# The Red Oak Microsoft Azure HPC Collaboration Centre in partnership with AMD

## Deploying, Benchmarking and Visualising workflows using an AMD HPC Cluster in Microsoft Azure

CA005PR002X003 – 2.0

Nick Skingle
Manveer Munde
James Page

Red Oak Consulting
Expert advice, exceptional delivery

# Release status

| | |
|---|---|
| This document is version: | 2.0 |
| Reason for release of new version: | Final draft for internal review |
| Ready for review: | Nick Skingle, 16-06-2022 |
| Technical review sign-off: | Dairsie Latimer, 16-06-2022 |
| Commercial review sign-off: | n/a |
| Copy edit review: | Vicki Lockhart, 16-06-2022 |
| Approved for release: | Dee Chadwick, 16-06-2022 |

# Contents

# Introduction

This white paper forms part of a series of offerings from Red Oak Consulting sponsored by Microsoft Azure and AMD. The aim is to demonstrate the performance and scalability of AMD EPYC™[1] based HPC virtual machines (VMs) in the Azure cloud using representative, readily available finite element analysis (FEA) and computational fluid dynamics (CFD) models.

An Azure DevOps (ADO) project[2] has been created and is available to clone from a public repository, which contains all the code and documentation needed to deploy an HPC cluster in the Azure cloud. An introductory video[3] is available that provides an overview of the ADO project and demonstrates the environment that will be deployed to the Azure subscription once the repository pipelines have been successfully run.

A second video[4] demonstrates and discusses benchmarking the representative CFD and FEA codes using a Python framework called ReFrame.[5] Two software packages have been selected for the simulations. The commercial software package Ansys LS-DYNA[6] for the simulation of both the car2car and Honda Accord crash models, and the open-source software package OpenFOAM[7] for the simulation of the motorbike model. The ReFrame framework is used to run the workflows on a SLURM[8] scheduler, with Microsoft CycleCloud[9] managing the orchestration of the cluster. Further details on the benchmarking results are discussed in greater detail in this white paper.

Using the benchmark results, we will discuss how scaling the number of cores to run a simulation can improve time to solution and increase productivity. We will also look at a comparison of the cost for time to solution, to demonstrate where on the curve a sweet spot can be found for optimising workflows. This can be reproduced using the provided information in the ADO repository.

The outputs of the workflows are visualised by deploying an Azure NVv4 VM that is configured with an AMD EPYC™ processor and AMD Instinct™ MI25 GPU.[10] The visualisation and analysis tools LS-PrePost[11] and ParaView[12] are used to demonstrate the capabilities of using a virtual desktop in the Azure cloud. The third and final video[13] provides a demonstration of these tools in action on the cloud VDI (Virtual Desktop Infrastructure) workstation. Later in this white paper, we discuss the benefits of leaving data in the cloud.

## Video series contents

- Video one: Deploying an AMD HPC Cluster in Microsoft Azure utilising the Red Oak Azure DevOps (ADO) Repository.
- Video two: Demonstration of using ReFrame to run and benchmark applications on the HPC cluster (Ansys LS-DYNA and OpenFOAM.com).
- Video three: Demonstration of visualisation of models on an AMD based remote GPU Workstation in Azure.

[1] https://www.amd.com/en/processors/epyc-server-cpu-family
[2] https://dev.azure.com/RedOakConsultingPublic/Microsoft%20Centre%20of%20Excellence/_git/CentreOfExcellence2022
[3] https://www.youtube.com/watch?v=qUQFWbVnbsU
[4] https://www.youtube.com/watch?v=w5UpOUwmRbI
[5] https://reframe-hpc.readthedocs.io/en/stable/index.html#
[6] https://www.ansys.com/en-gb/products/structures/ansys-ls-dyna
[7] https://www.openfoam.com/
[8] https://slurm.schedmd.com/documentation.html
[9] https://azure.microsoft.com/en-gb/features/azure-cyclecloud/#overview
[10] https://www.amd.com/en/products/professional-graphics/instinct-mi25
[11] https://www.lstc.com/products/ls-prepost
[12] https://www.paraview.org/
[13] https://www.youtube.com/watch?v=POSZxBrzGvs

# Deploying an HPC Cluster in Azure

As mentioned, an ADO repository containing all the resources required to deploy an HPC cluster in Azure has been made publicly available. The repository utilises readily available tools to build the operating system images, deploy the necessary Azure resources and configure a Slurm HPC cluster with CycleCloud.

We have discussed in previous white papers how to manually deploy CycleCloud and deploy an HPC cluster with a Slurm scheduler. This white paper, the associated ADO repository and the video demonstrations show how to automate the deployment of a similar HPC environment in Azure utilising AMD EPYC based HBv2 and HBv3 HPC instances. The HB-series VMs along with a Mellanox Infiniband network, have been optimised in Azure for HPC processing applications, such as CFD and FEA.

Figure 1 provides an example architecture design diagram of the environment being deployed in Azure.

