

**The
Alan Turing
Institute**

**Matching AI Research to
HPC Resource through
Benchmarking and
Processes**

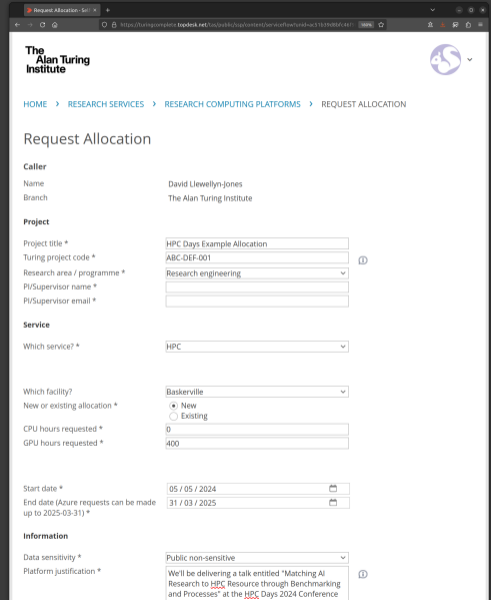
David Llewellyn-Jones, Tomas
Lazauskas



The Problem

Matching Projects to Systems

1. Many diverse users and projects
2. Many diverse systems and characteristics
3. Researchers aren't familiar with the system
4. Research Computing isn't involved with the project
5. Communication is through ticketing system



The screenshot shows a web browser window displaying the 'Request Allocation' form. The browser's address bar shows the URL: <https://turingexplorer.topdesk.net/You/0488/Sup/center/Service/nextord=431819056/481/>. The page header includes the Alan Turing Institute logo and a navigation menu with links for HOME, RESEARCH SERVICES, RESEARCH COMPUTING PLATFORMS, and REQUEST ALLOCATION. The form is titled 'Request Allocation' and is divided into several sections: 'Caller', 'Project', 'Service', and 'Information'. The 'Caller' section shows the user's name as David Llewellyn-Jones and their branch as The Alan Turing Institute. The 'Project' section includes fields for Project title (HPC Days Example Allocation), Turing project code (ABC-DEF-001), Research area / programme (Research engineering), and fields for PI/Supervisor name and email. The 'Service' section includes a dropdown for 'Which service?' (HPC), a dropdown for 'Which facility?' (Baskerville), and radio buttons for 'New' (selected) or 'Existing' allocation. It also has input fields for 'CPU hours requested' (0) and 'GPU hours requested' (400). The 'Information' section includes a dropdown for 'Data sensitivity' (Public non-sensitive) and a 'Platform justification' field with the text: 'We'll be delivering a talk entitled "Matching AI Research to HPC Resource through Benchmarking and Processes" at the HPC Days 2024 Conference'. The browser window also shows standard navigation icons and a search icon.

Request Allocation

Caller

Name: David Llewellyn-Jones
Branch: The Alan Turing Institute

Project

Project title *: HPC Days Example Allocation
Turing project code *: ABC-DEF-001
Research area / programme *: Research engineering
PI/Supervisor name *:
PI/Supervisor email *:

Service

Which service? *: HPC
Which facility? *: Baskerville
New or existing allocation *: New Existing
CPU hours requested *: 0
GPU hours requested *: 400

Start date *: 05 / 05 / 2024
End date (Azure requests can be made up to 2025-03-31) *: 31 / 03 / 2025

Information

Data sensitivity *: Public non-sensitive
Platform justification *: We'll be delivering a talk entitled "Matching AI Research to HPC Resource through Benchmarking and Processes" at the HPC Days 2024 Conference

A Diverse Institute

1. Over 400 researchers
2. Data science, machine learning, AI
3. Grand challenges
 - 3.1 Defence and national security
 - 3.2 Environment and sustainability
 - 3.3 Transformation of health
4. Digital society and policy

The screenshot shows the Alan Turing Institute website with a grid of 12 research project cards. Each card features a representative image, a title, and a brief description of the project.

