Pinky: Interactively Analyzing Large EEG Datasets

by

Rohan Mahajan

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Computer Science and Engineering

at the Massachusetts Institute of Technology

June 2016

Author . . . .
Department of Electrical Engineering and Computer Science
June 4, 2016

Certified by . . . .
Prof. Matei Zaharia
Thesis Supervisor

Accepted by . . . .
Dr. Christopher J. Terman
Chairman, Masters of Engineering Thesis Committee

Pinky: Interactively Analyzing Large EEG Datasets

by

Rohan Mahajan

Submitted to the Department of Electrical Engineering and Computer Science
on June 4, 2016, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Computer Science and Engineering

Abstract

In this thesis, I describe a system I designed and implemented for interactively analyzing large electroencephalogram (EEG) datasets. Trained experts, known as encephalographers, analyze EEG data to determine if a patient has experienced an epileptic seizure. Since EEG analysis is time intensive for large datasets, there is a growing corpus of unanalyzed EEG data. Fast analysis is essential for building a set of example data of EEG results, allowing doctors to quickly classify the behavior of future EEG scans. My system aims to reduce the cost of analysis by providing near real-time interaction with the datasets. The system has three optimized layers handling the storage, computation, and visualization of the data. I evaluate the design choices for each layer and compare three different implementations across different workloads.

Acknowledgments

This work is dedicated to Herbert Blum.

First, I would like to thank my family for their enduring support and love throughout my time at MIT.

I would like to thank Professor Sam Madden, Dr. Brandon Westover and Professor Mark Silberstein for their guidance and support while advising me throughout this project. Their insights and suggestions greatly helped shape this work.

I would also like to thank Amir Watad, Sagi Shahar, and Feras Daoud for helping me have a home away from home while collaborating at the Technion. At MIT, my work would never have been completed if not for the great friendship and support of Tal Tchwella, Stephanie Wang, Max Kanter and Neha Patki. I would also like to thank Adam Marcus, Lydia Gu, and Eugene Wu for our initial discussions of research topics and continuing support throughout the project.

In addition, I would like to acknowledge collaboration with Stavros Papadopoulos on the TileDB project, Stephanie Wang with the Visgoth system, Siddharth Biswal and his help with algorithms for processing EEGs, Bastian Bechtold and his WebGL-Spectrogram implementation, and Ole Christian Eidheim for support with the websocket server.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A number of applications require a domain expert to visually inspect and process a stream of incoming data. The problem with manual inspection is the inability to scale as datasets grow exponentially [19]. As the dataset grows, it becomes difficult to visualize interactively [39]. In this thesis we focus on medical data, where doctors have to analyze a patient's data and extract relevant information for treatment. Specifically, we focus on electroencephalogram (EEG) readings, a test which used to detect abnormalities related to the electrical activity of the brain.

Today, doctors store large amounts of patient data that they cannot analyze because they lack tools to efficiently view datasets at scale. To address this issue, we have designed and implemented Pinky, a system for processing large amounts of EEG data, allowing near real-time interactive analysis.

1.1 Pinky

Pinky is a doctor's newest tool for analyzing the brain, see Figure 1-1. Working with a team of researchers at Massachusetts General Hospital (MGH), we have designed and implemented the system to handle the fast growing corpus of collected EEG data. This end-to-end system handles the storage, processing, and visualization of EEG data. The goal of the system is to provide a scalable architecture for concurrent analysis of patient records with near real-time interactivity. Each layer of the system is optimized for use and evaluated across hundreds of gigabytes of patient data.

