In This Edition

"Welcome to the first edition of our Research Computing newsletter, each month we will explore how HPC and machine learning are being used by researchers at Bath. In this issue, we spotlight the ACORN research team's work on sustainable construction materials and how HPC shortened their time to results. We also have a piece around the recently added support for containerization on Nimbus with Singularity, enhancing code portability and reproducibility across complex computational environments. It would be great to hear your feedback on this issue and what you'd like to see in future issues."

Roger Jardine, Research Computing Director

Crushing Numbers, Breaking Concrete: Using Nimbus for the Finite Element Analysis of Concrete Thin-Shells

Maxime Pollet (PhD Candidate, Department of Architecture and Civil Engineering)
Email: *mp2333@bath.ac.uk*

Maxime is a third year PhD student within the Department of Architecture and Civil Engineering. His research is aligned with the Automating COncrete conStRuctioN (ACORN) research team, which investigated the use of concrete thin-shells as a flooring system.
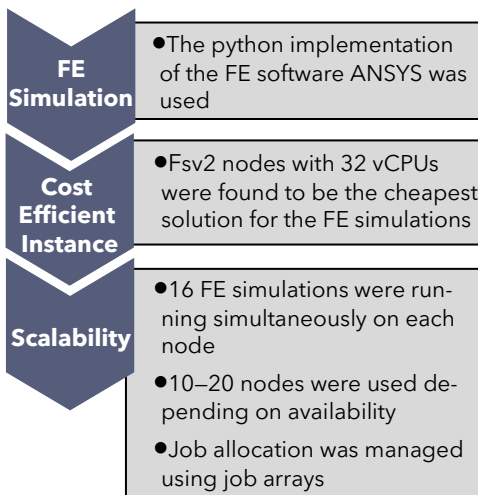
Concrete thin-shells (as shown in Figure 1) have an optimised shape, which allows them to be thinner than traditional flat concrete slabs. As a result, the quantity of concrete used in the production of materials can be reduced, which is of particular importance as Portland cement (one of the basic ingredients of concrete) is responsible for approximately *8% of global CO_2 emissions every year* [1]. In the case of the ACORN prototype, this results in a *reduction of about 50%* of cradle-to-gate embodied carbon compared to regular flat slabs [2]. However, the structural behaviour of concrete thin-shells is complex. It is typically assessed using Finite Element (FE) analysis, which can be computationally expensive and time-consuming.

Maxime's research investigates the use of Machine Learning (ML) to enable rapid estimation of the structural behaviour of concrete thin-shells. The concept consists of pre-running a large number of FE analyses for thin-shells with various shapes and material properties, and then training ML models to predict the outputs of the FE analyses as shown in figure 2. Once trained, the models can be used to estimate the structural behaviour of new samples. While ML models are not as accurate as FE analysis, they can make predictions in less than a second. Such a trade-off between speed and accuracy would enable several new applications in the design and production

"Nimbus is assisting Maxime in his research on concrete thin-shells, delivering a 50-fold performance boost"

Figure 1: ACORN's demonstrator of a concrete thin-shell ©John Orr.

Figure 2: Output of the FE analysis (z-component displacement) for one of the 40,000 samples

phases. The overview of Maxime's workflow that he has been using to effectively run hundreds of simulations on Nimbus is explained here:

FE Simulation	• The python implementation of the FE software ANSYS was used
Cost Efficient Instance	• Fsv2 nodes with 32 vCPUs were found to be the cheapest solution for the FE simulations
Scalability	• 16 FE simulations were running simultaneously on each node • 10–20 nodes were used depending on availability • Job allocation was managed using job arrays