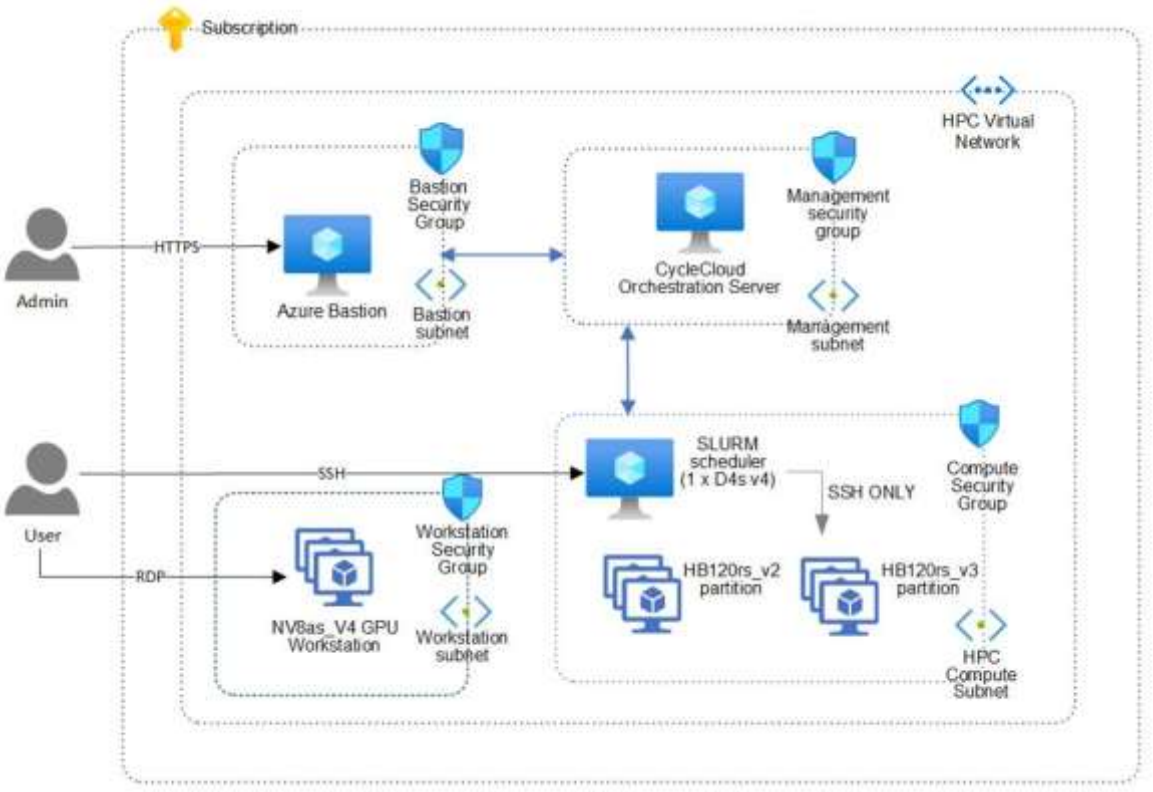


**Figure 1: an example design schematic of an Azure HPC environment**

The ADO repository contains the YAML files required to create the pipelines for deploying the infrastructure. The pipelines create the HPC compute node and CycleCloud orchestration server images, deploy the infrastructure to Azure, and configure the CycleCloud cluster.

The ReFrame framework software is installed on the HPC image as part of the image creation. ReFrame is a Python framework that is designed for running regression testing in an HPC environment. For the white paper, it is used to run and collect the performance metrics for each of the benchmarked workflows.

The README file in the repository contains a full explanation of how the code has been set up to deploy the infrastructure using the Azure native Bicep language.

Table 1 provides details of the specification of the HB-series of VMs optimised to run HPC workloads being used for benchmarking.

| Instance Size | vCPU | Processor | Memory | GHz Peak | RDMA (Gb/s) |
|---|---|---|---|---|---|
| Standard_HB120rs_v3 | 120 | AMD EPYC 7V73X | 448 | 3.5 | 200 |
| Standard_HB120rs_v2 | 120 | AMD EPYC 7V12 | 456 | 3.3 | 200 |

**Table 1: AMD HB-series instance specifications**

The GPU VDI instance is deployed using the Azure portal, with LS-PrePost and Paraview being installed manually. The specification of the AMD based VDI is shown in Table 2.

| Instance Size | vCPU | Processor | Memory | GPU | GPU Memory (GiB) | GPU Partition |
|---|---|---|---|---|---|---|
| Standard_NV8as_v4 | 8 | AMD EPYC 7V12 | 28 | AMD Instinct MI25 | 4 | 1/4 |

**Table 2: AMD NVv4-series instance specification**

The ADO repository contains three yaml files that build the pipelines for deploying the HPC cluster, which should be run in the following order:

- images.yml – builds and creates the HPC and the CycleCloud orchestrator OS images.
- infrastructure_bicep.yml – deploys the Azure resources for the cluster.
- cycle.yml – initialises and configures CycleCloud, Slurm and ReFrame.

The second video in the series demonstrates how to use ReFrame to schedule and run the workflows.

## Customising the HPC image

In most production circumstances we would recommend the installation of the modelling software on a shared filesystem such as Azure NetApp Files (ANF), mounted across all the nodes in the HPC cluster. For this white paper, it was agreed to install the requisite software on the HPC image. The software processing packages compiled and installed manually on the HPC image are:

- Ansys LS-DYNA (version 2022R1);
- OpenFOAM.com (version 2012).

All the instructions for customising the image are supplied in the ADO repository README file, including using Spack[14] recipes for compiling the OpenFOAM binaries using the AMD AOCC 3.2 compiler.

[14] https://spack.io/

# Benchmarking the AMD based HPC clusters

The performance ratio used for comparing the performance gain of the HBv3 instances is:

$$performance\ ratio = (('HBv2\ cpu\ time' - 'HBv3\ cpu\ time')/'HBv3\ cpu\ time') + 1$$

## Ansys LS-DYNA benchmarking

Ansys LS-DYNA was selected as it is a commonly used commercial software. It should be noted that a license is required to run the LS-DYNA application. Licenses can be obtained from Ansys. For these benchmarking tests, 720 core licences were available. It was also necessary to configure an Ansys license manager node that serves licenses to the HPC LS-DYNA client nodes. In this case, the CycleCloud orchestrator node was used as the license manager node.

