UNIVERSITY OF
CAMBRIDGE

The Laboratory for Scientific Computing
Cavendish Laboratory, JJ Thompson Avenue
Cambridge, CB3 0HE

Airblast modelling on multiple Tesla units

Sean Lovett

SUMMARY

We used NVIDIA Tesla GPUs to accelerate the solution of hyperbolic partial differential equations, with application to airblast modelling.

FORMULATION

Many physical systems can be represented by hyperbolic partial differential equations. For a model problem we consider the **Euler equations** which describe inviscid compressible gas dynamics:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0$$
$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \rho \mathbf{u}) + \nabla p = 0$$
$$\frac{\partial E}{\partial t} + \nabla \cdot (\mathbf{u}(E + p)) = 0.$$

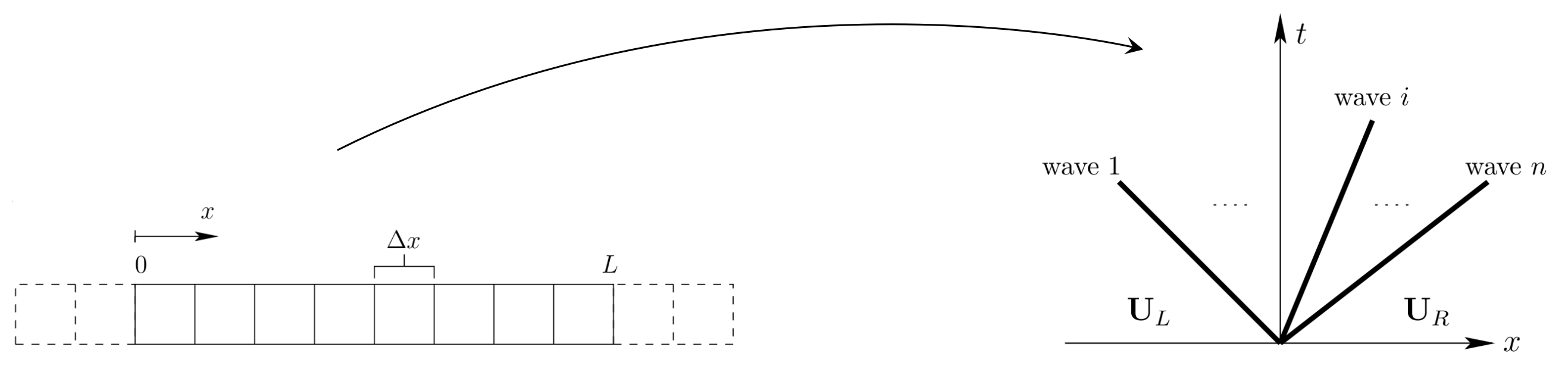
By linearising these equations we arrive at the **linear acoustic equations,** which describe incompressible sound propagation:

$$\frac{\partial p}{\partial t} + K_0 \nabla \cdot \mathbf{u} = 0$$
$$\rho_0 \frac{\partial \mathbf{u}}{\partial t} + \nabla p = 0$$

IMPLEMENTATION

We used C++ with CUDA to implement a second-order shock-capturing Riemann-problem-based method (the MUSCL-Hancock scheme) to solve the above equations [1,2].

This scheme solves a **Riemann problem** at each cell interface in order to compute flux between cells. The Riemann problem and the flux computation only require data local to the cell interface, making the problem well-suited to the CUDA model of computation.



1D domain showing computational cells. Fluxes are computed at cell interfaces by solving Riemann problems.

