Enabling the Acceleration of Dust Simulation using Job Scheduling Methods in a Cloud Environment

Manzhu Yu[1], Chaowei Yang[1], Zhenlong Li[1], Kai Liu[1], Songqing Chen[2]

[1]Center of Intelligent Spatial Computing, 4400 University Drive, Fairfax, Virginia 22030,
Email: myu7@gmu.edu

[2]Department of Computer Science, 4400 University Drive, Fairfax, Virginia 22030

Abstract

Dust modeling is highly computing intensive, and is normally executed through decomposition approach, by decomposing model domain evenly into multiple or many subdomains, which are then allocated to multiple computing nodes. The cost of data transfer due to communication among neighbor subdomains is a key efficiency issue because it adds significant overhead. In this research, two scheduling methods, including a K-Means and Kernighan-Lin combined heuristic algorithm and model default method, are applied to schedule the model run tasks onto multiple nodes. The overall results support that K&K can achieve better performance than default method. This research can also be applied to other similar computation problems, especially those related to big data.

Keywords: Domain Decomposition; Parallel Computing; Load Balancing; Optimization; Computing Algorithms.

1. Introduction

Dust impacts span a wide range of spatial and temporal scales and represents a serious hazard to health, property, environment and economy. Various dust models have been developed in the past several decades to predict dust emission, transport within the atmosphere and deposition. Dust models can be classified by their spatial coverage into regional or global coverage. Models with near global coverage, such as the Barcelona Supercomputing Centre-Dust Regional Atmospheric Model 8b v2.0 (BSC-DREAM8b) (Nickovic et al. 2001), are able to provide forecasts of the atmospheric life cycle of dust particles originating from deserts. As a regional model, Chinese Unified Atmospheric Chemistry Environment for Dust (CUACE/Dust) (Gong and Zhang 2001) model is an integral part of a real-time mesoscale sand and dust storm forecasting system for eastern Asia, which has an aerosol module that can differentiate the size of suspended particles. The model used in this study, NMM-Dust, is a meteorological core coupled with a dust module (Xie et al. 2010). The meteorological core is the non-hydrostatic mesoscale model (NMM), which is also used in US National Weather Service (NWS) operations. NMM-Dust can produce dust load and dust concentration in up to 3km spatial resolution.

However, dust modeling is highly computing intensive due to repetitive numerical calculations, vast dataset manipulations, and dust's intrinsic four-dimensional feature (a time dimension, two horizontal dimensions, latitude and longitude, and one vertical dimension) (Xie et al. 2010, Yang et al. 2011, Huang et al. 2013, Baillie et al. 1995). In

order to accelerate the simulation, parallelization is adopted by using message passing interface (MPI) programming model (Gropp et al. 1999). A Single Program Multiple Data (SPMD) decomposition approach is used to decompose the domain evenly into multiple or many subdomains, which are then allocated to multiple computing nodes. When two neighboring subdomains are allocated on two computing nodes separately, the intermediate data result of a subdomain generated on one computing node is transferred to another node across computer network. It inevitably introduces external communication. In the meantime, internal communication, in which data is transferred within a single computing node, occurs when neighboring subdomains are allocated on the same computing node. Comparing with external communication which introduces network I/O, internal communication cost (i.e., local disk/in-memory I/O) is trivial and can be ignored to some extent.

The cost of data transfer due to communication among neighbor subdomains is a key efficiency issue because it adds significant overhead (Baillie et al. 1997). Different subdomain allocation methods result in different communication overheads among the subdomains. Since MPI is not responsible for scheduling, how subdomains are allocated to the computing nodes are customized by model developers and engineers. If there is no customized allocation, the system dispatches the subdomains to the computing nodes sequentially row after row. By using cluster allocation and non-cluster allocation methods to run the same set of model simulation tasks under different decomposition granularities, Huang et al. (2013) validated this hypothesis and indicated that cluster allocation method achieved an average 20% performance improvement.

When dividing rules and the number of subdomains are specified for a given domain, the allocation method becomes a key issue to the simulation performance. Therefore, it is desirable and significantly useful to find an optimized case-dependent subdomain allocation method. An optimized allocation requires to best leverage the computing capacity gains and communication costs for minimizing numerical calculation time. The dynamic and heterogeneous features of environments as well as spatial and communicational constraints should be considered (Su et al. 2014). To address the critical issue of communication costs, it is demanded to take full advantage of modern task-scheduling approaches. In this research, two scheduling methods, including a K-Means and Kernighan-Lin combined heuristic algorithm and model default method, are applied to schedule the model run tasks onto multiple nodes.

