Proceedings of the 2005 International Conference on Simulation and Modeling
V. Kachitvichyanukul, U. Purintrapiban, and P. Utayopas, eds.

INPUT ASSIMILATION OF SOIL WATER ATMOSPHERE AND PLANT (SWAP) MODEL WITH GA USING CLUSTER COMPUTERS

Shamim Akhter
CSIM Program
School of Advanced Technology
Asian Institute of Technology
Pathumthani, Thailand 12120

Kiyoshi Honda
STAR Program
School of Advanced Technology
Asian Institute of Technology
Pathumthani, Thailand 12120

Yann Chemin
STAR Program
School of Advanced Technology
Asian Institute of Technology
Pathumthani, Thailand 12120

ABSTRACT

A method is proposed to optimize the input parameters of a one-dimensional crop model (SWAP) by assimilating simulated evapotranspiration with remote sensing data. The optimization is based on GA (Genetic Algorithm). However it requires huge computation time, which is one of the constraints for practical implementation of the method. This paper addresses the implementation of a GA optimization on SWAP model using cluster computers to reduce the time constraint. The most time consuming component in the process is the evaluation of the chromosomes, which is the running of SWAP for crop growth simulation, including water transport in soil column. This simulation needs to be calculated several times per generation of chromosomes. By distributing this evaluation process to multiple CPUs, the optimization time is reduced. A numerical experiment is presented in this paper to show the strengths and limitations of the proposed approach using 100 Mbps http://optima.ait.ac.th Cluster Computers.

1 INTRODUCTION

Running a crop growth model is useful for agriculture monitoring, prediction and management. A simulation model SWAP has been utilized for these purposes. SWAP (Soil, Water, Atmosphere, and Plant) is equipped with crop models and water management modules where the growth and development of a crop can be simulated under different climatic and environmental conditions (Van Dam et al. 1997).

One problem to run SWAP model is parameter identification, especially when the target area is large such as provincial or country level. Remote Sensing (RS) provides us with useful information over large area. RS cannot observe input parameters of SWAP directly, however, a method to estimate input parameters of SWAP from RS using data assimilation has been proposed (Amor V. M. Ines, 2002). This method optimize input parameters to minimize the difference of two (2) ETa, those are from remote sensing and SWAP model. GA is being used in this optimization process.

Even though the problem with larger RS data is solved, a practical issue arises with the overall calculations time load for assimilating data with remote sensing data. The calculation time for identifying SWAP parameters only for 1 pixel (1 square kilometer) will take 30 minutes and the most time consuming component in the process is the evaluation of the chromosomes, which is the running of SWAP for crop growth simulation, including water transport in soil column. This simulation needs to be calculated several times per generation of chromosomes, totaling several hundred times. Thus, a RS image of 1000 x 1000 square kilometer of 1000x1000 pixels will take more than 50 years (30 min. * 1000 * 1000) and that is not acceptable.

By distributing this evaluation process to multiple CPUs, the optimization time is reduced. A numerical experiment is presented in this paper to show the strengths and limitations of the proposed approach using Optima Cluster Computers.

2 TECHNICAL SPEC OF OPTIMA CLUSTER

Under the assumption that Cluster style computing will remove computational time constraints for SWAP-GA, a parallel one evaluation SWAP-GA procedure for remote sensed images can be considered. The AIT Cluster computer http://optima.ait.ac.th is used for the purpose of experiment.

Frontend: Athlon XP 1800+ 512 Mbyte RAM 4 of 80 Gbyte IDE disk
Compute Node: Athlon Xp 1800+ 512 Mbytes RAM 40 Gbyte IDE Disk
Interconnection: Fast Ethernet Swicth D-link DES1024R
NIC: SMC Fast Ethernet
Software: SCE 1.5, MPICH1.2.5, Pgapack
More information can be found at
http://optima.ait.ac.th/scmsweb/scms_home.html

3 DATA, SWAP MODEL AND GENETIC ALGORITHM

3.1 The ETa Data

The algorithm chosen to calculate evapotranspiration by remote sensing was laid down by Bastiaanssen (1995) and refereferred to as SEBAL (Surface Energy Balance Algorithm for Land). The latent Heat Flux is the resulting partition of the energy-balance at the time of the satellite overpass (typically 10.00-10.30AM for Landsat and 11.00-11.30AM for Modis). Eventually, the Latent Heat Flux (W/m^2) is converted in instantaneous Actual Evapotranspiration (mm). Thereafter, using net radiation information, the algorithm is extending the evapotranspiration to a daily value. The data to be used as target for the optimization process is:

- 46 images of Terra-Modis ETa average for 8 days (mm/day)
- 1 to 3 images of Landsat/Aster ETa for a day (mm/day)

3.2 SWAP Model

SWAP is a one-dimensional transient model (Figure 1) to simulate water flow in a heterogeneous soil-root system, which can be under the influence of groundwater (Feddes et al., 1978; Belmans et al., 1983). The model has been recently modified to include solute transport, heat flow and crop growth in the air-plant-soil environment (Van Dam et al, 1997).