Project Title	Description
Enhancing critical ecosystems	Using data analysis to enhance critical ecosystems like cities and farms, and the digital-physical systems that support them
Evaluating homomorphic encryption	Exploring different ways of encrypting sensitive data that can allow for secure, outsourced computation in the cloud
Complexity twin for resilient ecosystems	Understanding the resilience of the UK's critical infrastructure ecosystems, to develop investment strategies and improve safety
Distributed training for machine translation	Training neural networks, and developing related hardware, to be better at translating millions of words of online text
Data-driven nuclear management	Developing data-driven decision support systems to enable faster, more effective decision making for nuclear engineering operations
Disaster management	Developing novel machine learning approaches to data fusion, to aid with disaster management policy and response
Computational modelling of civil wars	Simulating and modelling civil conflicts in a data-driven way, to understand the dynamics of these events
Counterfactual fairness	Making algorithm-led decisions fair by ensuring their outcomes are the same in the actual world and a 'counterfactual world' where an individual belongs to a different demographic
Co-designing algorithms and computer architecture	Designing hardware to suit the needs of data science algorithms, which will similarly be designed to suit the capabilities of the hardware
Capturing complex data streams	Describing complex sequences of data from different sources, to gain insights and generate meaningful actions in a range of applications
London air quality	Developing machine learning algorithms and data science platforms to understand and improve air quality over London
Cancer pre-diagnostic analytics with AI	Creating a digital repository of a variety of tumour and immune cells and developing algorithms to recognise these cells automatically

Navigation: First Previous ... 13 14 15 16 17 18 19 20 21 Next Last

Core Capabilities

1. Research software engineering capability

Growing our core research software engineering capability to continue to contribute skills in research software engineering and data science in support of national priorities.

2. Open-source infrastructure

Expanding our work in the development and provision of open-source infrastructure that is accessible to all.









Our Approach

Our Approach

Four-pronged approach

1. Knowledge base and training
2. Structured onboarding
3. Trial access
4. Embedding in projects



Knowledge Base and Training

1. Walkthroughs
2. Periodic training
3. Developing benchmarking results

Walkthroughs

1. Most available systems have excellent docs
2. System-specific, but can't possibly cover all tools
3. The Turing has a narrower focus
4. Different tools have (mostly) excellent docs
5. But rarely HPC-specific (let alone system-specific)

```
lightning.pdf 100.0%
```

Configure the accelerator for use with XPU

```
# XPUAccelerator must be imported before PyTorch or Lightning
import xpuaccelerator as xpu

# Import OneCCL bindings for PyTorch
import onecccl_bindings_for_pytorch

# Import intel_extension_for_PyTorch
import intel_extension_for_pytorch as ipex
```

You'll need to copy the `xpuaccelerator.py` file somewhere Python can find it. For example, if you're including the project directory using `pip install -e .` for example, then you can copy the file directly into project's root directory.

Configure precision and callbacks

```
if torch.xpu.is_available():
    torch.set_float32_matmul_precision("high")
```

Optional callbacks for use on XPU

```
if torch.xpu.is_available():
    callback_list.append(callbacks.XPUMetricsCallback())

class XPUMetricsCallback(Callback):
    def on_train_epoch_start(self, trainer: "Trainer",
                             pl_module: "LightningModule") -> None:
        # Reset the memory use counter
        torch.xpu.reset_peak_memory_stats(self.root_gpu(trainer))
        torch.xpu.synchronize(self.root_gpu(trainer))
        self.start_time = time.time()

    def on_train_epoch_end(self, trainer: "Trainer",
                            pl_module: "LightningModule") -> None:
        torch.xpu.synchronize(self.root_gpu(trainer))
        max_memory = torch.xpu.max_memory_allocated(
            self.root_gpu(trainer)) / 2**20
        epoch_time = time.time() - self.start_time

        max_memory = trainer.strategy.reduce(max_memory)
```

