Parallelizing Electroencephalogram Processing on a Many-Core Platform for the Detection of High Frequency Oscillations

Gildo Torres, Paul McCall, Chen Liu, Mercedes Cabrерizo, Malek Adjouadi
Department of Electrical & Computer Engineering
Florida International University
Miami, Florida, USA
E-mail: {gtorr029, pmccat001, cliu, cabreriz, adjouadi}@fiu.edu

Abstract— EEG high frequency oscillations, known as ripples, in subdural Electroencephalography (EEG) have been associated to the seizure onset zone. Ripples, which can be visible in the frequency range from 80 to 250 Hz, are considered reliable biomarkers (like interictal EEG spikes) to identify the epileptic focus in the brain. Consequently, an automated detection method is proposed with the aim of identifying those electrodes that have a higher count of these events. The computational approach considered relies on processing the EEG records using the Intel Single-Chip Cloud Computer (SCC) platform. This new method preserves data coherency through the message-passing interface and utilizes dynamic voltage and frequency scaling (DVFS) capability of SCC, yielding both energy saving and performance benefit. The proposed SCC-based method for detecting high frequency oscillations (HFO) is validated by EEG experts at Miami Children's Hospital, and the location of the electrodes with higher counts will be compared with the 3-D source localization using interictal spikes to demonstrate the relation if any that exists between them.

Keywords: Cloud computing, Intel SCC, EEG processing, High frequency oscillation, Power-aware computing

I. INTRODUCTION

Epilepsy is a common medical condition characterized by a predisposition to unprovoked recurrent seizures. A seizure is the manifestation of an abnormal, hypersynchronous discharge of a population of cortical neurons [1]. Affecting over 60 million people around the world, Epilepsy is the second most frequent neurological disorder other than stroke.

Advanced clinical techniques are used to diagnose epilepsy, such as computed tomography (CT), encephalogram (EEG), magnetic resonance imaging (MRI), positron emission tomography (PET), and functional MRI, along with others. While the aforementioned techniques yield a coarse approximation of the epileptogenic region, they otherwise lack either the spatial or temporal resolution necessary to accurately determine the seizure focus location. When this is the case, invasive recording techniques such as intercranial-EEG (iEEG) or Electrocorticography (ECoG), which are characterized by the placement of electrode arrays on the cortex of the brain, are performed.

During ECoG recording it is routine clinical practice to place multiple electrode arrays on different areas of interest to the neurologists. Multiple arrays act to both eliminate certain cortical regions of interest and designate cortical areas where more analysis may be needed. For example, it is customary for a patient undergoing ECoG to have sixty-five or more implanted electrodes.

Within Epilepsy research, many relevant algorithms are useful throughout the process of localization of the seizure onset zone (SOZ). These include interictal spike detection, seizure onset detection, artifact detection and elimination, high frequency oscillations (HFO) detection as well as many others [2-6]. For HFO detection patients are monitored throughout the night and sleep ECoG is recorded for up to 10 hours. What’s more, high resolution clinical-use EEG machines have an eclipsing sampling rate of 2KHz. Due to these characteristics, it can be seen that any extensive time recordings will yield large amount of data which requires enormous computing power.

In the last several decades, we have seen how microprocessor performance have been dramatically improved by increasing the operating frequency, from 5MHz of Intel 8086 to the astounding 5.2GHz of IBM z196 [7]. Unfortunately, in recent years, power-thermal issues have limited the pace at which processor frequency can be increased [8]. In an effort to utilize the abundant transistor real estate offered by the Moore’s Law [9] and at the same time contain the power-thermal issues, current developments in microprocessor design favor increasing core counts over frequency scaling to improve processor performance and energy efficiency [10].

In the commercial field, it is common to have a 2, 4, 6 or even more cores housed in one chip nowadays. While the research community makes use of experimental many-core architectures containing tens or even hundreds of processors. Today, the challenge is not only how to develop powerful hardware architectures that satisfy the demands of high resources-consuming applications, but also the development of applications that could effectively explore the capabilities offered by many-core architectures. There are major benefits that can be obtained from parallel programs running on many-core platforms.
The Single-Chip Cloud Computer (SCC) experimental processor [10] is a 48-core "concept vehicle" created by Intel Labs as a platform for many-core software research. This system allows the implementation and study of parallel applications by supporting a message-passing programming model for communication among the cores. The SCC also includes hardware elements which support dynamic voltage and frequency scaling (DVFS) for improving energy efficiency.

As we mentioned earlier, ECoG recording and HFO detection would not be easily processed by desktop computers or even specialized software. Due to the limitations of the processing platforms that are available to researchers in this field, currently only a sub-section of this data, ranging between a few seconds to a few minutes, will be subjected to analysis while the rest of the data is discarded. In this paper we will illustrate the benefits of utilizing the SCC as the platform of choice for EEG algorithm implementation, by demonstrating the energy savings and computational benefits associated with Intel's 48-core Single-chip Cloud Computer.

