Science & Technology Facilities Council

Technical Report

DL-TR-2017-001

Weak and strong scaling of the CIAN2 code coupling mini-app

AD Taylor

June 2017

Enquiries concerning this report should be addressed to:

Chadwick Library
STFC Daresbury Laboratory
Sci-Tech Daresbury
Keckwick Lane
Warrington
WA4 4AD

Tel: +44(0)1925 603397
Fax: +44(0)1925 603779
email: librarydl@stfc.ac.uk

Science and Technology Facilities Council reports are available online at: http://epubs.stfc.ac.uk

ISSN 1362-0207

Weak and strong scaling of the CIAN2 code coupling mini-app

Andrew D. Taylor[1,2]

ABSTRACT

Numerical simulation of scientific systems often involves modelling different spatial and temporal scales and different physical processes to produce a single set of results. In order to achieve this, scientific codes need to be efficiently coupled so that the transfer of data between different codes can be performed effectively on large, multi-node computing clusters.

The following investigates the strong and weak scaling of the CIAN2 mini-app on an Intel Xeon IvyBridge cluster over a range of 3-4600 processes. The CIAN2 mini-app is a 'toy' application designed to demonstrate the principles of code coupling. Two 3D meshes are coupled together using MPI communication, with the user specifying the number of MPI processes and the size of the source and target mesh, among other things.

Results show that the application demonstrates excellent strong scaling, especially for larger problems. The overhead of communication is less significant for large problems ran across many processes than for small problems ran across many processes. Weak scaling is also good for CIAN2, with results showing that time to solution remains in the same order of magnitude when problem size and MPI process number is increased by the same factor.

The above results, while not scientifically significant, show a benchmark standard of parallel performance that other code coupling problems can compare their performance with on this system.

This work forms part of the EPSRC Service Level Agreement funded Software Outlook project

Keywords: Code Coupling, Weak Scaling, Strong Scaling, Intel Xeon (IvyBridge), CIAN2, CCP

June 12, 2017

[1]The Hartree Centre, Science and Technology Facilities Council, Daresbury Laboratory, Sci-Tech Daresbury, Daresbury, Warrington, WA4 4AD, UK.

[2]Correspondence to: andrew.d.taylor@stfc.ac.uk

1 Introduction

Numerical modelling of scientific systems involves considering different spatial and temporal scales and different physical processes that often need to be linked in order to gain a better understanding of the interactions between different processes. There are many examples of these linked systems problems, such as earth systems modelling, where atmospheric, oceanic and land surface processes are coupled with each other to, for example, simulate the Earth's future climate for centuries.

Code coupling can involve multi-scale modelling (for example, molecular dynamics coupled to larger scale material properties), multiphysics modelling (for example, linking chemical kinetics and fluid dynamics) and/or a combination of both. Code-coupling problems on large high performance computing (HPC) clusters need to scale successfully across multiple CPUs and nodes in order for them to make the most of the computing power available. Ideally, for a given problem size, as one doubles the number of CPUs available, time to solution should halve, indicating a strongly scaling code. Additionally, if one increases problem size and CPU count by the same factor, performance should be comparable, thus demonstrating a weakly scaling code.

2 CIAN2 mini-app

The CESAR Integrated Analytics Proxy Applications (CIAN2) Mini-app from Argonne National Laboratory, USA (`https://cesar.mcs.anl.gov/content/software/coupling_analytics`) is a set of 'toy' applications that are used to demonstrate the principles of code coupling.

Code coupling occurs between two different 3D mesh geometries using MPI communication. The user specifies a number of parameters for each experiment. These are:

1. Source and target mesh types - either hexahedral or tetrahedral.
2. Mesh size (regular grid vertices per side; all problems use same dimensions in x,y and z).
3. Number of MPI processes.
4. Number of iterations to simulate convergence to solution.
5. The 'slabbiness' of the source and target domain decomposition: this can be either homogeneous cubes or heterogeneous slabs.

Two tests are performed to determine the scaling performance of CIAN2. In both, the source mesh is tetrahedral and the target mesh is hexahedral, with both meshes utilising identical sizes for each experiment. Domain decomposition is homogeneous cubes with each experiment iterating 10 times.

3 Results

The following set of results were performed on the Napier Hartree Centre system, a NextScale cluster of 360 nodes. Each node has 2 x12 core Intel Xeon processors (E5-2697v2 2.7GHz) and 64GB RAM. Interconnect is Infiniband from Mellanox (FDR Connect-IB 56 GB/s).

3.1 Strong Scaling

Figure 1 shows the result of a strong scaling test for different mesh sizes on the Napier system. The CIAN2 mini-app scales very well up to a large (>1500) number of MPI processes for a range of grid sizes from 48^3 to 576^3. Scaling for the smallest grid size levels off once 192 processes is exceeded, but for all other problem sizes the points all lie on a straight line on the log-log plot.