System-Tooling Matrix

Tool	JADE2	Baskerville	COSMA8	Azure
PyTorch	✓	✓	✓	
Lightning	✓	✓	✓	
⚡ Fabric				
Deepspeed	✓	✓	✓	
FSDP	✓	✓	✓	
Tensor Parallel				
MPI	✓	✓	✓	
oneCCL				
Accelerate				
AzureML	✗	✗	✗	✓

Periodic Training

1. Annual training from the Baskerville Team
2. Bespoke in-house training
3. Knowledge-sharing tech-talks and reading groups



Developing Benchmarking Results

Benchmarking different HPC systems

1. Tasks relevant for AI workloads
2. Develop walkthroughs in parallel

Help users and projects select appropriate systems

1. Hard to compare based on individual systems' websites
2. Different systems have quite different characteristics for different workloads
3. Will give some results later

Structured Onboarding

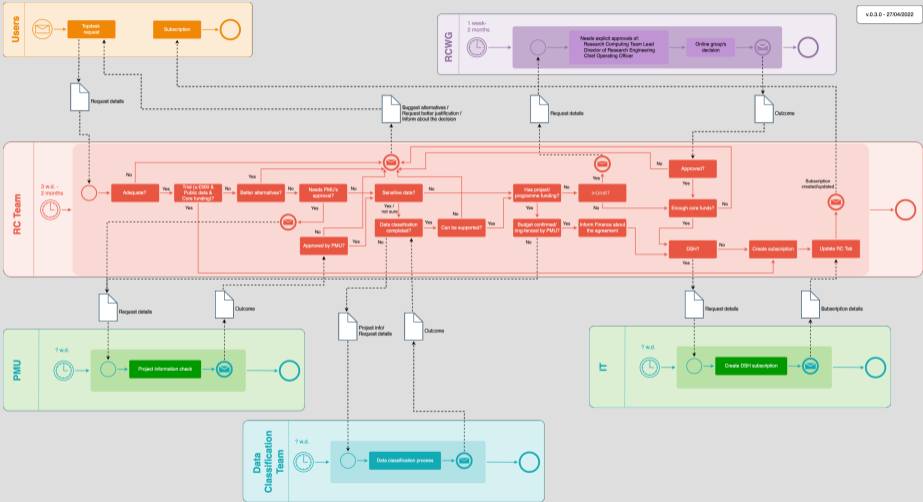
Flowchart processes

1. Modelled on a successful internal allocation flowchart
2. Built around a set of Intranet pages
3. Originally a wall of text

Ticketing system

1. Works okay but introduces project ping-pong
2. Want to avoid by providing more accessible material up-front

Backed up with drop-in sessions



Flowchart Processes

1. Developed separate user-facing flowchart
2. Plan to develop into an interactive flow
3. Plus drop-in sessions

Research Compute Res... x +

https://mathison.turing.ac.uk/page/2430

132%

Mathison@ The Alan T Institute

What's happening v You at work v Science v About us v Finance, Legal & Policies v

Q 🔔 👤

Applying for allocations

Requests should be submitted through [Turing Complete \(https://turingcomplete.topdesk.net/\)](https://turingcomplete.topdesk.net/): Research Services -> Research Computing Platforms -> Request Allocation).

If it is your first time completing the request form, you can refer to this image to get some idea of which service to apply for. These are only recommendations and you may apply for access to any computing service. Each decision corresponds to a subheading further down the page.

```

    graph TD
      Start([Start]) --> Q1{Do I need Azure?}
      Q1 -- Yes --> Q2{Who pays for it?}
      Q1 -- Maybe --> A1[Get a 300GBP Azure Trial]
      Q1 -- No --> Q3{Am I using Notebooks?}
      Q2 -- "The Project/programme" --> A2[Submit a brief Azure request]
      Q2 -- "Core funds" --> A3[Submit a full Azure request]
      Q3 -- Yes --> A4[Request time on Baskerville]
      Q3 -- No --> Q4{Are V100s Sufficient?}
      Q4 -- Yes --> A4
      Q4 -- No --> A5[Request time on JADE2]
  
```

Do I need Azure?

For your computing needs, you have the option of using cloud computing (namely, Microsoft Azure) and/or HPC (namely, Baskerville and JADE2) systems.