For benchmarking with LS-DYNA two FEA representative models were selected, these being the Honda Accord[15] and car2car[16] models. The Honda Accord is a more complicated model than car2car with higher geometry counts in the mesh and will take almost eight hours to run on a single HBv2 instance.

The performance metrics collected for both models were the elapsed time and the element processing time. The elapsed time is the total time it takes for the benchmark to run both the parallel and serial sections of the workflow. The element processing time is a parallel processing part of the simulation which provides information to analyse the parallel scaling for the model.

### Honda Accord results

For the Honda Accord model, Figure 2 shows a comparison of the two instance types for the elapsed time and Figure 3 shows the comparison of the parallel element processing for each instance type.

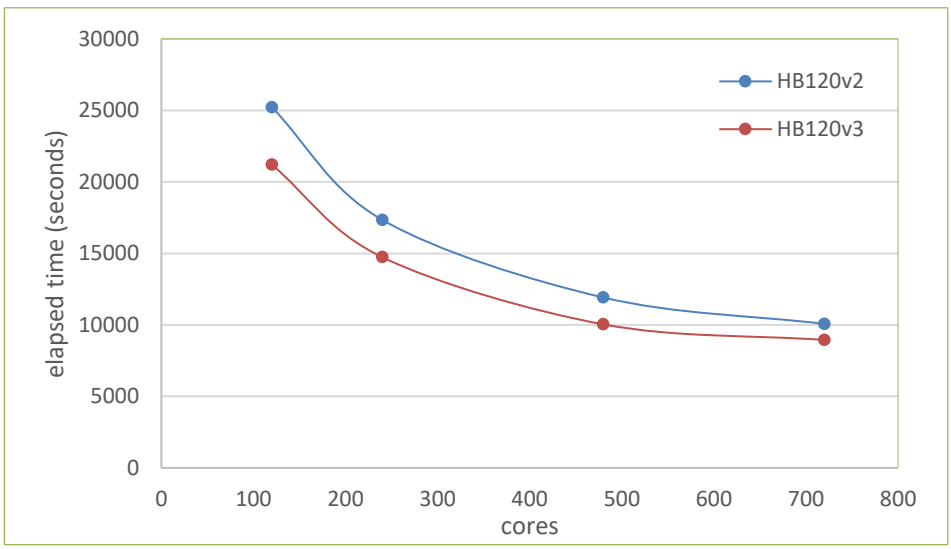


**Figure 2: Elapsed time for LS-DYNA Honda Accord model**

[15] https://www.nhtsa.gov/crash-simulation-vehicle-models
[16] https://ftp.lstc.com/anonymous/outgoing/topcrunch/car2car/

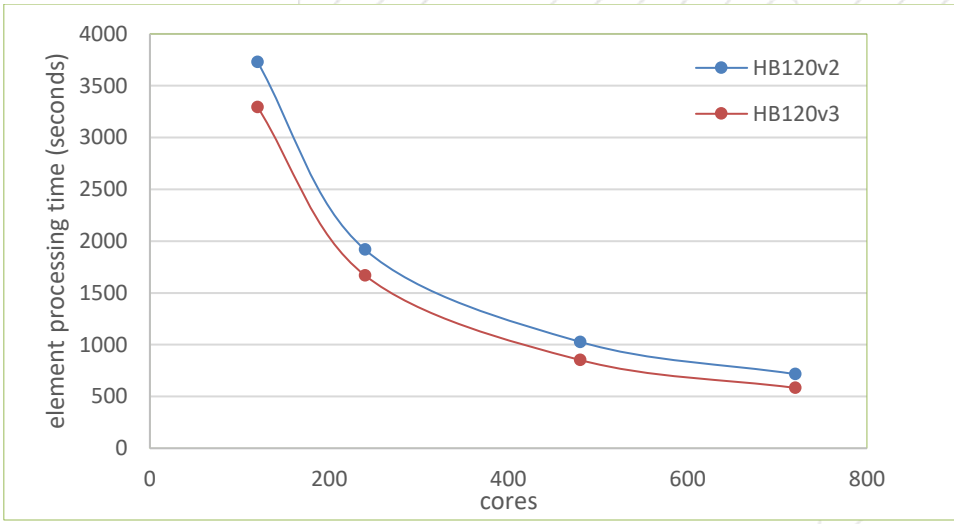


**Figure 3: Element time for LS-DYNA Honda Accord model**

The element processing results of the LS-DYNA Honda Accord model show that the performance gains for the HBv3 instances range between a ratio of 1.13 to 1.23, with the greatest performance gains utilising the full 720 cores. The performance improvement for the elapsed time is reasonably even between all the runs, ranging from a 1.16 to 1.19 gain for the HBv3 instances and the greatest gain seen with 120 cores. It is worth noting that the time reduction from 120 cores to 240 cores can be utilised to allow an engineer to submit multiple design iterations per day and thereby potentially increasing their productivity.

The results show that the Honda Accord model scales well for the parallel element processing stage, though the scaling is not linear after 240 cores. The HBv3 instances do continue to show performance benefits over the HBv2 instances up to and including the maximum licensed core count.

**Car2car results**

The car2car model benchmark results for the elapsed time are shown in Figure 4 and the element processing time is shown in Figure 5.