One of the challenges encountered by Maxime involved effectively managing the allocated storage space on Nimbus. The sheer size of the input and output files of the 40,000 samples exceeded the storage capacity on Nimbus. Consequently, Maxime adopted a strategy of dividing the workload into batches of 480 samples, necessitating frequent movement of files between Nimbus and local storage. To streamline this
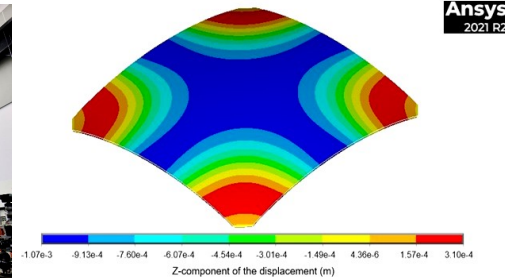
process, Maxime developed a scripted solution. This simple script routinely checked the status of the batch analyses running on Nimbus at 10-minute intervals. Upon completion of a batch, the script initiated the transfer of files back to the local storage and then dispatched a new batch to Nimbus. This automation significantly enhanced the efficiency of the workflow.
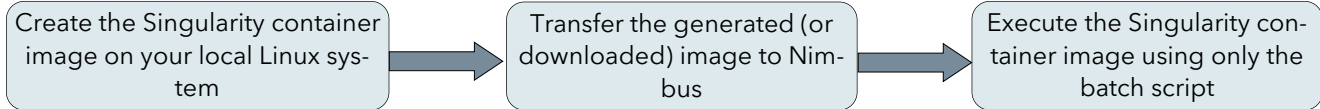
Maxime provided the following feedback of the experience of using Nimbus: *"The implementation of the entire workflow was easy thanks the Research Computing Team. They provided help and advice on several occasions, providing insights into the most efficient ways to use Nimbus. Using Nimbus made the constitution of my training set a lot quicker. On my university computer, I was able to complete on average 4.5 FE analyses per hour. On Nimbus, the average speed was of 240 analyses per hour. In other words, Nimbus allowed me to complete my task more than 50 times quicker than I would have on my university computer."*

References
1. Lehne, J. and Preston, F. (2018). *Making Concrete Change Innovation in Low-carbon Cement and Concrete #ConcreteChange. CHATHAM HOUSE.*
2. Oval, R., Nuh, M., Costa, E., Madyan, O.A., Orr, J. and Shepherd, P. (2023). *A prototype low-carbon segmented concrete shell building floor system. Structures, 49, pp.124–138.*

Scientific Applications Containerization with Singularity on Nimbus

Executing user codes in an HPC environment often demands extensive input due to compiler and library dependencies. Enhancing code portability and reproducibility is achievable through containers. Singularity, an open-source tool, enables users to encapsulate applications and dependencies into a single image file. Notably, Singularity is designed for HPC systems where users lack root privileges. To learn more about Singularity, view the documentation: *https://docs.sylabs.io/guides/3.5/user-guide/introduction.html.*

Create the Singularity container image on your local Linux system	→	Transfer the generated (or downloaded) image to Nimbus	→	Execute the Singularity container image using only the batch script

What is Containerization?

The process of creating containers (image files) that store all code and its dependencies for running applications on HPC.

When to use Containers?

There are cases mostly in AI workloads where the dependencies of an application are either too complex to install or not compatible with the host operating system (OS). In such cases, creating singularity containers can help by building a container with a compatible OS and then running the container image file on HPC.

Running applications binaries with OS requirements that are incompatible with the host system	Containers support both CPU and GPU applications, facilitating MPI runs across compute nodes as well.

How to implement on Nimbus?

Nimbus has singularity module (Singularity/4.0.0) available for HPC Users to run the container image files built with singularity software only. The container images cannot be built on NIMBUS by the users. The user can get the container images of the required applications using any of the following methods:

```
# Singularity Module available on Nimbus
[user@nimbus-1-login-1~]$ module avail Singularity
Singularity/4.0.0
```

The container images built by RC Team are available at the shared directory: /home/application/singularity_images/….	For many applications, the container images can be directly downloaded from: Singularity hub, NVIDIA GPU, etc.
For creating open source application containers, the users may request the RC Team or create one themselves.	For inhouse developed applications, the RC Team may help the users create one to improve portability across HPC clusters.

HPC News and Updates

Apply for **University Accelerator Awards** to secure funding for your research project utilizing Nimbus. Submit your application through this link: *https://forms.office.com/e/Sci494LC31*

HPC training sessions are re-starting soon and we plan to conduct the first in mid-March 2024.

Tip of the Month

The Linux command *"ldd"* can help in identifying shared library dependencies and users can handle library paths for executables before running.

```
[user@login1 ~]$ ldd a.out

# This will list out all the required shared
libraries of a.out.
```

Acknowledgements

The Research Computing team would like to thank all contributors for the current issue of *HPCBytes*.

- If you would like to contribute a case study or article to be featured in *HPCBytes*, please get in touch with the Research Computing team.
- If you would like to hear more, please subscribe to the Research Computing mailing list here: *https://forms.office.com/e/rF8rLWbakA*

Contact us

Research Computing Team
Digital, Data and Technology
Email: *it-hpc@bath.ac.uk*