SUPIM-DAVS IONOSFERIC MODEL FOR VECTOR PROCESSING

Luiz Henrique Broch Lago[1] - lhlago@inf.ufsm.br
João Vicente Ferreira Lima[1] - jvlima@inf.ufsm.br
Jonas Noboru Tamoki[2] - jonas@hpctamaoki.com.br
Adriano Petry[3] - dr.adriano.petry@gmail.com
Haroldo F. de Campos Velho[3] - haroldo.camposvelho@inpe.br

[1]Universidade Federal de Santa Maria – UFSM, Santa Maria, RS, Brazil
[2] Independent HPC Analyst, São Paulo, SP, Brazil
[3]Instituto Nacional de Pesquisas Espaciais – INPE, Santa Maria, RS / São José dos Campos, SP, Brazil

Abstract. *The SUPIM-DAVS is an operational system to simulate ionosphere dynamics. The system is one of the activities of the space weather program (Embrace: Estudo e Monitoramento Brasileiro do Clima Espacial) from the National Institute for Space Research INPE: (INPE: Instituto Nacional de Pesquisas Espaciais). The SUPIM-DAVS is executed every day to carry out the operational forecasting 24 hours ahead. This paper reports the performance of the SUPIM-DAVS computer code on the new vector processing architecture of the NEC SX-Aurora TSUBASA / Vector Engine. This vector computer can be a good option for intensive computing because this machine combines high computer power with lower energy demand. One copy of the SUPIM-DAVS was executed in a cluster with CPU multi-core and in the NEC SX-Tsubasa. The obtained CPU times were 4730 seconds (multi-core CPUs) and 1050 seconds (vector computer). The paper describes the strategy for migrating the CPU multi-core machine to the vector engine processing to reach the reported performance.*

Keywords: *Ionospheric dynamics, SUPIM-DAVS model, Parallel processing, Vector processing*

1. INTRODUCTION

The Brazilian Space Weather program started in 2007, named *Estudo e Monitoramento Brasileiro do Clima Espacial* (Embrace) – Brazilian study and monitoring of the space weather. One activity of this program was the development of an operational system to predict ionosphere dynamics. The Sheffield University Plasmasphere–Ionosphere Model (SUPIM) – see: Bailey and Sellek (1990), Bailey et al. (1993), Bailey and Balan (1996), Souza et al. (2000, 2010) – was the selected computer model for the task. SUPIM has been changed with several improvements looking at the operational activities to the medium range for ionospheric dynamics forecasting (Petry et al., 2014). The improvements were the development of the data assimilation module –

Newtonian relaxation scheme, physical constraint on electron concentration for higher altitudes, and a parallel version of the code (Petry, 2010).

The SUPIM code solves evolution equations for several ions and electron concentrations by the numerical method of finite differences. Before starting the time evolution simulation, a balance is computed among the fields under influence of the extreme ultraviolet (EUV) from the Sun, the electromagnetic radiation in the spectrum wavelengths interval from 10 nm up to 124 nm. The EUV must be understood as the main ionization forcing. The dynamics of the electron and ion concentrations are under the action of electromagnetic forces, and the Maxwell equations must be taken into consideration. The simulation is computer intensive, motivating a permanent investigation of new algorithms and new high-performance computer (HPC) architectures to speed up the code performance.

The operational task employs the updated SUPIM to describe the total electron content (TEC) maps for predicting this field up to 24 hours ahead, with outputs for each hour. The forecasting of the TEC maps on South America is available every day on a webpage: http://www2.inpe.br/climaespacial/portal/tec-supim-prevision/. The operational forecasting is executed in a cluster with 29 multi-core CPU's. Using magnetic coordinates, the 3D model can be split into many magnetic field lines (*copies*), where each magnetic field line is a 2D mathematical model (one *copy*). The SUPIM simulation over South America for 24 hours is executed in a cluster with CPU-time 01:18:50 (hours:minutes:seconds) = 4730 seconds. Looking for a reduction in the CPU-time, the operational SUPIM version was mplemented on a vector processor. The performance of the operational SUPIM optimized by the NEC SX-Tsubasa computer is the goal of this paper, where all steps for that are shown.