The upper boundary conditions of the system are determined by the potential evapotranspiration rate (ETpot), rainfall and irrigation. SWAP uses daily meteorological data to calculate ETpot according to the Penman-Monteith equation but opts for alternative procedures. The ETpot is used to calculate the potential soil evaporation rate (Epot) and potential transpiration rate (Tpot) according to the leaf area index development. Epot is reduced to actual soil evaporation rate (Eact) by taking the minimum of the evaporation rate, the maximum soil water flux in the top soil according to Darcy assuming a minimum allowed pressure head in the atmosphere, or an empirical reduction function (Black et al., 1969; Boesten and Stroosnijder, 1986). Actual transpiration rate is equal to the root water uptake rate being on turn a function of soil matric, osmotic potential and Tpot.

The crop model inside SWAP is WOFOST (ALTERRA, 2004), simulating the daily growth of a specific crop, given the selected weather and soil data. For each simulation, users select specific boundary conditions, which consist of the crop calendar and the soil's water and nutrient status. WOFOST follows the hierarchical distinction between potential and limited production. Light interception and CO2 assimilation are the growth driving processes, and crop phonological development is the growth controlling process.

An extensive description of the format of input and output is given by Kroes et al (1999). A summery of al input files the model can handle is given in Table 1. Some files are required, others are optional. A minimum dataset exists of 4 data files (see Table 1). The model requires daily meteorological data as input. Soil hydraulic functions are required as well as boundary conditions regarding interaction with deeper soil layers and surface water systems. For irrigated conditions irrigation timing and depth criteria can be input or generated by the model. Detailed rules for the formats of all input data files are given by Kroes et al (1999).

The model generates water and salt balance over a flexible time period ranging from days to several years. Balance terms include evaporation by intercepted rainfall, bare soil and crop, irrigated (computed) gifts, runoff, infiltration and drainage. Other examples of output are water contents, pressure heads, LAI, soil temperatures, which can be generated at time scales varying from one day to a maximum of one year.

The program may optionally generate various ASCII output files:

Water balance with cumulative data(*.wba).
Water balance with data for time increments (*.inc).
Water balance with data cumulative over time and vertical space (*.bal).
Solute balance (*.sba).
Soil temperature (*.tep).
Soil moisture, solute and temperature profiles (*vap).
Waterfluxes to/from surface water (Extended Drainage) (*.drf,*.swb,*.man).
Simulated irrigation demands/gifts(*.sc1,*.sc2,*.sc3).
Crop growth state parameters (*.cr1,*.cr2,*.cr3).
Export files with data that cover the entire simulation period (*.afo,*.aun,*.ate,*.air).
The export files can be directly used as input for pesticide and nutrient models.

Sensitivity and Limitations:

Listed below are excerpt from the study of (Wesseling and Kroes, 1998) on the global sensitivity of SWAP model:

- Boundary conditions (both upper and lower) are of crucial importance when applying the model.
- For all soil-crop combinations, the soil and crop evaporation were strongly dependent on the function describing the leaf area index(LAI).
- Drainage, simulated as lateral discharge, is very sensitive to surface water levels.
- High groundwater levels are strongly related to surface water levels; low groundwater levels depend on the combination of LAI, soil physical parameters and surface water levels; the average groundwater level is mainly determined by the level of primary drainage system.
- At low values for the saturated hydraulic conductivity, the model did not succeed in

finishing the simulations within one hour CPU-time; this occurred for peat at values below 0.1 cm d^{-1} and for clay at values below 0.06 cm d^{-1}. At these low values, the Richards' equation cannot be solved at the specific CPU-time.

Moreover, SWAP does not consider:

- The simulation of regional groundwater hydrology.
- The interaction between crop growth and Nitrogen availability.
- Non-equilibrium sorption of pesticides and the simulation of metabolites (Kroes et al., 1999).

