Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Wind field parallelization based on Schwarz alternating domain decomposition method

Gemma Sanjuan, Tomàs Margalef*, Ana Cortés

High Performance Computing Applications for Science and Engineering research group, Universitat Autònoma de Barcelona, Spain

HIGHLIGHTS

- Preconditioned Conjugate Gradient solver was parallelized using OpenMP.
- Jacobian preconditioner offers better scalability than SSOR.
- The Schwarz alternating domain decomposition method has been applied.
- Combining Schwarz method and PCG parallelization reaches operational execution time.

ARTICLE INFO

Article history:
Received 23 February 2016
Received in revised form
2 December 2016
Accepted 29 December 2016
Available online xxxx

Keywords:
Wind field
Forest fire
Preconditioned Conjugate Gradient
SSOR
Jacobi
Domain decomposition
Schwarz method

ABSTRACT

Wind field is a critical issue in forest fire propagation prediction. However, wind field calculation is a complex problem that, for large terrains, involves solving huge linear systems. To solve such systems, the Preconditioned Conjugate Gradient (PCG) solver is applied. SSOR and Jacobi preconditioners are usually used, but solving such systems takes too much time and makes the approach unfeasible in real time operation. Parallelization appears as a way to make the approach operational in real time. The PCG with both preconditioners has been parallelized to accelerate the execution. However, the improvement in execution time is not enough, and the Schwarz alternating domain decomposition has been applied to exploit a second level of parallelism. Using this method, the linear system is decomposed in a set of overlapped subdomains that can be solved in parallel using a Master/Worker paradigm, where each worker exploits the PCG solver parallelism. As a result, the wind field calculation time is significantly reduced; for example, a large map of 1200 × 1200 cells, whose solution took more than 2000 seconds in the original WindNinja, can now be solved in less than 240 seconds using 4 subdomain and 4 cores per subdomain.

1. Introduction

Forest fire is a natural hazard that destroys ecosystems and prolific areas and causes significant economic and societal losses every year. Therefore, it is necessary to provide extinction services with the best means of mitigating the effects of such hazards. In this context, forest fire propagation prediction appears as a significant contribution to allowing extinction services to use the available resources in the best possible way.

Several models have been developed and integrated into computer simulators (FARSITE [1], FireStation [2], Wildfireanalyst [3] or CARDIN [4]) to estimate forest fire propagation. These propagation models, and the consequent simulators, need a large set of input parameters describing the actual scenario in which the fire is taking place. These parameters include a terrain elevation map, a vegetation map and the features of said vegetation, an initial fire perimeter, and meteorological conditions.

Two of the parameters that most significantly affect fire behavior are wind speed and direction. Therefore, providing accurate values for these parameters is a crucial issue in obtaining accurate forest fire propagation predictions. The values provided by meteorological services are obtained from meteorological stations located at particular sites or provided by meteorological models at a very low resolution (i.e. 4 km). However, the meteorological wind is significantly modified by the topography of the terrain, and each point of the terrain receives particular wind speed and direction. Therefore, it is necessary to apply a wind field model that provides the wind speed and direction at a high resolution (i.e. 30 m). This wind field model must be coupled with the forest fire propagation model to improve the accuracy of the predictions.

* Corresponding author.
E-mail addresses: gemma.sanjuan@caos.uab.es (G. Sanjuan), tomas.margalef@uab.es (T. Margalef), ana.cortes@uab.es (A. Cortés).

http://dx.doi.org/10.1016/j.future.2016.12.041

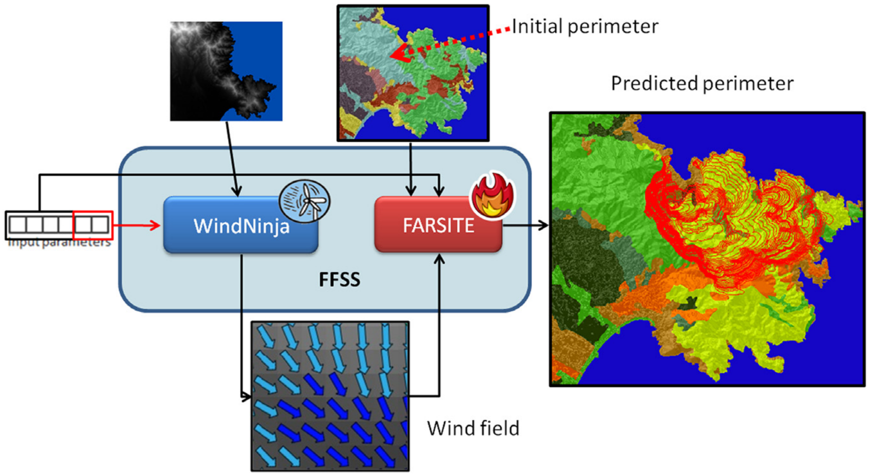

Fig. 1. Coupling wind field and forest fire propagation simulators.