2. SUPIM-DAVS SYSTEM

The operational version for SUPIM is named SUPIM-DAVS (SUPIM / Data Assimilation and Visualization System). The model simulates the time and space for temperature and concentrations of the electrons and the O^+, H^+, N^+, He^+, and NO^+ ions, considering the production and lost terms in the numerical solution. The input parameters for the SUPIM can be summarized as follows (Souza et al., 2000; Petry et al., 2014): date/time desired for simulation, magnetic field, neutral atmosphere, thermospheric neutral wind, vertical drift, and information about solar fluxes at different frequency bands (EUV, F10.7, and F10.7A). The F10.7 and F10.7A indexes are the solar radio flux of 10.7 cm wavelength (measured in solar flux unit (SFU)[1]), and F10.7A is the average of 81-day records of the daily F10.7 (Ikubanni and Adeniyi, 2017).

As already mentioned, some improvements were added to the SUPIM-DAVS system (Petry et al., 2014). To have a more numerically stable code, avoiding negative temperatures or negative ion concentrations, an exponential decay term for the ion-neutral collisional frequency was employed. The collision frequency decay (f_{in}) is given by:As already mentioned, some improvemnts were added to the SUPIM-DAVS system (Petry et al., 2014). In order to have a more numerically stable code, avoiding negative temperatures or negative ion concentrations, an exponential decay term for the ion–neutral collisional frequency was employed. The collision

[1]Note: 1 SFU = 10^{-22} $Wm^{-2}Hz^{-1}$.

frequency decay (f_{in}) is given by:

$$f_{in} = f_0 \, e^{-\Delta s / H_i} \tag{1}$$

being the initial ion-neutral collisional frequency represented by f_0, the height variation along the magnetic field line is denoted by Δs, and H_i is the ion scale height. In addition, the Newtonian relaxation method for data assimilation, combining the previous prediction and observation, is applied to compute the state $x_a(j)$ – the *analysis*: the initial condition for the forecasting process, being j a coordinate, from the background $x_b(j)$ (previous prediction) added to the innovation (difference between background and observation). The analysis is computed as follows:

$$x_a(j) = x_b(j) + g_\alpha \frac{\sum_i w_{ij}^2 \gamma_i [y(i) - x_b(i)]}{\sum_i w_{i,j}^2} \tag{2}$$

where g_α is the nudging parameter, $\gamma_i = \{1, 0\}$ for accepting if the observation $y(i)$ will be included (1) or not (0) in the assimilation process, and the w_{ij} is a weight calculated by:

$$w_{ij} = \max\left(0, \frac{R^2 - d_{ij}^2}{R^2 + d_{ij}^2}\right) \tag{3}$$

with R the influence radius of the observation, and d_{ij} the distance between the i-th observation up to j-th coordinate.

The SUPIM-DAVS system is executed on a parallel version in a cluster of multi-core CPUs, with a set of scripts mapping all the fields from geographical coordinates into magnetic ones – and vice-versa –, and applying an interpolation scheme to the computed fields if necessary. Finally, there is a script to generate TEC forecasting maps.

3. THE VECTOR PROCESSING

The vector processing version for the SUPIM is implemented on the NEC SX-Aurora TSUBASA vector processor. The SX-Aurora TSUBASA architecture links a vector engine (VE), where the vector processor is resident, with a vector host (VH). The VH is a processor x86 (architectures from a family of complex instruction set computer (CISC)) Linux hosts connected to the VE by PCI Express (PCIe) interconnect – see Figure 1. The clock frequency depends on VE CPU (1.4 or 1.6 GHz), and the VH CPU can range from 2.4 up to 5.5 GHz. In this architecture, a full application is executed on VE, while the VH manages the processing system calls needed by the application (Komatsu et al., 2018). Different from other co-processors, the idea is to avoid systematic data transfer between the VE and its VH.

