UK Atomic Energy Authority

S.Pamela

Milestone Xmas Report: Exploitation of SGI UV-2000 Shared Memory Nodes

This work was funded by the RCUK Energy Programme
[Grant number EP/P012450/1]

The Machines

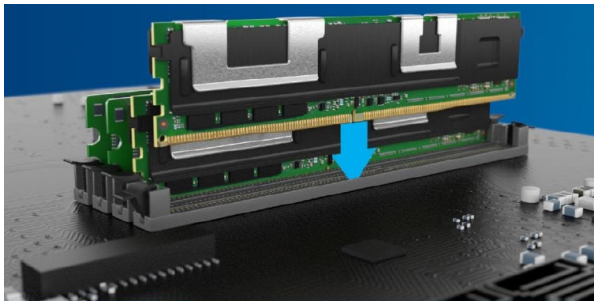
UK Atomic Energy Authority

SGI UV-2000 nodes (CCFE)

- 3TB (on main node)
- 768 intel-X-E5-4650L
- 6TB nodes not yet connected

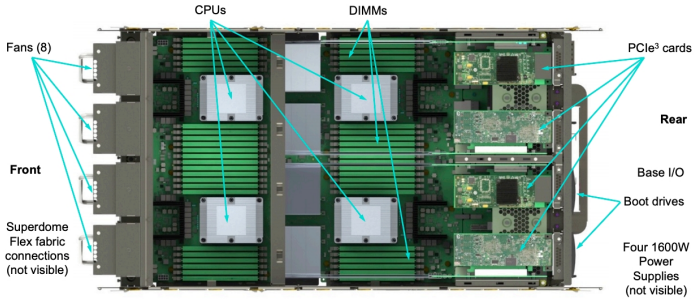
Intel-Optane (Swindon)

- 6TB
- 48 dual cores



HPE SuperdomeFlex (Leicester)

- 6TB
- 144 intel-X6154 cores



Marconi (Cineca, Italy)

- 196GB/node
- 48 cores/node

(conventional HPC reference)

CCFE CULHAM CENTRE FOR FUSION ENERGY

UV-2000 specifics

UK Atomic Energy Authority

Some of the SGI-NUMA optimisation functions are not available!

- eg. dplace, omplace, numactl…

But, the most important one is: taskset

- specifies list of cores we want to use in mpirun
- required for openMPI (tells MPI where it can find OMPs), eg.

```
>> export OMP_NUM_THREADS=50
>> mpirun -np 14 taskset -c 0-7,384-391,8-15,392-399,16-23,400-407,24-31,408-415,32-39,416-423,40-47,424-431,48-55,432-439,56-63,440-447,64-71,448-455,72-79,456-463,80-87,464-471,88-95,472-479,96-103,480-487,104-111,488-495,112-119,496-503,120-127,504-511,128-135,512-519,136-143,520-527,144-151,528-535,152-159,536-543,160-167,544-551,168-175,552-559,176-183,560-567,184-191,568-575,192-199,576-583,200-207,584-591,208-215,592-599,216-223,600-607,224-231,608-615,232-239,616-623,240-247,624-631,248-255,632-639,256-263,640-647,264-271,648-655,272-279,656-663,280-287,664-671,288-295,672-679,296-303,680-687,304-311,688-695,312-319,696-703,320-327,704-711,328-335,712-719,336-343,720-727,344-351,728-735,352-359,736-743,360-367,744-751,368-375,752-759,376-383,760-767      PROGRAM
```

- Doesn't accept filename (I think?)
- Not sure if order is important
- The above list is how CPUs are ordered on the node. ie.

```
NUMA node0 CPU(s):     0-7,384-391
NUMA node1 CPU(s):     8-15,392-399
NUMA node2 CPU(s):     16-23,400-407
NUMA node3 CPU(s):     24-31,408-415
NUMA node4 CPU(s):     32-39,416-423
```

- Still need to check if efficiency affected with

```
>> mpirun -np 14 taskset -c 0-767      PROGRAM
```

- Still need to check if multiple jobs on separate CPUs run OK

→ requires not just separation of cores, but also memory!

