ELSEVIER

Parallel Computing 28 (2002) 35–52

PARALLEL COMPUTING

www.elsevier.com/locate/parco

Applications

Distributed-memory concepts in the wave model WAVEWATCH III ☆

Hendrik L. Tolman *

SAIC/GSO at NOAA/NCEP, Environmental Modeling Center, 5200 Auth Road, Room 209, Camp Springs, MD 20746, USA

Received 4 January 2001; received in revised form 14 June 2001; accepted 31 August 2001

Abstract

Parallel concepts for spectral wind-wave models are discussed, with a focus on the WAVEWATCH III model which runs in a routine operational mode at NOAA/NCEP. After a brief description of relevant aspects of wave models, basic parallelization concepts are discussed. It is argued that a method including data transposes is more suitable for this model than conventional domain decomposition techniques. Details of the implementation, including specific buffering techniques for the data to be communicated between processors, are discussed. Extensive timing results are presented for up to 450 processors on an IBM RS6000 SP. The resulting model is shown to exhibit excellent parallel behavior for a large range of numbers of processors. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Ocean wind-wave modelling; Distributed memory computing; Message passing

1. Introduction

For several decades, numerical wind-wave models have been an integral part of weather prediction at weather forecast centers around the world. Major meteorological centers now rely on the so-called third-generation wave models like WAM [3,9], or WAVEWATCH III [7,8]. In such models, all physical processes describing wave growth and decay are parameterized explicitly. Compared to previous first- and second-generation models, which parameterized integral effects of the physics rather

☆ OMB contribution No. 201.

* Tel.: +1-301-763-8133; fax: +1-301-763-8545.
E-mail address: hendrik.tolman@noaa.gov (H.L. Tolman).

0167-8191/02/$ - see front matter
PII: S0167-8191(01)00130-2

than the physics itself, this is computationally expensive due to the explicit calculation of nonlinear wave–wave interactions, and due to the relatively small time-steps required by third-generation models. Although such models are still less computationally intensive than atmospheric models, they nevertheless require state-of-the-art supercomputer facilities to produce forecasts at acceptable resolutions and in a timely fashion.

The first supercomputers utilized the concept of vectorization. Such vector computers achieved increased computational performance by efficiently performing identical calculations on large sets of data. Conversion of computer models to such computers generally required systematic reorganization of the programs to generate long loop structures. Initial vector computers also required much hardware-dependent calls for basic operations. In later vector computers, additional programming was essentially limited to the inclusion of compiler directives in the source code.

The second supercomputing paradigm is that of parallelization. In this case the work is spread over multiple processors. In the simplest form (from a user's perspective), the processors share memory. As with vectorization, the success of such parallelization depends on the general structure of the program. If this structure is conducive to parallelization, modifications to the program for shared-memory parallel computers are generally small, and are usually limited to adding compiler directives to the program. Sharing memory between processors, however, requires additional logistics in the computer, which generally limits the number of processors in shared-memory parallel computers to about 16. Much more massively parallel computers with up to $O(10^3)$ processors can be constructed if the processors do not share their memory. Such distributed-memory parallel computers represent the latest development in supercomputing. Efficient application of models to such computers requires that the communication between processors becomes an integral part of the source code. Application to distributed memory computers therefore requires major code conversions, even for programs that are already applied to shared-memory parallel computers.

The present paper describes the conversion of the operational NOAA implementation of the third-generation wind-wave model WAVEWATCH III (henceforth denoted as NWW3) to a distributed memory computer architecture at the National Centers for Environmental Prediction (NCEP). The program is written in FORTRAN. For message passing between processors the message passing interface (MPI) standard has been used (e.g., Gropp [2]). In Section 2, a brief description of the model is given, together with previously used vectorization and parallelization approaches. In Sections 3 and 4, the basic distributed memory design and model modifications are discussed, as well as optimization considerations. In Section 5 the performance of the parallel code at NCEP is discussed. Sections 6 and 7 present a discussion and conclusions.

2. The wave model

In contrast to numerical models for the atmosphere and ocean, which provide a deterministic description of both media, wind-wave models provide a statistical

description of the sea state. The spatial and temporal scales of individual waves make it impossible to deterministically predict each individual wave for an entire ocean or sea. The random character of wind-waves makes it undesirable to deterministically model individual waves. Most statistical properties of wind-waves are captured in the distribution of wave energy over wave frequency (or wavenumber) and wave propagation direction, in the so-called wave energy density spectrum, or for short, the energy spectrum. Wave models generally predict the evolution in space and time of the energy spectrum, or alternatively, of the action spectrum. The action spectrum is the energy spectrum divided by the intrinsic frequency of the spectral components. The action spectrum is used in recent models as it allows for the transparent inclusion of effects of mean currents on the evolution of the wave field.

NWW3 predicts the evolution in the two-dimensional physical space $\mathbf{x}$ and time t of the wave action density spectrum A as a function of the wavenumber k and direction θ, as governed by the conservation equation

$$\frac{DA(k,\theta;\mathbf{x},t)}{Dt} = S(k,\theta;\mathbf{x},t). \tag{1}$$

The total derivative on the left represents the local change and effects of wave propagation. The function S represents source terms for wave growth and decay, that are governed by the direct action of wind, the exchange of action between components of the spectrum due to nonlinear effects, the action loss due to white-capping, and from additional shallow water processes if applicable. Physical space ($\mathbf{x}$) and spectral space (k,θ) are discretized, and the equation is solved marching forward in time t, using a (global) time-step Δt_g. Below, i, j and m will denote discrete grid counters in k-, θ- and $\mathbf{x}$-space, respectively. In NWW3, $\mathbf{x}$-space consists of a regular longitude–latitude grid. To reduce memory requirements of the model, spectra for grid points that are located on land are not stored, reducing m to a one-dimensional counter in the two-dimensional $\mathbf{x}$-space.

To facilitate an economical solution, and at the same time to simplify the numerical approaches, Eq. (1) is solved in several consecutive fractional steps (e.g., [10]). A fractional step method is used in virtually every spectral wave model. Details, however, differ between models. In NWW3, the fractional step approach addresses spatial propagation, spectral propagation and source terms separately. Thus the following three equations are solved consecutively:

$$\frac{\partial A}{\partial t} + \nabla_{\mathbf{x}} \cdot \mathbf{c}_{\mathbf{x}} A = 0, \tag{2}$$

$$\frac{\partial A}{\partial t} + \nabla_{k,\theta} \cdot \mathbf{c}_{k,\theta} A = 0, \tag{3}$$

$$\frac{\partial A}{\partial t} = S. \tag{4}$$

Here the functional dependencies of A and S have been dropped for convenience. $\nabla_{\mathbf{x}}$ and $\nabla_{k,\theta}$ represent differential operators in physical and spectral spaces, respectively. $\mathbf{c}_{\mathbf{x}}$ represents the propagation velocity vector in physical space, that is a function of