The VH can control up to 8 VE, and each VE can have up to 8 vector kernels. The vector system can be seen as 8 processing nodes with distributed memories, where each node has 8-cores for processing with shared memory. Each VE has 48 GB memory, and the VH has 196 GB memory.

Another important difference of other co-processors to develop scientific applications is that there is no need to use a diverse programming standard. The SUPIM kernel was developed in Fortran language. Therefore, only code optimization looking at the best code execution on the vector architecture is necessary.

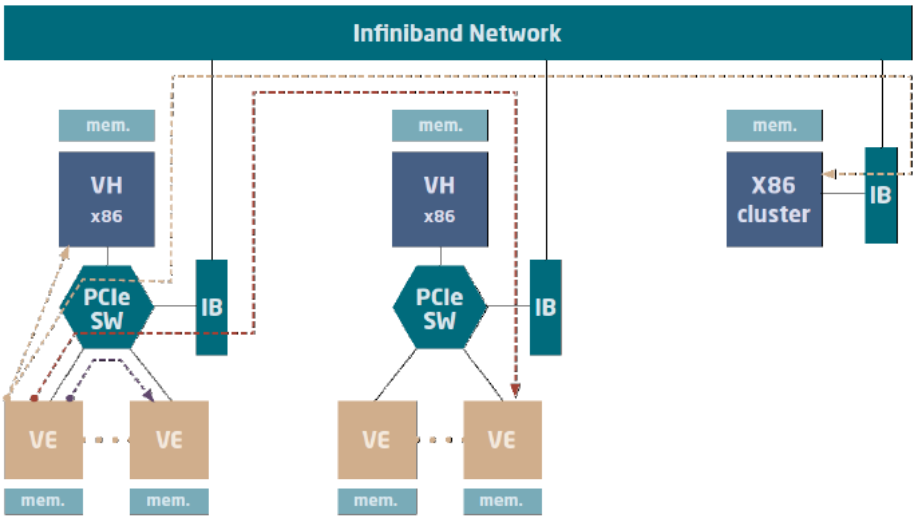

Figure 1- NEC Sx-Aurora TSUBASA architecture – sketch extracted from the NEC's webpage.

3.1 SUPIM for vector processing

The SUPIM code was installed in an NEC SX-Aurora TSUBASA computer, where the option "- *minit-stack=zero*" was employed to maintain the zero values for all local variables as default values. First, one execution was carried out without any optimization directives to compile the SUPIM. Secondly, compilations were performed under different optimization levels: O1, O2, O3, and *inline* compilation directives.

Finally, a profiler was applied to identify the processing demands for different routines from the SUPIM. The profiler identified the following processing demands shown in Table 1.

Table 1- Routines and processing demands.

Importance	Routine	Demands
(1)	GLOBE7:	28.7%
(2)	GTS7:	21.7%
(3)	DENSU:	16.8%
(4)	SPLIN:	13.1%

3.2 Performance for the SUPIM vector version

To illustrate a SUPIM-DAVS simulation, figure 2 shows the result of the vertical-TEC (VTEC) – integrating the electronic concentration on the vertical coordinate, as well as ion concentrations over South America. The figure displays the simulated VTEC (upper) and the same field measured by observational sensors (bottom) (Petry et al., 2014). From the figure, it is possible to realize the enhancing the electron concentration from the night period up to the afternoon, showing good similarities between the simulation and observations. More details can be seen in the paper from Petry and co-authors (2014).

As mentioned in the previous Section, several compilations were executed to the SUPIM. After the compilations with different levels for optimization directives, the code was worked to change each routine identified with higher processing demands. Each routine was modified

