Geosci. Model Dev. Discuss., 7, 7395–7425, 2014
www.geosci-model-dev-discuss.net/7/7395/2014/
doi:10.5194/gmdd-7-7395-2014

Geoscientific
Model Development
Discussions
Open Access

Efficient performance of the Met Office Unified Model v8.2 on Intel Xeon partially used nodes

I. Bermous

Centre for Australian Weather and Climate Research, the Australian Bureau of Meteorology, Melbourne, Australia

Received: 9 September 2014 – Accepted: 1 October 2014 – Published: 6 November 2014

Correspondence to: I. Bermous (i.bermous@bom.gov.au)

Published by Copernicus Publications on behalf of the European Geosciences Union.

Abstract

The atmospheric Unified Model (UM) developed at the UK Met Office is used for weather and climate prediction by forecast teams at a number of international meteorological centres and research institutes on a wide variety of hardware and software environments. Over its 25 year history the UM sources have been optimised for a better application performance on a number of HPC systems including NEC SX vector architecture systems and recently the IBM Power6/Power7 platforms. Understanding the influence of the compiler flags, MPI libraries and run configurations is crucial to achieving the shortest elapsed times for a UM application on any particular HPC system. These aspects are very important for applications that must run within operational time frames. Driving the current study is the HPC industry trend since 1980 for processor arithmetic performance to increase at a faster rate than memory bandwidth. This gap has been growing especially fast for multicore processors in the past 10 years and it can have significant implication for the performance and performance scaling of memory bandwidth intensive applications, such as the UM. Analysis of partially used nodes on Intel Xeon clusters is provided in this paper for short and medium range weather forecasting systems using global and limited-area configurations. It is shown that on the Intel Xeon based clusters the fastest elapsed times and the most efficient system usage can be achieved using partially committed nodes.

1 Introduction

The Unified Model (UM) numerical modelling system (Brown et al., 2012) is used for short and medium range weather forecasting at high resolutions as well as for climate modelling on relatively powerful HPC systems. Since September 2009 the UM has been used in the numerical weather prediction (NWP) component of the Australian Community Climate and Earth System Simulator (ACCESS; Puri et al., 2010) at the Australian Bureau of Meteorology (BoM). As a number of UM applications are being tested for the coming upgrade of the BoM operational systems, finding the most efficient run configurations becomes critical.

Current operational systems at BoM are based on the UM version 7.5 (vn7.5). At this stage it is planned that with the next operational systems upgrade, UM vn8.2 or a more recent software version will be used.

There have been substantial optimisations undertaken between these two versions. In particular, the model performance scalability has been significantly improved with the introduction of the asynchronous I/O feature from UM release vn7.8 (Selwood, 2012). The use of asynchronous computation and I/O in the model reduces the elapsed time of the model close to the NO-I/O case. Further optimizations are available under the recently released UM vn9.0; however these are not expected to significantly impact on the major results of this paper.

Leading HPC systems have from tens of thousands to several million very powerful cores. Since 1980 as per trend in the HPC development, the available processor performance has been increasing at a greater rate than the available memory bandwidth (Graham et al., 2005, pp.106–108). The authors concluded that a growing gap between processor and memory performance could become a serious constraint in performance scaling for memory bound applications. The gap between processor performance and memory bandwidth has been growing especially quickly for multicore processors in the past 10 years (Wellein et al., 2012). Limited memory bandwidth forces the cores on a node to compete for the same memory causing memory contention. For memory bandwidth intensive applications such as the UM, this can become a major problem.

Finding the most efficient usage of the system for a particular application, and the shortest elapsed times varies depending on whether the application is run on all node cores (fully committed case) or on a subset of the cores (partially committed case) available on each node. The placement of the threads and/or MPI processes across partially committed nodes (sockets) also needs to be done carefully taking into consideration all shared resources available to these cores.

Another practical aspect of the performance analysis discussed in the paper is to estimate whether the coming upgrade for the BoM's operational models will be feasible given the operational time windows and available HPC resources.

The performance analysis described here shows that on some modern systems the fastest and most efficient performance for a UM application can be achieved with the usage of partially committed nodes. The concept of using partial nodes for UM applications allows reduced memory contention and improves application performance (Bermous et al., 2013).

2 Description of the models

Regular upgrades to operational NWP systems are driven by the improvements made in both the NWP software and the underlying science. The BoM is currently planning for the next APS2 (Australian Parallel Suite 2) upgrade. The operational suite combines a number of short and medium range weather forecasting systems based on the UM software. These systems include the global NWP system (ACCESS-G), the regional NWP system (ACCESS-R, 12 km), the tropical-cyclone forecast system (ACCESS-TC) and several city forecast-only systems (ACCESS-C).

The current APS1 weather forecasting operational systems are based on UM vn7.5 for the global (Fraser, 2012) and vn7.6 for the city systems (BoM, 2013). At this stage it is planned that the operational weather forecasting software in APS2 will be upgraded to at least UM vn8.2. This software includes improvements to physical parameterisations, computational performance and performance scaling. Most of the scaling improvement is due to the introduction of asynchronous I/O. With this upgrade the model resolutions for the Global and City models will be increased. An increase in the model resolutions presents a challenge to fit the model runs into the required operational time windows. As a result an initial analysis of the performance measurements of the models is needed. In the current paper we will consider two types of the weather forecasting models: a medium range N512L70 Global model and a short range, limited area "City" model.

2.1 Global N512L70 model

The resolution of the currently used Global N320 (40 km) model with 70 vertical levels in APS1 will be upgraded to N512 (25 km) 70 levels in APS2. With a finite difference discretisation of the UM mathematical model the latest Global model has a horizontal grid of West–East × South–North of 1024 × 769. The existing "New Dynamics" dynamical core with semi-implicit and semi-Lagrangian time integration (Davies et al., 2005) was used.

The operational systems run 4 times daily include two different Global model runs for 3 and 10 days. In this paper performance scaling analysis for a 3 day simulation with a 10 min time step was used. With the operational model settings this system produces 137GB of output data. It is very important that the async I/O capability is used.

2.2 UKV city model

APS1 ACCESS-C operational systems nested in the ACCESS-R cover 5 Australian major city domains: Adelaide, Brisbane (South-East Queensland), Perth, Sydney and Victoria/Tasmania (Melbourne, Hobart). Each domain in the APS1 ACCESS-C has a horizontal resolution of approximately 4 km with 70 vertical levels. The corresponding short range city models are set to run 39 h forecasts 4 times daily. A significant horizontal resolution increase is planned for the APS2 upgrade by reducing the horizontal resolution from 4 km to either 1.5 km or 2.2 km. As a result some initial performance analysis for the city models is required to find the most efficient run configurations and arrangement within the operational run-time schedule.

