sbPOM: A parallel implementation of Princenton Ocean Model

Antoni Jordi[1] and Dong-Ping Wang[2]

[1]Institut Mediterrani d'Estudis Avançats, IMEDEA (UIB-CSIC), Miquel Marqués 21, 07190 Esporles, Illes Balears, Spain

[2]School of Marine and Atmospheric Sciences, Stony Brook University, Stony Brook, NY 11794, USA.

Corresponding author: Antoni Jordi

Institut Mediterrani d'Estudis Avançats, IMEDEA (UIB-CSIC), Miquel Marqués 21, 07190 Esporles, Illes Balears, Spain

E-mail: toni@imedea.uib-csic.es

Phone: +34 971 611 910

Fax: +34 971 611 761

Abstract

This paper presents the Stony Brook Parallel Ocean Model (sbPOM) for execution on workstations, Linux clusters and massively parallel supercomputers. The sbPOM is derived from the Princenton Ocean Model (POM), a widely used community ocean circulation model. Two-dimensional data decomposition of the horizontal domain is used with a halo of ghost cells to minimize communication between processors. Communication consists of the exchange of information between neighbor processors based on the Message Passing Interface (MPI) standard interface. The Parallel-NetCDF library is also implemented to achieve a high efficient input and output (I/O). Parallel performance is tested on an IBM Blue Gene/L massively parallel supercomputer, and efficiency using up to 2048 processors remains very good.

Keywords. Ocean circulation: Numerical model; Parallel computing

Software availability

- *Software name*: sbPOM

- *Developers*: Antoni Jordi and Dong-Ping Wang

- *Contact*: toni@imedea.uib-csic.es

- *Software requirements*: Linux or Unix, fortran compiler, mpich 2, parallel NetCDF

- *Programming language*: Fortran 77

- *Availability and cost*: Test cases are available at http://imedea.uib-csic.es/users/toni/sbpom at no cost

1. Overview

Ocean circulation models are an integral part of the ocean observing system, which play a central role in the study of global climate and biogeochemical cycles. There are many ocean models using different approaches to spatial discretization and vertical coordinate treatment, numerical algorithms for time-stepping, advection, pressure gradient, and subgrid-scale parameterizations. Reviews on ocean models and recent developments can be found in Griffies et al. (2000) and Ezer et al. (2002). Among them, the Princeton Ocean Model (POM) is widely adopted (http://www.aos.princeton.edu/WWWPUBLIC/htdocs.pom/). The POM community has developed several coupled models for biogeochemical (Chau, 2003; Vichi et al., 1998), sediment (Liu and Huang, 2009; Xu et al., 2010), and operational applications (Nittis et al., 2006; Price et al., 2006). Recent advances include the wetting and drying scheme (Oey, 2005) and surface wave-ocean current coupling (Mellor et al., 2008). In addition, several parallel POM codes have been described (Boukas et al., 1999; Giunta et al.,

2007). However, these codes are not publicly available, are limited to the use of specific hardware architectures, or are extensively modified from the serial code. In response to these limitations, we developed a new parallel version of POM (the sbPOM) available to the general public, which minimizes the modification to the POM code and achieves good scalability on a wide range of number of processors.

2. Model description

The procedure for the parallelization of POM is the implementation of a message-passing code using two-dimensional data decomposition of the horizontal domain. This approach ensures the portability across a large variety of parallel machines, and allows to maintain the same numerical algorithms used in the serial code. The horizontal global domain is thus partitioned into two-dimensional local domains using a Cartesian decomposition, and the vertical dimension is not divided. Each local domain (i.e. each processor) integrates independently the code on the local domain. The computation applied to each local domain is the same as that applied to the entire global domain with the serial code. The horizontal arrays assigned to each local domain are expanded by one grid point in each horizontal dimension, creating a halo of ghost cells (Fig. 1). The two- and three-dimensional arrays in these ghost cells are exchanged between neighbor local domains using the MPI standard interface (http://www.mcs.anl.gov/research/projects/mpich2/).

The sbPOM has implemented Parallel-NetCDF (http://trac.mcs.anl.gov/projects/parallel-netcdf), which provides high-performance parallel I/O while still maintains file-format compatibility with Unidata's NetCDF (http://www.unidata.ucar.edu/software/netcdf/). This makes the files space-efficient,

self-describing and machine independent. NetCDF is also recognized by many graphics and post-processing utilities.

