iJET EXPLORE International Journal of Engineering & Technology
CENTRE OF PROFESSIONAL RESEARCH PUBLICATIONS
ISSN 2049-3444

International Journal of Engineering and Technology Volume 2 No. 11, November, 2012

Parallel Computation of Spatio-Temporal Variations of Geomagnetic Field Using One-Minute Time Resolution Dataset

Felix Ale[1], Oye Ibidapo-Obe[1], Theophilus A. Fashanu[1], Olufemi A. Agboola[2]
[1]Department of Systems Engineering, University of Lagos, Nigeria
[2]Department of Engineering and Space Systems,
National Space Research & Development Agency, Abuja, Nigeria

ABSTRACT

This work presents a novel method for analyzing spatio-temporal variations of geomagnetic field using parallel computing. One-minute resolution geomagnetic dataset of 1996 was obtained from INTERMAGNET global network of 64 observatory stations. In effect, large and three dimensional arrays of geomagnetic observations are to be processed. Thus, sequential and parallel algorithms were developed using MATLAB 2012a, interfaced with the well-known kriging method for efficient geostatistical data gridding and mapping of solar quiet (Sq) daily variations. The runtime of the sequential algorithm on a processor took 18.5 minutes while the corresponding parallel algorithm took 2.95 minutes using eight Intel Xeon E5410 2.33GHz processors in parallel. The efficiency profile of the model is logarithmic in nature. This was further optimized using quadratic polynomial interpolation. The results show that 13 processors will process the Sq in less than one minute, thus providing effective near real-time observation of Space weather. In addition, the foci of the generated finer Sq(H) plots revealed temporal variability that is consistently maximized at local noon at every location on this geomagnetic environment as demonstrated in the 2D visual display.

Keywords: *Parallel Computing, Multi-core technology, Geomagnetic Field, Performance, Benchmark, Space Weather*

1. INTRODUCTION

Although, parallel computing has been an age long method in computing, however its effects, applications and versatility have not been adequately explored and exploited by the developing nations as an emerging and effective tool for solving some challenging scientific and engineering problems. Parallel computing which is the thrust of high-performance computing, involves the use of multiple high capacity processing units such as processors, memories, faster interconnect and programming structures as workhorse for solving complex and/or large problems concurrently. This approach facilitates and optimizes timely computational results.

Michael (1994) and George et al (1994) defined parallel computing as an aspect of computer science that deals with system architecture and software issues related to concurrent execution of applications. Supercomputer with shared memory technology was the forerunner of parallel computing in the mid-1950's. Later in the 1980's, Caltech Concurrent Computation project built a supercomputer for scientific applications from 64 Intel 8086/8087 processors with higher performance off the shelf microprocessors that were also known as massively parallel processors (MPPs). ASCI Red supercomputer was invented in 1997 with one trillion floating point operations per second in addition to clusters computing technology which was introduced in the 1990's. Ever since, parallel computing with varying architectures has continued to grow in processing speed, memory and interconnectivity.

Very affordable multi-core technology now comes with modern desktop and laptop systems. Chip manufacturers have begun to increase overall processing performance by adding additional CPU cores. The reason is that increasing performance through parallel processing can be far more energy-efficient than increasing microprocessor clock frequencies since the profile of the latter is asymptotic in nature. Fortunately, the continued transistor scaling predicted by Moore's Law will allow for a transition from a few cores to many (Intel, 2012). The shift in computing paradigm from sequential to parallel programming was necessitated by the yet evolving parallel hardware of miniaturized processors and memory technologies by the semiconductor industries.

The need to process large applications and dataset also gave rise to the advent of parallel computation. However, parallel programming and algorithms that take advantage of parallel hardware are more complex to develop compared with the corresponding sequential ones. This is due to the complexity of inter-processors communication along with synchronization of concurrent tasks and shared data. Al Geist et. al (1994) explained Message passing interface (MPI) and Parallel Virtual Machine (PVM) as platform independent standard interfaces that were

developed in 1990's to ease the development of parallel programs for massively parallel programs and clusters. Additionally, several parallel programming models and languages have been developed that are based on non-parallel languages such as the native C for the development of parallel programs with efficient and optimized solutions. Modern applications of parallel processing include computational fluid dynamics, thermal analysis of data centre, simulation of energy and heat equations deriving from the Navier-Stokes' as well as heliophysical studies. In addition, orbital tracking of navigation satellites, ephemeris data acquisition, photorealistic graphics, computational visualization and machine perception are other specialties that require high level of parallel algorithms to produce optimized results in a reasonable time.

The theme of this work is the application of parallel processors to solve for solar quiet (Sq) variations using one-minute resolution of geomagnetic field data. The generated Sq daily variations results from electric field dynamo at the E-region of ionosphere is known to be directly responsible for the magnetic perturbation on the Earth surface being measured by ground-based magnetometers. The Sq focus helps to observe the intensity of solar radiations and the geomagnetic field variability in the ionosphere. The overall conditions of the Sun in the interplanetary Space and Sun-Earth environment are generally known as Space weather. It has been established in a study carried out by James (2007) and Pekunlu (1999) that solar radiation takes 2-3 days while solar flares effect take less than 9 minutes to reach Earth.

Solar activities are known to be responsible for sudden explosion of some energetic particles which lead to solar flares, coronal mass ejection or magnetic storms depending on their magnitude and frequency (Hargreaves, 1979), (Oslen, 1996). Evidently, human lives and activities now depend largely on space-borne technological systems and applications such as satellite, GPS, mobile GSM, internet banking, weather forecast among others. Consequently, adverse effects of Space weather often inflict damages on space-borne and ground-based technological systems, thereby compromising socio-economic well-being. Therefore, there is a need to continuously improve on Space weather forecast for the realization of near real-time prediction of solar activities for adequate and efficient mitigation of natural disasters emanating from solar flares.

The paper is structured as follows. In the next section, we describe briefly the techniques employed in this work. Section 3 shows method of analysis and Sq parallel algorithm. Section 4 show the model and other results obtained, Section 5 discusses the results and we conclude the paper in Section 6.