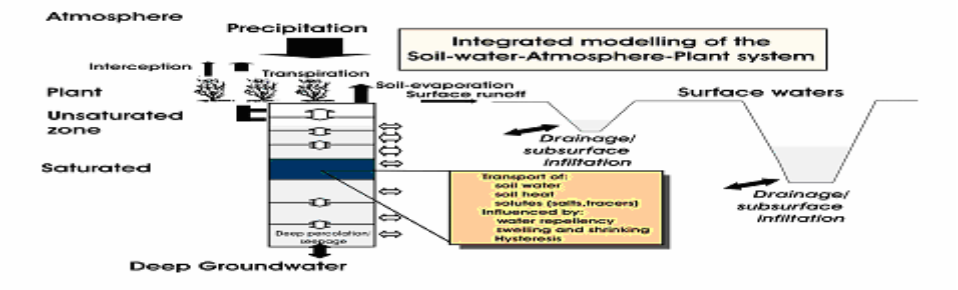

Figure1: A schematized overview of the modeled system

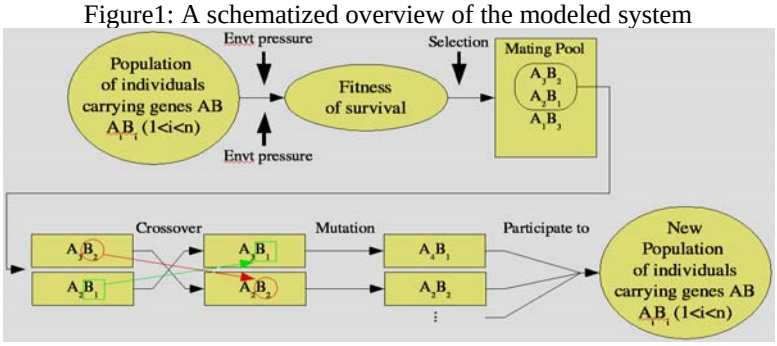

Figure 2: Genetic Algorithms

Table 1: Summary of Input File Requirements for SWAP Model

Kind of data	Description of file-content (kind of parameters)	Filename	Required	Optional
General	Simulation and I/O-options	Swap.key	+	
Meteo	Daily data	Hupsel.yyy	+	
	Detailed rainfall	Hupselir.yyy		+
Irrigation	Irrigation fixed	Hupsel.irg		+
	Irrigation calculated	Irrig.crp		+
Crop	Rotation	Yer80.cal		+
	Detailed non-grass	Maize.crp		+
	Detailed grass	Grass.crp		+
	Simple crop model	MaizeS.crp		+
Soil related	Soil water	Hupsel.swa	+	
	Soil hydraulic functions	Sandt.sol	+	
	Drainage lateral: basic	Hupsel.drb		+
	Drainage extended: surface water	Hupsel.dre		+
	Bottom boundary conditions	Hupsel.bbc	+	
Heat	Heat flow	Hupsel.hea		+
Solute	Solute transport and transformation	Hupsel.slt		+

3.3 The Genetic Algorithm Used with SWAP

The genetic evolutionary processes, that tend to modify the individuals in an isolated population having a stable pool of alleles. Four main genes modifiers can be identified. The mutation is a random change that occurs in the characteristic of the genes. This modification may be passed to the offspring. A gene flow is the introduction of new genetic material in the population by including an external individual into the population. Genetic drift is the occlusion of alleles, and happens only in small populations. Natural selection operates to choose the fittest individual for further reproduction, as response of environmental pressure on the individual survival.

All of these changes in the genotypes of the individuals of the population permits the selection of certain alleles that are most important in fitting the environment pressure on the population. When this environment pressure is the optimization function of the SWAP model with observed outputs (Figure 2), it becomes clear that the individuals are being selected to fit the input parameters.

The genetic algorithm used in this study is a simple implementation of the above-mentioned processes of evolution. It is not using the original binary-coded system implemented on similar optimizations by (Amor V. M. Ines, 2002), though the application is following a similar type of interaction with SWAP and evapotranspiration target values. The real-coded genetic algorithm (Michalewicz, 1996) is removing one layer of programming that is the coding/decoding to-from binary of the alleles.

The parameters under optimization are the starting date of cropping, the time extent of cropping and the groundwater depth in 1st January and in 31st December. It is expected to use rice pixels with double cropping as a case study.

The search domains for the dates of starting of cropping will require a non-overlapping restriction of about 90-100 days for soil preparation essentially. The time extent of the cropping season will be between 3 to 5 months. The groundwater level maybe ranging from 0 to 500cm depth but for the purpose of the case study it may be narrowed according to some general information about the area in order to improve the time efficiency for convergence.

Consider C the cost function, having (*x*,*y*,*d*,*p*) parameters, *x* the *longitude [0-180/E-W]*, *y* the *latitude [0-90/N-S]*, *d* the *date [yyyymmdd]* and *p* the *pixel size [90/1000]*.

For low spatial resolution satellite data (i.e. Modis): p=1000, *d ?* [e,...,f], with e to f being the different satellite overpass dates.

$$C_{xyd1000} \quad \sum_{e}^{f} ETa_{xyd1000} \; ETa_{SWAP_{xyd1000}} \qquad \text{(mm/d)} \; \textbf{(1)}$$

For high spatial resolution satellite data (e.g. Aster): p=90, $d \in$ [i,...,j], with i to j being the different satellite overpass dates.

$$C_{xyd90} \quad \sum_{i}^{j} ETa_{xyd90} ETa_{SWAP_{xyd90}} \qquad (mm/d) \; \textbf{(2)}$$

For our experimental purpose we are using Modis data with pixel size 500 and the cost function is

$$C_{Xyd500} = \sum_{i}^{j} \left| ETa_{xyd500} - ETa_{SWAP\,xyd500} \right| \qquad (mm/d) \; \textbf{(3)}$$

The fitness of an individual having *xydp* characteristics will be the inverse of the cost function aiming at minimizing the differences between SWAP simulation and target ETa, i.e. $F_{xydp} = 1 / (\text{cost} + \text{constraint})$.

$$F_{Xyd500} = 1 / (C_{Xyd500} + constraint) \qquad (mm/d) \; \textbf{(4)}$$

```
constraint=TECdoy-90.0; /* TEC should > 90 */
 if (constraint >= 0.0){
        lambda=0.0;
        const1=lambda*(pow (constraint,2));
 }
 else {
        lambda=10.0;
        const1=lambda*(pow (constraint1, 2));
 }
constraint=const1;
```

