Environmental Modelling & Software 41 (2013) 231–238

Contents lists available at SciVerse ScienceDirect

Environmental Modelling & Software

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

Large-scale, high-resolution agricultural systems modeling using a hybrid approach combining grid computing and parallel processing

Gang Zhao [a,b,*], Brett A. Bryan [b], Darran King [b], Zhongkui Luo [c], Enli Wang [c], Ulrike Bende-Michl [c], Xiaodong Song [b,e], Qiang Yu [a,d]

[a] Institute of Geographic Sciences and Natural Resources Research, Chinese Academy of Sciences, 11A Datun Road, Anwai, Beijing 100101, China
[b] CSIRO Ecosystem Sciences, Waite Campus, Urrbrae, SA 5064, Australia
[c] CSIRO Land and Water, Black Mountain, Canberra, ACT 2601, Australia
[d] Plant Functional Biology and Climate Change Cluster, School of the Environment, University of Technology, Sydney, Broadway 2007 NSW, PO Box 123, Australia
[e] Institute of Urban Environment, Chinese Academy of Sciences, Xiamen 361021, China

ARTICLE INFO

Article history:
Received 17 January 2012
Received in revised form
7 August 2012
Accepted 20 August 2012
Available online 10 September 2012

Keywords:
Grid computing
Parallel programming
APSIM
Agricultural modeling
High-performance computing
Food security
Crop modeling

ABSTRACT

The solution of complex global challenges in the land system, such as food and energy security, requires information on the management of agricultural systems at a high spatial and temporal resolution over continental or global extents. However, computing capacity remains a barrier to large-scale, high-resolution agricultural modeling. To model wheat production, soil carbon, and nitrogen dynamics in Australia's cropping regions at a high resolution, we developed a hybrid computing approach combining parallel processing and grid computing. The hybrid approach distributes tasks across a heterogeneous grid computing pool and fully utilizes all the resources of computers within the pool. We simulated 325 management scenarios (nitrogen application rates and stubble management) at a daily time step over 122 years, for 12,707 climate–soil zones using the Windows-based Agricultural Production Systems SIMulator (APSIM). These simulations would have taken over 30 years on a single computer. Our hybrid high performance computing (HPC) approach completed the modeling within 10.5 days—a speed-up of over 1000 times—with most jobs finishing within the first few days. The approach utilizes existing idle organization-wide computing resources and eliminates the need to translate Windows-based models to other operating systems for implementation on computing clusters. There are however, numerous computing challenges that need to be addressed for the effective use of these techniques and there remain several potential areas for further performance improvement. The results demonstrate the effectiveness of the approach in making high-resolution modeling of agricultural systems possible over continental and global scales.

Software availability

Software: Grid-Parallel-APSIM
Developer: Gang Zhao
Contact address: Gang Zhao, CSIRO Sustainable Ecosystems PMB 2, Waite Campus, Urrbrae, SA 5064, Australia. Gang.Zhao@csiro.au
Software and hardware requirements: APSIM, Windows XP or higher, Condor, python, multi-core computer or CPU cluster
Language: Python 2.7
Availability: The software is free to use for educational and research purposes.

1. Introduction

Agricultural land plays a key part in the global issues of food and energy security (Foley et al., 2011). Pressure on land resources is expected to continue growing in the coming decades as a result of expanding population, changing food consumption patterns, and competition from alternative land uses such as biofuel feedstocks (Foley et al., 2011; Tilman et al., 2009). Risk factors such as climate change will continue to challenge agricultural productivity in the major agricultural regions of the world (Luo et al., 2005a). The response of agricultural systems to these drivers needs to be

* Corresponding author. CSIRO Ecosystem Sciences PMB 2, Waite Campus, Urrbrae, SA 5064, Australia. Tel.: +61 08 8303 8679, fax: +61 08 8303 8582.
E-mail address: Gang.Zhao@csiro.au (G. Zhao).

1364-8152/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.envsoft.2012.08.007

understood and predicted to inform policy for managing land resources and increasing the resilience of the land system to various risk factors (Bryan et al., 2010).

Process-based models such as the Agricultural Production Systems SIMulator (APSIM) (Keating et al., 2003) and Environmental Policy Integrated Climate Model (EPIC) (Liu, 2009; Williams et al., 1989) have been increasingly used to simulate aspects of agricultural systems including yields, soil organic carbon, water use efficiency, nitrogen use efficiency, greenhouse gas emissions, and energy balance (Gaiser et al., 2010; Luo et al., 2011; Paterson and Bryan, in press). Responses of agricultural systems to changes in external drivers such as management and climate have also been predicted (Luo et al., 2005a, 2007). Whilst most of these models have been designed for and used in simulating plant-environment processes at high temporal resolution (e.g. daily time step) at the plot scale, this information is required over large extents to inform policy. Drivers of the agricultural processes of yield and soil carbon—such as soil and climatic conditions—vary across the landscape (Hansen and Jones, 2000; Luo et al., in review). Equally, agricultural management practices need to be assessed for their impact on agricultural systems at a fine level of granularity (i.e. what fertilizer/pesticides to apply, how much and when to apply them, what cultivars to use etc.). The influence of these practices also varies with soil and climatic conditions (Akponikpè et al., 2010; Basso et al., 2010; Goulding et al., 2008; Zhao et al., in review). Thus, accurate representations of agricultural systems for addressing the challenges discussed above require the exploration of a high-dimensional management scenario space (Smit and Skinner, 2002), at high spatial resolution (Bryan et al., 2011; Folberth et al., 2012), over large areas (Wang et al., 2009). However, applying high-resolution spatio-temporal process-based models over large extents presents significant computational challenges (Nichols et al., 2011). Whilst a single plot-based scenario may be completed within just minutes, many hundreds of scenarios may be required over many thousands of spatial units. This cannot be done within an acceptable time period using traditional computing methods.

One way to meet this computational demand is to process the simulations in parallel using high-performance computing approaches (Wang et al., 2011). Cluster and grid computing are the most common approaches. Clusters use a collection of linked, homogenous computers working together as a single system with tasks scheduled through job management software. Nichols et al. (2011) demonstrated the potential of computing clusters for modeling agricultural systems, achieving a 40 times speedup in running 140,000 EPIC simulations concurrently on a Linux-based computing cluster. However, most agricultural models are built for the Windows operating systems rather than Linux which is the most common operating system for large clusters, and non-trivial costs are associated with translating the software to another operating system. Grid computing can offer a viable alternative to clusters for high-throughput computing without the need to translate Windows-based models to another operating system.

