Parallel Computing Applied to Satellite Images Processing for Solar Resource Estimates

Rodrigo Alonso[1] Sergio Nesmachnow[2]

[1] Instituto de Física, Facultad de Ingeniería, Universidad de la República

[2] Centro de Cálculo, Facultad de Ingeniería, Universidad de la República

July 23, 2012

[1] rodrigoa@fing.edu.uy
[2] sergion@fing.edu.uy

1. Introduction
   - General introduction
   - Solar irradiation model and satellite data bank
2. problem description
   - problem overview and description
3. parallel implementation
   - parallel model
   - algorithm description
4. performance evaluation
   - Test set, methodology and environment
   - Evaluation results
5. Conclusions
   - Conclusions and future work

Introduction: solar resource overview

Objective:

- Estimate the amount of solar energy that reachs ground level.
- Solar energy applications, meteorology, agronomical planning.
- This work: parallel computing for satellite-based estimation.
- This work: first step for migrating to a C platform using parallel computing strategies.

Uruguay's solar resource knowledge context:

- First Uruguayan Solar Map (2009).
- First satellite-based solar resource model (2011).
- Next: improvement (2012), optimization and operation.

Why a satellite-based irradiation model?

- Spatial resolution ≈ 2 km. Hourly or intra-hourly estimates.
- Satellite-based estimates provides better accuracy than interpolation of ground measurements for sites located more than 30 km away from the measuring stations.

Introduction: solar resource model

- Justus, Paris and Tarpley hourly irradiation model (1986).
- Statistical model which relies on a multiple regression.
- Adjusted for the target region using ground measurements.
- A spatial neighborhood of sites is used (**cells**).

Multiple regression model:

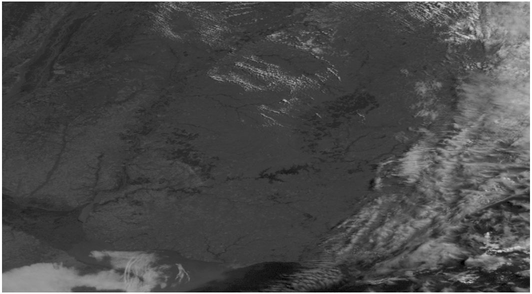
$$I = I_{sc}\left(\frac{r_0}{r}\right)^2 \cos\theta_z \left(a + b\cos\theta_z + c\cos^2\theta_z\right) + d\left(B_m^2 - B_0^2\right)$$

- I_{sc} - Hourly value of the solar constant.
- $(r_0/r)^2$ - Eccentricity factor (due to Sun-Earth distance).
- $\cos\theta_z$ - Cosine of the solar zenith angle.
- B_m - Mean brightness at the observed cell.
- B_0 - Clear-sky brightness for the observed cell.

Introduction: satellite images database

- Cloud information is taken from the visible channel.
- More than 10 years of GOES-East images are available.

satellite	start date	end date	images
GOES 8	01/01/2000	31/03/2003	24750
GOES 12	01/04/2003	14/04/2010	51900
GOES 13	14/04/2010	30/04/2012	15300
total	01/01/2000	30/04/2012	**91950**



Mean Brightness B_m

- Is calculated for every site.

Clear-sky Brightness B_0

- A model is trained for every site.

Problem overview

- Compute B_m mean brightness for every cell as an average.
- NetCDF format: standard machine-independent data format.
- Every image contains data and navigation information.

How many cells and images?

- Regular grid with a 5×5 latitude-longitude spacing.
- Target region varies between: 30 and 35 degrees South.
  53 and 59 degrees West.
- A total of $(5 \times 30) \times (6 \times 30) = 27000$ sites for every image.

A total of 27000 sites shall be processed for more than 90000 images. Thus, approximately **2430 millon averages** are to be computed in order to process the whole satellite database.

- Without using parallel computing: **15 hours** are needed to process the data bank in a single PC/server machine.