2. EVALUATION METRICS FOR HIGH-PERFORMANCE COMPUTING

Today, high performance (parallel) computers (HPC) are being used in a wide variety of disciplines. These disciplines include financial modeling in business industry, the meteorological prediction of tornadoes and thunderstorms, analysis of DNA sequences in computational biology. The pharmaceutical companies also use parallel computing in the design of new drugs, oil companies carry out seismic survey and exploration using parallel computation. NASA uses them for aerospace vehicle design, while the entertainment industry uses them for special effects in movies and commercials. The common attribute of these complex financial, scientific and engineering designs applications is the need to perform computations on large datasets and/or complex equations.

For performance analysis of parallel computing, Amdahl (1967) and Gustafson (1988) stated benchmark metrics for evaluation. Essentially, the speedup, ***Sp***, determines how many times the parallel program is faster than the fastest sequential using identical problem and hardware specifications. Other benchmark metrics include efficiency,$\boldsymbol{\lambda}$, which is the ratio of speedup to the number of processors (p), used for the computation. These major metrics can be mathematically represented as follows:

$$S_p = \frac{T_s}{T_p} \quad or \quad \frac{T_s}{T_1} \tag{1}$$

$$\boldsymbol{\lambda}_p = \frac{S_p}{P} * 100\ \% \tag{2}$$

Where Ts and Tp are the execution time for fastest sequential and parallel programs respectively.

The metrics often used to evaluate the performance of a well-configured parallel computing architecture are execution time, speedup, efficiency and cost. The execution time is the time elapsed when first processor starts the execution and when the last processor completes it. This time include the computation time, communication time and idle time.

Although much research work has been on the analysis of Sq using seasonal, hourly and diurnal dataset, it is obvious that more phenomena can be properly observed when using one-minute time resolution data. Such work were carried out by Chapman (1951), Chapman and Rajarao (1965), Campbell and Shiffmacher (1987), Onwumechili (1997), Okeke et al(1998), Rabiu (2001), Osborne (1964). Till date, due to challenges involved in parallel computation, a one-shot simulation for Sq foci using one-minute magnetic data has not been achieved. A major constraint in this regard is the multi-dimensional nature of the components of geomagnetic field. To explain this

problem, Campbell (1997) described the orientation of geomagnetic field components in as shown in fig.1.

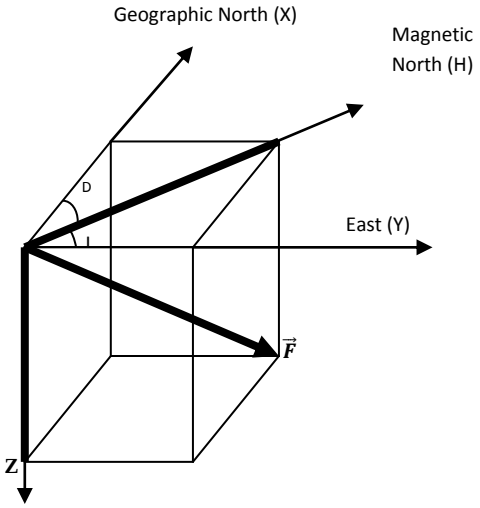

Fig.1: Components of Geomagnetic field

3. PRE-PROCESSING AND DATA ANALYSIS

The primary data for this work was obtained from INTERMAGNET global network. The Network involved 64 stations from 20 countries across the globe in 1996. One-minute time resolution geomagnetic dataset was extracted and converted to horizontal, vertical and angular declination orientation. In algebraic form, equation (3) shows the breakdown of the matrices of the computation in the sub-tasks.

$$\begin{matrix}\text{Cases} \\ \left\{\begin{matrix}64 \times 3 \\ 70272 \times 1440 \\ 20304 \times 1440 \\ 20304 \times 1440\end{matrix}\right.\end{matrix} \quad \begin{matrix}\text{Components} \\ \left\{\begin{matrix}H \\ D \\ Z\end{matrix}\right.\end{matrix} = \begin{matrix}\text{Objects} \\ \left\{\begin{matrix}SqH \\ SqD \\ SqZ\end{matrix}\right.\end{matrix} \tag{3}$$

Out of the 64 observatories, 44 stations represented their geomagnetic field dataset in horizontal or magnetic north (H), angle of declination (D), vertical (Z), total intensity (F) components while 20 stations represented the dataset in geographic north (X), geographic east (Y), vertical (Z), total intensity (F) according to magnetometer sensors orientation at the observatories. The relationships between the geographic and magnetic orientations were stated by Vestine (1947) in equations (4) and (5) as well as Campbell (1997) in figure 1. The numerical values of the components were generated and recorded in time frame by ground-based magnetometers. The importance of the equations is to use geomagnetic orientation (HDZ) as uniform platform for the computation.

$$F^2 = X^2 + Y^2 + Z^2 \tag{4}$$

$$H^2 = X^2 + Y^2,\ H = \sqrt{X^2 + Y^2} \tag{5}$$

3.1 Framework for the Computational Analysis

To proceed, we present in Table 1, the configuration and specifications of the high-performance parallel computing hardware that is deployed in this work.

Table 1: Systems Parameters

Feature	Specification
Model	Dell PowerEdge 2950
Processor clock speed	Intel(R) Xeon(R) CPU E5410 @ 2.33GHz (8 CPUs)
Front-side Bus speed	1333MHz
L2 Cache	4MB per core
Memory	16378MB RAM
Operating System	Windows Server® 2008 HPC Edition (6.0, Build 6001) (64-bit)

For the algorithm, Matlab m-files and functions were developed with the implementation of kriging method for geostatistical data gridding using weighted least square approximation.

The details of the algorithm developed and the inter-processor communications are shown in figure 2.