Large organizations can have many Windows-based desktop computers connected through high-speed networks which, with significant idle time, commonly operate at only a fraction of their processing potential (Huang and Yang, 2011). A key advantage of grid computing is that it can effectively coordinate loosely coupled, heterogeneous, and geographically dispersed computing resources over multiple administrative domains to achieve a common computing goal (Jeffery, 2007; Schwiegelshohn et al., 2010). Grid computing is highly anticipated by those facing compute-intensive problems with no access to local clusters (Sulis, 2009) and has been demonstrated to achieve significant computation efficiencies in environmental modeling (Fernández-Quiruelas et al., 2011; Mineter et al., 2003; Sulis, 2009).

Grid computing attains significant computing efficiencies through coordinating many idle computers. However, grid computing alone can still fail to utilize the full processing capacity of individual nodes in the grid pool. Although many modern computer workstations are equipped with multi-core CPUs, most grid middleware can only allocate one serial application to each node (Huang and Yang, 2011). Parallel programming methods can run multiple instructions simultaneously and take advantage of multi-core hardware and accelerate processing in proportion to the number of CPU cores (Hillar, 2010; Rouholahnejad et al., 2012). Parallel programming has been frequently demonstrated to improve the computing efficiency of models (Elaine, 2005; von Bloh et al., 2010). Embedding parallel programming techniques into grid computing could also substantially improve the throughput of grid computing.

In this study, we confronted a computing challenge of executing more than four million simulations of the APSIM process-based agricultural systems model. Processing each simulation takes 4 min on a single CPU core. We simulated wheat productivity on a daily time step over 122 years for 325 management scenarios in 12,707 spatial units covering Australia's cereal cropping regions. A hybrid computing approach was developed to distribute tasks to idle computers in a computing grid via the middleware, Condor. The approach also employed parallel processing methods to improve the throughput of grid computing. The agricultural systems modeling context and workflow of the hybrid computing approach is provided in the Methods section. The performance of the hybrid computing approach is presented in the Results section in addition to some illustrative model outputs. We demonstrate the potential for high spatial and temporal resolution modeling of agricultural systems over very large spatial extents for addressing global challenges affecting the land system such as food and energy security.

2. Methods

2.1. Study area and spatial units

A range of biophysical factors related to soil and climate determine the suitability of land for growing wheat (Nicholls, 1997). Suitable conditions prevail on mainland Australia in an area west of the Great Dividing Range from Central Queensland through New South Wales and Victoria and on to South Australia and in the south west of Western Australia. The study area was the area currently under wheat production buffered by 100 km (ABARE, 2006). We divided the study area into climate–soil units (CS units) that are relatively homogenous at the scale of analysis. To define unique climate–soil combinations (CS units), we identified climate domains by *k*-means clustering (Hartigan and Wong, 1979) of relevant climate factors (rainfall; minimum, average and maximum temperature) and overlaid them with a soil type classification layer (Johnston et al., 2003) (Fig. 1). Spatial resolution is a key decision in agricultural systems modeling over large extents (Liu et al., 2007; Luo et al., 2005b; Zwart et al., 2010). Selecting a spatial resolution for modeling depends on the questions asked and the scale and confidence in available input data (Adam et al., 2011; Folberth et al., 2012). Too low a resolution causes homogenization of areas with meaningful spatial variation. Too high a resolution could lead to over fitting, a lack of confidence in input data, and present increased computational demand for little gain in modeling detail. The climate–soil domain separation method presented here used irregular polygons to define basic modeling units thereby eliminating the potential redundancies of using raster tessellation (Liu, 2009; Reidsma et al., 2009). This captured the spatial resolution contained in soil and climate data whilst minimizing the number of spatial units and resultant computational load.

2.2. Agricultural systems modeling

APSIM simulates biophysical processes in farming systems (Keating et al., 2003). We used APSIM version 7.3, including its Wheat, SoilN, SoilWater, and Surface Organic Matter (SurfaceOM) modules to model agricultural systems in our study area. The APSIM-Wheat module simulates the growth and development of a wheat crop on a daily time step in response to weather (radiation, temperature), soil water, nitrogen, and residue and crop management. The modules of SurfaceOM and SoilN simulate the decomposition of surface organic matter and the dynamics of carbon

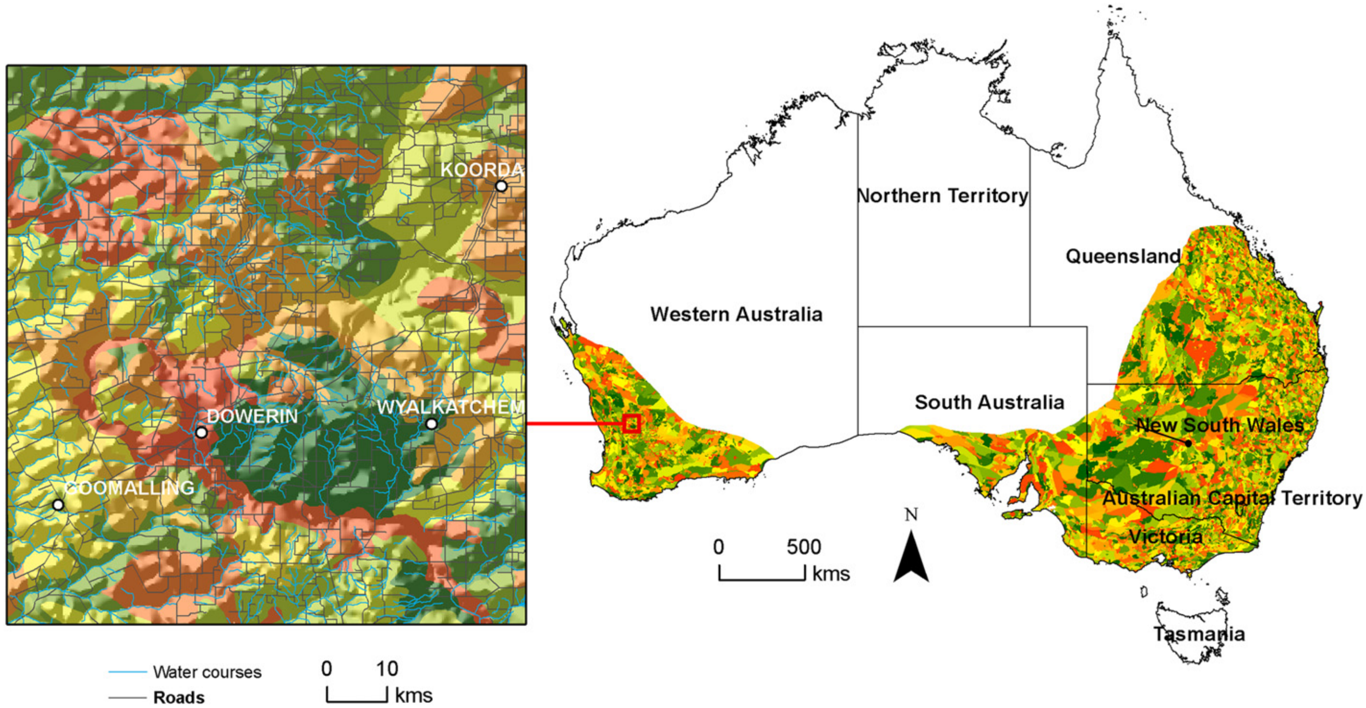

Fig. 1. Study area and climate–soil units. Each CS unit is demonstrated by one unique color. The cropping area in Tasmania was not included in this study. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