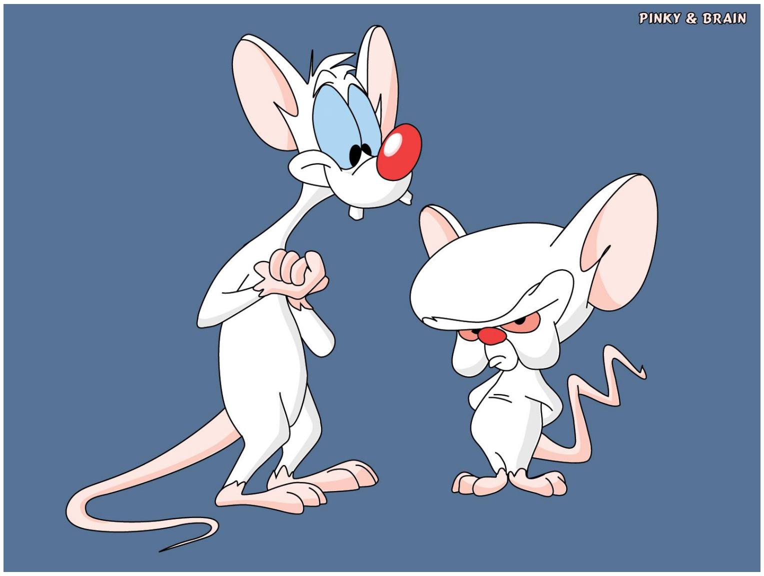


Figure 1-1: Etymology of Pinky's name.

1.2 Overview of EEG Analysis

A seizure is a transient aberration in the brain's electrical activity. People with the central nervous system disorder epilepsy suffer from recurrent seizures, often happening suddenly and at unpredictable times. A seizure can vary from a lapse of attention to a whole-body convulsion. Frequent seizures are dangerous, as they can increase risk of sustaining physical injuries and can even result in death [29].

One method for detecting the onset of epileptic seizures is the analysis of scalp EEG data, a non-invasive measure of the brain's electrical activity. Continuous EEG (cEEG) data is typically recorded using 19 silver/silver chloride electrodes, affixed to the scalp [36]. Figure 1-2 shows a drawing of the placement of sensors on a patient's scalp.

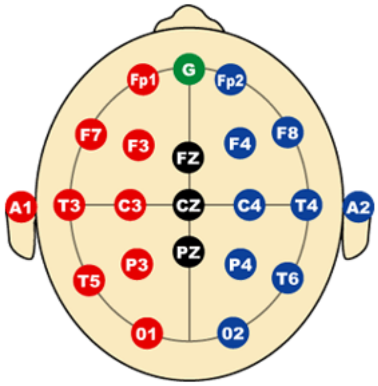
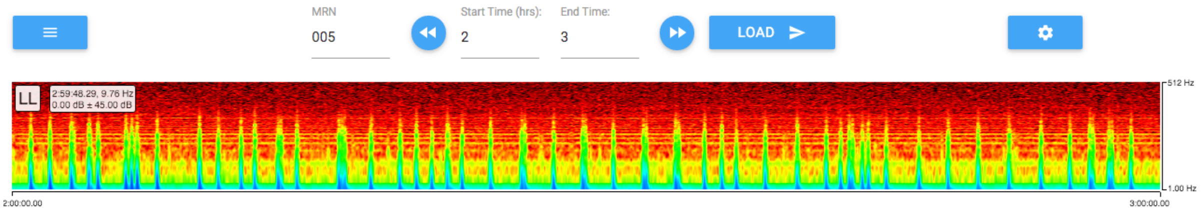


Figure 1-2: EEG electrode placement on a patient's scalp.

Trained individuals, such as attending physicians, epilepsy/neurophysiology fellows, or registered EEG technicians (encephalographers), review and screen EEG recordings, which typically take place over a continuous 24-hour period [10]. Unlike traditional epilepsy monitoring units which focus on provoking and capturing seizures, the goal of cEEG studies is to efficiently identify future seizures and prevent them. This leads to an increase in the number of cEEG recordings for preventative measures. Intensive care unit centers are subsequently overwhelmed with the analysis of the growing dataset due to the small number of available trained individuals. Methods to screen long EEG recordings without sacrificing accuracy are necessary to be able to efficiently process this data.

Typically, EEGs displays show no more than 10 to 15 seconds of data per screen of raw voltage readings and requires an analyst to simultaneously inspect multiple channels. In contrast, a compressed spectral array [9] or spectrogram display may show 2 to 8 hours of data on a single color map [10]. This allows analysts to quickly screen long periods of EEG data, determining which segments, if any, require direct review of the raw data. Spectrogram review reduces cEEG review time by 78% [26], with minimal loss of sensitivity compared with conventional review. For these reasons, we focus on building a tool to rapidly analyze spectrogram data.

