Accelerating the Global Nested Air Quality Prediction Modeling System (GNAQPMS) model on Intel Xeon Phi processors

Hui Wang[1,2,4,*], Huansheng Chen[2,*], Qizhong Wu[1,4], Junmin Lin[3], Xueshun Chen[2], Xinwei Xie[3], Rongrong Wang[1,4], Xiao Tang[2], Zifa Wang[2]

5 [1]College of Global Change and Earth System Science, Beijing Normal University, Beijing 100875, China
[2]LAPC, Institute of Atmospheric Physics, Chinese Academy of Sciences, Beijing 100875, China
[3]Intel Corporation
[4]Joint Centre for Global Changes Studies, Beijing Normal University, Beijing 100875, China
*These authors contributed equally to this work and should be considered co-first authors

10 Correspondence to: zifawang@mail.iap.ac.cn

Abstract. The GNAQPMS model is the global version of the Nested Air Quality Prediction Modelling System (NAQPMS), which is a multi-scale chemical transport model used for air quality forecast and atmospheric environmental research. In this study, we present our work of porting and optimizing the GNAQPMS model on the second generation Intel Xeon Phi processor codename "Knights Landing" (KNL). Compared with the first generation Xeon Phi coprocessor, KNL introduced many new hardware

15 features such as a bootable processor, high performance in-package memory and ISA compatibility with Intel Xeon processor. In particular, we described the five optimizations we applied to the key modules of GNAQPMS - CBM-Z gas chemistry, advection, convection and wet deposition. These optimizations work well on both the KNL 7250 processor as well as the Intel Xeon processor E5-2697 V4. They include: 1) updating the pure MPI parallel mode to hybrid parallel mode with MPI and OpenMP in emission, advection, convection and chemistry modules; 2) fully employ the 512-bit wide vector processing units (VPU) on the KNL platform;

20 3) reducing unnecessary memory access to improve caches efficiency; 4) reducing thread local storage (TLS) in CBM-Z gas phase chemistry module to improve its OpenMP performance; 5) changing global communication from interface-files writing/reading to using Message Passing Interface (MPI) functions to improve the performance and the parallel scalability. These optimizations improved GNAQPMS performance great. The same optimizations also work well for the Intel Xeon Broadwell processor, specifically, E5-2697v4. Compared with the baseline version of GNAQPMS, the optimized version is 3.34x faster on KNL and

25 2.39x faster on CPU. Furthermore, the optimized version on KNL runs at 26% lower average power compare to CPU. Combining the performance and energy improvement, the KNL platform is 47% more efficient compare to the CPU platform. The optimizations also enables much further parallel scalability on both the CPU cluster and KNL cluster – scale to 40 CPU nodes and 30 KNL nodes, with a parallel efficiency of 70.4% and 42.2%, respectively.

1 Introduction

30 Insatiable compute demand is driven by the ever increase in scientific demands found in many research codes such as the climate model Community Earth System Model (CESM) and the weather model Weather Research and Forecasting Model (WRF). In early days of computing, when there were not enough computation capability, scientists have to made trade-offs to fit the computation in limited budget. One example is the physical, chemical and dynamical processes in the models are simplified to adapt to the limited computation capability. Another example is the horizontal and vertical resolutions are also sacrificed due to

35 the limited computation capability. This means many details would be neglected or simplified in model, and the simulation ability of models is constrained.

Until early 2000, application performance can easily be increased by using higher frequency processors. As the semiconductor manufacturing technology improves, we reached the power-density and thermal limitation of silicon technology in early 2000 for single core processor design. The industry has taken a "right-hand turn" to deliver performance through more compute cores rather

5 than increasing processor frequency. As a result, applications need to embrace parallelism to achieve higher performance. And at the same time, heterogeneous computing was widely used in scientific computing area. The typical examples of many core architecture include Graphics Processing Unit (GPU) and Intel Many Integrated Core (Intel MIC) [Chrysos G, 2014].

With the popularity of new architecture, geo-scientific models have been partially or fully ported to the Graphics Processing Unit (GPU) and MIC heterogeneous computation platform to get better computation performance. There are many reports about porting

10 models to the GPU heterogeneous platform. The Princeton Ocean Model (POM) (Xu et al., 2015), except the initialization and input/output modules, was fully ported to the GPU by using CUDA-C and the model computation performance was improved both on single node and clusters. For the atmospheric chemistry models, the RADM2 chemical scheme in WRF-CHEM model (Grell et al., 2005) was ported to different multi-core platforms, and the GPU version got a speedup of 8.5x when compared with its serial version because of the limitation of the on-chip memory (Linford et al., 2009). Similar to the GPU, the first generation Intel Xeon

15 Phi coprocessor (codename Knights Corner or KNC) is connected to the mainboard via the Peripheral Component Interface Express (PCI-E) bus (Xu et al., 2015), and the bandwidth of PCI-E becomes the new performance bottleneck for some memory bandwidth bounded softwares, e.g., the popular atmospheric model WRF on KNC (Meadows, 2012). Mielikainen et al.(2014a,b,c,2015a,b) did a series of work to transplant the physical schemes to the KNC platform in WRF, including the Goddard microphysics scheme, the Thompson microphysics scheme, the Goddard shortwave radiation scheme, and the advection scheme in the model dynamic

20 core. Among these work, the Goddard microphysics scheme (Tao and Simpson, 1993; Khain et al., 2003) got 4.7x speedup on KNC and 2.8x speedup on CPU compared with its baseline version, on the same X86 architecture of CPU and MIC chips and sharing the same modern hardware features, respectively. In addition, this phenomenon of performance improvement also appeared in the optimization work of Thompson cloud microphysics. In our work, the global atmospheric chemistry model GNAQPMS also gets speedup on both CPU and MIC after optimization.

25 As emphasized by Mielikainen (Mielikainen et al., 2014), making full use of new hardware features of chips is the key to get the performance improvement on MICs. The Knights Landing (KNL) is the second-generation Intel MIC architecture processor (Sodani, 2015). Compared with the CPU, the KNL has more cores, 16GB on-chip Multi-Channel Dynamic Random Access Memory (MCDRAM), wider vector register and AVX-512 instructions support, and other minor architectural features. Compared with the GPU, the KNL is a bootable processor and can work alone without a host CPU and the bottleneck of the PCI-E bandwidth

30 is eliminated. In addition, the KNL adopts the x86 architecture and share the same programming model of the Intel processors. This study focuses on the optimization of the GNAQPMS model to fully utilize the features provided by modern (and future) processors. These optimizations not only improve performance of GNAQPMS on the KNL platform, but also they also work for our current and future generation processors. The optimization methods in this paper are also suitable for other atmospheric chemistry transport models, which use the similar chemistry or physical schemes to the GNAQPMS model.