and nitrogen in the soil. The SoilWater module calculates soil water balance. The APSIM-Wheat module has been widely validated against field and experimental data under different climate and soil conditions across Australia and in China (Asseng et al., 1998; Chen et al., 2010b). Numerous studies have applied APSIM in optimizing agricultural practices (Chen et al., 2010a; Sadras and Rodriguez, 2010), evaluating the risk of wheat production under climate change scenarios (Ludwig and Asseng, 2006; Luo et al., 2007, 2005c), and in assessing land use and management options (Bryan et al., 2011, 2008).

APSIM requires detailed climate and soil data. Soil data was sourced from the Australian Soil Resource Information System (ASRIS). We used the Level 4 data which provides information on 20 soil properties at five depths. Continental-scale daily climate data layers at 0.05° spatial resolution and spanning 122 years (Jeffrey et al., 2001) were sourced from the Australian Bureau of Meteorology (http://www.bom.gov.au). The daily maximum and minimum temperature, total solar radiation, rainfall, and evaporation were averaged for each CS unit using these daily climate layers (Zhao et al., in press). A soil parameter file and climate data files for each CS unit were stored on a data server (Fig. 2).

To simulate the detailed effects of varying fertilizer, residue, and tillage management on wheat productivity, soil carbon, and nitrogen dynamics, we simulated nitrogen application at 13 levels (0–300 kg ha^{-1} in 25 kg ha^{-1} increments), residue removal rates at 5 levels (0%–100% in 25% increments), and residue incorporation rates at 5 levels (0%–100% in 25% increments). This totaled 325 (13 × 5 × 5) management scenarios for every CS unit.

2.3. Grid computing and workload management

2.3.1. Grid computing with Condor

Grid computing middleware (e.g. ARC, Condor, gLite, Globus) can utilize the idle time of computing resources in organizations by coordinating heterogeneous machines as nodes on a computing grid and allow users to submit requests to execute applications. Condor, developed at the University of Wisconsin–Madison, is a mature, open source workload management system (Litzkow et al., 1988). Condor schedules the timing of jobs and allocates nodes based upon a predefined scheduling strategy and priority scheme. Job status monitoring is enabled in real-time through the regular logging of job outputs, errors, and processing progress. Condor facilitates high-throughput computing using idle desktop machines in a process called *cycle harvesting*. Desktop computer users always have priority of use on their own machines such that Condor processes will be evicted on detection of user activity.

Condor has several *universe* options for running applications in different environments. Condor's standard universe provides checkpointing where jobs can be submitted to a different node and continued following an interruption using a checkpoint image—but it does not allow parallel processing. We used the vanilla universe which allows parallel processing but does not support checkpointing, leaving interrupted jobs to be restarted from the beginning on another node in the pool (Condor Team, 2011).

CSIRO's Condor grid computing pool which contains more than 5000 machines was used to run APSIM simulations in this study. There are typically around 1800 idle machines during weekends and 2000 machines on weeknights. Jobs are

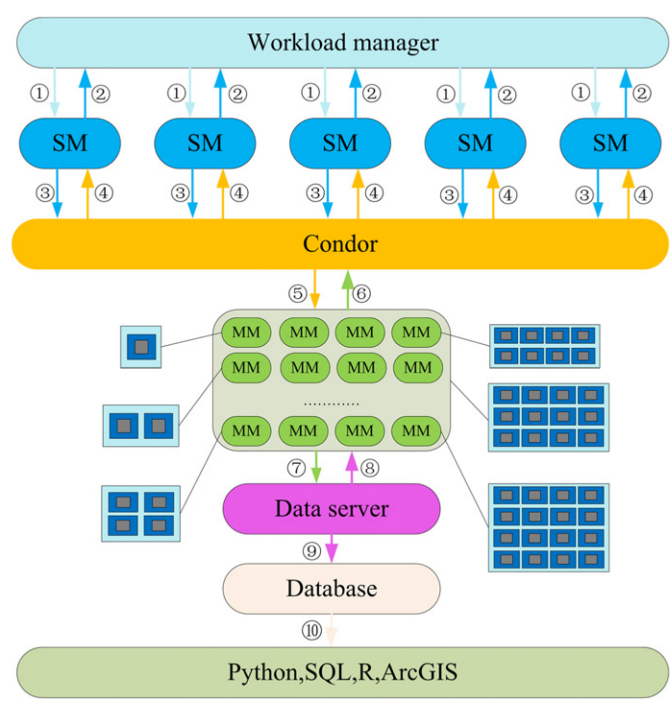

Fig. 2. Structure of the hybrid computing approach, SM (submitting machine), MM (multi-core machine). ① Copy and run monitoring application to submitting machines; ② gather logs from submitting machines to workload manager; ③ job description producing and submitting; ④ Condor logging to submitting machines; ⑤ jobs scheduling and data transferring; ⑥ communication between processing nodes and Condor; ⑦ transfer results back to data server; ⑧ transfer APSIM and input data to processing nodes; ⑨ import results into database; ⑩ results analysis and visualization.

normally permitted to run at anytime during weekends but are restricted to the hours between 6:00pm and 8:00am on weeknights.

The structure of the Condor submission is illustrated in Fig. 2 which demonstrates the relationships between the workload manager, submitting machines, Condor, multi-core grid nodes, data server, relational database, and data analysis tools. These elements are described in detail below.

2.3.2. Task partitioning

There is a workload balance challenge to be addressed when partitioning and assigning simulations to grid nodes and CPU cores within each node. Task partitioning influences the balance between job size and execution efficiency. Partitioning a smaller number of larger jobs increases the risk of interruption from user activity in a grid computing environment. Reducing the job size increases the number of jobs which, in turn, increases the data transfer overheads and the difficulty of job management with restrictions on the maximum number of concurrent jobs (400) from each submitting machine.

