

HF node scaling

For large HF jobs it is desirable to increase the number of nodes the job is submitted to in order to decrease the wall clock time of the job.

Scaling has already been implemented for LF jobs using an iterative approach based on the wall clock time estimation.

Previous implementation

Given a node time threshold factor we wish to minimise the number of nodes such that the WCT taken to complete the job is less than the number of nodes multiplied by the threshold time factor:

```
WCT < nodes * node_threshold
```

This time factor means that as the number of nodes used increases the allowable WCT also increases.

Currently the threshold factor for LF jobs is 15 minutes per node, with a minimum of 4 nodes.

An iterative approach was used where each time a node was added the WCT was estimated again.

This had the issue that WCT estimation is not accurate outside of the bounds the underlying neural network was trained on.

HF implementation

In order to get a more accurate estimation of the WCT taken for a job to complete it is assumed that the number of core hours required for a job remains constant, regardless of the number of nodes used.

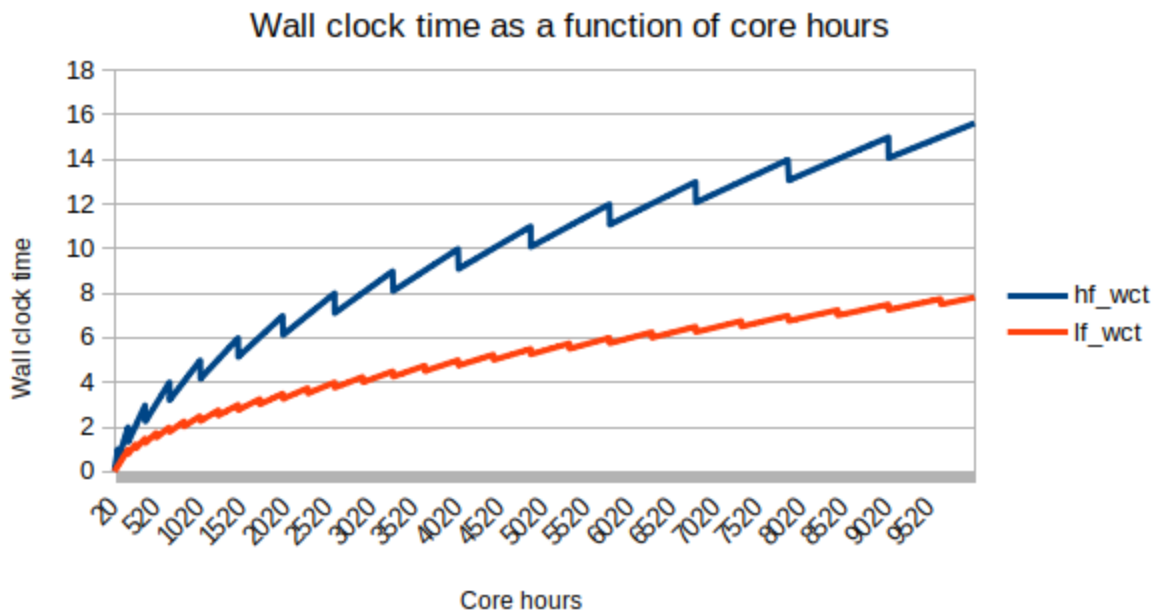
This allows us to use the algebraic expression:

```
core_hours = nodes * nodes * node_threshold * number_of_cores_per_node
```

We can now minimise the number of nodes such that the first formula holds.

We set the node_threshold for HF jobs to be one hour per node.

The new implementation has the following characteristics:



Error checking

In the case that the scaled number of hours is below the initial number, or above the maximum allowed of 66, then the nearest within that range is used.

If the WCT is calculated to be longer than 24 hours, then the minimum number of cores to bring it below 24 hours is used.

Test results

A test run was done with the WairarapNich_HYP01-47_S1244 realisation from Cybershake.

HF calculations were done using a single node taking approximately 8hrs 50mins, and then with 3 nodes, taking 1hr 4mins.

The resulting HF.bin files were determined to be identical.

Further testing showed either conservation or a reduction in the total number of core hours used.