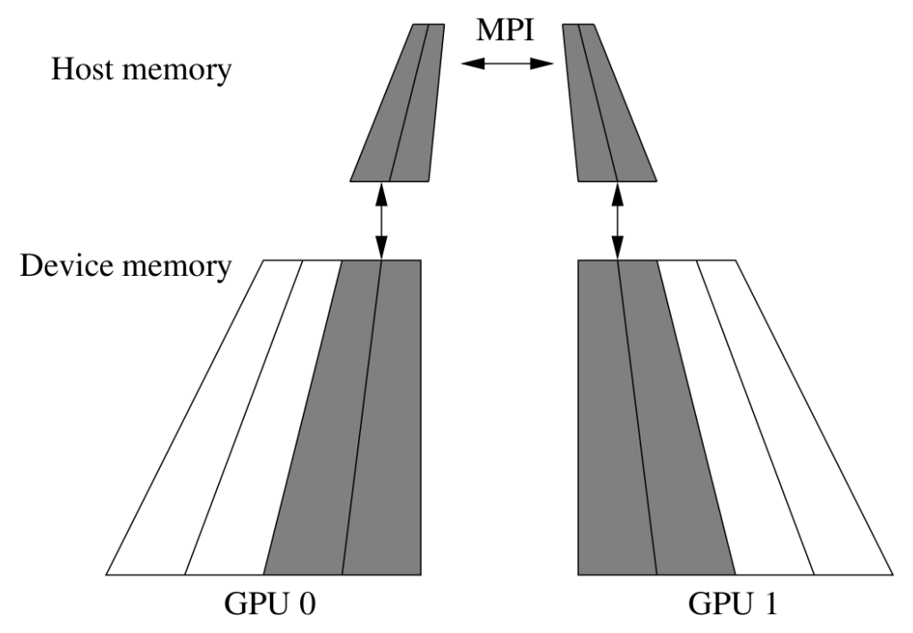
Structure of a general Riemann problem solution with initial data U_L and U_R.

A simplified form of the method is given below. Steps in red are parallelised on a thread-per-cell basis using CUDA.

```
Input:  Solution $U^n$ at $t^n$
Output: Solution $U^{n+1}$ at $t^{n+1}$
  Initialise ghost cells
  wavespeeds ← get_wavespeed_per_cell()
  dt ← cfl*dx/reduce(wavespeeds, max)
  fluxes ← get_flux_per_cell()
  $U^{n+1}$ ← $U^n$ + dx/dt*fluxes
  $t^{n+1}$ ← $t^n$ + dt
```