In this paper an example of a short-range limited-area forecasting is taken from a 1.5 km high-resolution system for Sydney domain (Steinle et al., 2012). The experimental system was created in 2012 and it is currently running 24 times per day, with hourly cycling 3D-VAR. The range of the forecasts provided by the system varies between 12 and 39 h. The corresponding atmospheric model is nested within the N512L70 global model and based on the UM vn8.2 sources using its variable resolution version (the UKV) with the "New Dynamics" dynamical core.

The UKV modelling concept includes a high resolution of 1.5 km in the inner domain, a relatively coarse resolution of 4 km near the boundary of the main domain and a transition zone of using a variable grid size connecting the inner domain of 1.5 km with the "outer" domain of 4 km.

The Sydney UKV model had a horizontal grid of E–W × N–S of 648 × 720 with 70 vertical levels. The related forecast job was set to run a 25 h simulation with a time step of 50 s giving in total 1800 time steps for a run. The I/O in the job producing only 18 GB of the output data is relatively small in comparison to the size of I/O in the global model job. Therefore the usage of I/O servers does not have any major impact on the job performance, even when a large number of cores are utilised.

3 Description of HPC clusters and software used

This section includes hardware specifications and details of the software environment for the HPC clusters used.

3.1 Hardware: specifications of HPC clusters

Numerical results have been obtained on three HPC clusters with Intel® Xeon® processors. The first Constellation Cluster (Solar) with Intel Nehalem processors was installed by Oracle (Sun) in 2009. This system was upgraded by Oracle to a new cluster (Ngamai) with Sandy Bridge processors in 2013.

In addition to the Ngamai system, the BoM also has access to a Fujitsu system (Raijin) installed at the National Computational Infrastructure (NCI) at the Australian National University (ANU) in Canberra, and it is also based on Intel Sandy Bridge processors. The NCI system at 1.2 Pflops was the fastest HPC system in Australia in 2013. Technical characteristics of these three systems are provided in Table 1.

A node Byte/Flop value in the table was calculated as a ratio of the node maximum memory bandwidth and the node peak performance. It should be noted that this ratio has been reduced by over 2 times on the latest Ngamai and Raijin systems in comparison with Solar. All three clusters have Lustre file systems.

3.2 Software: compiler, MPI library

The HPC systems described in Table 1 support both Intel and GNU compilers; however, extensive testing has shown that the Intel compiler provides more advanced optimisation. With the UM vn8.2 sources separate executables were required for each model type: global and limited-area. The Intel compiler version 12.1.8.273 was used to produce UKV executables.

In order for the global N512L70 system to use the UM async I/O features the Intel 14.0.1.106 compiler release was needed to avoid crashes due to interaction between the older compiler and the new I/O code.

On the BoM HPC systems OpenMPI was the only library available. The Intel MPI library was available on Raijin; however, testing showed a 10–20 % degradation in performance in comparison with OpenMPI. For this reason and to maintain compatibility between the NCI and BoM systems, OpenMPI was used for all comparisons presented below.

The UKV executable was built with OpenMPI 1.6.5. The usage of a UM async I/O feature requires at least SERIALIZED thread safety level. Therefore the OpenMPI 1.7.4 library was needed to enable the async I/O feature of UM vn8.2.

3.3 Intel compiler options

The UM sources were compiled with the highest optimisation level of `-O3`. However, in order to maintain reproducibility between runs on a given platform, the `-fp-model` `precise` option (Corden et al., 2012) was also used. Reproducibility of results between Ngamai and Raijin is critical for our research and development process. This reproducibility along with compatibility of executables across these machines was achieved by removing `-xHost` compiler option used on Solar.

The compilation flag `-xavx` for advanced vector extension was included in the building procedure on Ngamai and Raijin. This flag can be used for Intel Xeon Processor E5 family. Also the model sources were compiled with `-g -traceback` options which in case of a run time crash problem provide useful information on a failed subroutine call sequence. The "`-g -traceback`" options had no impact on the application performance.

4 Description of the performance results

Jobs were submitted from login nodes to compute nodes, and job scheduling was done via Sun Grid Engine (SGE) software on Ngamai (earlier on Solar) and Portable Batch Systems Professional Edition (PBS Pro) job scheduler on Raijin.

All model runs were made on very busy multi-user systems using the standard normal queue and a shared Lustre file system for I/O with a potential impact of the I/O contention on the performance results. At the same time each model run used exclusive nodes without being effected by suspend and resume functionality on the systems.

Fluctuations in the elapsed times were usually around 3–5 %, but they were up to 50 % in 3–5 % of the runs. This was particularly noticeable on the Raijin system which had consistent utilisation above 95 %. Using the environment setting

```
OMPI_MCA_hwloc_base_mem_alloc_policy=local_only
```

greatly improved stability of the run times. The best run times were initially taken from 3 or 4 runs. If this initial estimate appeared to be an outlier from the estimated performance scaling curve, further runs were made. It is noteworthy that the fluctuations in elapsed times were much higher on all systems when more than 2000 cores were used. The cause of these large fluctuations was not investigated.

Choosing the best timing from a number of runs has been shown to provide reliable estimates of timings under operational conditions – use of the highest priority queue and dedicated file systems to avoid I/O contention and reserved and exclusive use of sufficient nodes to run operational systems. These arrangements result in variations in elapsed times of a few percent.

The MPI decomposition in the Unified Model is based on horizontal domain decomposition where each subdomain (MPI process) includes a full set of vertical levels. Within the computational core of the UM, OpenMP is generally used to parallelise loops over vertical dimension.

Lustre striping via

```
lfs setstripe -s 4M -c 8 <run_directory>
```

was also used to optimize I/O performance.

4.1 UKV performance results

The heritage of the UM beginning on massively parallel systems with no shared memory required a focus on a single level of parallelism using MPI. The use of hybrid parallelism in the UM is relatively recent and it still being improved. Furthermore the efficiency of pure MPI vs. the hybrid parallelism depends on the implementation, the nature of a given problem, the hardware components of the cluster, the network and the available software (compilers, libraries). As a result, there is no guarantee hybrid parallelism will improve performance for every model configuration.

Comparison of the best elapsed times produced by running the UKV model on Raijin and Ngamai is given in Figs. 1 and 2 correspondingly. On each figure the best elapsed times are provided for comparison between the efficiency using pure MPI vs. MPI/OpenMP hybrid parallelism. For simplicity the elapsed times are provided for 4 different decompositions starting from the usage of 384 cores with a stride of 384. The

Title Page
Abstract Introduction
Conclusions References
Tables Figures
◀I ▶I
◀ ▶
Back Close
Full Screen / Esc
Printer-friendly Version
Interactive Discussion

run decompositions were 16 × 24, 24 × 32, 32 × 36 and 32 × 48 with pure MPI usage. In case of the hybrid parallelism, two OpenMP threads were used and the related run configurations using the same number of cores as in the pure MPI case were 2 × 6 × 32, 2 × 12 × 32, 2 × 16 × 36 and 2 × 16 × 48, where the first value is the number of threads used. Figures 1 and 2 include results for two cases: fully and partially committed nodes. With partially committed nodes the application was running with the same decomposition as in the fully committed node case, but only a part of each node was used: 8 cores from 12 on Ngamai and 12 cores from 16 cores on Raijin.