The list will need to be managed & hidden from user in #PBS...

CCFE CULHAM CENTRE FOR FUSION ENERGY

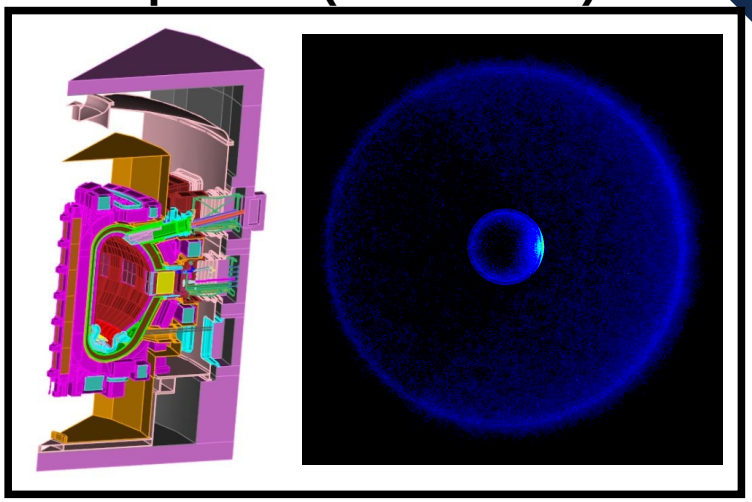
The Codes

UK Atomic Energy Authority

OpenMC:

- memory-intensive for large CADs
- CAD stored in each MPI process
  - → memory usage linear with MPI
  - → low memory for pure OMP

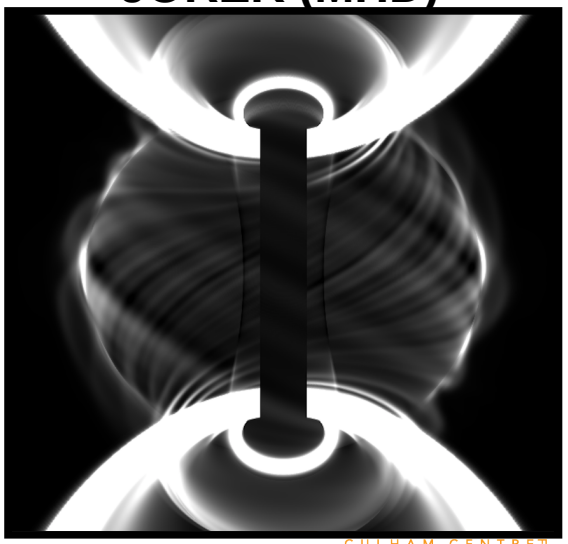
OpenMC (neutronics)



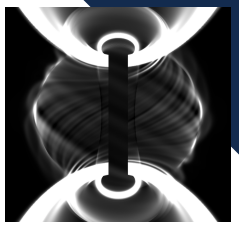
JOREK:

- memory-intensive for large matrix solves & preconditioner storage
- memory usage ~independent of MPI/OMP (relative to OpenMC)
- but still increases with MPI

JOREK (MHD)

