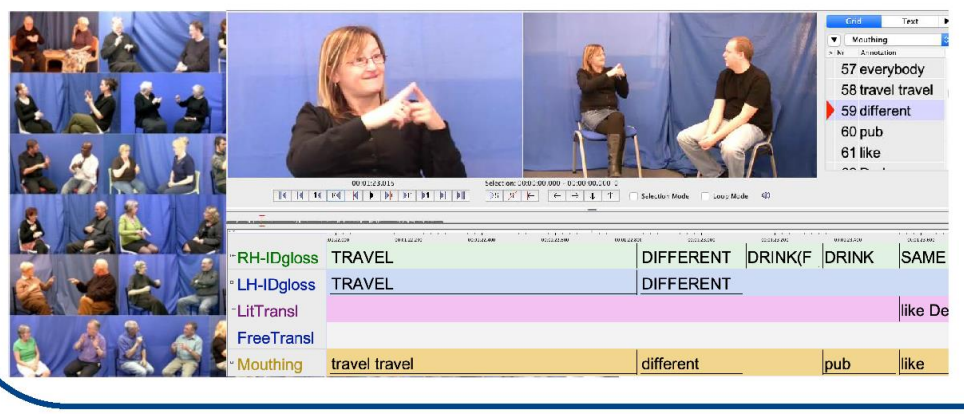
Research Questions

1. What linguistic annotation is needed for reliable machine translation from sign to spoken/written language?
2. What automatic annotation tools will provide the most utility to linguistic study?
3. How can we combine visual evidence from multiple articulators (hands, body, face) to achieve end-to-end translation?
4. What can machines do better than humans, and vice-versa?

Existing British Sign Language data

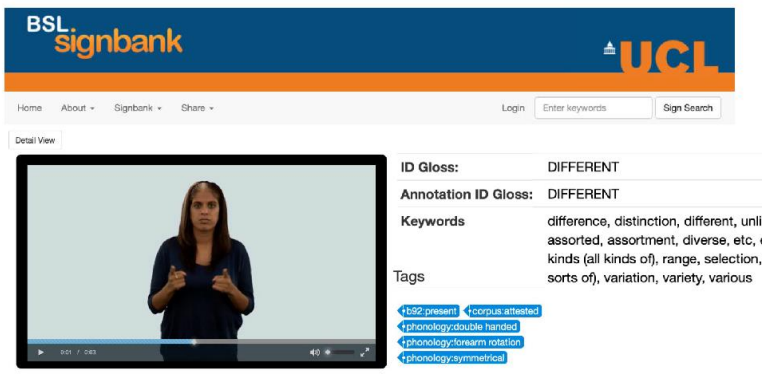
BSL Corpus

- Collection of ~125 hours of videos & metadata from 249 deaf signers from 8 UK regions: www.bslcorpusproject.org
- Videos online since 2011, initial annotations since 2014
- Annotation is SLOW – in 2022, 2/3 remains unsegmented



BSL Signbank

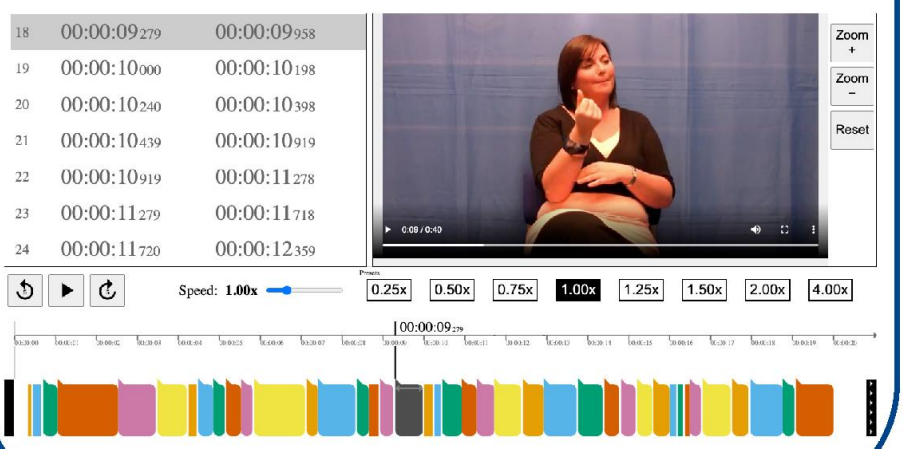
- Usage-based dictionary of BSL based on BSL Corpus
- Also lexical database, containing ID glosses and keywords
- Launched online in 2014 with 2500+ BSL signs & growing
- www.bslsignbank.ucl.ac.uk