In this study, considering the heterogeneous multi-node, multi-core computing environment and the average run time of each simulation (about 4 min), the simulations were partitioned at two levels: at the CS unit level and, within a single CS unit, at the management scenario simulation level. To minimize the amount of data transfer, a single CS unit with unique climate and soil data including all management scenarios was assigned to a single node in the Condor pool. As the 325 management scenarios for a single CS unit require no inter-process communication, we parallelized the simulations to take advantage of multi-core nodes in the Condor pool.

2.3.3. Workload management

Condor defines the run time environment using a job submission description file which comprises a combination of commands and keywords (Fig. 3, left). These specify the universe, arguments, node requirements, file transfer policy, and other characteristics of the job. To overcome the limited bandwidth of the submitting machines, the program and data were stored on a high bandwidth server (Fig. 2). We transferred only a small batch file to each node (Fig. 3, right) which connects to the server, downloads both the application and data, runs the simulations, and uploads the results.

To overcome the operating system limit of 400 concurrent jobs per submitting machine we submitted the 12,707 jobs using five submit machines (Fig. 2), thereby running 2000 jobs concurrently. Job description files were produced by the submitting machine. The submitting machine starts a process for each job to update and log its state. In this way, Condor logs the progress of each individual job thereby enabling job failure detection, diagnosis, and resubmission. Nodes with Windows 7 were excluded in this study as the Condor version used could not interrupt multiprocessing jobs on detection of user activity. The job submission workflow is described in detail in Fig. 4.

A fault-tolerant job management script was developed to coordinate the five submitting machines. The script monitored and checked the state of all jobs, quickly released suspended jobs, and resubmitted failed jobs automatically (Fig. 4). The script compared the Condor queue with the finished and unfinished job list. If jobs had not finished and were not in the Condor queue, it submitted them; if jobs were finished but were still listed in the queue, it removed the jobs from the queue. In order to avoid jobs languishing on slower nodes when faster machines were available, the script removed jobs which had run for more than 24 h.

2.3.4. Results management and analysis

In each growing season, there were outputs at three stages: sowing, end of year, and harvesting. Each output line contains 60 fields. Thus, over 90 billion (60*3*122*325*12,707) records were transferred back to the data server. A Microsoft SQL Server 2008 R2 (64-bit) database was used to manage and analyze these model outputs. A Python script was developed to import the APSIM outputs for each CS unit (50 MB) using the pyodbc package with bulk insert. Further validation, statistical analysis, and visualization were conducted with Matplotlib, Numpy, Scipy, R, and ArcGIS (Fig. 2).

2.4. Full utilization of multi-core nodes with parallel processing

APSIM simulations were run in parallel on multi-core grid pool nodes using a Python script. The script produces APSIM project files, processes the simulations in parallel, and collects the results. Each APSIM simulation was specified using an XML project file which contains parameters. In the script, the master process first constructs the 325 management scenario simulations for the CS unit using an APSIM project file template and queues them in a list. It discovers the number of CPU cores on the node and sets up a pool of worker processors. The script then uses the Multiprocessing package for the efficient and load-balanced processing of the 325 management scenarios using APSIM. When the simulations have finished, the master process collects the results (Fig. 4). In order to run the script and the APSIM model on a heterogeneous computing environment where nodes probably don't have APSIM or Python installed, the script and dependent libraries were compiled to a system-independent executable application using Py2exe.

3. Results

3.1. Condor job execution statistics

The total computing time taken to complete all simulations was 10.5 days. The number of concurrently-running jobs peaked Saturday night after submission earlier that afternoon (Fig. 5). There were several processing troughs during working hours on weekdays and a long tail at the end before all jobs were finished.

Compared to the average computing time of single-core nodes, the average speedup ratio of multi-core machines was proportional to the number of CPU cores in grid nodes (Fig. 6). There is an exception that the 16-core nodes were slower than the 12-core nodes. Several simulations on both 2-core and 4-core nodes were very slow as shown by the outliers in Fig. 6.

Only about 30% of the Condor pool was utilized in our simulations (Table 1). Around 87% of jobs were completed by 2 and 4-core nodes. Whilst 8, 12, and 16-core nodes formed a very small proportion of the Condor pool, due to the exploitation of their multiple CPU cores by the parallel processing script, they were able to complete a disproportionately high number of simulations.

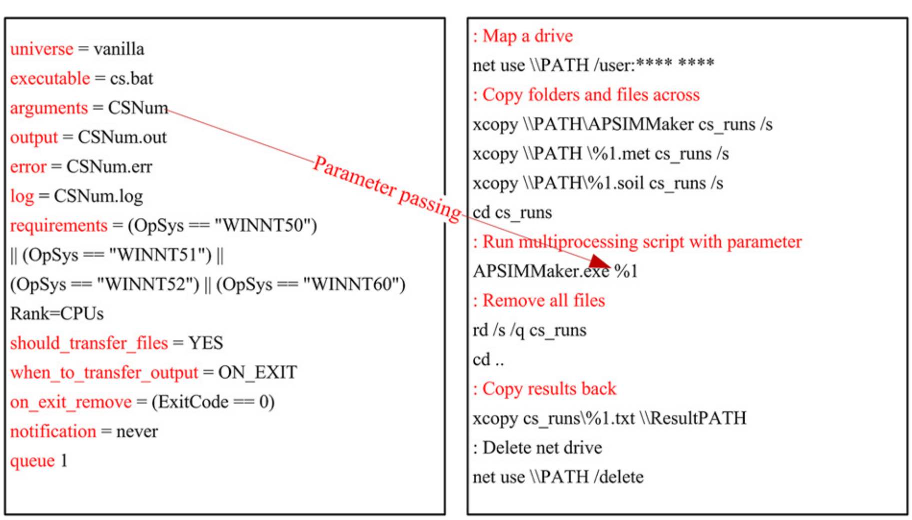

Fig. 3. Condor job submission description file (left) and batch file for transferring input data and application (right). APSIMMaker is the multiprocessing application mentioned following, the batch file accepts the CS unit number parameter from the job description file.