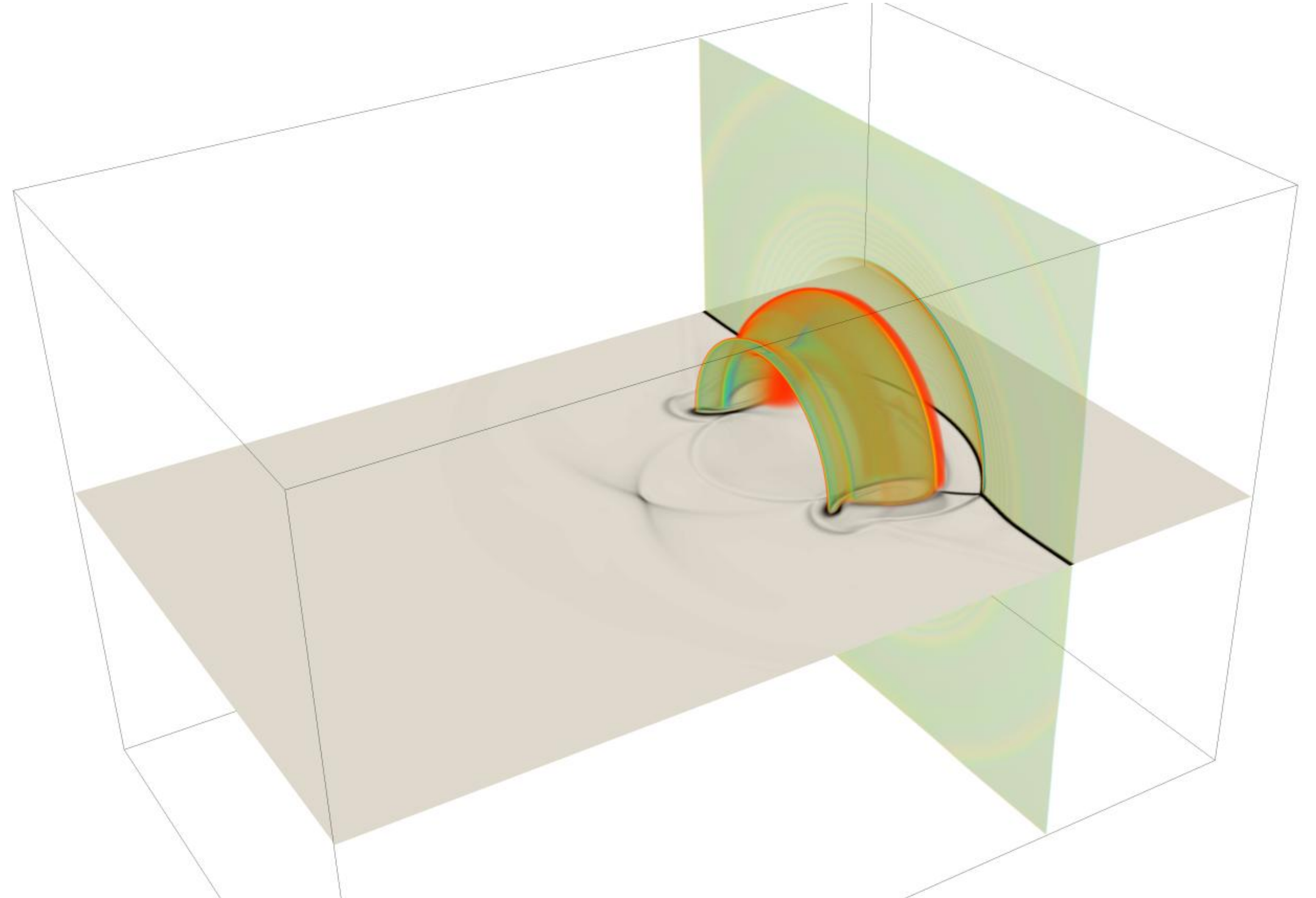
MPI IMPLEMENTATION

We parallelise the computation on the coarse scale using MPI. The figure on the right shows the memory transfer route between sub-domains for MPI computations. Currently we are investigating concurrent kernel execution and memory transfer for efficient scaling over many MPI processes.