35 In general, the optimization processes include three steps: 1) testing the baseline version codes and searching the performance bottleneck; 2) discovering and applying the optimization solutions according to the specific performance bottleneck; 3) testing the codes and validating the new version codes. The optimization process is iterative, that is, these steps would be repeated until the peak performance is reached. The single node performance should be optimized in prior to the multi-node optimization. The more details of common ways to modernize the codes can be found on Intel websites (https://software.intel.com/en-us/modern-

code/training/short-video-series). The organization of this paper is as follows: Section 2 introduces the GNAQPMS model and KNL processor. Section 3 presents the optimization processes for GNAQPMS. Section 3.1 shows the methods and tools to test the baseline codes and find the bottlenecks, followed by subsections describing the optimization measures in detail. The numerical experiments of performance testing are presented in Section 4, which include the result validations in Section 4.2 and the

5   performance tests in Section 4.2 and 4.3. The conclusions are given in Section 5.

2 Model and KNL description

The GNAQPMS model is a global multi-scale chemical transport model developed by the Institute of Atmospheric Physics, Chinese Academy of Sciences (Chen et al, 2015). The baseline version works on x86 CPU platform. As far as we know this is the first work that ports and optimizes GNAQPMS on the KNL platform. The model description of GNAQPMS and KNL is presented

10   as follows.

2.1 Model description of GNAQPMS

GNAQPMS is the global version of the Nested Air Quality Prediction Modelling System (Chen et al, 2015; Wang et al., 2006). Figure 1 shows the framework of the GNAQPMS model, its model inputs includes meteorology field and emissions, and its physical/chemical processes include dynamic emissions with profile assigned, advection, diffusion and convection due to

15   meteorology field, and gas chemistry, aerosol module, mercury chemistry and dry/wet deposition processes. The GNAQPMS model has several key techniques, including process analysis and tracer-tagging techniques, which will help to assess the contribution to emissions sources (Wu et al., 2011). It is also a multi-scale nested and parallel computation model, and can coupled with regional model to simulate the air pollution from global scale to regional scale, and even city-scale with MPI functions on the high performance parallel computation platform. The air pollutant concentration, depositions and source apportionment results

20   will be outputted after the simulation.

As mentioned above, the key chemical processes in GNAQPMS contain gas-phase chemistry, aqueous phase chemistry and aerosol chemistry. The gas-phase chemical module is the CBM-Z mechanism (Zaveri and Peters, 1999), with the solver module updated by Fan Feng (Feng et al., 2015) by using Modified-Backward-Euler (MBE) Method. In this study, the CBM-Z module is optimized heavily, as it is one of the most time consuming module in GNAQPMS as shown in Figure 2. Other chemical reaction modules

25   like aqueous phase and aerosol chemical module are relatively minor time consuming modules compared with CBM-Z module. The wet deposition module and aqueous phase chemical module use the RADM2 mechanism (Ge et al., 2014; Chang et al., 1987; Wang et al., 2002), and wet deposition is also a hotspot in GNAQPMS, which gets a really good performance after optimized. The other physical processes in GNAQPMS include dry deposition (Wesely, 2007), advection (Walcek and Aleksic, 1998; Walcek, 2000), diffusion and convection, and all of these modules are also important hotspots.

30   ### 2.2 KNL description

The targeted many-core processor we used in this study is the Intel Xeon Phi processor KNL 7250. Compared with the first generation MIC coprocessor KNC, KNL has many improvements. Similar to the GPU, KNC is a coprocessor and it can't work alone. KNC needs a host CPU and connected to the mainboard via PCI-E interface, and the bandwidth of PCI-E should be taken into account when designing the codes for KNC. Contrary to the KNC, KNL is able to work alone as a processor like the normal

35   CPU, which means more effective memory access. Moreover, KNL is equipped with a 16G MCDRAM, whose bandwidth is higher

than the normal DDR4 yet lower than the on-chip caches. The MCDRAM is designed to bridge the bandwidth gap between DDR4 and on-chip cache. MCDRAM on KNL can be configured in three modes for different application, including cache mode, flat mode and hybrid mode. Since the memory pressure of GNAQPMS is not dominant, the cache mode is chosen in our experiment. The core number and clock speed in KNL is also improved. The core number increases from 61 to 68, and the frequency of each

5   core increase from about 1.2 GHz to 1.4 GHz at the same time. More details about the KNL can be found in the homepage (http://www.intel.com/content/www/us/en/processors-/xeon/xeon-phi-detail.html).

3 Optimization technology

In this study, some optimization measures are used when porting the GNAQPMS to the KNL platform, including updating the pure MPI to hybrid parallel mode, strengthening vectorization, reducing unnecessary memory access, reducing TLS and changing

10   the way of global communication in the GNAQPMS model.

3.1 Baseline performance test

The first step of optimization was to test the baseline version of GNAQPMS (marked as "Base-V") and to identify the hotspots of the model. As shown in Figure 2, the run time breakdown of each section of the Base-V GNAQPMS was measured and calculated by the MPI function mpi_wtime in the experiment on the x86 CPU platform. The top five time consuming sections are CBM-Z

15   chemistry, diffusion, wet deposition, advection, and emission modules. In order to analyse the insight performance bottleneck of GNAQPMS, the Intel tool Vtune (https://software.intel.com/en-us/intel-vtune-amplifier-xe/) was used to investigate the hotspot functions in the model, and these hotspots are the targets to be optimized in priority. Hotspots are the segment codes that cost most of time during the model running. And optimizing these hotspot parts will be more efficient and helpful to improve the speed of the model codes.

20   To achieve the goal of porting the GNAQPMS model from the CPU platform to KNL platform, the basic idea is to fully use the hardware features of KNL, e.g. multi-hyper threads, vector computing units, MCDRAM and multi-level caches. Accordingly, the main optimization technologies include changing the parallel mode, vectorizing the codes and improving the caches hit rates. The Base-V GNAQPMS uses only MPI parallel mode which would ignore the hyper threads of CPU as well as KNL, and may greatly limit the scalability due to expensive communication as the number of processes increases. And at the same time, the file reading

25   and writing way to do the global communication is used in the Base-V GNAQPMS model, which directly affects the speed and limits the scalability.

3.2 Main optimization methods

According to the performance of the Base-V GNAQPMS on the CPU platform, the following optimization measures were conducted: 1) updating the pure MPI parallel mode to hybrid parallel mode with MPI and OpenMP functions; 2) manually

30   strengthening vectorization the model code to fully use the vector computation on KNL platform; 3) reducing unnecessary memory access to improve utilization efficiency of caches; 4) reducing TLS for common variables of each OpenMP thread; 5) changing the way of global communication from interface-files writing/reading to MPI functions.