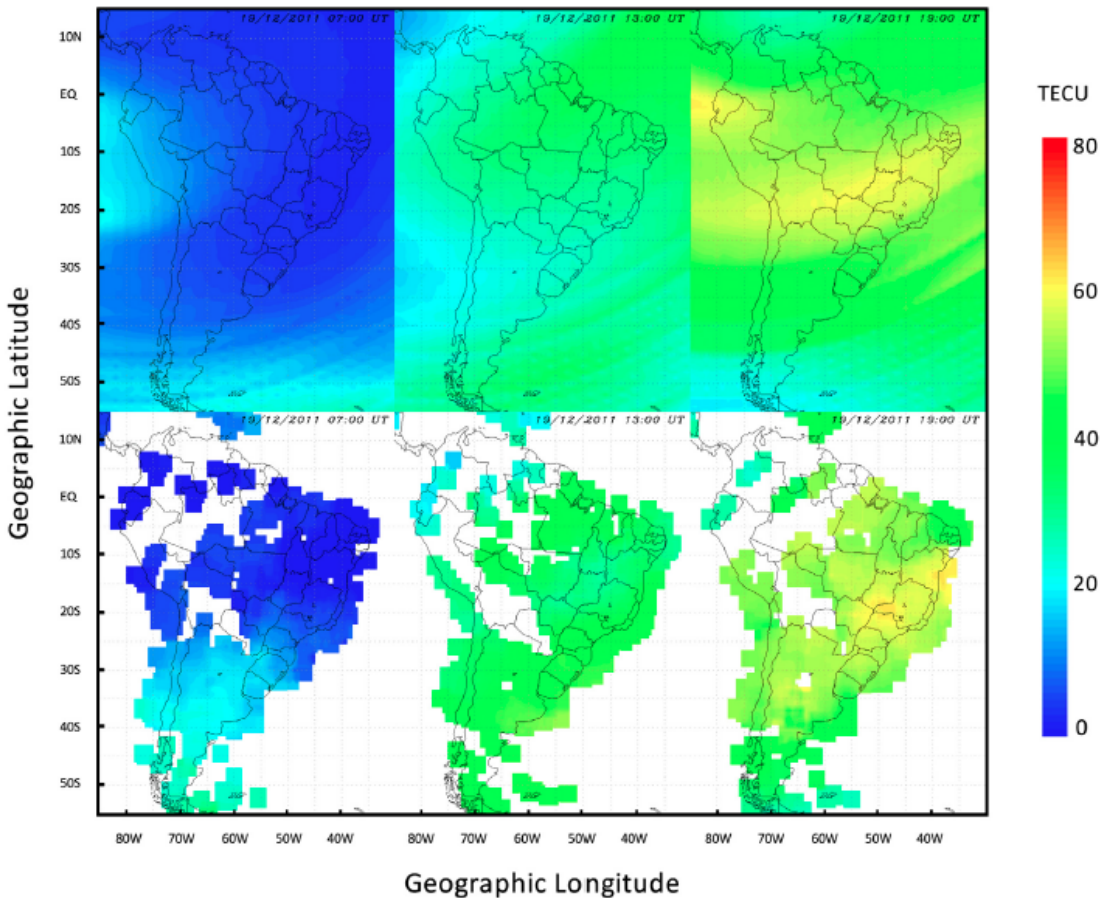

Figure 2- VTEC map from the SUPIM-DAVS simulation (upper) and ground-based GPS stations (bottom) at 7, 13, and 19 h UT on December 19th, 2011.

separately. Basically, variables were identified and changed into higher-dimensional vectors, and sometimes the order of the loops was changed to allow a better parallel processing.

Table 2 shows the CPU-time (hours:minutes:seconds) under different options for parallelism to the SUPIM. The compilation with directives O1, O2, and O3 did a significant improvement to the SUPIM performance, reducing $\sim$ 39 minutes for the CPU-time. A harder work was required for dealing with the optimizations of the routines because these imply changes for the code programming. However, a significative better performance was obtained: additional 22 minutes for speeding-up after optimization of all listed routines (Table 1) – see the CPU-time in the last row of Table 2 considering all optimizations.

Table 2- Parallel optimizations and CPU-time.