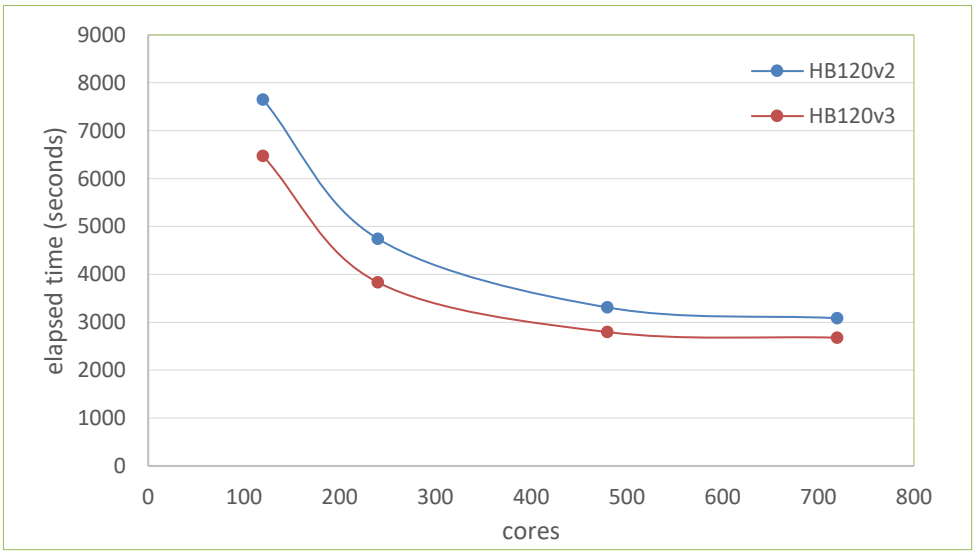


**Figure 4: Elapsed time for LS-DYNA car2car model**

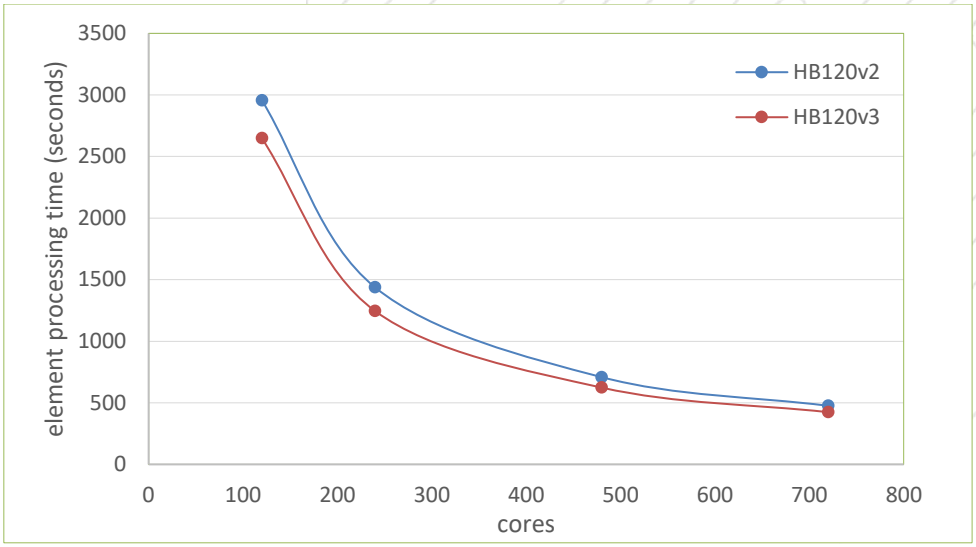


**Figure 5: Element time for LS-DYNA car2car model**

The car2car model results are showing a smaller, though still significant, improvement in performance with the HBv3 instances for both the element processing time and the elapsed time. The element processing performance gains start at a ratio of 1.12 using 120 cores, peaking at a 1.15 performance gain using 240 cores. Whilst the elapsed time ranges from gains of 1.15 to 1.24 performance gains, the largest gains are seen when utilising 240 cores.

It can be seen from the results that the element processing stage of the workflow continues to scale for both instance types, though the scaling does fall off as the core count increases. The reduced amount of scaling shown in the elapsed time graph can be accounted for by the serial part of the process getting slower as the domain decomposition, being a serial process, is taking longer.

Using the sweet spot found by running benchmarks, productivity gains can be achieved for running workloads with the right CPU type and core count combination. The workflows demonstrated here, show the job runtime can be significantly reduced from 50 minutes down to 21 minutes by scaling to 240 cores. Comparing the cost of an Engineer per hour and the cost of two VMs per hour, could provide opportunities for Engineers to be more productive by utilising readily available cloud resources which in turn provides a real return on investment.

## OpenFOAM benchmarking

OpenFOAM.com was selected as it is a free open-source CFD software with a large user base across multiple disciplines of science and engineering. The popular motorbike model has been selected, from the OpenFOAM HPC benchmark suite.[17] The suite provides small, medium and large models. The large model, used for our benchmarking, contains four motorbikes that have been mirrored from the single motorbike model. Using an increased domain size and a finer background mesh, the large motorbike model has 34 million cells in the mesh.

[17] https://develop.openfoam.com/committees/hpc/-/tree/develop

Block meshing is a serial process which proved to have a flat scaling response across multiple cores, levelling off at around 100 cores. The snappyHexMesh, parallel portion, showed better scaling which began to level off around the 40-core mark. When using OpenFOAM.com it is currently recommended to use the hierarchical method of decomposition for the mesh phase, as it performs better for the meshing process than the scotch method. It can be seen in Figure 6 and Figure 7 that there are performance benefits using the HBv3 VM, though there are currently no benefits to scaling the meshing process to more than a single node.