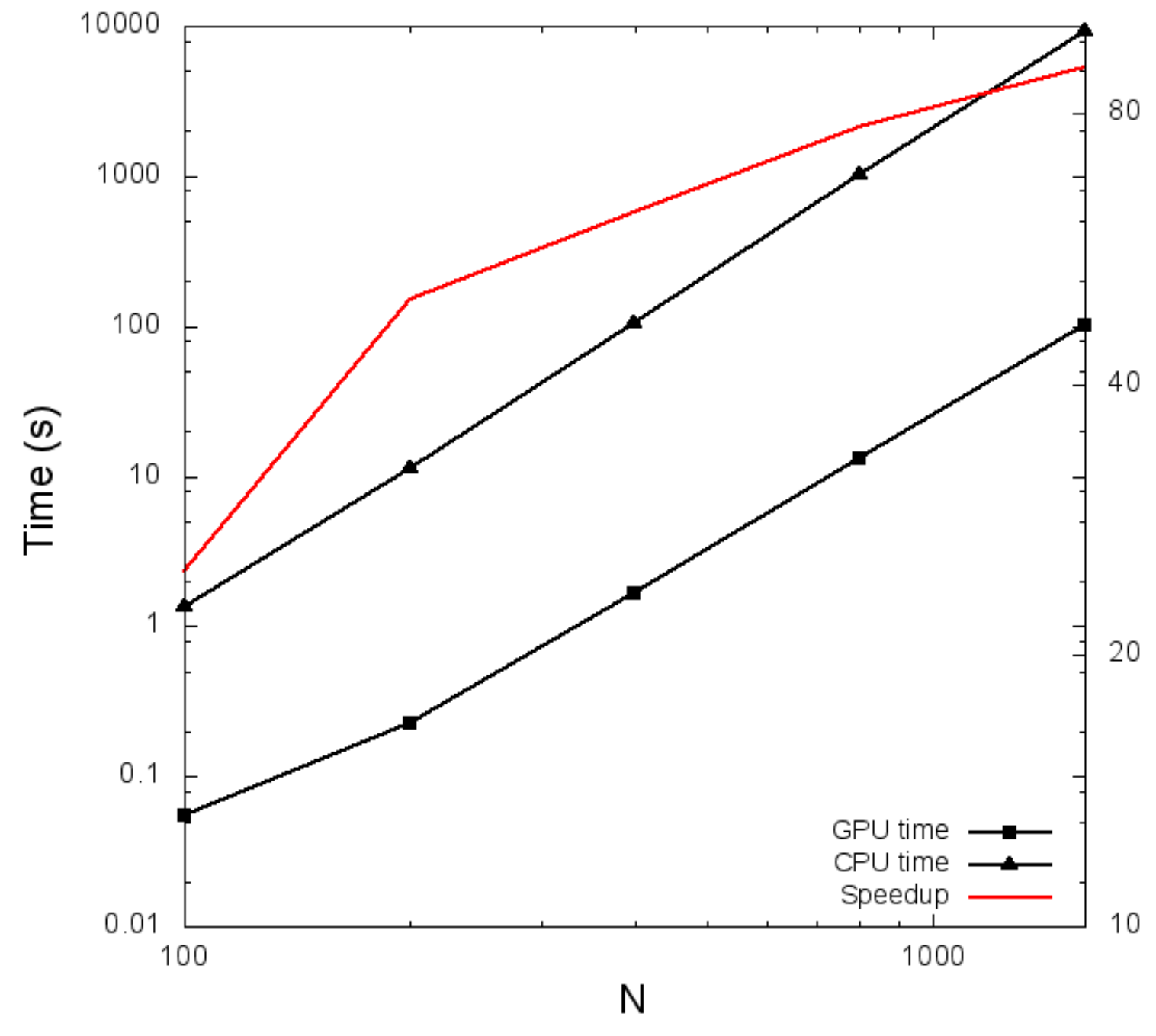


VALIDATION AND EVALUATION

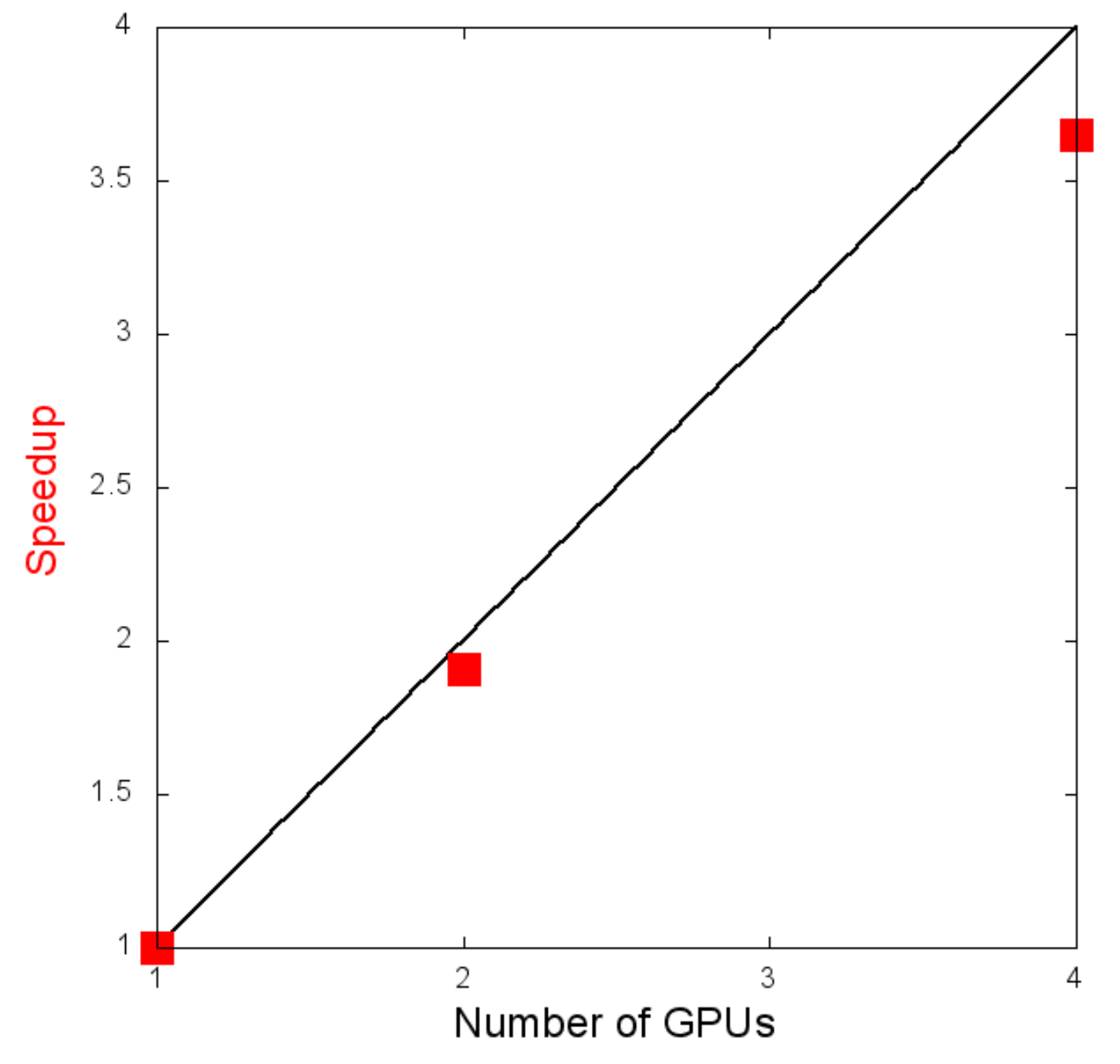
The resulting code was extensively validated and evaluated for accuracy and speed. For these tests we used an **exact Riemann solver** for the Euler equations. Sample results are shown in the figures below.