4 THE METHODOLOGY TO IMPLEMENT IN CLUSTER

4.1 Serial SWAP_GA:

Serial SWAP-GA is a combination of SWAP crop model, Real Coded GA and ETa simulated data. ETa data is inputted and GA is used for finding the best parameters for SWAP model (to find the crop growth rate) so that the output of SWAP will be assimilated with ETa data. The procedure of assimilating is shown in (Figure 3). The difference of SWAP-GA ETa data and actual ETa RS data is the cost value (Equation 3) and by inversing the summation of cost and constraints, a fitness value (equation 4) will be generated. The highest fitness value will give better assimilation. To implement SWAP-GA in parallel using Cluster computers the following approach is proposed.

> All populations will be distributed properly among available salves by Master node. Slaves do the evaluation, generate fitness and sent back the population (with fitness) to Master.

4.2 Parallel SWAP-GA with Population Distribution:

To parallelize the SWAP-GA the tool used is MPI (PACS Training Group). MPI stands for “Message Passing Interface”. MPI is a library of functions (in C) or subroutines (in FORTRAN) that one can insert into the source code to perform data communication between

processors (Hoffman and Hargrove, 2000). At first, the GA evaluations run parallel for only one generation and according to population size # of processors are selected. For example, the population size is 5; there will be 6 processors (one master and 5 slaves). To access the input files in parallel, different directories (different name) are created in each slave local memory and copy all the input files inside these directories. The directory name should follow the processor name convention (pro1 is a directory under processor 1 ranked by MPI).