With the use of partially committed nodes, the placement/binding of cores to the nodes/sockets should be done in a symmetrical way to give the same number of free cores on each socket. This allows for a better usage of the shared L3 cache on each socket.

Based on the performance results plotted in Fig. 2, the usage of pure MPI gives shorter elapsed times than with the usage of the hybrid parallelism for all decompositions on Ngamai. Figure 1 shows that a similar conclusion can be made for the elapsed times obtained on Raijin, excluding cases of fully committed nodes with the usage of over 768 cores. At the same time the fastest elapsed times on Raijin are achieved using pure MPI on partially committed nodes (12 cores-per-node) for the whole range of the used cores. The usage of more than 2 threads (3 or 4) showed no improvement in the elapsed times obtained with 2 threads.

Comparing the actual set of the obtained elapsed times between the two systems with the usage of pure MPI on fully committed nodes of (3252; 1733; 1290; 1058) s on Ngamai and (3052; 1677; 1209; 1048) s on Raijin shows that performance scaling on Raijin is slightly worse than on Ngamai. At the same time comparing the corresponding elapsed times of (2974; 1620; 1215; 1030) s on Ngamai and (2616; 1380; 1003; 812) s on Raijin with the usage of partially committed nodes, scaling improves on Raijin. The elapsed time with 1536 cores on Raijin is 27 % better than the corresponding elapsed time on Ngamai. This improvement reduces with the number of cores used, with only a 13.7 % faster time on 384 cores. Contributing factors to this include the Raijin cores being slightly faster than Ngamai cores and turbo boost was not enabled in BIOS on Ngamai. With the use of partially committed nodes, memory contention between the processes/threads running on the same node is reduced, improving performance for memory bandwidth-intensive applications. At the same time enabling turbo boost can increase processor performance substantially, reaching peak speeds of up to 3.1 GHz on Raijin using 12 cores-per-node (derived from information in Table 1).

Elapsed times for the UKV model with pure MPI usage on partially committed nodes on all 3 systems are provided in Figs. 3–8. Each pair of figures (Figs. 3 and 4 for Raijin; Figs. 5 and 6 for Solar and Figs. 7 and 8 for Ngamai) shows speedup as a function of the number of cores actually used as well as a function of the reserved cores (i.e. total number of cores allocated to the run, both used and unused).

For example, a 12 cores-per-node case on Raijin and a 6 cores-per-node case on Solar each reserved full nodes (16 and 8 cores respectively), but left a quarter of cores unused. This indicates a requirement of specifying by 1/3 of more cores using `-npersocket` or `-npernode` option of the mpirun command in comparison with the fully committed case using the same run configuration. For example, running the model with pure MPI on Raijin and using 12 cores per node the following options

```
mpirun -npersocket 6 -mca orte_num_sockets 2
                     -mca orte_num_cores 8 ...
```

were used in the mpirun command with OpenMPI 1.6.5. The last two options were required to specify to avoid the related bugs found in the OpenMPI software.

Figure 3 shows that the usage of partially committed nodes on Raijin. Even using 12 cores-per-node significantly improves the model scaling, and this improvement increases with the number of cores used. The improvement reaches 1.6 times the performance of using 1536 fully committed nodes. The usage of 8 cores-per-node on Raijin gives an additional performance improvement in comparison with the 12 cores-per-node case varying from 1.2 times at 96 cores to 1.1 times at 1728 cores. Examining the same performance results on the reserved cores basis as in Fig. 4 shows that above 576 cores it is more efficient to use 12 cores-per-node than 16. Note that the ratio of these two efficiency measures continues to increase with an increasing number of reserved cores. Even the usage of half nodes with 8 cores-per-node becomes more efficient in comparison with the fully committed node case for decompositions using over 1200 cores.

Figure 5 shows that the usage of partially committed nodes on Solar improves the runtimes with 6 cores-per-node by 6.9–16.5 % and a further reduction of 7.3–10.4 % is achieved with the usage of 4 cores-per-node.

The speedup curves as a function of used cores on Ngamai shown in Fig. 7 indicate that the model runs 10.4–14.6 % faster with the 8 cores-per-node. Unlike the other two systems (Raijin and Solar), the usage of half utilised nodes with 6 cores-per-node on Ngamai gives only a very modest reduction of no more than 5.5 %. These latter results indicate that a reduction in memory contention with the 6 cores-per-node case has almost no impact over using 8 cores-per-node.

Speedup curves as functions of the reserved cores for Solar (Fig. 6) and Ngamai (Fig. 8) show that unlike Raijin, the efficiency gains on partial nodes were not achieved on up to 1152 reserved cores on Solar and 1728 on Ngamai.

Using partially committed nodes can still be beneficial if there are restrictions within the numerical algorithms or software that limit the number of parallel processes. If this limit is below the level where the performance scaling curves have begun to level off, then partially committed cores will reduce the run times. The current UKV model resolution case actually satisfies this criterion. Due to semi-Lagrangian dynamics implementation, the halo size for each sub-domain limits the maximum number of subdomains in the North–South direction for an MPI decomposition to 48. So a 36 × 48 decomposition of 1728 cores represents near the maximum number of cores which can be used to achieve the best performance for the application. At the same time with the usage of partially committed nodes on Raijin an elapsed time of 1107 s obtained on fully committed nodes for a decomposition 36 × 48 can be improved by 37 % (696 s) on 2304 reserved cores using 12 cores-per-node or even by 43 % (632 s) on 3456 reserved cores, using 8 cores-per-node.

4.2 N512L70 performance results

As mentioned earlier, the UM async I/O feature (Selwood, 2012) was used to obtain good performance and especially performance scaling when running the model on more than 1000 cores. Using UM multithreaded I/O servers feature, all model runs were made with 2 OpenMP threads. The usage of 3 or even 4 threads showed no improvement in model performance.

Elapsed times from runs without I/O were used as a target for the async I/O case. The run times with full I/O in the fully committed node case were within 5–10 % of those without I/O. Note that on a very busy such as Raijin system some improvement in the runtimes for a few cases were achieved using different Lustre striping parameters, namely

```
lfs setstripe -s 8M -c <number_of_IO_servers>  <run_directory>
```

Performance results for the model were obtained with up to 3500 cores on Ngamai. Due to large variations in the run times with over 4500 cores on Raijin, the related model performance results are provided for only up to 3840 cores.

The best elapsed times obtained on Raijin are provided in Table 2, where the I/O servers configuration of ($m \times n$) included in the third column of the table have the following meaning: m is a number of the I/O server groups, n is a number of I/O tasks per server. For up to 2688 cores the best performance was achieved on fully committed nodes. For 3072 or more cores the best performance results were achieved using partially committed nodes.