To investigate how the individual functions within the CIAN2 coupling code perform an additional set of experiments were run for the 384^3 grid size using instrumentation from the TAU profiler. Figure 2

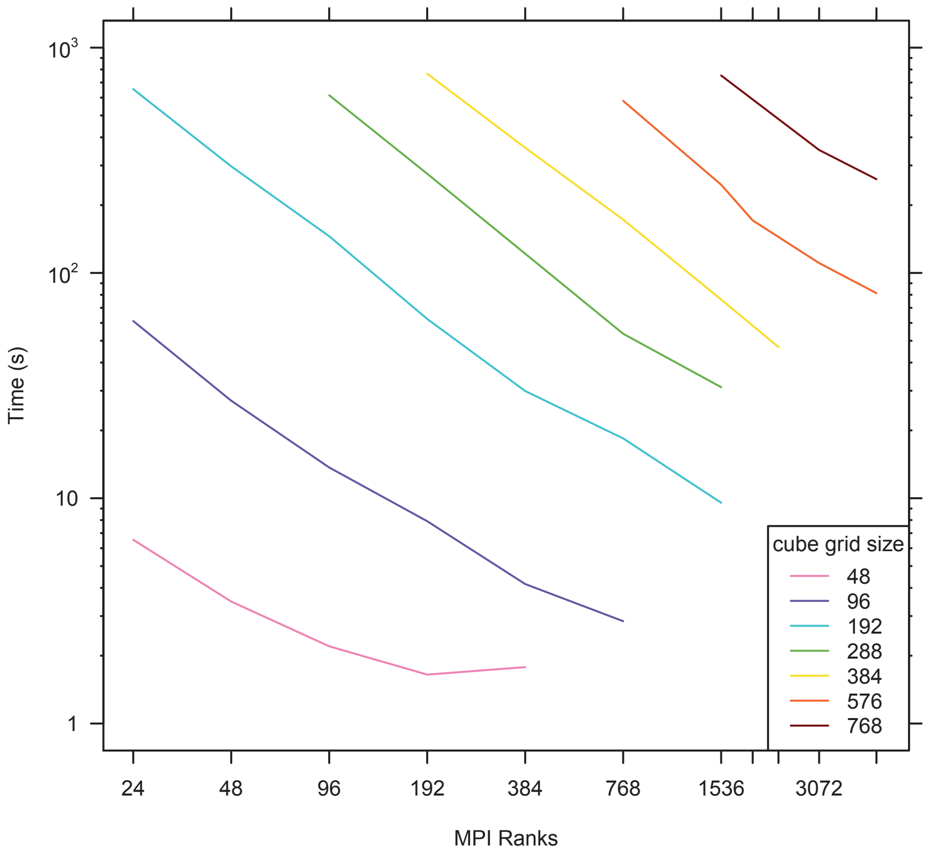

Figure 1: Strong scaling of CIAN2 mini-app on Napier NextScale cluster

shows the results of this for a range of different MPI process numbers. Time-dominant functions of the CIAN2 source code such as 'coupling', 'initProjection' and 'projectField' scale very well with increasing MPI processes. The function 'prepMeshes' reaches a performance limit above 384 MPI ranks. Finally, the MPI library function MPI_Init increases with increasing MPI processes. Thus, as total time decreases with increasing MPI processes, MPI_Init takes up 1 % of run-time at 192 ranks but 10 % of run-time at 1536 ranks. This suggests that MPI_Init becomes a limiting factor to strong scaling as one increases MPI processes.

3.2 Weak scaling

To test weak scaling, the number of MPI processes and total grid points was concurrently doubled from 3 to 1536 processes and from 1.1 x 10^5 to 5.7 x 10^8 elements. Ideal weak scaling should show that each of these runs takes approximately the same time. Figure 3 shows the result of this analysis using the TAU profiler. Total time (represented by the 'coupling' function) lies in one order of magnitude (between 40-80 seconds) for all experiments. There is a tendency for experiments with larger numbers of MPI ranks and elements to take slightly longer; this can be partially explained by the increase in time spent in the projectField function as one increases MPI processes and elements.