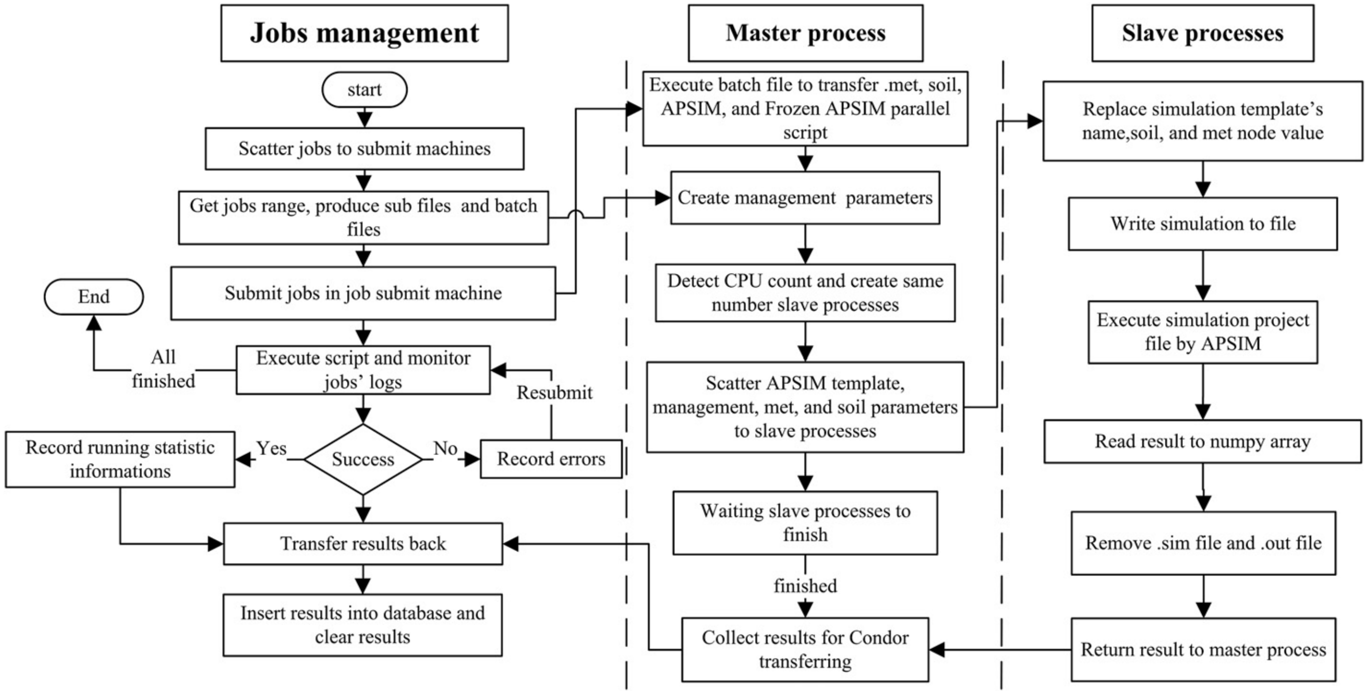

Fig. 4. Workflow of job management, task partitioning and parallel programming. The left part is the job management at the submitting machine side, middle and right part is parallel processing part realized by python Multiprocessing packages.

3.2. Preliminary model results

Deep analysis of the agricultural modeling outputs is not within the scope of this paper and only illustrative results are shown here. The great benefit of our computing approach is that we can begin to understand the spatial and temporal variability in many aspects of agricultural systems at resolutions that were previously impossible. To illustrate this, maps of the spatial distribution of crop yield, the impact of management, and the inter-annual variability are presented (Fig. 7).

4. Discussion

To inform decisions on the policy and management of agricultural systems, detailed spatio-temporal information is required over large extents. Our hybrid approach, combining grid computing and parallel processing, enabled the modeling of agricultural systems at a level of spatial and temporal detail never before assessed over a continental extent. Extrapolating the results of the processing time suggests that on a single CPU core (21.1 h for one CS unit, Table 1), modeling the management of agricultural systems at

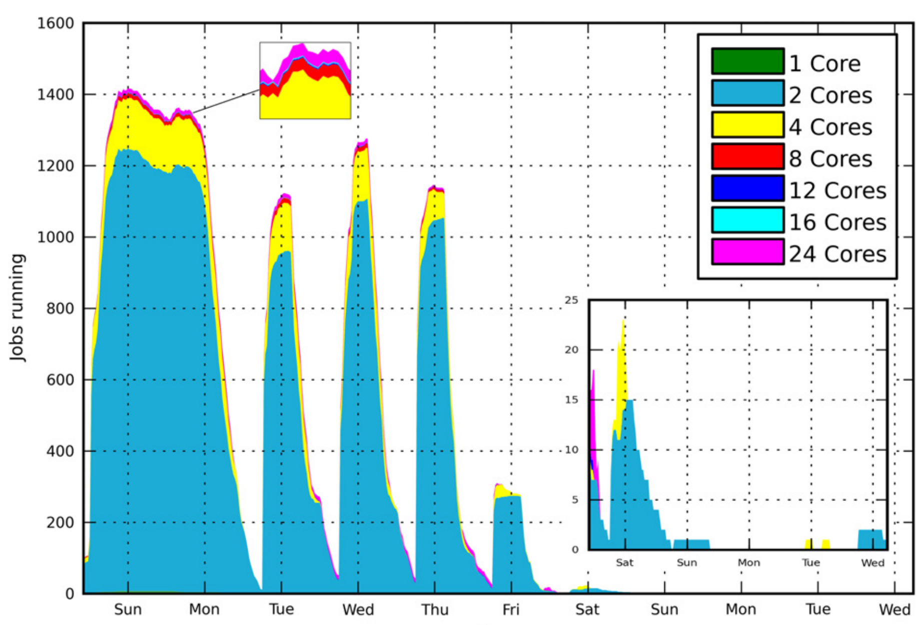

Fig. 5. Number of running jobs and different types of machines over time. The small inset graph shows the long tail in run time.

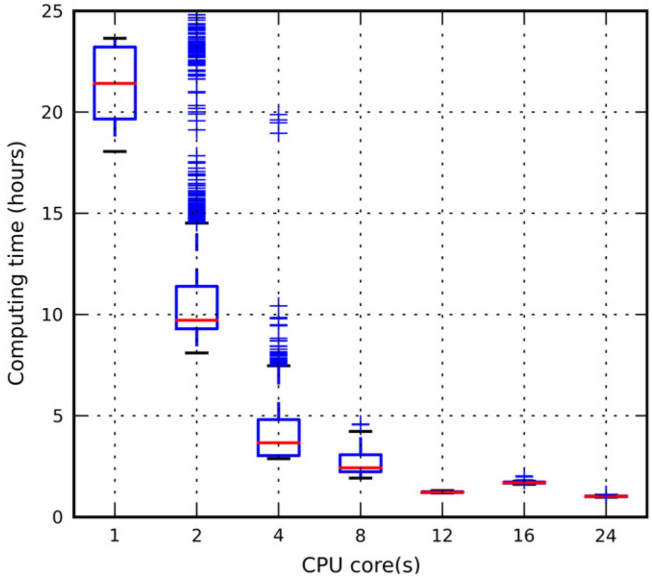

Fig. 6. Variation of computing times of machines in Condor pool with different number of CPU cores.

the spatial and temporal resolution undertaken in this study would have taken more than 30 years. Our hybrid approach completed the modeling in 10.5 days. To achieve the equivalent computing performance on a Windows cluster would require 93 × 12-core compute nodes, assuming we could get sole access to the entire cluster. Windows clusters of this size are uncommon and clusters are usually shared—so processing times would depend on demand. Even so, occupying the cluster for such a long time is usually not realistic especially when the simulations need to be run several times.