Performance scaling of the model as a function of a number of the reserved cores for fully committed node case, 12 cores-per-node and 8 cores-per-node cases is shown in Fig. 9. The curves clearly show that the most efficient system usage with 3072 cores or higher is achieved running the application on partially committed nodes with 12 cores on each node from 16 available. The curves corresponding to 12 cores-per-node and 8 cores-per-node cases show a reasonably good scaling of the model with the usage of up to 4000 cores. Note that using partially committed nodes the model performance is slightly worse when core usage is in the range 384 to 2688.

The best elapsed times obtained on Ngamai are provided in Table 3. On this system in contrast with Raijin, the most efficient usage is achieved using fully committed nodes. Performance scaling of the model as a function of a number of the reserved cores for fully committed node case and with the usage of 8 cores-per-node case is provided on Fig. 10. For the fully committed node case a relatively good performance scaling is achieved with the usage of up to 1920 cores, after that performance scaling degrades slowly with the usage of 2304 and 2688 cores and levels out by 3072 cores. Based on the elapsed times produced with the usage of up to 2688 cores, the most efficient usage of the system is with fully committed nodes. At the same time the usage of 8 cores-per-node for up to 3456 reserved cores has relatively good performance scaling and from the efficiency point of view runs with 3072 reserved cores and higher should use partially committed nodes. Our expectations are that this efficiency in partial used nodes could be improved if turbo boost was enabled.

Performance results for a 6 cores-per-node case are not presented in Fig. 10. Having only 6 cores per node or 3 cores per socket to be used with the hybrid parallelism gives only two possible combinations for running the application: symmetrical case with 1 MPI process and 3 OpenMP threads running on each socket or a non-symmetrical case with 2 MPI processes running on one socket and 1 MPI process running on another socket with 2 threads per each process. As for the UKV case, the use of 3 threads per MPI process provides slower runtimes in comparison with 2 threads. With symmetrical usage of 3 threads per MPI process run on each socket, the model performance was even worse in comparison with the fully committed node case. At the same time the non-symmetrical usage with 3 MPI processes and 2 threads gave similar performance results as in the 8 cores-per-node case.

5 Conclusions

With a trend in the HPC industry of decreasing the Byte/Flop ratio on systems, the most efficient system usage by memory bandwidth intensive applications can be achieved with the usage of partially committed nodes. In other words, increasing memory bandwidth per core can more than compensate for the reduced number of cores in action. This approach can improve an application performance and most importantly the application performance scaling. A conclusion on whether a specific application should be running on fully or partially committed nodes depends on the application itself as well as on the memory bandwidth available per node. Other factors such as availability of turbo boost on the system and type of node interconnect can also influence the best choice. The usage of partially committed nodes can further reduce elapsed times for an application when the corresponding performance scaling curve has flattened.

Another example when the use of partially committed nodes can reduce run times is when the performance scaling has not flattened but the number of used processors cannot be increased due to other constraints in the application. This case was illustrated by the UKV model example in Sect. 4.1.

The approach of using partially committed nodes for memory bandwidth-bound applications can have a significant practical value for efficient HPC system usage. In addition, this can also ensure the lowest elapsed times for production runs of time-critical systems. This approach is a very quick method for providing major performance improvements. In contrast, achieving similar improvements through code related optimisation can be very time consuming and may not even be as productive.

The usage of partially committed nodes has been successfully employed to improve run times for other two tasks run on Raijin, including a coupled climate model running at a relatively low resolution of N96. Adding a couple of hundred cores gave a significant reduction in elapsed time of over 20 %. This level of improvement is very important for climate modelling experiments that usually require many months of elapsed time.

A second application was a UK Met Office four-dimensional variational analysis system at a resolution of N320L70. The usage of partially committed nodes gave up to a tenfold improvement in performance scaling for used core counts in the range between 500–1500.

Code availability

The Met Office Unified Model is available for use under licence. The Australian Bureau of Meteorology and CSIRO are licensed to use the UM in collaboration with the Met Office to undertake basic atmospheric process research, produce forecasts, develop the UM code and build and evaluate Earth System models. For further information on how to apply for a licence see http://www.metoffice.gov.uk/research/collaboration/um-collaboration.

The Supplement related to this article is available online at doi:10.5194/gmdd-7-7395-2014-supplement.

Acknowledgements. The author thank Yi Xiao (BoM) for providing the UM job settings used in the paper; Peter Steinle (BoM), Robert Bell (CSIRO), Paul Selwood (UK Met Office) and Gary Dietachmayer (BoM) for their useful comments; Jörg Henrichs (Oracle, Australia), Dale Roberts (NCI, ANU) and Ben Menadue (NCI, ANU) for continuous application support provided on all systems. A useful discussion with Dale Roberts on the Lustre striping parameter settings helped improve elapsed times for a few configurations where the original settings gave small anomalies (a few percent) in scaling results.

References

Bermous, I., Henrichs, J., and Naughton, M.: Application performance improvement by use of partial nodes to reduce memory contention, CAWCR Research Letters, 9, 19–22, 2013.

Brown, A. R., Milton, S., Cullen, M., Golding, B., Mitchell, J., and Shelly, A.: Unified modelling and prediction of weather and climate: a 25 year journey, B. Am. Meteorol. Soc., 93, 1865–1877, 2012.

Corden, M. and Kreitzer, D.: Consistency of Floating-Point Results using the Intel® Compiler or Why does not my application always give the same answer?, available at: https://software.intel.com/en-us/articles/consistency-of-floating-point-results-using-the-intel-compiler (last access: 3 September 2014), 2012.

Davies, T., Cullen, M. J. P., Malcolm, A. J., Mawson, M. H., Staniforth, A., White, A. A., and Wood, N.: A new dynamical core for the Met Office's global and regional modelling of the atmosphere, Q. J. Roy. Meteor. Soc., 131, 1759–1782, 2005.

Fraser, J.: NMOC Operations Bulletin Number 93, available at: http://www.bom.gov.au/australia/charts/bulletins/apob93.pdf (last access: 3 September 2014), 2012.

Graham, S. L., Snir, M., and Patterson, C. A.: Getting Up to Speed: the Future Of Supercomputing, National Academies Press, 289 pp., 2005.

Puri, K., Xiao, Y., Sun, X., Lee, J., Engel, C., Steinle, P., Le, T., Bermous, I., Logan, L., Bowen, R., Sun, Z., Naughton, M., Roff, G., Dietachmayer, G., Sulaiman, A., Dix, M., Rikus, L., Zhu, H., Barras, V., Sims, H., Tingwell, C., Harris, B., Glowacki, T., Chattopadhyay, M., Deschamps, L., and Le Marshall, J.: Preliminary results from numerical weather prediction implementation of ACCESS, CAWCR Research Letters, 5, 15–22, 2010.

Selwood, P.: The Met Ofice Unified Model I/O Server, ENES Workshop: Scalable IO in climate models, available at: https://verc.enes.org/computing/hpc-collaborations/parallel-i-o/workshop-scalable-io-in-climate-models/presentations/Unified_Model_IO_Server_Paul_Selwood.ppt/view (last access: 3 September 2014), 2012.