Problem description

Performance oriented point of view:

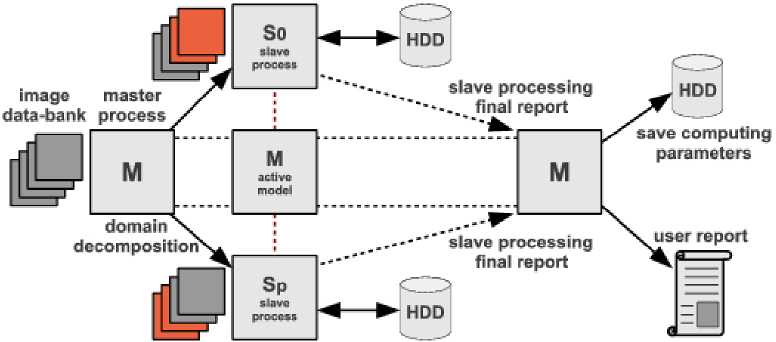
- Processing a single image takes a reduced execution time.
- Each image can be efficiently processed by a single processor.
- Complexity relies on the large number of images to process.

Context: first step towards a operational parallel platform

- Not only the main parallel algorithm was developed.
- A set of common libraries were built (for similar or future algorithms): implemented libraries are independent from any parallelization scheme.
- User parameters are introduced via plain text:
    - latitude-longitude spacing, target window.
    - folders, years of images to process, etc.
- Platform selection: C lenguaje and MPI standard.

Parallel model

- A data-parallel scheme was adopted.
- Master process performs the domain-decomposition.
- An active master-slave model is used.
- Several processes are used which execute in differents nodes.
- Every process receives a reference of the first and last image.



Implemented libraries

Modularity: three main libraries were developed.

(a) a *processing* library.

(b) an *assigment* library.

(c) a *calibration* library.

Processing library

- Implements all functionalities required for image processing.
- Features: interact with NetCDF files and disk writting.
- Start and finish image is received as a parameter (assigment).

Assigment library

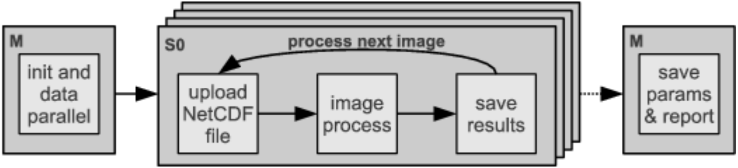
- Able to count how many images were requested to process.
- Logic to assess the domain-decomposition.

Calibration library

- Satellite calibration and count-to-radiance conversion.

Algorithm description

- The well-known MPICH implementation was used (1.2.7p1).
- Algorithm flow:
  - (a) initialization and data parallel distribution.
  - (b) image upload from hard-disk.
  - (c) computation of the B_m values for each assigned image.
  - (d) save processed data to hard-disk.
- Master process is in charge of the initialization.
- Every slave process knows all the data to work independently.
- When a process finishes it sends a notification to the master.



Load balancing and output

Load balancing

- All the images in the data-bank:
  - have the same size and data.
  - image processing is exactly the same.
- The parallel algorithm was conceived to execute in a dedicated parallel computing infrastructure.
- A static load balancing was adopted.
- In practice, when processing ~ 7700 images (one year)
  - 20 images maximum deviation from the ideal situation.
  - correspond to a negligible value of ~ 10 seconds.

Algorithm's output

- Six plain text files with the grid values.
- Brightness counts and subproducts from the image calibration.
- Mask for corrupted values. The amount of pixels for average.

1. Introduction
   - General introduction
   - Solar irradiation model and satellite data bank
2. problem description
   - problem overview and description
3. parallel implementation
   - parallel model
   - algorithm description
4. performance evaluation
   - Test set, methodology and environment
   - Evaluation results
5. Conclusions
   - Conclusions and future work

Test set, methodology and environment

Execution platform: Cluster FING

- Server with two Intel quad-core Xeon processors at 2.6 GHz, with 8 GB RAM, CentOS Linux, and Gigabit Ethernet.

Image test set:

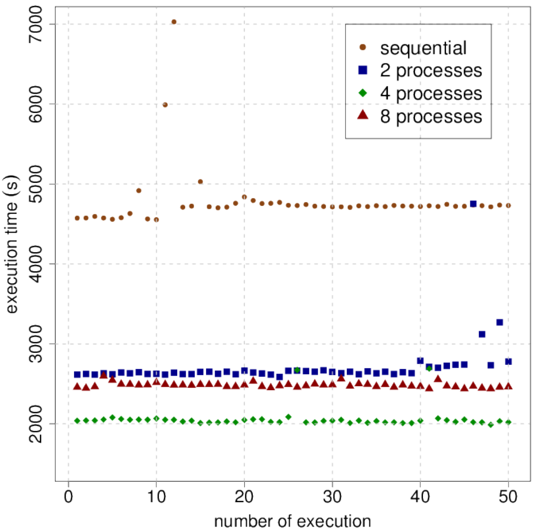
- Satellite images of the year 2011 were used (7670 images).
- Disk capacity of 60GB. Local to the node used for evaluation.