Beyond increased processing efficiencies, the key advantage of grid computing in organizations is that it can offer substantial computational capacity for running Windows-based models. The size of this computational resource can be many times greater than available Windows-based clusters. APSIM was developed on and for the Windows platform. Like many other environmental models, APSIM has a complicated structure combining modules written in several different languages, including Windows-specific languages such as .NET. Hence, to port or compile it for other operating systems such as Linux, for running on large computing clusters is a significant undertaking. Further, these models are often constantly updated. Maintaining a version of a complex model in another language requires ongoing effort. Our hybrid computing approach eliminates the need to translate and maintain versions of the models (including incorporating updates) running under other operating systems.

Further enhancements to processing efficiency are possible, particularly through reducing the long tail in job processing time or

Table 1
Statistics of jobs in Condor and Condor pool.

CPU core(s)	Finished jobs	Percentage of total jobs	Distinct machines	Machines in pool	Average time (h)	Clock speed (MHz)	
						Min	Max
1	15	0.11	8	149	21.1	2327	2992
2	8548	67.27	1773	4180	10.5	1994	4094
4	2558	20.13	181	518	4.1	1994	3093
8	349	2.74	16	185	2.6	2261	3392
12	74	0.58	1	47	1.2	2660	2660
16	39	0.31	1	29	1.7	2527	2527
24	1124	8.85	12	25	1.02	2660	2660

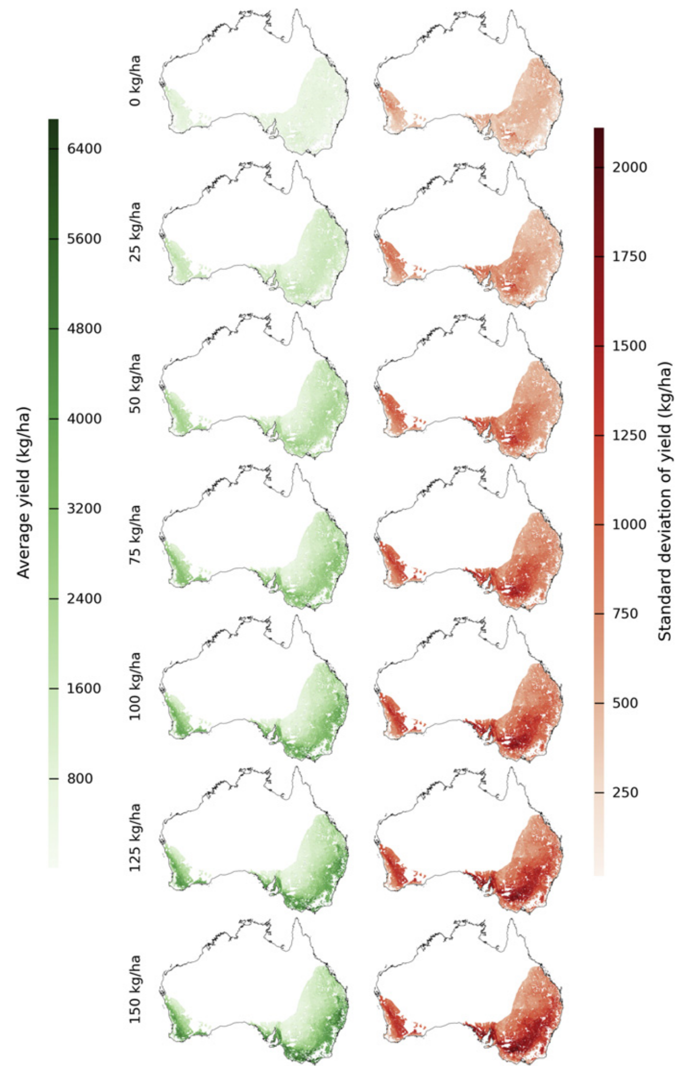

Fig. 7. Average (left) and standard deviation (right) of yield at nitrogen applying levels from 0 to 150 kg ha^{-1}. The areas of remnant vegetation are excluded in the results.

enhancing the computing efficiency of APSIM application. A large proportion of the APSIM simulations were completed over the first weekend. However, most nodes were in use on weekdays, there were several troughs during working hours. As the average processing time of dual-core machines was 10.5 h per job, jobs running overnight could be evicted the next morning prior to finishing. The long tail was caused by a few jobs which were allocated to slower dual-core machines failing several times during weekday processing. The ability to rank machines by the number of CPU cores instead of the number of CPUs could shorten the run time by speeding processing time and increasing the chance of success of jobs in the tail. Smaller job size partitioning is another potential solution for the long tail. Smaller jobs need shorter run times which will reduce the risk of eviction of jobs that have been running on slower nodes. However, this increases the complexity of task partitioning, and time spent in data transmission, job management, and result collection. The ability to access nodes running Windows 7 would provide access to many of the newer, higher-performance pool machines and further increase efficiency. Finally, 4 min for one daily simulation over 122 years is a relatively long execution time. If the efficiency of the APSIM model could be improved, the total computing time could be further reduced.

Whilst the hybrid approach dramatically improved the processing efficiency, there are also several limitations. Firstly, software cannot be installed on nodes through grid computing middleware. Although the middleware can choose machines with specific software installed, this restricts the pool to those nodes with the particular software. This also reduces the flexibility of the computing resource. In order to maximize the processing capacity of the computing grid, developing applications and models that work under diverse hardware and software environments is a key challenge. Secondly, grid computing is generally only suitable for tasks which are run many times with different input data and/or parameters and which require no inter-process communication, such as the agricultural systems simulation presented here. These jobs are commonly called embarrassingly parallel (Agullo et al., 2011). For tasks requiring communication between processes, message-passing programming like the Message Passing Interface (MPI) in a cluster environment is more appropriate option than grid computing (Gropp et al., 1999; Keller et al., 2003). Lastly, with simple models that need to be run many times, Graphics Processing Units (GPU) may be a better option than grid computing (Bryan, in press) as the job coordination and data distribution overheads may limit the efficiencies gained through grid computing.

Nevertheless, the hybrid computing framework combining grid computing and parallel processing improved the computational efficiency and made high resolution simulation of agricultural systems possible at a national scale. This can better support the analysis of global challenges of land use, food security, soil carbon, and energy use in agriculture.