Staniforth, A. and Wood, N.: Aspects of the dynamical core of a nonhydrostatic, deep-atmosphere, unified weather and climate-prediction model, J. Comput. Phys., 227, 3445–3464, 2008.

Steinle, P., Xiao, Y., Rennie, S., and Wang, X.: SREP Overview: Meso-scale Modelling and Data Assimilation, available at: http://cawcr.gov.au/research/esm/Modelling_WS/10_MEW/Steinle_P.pdf (last access: 3 September 2014), 2012.

The Australian Bureau of Meteorology (BoM): NMOC Operations Bulletin Number 99, available at: http://www.bom.gov.au/australia/charts/bulletins/apob99.pdf (last access: 3 September 2014), 2013.

Wellein, G., Hager, G., and Kreutzer, M.: Programming Techniques for Supercomputers: Modern Processors, available at: https://wiki.engr.illinois.edu/download/attachments/217842128/performance-bandwidth.pdf?version=1&modificationDate=1359049396000 (last access: 3 September 2014), 2012.

Table 1. Intel Xeon Compute System Comparison.

	Solar, BoM	Ngamai, BoM	Raijin, NCI (ANU)
Processor type	Intel Xeon X5570	Intel Xeon E5-2640	Intel Xeon E5-2670
Number of compute nodes	576	576	3592
Total number of cores	4608	6912	57 472
InfiniBand interconnect	QDR	QDR	FDR
Memory size per node	24GB	64GB	32GB on 2395 nodes (67 %) 64GB on 1125 nodes (31 %) 128GB on 72 nodes (2 %)
Node processor cores	2 × (2.93 GHz, 4-core)	2 × (2.5 GHz, 6-core)	2 × (2.6 GHz, 8-core)
Max turbo frequency	3.333 GHz	3 GHz	3.3 GHz
Node cache size	2 × 8MB	2 × 15MB	2 × 20MB
Memory type	DDR3-1333 MHz	DDR3-1333 MHz	DDR3-1600 MHz
Number of memory channels	3	4	4
Node peak performance (base)	85 GFlops	240 GFlops	332.8 GFlops
Node max memory bandwidth	64 GB s^{-1}	85.3 GB s^{-1}	102.4 GB s^{-1}
Byte/Flop	0.753	0.355	0.308
Turbo boost additional multipliers	2/2/3/3	3/3/4/4/5/5	4/4/5/5/6/6/7/7
Usage of turbo boost	No	No	Yes
Usage of hyper threading	No	No	No

Table 2. The best elapsed times for N512L70 on Raijin using 2 threads.

Total reserved cores	Decomposition	I/O servers configuration	Cores used per node	Elapsed time (s)
384	8 × 23	2 × 4	16	2881
768	12 × 31	2 × 6	16	1575
1152	14 × 40	2 × 8	16	1131
1536	16 × 47	2 × 8	16	927
1920	16 × 59	2 × 8	16	802
2304	18 × 63	2 × 8	16	701
2688	20 × 66	2 × 12	16	640
3072	18 × 63	2 × 8	12	603
3584	20 × 66	2 × 12	12	543

Table 3. The best elapsed times for N512L70 on Ngamai using 2 threads.

Total reserved cores	Decomposition	I/O servers configuration	Elapsed time (s)
384	8 × 23	2 × 4	3068
768	12 × 31	2 × 6	1639
1152	14 × 40	2 × 8	1217
1536	16 × 47	2 × 8	1026
1920	18 × 52	2 × 12	877
2304	18 × 63	2 × 9	805
2688	20 × 66	2 × 12	759
3072	22 × 69	2 × 9	756

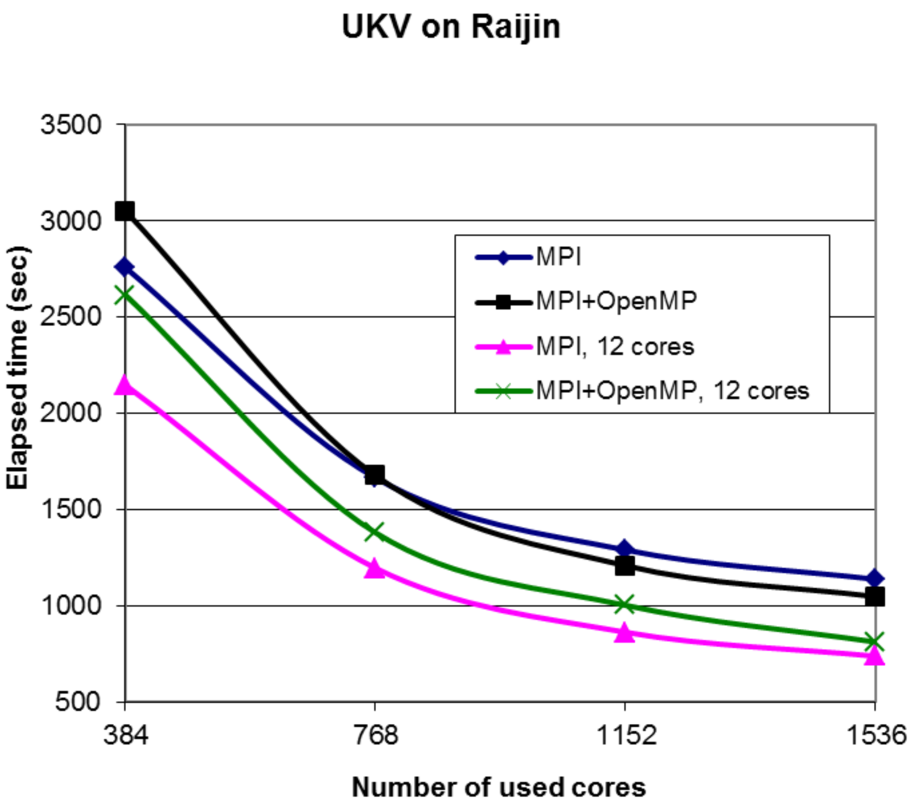

Figure 1. Elapsed times for the UKV model runs on Raijin vs. the number cores actually used in each run.