2. Methodology

K-Means and Kernighan-Lin combined algorithm (K&K) integrates K-Means clustering partitioning method with Kernighan-Lin local partitioning method. K-Means can leverage the locality and shapes of components based on the graphical distribution of subdomains in general, but it can neither balance the number of subdomain, nor reduce connections between components. While Kernighan-Lin is a bi-partitioning method which can reduce connections between components effectively, it may get trapped to a local minima (Schloegel et al. 2000) and may not produce good results for multiple components partitioning since the algorithm cannot consider partitioning shapes of the entire domain at one time. Therefore, a combination algorithm could leverage advantages and reduce the drawbacks of both methods.

On-demand cloud computing has been offering a new dimension to High Performance Computing (HPC) applications, in this case dust modelling, in which virtualized resources can be sequestered, in a form customized to target a specific scenario, at the time and in the manner they are desired. In order to test the performance improvement using K&K compared to model default method, a cloud cluster is needed to be created that could run the model typically run on more traditional HPC platform and take advantage of the interactive, on-demand capabilities the resource provides. The creation and customization of cloud cluster is conducted according to the execution environment of NMM-dust.

The great challenge to using multiple compilers is that the MPI runtime needs to be rebuilt for the new compiler every time. In order to effectively test the job scheduling effect, an automated workflow is utilized to run and provision tasks from the start to the end of each experiment. Based on subdomain decomposition and the number of computing node, job scheduling algorithm is executed to produce the allocation solution. The allocation solution is then utilized onto model execution.

Regarding to Input/Output and filesystem, the proposed approach utilizes the most straightforward way of providing a shared diskspace for parallel processes running on cloud cluster on demand, exporting home (and other if necessary) directory space over NFS to all nodes. It is a very basic configuration that is always desirable as it enables ease of parallel operation (no need to propagate executables to all nodes etc.) although the performance limitations of NFS would come into play when the number of participating nodes becomes very large.

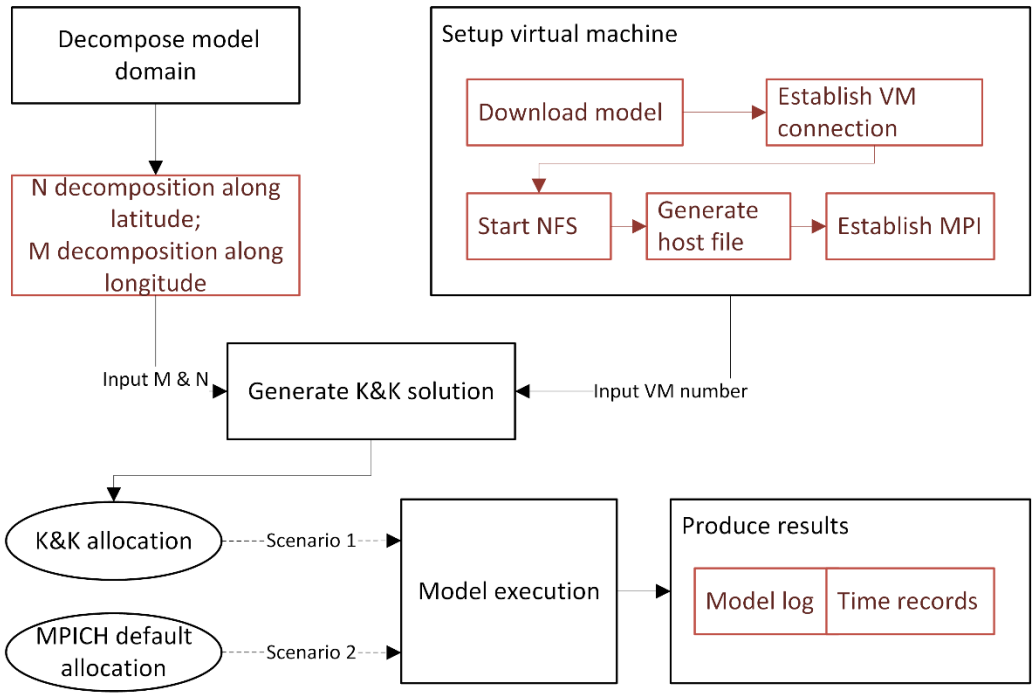

Figure 1. Experiment workflow.