Acknowledgments

This work was supported by the National Basic Research Program of China (Grant No. 2012CB955304), CSIRO's Integrated Carbon Pathways (ICP) initiative and Sustainable Agriculture Flagship. We thank David Jacquier of CSIRO Land and Water, Canberra for providing soil texture data. We also thank Greg Hitchen of CSIRO Information Management & Technology for technical support of Condor job submitting and management. Efforts and comments from Andrew Moore, Neville Crossman, and three anonymous reviewers greatly improved the manuscript.

References

ABARE, 2006. Australian Crop Report. Australian Bureau of Agricultural and Resource Economics and Sciences.

Adam, M., Van Bussel, L.G.J., Leffelaar, P.A., Van Keulen, H., Ewert, F., 2011. Effects of modelling detail on simulated potential crop yields under a wide range of climatic conditions. Ecological Modelling 222 (1), 131–143.

Agullo, E., Coti, C., Herault, T., Langou, J., Peyronnet, S., Rezmerita, A., Cappello, F., Dongarra, J., 2011. QCG-OMPI: MPI applications on grids. Future Generation Computer Systems 27 (4), 357–369.

Akponikpè, P.B.I., Gérard, B., Michels, K., Bielders, C., 2010. Use of the APSIM model in long term simulation to support decision making regarding nitrogen management for pearl millet in the Sahel. European Journal of Agronomy 32 (2), 144–154.

Asseng, S., Keating, B.A., Fillery, I.R.P., Gregory, P.J., Bowden, J.W., Turner, N.C., Palta, J.A., Abrecht, D.G., 1998. Performance of the APSIM-wheat model in western Australia. Field Crops Research 57 (2), 163–179.

Basso, B., Cammarano, D., Troccoli, A., Chen, D., Ritchie, J.T., 2010. Long-term wheat response to nitrogen in a rainfed Mediterranean environment: field data and simulation analysis. European Journal of Agronomy 33 (2), 132–138.

Bryan, B.A., 2012. High-performance computing tools for the integrated assessment and modelling of social-ecological systems. Environmental Modeling & Software. http://dx.doi.org/10.1016/j.envsoft.2012.02.006.

Bryan, B.A., King, D., Wang, E., 2010. Potential of woody biomass production for motivating widespread natural resource management under climate change. Land Use Policy 27 (3), 713–725.

Bryan, B.A., King, D., Ward, J.R., 2011. Modelling and mapping agricultural opportunity costs to guide landscape planning for natural resource management. Ecological Indicators 11 (1), 199–208.

Bryan, B.A., Ward, J., Hobbs, T., 2008. An assessment of the economic and environmental potential of biomass production in an agricultural region. Land Use Policy 25 (4), 533–549.

Chen, C., Wang, E., Yu, Q., 2010a. Modelling the effects of climate variability and water management on crop water productivity and water balance in the North China Plain. Agricultural Water Management 97 (8), 1175–1184.

Chen, C., Wang, E.L., Yu, Q., Zhang, Y.Q., 2010b. Quantifying the effects of climate trends in the past 43 years (1961–2003) on crop growth and water demand in the North China Plain. Climatic Change 100 (3–4), 559–578.

Elaine, W., 2005. Eclpss: a java-based framework for parallel ecosystem simulation and modeling. Environmental Modelling & Software 20 (9), 1081–1100.

Fernández-Quiruelas, V., Fernández, J., Cofiño, A.S., Fita, L., Gutiérrez, J.M., 2011. Benefits and requirements of grid computing for climate applications. An example with the community atmospheric model. Environmental Modelling & Software 26 (9), 1057–1069.

Folberth, C., Yang, H., Wang, X., Abbaspour, K.C., 2012. Impact of input data resolution and extent of harvested areas on crop yield estimates in large-scale agricultural modeling for maize in the USA. Ecological Modelling 235–236 (0), 8–18.

Foley, J.A., Ramankutty, N., Brauman, K.A., Cassidy, E.S., Gerber, J.S., Johnston, M., Mueller, N.D., O/'Connell, C., Ray, D.K., West, P.C., Balzer, C., Bennett, E.M., Carpenter, S.R., Hill, J., Monfreda, C., Polasky, S., Rockstrom, J., Sheehan, J., Siebert, S., Tilman, D., Zaks, D.P.M., 2011. Solutions for a cultivated planet. Nature 478 (7369), 337–342.

Gaiser, T., de Barros, I., Sereke, F., Lange, F.-M., 2010. Validation and reliability of the EPIC model to simulate maize production in small-holder farming systems in tropical sub-humid West Africa and semi-arid Brazil. Agriculture, Ecosystems & Environment 135 (4), 318–327.

Goulding, K., Jarvis, S., Whitmore, A., 2008. Optimizing nutrient management for farm systems. Philosophical Transactions of the Royal Society B: Biological Sciences 363 (1491), 667–680.

Gropp, W., Lusk, E., Skjellum, A., 1999. Using MPI: Portable Parallel Programming with the Message-passing Interface, second ed. The MIT Press.

Hansen, J.W., Jones, J.W., 2000. Scaling-up crop models for climate variability applications. Agricultural Systems 65 (1), 43–72.

Hartigan, J.A., Wong, M.A., 1979. Algorithm AS 136: a *k*-means clustering algorithm. Journal of the Royal Statistical Society Series C: Applied Statistics 28 (1), 100–108.

Hillar, G., 2010. Professional Parallel Programming with C#: Master Parallel Extensions With.NET 4. Wrox.

Huang, Q., Yang, C., 2011. Optimizing grid computing configuration and scheduling for geospatial analysis: an example with interpolating DEM. Computers & Geosciences 37 (2), 165–176.

Jeffery, K.G., 2007. Next generation GRIDs for environmental science. Environmental Modelling & Software 22 (3), 281–287.

Jeffrey, S.J., Carter, J.O., Moodie, K.B., Beswick, A.R., 2001. Using spatial interpolation to construct a comprehensive archive of Australian climate data. Environmental Modelling & Software 16 (4), 309–330.

Johnston, R., Bui, E., Simon, D., Carlile, P., Henderson, B., Imhoff, M., Howe, D., Schoknecht, N., Powell, B., Bleys, E., 2003. ASRIS: the database. Australian Journal of Soil Research 41 (6), 1021–1036.

Keating, B.A., Carberry, P.S., Hammer, G.L., Probert, M.E., Robertson, M.J., Holzworth, D., Huth, N.I., Hargreaves, J.N.G., Meinke, H., Hochman, Z., McLean, G., Verburg, K., Snow, V., Dimes, J.P., Silburn, M., Wang, E., Brown, S., Bristow, K.L., Asseng, S., Chapman, S., McCown, R.L., Freebairn, D.M., Smith, C.J., 2003. An overview of APSIM, a model designed for farming systems simulation. European Journal of Agronomy 18 (3–4), 267–288.