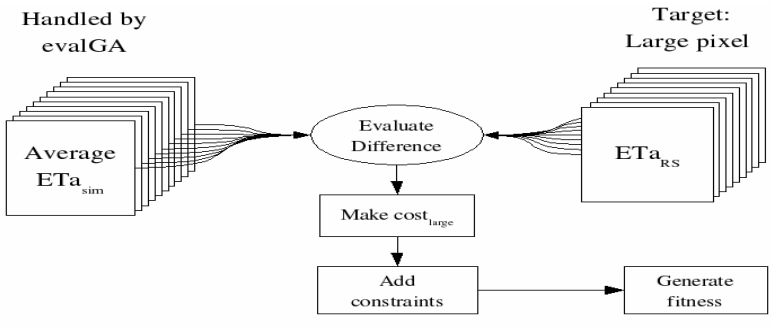

Figure 3: Serial SWAP-GA Model Structure Diagram

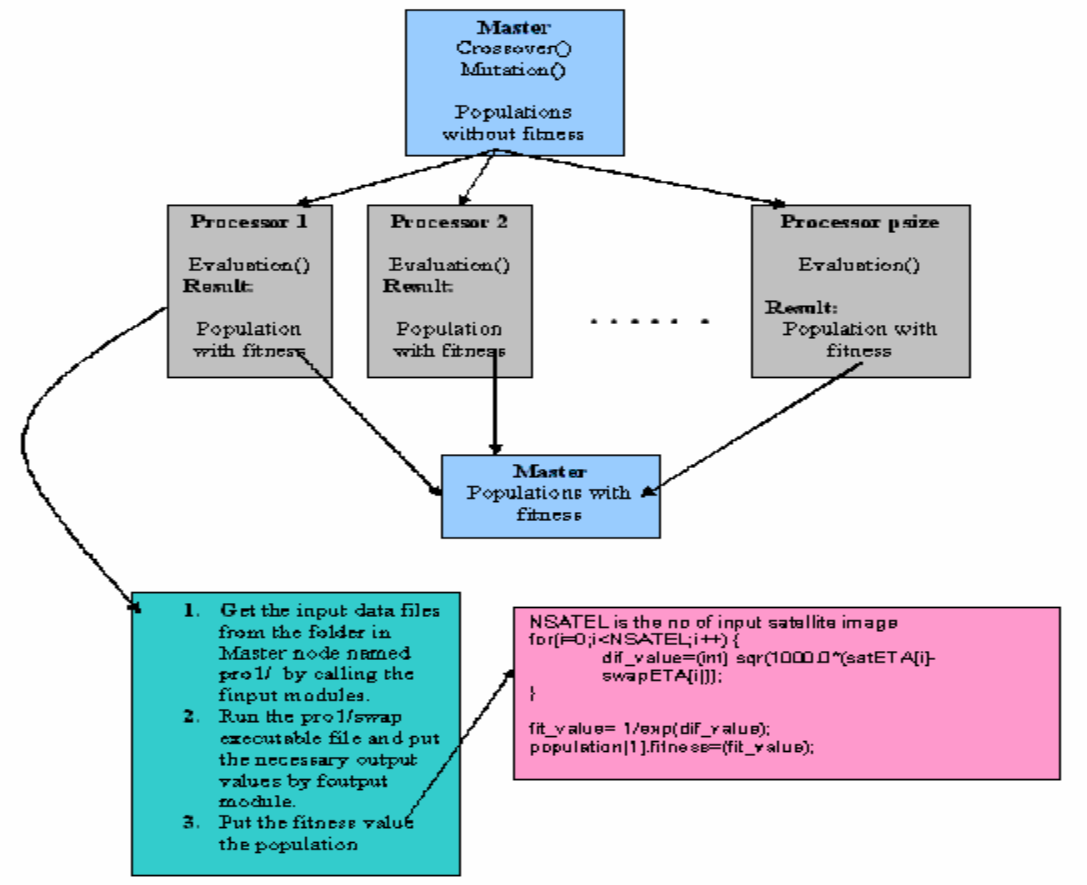

Figure 4: Parallel SWAP-GA Programming Structure Diagram

Master reads all the input files one time and calls the initialization module to initialize randomly the values of all populations. Then it sends each population to separate slaves (Figure 4 and 5) to be evaluated for getting the fitness values. Each slave then makes some internal process, finds the fitness value for its population and sends

the population back to the master nodes. Than master node decides the best populations match with the RS ETa data. Increasing the generation additional step is necessary that is to add the internal loops procedure *while (generation < MAXGENS) in* both master and slaves. MAXGENS is a global variable, therefore no need any further coding for broadcasting or sending data from master to slaves or vice versa. A profile is made with the experiment results. In order to analyze the running process for finding any better option. Also need to implement the procedure to copy the whole input files into slaves' local memory at very beginning of running code. So that slaves run the files internally into their own memory. This procedure will give less time than the previous one.

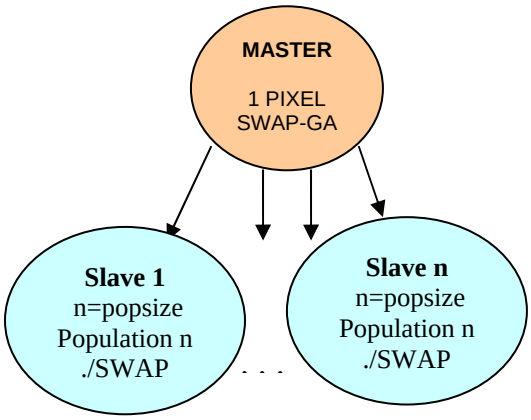

Figure 5: Parallel SWAP-GA Model Structure Diagram.

The following algorithm presents when number of populations are equal to number of slaves then there will be less headache, just distributed the equal number of population to each salves. But if more than number of slaves than the total population or pixel numbers will be equally distributed over the slaves (Master just send and receive population in each generation) and rest populations (left_population) are distributed sequentially from slave's rank 1 to left_population. In this case, for the marked slaves (rank 1 to left_population) the population_send_no in MPI_Send, is just added by one and passed to slaves. Otherwise, total population number will less than available slaves number than each population will be distributed over the equal number of slaves and rest slaves will do nothing and finish their jobs.

Algorithm and Pseudo Code:

```
POPSIZE=No of Population;
N_Pro=Total No of Available Processor;
N_Slaves=N_Pro-1;

if(POPSIZE>N_Slaves){
        population_send_no=POPSIZE/(N_Slaves);
       left_population=POPSIZE- ((N_Slaves)* population_send_no);
}
else {
       population_send_no=1;
       N_Slaves=POPSIZE;
}
if (MASTER)
{
        //The following code will be executed for each generation (Maximum generation is used as a //threshold value to break the loop )
        first_index=0;
        for(i=1;i<=N_Slaves;i++){
              if(i<=left_population){
                          MPI_Send(&population[first_index],population_send_no+1,genstype,i,1,MPI_COMM_WORLD);
                      first_index=first_index+population_send_no+1;
                  }
             else{
                          MPI_Send(&population[first_index],population_send_no,genstype,i,1,MPI_COMM_WORLD);
                          first_index=first_index+population_send_no;
                  }
          }
        first_index=0;
        for(i=1;i<=N_Slaves;i++){
        if(i<=left_population){

                          MPI_Recv(&population[first_index],population_send_no+1,genstype,i,1,MPI_COMM_WORLD,&status);
                     first_index=first_index+population_send_no+1;
                }
                else {
                          MPI_Recv(&population[first_index],population_send_no,genstype,i,1,MPI_COMM_WORLD,&status);
```

```
                        first_index=first_index+population_send_no;
                    }
                }
// Best populations will be chosen after crossover, mutation, elitist for next generation.
}
if (Slave){
    //The following code will be executed for each generation.
    MPI_Recv(&population[0],population_send_no,genstype,0,1,MPI_COMM_WORLD,&status);
    //Analyze all received population's gens //Execute SWAP and find out the cost //Evaluate the fitness
    MPI_Send(&population[0],population_send_no,genstype,0,1,MPI_COMM_WORLD);
}
```

5 EXPERIMENTS AND RESULT

Table 2: Serial Time for 1 evaluation Ethernet Cluster (Optima)

Reading From File		**Writing in File**		**Execute SWAP**	
Reading file size KB	Accessing Time (s)	Reading file size KB	Accessing Time (s)	SWAP file size KB	Execution time (s)
Out.inc=34.9	.000551	Roodc.cal=0.154	.000103	307	1.753
Satellite.dates=0.129	.006006	Roodc.bbc= 0.438	.000095		
Satellite.eta=0.082	.000137	Crop.crp=0.87	.000134		

SWAP-GA Implementation in Optima (100 Mbps Cluster) with 1 Master node and 2 slave nodes. Here (Figure 9), P0, P1 and P2 are indicating the Master, Slave1 and Slave2 respectively. t_x, t_{1x} and t_{2x} represent corresponding processors P0, P1 and P2's the execution times.

