Memory Intensive Workflows in Scientific Computing

Background and Methodology

This report describes Work Package 1.5 of the DiRAC Federation Project, whose goal is engagement with computational scientists in other fields (biomedical, climate, environmental, engineering, materials, energy) to share and exchange insight and experience, and in particular to explore the extent to which common workflows can be the basis of defining UKRI-wide computing services in a future UKRI Digital Research Infrastructure. Following consultation with senior figures within DiRAC, individual scientists in these domains were approached on the basis of their expertise and/or standing, and then invited to one of two one-day workshops held in central London during February 2022. This report covers the first workshop, held on 9th February, focussing on "memory intensive workflows'', by which is signified memory-bound problems exemplified by computational cosmology, currently served by the *COSMA* facility in Durham.

Due to the prevalent COVID situation, not everyone was able to attend the meeting in person, so the event was run in hybrid mode, with remote participants attending via Zoom, and questions for speakers submitted via a web-based interface.

In advance of the workshop, participants within a particular discipline were organised into groups and invited to collaborate on presenting and summarising the state-of-the-art in their particular field in a specified timeslot during the morning of the workshop, prompted by the question "what kind of machine is ideally suited to your problem?". The participants and groupings are listed in Appendix A. Stress was laid on the importance of explaining the problems being tackled to a non-expert, if informed and receptive, audience, stressing computational techniques, challenges and bottlenecks. Scientific quality, competitiveness and importance was understood; speakers were urged not to spend precious time underscoring those aspects. Presentation slides are available to view at www.dirac.ac.uk/federation-project-workshops-wp1-5/.

During the afternoon, participants broke up into groups to consider and discuss particular issues of interest. In the first half of the afternoon the questions were scripted, but participants were given the opportunity to propose their own questions, and the most popular (on the basis of a web-poll) were discussed in the second half of the afternoon, with participants free to join whichever group best matched their interests. All questions considered are listed in Appendix B. Responses were collected, again using a web-based tool, and form the basis for the discussion summary in the second part of this report.

Scientific Fields - State of Play

Cosmology

- Model the evolution of the Universe from well-understood initial conditions based on observations of the Cosmic Microwave Background to the present epoch. The core physics input is General Relativity, the Standard Model, and Λ-CDM. Beyond GR, models may be supplemented by hydrodynamics, magnetic fields, radiative transfer, and cosmic rays. Output questions addressed include galaxy morphology, development of super-massive black holes, star formation, metal abundance, and dust formation.
- State-of-the-art simulations may contain 8×10^{12} particles (Euclid). There are several codes for GR evolution, yielding consistent results. GR is *long-ranged*, hence requires keeping the whole system in memory with all-to-all communication, and *attractive* so care is needed to control accumulating error. Particular challenges are exerted by huge dynamical ranges: 10^{10} (length scales), 10^{12} (time), 10^{20} (density).
- Handling dynamic length scale ranges requires techniques such as adaptive mesh refinement, particle splitting in mesh-free hydrodynamic codes, semi-Lagrangian techniques, and codes capable of focussing compute resource on interesting regions. The time-evolving nature of the

problem requires techniques such as task-based parallelism, genetic algorithms and per-particle (ie. local) timestepping. Often program logic reorganisation (eg. from N^2 loops to tree-code) results in faster time-to-solution at the expense of scalability. Data movement is an important bottleneck. State-of-the-art projects require O(1PByte) fast disk storage, enabling fast checkpointing with one file output per MPI rank. It is anticipated that new numerical algorithms are crucial in order to enter the exascale era.

Chemistry and Materials Science

- Study and modelling of materials has a plethora of applications in both pure and applied science: structure prediction, phase diagrams and phase transitions, material failure and crack propagation, study of defects and radiation damage, protein-protein interactions, enzyme catalysis, drug design, interactions/chemistry at interfaces such as molecular adsorption, photocatalysis. There is a huge variation in length/timescales: nm/fs for molecules to m/s for corrosion studies.
- The approaches used depend on the scale of the problem: Density Functional Theory (DFT) is an *ab initio* method solving the electronic Schrödinger equation; atomistic and Molecular Dynamics (MD) methods use empirical forces to model mechanical and dynamical properties; more coarse-grained problems tackled using continuum approaches such as Finite Element Method (FEM) such as Lattice Boltzmann for fluid simulation.
- DFT employs an expansion of N_e~O(10^4) individual electron wavefunctions over a N_b-dimensional basis, with N_b~O(10^3-10^6), requiring storage/handling of $N_b(N_b+N_e)$ variables. Depending on the problem the basis set may be local (e.g. CRYSTAL, NWCHEM, GAMESS, FHI-Aims, Gaussian) or plane wave (hence not tied to atomic positions, e.g. CASTEP, VASP, ABINIT, Quantum Espresso). Some efficiencies follow if the resulting matrices to be diagonalised are sparse (eg. CP2K, bigDFT, ONETEP). More accurate *ab initio* methods (eg. Questaal Suite, Coupled Cluster) are applicable for smaller systems, and there are also QM/MM codes embedding quantum approaches within a classical molecular framework (ChemShell).
- MD simulations integrate equations of motion for O(10^6) particles, interacting via both short and long-ranged forces. Typical MPI codes suffer from memory bandwidth issues.