In order to assess the performance of sbPOM, we used the seamount test case (Ezer et al., 2002; Mellor et al., 1998) which is made available with the code. It is a stratified Taylor column problem which simulates the flow across a seamount. The test case has 1026 × 770 global grid points and 31 vertical sigma levels. The simulation time is measured while varying the number of processors on a Blue Gene/L massively parallel supercomputer housed in the New York Center for Computational Sciences (http://www.newyorkccs.org/). The problem size for the test case is too large to run on one or two processors due to memory allocation. Simulations are therefore run with a number of processors ranging from 4 to 2,048 (2K), and the efficiency is measured with respect to the four processors, $t_4/t_{nproc} \cdot (nproc/4)$, where t_{nproc} is the simulation time when using *nproc* processors. The efficiency is very high even for 2K processors (~ 0.8) (Fig. 2).

3. Concluding remarks

POM is a relatively simple community model with very few options for numerics, physics, boundary conditions, and I/O. In this regard, sbPOM also is a relatively simple parallelization in order to maintain the POM legacy. This should provide POM developers/users with a familiar code that they can easily use or modify. The user community is encouraged to parallelize recent advances in POM and to develop new parallel techniques. Despite its simplicity, sbPOM test results on IBM BlueGene/L show a very high efficiency. The supreme CPU performance on supercomputers opens a new era for high-resolution ocean modeling. An example is application of sbPOM with

high spatial resolution to simulate submesoscale processes of surface frontogenesis of the Mid-Atlantic Bight (U.S.A) shelfbreak front (Wang and Jordi, 2011).

Acknowledgments

This research utilized resources at the New York Center for Computational Sciences at Stony Brook University/Brookhaven National Laboratory which was supported by the U.S. Department of Energy under Contract No. DE-AC02-98CH10886 and by the State of New York. A. Jordi's work was supported by a Ramón y Cajal grant from MICINN.

References

Boukas, L.A., Mimikou, N.T., Missirlis, N.M., Mellor, G.L., Lascaratos, A., Korres, G., 1999. The Parallelization of the Princeton Ocean Model, In: Amestoy, P. (Ed.), Lecture Notes in Computer Science. Springer-Verlag: Berlin, pp. 1395-1402.

Chau, K.W., 2003. Manipulation of numerical coastal flow and water quality models. Environmental Modelling & Software 18(2) 99-108.

Ezer, T., Arango, H., Shchepetkin, A.F., 2002. Developments in terrain-following ocean models: intercomparisons of numerical aspects. Ocean Modelling 4(3-4) 249-267.

Giunta, G., Mariani, P., Montella, R., Riccio, A., 2007. pPOM: A nested, scalable, parallel and Fortran 90 implementation of the Princeton Ocean Model. Environmental Modelling & Software 22(1) 117-122.

Griffies, S.M., Boning, C., Bryan, F., Chassignet, E.P., Gerdes, R., Hasumi, H., Hirst, A., Treguier, A.M., Webb, D., 2000. Developments in ocean climate modelling. Ocean Modelling 2 123-192.

Liu, X.H., Huang, W.R., 2009. Modeling sediment resuspension and transport induced by storm wind in Apalachicola Bay, USA. Environmental Modelling & Software 24(11) 1302-1313.

Mellor, G.L., Donelan, M.A., Oey, L.Y., 2008. A Surface Wave Model for Coupling with Numerical Ocean Circulation Models. Journal of Atmospheric and Oceanic Technology 25(10) 1785-1807.

Mellor, G.L., Oey, L.Y., Ezer, T., 1998. Sigma coordinate pressure gradient errors and the Seamount problem. Journal of Atmospheric and Oceanic Technology 15(5) 1122-1131.

Nittis, K., Perivoliotis, L., Korres, G., Tziavos, C., Thanos, I., 2006. Operational monitoring and forecasting for marine environmental applications in the Aegean Sea. Environmental Modelling & Software 21(2) 243-257.

Oey, L.Y., 2005. A wetting and drying scheme for POM. Ocean Modelling 9(2) 133-150.

Price, J.M., Reed, M., Howard, M.K., Johnson, W.R., Ji, Z.G., Marshall, C.F., Guinasso, N.L., Rainey, G.B., 2006. Preliminary assessment of an oil-spill trajectory model using satellite-tracked, oil-spill-simulating drifters. Environmental Modelling & Software 21(2) 258-270.

Vichi, M., Pinardi, N., Zavatarelli, M., Matteucci, G., Marcaccio, M., Bergamini, M.C., Frascari, F., 1998. One-dimensional ecosystem model tests in the Po Prodelta area (Northern Adriatic Sea). Environmental Modelling & Software 13(5-6) 471-481.

Wang, D.P., Jordi, A., 2011. Surface frontogenesis and thermohaline intrusion in a shelfbreak front. Ocean Modelling 38(1-2) 161-170.