Spectrograms are the most widely used compressed data format for EEG data [36]. A spectrogram consists of three-dimensional plots with time on the x-axis, frequency on the



MRN: 005

Start Time (hrs): 2

End Time: 3

Figure 1-3: Spectrogram for one hour window of EEG data of the `LL` region of the brain.

y-axis, and EEG power on the z-axis. Figure 1-3 shows an example of rendered spectrogram data. An analyst typically views four spectrograms concurrently, mapped to different regions of the brain. Each region is formed by using multiple EEG channels where an EEG channel is the difference between voltages measured at two electrodes. This captures the summed potential of millions of neurons [29]. Figure 1-2 shows the electrode placement on the patient's scalp, yielding four regions for analysis: left lateral power, `LL`, (Fp1-F7, F7-T3, T3-T5, T5-O1), left parasagittal power, `LP`, (Fp1-F3, F3-C3, C3-P3, P3-O1), right lateral power, `RL`, (Fp2-F8, F8-T4, T4-T6, T6-O2), right parasagittal power, `RP`, (Fp1-F4, F3-C4, C4-P4, P4-O2).

Data from a single patient can vary in size from tens to hundreds of gigabytes and the number of EEG tests performed each year is estimated to be between 10 and 25 million [15]. As this corpus of data collected at the ICU continues to grow, efficient mechanisms to store and visualize this data at scale are key for analysts to quickly view patient screenings. Pinky aims to provide this for analysts by giving them a simple yet powerful interface to view spectrogram data.

1.3 System Architecture

Pinky is comprised of three coupled layers which handle storage, computation and visualization. Figure 1-4 shows the overall architecture of the system.

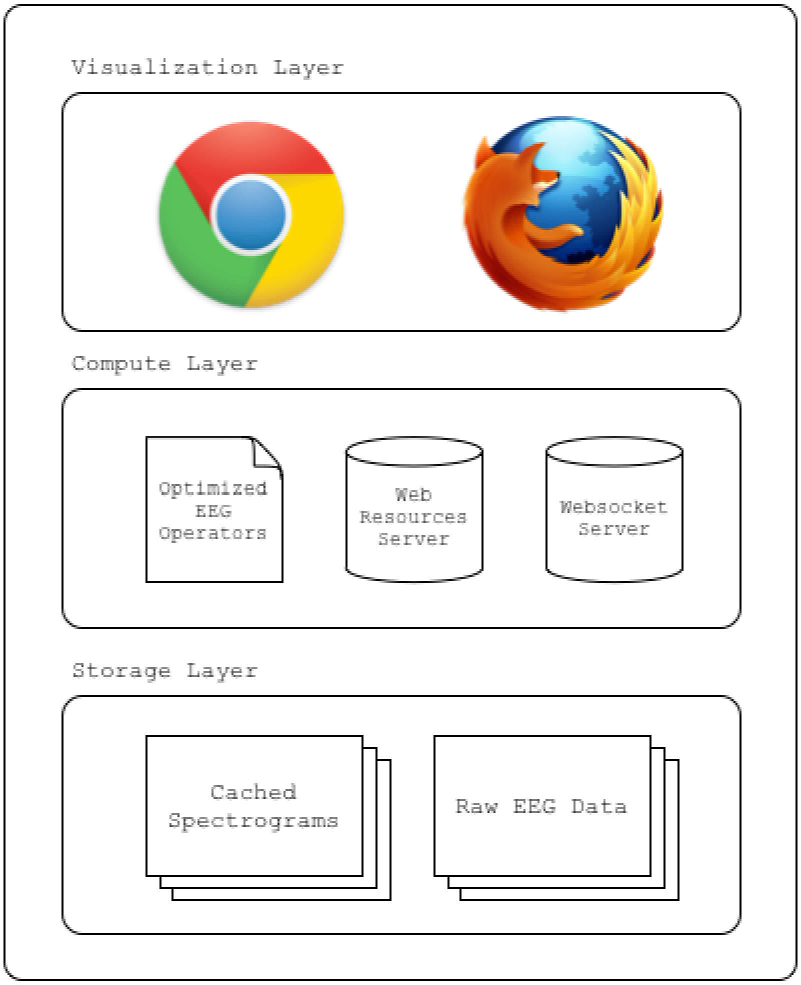


Figure 1-4: Pinky system architecture.

1.3.1 Storage Layer

The storage layer, discussed in detail in Chapter **??**, is responsible for storing raw EEG patient data and the calculated spectrogram. This datastore must optimize both reads and writes of array based data for multidimensional arrays on the order of tens to hundreds of gigabytes.

