Scaling Cellular Automata beyond 100,000 cores
Coarrays vs MPI + OpenMP + do concurrent
Anton Shterenlikht
The University of Bristol

Abstract
CASUP (pack.of.coarrays) is a generic, HPC cellular automata (CA) library. In this work it was applied to 3D Ising magnetisation calculations. Scaling of two halo exchange (HX) methods (Fortran coarrays, MPI) and of three CA loop routines (simple nested loop, do concurrent, OpenMP) was measured on ARCHER up to full machine capacity, 4544 nodes (109,056 cores). Ising energy was calculated with MINIMAL and Fortran 2018 COARRAY collectives. Using fully populated nodes and no threading gave the highest performance, probably because the Fortran model is perfectly load balanced. MPI HX scaled better than coarray HX, which is surprising because both algorithms are pair-wise "handshakes" (SCREW/SEND/RECV/ATTALL vs SYNC IMAGES). The University of Bristol

Introduction
CASUP solidification scaled to 32k cores on HECToR [1]. CA SuperFlowEM (CAFES, coarray) cellular automata finite element method (CAFE) multi-scale fracture simulation scaled to 8k cores on ARCHER [2]. In this work CASUP has been completely re-designed:

1. All synchronisation is now hidden from the user. CASUP automatically inserts the minimum number of sync calls to ensure data integrity. The user never needs to worry about data integrity, i.e., no sync calls in CASUP miniapps.

2. CASUP is now completely modular, i.e., the CA kernels, the halo exchange or collective routines can be swapped with ease, allowing performance analysis of multiple combinations of Fortran coarray, MPI, OpenMP and do concurrent (Fig. 1).

Halo exchange (HX)
Two possibilities for HX exist when using coarrays: (1) creating the whole of the CA model of coarrays, and (2) using coarrays only for CA halos. HX in the second method involves a separate step, copying of the boundary CA data into coarray arrays, as shown in Fig. 2.

3D Ising magnetisation
Extension of the QXR Vichniac's 2D rule [3] to 3D was implemented:

- Spin of 0 (down) or 1 (up).
- 3D mask array.
- A spin flips if 3 neighbours are 0 and the other 3 are 1.

Results

Conclusions

- MPI HX performs better than Fortran coarrays HX.
- Virtuous loop: more coarray usage → better support by vendors → higher performance → more coarray usage, etc.
- Whole model coarrays perform worse than when coarrays used only for halos. This is unexpected.
- Any amount of threading lowers performance. This is likely due to very little load imbalance in Ising magnetisation CA.

Future

- 1D vs. 3D domain decomposition will be explored. 3D decomposition uses smaller messages, but x3 more messages:

  - Asynchronous coarrays HX using F2018 EVENTS + atomic. This might improve performance, particularly if load balancing is poor.
  - Allocatable vs F2008 CONTIGUOUS assumed shape arrays.

Acknowledgments
This work used the ARCHER UK National Supercomputing Service (http://www.archer.ac.uk).

References