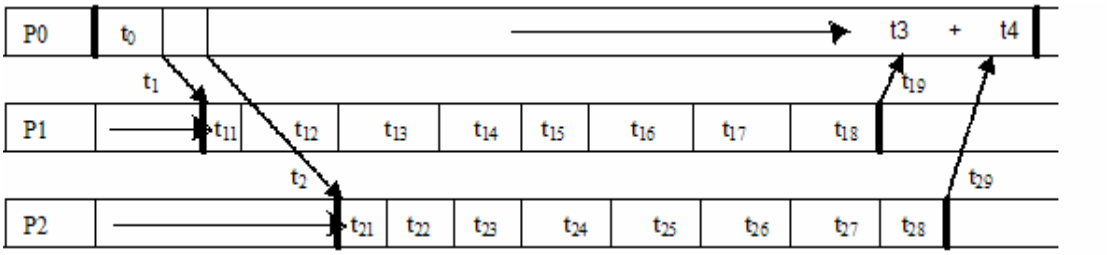

Figure 6: Flow Diagram for SWAP-GA implementation in Optima Cluster computer system

Table 3: Parallel SWAP-GA execution analysis for 1 generation and 2 populations SWAP-GA running in master with 2 slaves

Symbol	**Operations on time**	**Size(KB)**	**Time(s)**	**Rate(KB/s)**
t0	Master Init.+ 2 serial evaluations		3.93	Master Processor speed
t1+t2	Sending data to slave1 and slave2	0.125	.000141	886.52
t11	Data receiving by slave1	0.061	3.92	Queuing time in Slave1
t12	Writing data file Roodc.cal	0.154	.0003	Slave1 Processor speed
t13	Writing data file Rood.bbc	0.438		
t14	Writing data file Crop.crp	0.87		
t15	Executing swap	307	1.54	
t16	Reading data file Out.inc	35	.0035	
t17	Reading data file Satellite.dates	0.129		
t18	Reading data file Satellite.eta	0.082		
t19	Data sending from slave1 to master	0.061	.000044	1367.06
t21	Data receiving by slave2	0.061	3.92	Queuing time in Slave2
t22	Writing data file Roodc.cal	0.154	.0004	Slave2 Processor speed
t23	Writing data file Rood.bbc	0.438		
t24	Writing data file Crop.crp	0.87		

t25	Executing swap	307	1.81	
t26	Reading data file Out.inc	35	.0044	
t27	Reading data file Satellite. Dates	0.129		
t28	Reading data file Satellite.eta	0.082		
t29	Data sending from slave2 to master	0.061	.000058	1043.91
t3+t4	Receiving data by master	0.125	1.82	Queuing time in Master
Total	6.18 Seconds			

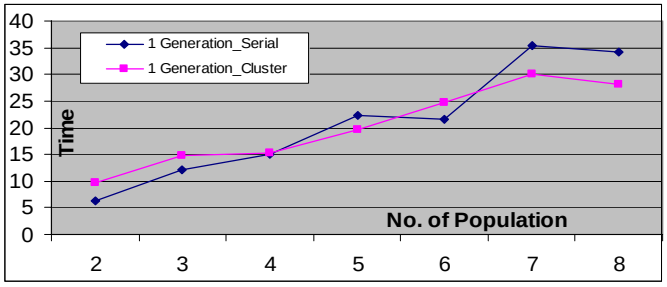

Figure 7: Timing Diagram: Cluster vs Serial Computer for 1 Generation.