3. Experiments

Model domain is evenly divided into 4 to 128 subdomains along latitude and longitude directions. Model run experiments are conducted in a cloud environment managed by Eucalyptus[1] in a physical cluster (Table 1). Up to 8 virtual machines are used according to experimental scenario and each one composed by 2 CPUs and 2048 MB of memory. In all executions, we simulated dust condition for 3 km resolution, 4.8 by 4.8 degree domain size for 72 hours. The vertical atmosphere layer is divided into 45 layers. For the same computing node number and the same subdomain setting, we ran the model twice, using the default allocation and K&K result respectively. Performance is measured by recording the start and end of execution time of each subroutine throughout the model execution.

Experiment	Virtual machine	Subdomain
K&K vs. default	2-8 node	4(2*2) 8(4*2) 16(4*4) 32(8*4) 48(8*6) 64(8*8) 80(10*8) 96(12*8) 104(13*8) 112(14*8) 120(15*8) 128(16*8)

Table 1. Experiment design.

4. Results

The overall results support that K&K can achieve better performance than default method. Two series of plots, subdomain number vs. time plot (Figure 2) and node number vs. time plot (Figure 3), are used to demonstrate the patterns of subdomain number, node number, and execution time (displayed as bar plot). Besides, we illustrate a performance improvement factor on the plots for better demonstration (displayed as grey lines). Here we define performance improvement factor as Eq. 1:

$$f = \Delta t / t_{default} \qquad (1)$$

where $t_{default}$ is default allocation runtime, and Δt is the difference between default allocation runtime and K&K allocation runtime.

By controlling the same number of computing node, execution time using different number of subdomain are recorded and compared. Results show (Figure 2) that the overall execution time increases, while in some cases decreases first and then increases,

[1] http://www.eucalyptus.com

when the number of subdomain increases. The pattern suggests that dividing a domain into finer scale subdomains cannot necessarily reduce execution time, especially when the subdomain number is substantially larger than the node number.