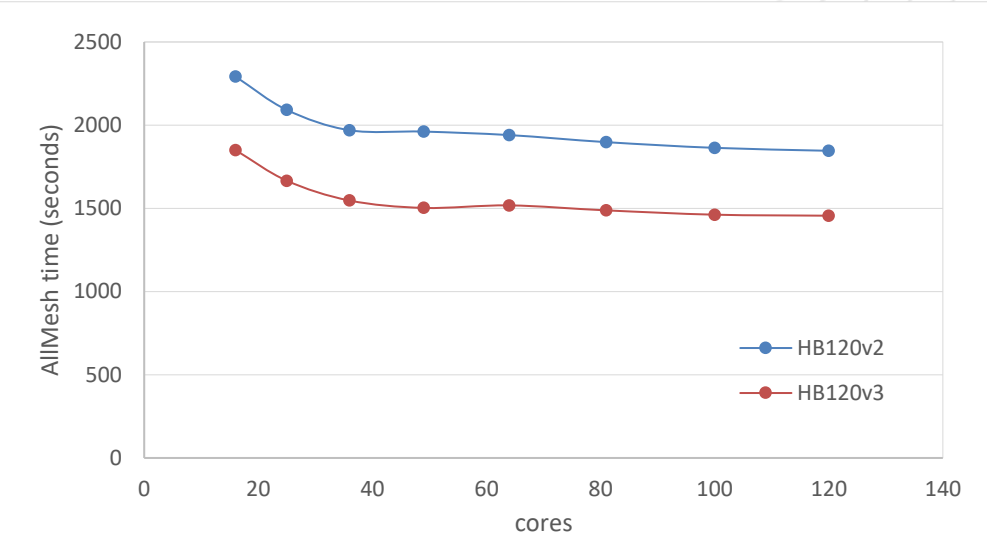


**Figure 6: The AllMesh performance HBv2 vs HBv3**

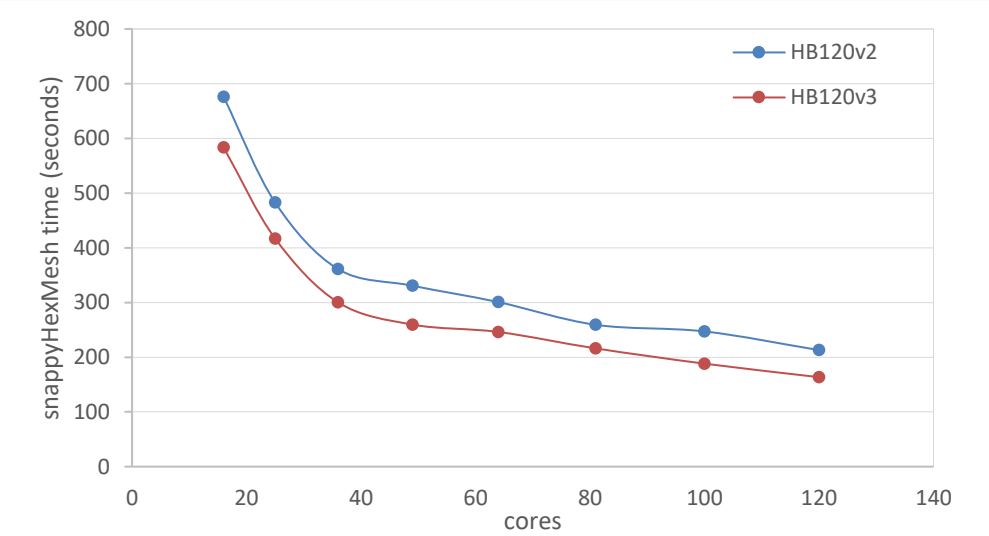


**Figure 7: The snappyHexMesh performance HBv2 vs HBv3**

The OpenFOAM SimpleFoam solver process is the parallel part of the motorbike workflow. Analysing the metrics of the SimpleFoam process will allow us to compare the performance of the different nodes, review the scaling characteristics and calculate the best cross over for core count price/performance. It has been found that the best decomposition method for the solver portion of the process is the scotch method.

Figure 8 shows the runtime per core for running the parallel solver stage of the large HPC motorbike model.

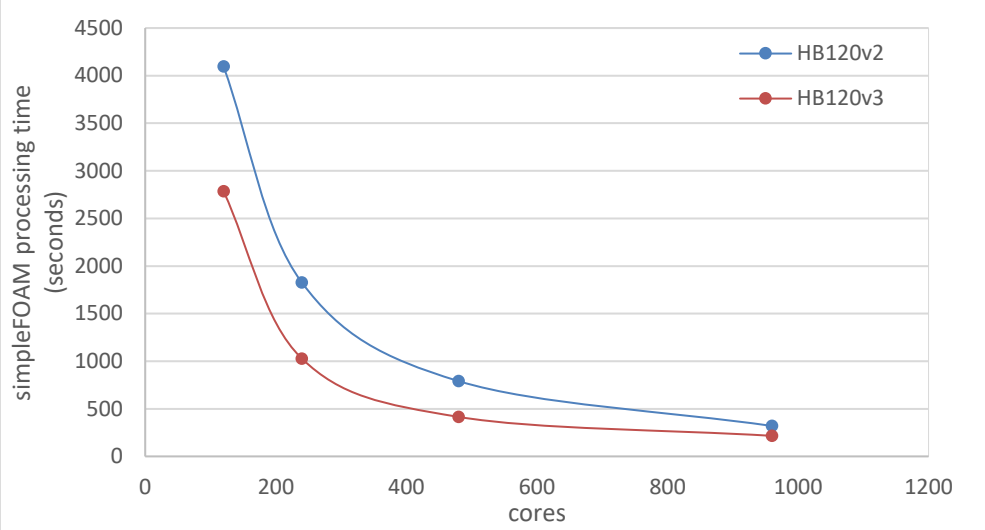


**Figure 8: SimpleFoam performance HBv2 vs HBv3**

The scaling for both the HBv2 and HBv3 instances starts to level out at just over the 240 cores. The results show the performance ratio for the solver stage ranges from 1.45 to 1.91 for the HBv3 instances. The highest performance gain for the HBv3 instances is at 480 cores and the lowest at 120 cores. Though the performance gain reduces as the scaling increases beyond 480 cores, there is still a significant 1.48 performance increase using 960 cores for the HBv3 instances.