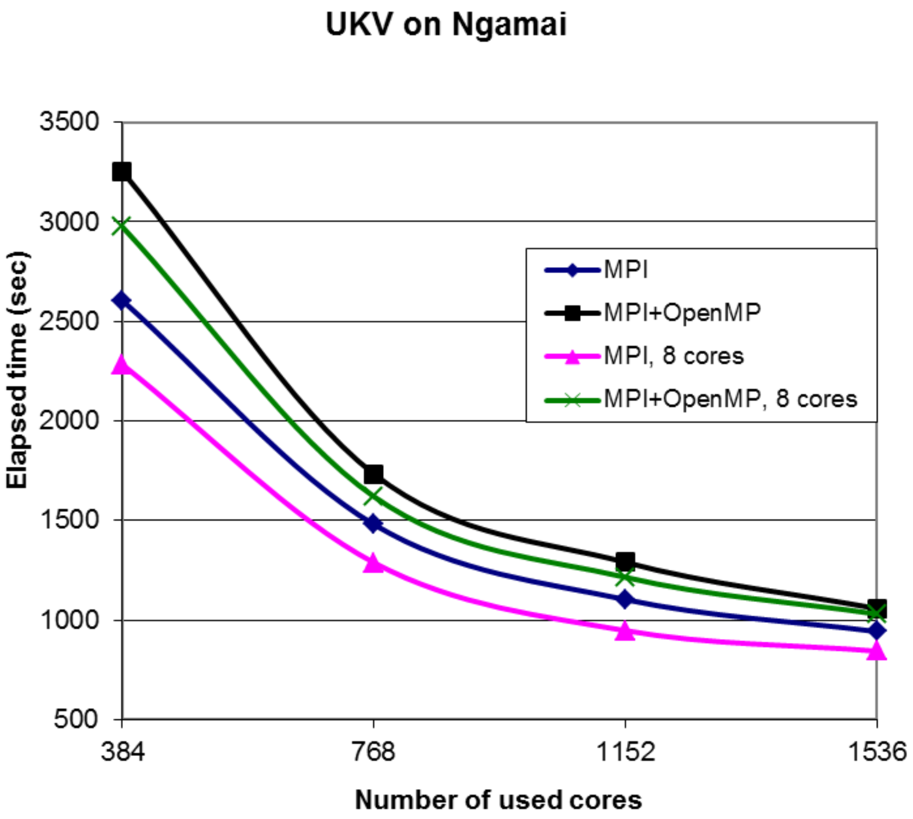

Figure 2. Elapsed times for the UKV model runs on Ngamai vs. the number cores actually used in each run.

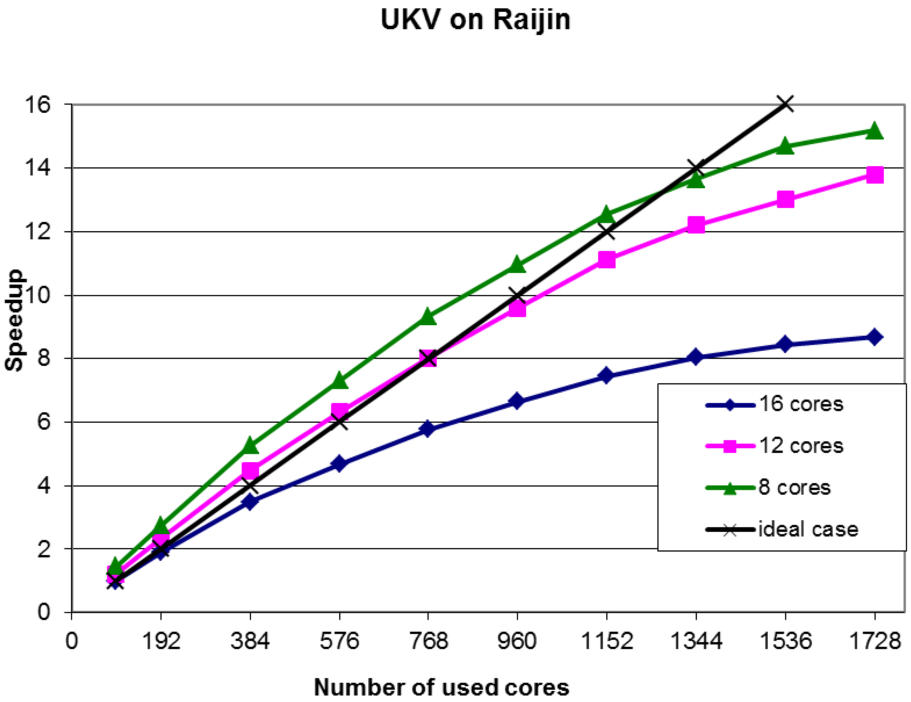

Figure 3. Speedup as a function of number of used cores on Raijin. Speedup was calculated in relation to the elapsed time of 9598 s obtained for a 96 core run on fully committed nodes.

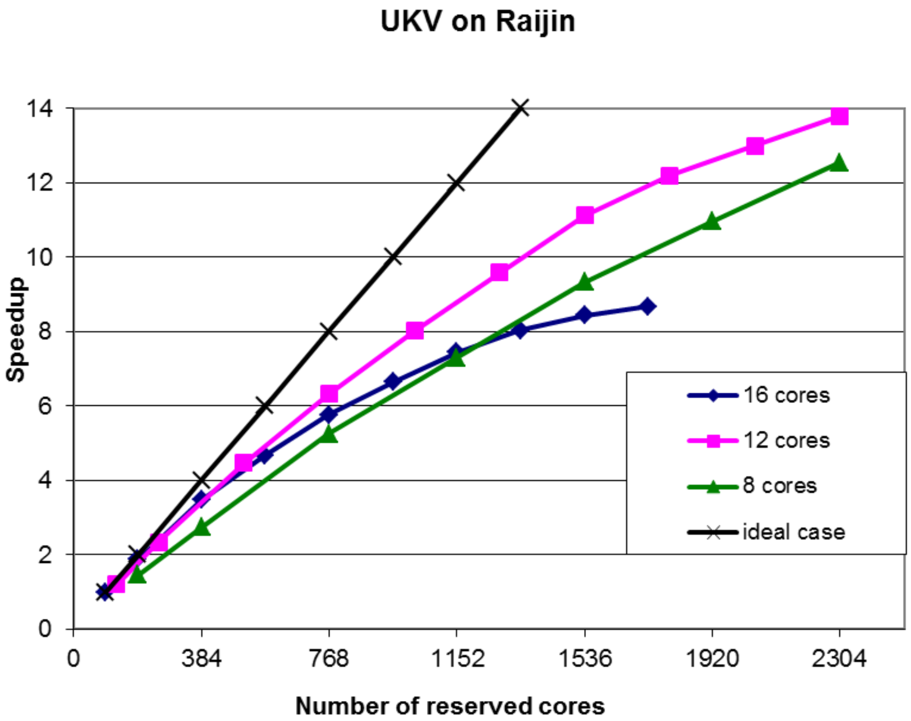

Figure 4. Speedup as a function of number of reserved cores on Raijin. Speedup was calculated in relation to the elapsed time of 9598 s obtained for a 96 core run on fully committed nodes.