A 3D shock-bubble interaction computed with 400x400x600 cells. The compute time on 4 Tesla GPUs was 24 minutes; the same computation on 4 CPUs would take about 12 hours. Instantaneous density (volume render) and density gradient (surface) are visualised.



Scaling for the 2D Euler equations on a single GPU. N is the number of cells per dimension.



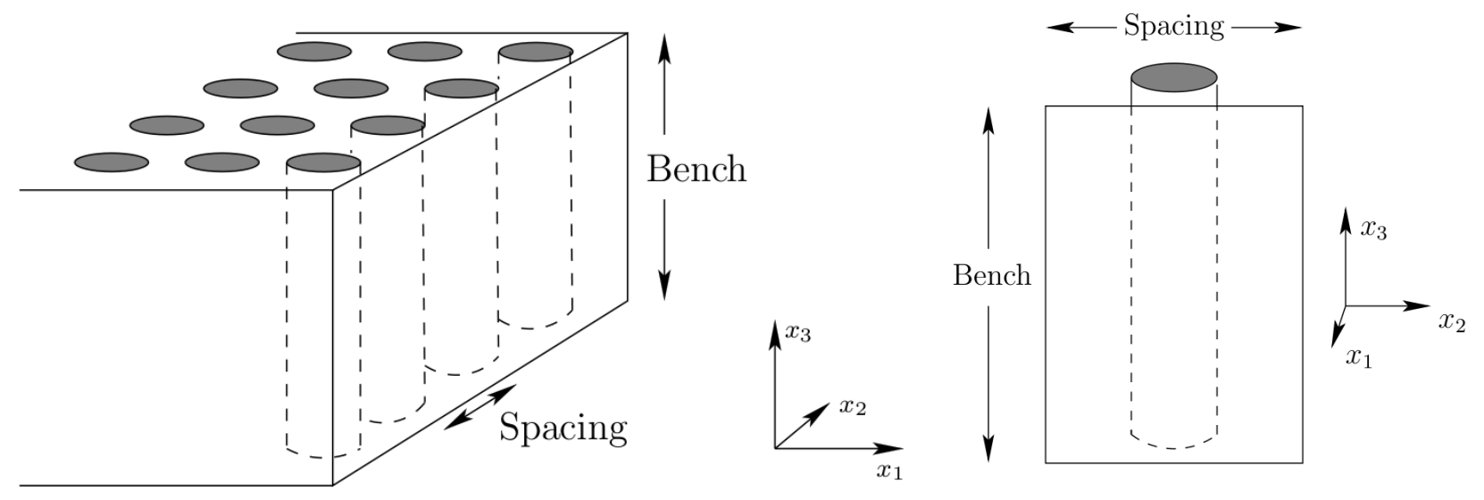
Strong scaling for the 3D Euler equations over multiple GPUs. Ideal scaling (line) and observed scaling (symbols).

AIRBLAST MODELLING

Here we demonstrate an application of this work to modelling airblast (acoustic waves) generated by bench mining operations [3]. Airblast is a primary source of noise and is governed by strict regulations. Fast time-to-solution is important when modelling this problem. Ideally models could be run directly on laptops in the field.