# Price and performance comparisons

It is important to appreciate price versus performance and the impact this can have on productivity when deploying HPC clusters, especially when considering the flexibility and options available in the cloud. We have so far compared the performance of the AMD based HBv2 and HBv3 instances in Azure. We should equally review the costs for running those workflows on the different instances. The cost analysis is based on a minimal set of metrics and is focused on compute costs only.

At the time of writing the price for each VM per hour in the South Central US region is shown in Table 3. Spot prices can vary however they average at 90% lower than PAYG pricing.

| SKU | Price in USD for PAYG per hour | Price in USD for Spot per hour |
|---|---|---|
| HB120v2 | 3.96 | 0.396 |
| HB120v3 | 3.96 | 1.584 |

**Table 3: PAYG instance price per core hour**

## Pay as you go (PAYG) instances

Comparing the PAYG prices for the LS-DYNA Honda Accord benchmark shows the performance gains with the HBv3 make these instances more cost-effective to run these workflows. Figure 9 shows the cost comparison of the HBv2 and HBv3 instances running the Honda Accord Model.

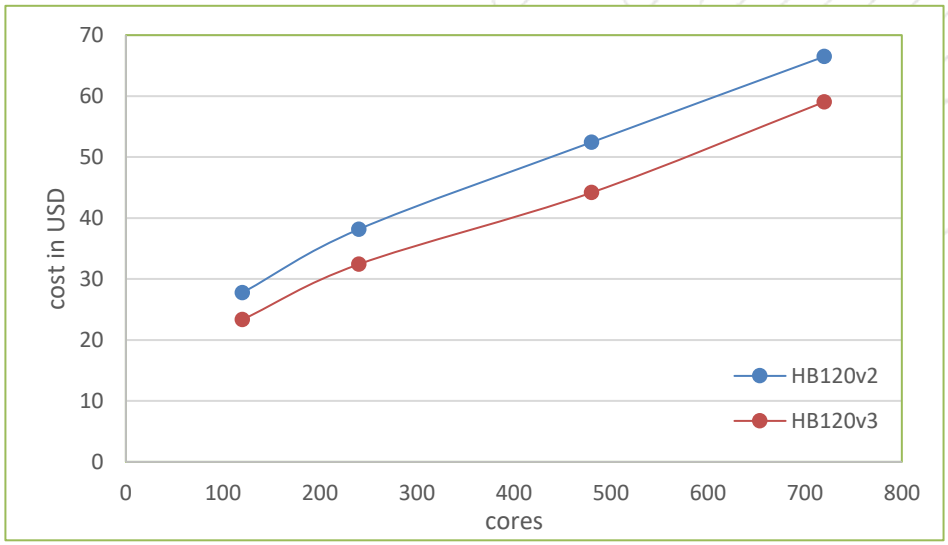


**Figure 9: Honda Accord PAYG cost in USD HBv2 vs HBv3**

The same cost comparison can be made with the car2car workflow, which shows a very similar trend to the Honda Accord costs. Figure 10 shows the cost benefits of using the HBv3 instances when running the car2car model.

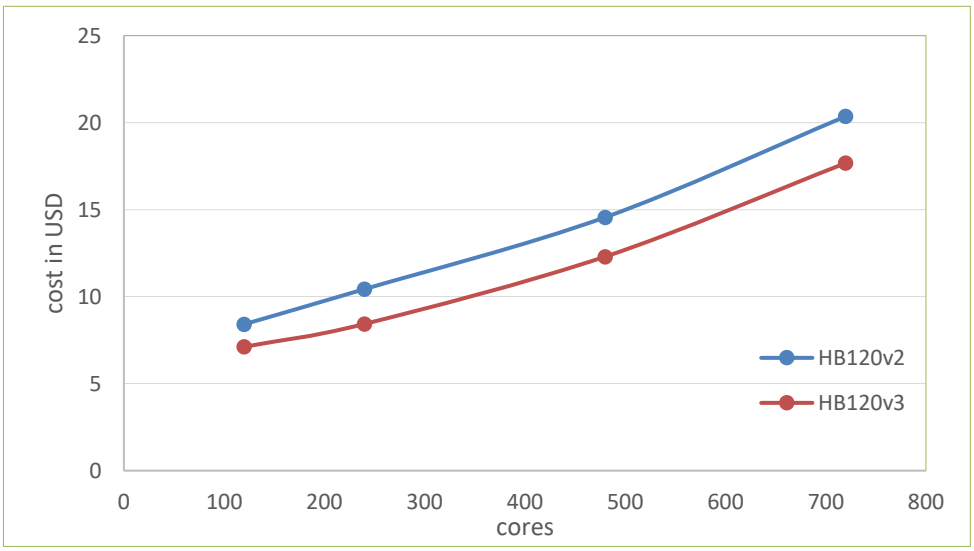


**Figure 10: car2car PAYG cost in USD HBv2 vs HBv3**

A similar trend was seen with the simpleFoam and snappyHexMesh stages of the OpenFOAM motorbike model. However, it should be noted that as the cluster was scaled up, the costs increased significantly for the meshing stage. This was due to there being no performance benefits to scaling beyond one node due to the serial nature of the process. It would therefore be favourable to run the two stages as separate jobs (ie meshing on one node and then solving on several).

### Spot instances

The Azure spot costs should not be ignored, as it is preferable to run workloads on spot instances whenever the situation allows. Table 2 included the prices for spot instances in the South Central US region for the HBv2 and HBv3 instance types, which can be significantly cheaper than PAYG. Figure 11 compares the spot costs for the Honda Accord benchmarking tests as core count increases.