As shown in Table 1, the optimization measures and the corresponding speedup for each hotspot section is presented, and the optimization steps in the heading of Table 1 refer to the optimization measures mentioned in the preceding paragraph. The OpenMP

35   was added to the sections including emission calculation, advection and convection, diffusion, gas phase chemistry and wet

deposition modules. Other sections are not involved in OpenMP optimization because of relatively low calculation density and time consumption. The cost of establishing and destroying threads in these sections are larger than the benefits gained from OpenMP. Therefore, OpenMP may lead to a decline of performance for these modules. To ensure the peak performance of OpenMP is fully achieved, the TLS of common variables in the CBM-Z module was removed, and this is effective to reduce the overhead

5   of establishing the threads by reducing the procedure to copy the common variables for each thread.

For global communications, the improvement was achieved by changing how communication is taken place. The original way for global communication is writing the messages that needed to broadcast to other processes into the files and processes need to read the file to get the message through I/O channel, which is a bottleneck in the model. The old way has relatively low efficiency and will impact the performance greatly, especially in the initialization module. Multiple processes read the same file will make this

10   file a critical section and the limitation of I/O bandwidth would also slow the speed. This problem is introduced by lacking consideration of parallel computation in the early development of the GNAQPMS model. Instead of the interface-files writing/reading, the new way is to use MPI_ALLREDUCE and MPI_GATHERV functions to perform global communications.

Manual vectorization is used in the sections including emission, advection, diffusion, and CBM-Z gas phase chemistry. KNL supports 512-bit vector operations and data path. It consists of two Vector Processing Units (VPUs) that can perform up to two

15   512-bit vector operations per cycle. Previous study about optimization of the physical schemes in WRF (Mielikainen et al., 2014) included plenty of work to vectorize the code and aligned the data for vectorization, which was prepared for the coming unified AVX-512 instruction on KNL and Skylake architecture CPUs. Although the compiler can automatically vectorize loops with no obvious data dependence, there are still many loops that can't be optimized automatically. As a result, manually vectorization directives are needed. During this process, different optimization tips were used for various scenes, and typical vectorization

20   techniques are introduced in Section 3.4 and 3.5. As mentioned by (Mielikainen et al., 2014), to ensure vectorization to get the peak performance, alignment directives were added in the codes. For KNL, if the data is aligned and padded to 64-byte boundaries, the efficiency of data access can be improved, and vector operation can be executed with high efficiency. This operation is treated as part of the vectorization optimization and is isolated as an independent optimization measure.

Memory optimization is also a critical spot that is concentrated on. As mentioned in Section 2.2, the MCDRAM on KNL platform

25   can be configured in three modes. Since the internal memory is not the bottleneck for GNAQPMS, 16GB MCDRAM is used as the last level cache for GNAQPMS. To utilize the two level caches and MCDRAM well, some unnecessary memory accesses were removed via optimization, and some temporary arrays were cut off. In the original code, some array variables are allocated, used, and de-allocated many times in the outermost loop of time step. In the optimized code, these variables are allocated and de-allocated only once outside the time step loop. Besides, reforming the loop order to realize vectorization, the cache hit rates also get improved

30   at the same time.

The optimization details for the typical physical and chemical modules, including initialization, emission, advection, convection, diffusion, chemistry and deposition modules in GNAQPMS model, are presented in the following sections.

3.2.1 Global Communication

The global communication of model parameter in Base-V GNAQPMS is realized thought writing and reading interface-files in

35   MPI parallel computing. The GNAQPMS does many global communications when the model was initialized for the defined model domains, grids and boundaries setting and such model parameter. Thus, the model initialization got a good speedup trough this optimization method, which can save the time consuming in input/output resources. According the performance experiment shown in Table.1, the speedup for this section reaches 1.31 on CPU and 1.1 on KNL compared with the Base-V model on CPU platform.

The KNL has more processor than the CPU, and will involved more MPI tasks, and more communication between each task, thus, the KNL get limited benefit through this measure on the single-node. However, this optimization method improved the model scalability greatly in the multi-node testing. The scalability test results are shown in Section 4.4.

3.2.2 Emission process section and typical vectorization

5   In the GNAQPMS model, the emission process section would read external emissions file, and assign them to emission variables, and increase the relevant pollution concentration when the model running. In a word, the emission process section prepares emission data for the GNAQPMS model. Therefore, it is the first section in the calculation loop of one time step. The emission section calculates and distributes the emission rates of the relevant species for each vertical and horizontal layer, and also completes the unit conversion.

10   In our study, the manual vectorization and multithreading were added in the emission section. The sample code of this work is shown in Figure 3. We changed the cyclic order of loops from **j**, **i**, **igas** to **igas, j, i ,** which ensured the data to be continuously accessed and improved the efficiency of caches. At the second step, we cancelled the calling of the subroutine get_ratio_emit() in the original code, and made it an internal function in main program, to improve the calling efficiency and facilitate the vectorization. Thirdly, vectorization was involved in emission section in the model by using parameters to convert scalar structure to vector

15   structure. At the end, we added the directives, clauses, declaration and syntax comment of OpenMP outside the outermost loop as shown in box "4". According to the performance testing of the sample code of this hotspot, it can get 8.57 speedup on CPU (E5-2697 V4) with 2 OpenMP threads. However, in the actual application, the number of OpenMP threads should fit the whole application to get the peak performance. This kind of optimization is common in the Opt-V GNAQPMS. With the double wider vector registers in KNL and OpenMP optimization, the speedup of the whole emission section reached 10.37.

20   Besides, the allocatable arrays loading the emission rates of all species were kept, which had been de-allocated at the end of the emission module in the Base-V GNAQPMS, since they would be used again in the gas phase chemistry section in the same way. Therefore, the cost of allocating, assigning and de-allocating these arrays for the second time in the section of CBM-Z is saved by preserving the variable across function. Finally, the initialization of these arrays was also updated from one statement to assign the whole four-dimension arrays to loops with OpenMP to initialize the values, improving the efficiency of initialization.

25   ### 3.2.3 CBM-Z gas phase chemistry section

The gas phase chemistry module is the key module in the chemical transport model, in GNAQPMS model, the gas phase chemistry module uses CBM-Z (Zaveri and Peters, 1999; Chen et al., 2015) scheme. According to the performance analysis with MPI timing function shown in Figure 2, the CBM-Z module is one of the most important and sophisticated hotspots in GNAQPMS.