Table 2: Sequential to Sq parallel algorithm methodology

Serial computation	Parallel computation
tic % start CPU timer	
	Matlabpool p open
	parfor i=1:n
For n = 1:n	*% slicing, indexing variables*
$t_s = \sum_{1}^{n} Sq_n$	$t_p = \max_n(Sq_n)$
end	*End*
toc % stop CPU timer	*Matlabpool close*

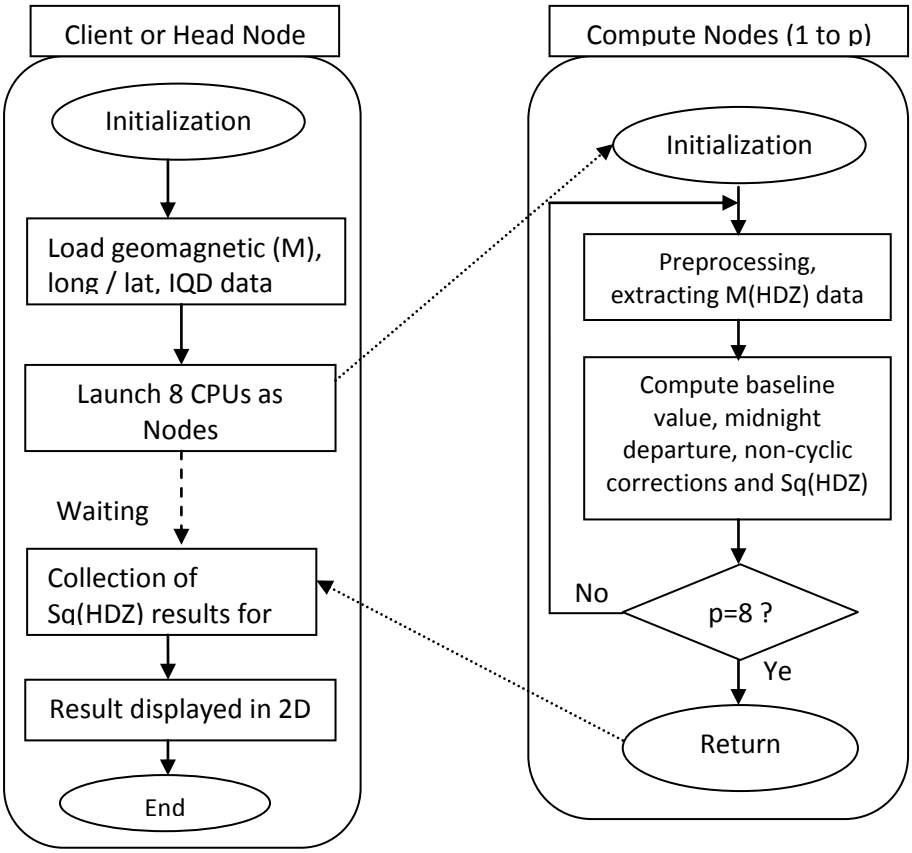

Fig.2: Sq parallel algorithm

Summarily, the overall framework for this work is as shown in the figure 3.

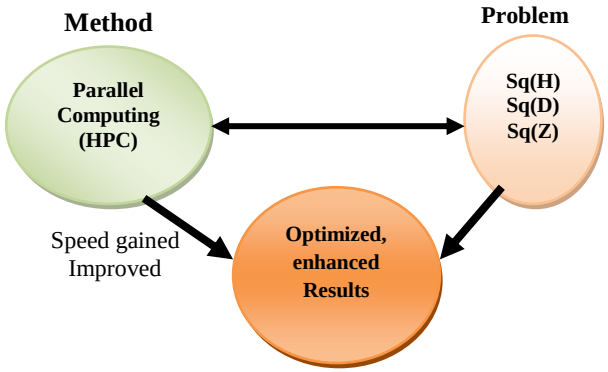

Fig.3: Framework of Sq the computation

3.2 Tasks Allocation and Matrices Partitioning

Load balancing and concurrency were identified as major problems in the algorithm optimization. One of the solutions to load balancing is by partitioning large matrices for equal assignment of work to the processors in parallel. The algorithm shown below was developed to balance the load of processing of Sq among the eight (8) processors. The success of task allocation minimized the processor idle time and time spent on inter-processor communication. To proceed, dimensions of acquired dataset, matrices i.e. load for processing are highlighted below.

- Set of matrices used
    - 64 x 3 (for H, D, Z components by 64 observatories)
    - 192 by 1440 (276480 cases)
    - 366 x 276480
- For the partitioning procedure, we employ the following format:

Let V = (366-by-276480) matrix

V was distributed to the eight (8) CPUs, a MATLAB algorithm was developed to partition the two-dimensional array horizontally into segments and assigns one segment of the array to each CPU. Consequently, the two-dimensional array was partitioned horizontally (by rows) by assigning columns of the V-array of 366-by- 34560 to each CPU. The algorithm below shows how V-array was distributed to eight (8) CPUs.

```
spmd

X = zeros(366, 34560);
Y = codistributed(X);
# with the use of 8 processors (p)
computeNode 1: This node stores Y(:,1: 34560);
computeNode 2: This node stores Y(:,34561: 69120);
computeNode 3: This node stores Y (:,69121: 103680);
computeNode 4: This node stores Y(:,103681: 138240);
computeNode 5: This node stores Y(:,138241: 172800);
computeNode 6: This node stores Y(:,172801: 207360);
computeNode 7: This node stores Y(:,207361: 241920);
computeNode 8: This node stores Y(:,241921: 276480);

end
```

This algorithm created a 366-by-276480 replicated array locally and assigned it to variable X. The second command distributes X and created Y = 366-by-34560 single array that got distributed to the eight (8) CPUs. Each CPU has access to all segments of the array as a local variable so as to run faster rather than sending/ receiving data between CPUs which would consume more time.

The generic algorithm responsible for this task is given as: matlabpool open config p

```
        spmd
A = zeros(n, m);
B = codistributed(X);
# with the use of 8 processors (p)
a=m/p;  z=0;z1=1;
For p=1:8
z = z+a;  computeNode p: This node stores B(:,z1:z);
                z1=z+1;
end
        end
        matlabpool close
```

The developed sequential code was modified by removing data dependencies and other inhibitors of parallelism. This way, some variables were sliced and indexed to permit parallelization.

3.3 Performance Analysis of Sq Parallel Algorithm

The execution time with respect to the number of processors used is stated in table 3. The execution time is the total of computation time and overhead time.

Table 3: The Observed Execution Time

Processor (ρ)	Time(sec) (t)
1	801.549
2	409.737
3	302.236
4	242.909
5	218.074
6	208.901
7	192.848
8	185.517

As an outcome of table 3, eight (ρ, t) data points were obtained. Where ρ = number of process, t = time frame.