Examples of resources provided by ExTOL project

Sign Segmentation Tool: VIA-SLA

- Auto segmentation of BSL signs
- Segmentation can be edited via online browser
- Reasonable performance in other signed languages
- Annotations can be exported to ELAN
- www.robots.ox.ac.uk/~vgg/research/signsegmentation/



BOBSL: BBC-Oxford BSL Dataset

- 1,400 hours BSL-interpreted BBC broadcast inc. English subtitles
- BOBSL v1_2 challenges for the ECCV SLRTP 2022 workshop
- Useful resource for researchers in the computer vision, natural language processing and sign linguistics communities
- www.robots.ox.ac.uk/~vgg/data/bobsl/

DCAL: Deafness, Cognition & Language Research Centre

Email: dcal@ucl.ac.uk Website: http://www.dcal.ucl.ac.uk

Real-Time Particle Filtering Data Assimilation

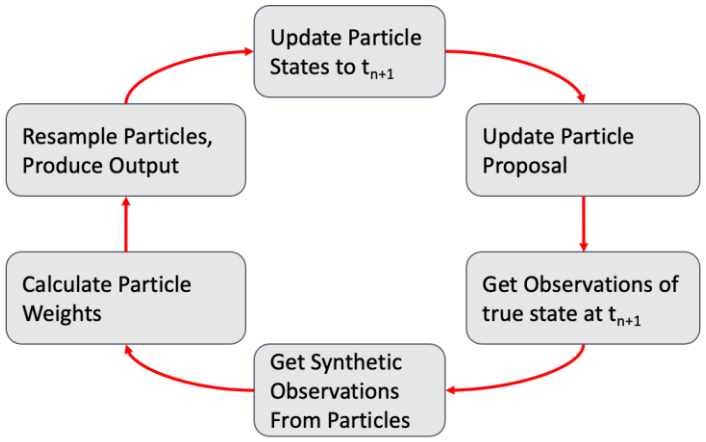
D. Giles[1], M. Graham[2], M. Giordano[2], T. Koskela[2], A. Beskos[1], S. Guillas[1,2]

1. Department of Statistical Sciences, University College London
2/ Centre for Advanced Research Computing, University College London

Abstract

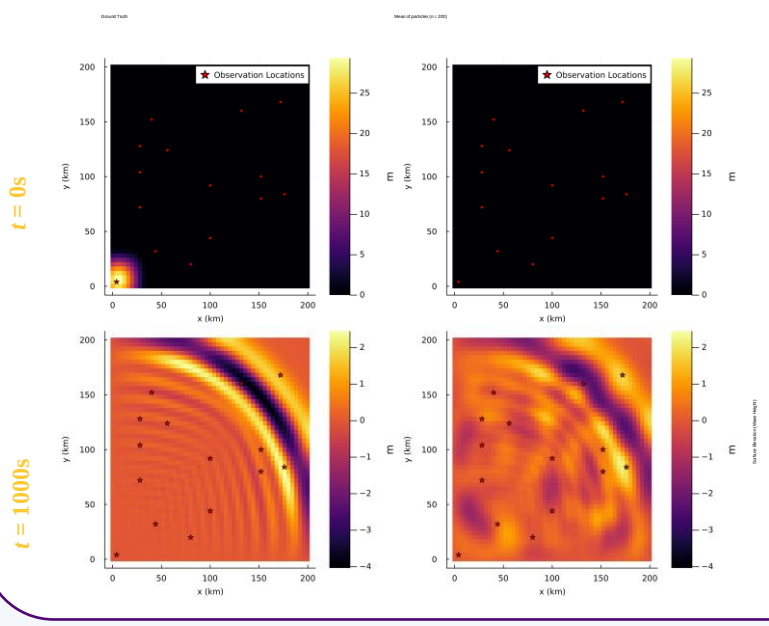
Data Assimilation (DA) is a technique for updating numerical model predictions with real-time observations. It is commonly used in e.g. weather forecasting. Particle Filtering is a DA method that makes very few assumptions about the uncertainty of observations, however it can easily become extremely computationally expensive. In this project, the RSE team at UCL ARC helped researchers in statistics to develop a **highly efficient Particle Filtering package in Julia**, optimized for parallel processing on HPC systems using MPI and multithreading. The package has been applied to tsunami and climate models, and is being generalised further to allow coupling with more applications.