The framework of the CBM-Z module is shown in Figure 4. It contains many complicated subroutines to calculate the gas phase

30   species concentration. The analysis of algorithms and code structure is in the first place before optimization, and the flow charts of the module is presented in Figure.3.Deep analysis with Intel Vtune tool shows that, the most complicated and important hotspot is the IntegrateChemistry, containing subroutines of SelectGasRegimes, PeroxyRateConstants, GasRateConstants, Setgasindices, MapGasSpecis and MBEsolver. The function of subroutine SelectGasRegime is to choose the optimum combination of gas-phase chemistry mechanisms based on the concentrations and emissions of different gas species. The selection of different gas-phase

35   chemistry mechanisms controls the following progresses in the subroutine IntegrateChemistry. The subroutines PeroxyConstants and GasRateConstants calculate the gas reaction rates for the selected chemistry mechanisms relying on the result of subroutine SelectGasRegimes. And then, the following subroutine SetGasIndices prepares the index of local concentration and emission

variables, and MapGasSpecies converts the global gas species concentration variables to the local concentration variables when it is called at the first time. After that, the MBE-solver would calculate the ODE functions of gas phase chemistry reactions. The second calling to MapGasSpecies returns the new values of the global gas concentrations.

According to the code structure of CBM-Z module, the optimization work was done step by step. At first, because of relatively simple structures and functions, the subroutine Setrunparameters and SolarzenithAngle, as shown in red in Figure 4, were removed and made into inline functions to improve the calling efficiency. For the subroutines in yellow in CBM-Z module, including the PrintResult and IntegrateChemistry subroutines, manual vectorization was conducted. The PrintResult subroutine has a function of converting the units of gas concentration from molecules/cc to ppb with one loop, and a directive pragma was added for this loop to force the compiler to do the vectorization.

In the CBM-Z module, the core calculation is in the IntegrateChemistry subroutine, whose flow chart is also shown in the right plot in Figure 4. The optimization of this subroutine contributed the most remarkable performance improvement for GNAQPMS. The main optimization of IntegrateChemistry includes two parts of work, manual vectorization and removing the Thread Local Storage. The manual vectorization in IntegrateChemistry was realized through three aspects, 1) giving the directives for the loops to instruct the compiler to vectorize the codes, including declaring no-dependency and aligning the data for efficient data accesses; 2) updating some code segments to let the serial codes to construct vectorization; 3) in the original code segments in IntegrateChemistry subroutine, the exponential operation sometimes was used without base-e, and these code segments had been updated to the base-e exponential operation, which can be vectorized by the AVX-512 on the KNL platform. The second part in our work was removing the TLS for OpenMP threads. The TLS is designed to keep the data synchronization among the threads for the common variables in Fortran. At present, the same work to the TLS has been done by the compiler automatically in the way of adding codes to copy the common variables for each thread of OpenMP. The codes added by the compiler impacted the performance greatly, and it is necessary to remove the TLS. For example, A type structure named **cbmztype** was constructed to store the common variables. And the subroutines contain these common variables were rewritten at the same time to add a formal parameter **cbmzobj(cbmztype)** to deliver these private variables to the subroutines to replace the common variables in the original codes. After these work, the common variables became the private variables for each threads of OpenMP. Performance evaluation showed that the efficiency was greatly improved in this way.

Other optimization, including removing local variables to improve the memory accesses, was also used in CBM-Z to improve the cache using efficiency. After all the optimizations, the CBM-Z got 2.56x speedup on the CPU platform, and 3.14x speedup on the KNL platform, respectively, as shown in Table.1. Compared with the other modules, the OpenMP performance of CBM-Z module is still worse, taking up most of the time in the Opt-V GNAQMS, as shown in Figure 2. This is because of the high cost of copying the rest of common variables. More optimizations will be involved in the future.

3.2.4 Diffusion and wet deposition section

In the Base-V GNAQPMS model, the costs of diffusion and wet deposition modules were significant (Figure 2), which spent 10 percentages and 8 percentages, respectively. After optimization, the cost percentages decreases to 9 percentages and 6 percentages in Opt-V GNAQPMS, respectively.

Manual vectorization and global communication updated has been used in the optimization of the diffusion module. According the performance on the single node, the diffusion module can get 1.78 speedup on the CPU platform and 2.39 speedup on the KNL platform. The optimization of wet deposition module is relatively simple but more effective. The main optimization of the module is adding OpenMP pragma to enable the multithreading for the wet deposition module. During this process, the position of

allocating the private variables should be carefully chosen. The scalability of threads in the wet deposition is really good, which allows the OpenMP to get better performance on the KNL platform than on the CPU platform. Finally, the optimized wet deposition module got 5.11X speedup on the KNL platform, much higher than 2.30X on the CPU platform.

4 Performance evaluation

5   A 48-hour global atmospheric chemistry simulation was designed as the test case to test the Opt-V GNAQPMS. In the test case, The GNAQPMS model has full physical and chemical processes in one domain without nesting grids, which is easier to diagnose the elapsed time. The horizontal resolution of the model is 1°×1°, which indicates that the modelling domain contains 360×180 grids. And the number of vertical layers is 20, while the time step for integration is 600 seconds in the test case. The test case was designed to test the performance of GNAQPMS on single node of CPU and KNL platform, and multi-nodes on different platform
10   clusters.

Three aspects were considered to test the performance of the optimized version of GNAQPMS by comparing with the baseline model version: 1) validation of the modelling results, 2) speedup, 3) scalability, as discussed in the following section. This test case only focused on the calculation loops part except for the output part.

4.1 Platform Setup

15   The Intel Corporation provides the High Performance Computing environment for the test. There are two platforms, includes CPU and KNL nodes. The CPU node has 2.3 GHz 18-core Intel Xeon processor E5-2697 V4 CPU, and each board contains two sockets, and its operating system is CentOS release 6.7, similar to the Red Hat Enterprise Linux system. The KNL node has Intel Xeon Phi processor 7250, 1.40GHz 68-core, and its operating system is Red Hat Enterprise Linux 7.2. The network is using the latest Intel Omni-Path Architecture (OPA). Both the Base-V and Opt-V GNAQPMS are compiled with the Intel FORTRAN Compiler 2017
20   Update 1, and the Opt-V GNAQPMS has been compiled on CPU and KNL platform, respectively. The compile flags for GNAQPMS are shown in Table 2. For the Opt-V GNAQPMS, the -xCore-AVX2 and –xMIC-AVX512 compile flags were not used for the advection module because these compile flags might cause calculation accuracy difference.

The comparison of Opt-V and Base-V GNAQPMS was not only tested on the single-node, but also on the CPU and KNL clusters, respectively.