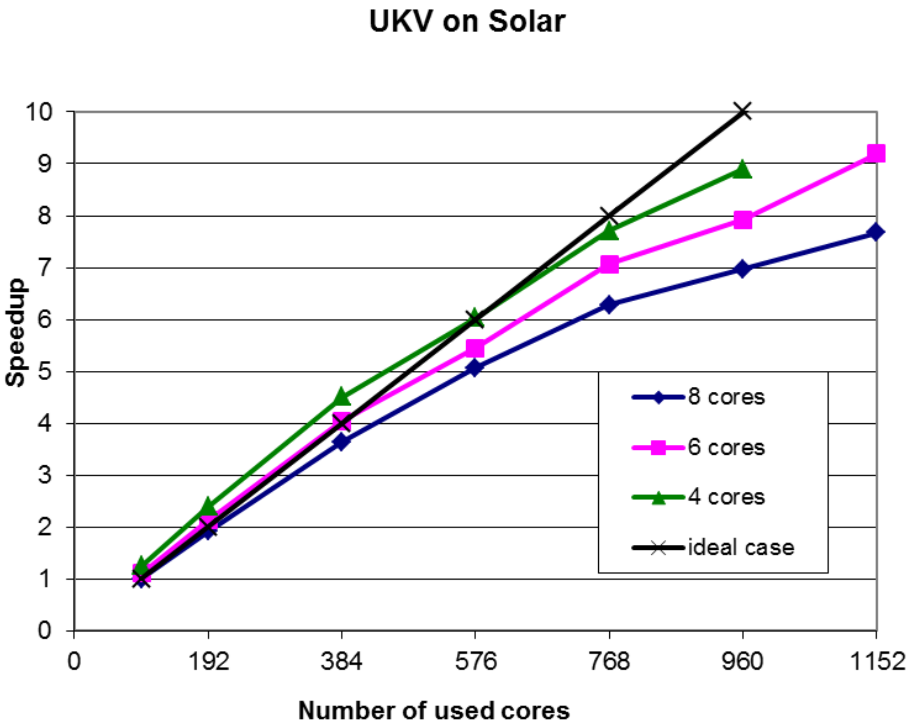

Figure 5. Speedup as a function of number of used cores on Solar. Speedup was calculated in relation to the elapsed time of 11 488 s obtained for a 96 core run on fully committed nodes.

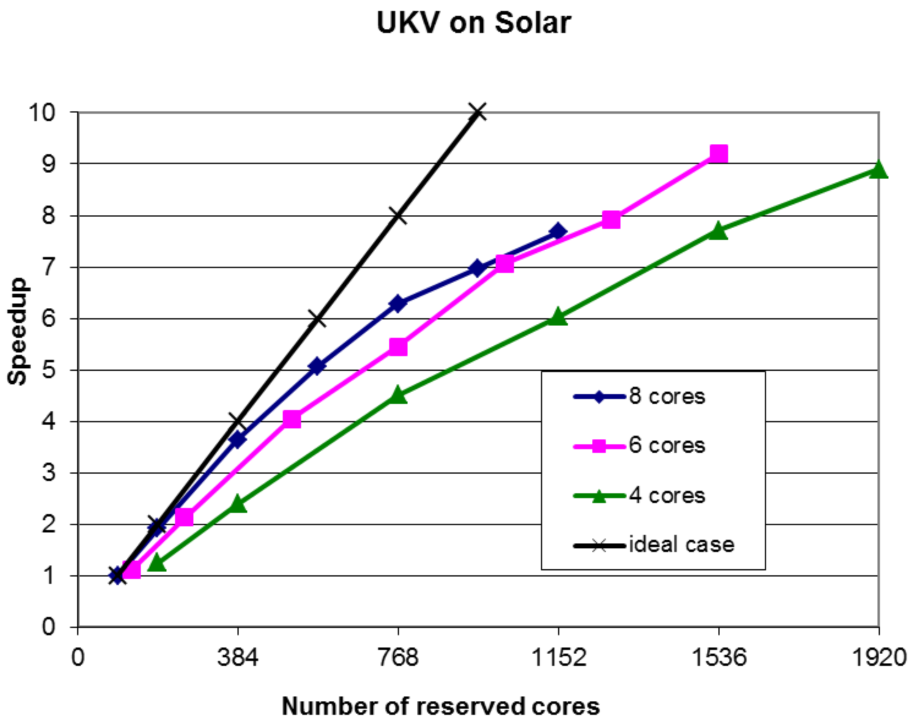

Figure 6. Speedup as a function of number of reserved cores on Solar. Speedup was calculated in relation to the elapsed time of 11 488 s obtained for a 96 core run on fully committed nodes.

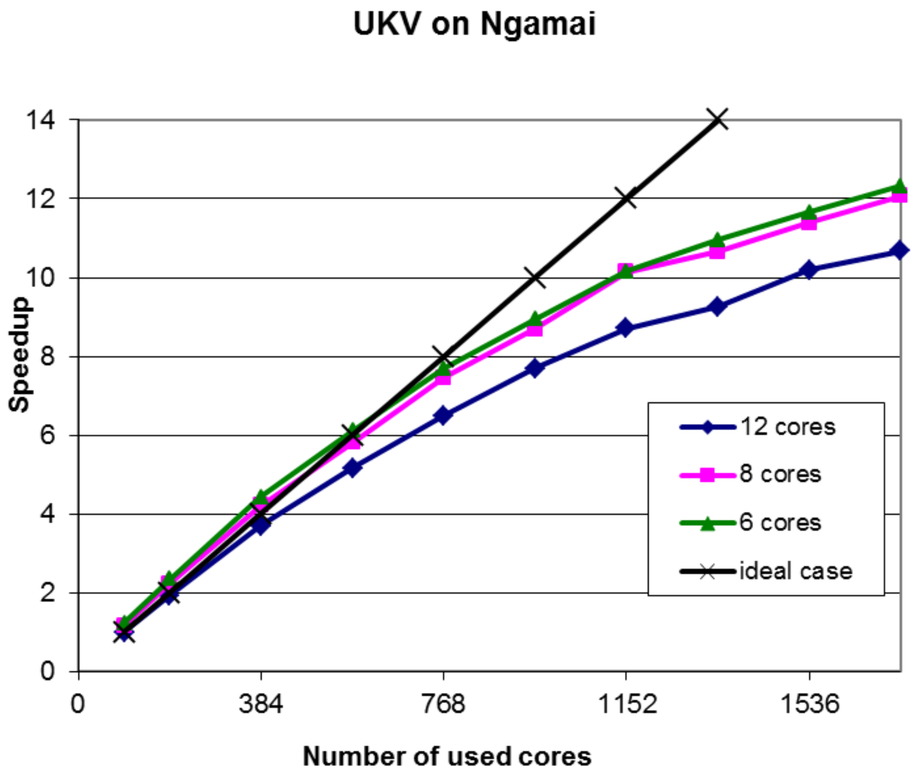

Figure 7. Speedup as a function of number of used cores on Ngamai. Speedup was calculated in relation to the elapsed time of 9608 s obtained for a 96 core run on fully committed nodes.