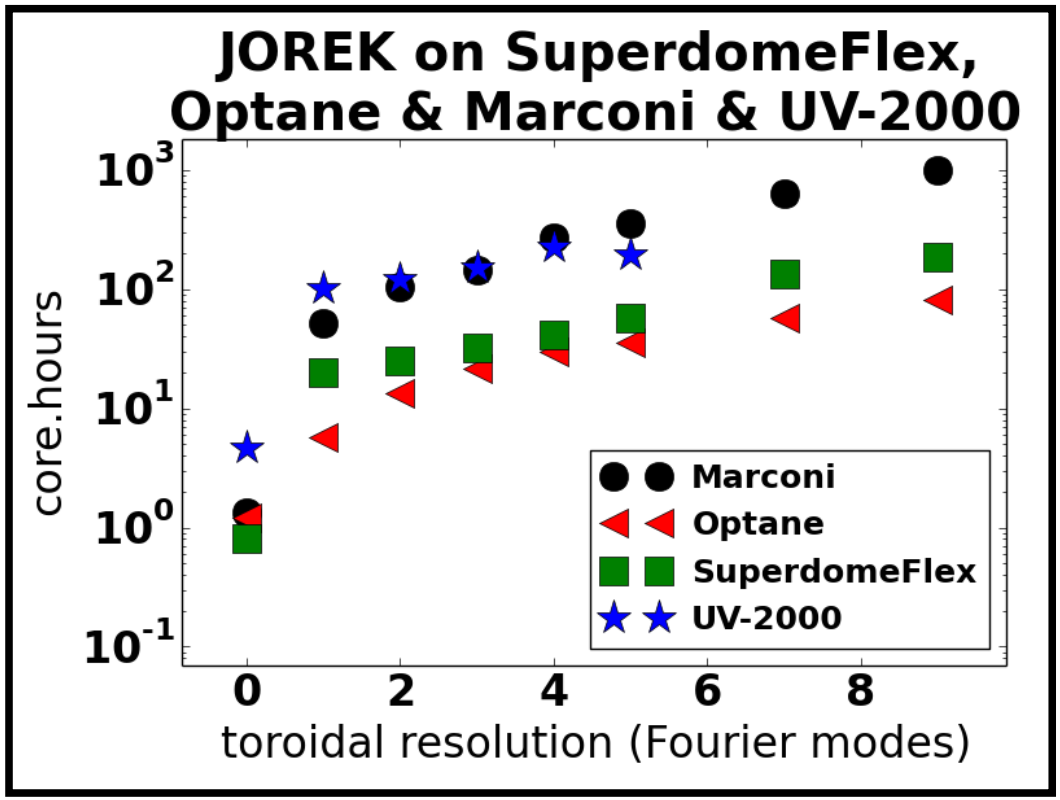


JOREK timing



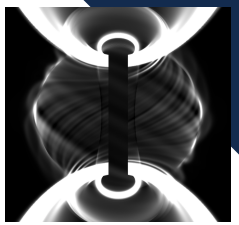
UK Atomic Energy Authority

- Large gain in terms of CPU compute time
- Optane doing slightly better than SuperdomeFlex
  (gain is more systematic)
- UV-2000 not doing as well as expected
  - → UV-2000 gain seems to improve for large cases
  - → need to test the other nodes (main node limited to 3TB)