4.2 Validation of Model Results

25   The spatial distribution of atmospheric chemistry was used for the validation, which is plotted from the binary files output by GNAQPMS model, as shown in Figure 4. The four species, including BC, CO, O_3, NO_2, were chosen to verify the model results by examining their value changes after optimization. According to the different reaction properties, these four species participate in different chemistry reactions. Black carbon (BC) is a component of fine particulate matter (PM2.5), consisting of pure carbon
30   in several linked forms, and is emitted in anthropogenic and naturally occurring soot. In GNAQPMS model, the BC hardly gets involved in chemical reactions, and can stay in the atmosphere for several days or even weeks. Carbon monoxide (CO) is spatially variable and short lived, playing a role in the formation of ground-level ozone (O_3), and its spatial distribution is predominated by the emissions. Nitrogen dioxide (NO_2) is one of ground-level ozone (O_3) precursor, participating in the photochemical reaction with ozone (O_3). Thus, CO, NO_2 and O_3 will be calculated in the gas phase module CBM-Z of GNAQPMS. Because of this kind
35   of species diversity, the model modules can be fully covered and tested, to ensure that the model results have no change with the

step-by-step optimization. By comparing the model output results and plotting the spatial distribution images shown in Figure 5, the results between Base-V and Opt-V GNAQPMS were confirmed to be identical. The optimization does not introduce the "erroneous" concentration for any atmospheric specie, and therefore it is reliable.

4.3 Speedup Performance

5 The run time breakdown of Base-V and Opt-V GNAQPMS on the single-node CPU platform is shown in Figure 2. Due to the different speedups of main modules (Table 1), the ranks of most time consuming modules have changed after optimization. The ranked third has changed from "wet deposition" to "advection and convection" module, while the percentages of the diffusion, wet deposition and emission modules decreased. In both Base-V and Opt-V GNAQPMS, the CBM-Z module played the most significant role for performance, and the absolute performance improvement for CBM-Z was remarkable after optimization. The

10 better vector processing performance helped the KNL to get a better speedup (3.14) than the CPU (2.56) in CBM-Z. However, compared with the other modules (e.g. emission and wet deposition), the acceleration of CBM-Z is limited, this is mainly caused by the parallelization overhead of OpenMP for CBM-Z module when establishing OpenMP threads. The total performance on the single node is showed in Figure 6. The speedup of Opt-V GNAQPMS reached 3.34X on the KNL and 2.39X on the CPU, compared to the Base-V GNAQPMS, and the KNL platform has an advantageous speedup of 1.39X over the CPU platform. At the same

15 time, the average power is 440W and 324W for CPU and KNL platform, respectively. Therefore, the average power of KNL is 26% lower and the average energy consumption is 47% lower than that of the CPU platform. The faster speed and lower energy consumption make KNL outperform CPU on single node.

4.4 Scalability on Cluster

20 The cluster performance of the atmospheric model was measured by strong scalability. Strong scalability means how many computing resources can be used when the computing scale is fixed, which can be measured by the speedup with increasing node number. The better scalability means the model can use the more computing resources to deal with the task, and complete the task in a shorter time. The scalability was measured by recording speedup of the core calculation portion of the model on clusters. As showed in Figure 7, the Base-V GNAQPMS can maximally use eight two-socket CPU nodes for the test case. After optimization,

25 the parallel scalability of GNAQPMS is greatly improved on both the CPU and KNL cluster, scaling to 40 CPU nodes and 30 KNL nodes, with a parallel efficiency of 70.4% and 42.2%, respectively The Opt-V GNAQPMS can use more than 40 two-socket CPU nodes for the same test case. The test of scalability on CPU cluster is constrained by limited computing resources, and the scalability could be expected to extend to more than 40 nodes. However, the scalability curve of KNL cluster is lower than the one of CPU as shown in Figure 7, that the Opt-V GNAQPMS can use 30 KNL nodes at most. On single node, the Opt-V GNAQPMS model

30 on KNL platform has higher performance than that on CPU platform, when the number of nodes reaches to 12, the speed of the Opt-V GNAQPMS on KNL platform is lower than that on CPU platform. This is mainly caused by too many MPI processors and not good enough performance of GNAQPMS OpenMP code segments on KNL, which has 68-core and more than four times the CPU. According to the above test, the further optimization of OpenMP is needed to improve the cluster performance of Opt-V GNAQPMS on KNL cluster.

5 Conclusions

In this study, the global chemistry transport model GNAQPMS is optimized to run on the Intel second-generation MIC architecture processor KNL and get the acceleration. The main optimization methods and tips were used including 1) updating the pure MPI parallel mode to hybrid parallel mode with MPI and OpenMP; 2) manually vectoring the codes in GNAQPMS model to make full use of the 512-bit wide VPU on the KNL platform; 3) reducing unnecessary memory access to improve utilization efficiency of caches; 4) removing TLS for common variables with each OpenMP thread to improve the OpenMP efficiency; 5) changing the way of global communication from interface-files writing and reading to MPI functions.

The tests of Opt-V GNAQPMS were conducted on the latest Xeon E5-2697 V4 and KNL 7250 clusters. Both single node and multi-node cluster performances were tested. On single node, the Opt-V GNAQPMS got a speedup of 2.39 on the CPU platform and 3.34 speedup on the KNL platform compared with the Base-V model on the CPU platform. The power and energy consumption of KNL is 26% and 47% lower when compared with CPU respectively. Compared with the CPU platform, the KNL has obvious advantage with fast speed and lower energy consumption. The cluster test results showed that the scalability of GNAQPMS is largely increased from 8 nodes to up to 40 nodes on the CPU platform, but the scalability on the KNL cluster is not as good as the CPU platform due to the bottleneck of MPI global communication and fragmental OpenMP parallel regions. Therefore, further work will be focused on merging OpenMP parallel regions and optimize global communication. Besides, the I/O optimization was not considered in this study and it should be taken into account in the future.

The general suggestions we could give for the optimization of other models on KNL are as follows: 1) focusing on vectorization of codes and 2) the performance of OpenMP is very important for KNL, which needs the coder to design more efficient parallel regions for OpenMP.

Code availability

The source code of Base-V and Opt-V GNAQPMS are available online via ZENODO (https://doi.org/10.5281/zenodo.290203) at doi:10.5281/zenodo.290203.

Acknowledgements:

The National Natural Science Foundation of China (41305121 and 41405119), the National Major Research High Performance Computing Program of China (2016YFB0200800), and the Environmental Public Welfare Research Project of China (201509014) fund this work. The authors would like to thank the Intel Corporation and Software Support Group (SSG) group for providing the High Performance Computing (HPC) environment and technical support for the case testing of GNAQMS model. Special thanks should go to the Victor Lee for providing the professional English writing instruction.

References

Chang, J. S., Brost, R. A., Isaksen, I. S. A., Madronich, S., Middleton, P., Stockwell, W. R., and Walcek, C. J.: A three‐dimensional Eulerian acid deposition model: Physical concepts and formulation, J. Geophys. Res.-Atmos, 92, 14681-14700, doi:10.1029/JD092Id12p14681, 1987.

Chen, H. S., Wang, Z. F., Li, J., Tang, X., Ge, B. Z., Wu, X. L., Wild, O., and Carmichael, G. R.: GNAQPMS-Hg v1.0, a global nested atmospheric mercury transport model: model description, evaluation and application to trans-boundary transport of Chinese anthropogenic emissions, Geosci. Model. Dev., 8, 2857-2876, doi:10.5194/gmd-8-2857-2015, 2015.