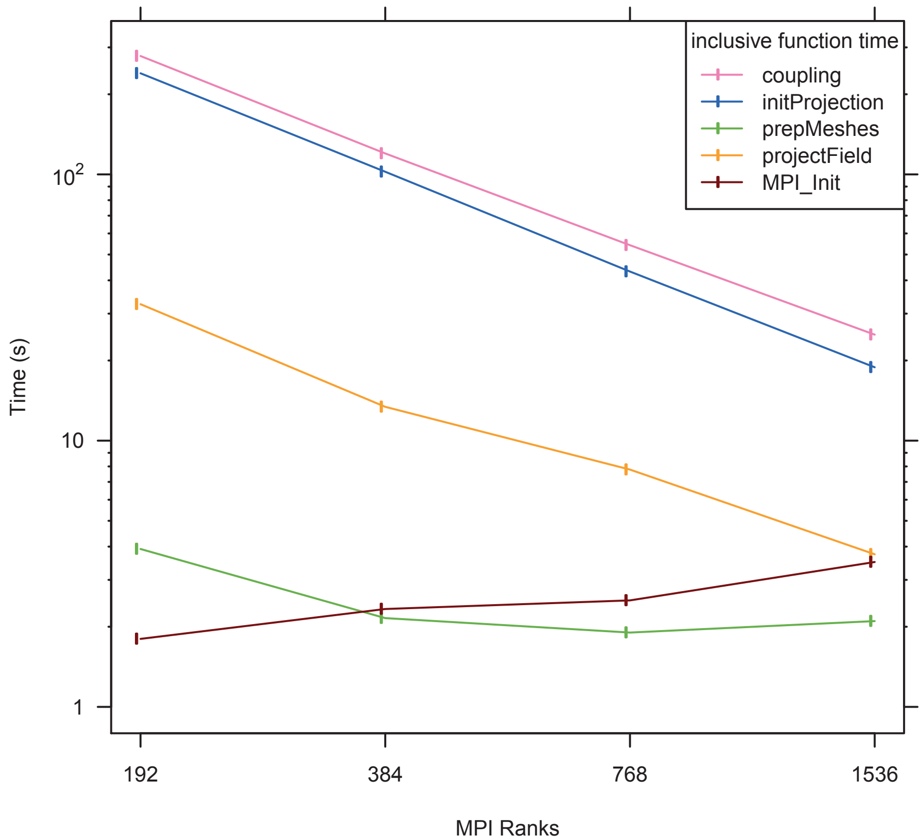

Figure 2: Strong scaling of CIAN2 mini-app inclusive function time on Napier NextScale cluster generated using TAU and a grid dimension of 384^3.

4 Conclusions

The above show that CIAN2 mini-app exhibits good strong and weak scaling across the Napier cluster from 1 to 200 nodes. This result is not surprising, because the CIAN2 mini-app is a 'toy' problem designed to demonstrate the principles of code coupling. Therefore, these results can be taken forward as a benchmark against which future code coupling work on the Napier cluster can be compared with. Real-world problems will likely not reach the level of performance demonstrated here, but the data here demonstrates the performance that is theoretically possible on Napier.

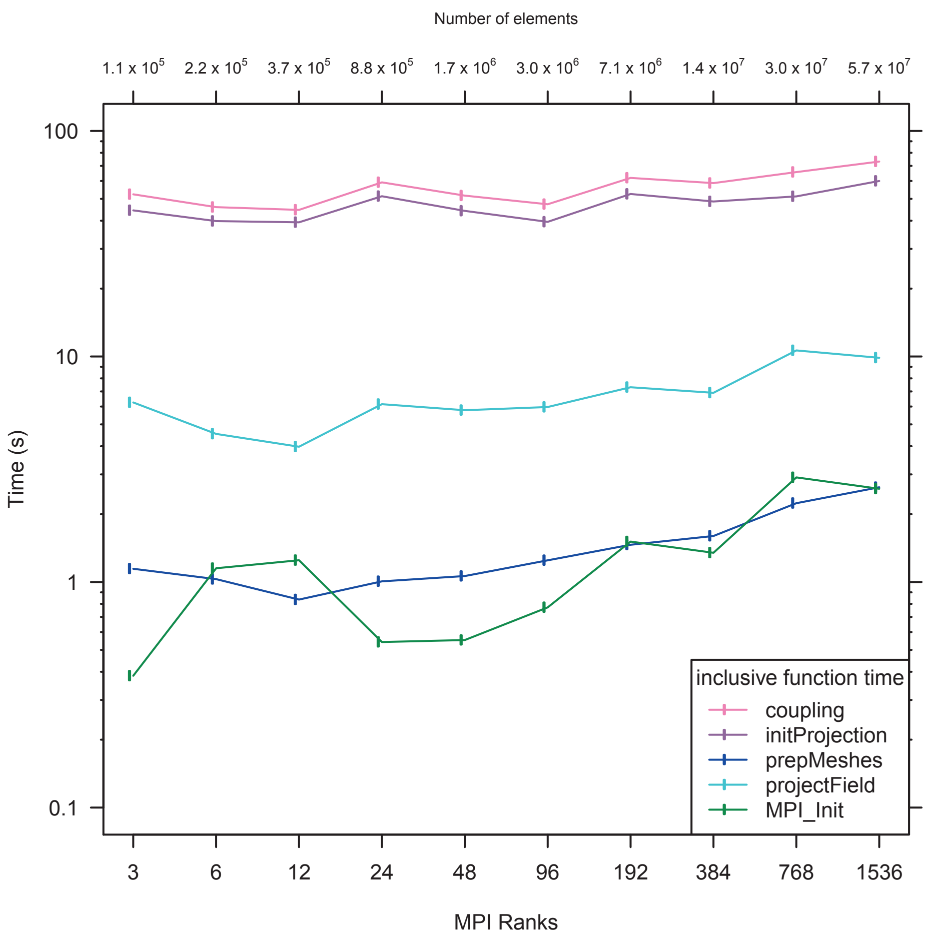

Figure 3: Weak scaling of CIAN2 coupling mini-app on Napier NextScale cluster using TAU profiler. Lower x-axis shows MPI Ranks, upper x-axis shows total number of elements.