CCFE CULHAM CENTRE FOR FUSION ENERGY

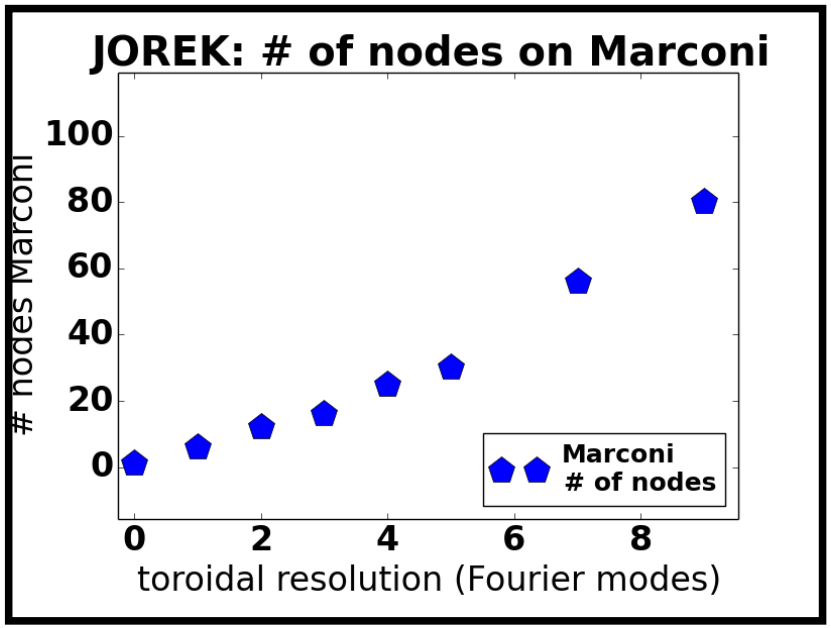
JOREK node-# on Marconi

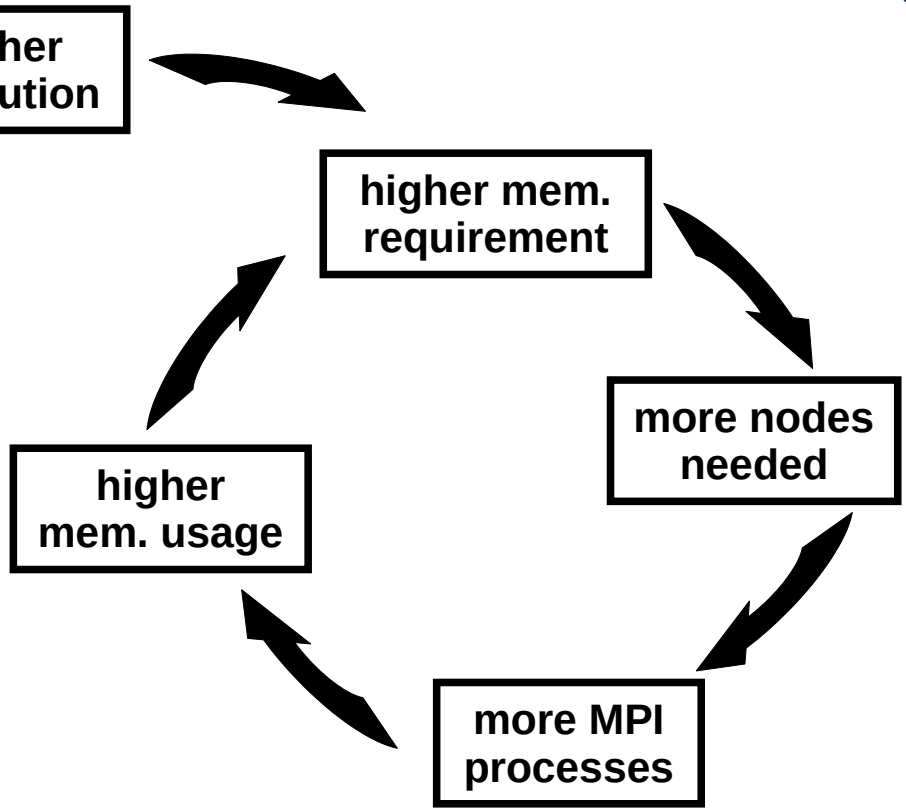


UK Atomic Energy Authority

On Marconi:

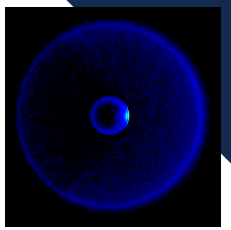
- relatively large memory/node (196GB/node)
- the vicious circle on conventional HPC:





CCFE CULHAM CENTRE FOR FUSION ENERGY

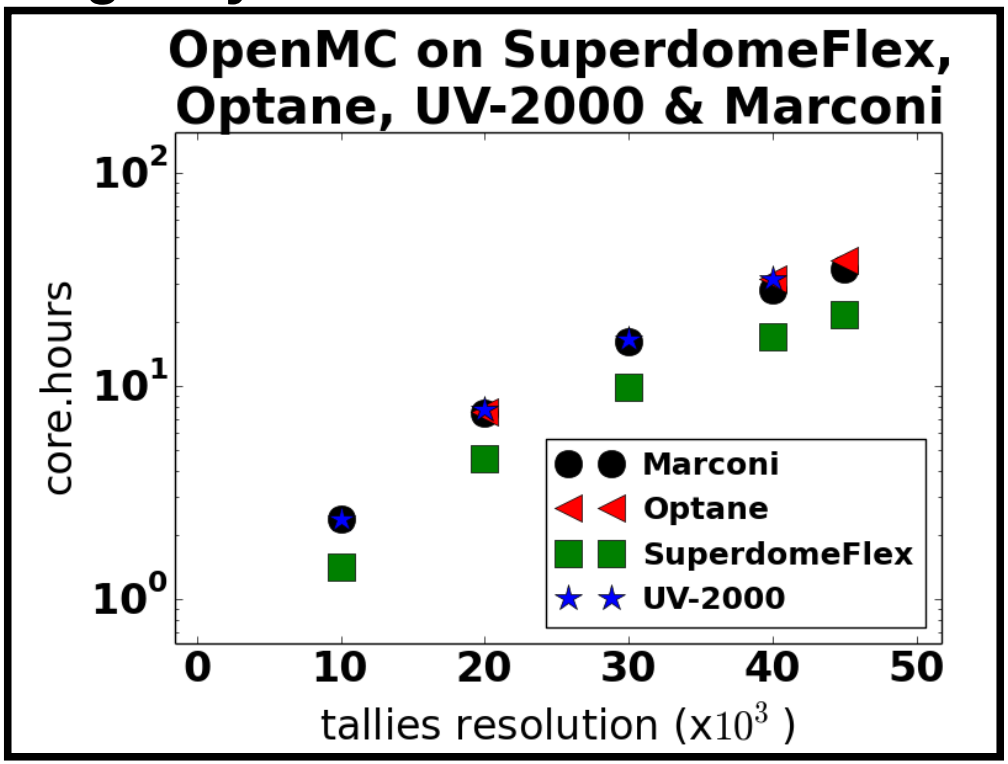
OpenMC tally-scaling



UK Atomic Energy Authority

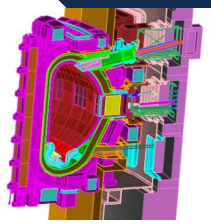
Using J.Shimwell's workshop Task-4 (simplified ST)

- **scaling in tally numbers (resolution)**
- **using pure-MPI → high memory usage**
  (ie. very artificial, ideally would use pure-OMP…)
- **SupernodeFlex faster**
- **UV-2000 doing very well**