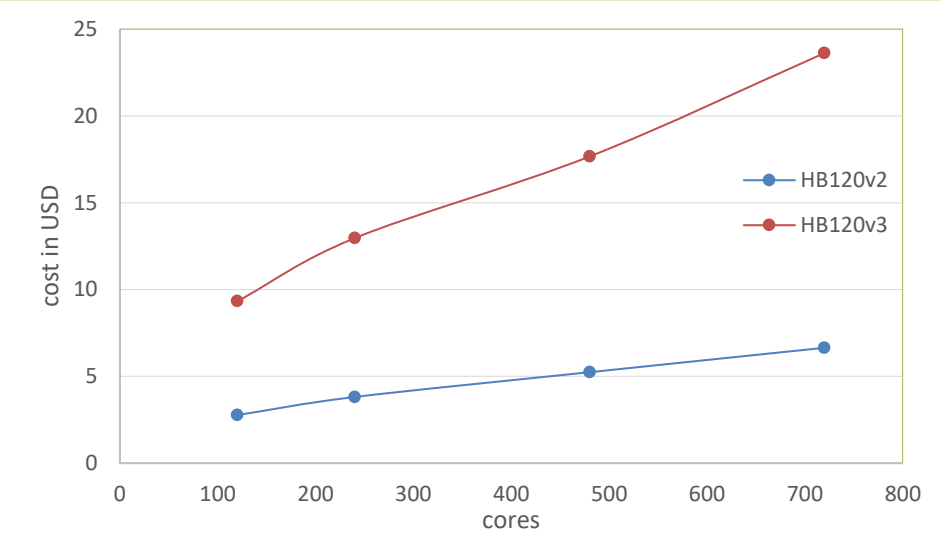


**Figure 11: Honda Accord spot cost in USD HBv2 vs HBv3**

It is clear from the graph in Figure 11 that the spot costs of the HBv3 instance types are currently more expensive than the HBv2. The spot pricing also reflects the current popularity of the newer HBv3 instances for running HPC workloads. As the number and availability of these instances increases in the Azure cloud, it is also likely that the spot price for the HBv3 instances will converge with the spot price for the HBv2 instances.

## Remote visualisation in Azure

The focus so far throughout this white paper has been the performance, cost, and productivity of an HPC cluster in the Azure cloud. One of the considerations when running HPC workloads in the cloud is the necessity for visualisation of the data. Now the question is where best should that visualisation take place? The downloading of data from Azure cloud storage to an on-premises storage location for visualisation introduces egress costs and possible delays due to the data transfer time.

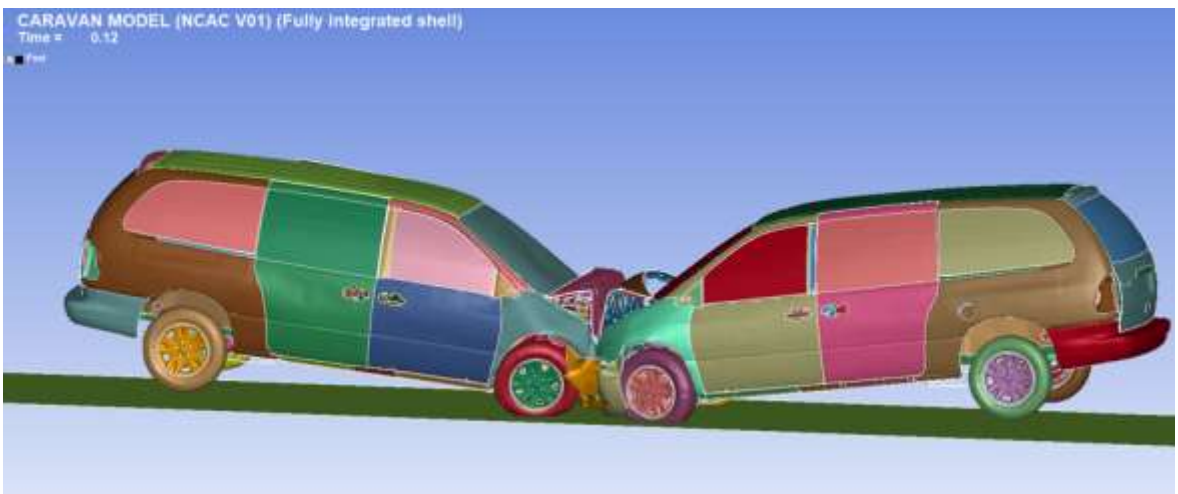


**Figure 12: LS-PrePost car2car on a remote VDI**

Azure now provides VDI workstations such as the NVv4 series VM based on the AMD Instinct™ MI25 graphics adapter. We have demonstrated in the final video that complements this white paper, that visualising the output from the workflows using Paraview and LS-Prepost is responsive and works well. The improvement in the capabilities of remote visualisation means costs of egress and lengthy download times can be avoided. Having the data readily available in the same location as the remote VDI provides an opportunity for improved productivity.