Optimazation type	CPU-time
Without optimization	01:18:50
Optimization directives	00:39:47
Optimized Globe7	00:26:06
Optimized GTS7	00:23:20
Optimized DENSU	00:20:25
Optimized SPLIN	00:17:30

4. CONCLUSIONS

SUPIM was a computer code originally codified for a sequential computer machine. The first parallel version tried to use a hybrid parallel programming standards: OpenMP and MPI (Petry, 2010). However, even using the exponential decay – see Eq. (1), sometimes unstable behaviour was registered. So, several SUPIM's copies are executed independently in a parallel way by using the OAR package (https://oar.imag.fr/) for jobs scheduling. The parallel SUPIM execution on a cluster with multi-core CPUs presents 4730 seconds for 24 hours of simulation. The SUPIM implementation on the NEC SX-Aurora TSUBASA vector processor allow us to obtain a signifitive reduction of the CPU-time to 1050 seconds: our SUPIM vector version is $\sim$ 4.5 faster than the sequential version.

In future work, we have the intention to analyze the power consumption for the SUPIM execution on the cluster with multi-core CPUs and energy demands by vector version on X-Aurora TSUBASA vector computer, in a similar way done by Komatsu et al. (2021).

Acknowledgements

The authors want to thank the NEC Corporation for allowing us to deal with the NEC SX-Tsubasa computer (Japan), at the first moment, and the NEC Latin-American (NEC LatAm) team for their agreement for our use of the vector machine in Brazil – in special, authors want to thank Mrs. Fernanda S. Mendes and Mr. Carlos B. Siqueira. Authors thank to the National Council for Scientific and Technological Development (CNPq, Brazil) for the research grant (CNPq: 422614/2021-1), and author HFCV would also like to thank the CNPq (Brazil) for the research grant (CNPq: 312924/2017-8).

REFERENCES

Bailey, G. J., Sellek, R. (1990), A mathematical model of the Earth's plasmasphere and its application in a study of He+ at L = 3. Annales Geophysicae, 8, 171–189.

Bailey, G. J., Sellek, R., Rippeth, Y. (1993), A modelling study of the equatorial topside ionosphere. Annales Geophysicae. 11, 263—272.

Bailey, G. J., Balan, N. (1996), A low-latitude ionosphere–plasmasphere model. In: Schunk, R.W. (Ed.), "STEP Handbook on Ionospheric Models". Utah State University, pp. 173–206.

Ikubanni, S. O., Adeniyi, J. O. (2017), Relationship between ionospheric F2-layer critical frequency, F10.7, and F10.7P around African EIA trough. Advances in Space Research, 59, 1014–1022

Komatsu, K., Momose, S., Isobe, Y., Watanabe, O., Musa, A., Yokokawa, M., Aoyama, T., Sato, M, Kobayashi. H. (2018): Performance evaluation of a vector supercomputer SX-Aurora Tsubasa. In SC'18: International Conference on High Performance Computing, Networking, Storage and Analysis – IEEE, 685–696. IEEE.

Komatsu, K., Onodera, A., Focht, E., Fujimoto, S., Isobe, Y., Momose, S., Sato, M., Kobayashi, H. (2021): Performance and power analysis of a vector computing system. Supercomputing Frontiers and Innovations, 8 – DOI: 10.14529/jsfi210205 (open access on the internet).

Petry, A. (2010), "Análise e paralelização do código SUPIM". São José dos Campos: [s.n.], 2010. Available at: http://urlib.net/8JMKD3MGP7W/37SMSTS.

Petry, A., Souza, J. R., Campos Velho, H. F., Pereira, A. G., Bailey G. J. (2014): First results of operational ionospheric dynamics prediction for the Brazilian space weather program. Advances in Space Research, 54, 22–36.

Souza, J. R., Abdu, M. A., Batista, I. S., Bailey, G. J. (2000). Determination of vertical plasma drift and meridional wind using the Sheffield University plasmasphere ionosphere model and ionospheric data at equatorial and low latitudes in Brazil: summer solar minimum and maximum conditions. Journal of Geophysical Research. 105, 12813-–12821.

Souza, J. R., Brum, C. G. M., Abdu, M. A., Batista, I. S., Asevedo Jr, W. D., Bailey, G. J., Bittencourt, J. A. (2010), Parameterized regional ionospheric model and a comparison of its results with experimental data and IRI representations. Advances in Space Research, 46, 1032–1038.