Performance metrics:

- 50 independent executions varying processes' amount.
- Speed up: compare the mean values of the sequential and parallel execution times (non-deterministic algorithms).
- Efficiency: is the normalized value of the speedup with the number of computing elements.

$$S_m = \frac{E[T_1]}{E[T_m]} \qquad e_m = \frac{S_m}{m}$$

Executions times (1)



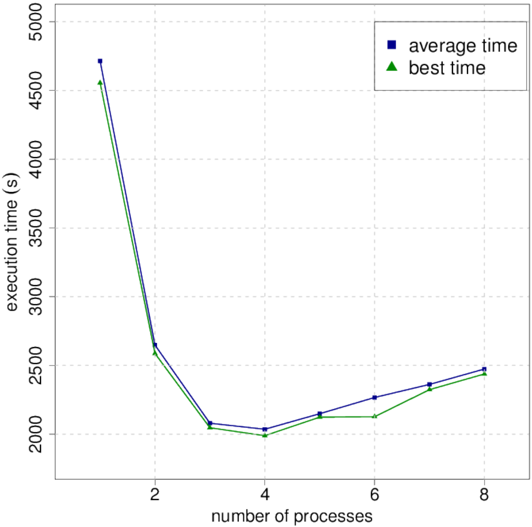
Executions times (2)

Experimental results

processes	execution time (s)			metric	
	best	avg	σ	speedup	efficiency
1 (sequential)	4555	4786	380	–	–
2	2587	2720	316	1.76	0.88
3	2047	2123	214	2.25	0.75
4	1989	2062	130	2.32	0.58
5	2124	2200	136	2.18	0.44
6	2127	2360	293	2.03	0.34
7	2324	2370	36	2.02	0.29
8	2438	2484	32	1.93	0.24

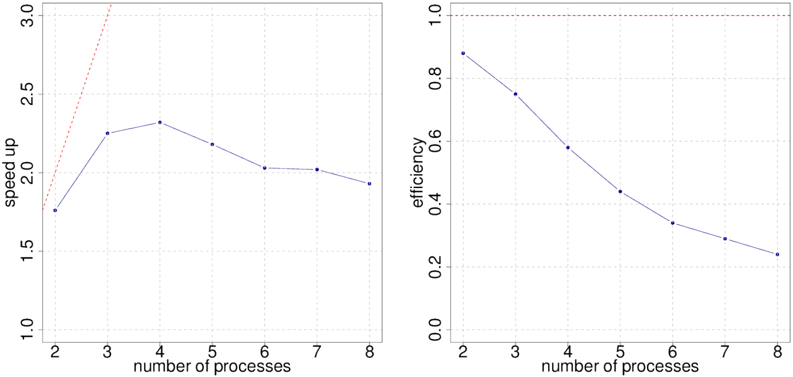
- Performance increase with processes' number for $p \leq 4$.
- A speedup of 1.76 is achieved when using $p = 2$.
- A maximum speedup of 2.32 is obtained at $p = 4$.
- When using more than 4 processes execution times increase.
- Standard deviation reduce when more processes are used.

Executions times (3)



Speed up and efficiency

- Speed up presents a maximum at $p = 4$.
- Efficiency presents a maximum at $p = 2$.
- Problem: limited input/output disk access bandwidth.



1. Introduction
   - General introduction
   - Solar irradiation model and satellite data bank
2. problem description
   - problem overview and description
3. parallel implementation
   - parallel model
   - algorithm description
4. performance evaluation
   - Test set, methodology and environment
   - Evaluation results
5. Conclusions
   - Conclusions and future work

Conclusions and future work

Conclusions:

- We present a parallel algorithm application to process satellite image information for solar resource estimates.
- The problem's complexity lies in the big amount of images.
- A parallel master-slave algorithm was developed based on image domain-decomposition.
- A maximum speedup of 2.32 was reached using 4 processes.
- Current version: it can't escalate beyond 4 processes because of the limited hard-disk drive input/output capacity.

Possible ways to improve algorithm's performance:

- Dynamic load balancing to execute in non dedicated infrastructures.
- Parallelization of hard-disk drive access.

¡Muchas Gracias!

Thank you!

Questions?