Chrysos, G., and Engineer, S. P.: Intel xeon phi coprocessor (codename knights corner), Proceedings of the 24th Hot Chips Symposium,2012.

Feng, F., Wang, Z., Li, J., and Carmichael, G. R.: A nonnegativity preserved efficient algorithm for atmospheric chemical kinetic equations, Appl. Math. Comput., 271, 519-531, doi:10.1016/j.amc.2015.09.033, 2015.

Ge, B. Z., Wang, Z. F., Xu, X. B., Wu, J. B., Yu, X. L., and Li, J.: Wet deposition of acidifying substances in different regions of China and the rest of East Asia: Modeling with updated NAQPMS, Environ. Pollut., 187, 10-21, doi:10.1016/j.envpol.2013.12.014, 2014.

Grell, G. A., Peckham, S. E., Schmitz, R., McKeen, S. A., Frost, G., Skamarock, W. C., and Eder, B.: Fully coupled "online" chemistry within the WRF model, Atmos. Environ., 39, 6957-6975, doi:10.1016/j.atmosenv.2005.04.027, 2005.

Khain, A., Lang, S., Lynn, B., and Shie, C. L.: Microphysics, radiation and surface processes in the Goddard Cumulus Ensemble (GCE) model, Meteor. Atmos. Phy., 82, 97-137, doi:10.1007/s00703-001-0594-7, 2003.

Linford, J. C., Michalakes, J., Vachharajani, M., and Sandu, A.: Multi-core acceleration of chemical kinetics for simulation and prediction, in: Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis/ACM, Portland, Oregon, 14-20 November 2009, 1-11, 2009.

Meadows, L.: Experiments with WRF on Intel® Many Integrated Core (Intel MIC) Architecture, in: 8th International Workshop on OpenMP/Openmp in A Heterogeneous World, Rome, Italy, 11-13 June 2012, 130-139, 2012.

Mielikainen, J., Huang, B., and Huang, A. H. L.: Intel Xeon Phi accelerated Weather Research and Forecasting (WRF) Goddard microphysics scheme, Geoscientific Model Development Discussions, 7, 8941-8973, doi:10.5194/gmdd-7-8941-2014, 2014a.

Mielikainen, J., Huang, B., and Huang, A. H. L.: Optimizing the updated Goddard shortwave radiation Weather Research and Forecasting (WRF) scheme for Intel Many Integrated Core (MIC) architecture, SPIE Sensing Technology+ Applications/International Society for Optics and Photonics, Maryland, United States, 20 April 2015, 9501, 2015a.

Mielikainen, J., Huang, B., and Huang, A. H. L.: Optimizing zonal advection of the Advanced Research WRF (ARW) dynamics for Intel MIC, SPIE Remote Sensing, International Society for Optics and Photonics, Amsterdam, Netherlands, 22 September 2014, 92470M, 2014b.

Mielikainen, J., Huang, B., and Huang, A. H. L.: Optimizing Weather and Research Forecast (WRF) Thompson cloud microphysics on Intel Many Integrated Core (MIC), SPIE Remote Sensing, International Society for Optics and Photonics, Maryland, USA, 5 May 2014, 91240Q, 2014c.

Mielikainen, J., Huang, B., and Huang, A. H. L.: Revisiting Intel Xeon Phi optimization of Thompson cloud microphysics scheme in Weather Research and Forecasting (WRF) model, SPIE Remote Sensing, International Society for Optics and Photonics, Toulouse, France, 21 September 2015, 964605-964605-14, 2015b.

Sodani, A.: Knights landing (KNL): 2nd Generation Intel® Xeon Phi processor, Hot Chips 27 Symposium (HCS), 2015 IEEE, 22-25 August 2015, 1-24, 2015.

Tao, W. K., and Simpson, J.: Goddard Cumulus Ensemble Model. Part Ⅰ: Model Description, Terr. Atmos. Oceanic Sci, 4, 1993.

Walcek, C. J.: Minor flux adjustment near mixing ratio extremes for simplified yet highly accurate monotonic calculation of tracer advection, J. Geophys. Res.-Atmos, 105, 9335-9348, doi:10.1029/1999JD901142, 2000.

Walcek, C. J., and Aleksic, N. M.: A simple but accurate mass conservative, peak-preserving, mixing ratio bounded advection algorithm with FORTRAN code, Atmos. Environ., 32, 3863-3880, doi:10.1016/S1352-2310(98)00099-5, 1998.

Wang, Z., Xie, F., Wang, X., An, J., and Zhu, J.: Development and application of Nested Air Quality Prediction Modeling System (in Chinese), Chinese Journal of Atmospheric Sciences, 30(5), 778-790, 2006.

5 Wang, Z., Akimoto, H., and Uno, I.: Neutralization of soil aerosol and its impact on the distribution of acid rain over east Asia: Observations and model results, J. Geophys. Res.-Atmos, 0345, 1-6, doi: 10.1029/2001JD001040, 2002.

Wesely, M. L.: Parameterization of surface resistances to gaseous dry deposition in regional-scale numerical models ☆, Atmos. Environ., 23, 1293-1304, doi:10.1016/0004-6981(89)90153-4,2007.

Wu, Q., Wang, Z. F., Gbaguidi, A., Gao, C., Li, L. N., and Wang, W.: A numerical study of contributions to air pollution in Beijing
10 during CAREBeijing-2006, Atmos. Chem. Phys., 11, 5997–6011, doi:10.5194/acp-11-5997-2011, 2011

Xu, S., Huang, X., Oey, L. Y., Xu, F., Fu, H., Zhang, Y., and Yang, G.: POM.gpu-v1.0: a GPU-based Princeton Ocean Model, Geosci. Model. Dev., 8, 2815-2827, doi:10.5194/gmd-8-2815-2015, 2015.

Zaveri, R. A., and Peters, L. K.: Anew lumped structure photochemical mechanism for long-scale applications, J. Geophys. Res.-Atmos, doi:10.1029/1999JD900876, 1999.