The rest of the paper is organized as follows. In Section II we present an overview of the SCC platform. Section III contains the methodology we propose for processing EEG data on the Intel SCC platform and Section IV describes the algorithm implemented for the detection of HFOs. In Section V we present and comment on the obtained results and we conclude in Section VI and give an outlook for future work.

II. COMPUTATION PLATFORM

The Single-Chip Cloud Computer contains 48 Pentium™ class IA-32 cores on a 6x4 2D-mesh network of tiled core clusters with high-speed I/Os on the periphery [10]. There is a unique hardware feature called Message Passing Buffer (MPB), shared by every two cores, that is optimized to support message passing programming model to communicate among all the cores.

The SCC platform used in this research can be considered a computational benefit to almost any parallel application performed on it due to the fact that it possesses 48 cores. However, there are two main benefits that make the SCC platform suitable for EEG signal processing: its inter-core communication or message-passing abilities, and the capacity for DVFS. These two aspects of the SCC make EEG processing a promising application because they address two problems that are inherent to processing of this type.

The first difficulty that occurs when processing EEG data is that a significant amount of time and energy is consumed upon the access and distribution of the data. This problem is magnified within a parallel architecture [11]; while a few cores are accessing and loading the data, many of the cores are running at the same power levels without contributing to the overall progress. As shown in Figure 1, every two cores form a frequency island and every eight cores form a voltage island. The SCC allows the user to fully control these voltage and frequency islands that are present on the chip.

In this way we can act to minimize wasted energy by setting cores to lower power states while accessing large amounts of data. A similar approach has been demonstrated by the use of multiple voltage-frequency gears that run at different segments throughout the program in order to maximize performance while saving energy in a PC cluster setting [12].

The second issue with EEG data is that there tends to be global parameters for most algorithm implementations. This means that the processing of one electrode may be dependent on a parameter defined by another electrode or a group of electrodes. This is due to the aggregate nature of the EEG signal itself. The EEG signal stems from a summation of neuronal activity; therefore a single phenomenon may have components in many surrounding electrodes. A program which distinguishes a particular activity of interest, such as interictal spikes, may need information from numerous electrode signals in order to confidently detect their presence within the data set. This establishes a need for effective and user-controllable inter-core and thus inter-electrode communication. When such algorithms are run in a parallel manner, this becomes a more detrimental issue. Without explicit user defined communication protocols, there would potentially be cache coherency issues and/or memory allocation issues due to variables growing inside loops. The SCC allows the user, through use of the RCCE library [13] (an API library for message passing programming model specifically designed for SCC), the ability to control and synchronize inter-core communication due to the message-passing benefits of the SCC platform.

III. METHODOLOGY

The proposed method for implementing EEG algorithms on the SCC platform is shown in Figure 2 and outlined below.

1. Electrode dependencies and any need for global parameters are identified. This knowledge will lead to an understanding of the appropriate inter-core communication that will be necessary for proper execution.

2. Once inter-core communication is understood, the program needs to be broken down into segments. These segments are divided into two categories,
communication-intensive and computation-intensive. Parts of code that accesses the data and distributes appropriate data to proper cores would be considered communication-intensive while parts of code that utilize many processing cores for filtering or detection of any kind would be considered computation-intensive. These segments would then be run using the appropriate Voltage-Frequency gear that would result in performance benefits while maintaining energy savings.

3. Once there is an awareness of the necessary communication between cores and the code has been segmented and assigned correct gears, the code needs to be modified with the correct RCCE library functions so as to be implemented and executed on the SCC platform.

The EEG data and results that are presented within this paper are gathered from a patient who was monitored overnight with multiple electrode arrays placed on his cortex totaling sixty-five electrodes at the Brain Institute in Miami Children’s Hospital (MCH) [14]. In the validation of this algorithm, 10-minute segments were analyzed consisting of 32 electrodes. This number of electrodes was analyzed for each run of the simulation because this is a preliminary analysis of EEG processing on the SCC platform. Processing of more electrode data will be discussed later in the Future Work portion of this paper. The sampling rate for data acquisition is 2 kHz. HFO results were validated by neurologists at MCH.

IV. HFO – EEG ALGORITHM

HFOs have been defined as spontaneous patterns in the range of 80 – 500 Hz that consists of at least 4 oscillations which can be distinguished from the background. However, this is not a quantitative definition, thus making accurate detection of HFOs both difficult and subjective. HFOs can be visually marked but tend to be highly time consuming, on the order of hours for the analysis of a few minutes of data [15]. Research has suggested that HFOs are possibly related with epileptogenesis [16]. Electrodes of interest, which correspond to the SOZ, have higher relative ripple counts when compared with electrodes that are associated with other cortex regions of normal neuronal activity [17, 18].