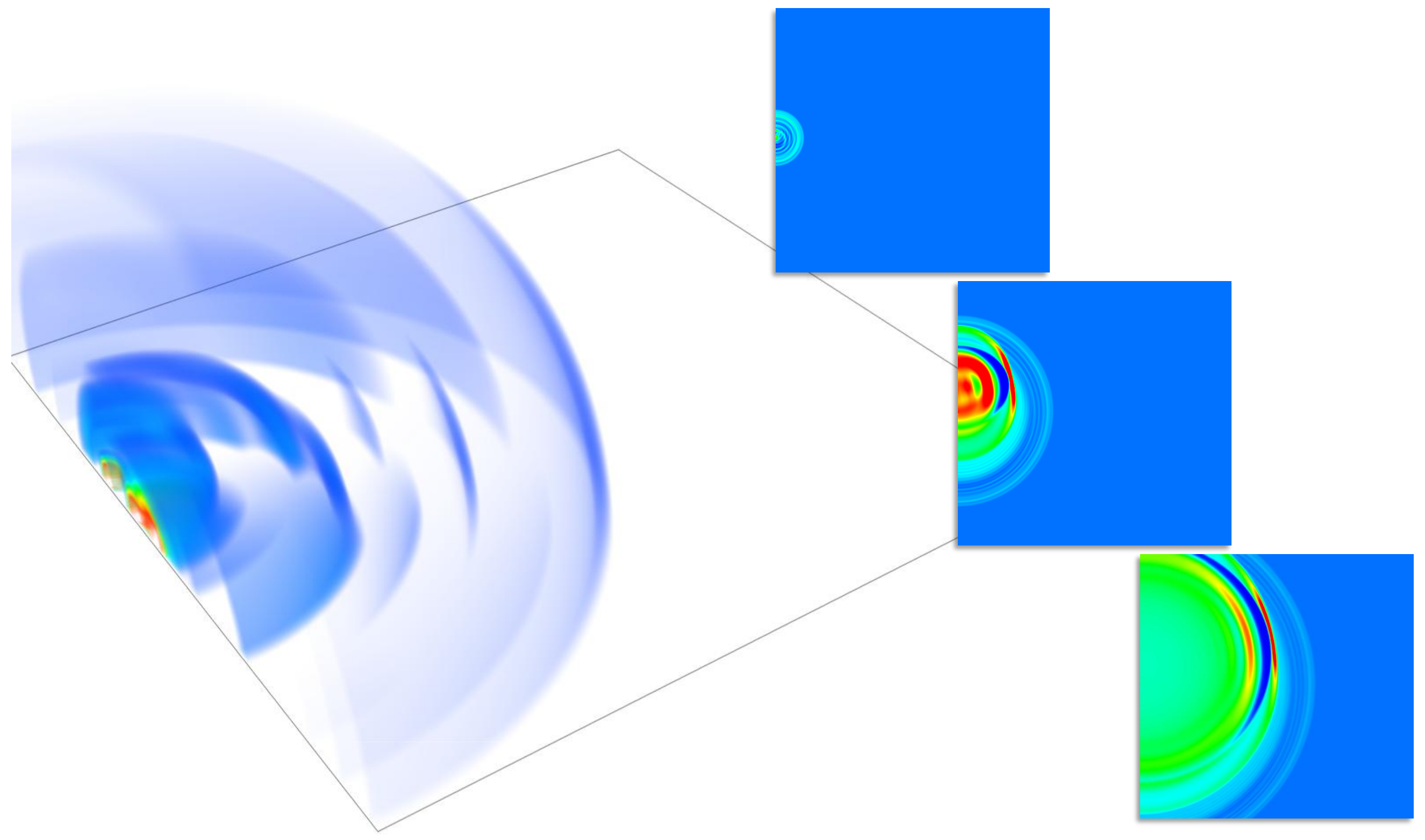
We consider a model in which airblast is generated by piston-like displacements of the free rock face:

A quarry bench blast in Australia (image courtesy Orica).



Piston model for airblast generation.

We solve the linear acoustic equations using these displacements as input. The relative timing of these displacements affects the direction of the resulting airblast. With faster simulations, more timing programmes can be tested to fine-tune the direction and intensity of the airblast.



A 3D airblast computation (left, 256^3 cells) and a 2D computation in plan view (right, 2000^2 cells). Instantaneous pressure is visualised.

REFERENCES

[1] S. Lovett. Many-Core Riemann-Problem-Based Methods for Compressible Flow. Master's Thesis, Department of Physics, University of Cambridge, 2009.
[2] S. Lovett and N. Nikiforakis. Implementation of Riemann problem based methods on GPUs (in preparation).
[3] S. Lovett, N. Nikiforakis and A. Minchinton. Airblast modelling on GPUs (in preparation).

Contact: sdl30@cam.ac.uk, http://www.lsc.phy.cam.ac.uk/people/sean_lovett.shtml

Academic supervisor: Dr N. Nikiforakis
Industrial supervisor: Dr A. Minchinton

Acknowledgements: This work was sponsored by Orica Mining Services
Tesla GPUs were donated by NVIDIA

ORICA