In this work, the forest fire simulator selected is FARSITE [1], because it has been extensively validated and is widely used throughout the firefighting community. Taking into account that the forest fire simulator used is FARSITE, the wind field simulator chosen is WindNinja [5,6], because it can accept the same input files as FARSITE and can generate wind field files that can be directly used by FARSITE. So, it is very easy to couple WindNinja and FARSITE [7]. This coupled scheme is shown in Fig. 1.

Calculating the wind field at high resolution requires the creation of a dense and large mesh describing the terrain and linear systems representing the physical principles involved, such as mass conservation. Such huge systems with millions of unknowns, in most cases, must be solved applying iterative methods that require a large number of iterations (over one thousand iterations) and take an amount of time (more than 30 min) that prevents their use in operational scenarios.

So, overlapping domain decomposition [8] appears as a promising approach to solving such large systems since the mesh is divided into subdomains and the linear systems involved are consequently reduced [9,10], implying a significant reduction in wind field calculation execution time. It must be considered that the complete linear system implies a huge matrix with millions of rows, where the elements in the main diagonal represent the nodes in the mesh. Such large matrices cannot usually be loaded in the main memory, and it is necessary to access data in the disk using swapping. So, using a domain decomposition approach implies a significant reduction in the size of each subdomain and this facilitates the storage of the submatrices in the nodes' memory, avoiding disk use. This work focuses on applying the Schwarz domain decomposition method with overlapping to accelerate wind field calculation and coupling the wind field simulator to a forest fire propagation simulator.

The rest of this paper is organized as follows: In Section 2, the main features and limitations of the WindNinja wind field simulator are described. Section 3 presents the Schwarz alternating domain decomposition method. Section 4 shows the application of domain decomposition with overlapping to the particular case of WindNinja. Section 5 presents the experimental results. Finally, Section 6 summarizes the conclusions of this work.

2. WindNinja wind field simulator

WindNinja [5] is a wind field simulator that takes a meteorological wind and determines the wind at each point of the terrain at a given resolution. To carry out such calculations, two main inputs are required: meteorological wind parameters and the digital elevation map of the terrain. The internal functioning of WindNinja can be summarized in five phases, as shown in Fig. 2:

1. It generates the mesh discretizing the terrain map.
2. It applies the mass conservation equations to each point of the mesh to generate a system of equations, $Ax = b$.
3. Matrix A is a sparse matrix that is stored in CRS (Compressed Row Storage) format. An in-depth study of WindNinja matrices has been carried out. WindNinja matrices always present the same pattern with only a few differences. Most of the elements, representing interactions among the nodes of the mesh, are concentrated around the diagonal and just very few elements are a little bit further. Each node of the mesh presents maximum 14 of interrelations with other nodes in the same subdomain or in the interface subdomain. So, the sparse matrix presents a well-defined pattern with the main diagonal and 13 subdiagonals on the upper triangular part of the matrix and the same values on the lower triangular part of the matrix.
4. It applies the Preconditioned Conjugate Gradient (PCG) [11] method to solve the system of equations. The PCG is an iterative method that can only be applied when the sparse matrix representing the system is symmetric and positive definite and real. It uses a matrix M as a preconditioner and iteratively approaches the solution by applying the algorithm. The preconditioner determines the convergence of the system.
5. The solution of the solver is used to construct the wind field.

2.1. The preconditioned Conjugate Gradient method

The simplest way to solve the system $Ax = b$ would be to calculate A^{-1}, the inverse of matrix A, and then multiply by this inverse matrix on both sides of the equation, so that Eq. (1) is reached:

$$A^{-1}Ax = A^{-1}b. \quad (1)$$

From this equation, it can be reached that:

$$x = A^{-1}b. \quad (2)$$

So, solving the system would be just to calculate the inverse matrix of A and then multiply it by vector b. However, when the matrices are very large, calculating the inverse matrix is a very time-consuming task that becomes impossible. So, iterative methods are introduced to approximate the solution of the system. A particular case is the Conjugate Gradient Method (CG). This method can be described intuitively, as shown in Fig. 3. In this figure, it can be observed that starting from an initial approximation of the solution x, an orthogonal vector g at the surface in x is obtained. Then, an orthogonal vector to g, q, is calculated. These two orthogonal vectors provide the