Microsoft Azure is a cloud computing platform, which provides services such as virtual machines, virtual networks and databases as well as services targeted at specific fields, such as analytics, machine learning, and internet of things. Microsoft Azure is particularly suitable if you need full control of the resource. For example, to host a publicly available web service, when software installation requires root access and for long-term storage of large data volumes. If you will be working with sensitive data, see [Trusted Research Environments](#).

Trial Access

All Turing users can request trial access

1. Minimal justification
2. Restricted to 400 GPU hours
3. Aimed at helping scope and specify requirements
4. Production systems

Can be converted to full subscriptions

Benchmarking

HPC Benchmarking

1. Explore real-world training performance
2. Use PyTorch Lightning for multi-GPU strategies
<https://github.com/Lightning-Universe/lightning-GPT>
3. Focus on GPT-2 (minGPT)

Model	Hidden layers	Attention heads	Embedding dim	Parameters (M)	16 bit Size (MB)
GPT2	12	12	768	85.21	170.51
GPT2-M	24	16	1024	302.51	605.16
GPT2-L	36	20	1280	708.64	1417.45
GPT2-XL	48	25	1600	1475.87	2951.96
GPT2-XXL	60	30	1920	2656.08	5312.43
GPT2-XXXL	84	40	2560	6609.33	12219.00

HPC Systems

Service	Name	Type	Accelerator	GB	Interface	Launched
JADE 2	J-V100-32	GPU	Nvidia V100	32	SXM2	06-2017
Baskerville	B-A100-40	GPU	Nvidia A100	40	SXM4	06-2020
Baskerville	B-A100-80	GPU	Nvidia A100	80	SXM4	06-2020
Stanage	S-H100-80	GPU	Nvidia H100	80	PCIe 4.0	03-2023
COSMA8	C-MI100-32	GPU	AMD MI100	32	PCIe 4.0	11-2020
COSMA8	C-MI210-64	GPU	AMD MI210	64	PCIe 4.0	03-2022
Graphcore	IPU-POD 16	IPU	IPU-M2000	14.4	RoCEv2	03-2021
Dawn	D-MX1550-128	GPU	Intel Max 1550	128	PCIe 5.0	03-2023

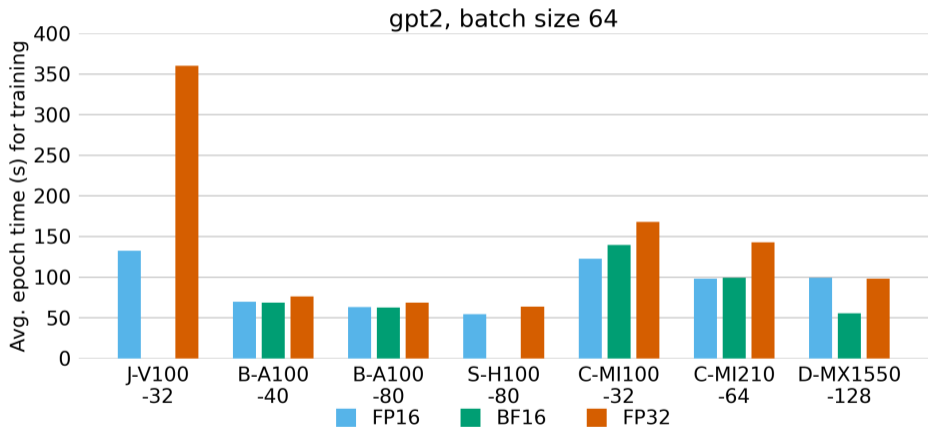
HPC Peak Performance on Paper (TFLOPs)

Service	Name	GB	FP16	BF16	FP32	FP64
JADE 2	J-V100-32	32	31.33	N/A	15.7	7.8
Baskerville	B-A100-40	40	312	312	19.5	9.7
Baskerville	B-A100-80	80	312	312	19.5	9.7
Stanage	S-H100-80	80	1513	1513	51	26
COSMA8	C-MI100-32	32	184.6	92.3	23.1	11.5
COSMA8	C-MI210-64	64	181	181	22.6	22.6
Graphcore	IPU-POD 16	14.4	3994	N/A	998	N/A
Dawn	D-MX1550-128	128	52.43	832	52.43	52.43