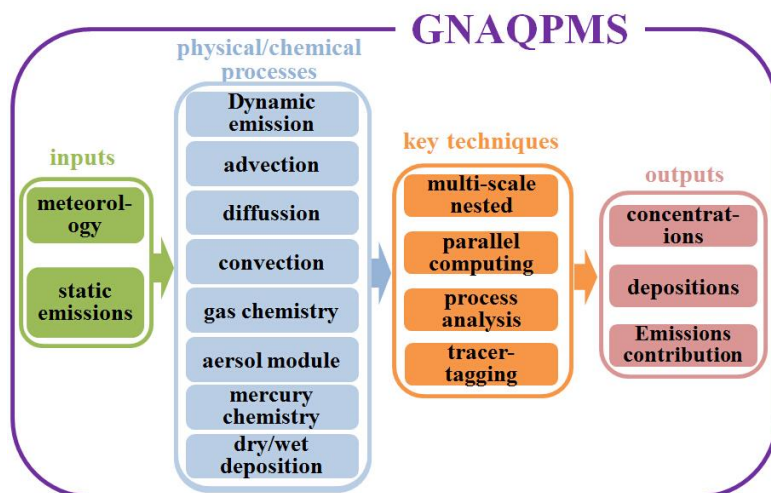



Figure 1. The framework of the Global Nested Air Quality Prediction Modeling System (GNAQPMS) model

Inner ring -- Opt-V GNAQPMS
Outer ring -- Base-V GNAQPMS

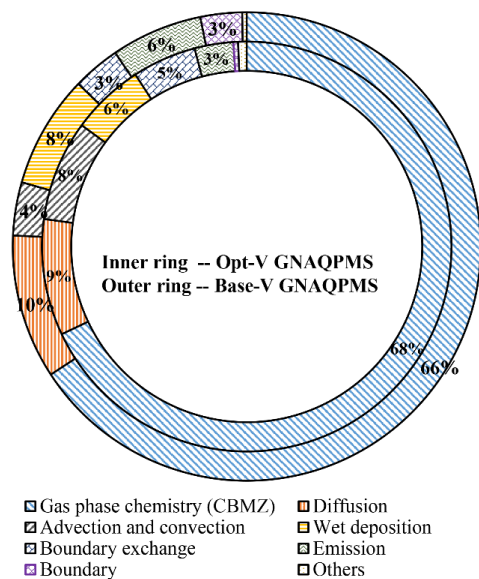
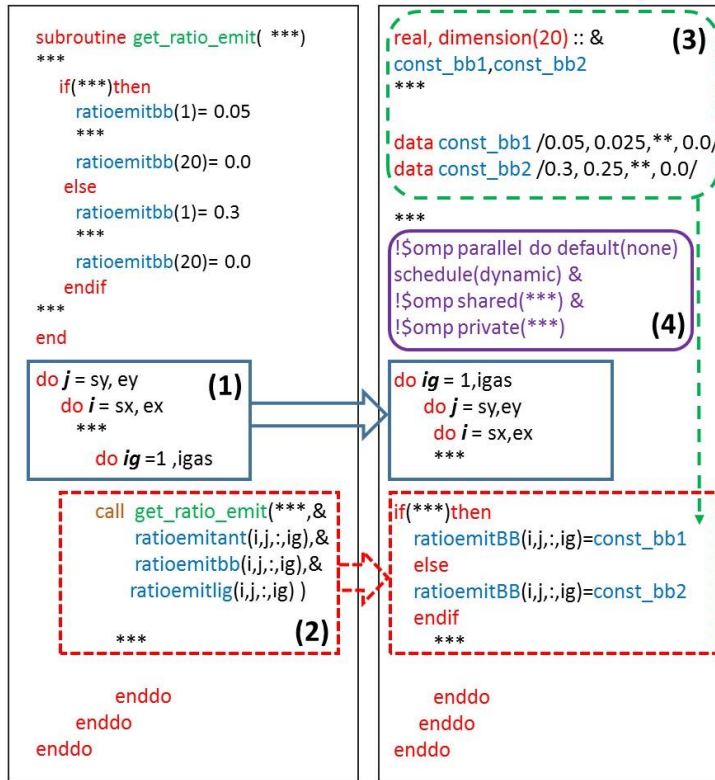
☒ Gas phase chemistry (CBMZ)   ☒ Diffusion
☒ Advection and convection   ☒ Wet deposition
☒ Boundary exchange   ☒ Emission
☒ Boundary   ☐ Others

5   **Figure 2. The overhead proportions of Base-V GNAQPMS (outer ring) and Opt-V GNAQPMS (inner ring). The top five most time consuming modules are CBM-Z, diffusion, advection, wet deposition, and boundary exchange.**

(a) Original code   (b) Optimized code

Figure 3 Simple codes contain some typical optimization methods. The part (a) is the original codes and part (b) is the optimized codes. Step (1) changes the order of the i, j, ig-loops, and step (2) make the subroutine inline. Step (3) uses the parameter to construct the scalar codes to vector codes, and step (4) adds the OpenMP pragmas.

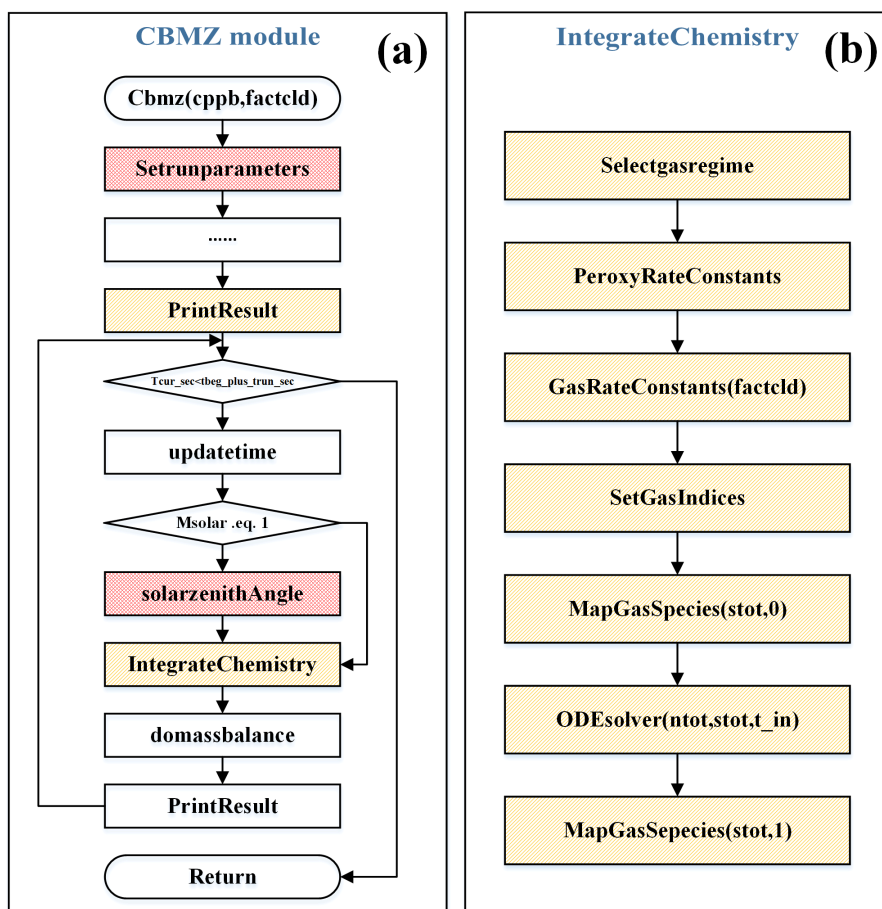



Figure 4. The flow charts of CBM-Z module (a) and the subroutine IntegrateChemistry (b), the red subroutines were removed and made into inline functions and the orange parts were modified for vectorization