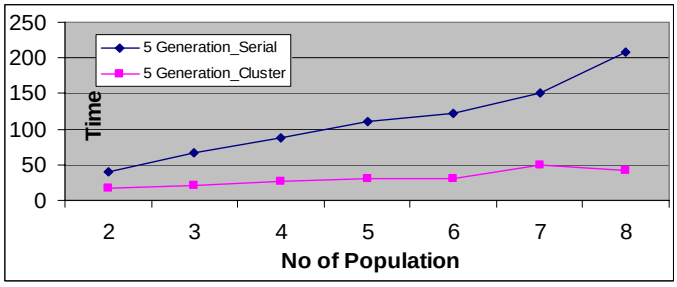

Figure 8: Timing Diagram: Cluster vs Serial Computer for 5 Generations

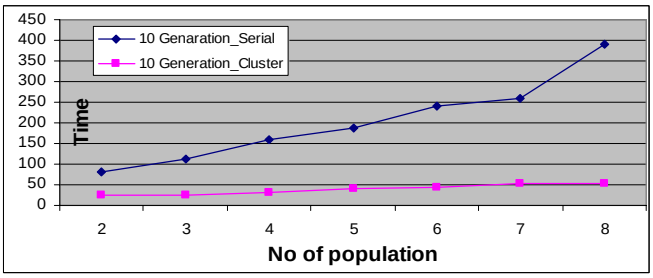

Figure 9 :Timing Diagram: Cluster vs Serial Computer for 10 Generations

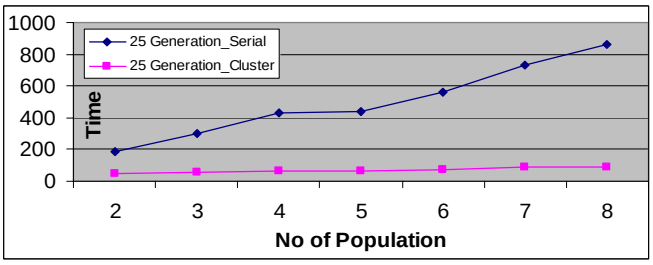

Figure 10: Timing Diagram: Cluster vs Serial Computer for 25 Generations

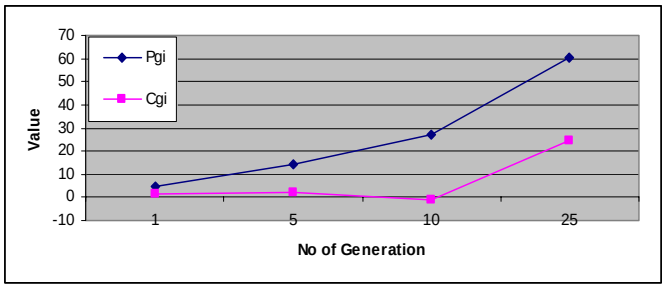

Figure 11 :Generation with Population (Pg_i) and Generation with Constant (Cg_i) Curve with Serial SWAP-GA

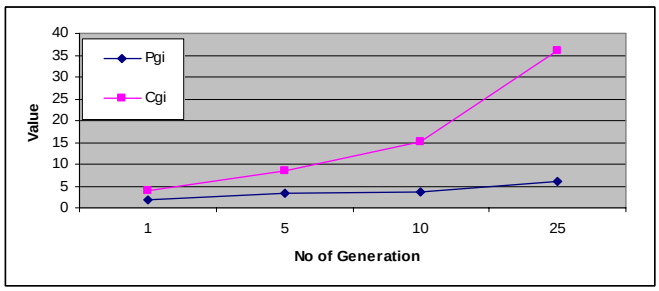

Figure 12 : Generation with Population (Pg_i) and Generation with Constant (Cg_i) Curve with Parallel SWAP-GA

Table2 shows some result values those describe the time for writing, reading and executing swap files while SWAP-GA model runs serially (GA works serial). And later we do the same activity (Table3) when running the SWAP-GA in parallel (GA works within cluster, maintaining master-slaves behavior). By analyzing these two tables' data we can easily understand how much time will kill for communication activities (sending and receiving data). In Table2 and Table3 the reading, writing and execution time are approximately same because all data files including swap executable file are copied in to each slave local memory. So, there will be no communication delay.

Actually, in Table3 the most time consuming part is for receiving data by slaves. It actually held for the internal queuing activities of MPI procedure.

Figure 7 to Figure 10 shows timing diagrams for both serial running SWAP-GA and Parallel running SWAP-GA by increasing the generation. Here at the 1 (Figure 7) and 5 (Figure 8) generations the parallel procedure will not show the improvement (a lot of diversities). But when we increase the generation higher then we gain the time efficiency. Moreover, in Figure 10 for 25 generations the curve for Parallel SWAP-GA shows like straight line that means that here the parallelism works with known communication delay.

Considering all curves (Figure 7 to Figure 12) as linear, the following equations will be generated.
The symbols, g_i is the generation number, Tg_i is the time in generation gi, Pg_i is the population number with generation g_i, Cg*i* is the constant for linear curve with respect to generation g_i,

For i ∈ [generation]

$$Tg_i = Ag_i\ Pg_i + Cg_i \quad (5)$$

Serial SWAP_GA Parameter Equations:

$$Ag_i = 17.95\ g_i - 18.26 \quad (6)$$

$$Cg_i = 6.69 g_i - 10.017 \quad (7)$$

Parallel SWAP_GA Parameter Equations:

$$Ag_i = 1.26\ g_i + 0.57 \quad (8)$$

$$Cg_i = 10.25 g_i - 9.715 \quad (9)$$

Equation 5 is the actual equation, which defines the time value for a specific generation number with known value of population number. By using generation number 1, 5, 10, 25 (Figure 7 to Figure 10) Figure 11 (Serial SWAP-GA) and Figure 12 (Parallel SWAP-GA) are created. From the Figure 11 and 12 the corresponding equations 6, 7 and 8, 9 are derived.

However, within this experiment highest 25 generations and 8 populations are used but equation 5, 6, 7, 8, 9 have given a generic model. For any value of generation and population number the running time in both parallel SWAP-GA and serial SWAP-GA can be measured by using these equations. Thus, the above equations can solve the complication of running parallel SWAP-GA within Clusters those hold limited number of slaves (PC).