3.4 Regression Analysis Using Least Square and Pearson's Methods

The correlation coefficient, R, measures the strength and direction of linear relationships between the observed runtime (t_1) and the predicted time (t_2). The coefficient of determination, R^2, measures the percentage of variation in the dependent variable that is explained by the regression or trend line. It has a value between zero and one. Here, a high value indicates a good fit. The equation for the Pearson product moment correlation coefficient, R, is given as:

$$= \frac{\sum(t_1 - \overline{t_1})(t_2 - \overline{t_2})}{\sqrt{\sum(t_1 - \overline{t_1})^2 \sum(t_2 - \overline{t_2})^2}}, \quad 0 \leq R \leq 1 \qquad (6)$$

The coefficient of determination is R^2 as indicated on the plots.

Non-linear least squares is the form of least squares analysis which is used to fit a set of *m* observations with a model that is non-linear in *n* unknown parameters ($m > n$). It is used in some forms of non-linear regression. The basis of the method is to approximate the model by a linear curve and to refine the parameters by successive iterations. The experimental result in Table 3 can be expressed in form (ρ_k, t_k) where k = 1,2,3, …, 8 is independent variable and, *t*, are dependent as shown in table 4.

Table 4: Dependent variable vs independent variable

ρ_k, t_k
(1, 801.55)
(2, 409.74)
(3, 302.24)
(4, 242.91
(5, 218.07)
(6, 208.90)
(7, 192.85)
(8, 185.52)

The graph plot of (ρ, t) shows a logarithmic curve which modeled a best-fit curved line to display the data values that decrease quickly before leveling out. Mathematically, the relationship conforms the following logarithmic function to compute the least squares fit through the observed points:

$$t = \alpha_1 \ln(p) + \alpha_2 \quad (7)$$

where α_1 and α_2 are constants that approximately solve the over-determined non-linear system.

Clearly, this is an over-determined system i.e. having eight (8) equations with two (2) unknowns. The least squares approach to solving this problem is to try to make as small as possible the sum of squares of errors between the right- and left-hand sides of the equations, that is, to find the minimum of the function.

Thus, the generated over-determined system can be expressed as:

$$t_i = \sum_{j=1}^{n} \rho_{ij}\, \alpha_j \qquad where \quad (i = 1,2,\dots,8)$$

The over-determined system is of *m=8* linear equations in *n=2* unknown coefficients, α_1, α_2, with *m* > *n*. This can be written in compact matrix form as:

$$\mathbf{t} = \boldsymbol{\rho} * \boldsymbol{\alpha} \quad (8)$$

where

$$\rho = \begin{pmatrix} \rho_{11} & \rho_{12} & \cdots & \rho_{1n} \\ \rho_{21} & \rho_{22} & \cdots & \rho_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{m1} & \rho_{m2} & \cdots & \rho_{mn} \end{pmatrix}, \ \alpha = \begin{pmatrix} \alpha_1 \\ \vdots \\ \vdots \\ \alpha_n \end{pmatrix}, \ t = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_m \end{pmatrix} \quad (9)$$

Here, the objective function *S* is given by

$$S(\alpha) = \sum_{i=1}^{m} |t_i - \sum_{j=1}^{n} \rho_{ij}\, \alpha_j|^2 = ||t - \rho\alpha||^2 \quad (10)$$

This minimization problem has a unique solution, provided that ρ is a full ranked matrix. The solution is given by the standard procedure:

$$(\boldsymbol{\rho^{-1}\rho})\hat{\boldsymbol{\rho}} = \boldsymbol{\rho^{-1}t} \quad (11)$$

In this case, the i[th] error or residual is given as:

$$r_i = t_i - \sum_{j=1}^{n} \rho_{ij}\, \alpha_j$$

Then *S* can be expressed as:

$$S = \sum_{i=1}^{m} r_i^2$$

Note that *S* is minimum when the gradient vanished. Since the model contains *n* parameters there are *n* gradient equations, so, elements of the gradient vector are the partial derivatives of *S* with respect to the parameters, that is:

$$\frac{\partial S}{\partial \alpha_j} = 2\sum_{i=1}^{m} r_i \frac{\partial r_i}{\partial \alpha_j} = 0 \ \ where \ (j = 1,2,\dots,n)$$

The minimum is determined by calculating the partial derivatives of $S(\alpha_1,\alpha_2)$ with respect to α_1 and α_2 and setting them to zero. This results in a system of two equations in two unknowns, called the normal equations, which results in

α_1 = -276.2 and α_2 = 686.4,

Consequently, t = -276.2ln(ρ) + 686.4 (12)

4. RESULTS OF THE WORK

In this section, results of the sequential and parallel runtimes of the developed algorithms using single and eight (8) processors respectively are presented. Speedup and efficiency as benchmark metrics are stated.

Table 5: Speedup and Efficiency of Processors utilization

CPU (*P*)	Time(s) (T_p)	Time (minute, T_p)	Speedup (S_p)	Efficiency (%)
1	801.549	13.40	1	100
2	409.737	6.83	1.96	97.8126
3	302.236	5.04	2.65	88.4021
4	242.909	4.05	3.30	82.4948
5	218.074	3.63	3.68	73.5117
6	208.901	3.48	3.84	63.9497
7	192.848	3.21	4.16	59.3768
8	185.517	2.95	4.32	54.0079