Keller, R., Gabriel, E., Krammer, B., Müller, M.S., Resch, M.M., 2003. Towards efficient execution of MPI applications on the grid: porting and optimization issues. Journal of Grid Computing 1 (2), 133–149.

Litzkow, M.J., Livny, M., Mutka, M.W., 1988. Condor-a Hunter of Idle Workstations. IEEE, pp. 104–111.

Liu, J., 2009. A GIS-based tool for modelling large-scale crop-water relations. Environmental Modelling & Software 24 (3), 411–422.

Liu, J., Williams, J.R., Zehnder, A.J.B., Yang, H., 2007. GEPIC – modelling wheat yield and crop water productivity with high resolution on a global scale. Agricultural Systems 94 (2), 478–493.

Ludwig, F., Asseng, S., 2006. Climate change impacts on wheat production in a Mediterranean environment in Western Australia. Agricultural Systems 90 (1–3), 159–179.

Luo, Q., Bellotti, W., Williams, M., Bryan, B., 2005a. Potential impact of climate change on wheat yield in South Australia. Agricultural and Forest Meteorology 132 (3–4), 273–285.

Luo, Q., Bryan, B., Bellotti, W., Williams, M., 2005b. Spatial analysis of environmental change impacts on wheat production in Mid-Lower North, South Australia. Climatic Change 72 (1), 213–228.

Luo, Q.Y., Bellotti, W., Williams, M., Cooper, I., Bryan, B., 2007. Risk analysis of possible impacts of climate change on South Australian wheat production. Climatic Change 85 (1–2), 89–101.

Luo, Q.Y., Jones, R.N., Williams, M., Bryan, B., Bellotti, W., 2005c. Probabilistic distributions of regional climate change and their application in risk analysis of wheat production. Climate Research 29 (1), 41–52.

Luo, Z., Wang, E., Bryan, B.A., King, D., Zhao, G., Pan, X., Bende-Michl, U. Metamodeling soil organic carbon sequestration potential and its application at regional scale. Manuscript in review.

Luo, Z., Wang, E., Sun, O.J., Smith, C.J., Probert, M.E., 2011. Modeling long-term soil carbon dynamics and sequestration potential in semi-arid agro-ecosystems. Agricultural and Forest Meteorology 151 (12), 1529–1544.

Mineter, M.J., Jarvis, C.H., Dowers, S., 2003. From stand-alone programs towards grid-aware services and components: a case study in agricultural modelling with interpolated climate data. Environmental Modelling & Software 18 (4), 379–391.

Nicholls, N., 1997. Increased Australian wheat yield due to recent climate trends. Nature 387 (6632), 484–485.

Nichols, J., Kang, S., Post, W., Wang, D., Bandaru, V., Manowitz, D., Zhang, X., Izaurralde, R., 2011. HPC-EPIC for high resolution simulations of environmental and sustainability assessment. Computers and Electronics in Agriculture 79 (2), 112–115.

Paterson, S.E., Bryan, B.A., 2012. Food-carbon trade-offs between agriculture and reforestation land uses under alternative market-based policies. Ecology & Society 17 (3), 21.

Reidsma, P., Ewert, F., Boogaard, H., Diepen, K.v, 2009. Regional crop modelling in Europe: the impact of climatic conditions and farm characteristics on maize yields. Agricultural Systems 100 (1–3), 51–60.

Rouholahnejad, E., Abbaspour, K.C., Vejdani, M., Srinivasan, R., Schulin, R., Lehmann, A., 2012. A parallelization framework for calibration of hydrological models. Environmental Modelling & Software 31 (0), 28–36.

Sadras, V.O., Rodriguez, D., 2010. Modelling the nitrogen-driven trade-off between nitrogen utilisation efficiency and water use efficiency of wheat in eastern Australia. Field Crops Research 118 (3), 297–305.

Schwiegelshohn, U., Badia, R.M., Bubak, M., Danelutto, M., Dustdar, S., Gagliardi, F., Geiger, A., Hluchy, L., Kranzlmüller, D., Laure, E., Priol, T., Reinefeld, A., Resch, M., Reuter, A., Rienhoff, O., Rüter, T., Sloot, P., Talia, D., Ullmann, K., Yahyapour, R., von Voigt, G., 2010. Perspectives on grid computing. Future Generation Computer Systems 26 (8), 1104–1115.

Smit, B., Skinner, M.W., 2002. Adaptation options in agriculture to climate change: a typology. Mitigation and Adaptation Strategies for Global Change 7 (1), 85–114.

Sulis, A., 2009. GRID computing approach for multireservoir operating rules with uncertainty. Environmental Modelling & Software 24 (7), 859–864.

Team, C., 2011. University of Wisconsin–Madison, Condor Version 7.6. 0 Manual.

Tilman, D., Socolow, R., Foley, J.A., Hill, J., Larson, E., Lynd, L., Pacala, S., Reilly, J., Searchinger, T., Somerville, C., Williams, R., 2009. Beneficial biofuels—the food, energy, and environment trilemma. Science 325 (5938), 270–271.

von Bloh, W., Rost, S., Gerten, D., Lucht, W., 2010. Efficient parallelization of a dynamic global vegetation model with river routing. Environmental Modelling & Software 25 (6), 685–690.

Wang, E., Cresswell, H., Bryan, B., Glover, M., King, D., 2009. Modelling farming systems performance at catchment and regional scales to support natural resource management. NJAS – Wageningen Journal of Life Sciences 57 (1), 101–108.

Wang, H., Fu, X., Wang, G., Li, T., Gao, J., 2011. A common parallel computing framework for modeling hydrological processes of river basins. Parallel Computing 37 (6–7), 302–315.

Williams, J., Jones, C., Kiniry, J., Spanel, D.A., 1989. The EPIC crop growth model. Transactions of the ASAE 32, 497–512.

Zhao, G., Bryan, B.A., King, D., Luo, Z., Wang, E., Song, X., Yu, Q. Impact of agricultural management practices on soil organic carbon dynamics in a continuous wheat cropping system of Australian croplands. Manuscript in review.

Zhao, G., Bryan, B.A., King, D., Song, X., Yu, Q. Parallelization and optimization of spatial analysis for large scale environmental model data assembly. Computers and Electronics in Agriculture, in press.

Zwart, S.J., Bastiaanssen, W.G.M., de Fraiture, C., Molden, D.J., 2010. A global benchmark map of water productivity for rainfed and irrigated wheat. Agricultural Water Management 97 (10), 1617–1627.