1.3.2 Compute Layer

The compute layer, discussed in detail in Chapter **??**, is an extensible module which handles the algorithms to calculate the spectrogram and other EEG related calculations. As we discuss in Section **??**, there are a number of extensions the project can take, thus it is important that an interested developer can easily add functionality to this layer. In addition, the compute layer contains two servers. One server interfaces with the optimized EEG algorithms and the storage layer to serve array based data. The second server is a lightweight server for the web resources of the visualization layer.

1.3.3 Visualization Layer

The visualization layer, discussed in detail in Chapter **??**, is a browser based module that renders the data to the client. The interface allows users to query based on a patient's id (medical record number, `mrn`) and view a spectrogram for a given time interval. An analyst may smoothly pan and zoom throughout the dataset.

1.3.4 Visgoth System

Since enabling interactivity is an important design criteria, we have designed and built an optimization module for browser based visualizations named Visgoth. The system uses profiling information from the client and server to suggest an adaptive scaling of the visualizations served in order to keep latency consistent, regardless of a client's hardware or network bandwidth. We discuss Visgoth in detail in Chapter **??**.

1.4 Usage

The project code base is available publicly on Github [17] at `https://github.com/joshblum/eeg-toolkit`, with documentation for installing the project for development. In addition, we have created Docker [25] images that can easily be installed for production use. Armed with a dataset, any curious doctor is able to install the images and load the data for analysis. The docker images are available for public use on DockerHub: `https://hub.docker.com/r/joshblum/eeg-toolkit-webapp` and `https://hub.docker.com/r/joshblum/eeg-toolkit-toolkit`. The Github project contains specific installation instructions.

1.5 Contributions

Pinky makes the following contributions:

- Implements an abstraction for array based storage systems.
- Implements three different backends which adhere to the abstraction.
- Evaluates the different backends for varying input ranges and workloads.
- Implements optimized algorithms for analyzing EEG data.
- Provides an extensible framework for accessing array based data and visualizing it in the browser.
- Implements scalable in-browser visualizations using the client's GPU.
- Implements a new system, Visgoth, for reducing latency for browser based visualizations.

These contributions enable doctors and medical expert analysts to interactively analyze EEG data at scale.

Bibliography

[1] Rajeev Agarwal, Jean Gotman, Danny Flanagan, and Bernard Rosenblatt. Automatic eeg analysis during long-term monitoring in the icu. *Electroencephalography and clinical Neurophysiology*, 107(1):44–58, 1998.

[2] Alvin Wang. Materialize, 2014-2016. https://github.com/Dogfalo/materialize.

[3] Armin Ronacher. Flask (A Python mircoframework), 2010-2016. http://flask.pocoo.org/.

[4] Armin Ronacher. Jinja2 (The Python Template Engine), 2010-2016. http://jinja.pocoo.org/.

[5] Bastian Bechtold. WebGL-Specotrogram, 2014-2016. https://github.com/bastibe/WebGL-Spectrogram.

[6] Leilani Battle, Remco Chang, and Michael Stonebraker. Dynamic prefetching of data tiles for interactive visualization. 2015.

[7] Ben Campbell. HappyHTTP, 2014-2016. http://scumways.com/happyhttp/happyhttp.html.

[8] Anant Bhardwaj, Amol Deshpande, Aaron J. Elmore, David Karger, Sam Madden, Aditya Parameswaran, Harihar Subramanyam, Eugene Wu, and Rebecca Zhang. Collaborative data analytics with datahub. *Proc. VLDB Endow.*, 8(12):1916–1919, August 2015.

[9] A Bricolo, S Turazzi, Fo Faccioli, Fo Odorizzi, Go Sciarretta, and P Erculiani. Clinical application of compressed spectral array in long-term eeg monitoring of comatose patients. *Electroencephalography and clinical neurophysiology*, 45(2):211–225, 1978.

[10] C. Carlson. Can We Screen EEGs More Efficiently? Spectrographic Review of EEG Data. *Epilepsy Curr*, 15(1):24–25, 2015.

[11] collectd. Collectd, 2008-2015. https://github.com/collectd/collectd.

[12] Michael Crosier and Lewis D Griffin. Using basic image features for texture classification. *International Journal of Computer Vision*, 88(3):447–460, 2010.

[13] Dropbox Inc. json11, 2013-2016. https://github.com/dropbox/json11.