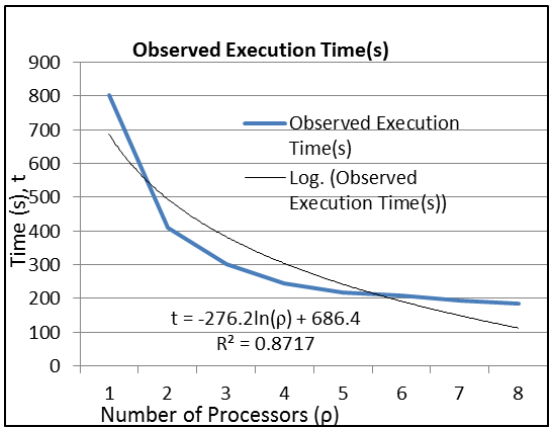

Fig.4: Execution time Vs number of Processors

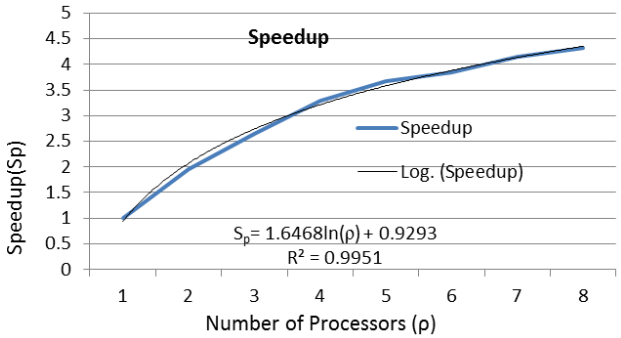

Fig.5: Execution time, Speedup Vs number of processors allocation

The new model exhibits coefficient of determination (R^2) = 0.8717, which indicates a strong linear relationship between the two time-series variables as indicated on the graph.

It was observed in this study that speedup, S_p = 4.32. This showed that the parallel algorithm is four times faster than the sequential algorithm. Thus Sp is less than p (S_p <p). However, theoretically, $S_p \leq p$. The difference can be attributed to parallel overhead.

To further explain the strength of parallel computing, the values of the performance metrics are predicted as follows:

$$t = -276.2\ln(\rho) + 686.4 \quad (15)$$

Using the absolute value,

$$t = abs(-276.2\ln(\rho) + 686.4) \quad (16)$$

In this regard, dataset performance metric values are given in Table 6 below.

Table 6: Performance metrics for the parallel algorithm

CPU (ρ)	Execution Time(s)	Time (min)	Speedup	Efficiency (%)	Logarithmic Model	Optimized 2nd order Polynomial
1	801.55	13.36	1.00	100.0	686.40	594.32
2	409.74	6.83	1.96	97.81	494.95	502.09
3	302.24	5.04	2.65	88.40	382.96	417.79
4	242.91	4.05	3.30	82.49	303.51	341.40
5	218.07	3.63	3.68	73.51	241.87	272.95
6	208.90	3.48	3.84	63.95	191.52	212.42
7	192.85	3.21	4.16	59.38	148.94	159.82
8	185.52	2.95	4.32	54.01	112.06	115.14
9	unobserved				79.53	78.39
10	unobserved				50.43	49.56
11	unobserved				24.10	28.66
12	unobserved				0.07	15.68
13	unobserved				22.04	10.63
14	unobserved				42.51	13.51
15	unobserved				61.56	24.31
16	unobserved				79.39	43.04
17	unobserved				96.13	69.69
18	unobserved				111.92	104.27

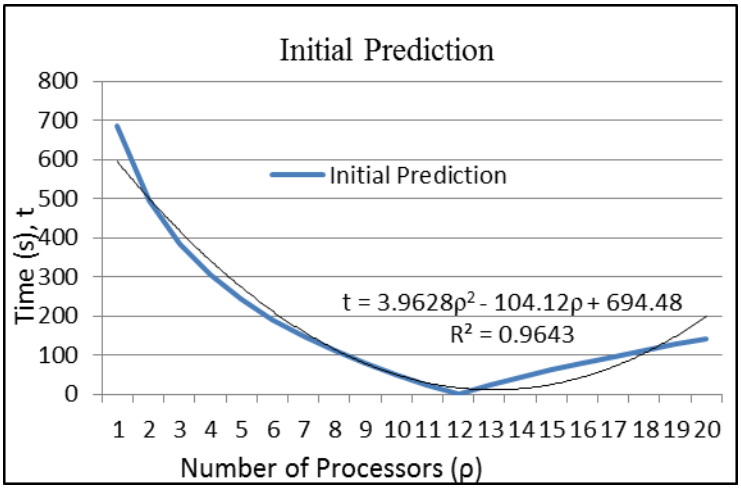

Fig.6: Prediction accuracy vs number of processor

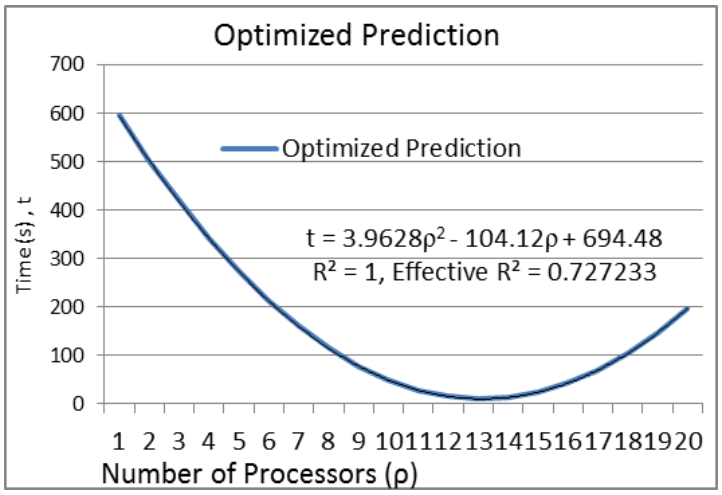

Fig.7: Determination of optimum number of processors for accurate prediction

To obtain the optimum parameter, p (number of processors, we consider the derivatives of their absolute values in the model as follows:

Derivative of the absolute values of the model gives,

$$t = \frac{d}{dt}(3.9628\rho 2 - 104.12\rho + 694.48)$$

This simplifies to

$7.9256\rho - 104.12 = 0$ for the optimum value of p.

The second order polynomial above describes the optimized processors usage and execution time. Hence, the maximum number of processors useful for this computation would be given as the value of p when the first derivative of the quadratic vanished;

That is,

$\frac{dT_p}{dP} = 0$, p = 13;

This implies that parallel slowdown occurs when at values p, the first derivative is greater than zero.