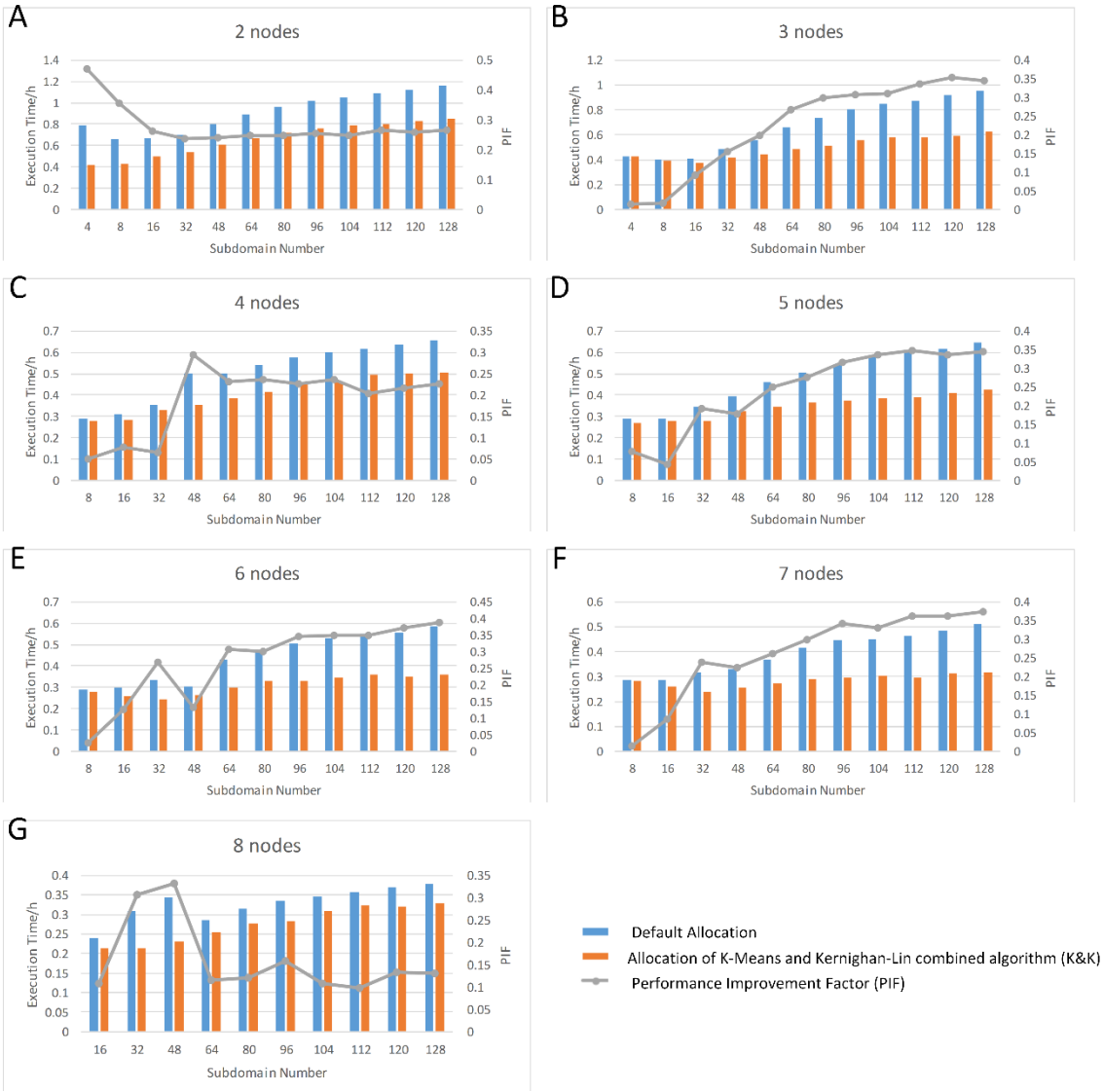

Figure 2. Subdomain Number - Time Plot.

By controlling the same number of subdomain, execution time using different number of node are recorded and compared (Figure 3). As the number of nodes increases for the first several nodes, there is an obvious pattern of decreased execution time. When the increasing node reach a certain point, the pattern of decreasing execution time turns out to be insignificant.