Fluid Dynamics

- Many applications in science and engineering. CFD solves a non-linear differential equation, e.g. Navier-Stokes. There are several distinct physical regimes: compressible; incompressible; multiphase; hypersonic. The associated physics or geometry may also present challenges.
- Four distinct computational approaches:
  - Finite Difference – high order methods based on structured grids, permitting high-order methods needed to capture turbulence. A compact stencil is suitable for effective parallelisation and/or accelerated architectures. Example: Xcompact3d
  - FEM/Finite Volume methods — flexible discretisation allows complex geometries needed for engineering applications. Solution is based on iterative application of large sparse matrices. Permits inclusion of complex physics packages. Example: Code_Saturne
  - Spectral methods, exemplified by hp-FEM (*h* is discretisation scale, *p* polynomial degree) achieve exponential convergence, with cost ~p^4, accuracy h^p. Example: Nektar++
  - Smoothed Particle Hydrodynamics (SPH—a meshless approach in which particles bearing properties such as density move along trajectories governed by dynamics, and fluid properties calculated via a weighted average using a kernel function.
- Implementations
  - Simple mesh-based codes can exhibit near-ideal scaling up to ~500k cores.
  - The arithmetic intensity of hp-FEM increases with *p,* so flop- rather than memory bandwidth-limited. FVM/FDM show similar trends. However higher-order methods fail to capture shock behaviour.

- SPH with a compact kernel support ideally suited to implementation on many-core GPU. Many inter-particle interactions (20-50 in 2D, 100-400 in 3D) creates a significant memory overhead. Real world problems require 10^8—10^{11} particles and often complex physics, hence many GPUs, or fewer particles/slower simulations. Multi-GPU simulations must address data orchestration issues via scalable offloading.

Neuroscience

- Goal is to model the human brain and brain function based on images obtained using Magnetic Resonance Imaging. MRI scan data highlight different body tissue content (e.g. T1 fat vs. T2 fat and water), exist in a variety of formats, and are subject to imaging artifacts. Datasets also capture temporal variation (fMRI) and may contain > 10^6 individual scans.
- The main approach is Machine Learning, either classical (statistical parametric mapping) or via deep neural network models. Typical deep 3D CNNs have 40M parameters with a 32GByte memory footprint.
- Work is performed on GPU clusters, using software such as Pytorch (Torchvision, Monai), and Dockers. There is a tension between the need to increase the dimensionality and resolution of the latent space, and/or data augmentation, and compute cost/memory bounds. Possible solutions under investigation include the use of cloud, mixed precision, invertible networks, and better application programming interfaces.

Climate and Meteorology

- Advances in global weather forecasting and climate modelling are characterised by increasing spatial resolution, which has a doubling time of 24 months (cf. the 18 months of Moore's Law). Ensemble forecasting, required to sample a range of initial conditions, produces 900 GB per hour, or 300 TB for a 15-day forecast. A 50-member global ensemble at the current state-of-the-art 9km resolution requires 1000 sustained Tflops. Future needs are driven by the high volume and source variety of input satellite data (200 TB/day in 2022), and the high volume of simulation data (9×10^9 field points @ 9km, 352×10^9 @1km)
- New tools in climate modelling: digital twins (e.g. NVidia's Earth-2) employ AI data pipelines. Data volumes for modelling weather extremes/hazard management and climate change adaption will rise from 105 TB/day (2200 TB/year online storage) in 2022, to 1300 TB/day (8230 TB/year) in 2024.
- Dealing with these volume and data-handling challenges requires memory-intensive workflows utilising both high performance *and* cloud computing. "Information density" within data is dynamic; smart tools, both algorithms and hardware, are needed for switching precision for optimal efficiency. Programming orthodoxy suggests minimising frequency of MPI messaging; recent studies suggest rather that more frequent, but well-timed data movement is more effective. Depending on the nature of data and the operations required it might even be more cost-effective to perform redundant computation, or to have compute units lying idle some of the time.
- An outstanding and key question: who will write the software needed for such new memory-centric machines?