Xu, F.H., Wang, D.P., Riemer, N., 2010. An idealized model study of flocculation on sediment trapping in an estuarine turbidity maximum. Continental Shelf Research 30(12) 1314-1323.

Figure captions

Fig. 1 Example of data decomposition scheme for a global size of 14 x 17, local sizes of 6 x 7, and 9 processors. Crosses represent the boundaries of the global domain, shadow areas are the ghost cells (or local boundaries) and arrows indicate the communication between local domains to exchange variables at the ghost cells.

Fig. 2 Performance for the test case. (a) Time (seconds per simulation day) as a function of the number of processors. (b) Parallel efficiency relative to performance on four processors. The black line (in both panels) is efficiency equal to one.

Figure 1

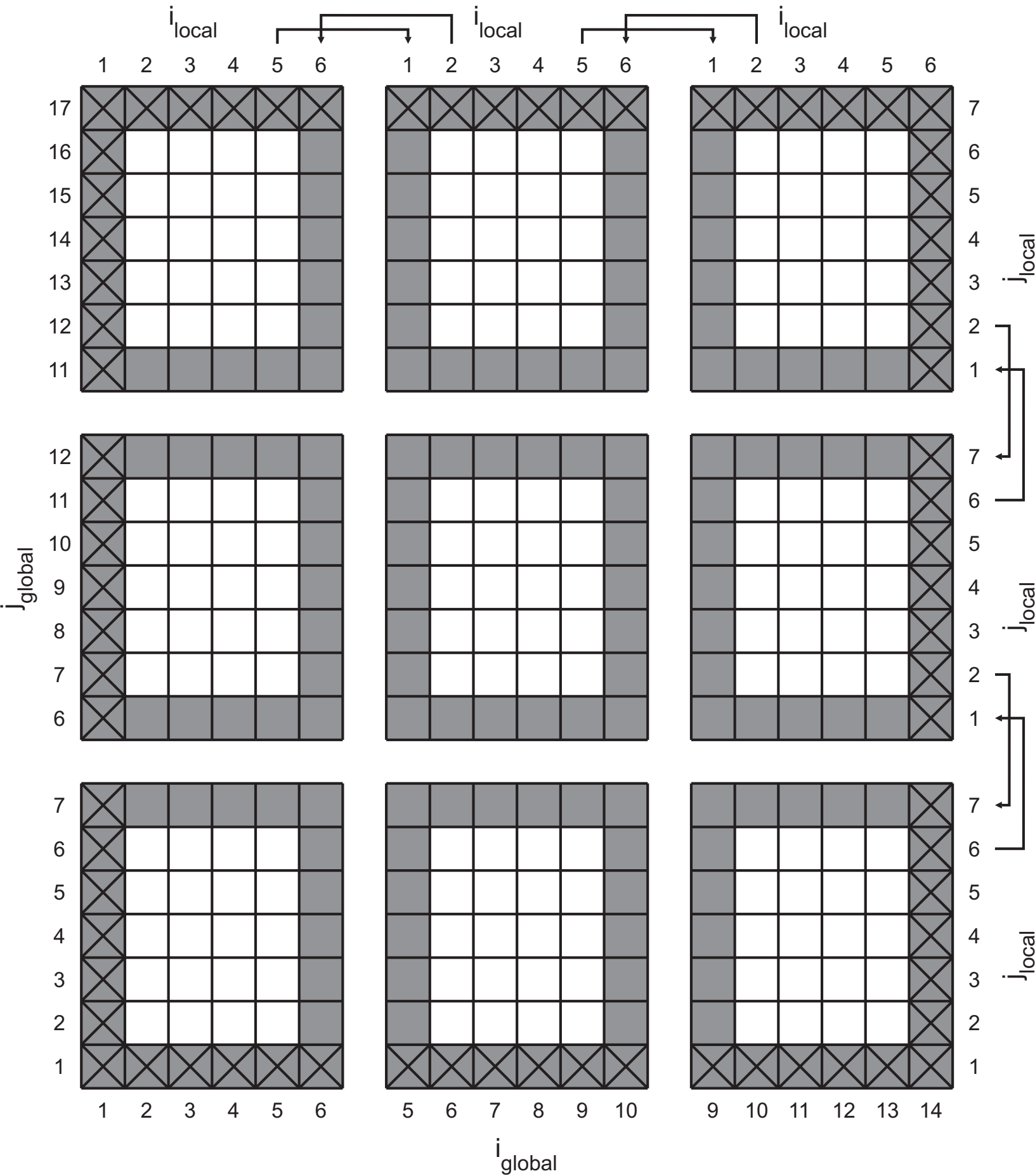
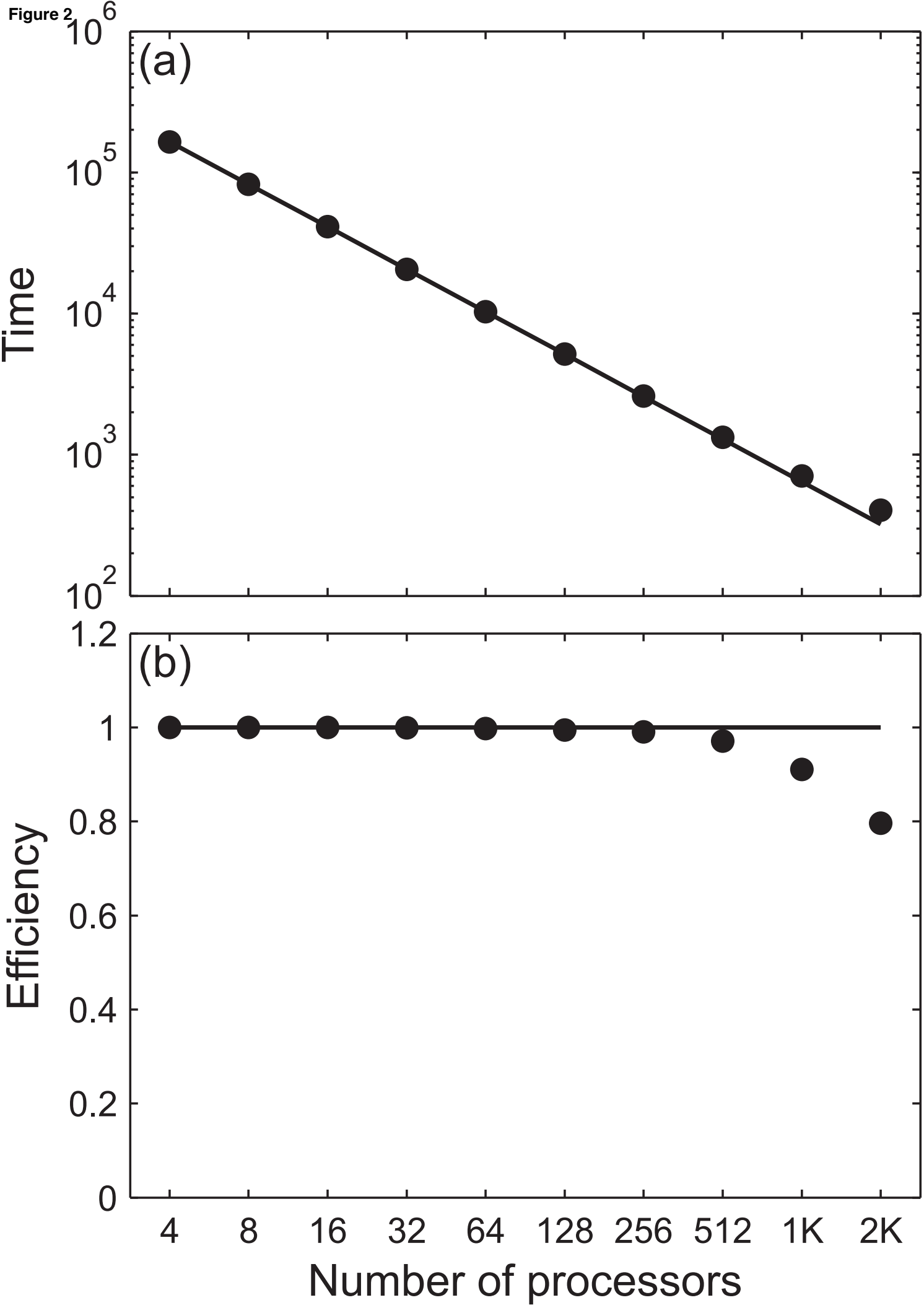

i local
i local
i local
1 2 3 4 5 6
1 2 3 4 5 6
1 2 3 4 5 6
17 16 15 14 13 12 11
7 6 5 4 3 2 1
12 11 10 9 8 7 6
7 6 5 4 3 2 1
7 6 5 4 3 2 1
7 6 5 4 3 2 1
j local
j local
j local
j global
1 2 3 4 5 6
5 6 7 8 9 10
9 10 11 12 13 14
i global

Figure 2


(a)
Time
10^6
10^5
10^4
10^3
10^2
(b)
Efficiency
1.2
1
0.8
0.6
0.4
0.2
0
4
8
16
32
64
128
256
512
1K
2K
Number of processors