CCFE CULHAM CENTRE FOR FUSION ENERGY

OpenMC Strong Scaling

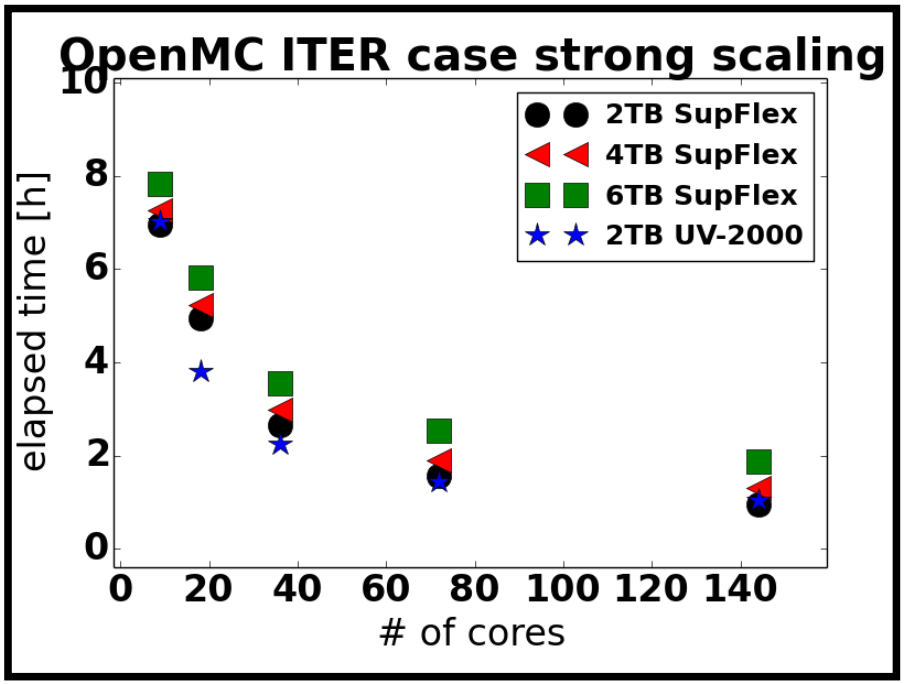


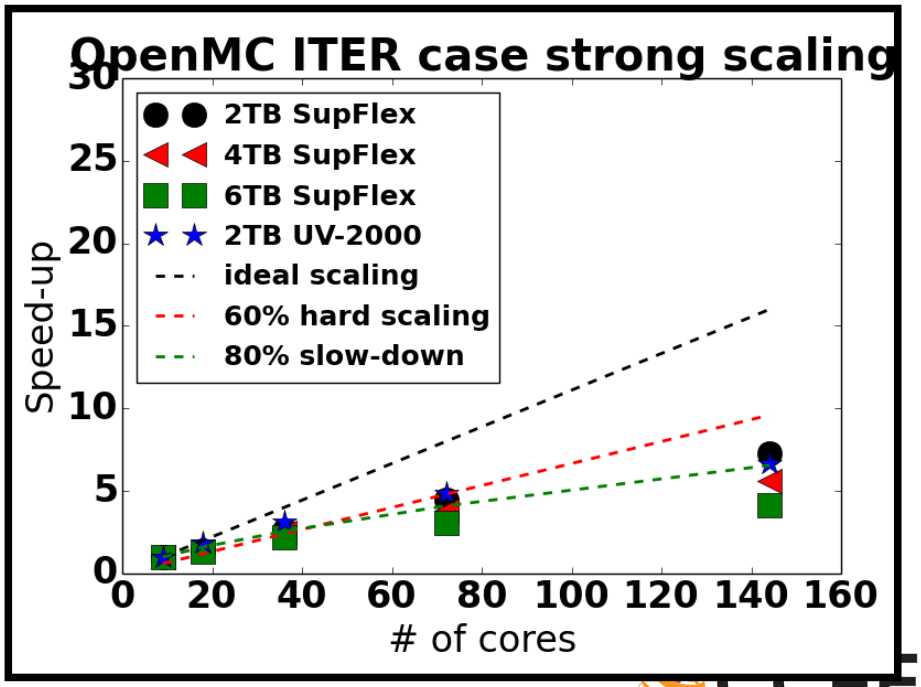
UK Atomic Energy Authority

Using A.Davis's ITER case

- using DAGMC with Double-Down to load CAD
- pure-MPI strong-scaling: 60% (80% for local scaling)
- increasing tallies from 2TB to 6TB
- here SuperdomeFlex not faster...
- UV-2000 doing very well

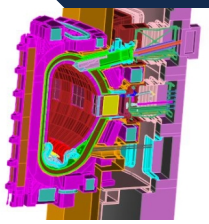
[note: 'hard scaling' means minimum-cores is reference
'slow-down' means the previous case is reference]