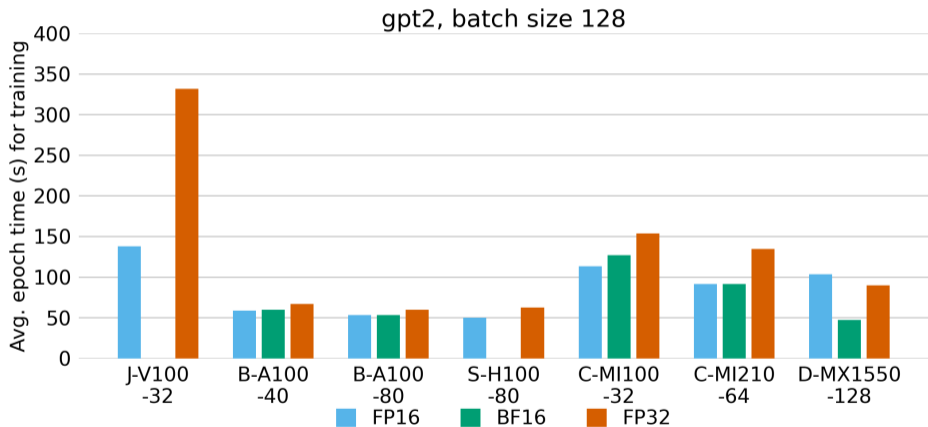
Strategies

1. Distributed Data Parallel
2. DeepSpeed ZeRO
3. Fully Sharded Data Parallel
4. Pipelined Execution
5. Sharded Execution