Plasma and Fusion Science

- UKAEA uses computer modelling to understand plasma physics, design and test reactor components, and improve fusion performance. Challenges include modelling electromagnetic forces, large heat and neutron fluxes, complex actively-cooled components, and new materials. Length scales range from atomic dimensions to full-scale CAD models of the machine, and from individual particle interactions at GHz frequencies to MHD instabilities at O(1Hz).
- CAD for machine components is done manually and serially, resulting in large datafiles. A typical application is neutronics simulations (using OpenMC code); this memory-intensive approach

assigns one CAD file to each MPI process. Realistic simulations of the ITER fusion experimental facility manifest strong scaling, with the best performance from large-memory nodes.

- Plasma simulations use semi-implicit codes requiring the inversion of a large sparse matrix (e.g. GYRO, JOREK). Again, it is found that the best performance as spatial resolution is increased results from high-memory nodes. Machine learning approaches employing large image databases are being explored; this has potential to reduce processing time for 3TB files from 15 minutes to a few seconds

Discussion Topics

Common Features of Workflows

All participants were asked to discuss this topic. Generally, workflows were found to fall into two camps: those that are “memory-intensive” in that they require large memory capacity, and those that are memory-intensive in that they require high memory performance (bandwidth or latency). Computational cosmology, the use case defining the DiRAC Memory Intensive Service, falls into the former camp. Machines specified for one use case will not be an ideal fit for the other, so greater precision in this definition will be important when defining services, and when organising future workshops.

A common feature among many fields is the need for code coupling, which was then discussed separately; see the full discussion below.

It was identified that many use cases represented at the workshop could be described as smooth-particle hydrodynamics (SPH). However, the relative computational requirements of SPH vary with length and time scales being studied, so an “SPH label” may be insufficient to define common requirements.

Access to technical expertise

The emergence of the Research Software Engineer (RSE) as a career pathway has helped, but not solved the problem of researchers and research groups having access to technical expertise to support their computational research. For technical expertise to be retained within a group—i.e. to retain a “standing army” of RSE skills—individual RSEs must be retained for long periods, or they need to train their replacements and ensure that there is knowledge transfer. The most prevalent current model of recruiting RSEs on short-term contracts with project funding is not conducive to either of these. A number of prerequisites to solving this problem were identified. Firstly, the perception of domain-specific RSE as being a “failed academic” must be dispelled; modern computational research requires more skills than any one person can be expected to have, and those holding the technical skills to enable the research to be conducted within the available computational resources are no less important than those providing other intellectual contributions. Secondly, and relatedly, software must be viewed as a first-class research output; so long as software is viewed as secondary to papers, then priority will be given to retaining staff who enable the latter, at the expense of the former, and technical expertise will be lost. The distribution of RSE funding across different areas of science is currently uneven, so EPSRC-aligned science is blessed with many RSE Fellows, while many other disciplines have none.

Data

Participants identified the need to be precise about the distinction between data management (e.g. FAIR data), data retention, data exhibition (e.g. open data), and data curation (ensuring that data remain readable on long timescales). Each of these is a specific skillset, and each project must decide which elements are required. An observation was made that while funding for work in these areas is written into grant applications, lack of ring-fencing or auditing means that funds can be redirected to other aspects of the project work, resulting in a token effort made addressing data-related issues.

Around data retention and publishing, for many memory-intensive workflows, data retention of all output data would be very expensive, and since maintenance would be required outside of the funded period to keep data available, would not be fundable by project funding. As such, a different model would be needed to fund maintaining availability of data. The push for digital twins is likely to further increase the volumes of data that must be stored.

It was however observed that where large simulation outputs have been made available, for example in the *Millennium* cosmology simulation, simulations that would otherwise be comparable or smaller in scale than contemporary competitors can generate significantly more outputs, as many researchers who do not have access to the computational resource to generate such a simulation are able to perform analysis on it.

Some participants had concerns around misinterpretation of data made public; the point was however made that the same could be said of published papers.

The challenges of having a single data policy that is appropriate for many science areas were raised—some disciplines deal with the most sensitive of personal data which cannot be published under any circumstances, potentially even with anonymisation, while for others this is not a problem, and in simulation-based fields the effort to store the data may become larger than that required to regenerate it from scratch within a few years of the work being published.