$\frac{dT_p}{dP} \geq 0$, p ≥ 13;

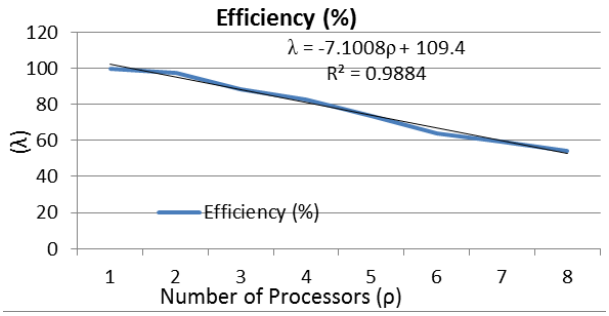

Fig.8: Efficiency of the parallel algorithm

Furthermore, it was observed in fig. 8 that the runtimes of the parallel algorithm decrease as the number of processors increases as expressed below:

$$p \propto \frac{1}{T_s} \qquad and \qquad T_s = f(\beta + \alpha)\ as\ Ts \rightarrow 0$$

Where β = workload (Sq computation) or size of the problem and α = parallel overload (p_o)

Results of Sq variations obtained from the application of our developed parallel computing algorithm are shown in figures 7 to 10.

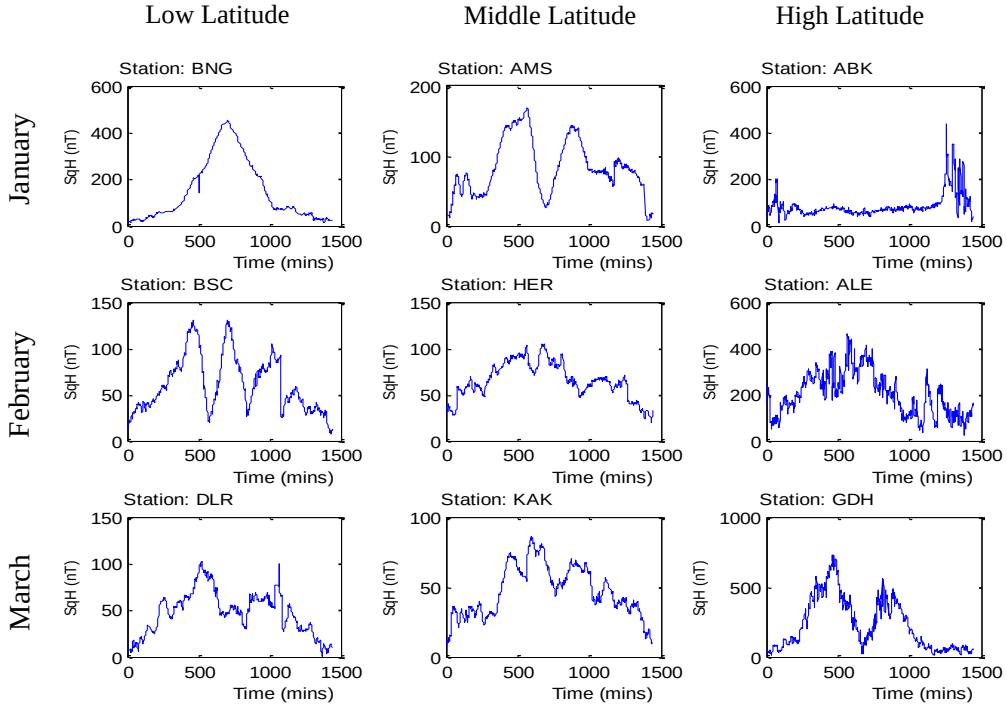

Fig.7: Sq(H) variations in January, February and March along latitudinal zones

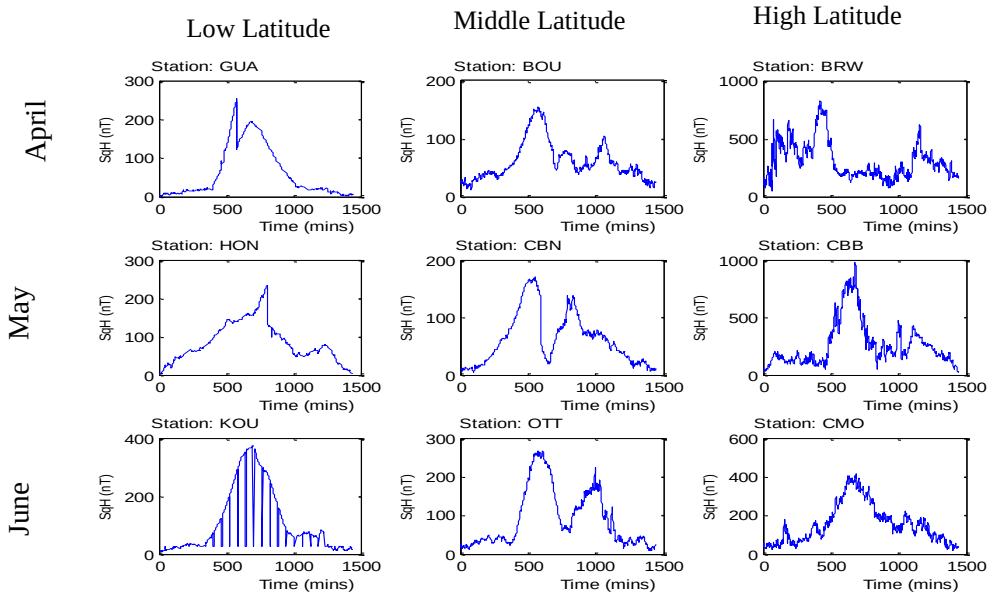

Fig.8: Sq(H) variations in April, May and June along the latitudes

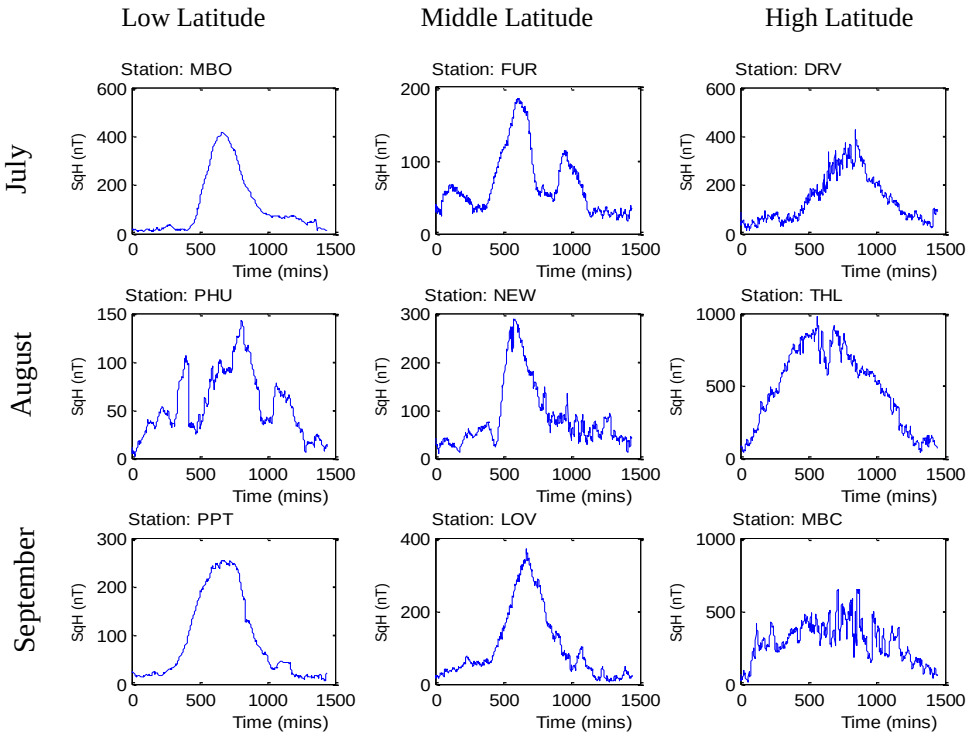

Fig. 9: Sq(H) variations in July, August and September along the latitudes

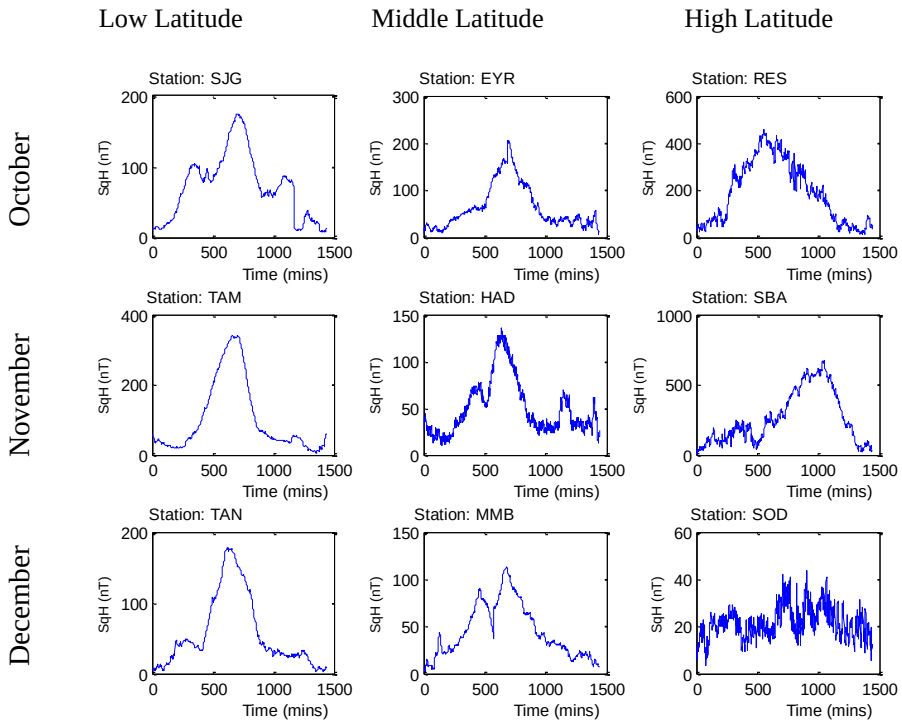

Fig.10: Sq(H) variations in October, November and December along the latitudes

5. DISCUSSION OF RESULTS

The results showed that execution time of running serial Sq algorithm on a sequential processor took 18.5 minutes while the runtime for corresponding parallel algorithm on eight parallel processors took 2.95 minutes. The model of parallel runtimes is logarithmic and over-determined in nature. Further analysis by extrapolation into few more points showed that 13 processors would generate runtime of less than one minute, thus providing optimum computing solution in terms of resources and time. The result of the extrapolation using least square method conformed to the differentiation of the logarithmic function at the minimum value. This was evident by the graphical representation of the speedup which showed that the parallel