6 CONCLUSION AND FUTURE WORK

In this paper, an approach for implementing data assimilation on an agro-hydrological model in a cluster computer has been studied. It is analyzing only one pixel of Remote Sensing data at a time. In the near future, implementation of the same procedure will be done for a Remote Sensing images of few hundred thousands pixels. For that purpose Grid computing can be used where Master node will distribute each pixel into available clusters and each cluster will execute the model for analyzing each pixel information.

REFERENCES

ALTERRA, 2004. WOFOST description web page. <http://www.alterra.wur.nl/NR/exeres/71D98F7E-48B7-4D11-8339-F6B8D5291FF4.htm>

Amor V. M. Ines, 2002, Improved Crop Production Integration GIS and Genetic Algorithms. Doctoral Thesis. AIT. Bangkok, Thailand. *AIT Diss no. WM-02-01*.

Bastiaanssen, W.G.M., 1995. Regionalization of surface flux densities and moisture indicators in composite terrain. A remote sensing approach under clear skies in Mediterranean climates. Agricultural Research Department, Report 109, Wageningen, The Netherlands.

Belmans, C., Wesseling, J.G. and R.A. Feddes.,(1983) Simulation of Water Balance of a Cropped Soil: SWATRE, journal of Hydrology., 63, 271-286.

Black, T.A., Gardner, W.R., and Thertell, W., 1969. The prediction of evaporation, drainage and soil water storage for a bare soil. Soil Sci. Soc. Am. J., 33:655-660.

Boesten, J.J.T.I., and Stroosnijder, L., 1986. Simple model for daily evaporation from fallow tilled soil under spring conditions in a temperate climate. Netherlands Journal of Agricultural Science, 4:75-90.

Feddes, R.A., Kowalik, P.J., and Zaradny, H., 1978. Simulation of water use and crop yield. Simulation Monography, Pudc, Wageningen, 188pp.

Hoffman. F.M., Hargrove, W.W., 2000, High Performance Computing: An Introduction to Parallel Programming with Beowulf, <http://climate.ornl.gov/~forrest/osdj-2000-11/>

Kroes, J.G., J.C. van Dam, J.Huygen, R.W.Vervoort, 1999. SWAP 2.0: User's Guide, Simulation of water flow, solute transport and plant growth in the Soil-Water-Atmosphere-Plant environment. Technical Department 53. Alterra (DLO Winand Staring Centre), Wageningen Report 81, Department Water Resources, Wageningen Agricultural University, Wageningen, 167 p.

Michalewicz, Z., 1996. Genetic algorithms + data

structures = evolution programs. 3rd reviewed and extended edition. Springer-Verlag, Berlin Heidelberg.

PACS Training Group. Introduction to MPI <http://pacont.ncsa.uiuc.edu:8900/public/MPI/>

Van Dam, J.C., Huygen, J., Wesseling, J.G., Feddes, R.A., Kabat, P., van Waslum, P.E.V., Groenendjik, P. and C.A. van Diepen 1997. *Theory of SWAP version 2.0: Simulation of water flow and plant growth in the Soil-Water-Atmosphere-Plant ebvironment. Technical Document 45*. Wageningen Agricultural University and DLO Winand Staring Centre. The Netherlands.

Yann Chenmin, Kiyoshi Honda, Amor V. M. Ines, 2004, Genetic algorithm for assimilating remotely sensed evapotransiration data using a soil-water-atmosphere-plant model. Implementation issues, *Proceedings of the 2004 FROSS/GRASS Users Conference, Bangkok, Thailand.*
<http://gisws.media.osaka-cu.ac.jp/grass04/viewabstract.php?id=27>

AUTHOR BIOGRAPHIES

Shamim Akhter is a Masters student in Computer Science and Information Management, School of Advanced Technologies, Asian Institute of Technology (AIT), Thailand. He completed his undergraduate from Department of Computer Science, American International University-Bangladesh (AIUB) in 2002. He is now doing his thesis "Implementing SWAP-GA Model in Cluster Computers" with Dr. Honda, STAR Program, AIT, Thailand. His email address is <shamim7862002@yahoo.com>

Kiyoshi Honda is an Associate Professor in Space Technology Applications and Research Program (STAR), School of Advanced Technologies, Asian Institute of Technology, Thailand. He received a Ph. D. from the Doctor of Engineering, University of Tokyo, Japan, 1992. He worked as Invited Scientist, National Space Development Agency of Japan (NASDA>JAXA), 2000-2002, Director, Asian Center for Research on Remote Sensing(ACRoRS),AIT,1997-2002, Japanese Faculty Representative, AIT, 1999.5 - 1999.10. His teaching and research interests include Remote Sensing and GIS, High performance computing and Application of Remote Sensing for Erosion Control. His email address is <honda@ait.ac.th>

Yann Chemin got his M.Sc. in Land and Water Resources Management from Silsoe College (UK), he is currently following his research period as Doctorate Student in the Asian Institute of Technology (Thailand). His research interests are in evapotranspiration by remote sensing and RS supercomputing. His email address is <yann.chemin@ait.ac.th>.