The definition used in this paper for the HFO detector is listed below and illustrated in Figure 3.

- HFOs are within the 80 – 250 Hz frequency band
- A global threshold based on standard deviation of a selected electrode is determined
- Three or more crossings of the global threshold within a 250 ms window will count as a HFO

![Figure 2. Proposed EEG-SCC Methodology](image)

![Figure 3. HFO Detection Program Executed on SCC](image)

Raw EEG data is passed into the SCC program, parsed up and distributed to the appropriate cores. Upon receiving the data, each core implements a 10th-order Butterworth IIR filter using cascaded Second Order Direct Form II sections. Once the filtering process is completed, a global maximum value is taken across all electrode signals. This electrode-max value is used to normalize the entire set of electrode readings. A sample segment of data is shown in Figure 4 after filtering and normalization.
After normalization, a standard deviation calculation is performed on the electrode signal that produced the electrode-max value. The global threshold defined for the program is calculated as a multiple of this standard deviation value. This threshold parameter shows that even for the simplest EEG algorithms, such as a rudimentary HFO detector, there exists a fundamental need to efficiently process and pass data between cores while the program is analyzing the data.

V. Results

This work acts to demonstrate the feasibility of the HFO algorithm implementation for EEG analysis on the SCC platform. In order to verify the correctness of our implementation, relative HFO counts per electrode were compared with the findings previously validated by neurologists at Children’s Hospital and found to match, with all electrode signals of interest being identified. The results were also validated when electrode locations with higher HFO counts from the algorithm were compared with 3D source localization using interictal spikes. This analysis was performed for 10-minute segments of data with all simulations taking less than 40-seconds of processing time. It is worth noting that when a replica of our algorithm is ran on Matlab, removing all Matlab optimized function calls, the processing time exceeded thirty minutes for the same amount of data.

As shown in Figure 3, the program is classified into the load region and the execution region. The load region includes the section of the program where the data file is being loaded and split by the master core. The data is then distributed to each processing core. This region was classified as a communication-intensive region, because the code spends most of the time accessing memory or transmitting information to other cores. Therefore, a high processing frequency is not required.

The execution region, explained in the previous section, is where the ripple-detection algorithm is executed. We classified this region as a computation-intensive region, where a high processing frequency is required.

We tested different setups of the SCC platform where the number of cores employed varied while three Frequency-Voltage configurations were used, as shown in Table 1. The Frequency-Voltage schemes tested were HIGH (800 MHz and 1.1V), LOW (533 MHz and 0.8V), the third and fourth one were mixed approaches, where the low gear was applied to the load region and the high gear to the execution region in the third configuration that we called MIX, and for the fourth one the high gear was applied first and then low for what we called the XIM gear. For both mixed gears we dynamically changed the Voltage and Frequency values during the transition between regions. The numbers of cores tested were 1, 2, 4, 8, 16, and 32.

The elapsed time when switching between different levels of voltage and frequency is not significant compared to the time consumed by each region, therefore it has no impact on the timing results associated with either region or the total execution.

It can be seen that total processing time for all simulation runs does not exceed 40 seconds. While this is a major improvement over serial Matlab implementation, it does not directly demonstrate the need for the parallel platform. This is because our proposed method is not specifically aimed at this algorithm implementation. The HFO algorithm is a simple computation task for the SCC, and immediate benefits lie in the DVFS and message-passing abilities of the system. However, it can be seen that as the execution region becomes more burdensome, the benefits of parallelizing the program can be easily seen. We can assume that processing benefits associated with the SCC based EEG analysis will become more substantial when running more complex algorithms on the platform.

From Figure 5 we can observe how significant the communication region is when compared to the total time for the simulations with the exception of the simulations completed with one core where no communication overhead exists because all the data manipulation is local to one core. This result shows the HFO detection program spends most of its execution time in communicating and transferring data among the cores. We can see as the number of cores increases, the communication overhead increases, even the time for completing the execution region decreases. Here the

<table>
<thead>
<tr>
<th>Gear</th>
<th>Voltage</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>HIGH</td>
<td>1.1 V</td>
<td>800 MHz</td>
</tr>
<tr>
<td>LOW</td>
<td>0.8 V</td>
<td>533 MHz</td>
</tr>
<tr>
<td>MIX</td>
<td>0.8/1.1 V</td>
<td>533/800 MHz</td>
</tr>
<tr>
<td>XIM</td>
<td>1.1/0.8 V</td>
<td>800/533 MHz</td>
</tr>
</tbody>
</table>

Table 1. Voltage - Frequency Gears
communication overhead has more weight than the computation overhead. So “the more the merrier” seems not work here. The implementation of the MIX gear tries to match the communication-intensive region with low power dissipation gear and the execution-intensive region with a high processing frequency in order to improve the power-energy efficiency without sacrificing performance. Based on the nature of the specific program we are running, we decided to try the XIM gear where we executed the communication-intensive region with high frequency and the execution-intensive region with the low frequency gear. We discuss our findings below.