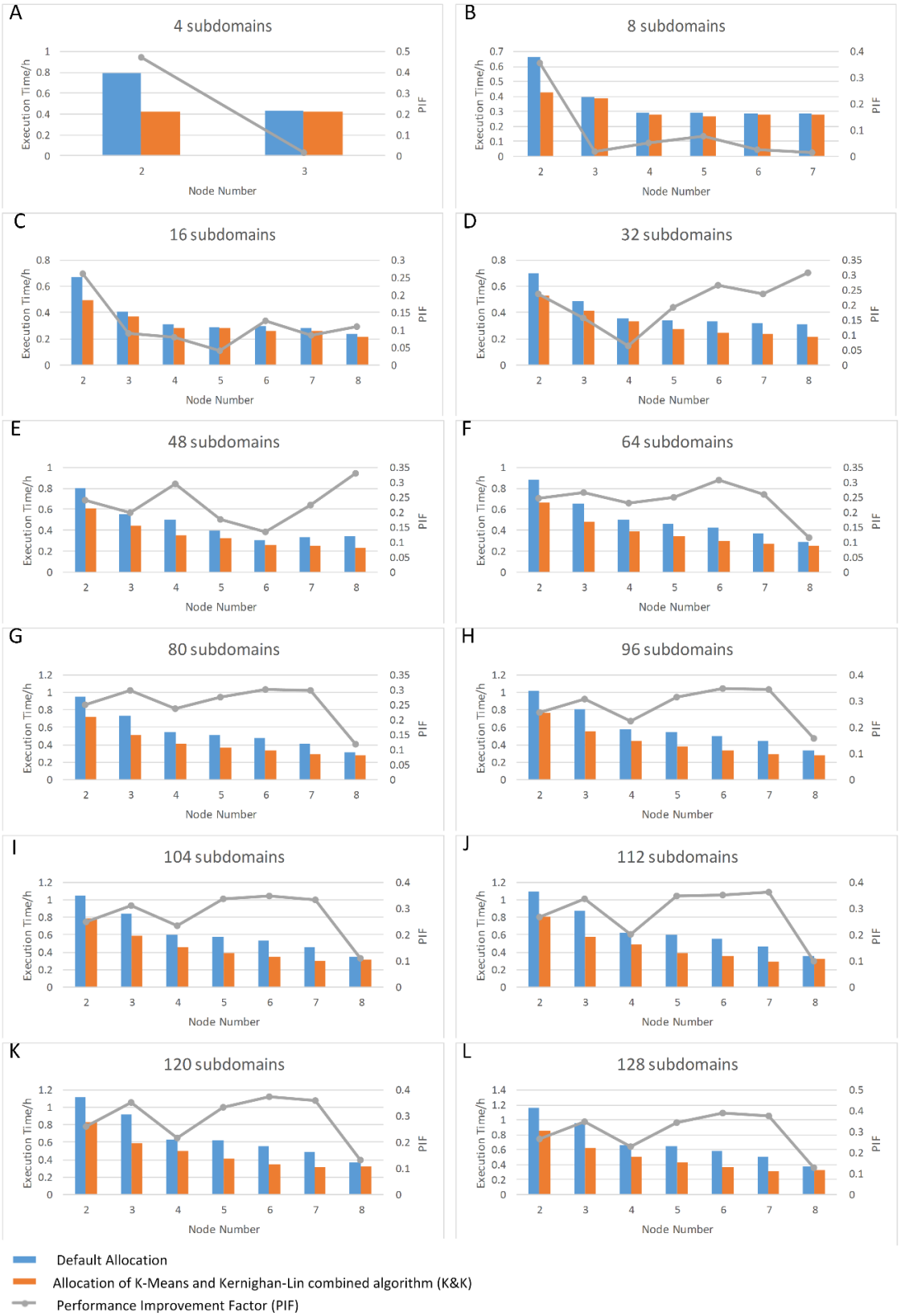

Figure 3. Node Number - Time Plot

Although PIF values indicate that the performance using K&K is overall improved, we can hardly summarize PIF as a simple function of node number or subdomain division. However, the unstable patterns of performance improvement provide the insight that different settings of node number and subdomain divisions generate performance improvements to varying degrees.

5. Discussion

This abstract described a combination of (1) a cloud on-demand cluster system with adequate performance (2) a custom image designed to make the system use as easy as possible (3) a job scheduling method (K&K) and (4) a dust model NMM-dust. This combination offers a compelling case for the acceleration of dust simulation using job scheduling methods in a cloud environment. Performance is comparable with low-cost cluster systems, and the results are encouraging. It may provide suggestions about the granularity of subdomain to achieve best resource usage and high efficiency.

6. References

Baillie, C, MacDonald, A & Sun, S 1995, QNH: a portable, massively parallel multi-scale meteorological model, in *Proceedings of the Fourth Int'l Conference on the Applications of High Performance Computers in Engineering*.

Baillie, C, Michalakes, J & Skålin, R 1997, Regional weather modeling on parallel computers, *Parallel Computing*, 23(14):2135-2142.

Gong, S & Zhang, X 2008, CUACE/Dust–an integrated system of observation and modeling systems for operational dust forecasting in Asia, *Atmospheric Chemistry and Physics*, 8(9):2333-2340.

Gropp, W, Lusk, E & Skjellum, A 1999, Using MPI: portable parallel programming with the message-passing interface, *MIT press*.

Huang, Q, Yang, C, Benedict, K, Rezgui, A, Xie, J, Xia, J & Chen, S 2013, Using adaptively coupled models and high-performance computing for enabling the computability of dust storm forecasting, *International Journal of Geographical Information Science*, 27(4):765-784.

Nickovic, S, Kallos, G, Papadopoulos, A & Kakaliagou, O 2001, A model for prediction of desert dust cycle in the atmosphere, *Journal of Geophysical Research: Atmospheres (1984–2012)*, 106(D16):18113-18129.

Schloegel, K, Karypis, G & Kumar, V 2000, Graph partitioning for high performance scientific simulations, *Citeseer*.

Su, X, Zhang, M, Ye, D & Bai, Q 2014, A Dynamic Coordination Approach for Task Allocation in Disaster Environments under Spatial and Communicational Constraints.

Xie, J, Yang, C, Zhou, B & Huang, Q 2010, High-performance computing for the simulation of dust storms, *Computers, Environment and Urban Systems*, 34(4):278-290.

Yang, C, Wu, H, Huang, Q, Li, Z & Li, J 2011, Using spatial principles to optimize distributed computing for enabling the physical science discoveries, *Proceedings of the National Academy of Sciences*, 108(14):5498-5503.