[14] Aaron Elmore, Jennie Duggan, Michael Stonebraker, Magdalena Balazinska, Ugur Cetintemel, Vijay Gadepally, Jeffrey Heer, Bill Howe, Jeremy Kepner, Tim Kraska, Samuel Madden, David Maier, Timothy Mattson, Stavros Papadopoulos, Jeff Parkhurst, Nesime Tatbul, Manasi Vartak, and Stan Zdonik. A demonstration of the bigdawg polystore system. *Proc. Very Large Database Endowment (PVLDB)*, 8(12), 2015.

[15] L. Fleming Fallon. Gale Encyclopedia of Surgery: A Guide for Patients and Caregivers, 2004. http://www.encyclopedia.com/topic/electroencephalography.aspx.

[16] Matteo Frigo and Steven G. Johnson. The design and implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231, 2005. Special issue on "Program Generation, Optimization, and Platform Adaptation".

[17] Github Inc. Github, 2008-2016. https://github.com.

[18] Pierre Granjon. The cusum algorithm a small review. 2012.

[19] Crossbow Technology Inc. http://xbow.com/, 2005.

[20] Joe Walnes. reconnecting-websocket, 2010-2016. https://github.com/joewalnes/reconnecting-websocket.

[21] Joshua Blum. EEGToolkit, 2014-2016. https://github.com/joshblum/eeg-toolkit.

[22] Uwe Jugel, Zbigniew Jerzak, Gregor Hackenbroich, Gregor Hackenbroich, and Volker Markl. M4: A visualization-oriented time series data aggregation. *Proceedings of the VLDB Endowment*, 7(10):797–808, 2014.

[23] Bob Kemp and Jesus Olivan. European data format 'plus'(edf+), an edf alike standard format for the exchange of physiological data. *Clinical Neurophysiology*, 114(9):1755–1761, 2015/12/03.

[24] Sangmi Lee, Sung Hoon Ko, and Geoffrey Fox. Adapting content for mobile devices in heterogeneous collaboration environments. Citeseer.

[25] Dirk Merkel. Docker: Lightweight linux containers for consistent development and deployment. *Linux J.*, 2014(239), March 2014.

[26] Lidia MVR Moura, Mouhsin M Shafi, Marcus Ng, Sandipan Pati, Sydney S Cash, Andrew J Cole, Daniel Brian Hoch, Eric S Rosenthal, and M Brandon Westover. Spectrogram screening of adult eegs is sensitive and efficient. *Neurology*, 83(1):56–64, 2014.

[27] Ole Christian Eidheim. Simple-WebSocket-Server, 2014-2016. https://github.com/eidheim/Simple-WebSocket-Server.

[28] Conrad Sanderson. Armadillo: An Open Source C++ Linear Algebra Library for Fast Prototyping and Computationally Intensive Experiments. Technical report, NICTA, September 2010.

[29] Ali H Shoeb and John V Guttag. Application of machine learning to epileptic seizure detection. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 975–982, 2010.

[30] Konstantin Shvachko, Hairong Kuang, Sanjay Radia, and Robert Chansler. The hadoop distributed file system. In *Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on*, pages 1–10. IEEE, 2010.

[31] Stavros Papadopoulos, Intel Labs. TileDB, 2016. https://github.com/stavrospapadopoulos/TileDB.

[32] Teunis van Beelen. EDFlib, 2009-2016. https://github.com/Teuniz/EDFlib.

[33] The HDF Group. Hierarchical Data Format, version 5, 1997-2016. http://www.hdfgroup.org/HDF5/.

[34] Manik Varma and Andrew Zisserman. A statistical approach to texture classification from single images. *International Journal of Computer Vision*, 62(1-2):61–81, 2005.

[35] Tom White. *Hadoop: The definitive guide*. " O'Reilly Media, Inc.", 2012.

[36] Craig A Williamson, Sarah Wahlster, Mouhsin M Shafi, and M Brandon Westover. Sensitivity of compressed spectral arrays for detecting seizures in acutely ill adults. *Neurocritical care*, 20(1):32–39, 2014.

[37] Yesudeep Mangalapilly. Watchdog, 2010-2016. http://pythonhosted.org/watchdog/.

[38] Matei Zaharia, Mosharaf Chowdhury, Michael J Franklin, Scott Shenker, and Ion Stoica. Spark: cluster computing with working sets.

[39] Biye Jiang Zhicheng Liu and Jeffrey Heer. immens: Real-time visual querying of big data. *Eurographics Conference on Visualization (EuroVis)*, 32(3), 2013.