As we expected, the MIX gear provides a performance in between the HIGH and LOW gear for the simulations run with fewer amount of cores, where T1 is lower and T2 bigger compared to those of simulations completed employing more cores. In the other hand, when using the XIM gear, we can see how the performance compared to the MIX gear improves as the number of cores increases. Executed with one core is the only case where the load and the execution region performances are in similar order. This happens because in this case we avoid the overhead of data transmission and because of the delay it takes the execution region to be completed. For optimal performance, we see that the best configuration for the SCC is having 8 cores running with the HIGH gear.

Differences in power consumption for both regions of the program are demonstrated in Figure 6. HIGH gear always consumes more power than LOW gear due to higher frequency and voltage. In the MIX gear mode, the execution region consumes similar power as HIGH gear mode, while load region consumes similar power as LOW gear mode, due to the dynamic change of operating frequency. The opposite happens for the XIM gear. From this graph we can observe that the execution region always consumes more power than the load region, as more computation is required. As we increase the number of cores, there is a resulting increase in power consumption associated with both the HIGH and the LOW gear. If optimal power is required, either 1 or 2 cores running with the LOW gear might be the right choice.

The energy consumed by each region, as well as the total energy consumption for each simulation, is shown in Figure 7. This graph reaffirms that the energy consumed by a system is dependent on the balance between performance and power, not simply processor speed or power dissipation alone. From this graph we can conclude that for energy considerations, the HIGH gear running with 1 and 2 cores is the most energy efficient configuration for this specific application. This is because in our HFO application, the program spends most of its execution time in communicating and transferring data among the cores, as we saw from Figure 5. When increasing the number of cores, the communication overhead and the time required for completing the load region increases as well as the energy associated with this part of the program.

The metric for measuring power-performance is very common. From the system point of view, minimizing the execution time may usually be the first priority. However, even if the total energy is minimized, the user may not be satisfied with extended system response time [19-20]. In Figure 8 we present the energy-delay product (EDP) for each used configuration. EDP metric takes into consideration both energy and execution time. This graph shows how both mixed gears generally have an EDP between the HIGH and the LOW gear as we anticipated. One thing worth mention is that in this case, similarly to the energy analysis, the HIGH
gear running with 1 and 2 cores provides the best of both worlds, user experience and energy consumption.

From observing the presented performance, power, and energy results, it is evident that adaptive scheduling of the voltage and frequency may reduce power dissipation and energy consumption without sacrificing performance significantly.

VI. CONCLUSIONS AND FUTURE WORK

The performance benefits of utilizing the SCC platform for HFO detection are substantial. The SCC can process data while implementing complex algorithms in short periods of time. In the area of HFO detection alone this is a remarkable advance. This would allow HFO analysis to be performed on extended durations of recordings, in hours instead of minutes. This increase in data processing capability will act as an analysis tool for neurosurgeons and neuroscientists in order to define the SOZ with higher resolution and confidence.

The advantages associated with EEG processing on the SCC platform can be employed on a myriad of EEG signal processing algorithms. The SCC proves to be an ideal platform on which to process multiple electrode recordings in an energy-efficient manner, while increasing the performance of analysis as a whole. The DVFS capabilities of the SCC allow the user to have full control of energy usage which can lead to total system energy savings when the code can be broken up into communication-intensive and computation-intensive segments and run with the appropriate Voltage-Frequency gears. The message-passing architecture, both at the hardware and software level of the SCC, allow for user-defined inter-core and therefore inter-electrode communication, which has been shown to be essential for EEG algorithms.

All blocks pertaining to the HFO-EEG algorithm are implemented and running on the SCC platform. This algorithm is not computationally burdensome to the SCC and is handled in a very efficient manner. As future work, we expect to implement a program been able to process 65 electrodes overall and more than 32 at a time. Further research can be done implementing more complex EEG detection algorithms. Other areas of interest would be systems and EEG algorithms that incorporate more message-passing or electrode-dependencies than are currently present in the HFO algorithm. The SCC platform allows for the development and implementation of more complex, data-dependent algorithms in which further neuronal phenomena can be examined. A system approach in which multiple algorithms are executed in parallel on the SCC would be beneficial to the field of neuroscience and epilepsy research as well.

ACKNOWLEDGMENT

This work is partly supported by the National Science Foundation and the Department of Defense (DoD) through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program. The authors are also grateful for the support provided by the National Science Foundation under grants ECCS-1125762, CNS-0959985, CNS-1042341, and HRD-0833093. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or Department of Defense.

REFERENCES