Particle Fliter Algorithm

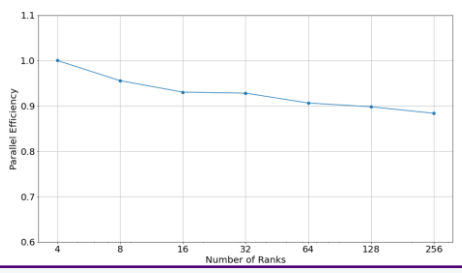


Tsunami Modelling

- $F(x_t)$ is an inbuilt solver of the linear shallow water equations
- Solved using a first order finite difference scheme (TDAC Model)



- Shared (threads) and distributed (MPI) parallelism
- Tests carried out using Tier2 CSD3 Peta4 KNL partition
- Performance scales well up to 260,000 particles, where we are using 256 ranks (4096 cores)



Atmospheric Modelling

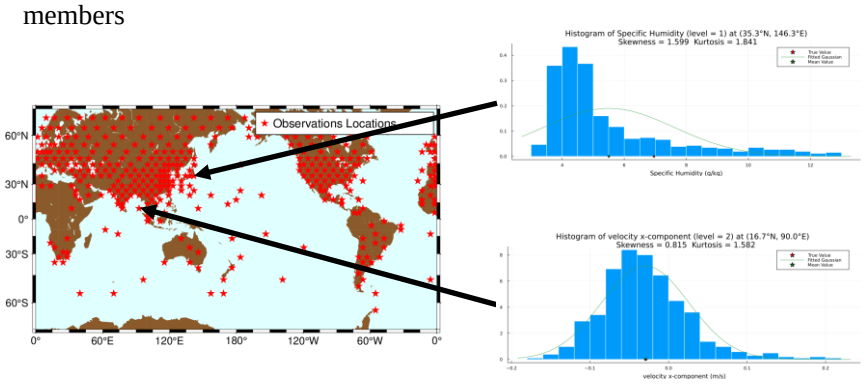
- Integration with SPEEDY, an atmospheric general circulation model which is written in Fortran
- Software considerations include:
  - Data transfer between the codes
  - Calling the SPEEDY model from Julia
- Prognostic variables: Zonal and meridional wind velocity components (u,v), temperature (T), specific humidity (q) and surface pressure (p)

Experimental set up

- Observations are generated from a nature run
- Spatial grid of dimensions: 96 × 48 x 8
- Observations of surface pressure at 416 locations
- Assimilating the surface pressure every 6 hours
- Number of particles (*n*) is 512
- Assimilation cycles: 125

Results

- Reduced errors when compared to a naive ensemble (not shown here)
- Existence of non-gaussian distributions with an ensemble of 512 members



ParticleDA.jl has developed to be **forward model agnostic**, ensuring that it is applicable to a wide range of modelling settings. Emphasis has been placed on numerical efficiency, scalability and optimisation for high performance computing frameworks. In the future, we plan to add localisation features (Couple with LETKF), integration it with more models and extend the optimal filter to unstructured meshes.

Project PI: s.guillas@ucl.ac.uk
RSE Contact: t.koskela@ucl.ac.uk
Project GitHub: https://github.com/Team-RADDISH/ParticleDA.jl