New architectures

Given the variety of architectures becoming available, it's likely that different parts of a workflow may perform better on different architectures; machines may need to be over-specified in order to facilitate maximal performance. Failing this, in a heterogenous machine, job schedulers will need to be more intelligent in order to allocate job-steps to the right resource at the right time. The software effort required is likely to be large, comparable to or greater than that for the transition to GPUs—which itself in some disciplines is still ongoing. It's possible that programming models such as OpenMP and OpenACC may make this easier.

GPUs

GPUs provide a significant improvement in compute intensity and energy efficiency over more traditional CPU-based compute, but for memory-intensive workflows these improvements can be difficult to achieve. Since GPUs have significantly less memory (80GB maximum for current-generation hardware) than per-node RAM (1TB on COSMA8), in many cases communication time between device and host, or between devices, will dominate, giving a slowdown rather than a speedup. That said, aspects of problems that are less intensive on memory capacity can and are deployed on GPUs; this includes in coupled or multi-physics problems, where a less memory-intensive part of the problem can be offloaded to a GPU while the memory-intensive aspect runs—either concurrently, or sequentially—on the CPU with the larger system RAM. For example, machine learning tasks typically perform very well on GPU; if a workflow includes a machine-learning element then this can be offloaded. In other cases, a focus on vectorisation over GPU translation would be likely to give better performance gains. Developments from vendors in increasing the memory per GPU, and removing the overhead associated with host-device transfers, may make GPUs more deployable for the largest memory-intensive problems.

Training needs

There is a general agreement that new researchers are unfamiliar with computer architecture, knowledge of which is vital to be able to develop computational research software that is performant on modern hardware. This is becoming more noticeable as PhD students are recruited from more diverse backgrounds. The fact that RSEs are increasingly taking on the computational aspects of work can reduce this pressure, but it isn't clear whether non-RSE researchers can reasonably delegate all knowledge of architecture to RSEs. How and when this knowledge should be imparted (to researchers or RSEs) is not clear— suggestions included teaching it as part of undergraduate courses, as well as more traditional training courses for PhD students or postdocs.

Cloud HPC

There are more challenges to using commercial cloud providers for HPC than for more typical cloud workflows. Experience from the US was reported, where researchers tried using Google Cloud and had a negative experience due to unexpected costs. Using commercial cloud providers in an academic context requires significantly more paperwork and administration than in other sectors, in order to manage billing and budgeting arrangements, and discussion participants had not encountered a way to manage this. Additionally, there is a lack of unbiased information; most information given to institutions comes from vendors, who do not have an unbiased perspective.

Resource allocation

Most HPC resources have more demand for their time than they have capacity; as such, a process for prioritisation of access is needed. There was a general agreement that decision regarding the science must be deferred to the science community in question. Regarding technical aspects, there is a question of how to define efficiency for the purposes of prioritisation. Measuring "science per unit compute time" is most likely to match the priorities of funders, but "amount of science" is relatively unquantifiable, and researchers are less pressed to make efficient use of the machine in terms of the parallel efficiency of the algorithm and implementation. Measurement of parallel efficiency is relatively easier, but in some cases the algorithms with the shortest time to solution are less parallelisable than those that take longer, so prioritisation of parallel efficiency would in fact incentivise spending more compute time than necessary. In many cases it is possible to prototype workloads on smaller local facilities before moving to larger national machines, but this is not always true; some disciplines represented have huge requirements even for newcomers, and in other cases the performance characteristics of local machines aren't the same, so testing on larger machines to understand the performance of the algorithms and software is necessary before science-delivering work can commence.

Code coupling

Increasingly, research topics of interest (e.g. in weather and climate, fusion, aeronautics, materials science) require multiple phases of computation that are each sufficiently complex to have their own software ecosystem; there is broad interest in how to enable coupling of different codes together. In low-performance applications manual conversion of data on disk is sufficient to enable this, but at larger scales the time and storage requirements of this approach can be prohibitive. Having a common data format and common memory layout for classes of data that are shared between multiple software communities is likely to facilitate this, but it is not clear how such a format could be decided upon—whether one can be imposed *a priori*, decided by committee, or whether one can emerge from a particular community's work coupling their own software. Having appropriate standards for metadata tagging and data storage is likely to be invaluable in ensuring that the output of coupled software remains robust.