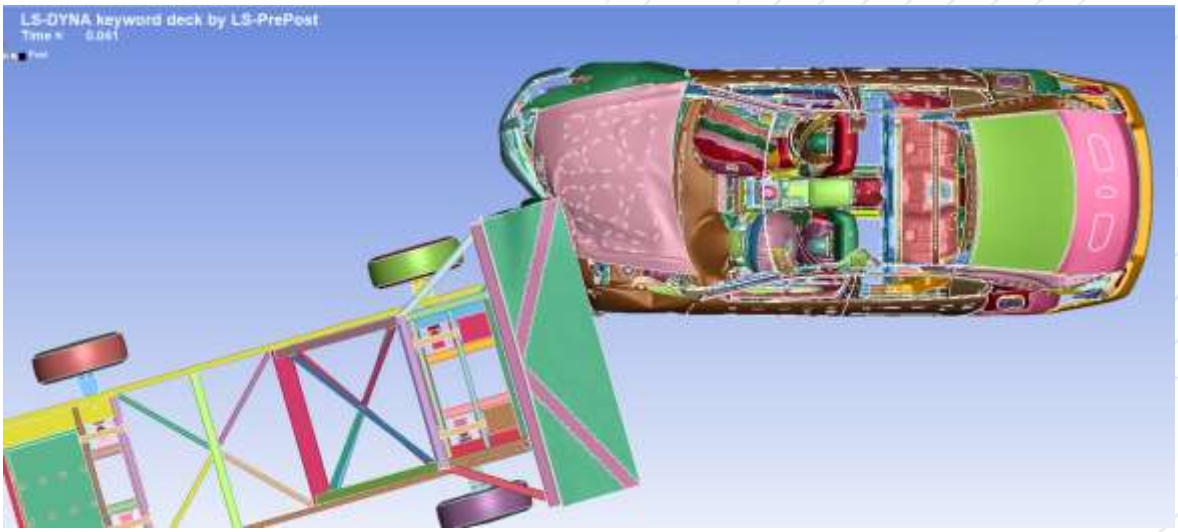


**Figure 13: LS-PrePost Honda Accord on remote VDI**

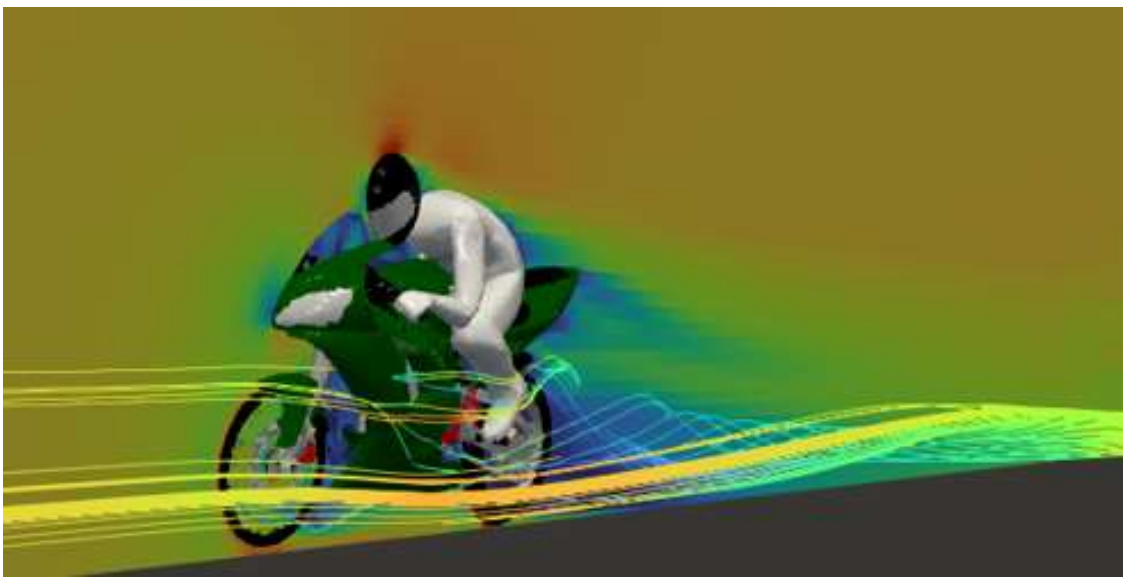

**Figure 14: ParaView motorbike on a remote VDI**

Figure 12, 13, and 14 are screenshots taken directly from the remote VDI.

To make the best use of cloud storage for visualisation, it is again necessary to understand your performance and cost requirements. For example, storing data on ANF volumes may not be needed if lower costs and less performance suit the business needs. In this case, blob storage may well be sufficient for your requirements

# Conclusion

When it comes to the best use of HPC to improve an engineer's productivity, there are many factors to consider, not least the cost per workflow, but also the time to solution. The methods of benchmarking and analysing the metrics shown in this white paper provide a starting point for finding the sweet spot for the time to run workflows versus cost. We can see that reducing the time to run a workflow may come with a greater cost, but if the time to solution is imperative then this becomes the overarching criteria. The HBv3 instances consistently outperform the HBv2 instances and at the current PAYG prices, the performance benefits are available with no increase in cost.

Productivity gains can be achieved by reducing the time taken to run a standard eight-hour workflow, normally run overnight for analysis the next day. Reducing the time to run this type of workflow to 3-4 hours, or even 2-3 hours could double or even triple an engineer's productivity as it would allow several iterations of a workflow to be run and analysed in a working day.

The results have also shown that by running much more complicated models with much higher cell or element counts, the serial decomposition or meshing processes take longer. The optimal balance of performance and price can be achieved by running the serial and parallel stages of the workloads separately on an appropriate number of cores.

We have seen that the Azure cloud provides resources to allow for the scaling up of workloads, maximising productivity. The cloud can also scale-out, simultaneously running multiple workloads that would otherwise be bottlenecked on a constrained on-premises shared HPC resource.

## About Red Oak Consulting

Red Oak Consulting's core business is high performance computing (HPC) consultancy. With a reputation for expert knowledge combined with real-world experience, we've delivered strategy, given procurement advice and helped to install more than 30 on-premises HPC systems (including several in the Top 10 of the Top500). To date, we have also completed more than 20 cloud-based projects, bringing first-class technical skills to working with 'early adopter' customers in both the higher education and commercial sectors.

All three white papers can be found at https://www.redoakconsulting.co.uk/whitepapers/. (*Note: The actual landing page address will be added before publication*).