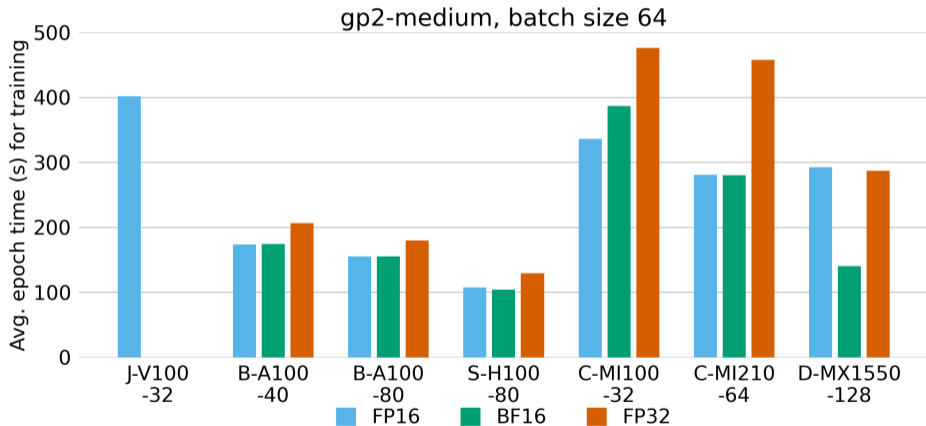
Single Accelerator Comparison



Single Accelerator Comparison



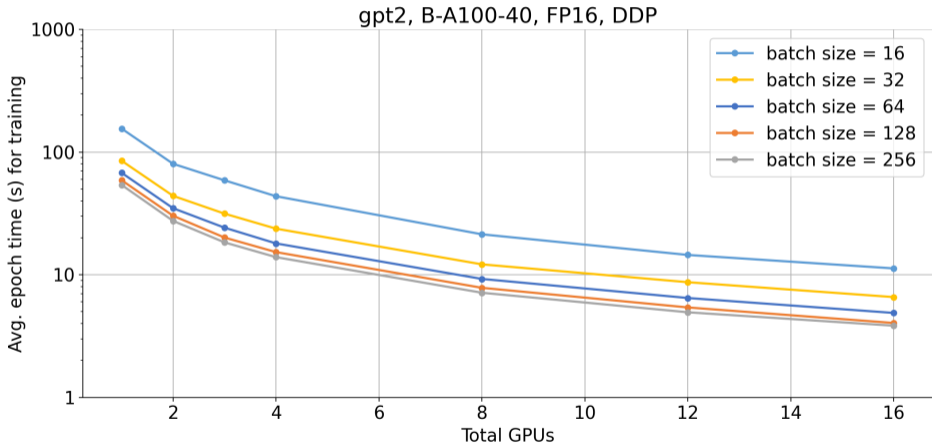
Single Accelerator Comparison



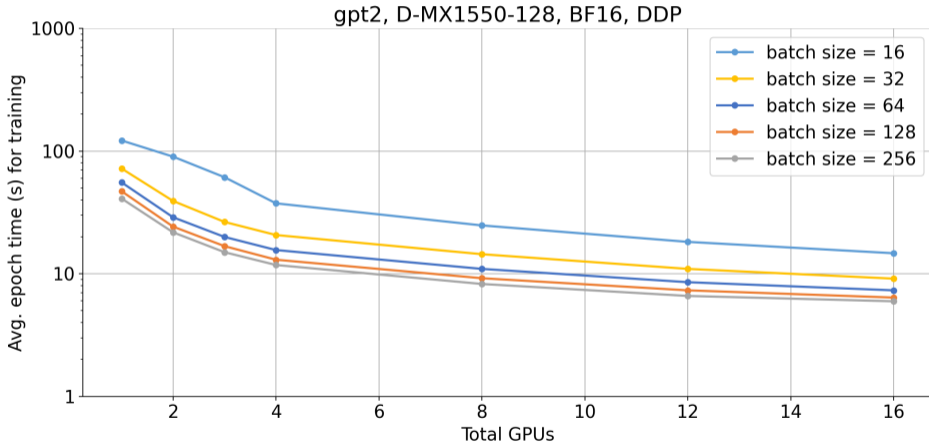
Single Accelerator Comparison - Observations

1. Nvidia H100 80 GB is the fastest GPU, theoretically and actually
2. Performance gap between H100, A100, Max 1550 not as large as expected
3. Depends on peak performance and precision
4. Difference between 16 and 32 bit less significant for smaller models, except for V100
5. Increased model size significantly increases training time
6. Doubling batch size does not significantly improve training time
7. GPT2-M and a batch size of 128 too large for 40 GB