CCFE CULHAM CENTRE FOR FUSION ENERGY

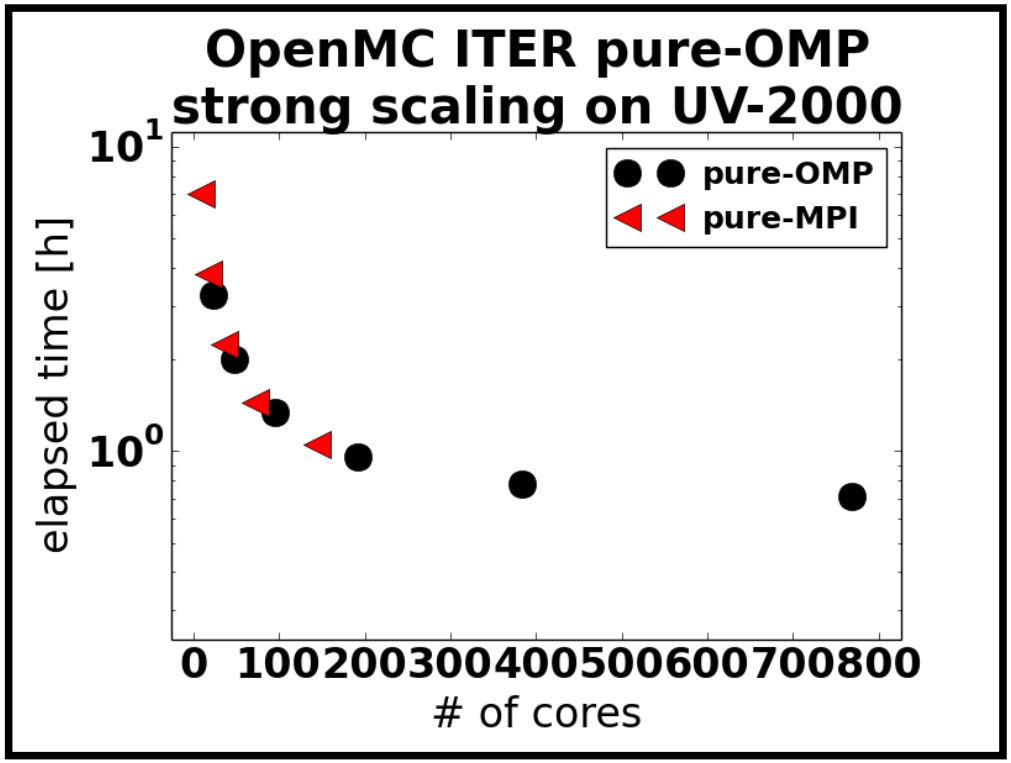
OpenMC: MPI vs. OMP



UK Atomic Energy Authority

Using A.Davis's ITER case

- **on the UV-2000**
- **strong-scaling using pure-MPI vs. pure-OMP**
- **no difference at all**
- **the limit on # of MPI processes (~200?) is determined by available memory of course (3TB on main UV-node)**



CCFE CULHAM CENTRE FOR FUSION ENERGY

In-Memory Data Transfer

UK Atomic Energy Authority

Developing C++ codes using the library BOOST-interprocess
https://www.boost.org/doc/libs/1_63_0/doc/html/interprocess.html

Codes developed can be found on github:
https://git.ccfe.ac.uk/pstanis/spamela_ukaea_logs/-/raw/master/UV2000_Milestone_2021/example_boost_interprocess.cpp
https://git.ccfe.ac.uk/pstanis/spamela_ukaea_logs/-/raw/master/UV2000_Milestone_2021/install_and_test.sh

- creation/retrieval of shared-memory blocks
- can handle any kind of data (eg. text, binary)
- tested with various binary files
images, tar-balls, zip-balls, executables, text-files
- up to the full memory capability of the node (3TB)
- results extremely satisfactory
fraction of a second to retrieve data
compared to 10s of minutes for file-readings

This technology could be extended to develop data-mirrors for large data-bases like the full MAST or MAST-U campaigns, enabling extensive data-mining with AI software.
Such plans are now being considered.

Summary

OpenMC:

- Milestone goals achieved, but could do more...
- Bigger CAD: ITER case uses 100GB in pure OMP
  - → whole argument is that large-CADs can only run on high-memory HPC...
  - → OR… need to solve long-int issue in OpenMC (with high tally-# OpenMC fails)
- Look at timing of OpenMC set-up time (not just particles)
  - → ie. dagmc.h5 load, initialisation, MPI comms etc.

JOREK:

- Compiler issues solved, and tests successful
- To bear in mind for deployment:
  - → intel compilers provided by CODAS broken (problem passing MPI comm from Fortran to C/C++)
  - → hopefully intel-2020 compiler works (action on D.Robson)

In-memory HDF5:

- BOOST-interprocess library gives very satisfactory result.
- Could provide a plugin for UDA and enable data-mirrors for AI data-mining
- Check with Leo Ma
  - → he had some ideas to use UV-2000 with SQL database, but said he needed to do some ground development first...