algorithm was four (4) times faster than sequential ones. The results also showed that the use of more than 13 processors to solve the Sq problem would generate parallel slowdown and under-utilization of processors and other computing resources due to increase in time for inter-processors communication and synchronization. Contrary to an expected speedup of eight (8), the overall speedup is 4.32 due to parallel overhead. At that level, the total performance gain would decrease. Thus, this reconciles Amdah's (1967) and Gustafason's laws (1988) about the diminishing effects of size and overhead of parallel computing.

On the other hand, computation of Sq variations using our developed algorithm showed that Sq focus exhibits variability and maximized at local noon of the observatory stations locality as demonstrated in the 2-D graphical representation. The stations were classified into three latitudnal zones viz – low latitude, mid-latitude and high latitude. Figure 7 shows the Sq(H) variations in 1st quarter of year 1996), we observed in January that BNG station (4.44^0) in the low latitude, the curve soared higher at noon (720minutes = 12noon local time), however the Sq(H) curve reduced towards evening and night. Evidently, this is due to intense activity of the Sun along the equatorial region when the Sun was directly overhead at noon lost focus with it at night. At the high latitude region, the curves of stations like ABK (Abisko, Sweeden – 68.36^0), being close to polar region, were rough and dwindled as a result of limited exposure to the Sun especially in January. Station AMS which, is located in the southern hemisphere in latitude (-37.83^0), had its curve nose-dived at noon (720min) due to reduced solar activities while its behavior was similar to KAK when observed in March.