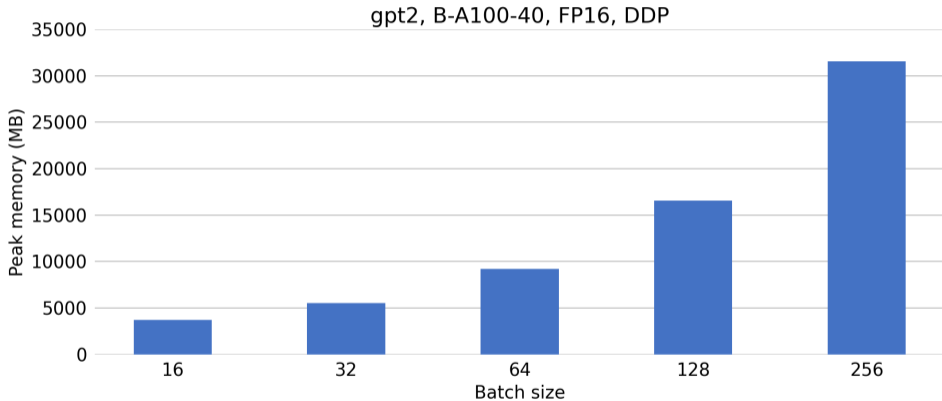
Scaling Up and Out with DDP



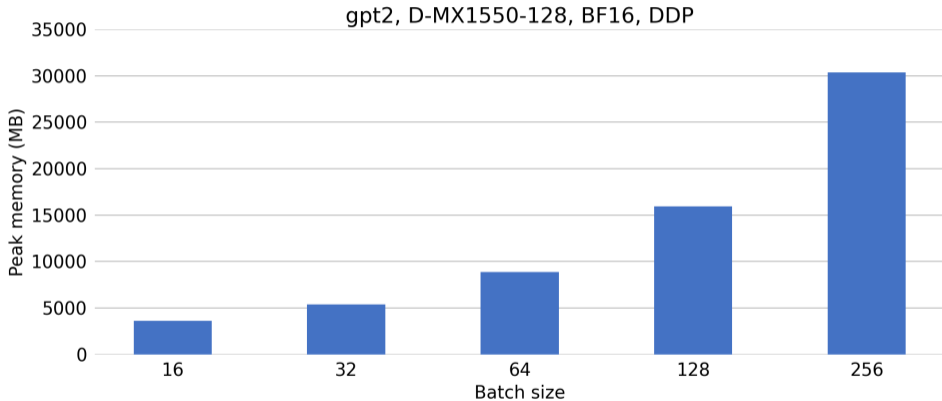
Scaling Up and Out with DDP



Scaling Up and Out with DDP



Scaling Up and Out with DDP



Scaling Up and Out with DDP - Observations

1. Scaling between 1 and 16 GPUs marginally sub-linear
2. Batch size 64 to 128 decreases training time by 15%.
3. Batch size 64 to 256 reduces training time by 22%
4. Batch size 64 to 32 increases training time by 31%
5. Batch size 64 to 16 increases training time by 137%
6. Fixed model size, limiting performance factor is batch size and GPU memory
7. Doubling batch size increases peak memory usage by a factor of 1.5
8. Peak memory usage did not significantly change between 1 to 16 GPUs

Conclusions

Conclusions - Benchmarks

1. BFLOAT16 peak performance better indicator for AI workloads than FP16 or FP32
2. Tensor Core on A100 enables mixed-precision training, better for AI workloads
3. MI100 32 GB and MI210 64 GB potentially more suitable for traditional HPC tasks
4. FSDP faster than DeepSpeed, but DeepSpeed Stage 3 more memory-efficient for largest models
5. Largest trainable model using DeepSpeed and FSDP is GPT2-XXXL
6. Balanced consideration of memory and time is needed especially for larger models

Conclusions - Process

1. Understanding HPC trade-offs is difficult for researchers
2. We use four approaches to try to help
 - 2.1 Knowledge base and training
 - 2.2 Structured onboarding
 - 2.3 Trial access
 - 2.4 Embedding in projects
3. These are all still work-in-progress

Acknowledgements

1. With thanks to Edwin Brown, Sheffield and Turing
2. Funded by The Alan Turing Institute under the EPSRC grant EP/N510129/1
3. Partially supported by Baskerville, a national accelerated compute resource under the EPSRC Grant EP/T022221/1
4. Partially supported by JADE: Joint Academic Data Science Endeavour - 2 under the EPSRC Grant EP/T022205/1, and The Exascale Computing: Algorithms and Infrastructures Benefiting UK Research (ExCALIBUR) program, which is funded under Wave 2 of the Strategic Priorities Fund (SPF)
5. Cambridge Service for Data Driven Discovery (CSD3) operated by the University of Cambridge Research Computing Service (<https://www.csd3.cam.ac.uk>), provided by Dell EMC and Intel using Tier-2 funding from the Engineering and Physical Sciences Research Council (capital grant EP/T022159/1)
6. The University of Sheffield for the provision of services for High Performance Computing
7. The Mandelbrot system at the UCL Centre for Advanced Research Computing and associated support services (<https://www.ucl.ac.uk/advanced-research-computing/advanced-research-computing-centre>)
8. DiRAC@Durham facility managed by the Institute for Computational Cosmology on behalf of the STFC DiRAC HPC Facility (<https://www.dirac.ac.uk>). The equipment was funded by BEIS capital funding via STFC capital grants ST/P002293/1, ST/R002371/1 and ST/S002502/1, Durham University and STFC operations grant ST/R000832/1. DiRAC is part of the National e-Infrastructure