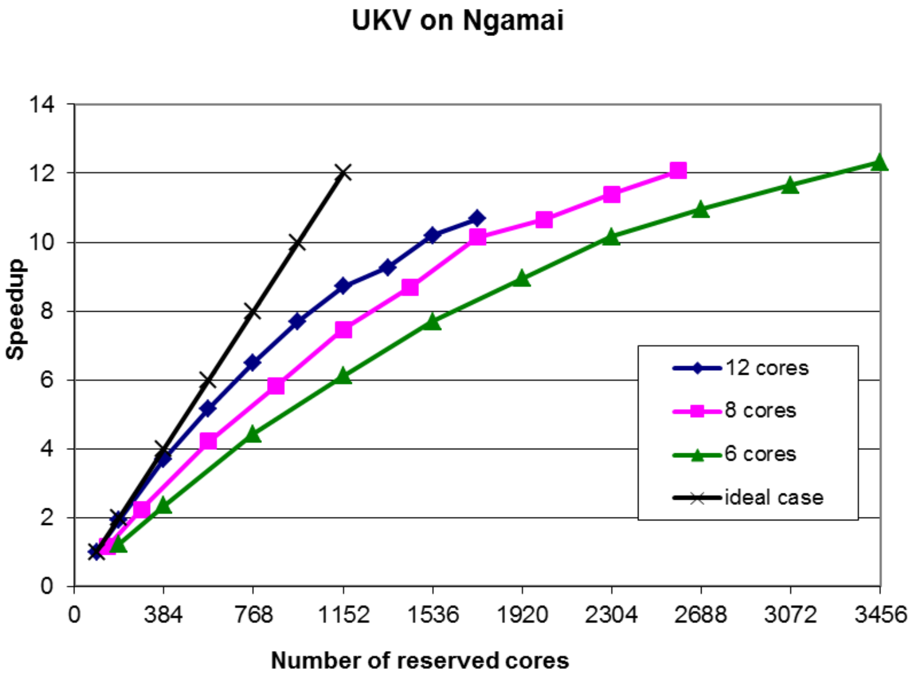

Figure 8. Speedup as a function of number of reserved cores on Ngamai. Speedup was calculated in relation to the elapsed time of 9608 s obtained for a 96 core run on fully committed nodes.

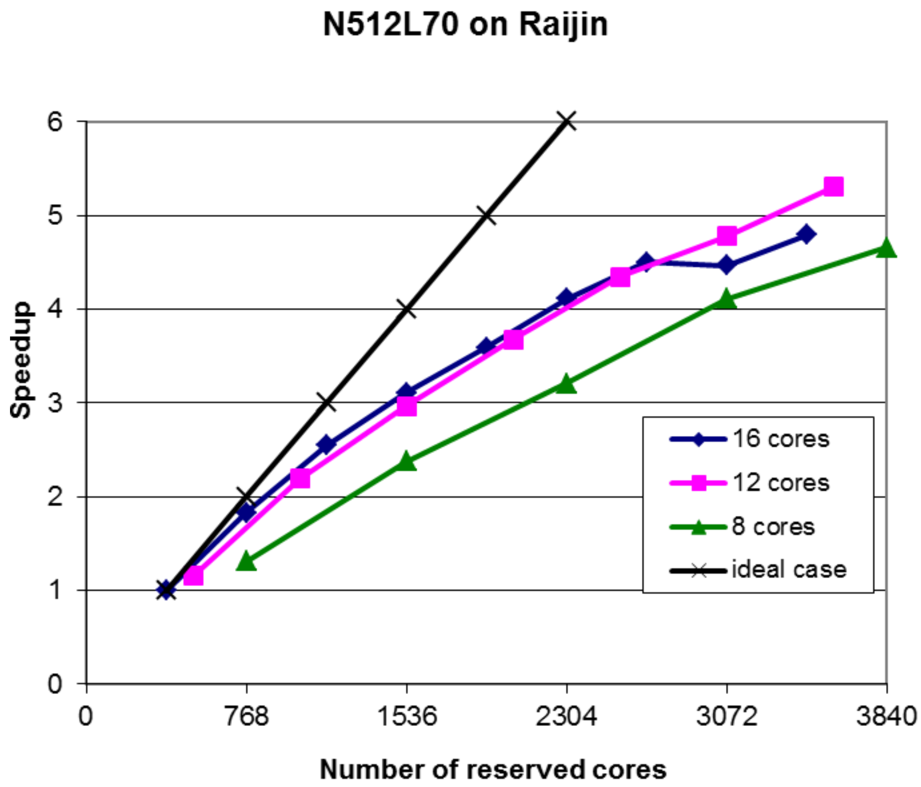

Figure 9. Speedup as a function of number of reserved cores on Raijin. Speedup was calculated in relation to the elapsed time of 2881 s obtained for a 384 core run on fully committed nodes.

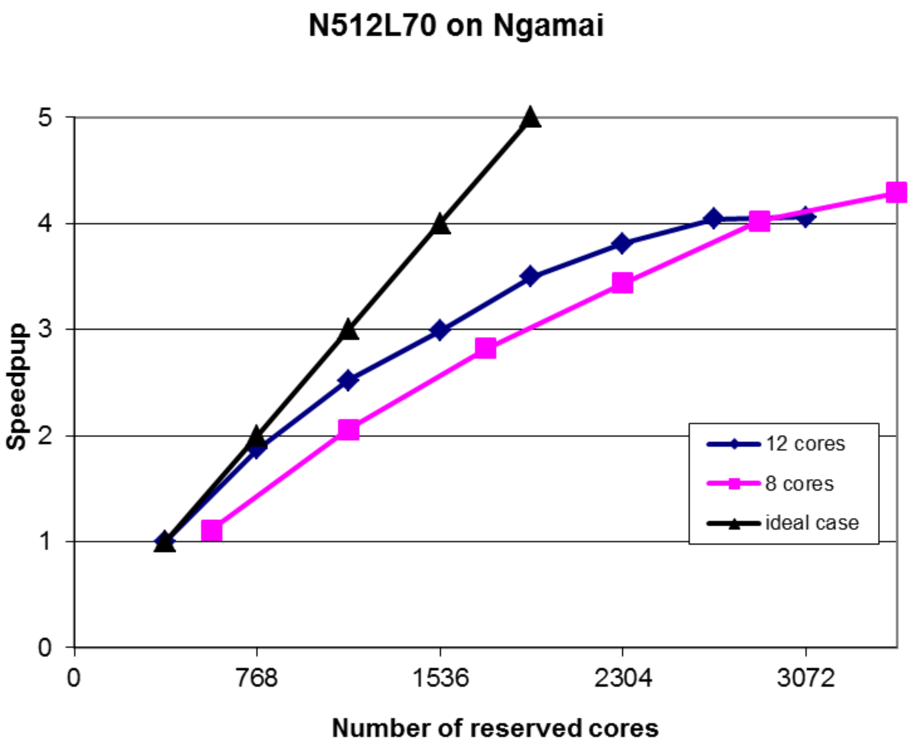

Figure 10. Speedup as a function of number of reserved cores on Ngamai. Speedup was calculated in relation to the elapsed time of 3068 s obtained for a 384 core run on fully committed nodes.