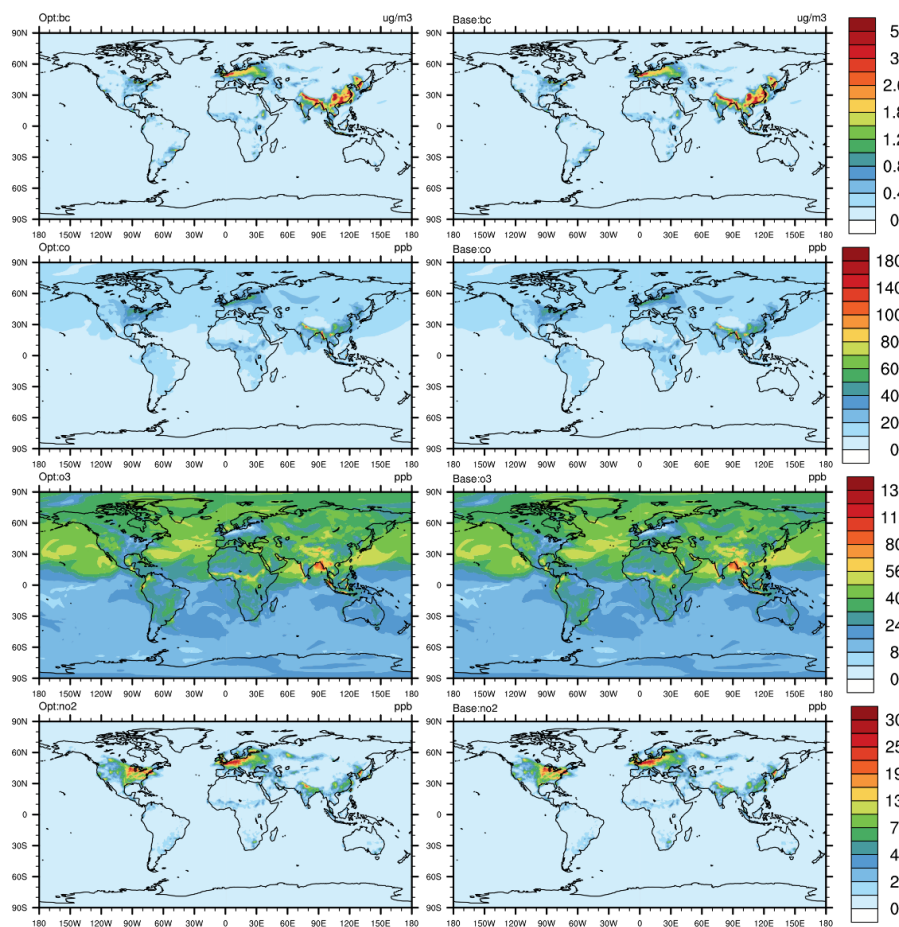



Figure 5. The spatial distribution of BC, CO, O3 and NO2 from Opt-V and Base-V GNAQPMS. The results are the same in spatial patterns and numerical values.

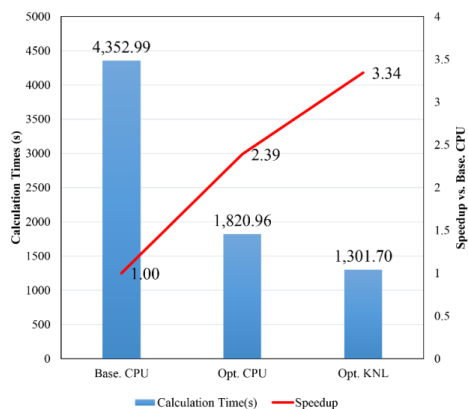



Figure 6. The performance test of Base-V and Opt-V GNAQPMS on CPU and KNL single node. Opt-V GNAQPMS got speedup of 2.39 and 3.34 on KNL platform respectively.

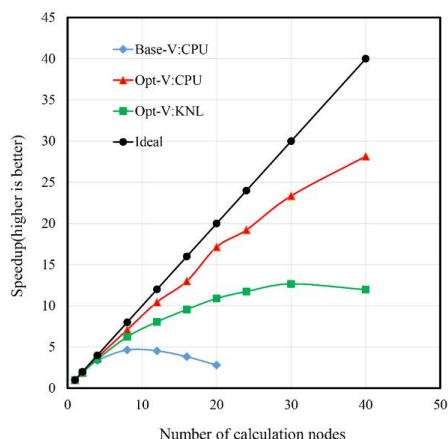



Figure 7. The scalability of the Base-V and Opt-V GNAQPMS on CPU and KNL cluster. The Base-V GNAQPMS on CPU cluster had the bad scalability and the performance nearly saturated on 8 nodes and Opt-V GNAQPMS can reach 40 nodes on CPU at least and 30 KNL nodes at most.

Table 1. The optimization measures for main modules and the speedup after optimization on CPU and KNL platform. The well parallelized modules (e.g. Emission module)can get a high speedup of 10.37x on KNL, and 3.14x speedup on KNL and 2.56x speedup on CPU were achieved for CBMZ gas chemistry module, the most time consuming part (66%) for the baseline version.

Items	OpenMP Optimization	Manual Vectorization	Removing TLS	Change global communication	Remove redundant calculation and memory access	Speedup on CPU	Speedup on KNL
Initialization				Yes		1.31	1.10
Emission	Yes	Yes		Yes	Yes	4.28	10.37
Advection and convection	Yes	Yes		Yes	Yes	1.54	2.91
Diffusion		Yes		Yes		1.78	2.39
Gas phase chemistry(CBM-Z)	Yes	Yes	Yes	Yes	Yes	2.56	3.14
Wet deposition	Yes			Yes		2.30	5.11

Table 2. Compile flags of Base-Version GNAQPMS and Opt-Version GNAQPMS on CPU and KNL platforms

VERSION OF GNAQPMS	INTEL COMPILER FLAGS
BASE-VERSION	-O3 –init=arrays –init=zero –fpp –w – traceback –ftz –fno-alias – fno-fnalias -g
OPT-VERSION(CPU PLATFORM)	–O3 –ip –init=arrays -xCore-AVX2 –fp-model fast=1 –O3 –ip – init=arrays –init=zero –qopenmp –ftz –fno-fnalias –fno-alias –g – w – traceback

OPT-VERSION(KNL PLATFORM)	–O3 –ip –init=arrays –MIC-AVX512 –fp-model fast=2 -align array64byte –qopenmp –ftz –fno-fnalias –fno-alias –g –w –traceback