Fig.8 shows the graphical representations of Sq(H) in 2D for the second quarter. It could be generally inferred that the minute variations of Sq(H) would enhance further understanding and observations of the Sq(H) variation at various seasons, monthly, hourly and on minute basis. It was observed that there is a relationship between the latitude amplitude and months in the year 1996. At the low latitude, the amplitude of Sq(H) dropped from about 400 in the first quarter of the year to about 300 in second quarter of the year while the values were averagely constant in the high latitude regions. At the stations' local time, the sun would always be overhead thereby having more intensity on the area compared with other minutes of the day as well as in the average over months.

Fig.10 shows the Sq(H) values along the low-, mid- and high-latitudes among the selected stations in the last quarter of the year 1996. It was evident that solar activities vary depending on solar intensity as well as other heliophysical phenomena such as cloud cover, total electron content, equatorial electrojet.

In general, we observe that the local time at each station witnessed higher amplitude in the Sq(H) variation across the whole latitudinal zones. The jigsaw, saw-toothed, zig-zagged curves are as a result of solar activities that release geomagnetic storm, solar flares, coronal mass ejection etc which generally characterize harsh nature of space weather.

6. CONCLUSION

Although solar quiet daily variations have been processed using sectorial and large time range, this study used parallel algorithm and parallel hardware to solve for Sq variations using higher time resolution of one-minute geomagnetic field dataset. Solar effect on ionosphere causes current that reflect as Sq, which in turn characterize condition of space weather. Adverse space weather can cause failure of technological systems. Thus, in this study, Sq variation as a phenomenon of space weather was simulated using parallel computation. Microscopic view of Sq substantiated the work of Chi and Russell (2005), *Manoj et al (2006)*, Courtillot and *Chulliat* (2008), Love (2008) and Love et al (2008) about the need for high time resolution for easy prediction of ionospheric travelling disturbance. The runtimes, speedup and efficiency of the parallel computation largely depend on problem size and load balancing. This optimal number of processors is essential for effective parallel computation.

REFERENCES

[1] Amdahl, G.M., "Validity of the Single-Processor Approach to Achieving Large-Scale

[2] Computing Capabilities," *Proc. Am. Federation of Information Processing Societies Conf*., AFIPS Press, 1967, pp. 483-485.

[3] Al Geist, Adam Beguelin, Jack Dongarra, Weicheng Jiang, Robert Manchek, Vaidy

[4] Sunderam, 1994, PVM: Parallel Virtual Machine - A User's Guide and Tutorial for Networked Parallel Computing. MIT Press.

[5] Campbell,W.H. & Shiffmacher, E.R. (1987), Quiet ionospheric currents and earth conductivity profile computed from quite-time geomagnetic field changes in the region of Australia. Aust. J. Phys. 40, 73-87

[6] Campbell,W.H,1997, Introduction to Geomagnetic Fields, Cambridge University Press.

[7] Chapman, S., The equatorial electrojet as detected from the abnormal electric current distribution above Huancayo, Peru, and elsewhere, *Arch.Meteorol. Geophys. Bioclimatol.* A, 4, 368–390, 1951.

[8] Chapman, S. and K. O. Rajarao, The *H* and *Z* variation along and near equatorial electrojet in India, Africa and the Pacific, *J. Atmos. Terr. Phys.*, 27, 559–581, 1965.

[9] Chi P.J. and C.T. Russell, 2005, Travel-time magnetoseismology: Magnetospheric sounding by timing the tremors in space. *Geophys. Res. Lett.* , 32.

[10] *Courtillot* V. and A. *Chulliat* (eds.), Observations magnétiques - Magnetic results 2005,

[11] Bulletin No 25, Bureau Central de Magnétisme Terrestre, Paris, *2008*.

[12] George S. Almasi and Allan Gottlieb, 1994, *Highly Parallel Computing, Second Edition* Benjamin/Cummings Publishers

[13] Gustafson, J.L., "Reevaluating Amdahl's Law," *Comm. ACM*, May 1988, pp. 532-533.

[14] Intel http://www.intel.com/pressroom/kits/upcrc/, Accessed, May 10 2012

[15] Hargreaves, J. (1979), *The Upper Atmosphere and Solar Terrestrial Relations. An*

[16] *Introduction to Aerospace Environment*, New York: Van Nostrand Reinhold Company Ltd.

[17] *Love J. 2008.* Magnetic monitoring of Earth and space, Physics Today, 61, 31-37.

[18] *Love*, J. J., Applegate, D. & Townshend, *J* .B., *2008*. Monitoring the Earth's Dynamic Magnetic Field, USGS Fact Sheet, 2007-3092.

[19] *Manoj, C., Kuvshinov, A., Maus, S., Lühr, H. (2006):* Ocean circulation generated *magnetic* signals. - Earth Planets and Space, 58, 4

[20] Michael J. Quinn, 1994, *Parallel Computing Theory and Practice*, McGraw-Hill, Inc.,

[21] keke, F. N., C. A. Onwumechili, and B. A. Rabiu, Day-to-day variability of geomagnetic hourly amplitudes at low latitudes,*Geophys.J.Int.*,134,484–500, 1998.

[22] Onwumechili, C. A., Spatial and temporal distributions of ionospheric currents in sub-solar elevations, *J. Atmos. Terr. Phys.*, 59, 1891–1899, 1997

[23] Osborne, D. (1964). Daily and seasonal changes of equatorial electrojet in Peru. *J. atmos. terr. phys , 26*, 1097-1105.

[24] Oslen, N. (1996). Magnetospheric contributions to geomagnetic daily variations. *Ann. Geophysicae , 14*, 538 - 544.

[25] Rabiu A.B., 2001, Seasonal Variability of Solar Quiet at Middle Latitudes, Ghana *Journal of* Science, 41, 15-22

[26] Vestine, E. H., Description of the earth's main magnetic field and its secular change, 1905-1945, Carnegie Institute of Washington Publication 578. Washington. D. C., 1947.

[27] Rennan Pekunlu (1999), Solar Flares, Tr. J. of Physics International Workshop: 23 (1999), 415-423.

[28] James A. Marusek, 2007, Solar Storm Threat Analysis, Impact, Bloomfield, Indiana 47424