Summary

Rather than attempt a narrative synthesis of the many topics and themes explored in the workshop, we summarise our findings in the form of a table with one column for each science domain and one row for each computational issue or workflow characteristic. The results of the companion workshop on Extreme Scaling Workflows are also included (right-hand panel). The corresponding element is populated based solely on the documentary evidence from the workshop; absence of an 'x' need not imply that a particular issue is not important or characteristic of a particular domain. Some overarching conclusions are clear, however: distributed-memory parallelism as a means to high performant computational science is almost ubiquitous, there is a clear direction of travel towards running high arithmetic intensity simulations on GPUs, and memory capacity and bandwidth, and storage capacity will continue to be significant bottlenecks.

	Cosmology	Chemistry/Materials MI	Fluid Dynamics	Neuroscience	Climate/Meteorology MI	Plasma/Fusion MI	Lattice QCD	Chemistry/Materials ES	Biomolecular	Climate/Meteorology ES	Plasma/Fusion ES
Unstructured mesh		x	x			x					x
Adaptive mesh refinement	x										
Mesh-free method	x	x	x					x			
Memory capacity constrained	x	x		x	x	x			x	x	x
Memory bandwidth/latency constrained	x	x	x		x			x	x	x	
Disk I/O constrained	x				x		x			x	
GPUs	x		x	x	x		x	x	x	x	x
Machine learning/artificial intelligence				x	x	x				x	x
Cloud				x	x						
Smart data movement/mixed precision	x			x	x						
Large sparse matrix operations		x	x			x	x				
Ensemble forecasting/time parallelism					x				x	x	x
Time critical					x					x	
Task-based parallelism	x										
Code coupling		x	x		x			x	x		
Workflow management system				x	x		x	x	x	x	
SIMD/SIMT							x				
Whole-machine via trivial parallelism				x				x	x		
Training needs					x			x	x		
Distributed-memory parallelism	x	x	x		x	x	x	x	x	x	x
Storage-constrained	x	x	x	x	x	x	x	x	x	x	x

It remains to thank the workshop participants for their willingness to rise to the challenge of an unusual exercise, before an unfamiliar audience, in some cases with previously-unmet collaborators; for generously sharing their time and their knowledge (both precious commodities); and finally for providing invaluable feedback for early drafts of this report. Needless to say, any remaining inaccuracies are entirely our responsibility.

Ed Bennett (STFC RSE Fellow)
Simon Hands (Community Development Director, DiRAC)

Appendix A: Names and affiliations of participants

Cosmology:	Alastair Basden	U. Durham
	Aidan Chalk	STFC Hartree Centre
	Matthieu Schaller	U. Leiden
	Debora Sijacki	U. Cambridge
Chemistry & Materials:	Ian Bush	STFC Scientific Computing
	Peter Coveney	UCL
	Sergei Dudarev	U. Oxford
	Alin-Marin Elena	STFC Scientific Computing
	Phil Hasnip	U. York
	Scott Woodley	UCL
Fluid Dynamics:	Stephen Longshaw	STFC Scientific Computing
	Benedict Rogers	U. Manchester
	Spencer Sherwin	Imperial College
Neuroscience:	Robert Gray	UCL
	Parashkev Nachev	UCL
Climate & Meteorology:	Nils Wedi	ECMWF
	Tobias Weinzierl	U. Durham
Plasma & Fusion Science:	Rob Akers	UKAEA
	Andy Davis	UKAEA
	Shaun DeWitt	UKAEA
	Stan Pamela	UKAEA
	Michael Ball	BBSRC
	James Richings	U. Edinburgh
DiRAC:	Ed Bennett	U. Swansea
	Simon Hands	U. Liverpool
	Mark Wilkinson	U. Leicester
	Jeremy Yates	UCL
	Laura Pecorone	
	Alastair Williams	

Appendix B: List of discussion questions (scripted and unscripted)

- Which aspects of the other workflows do you recognise? Is MI the right model?
- What can we learn from other fields?
- How is your community managing/planning the transition from petascale to exascale computing?
- What are the storage requirements of your field? To what extent is this a bottleneck/inhibitor for progress?
- How are you responding to the growing requirement for effective data management/curation?
- How would your community ensure the best science is given priority access to resource? How might a peer review process look?
- What features of cloud services would you expect to find useful?
- Will you be using Machine Learning/Artificial Intelligence to advance your research?
- Do you see an advantage in migrating from CPUs to GPUs? How are you managing the transition?
- What role do you foresee for emerging architectures, e.g. FPGAs, IPUs, Wafer Scale Engines?
- Do you foresee any role or need for quantum computing and/or quantum algorithms?
- Do you have sufficient access to technical expertise, e.g. RSE support, to pursue your goals effectively?
- What training needs does your research community have?