Rewriting an Antarctic Research Data Pipeline

Brian Dawn
Computer Science Department
Augsburg College
Minneapolis, MN 55454
dawn@augsburg.edu

Noel Petit
Computer Science Department
Augsburg College
Minneapolis, MN 55454
petit@augsburg.edu

Abstract

The PENGUIn Automated Geographical Observatories contain a wide variety of instruments which gather data pertaining to space physics. This data is made available via Augsburg College which handles the processing, graphing, and hosting of the data. The entire system which handles this was rewritten to be more modern and robust. Three new Linux servers were purchased to handle this. Data acquisition. graphing and processing software was completely rewritten.

Introduction

The Polar Experiment Network for Geophysical Upper atmosphere Investigations (PENGUIn) Automatic Geophysical Observatories (AGO) are unmanned Antarctic stations which contain a wide variety of instruments. Augsburg College is responsible for data acquisition and processing of these automated systems via the Iridium satellite system.

Background

In the early 1990s universities associated with the PENGUIn program deployed several AGOs for unmanned data collection. Each AGO contains many different instruments including wind and temperature sensors, magnetometers, and cameras. By 1998 six AGOs had been installed in Antarctica. [3]

Figure 1: AGO P1 in 2003. [1]

The Augsburg College Computer Science department is responsible for data acquisition and storage [2]. The old dialing system before this project was written approximately 20 years ago. The software running on the dialers was written in C++ and was poorly documented. The dialing machines were running legacy hardware and running Windows 2000.

The processing system was also poorly documented. Software was written in a variety of languages such as PHP, Bash, C, IDL, Matlab, Perl, and Java. Code was scattered across various directories in the processing server, which was also responsible for acting as a web server.

Development

The goals of the rewrite of this data pipeline were:

- to document the entire system,
- to have a file structure on the servers where code and data are easy to find, and
- to distribute the work into 3 separate servers to prevent security vulnerabilities

The three main processes for the system are data processing, data storage, and data serving. One server was needed to act as a web server, subversion server, and to perform data graphing, another as a data storage server, and lastly a server to handle data processing and to act as a gateway to our data server. These servers were named zspace, dspace, and wspace respectively. All the servers were setup with Ubuntu Server 10.04. The setup can be visualized in Figure 2.

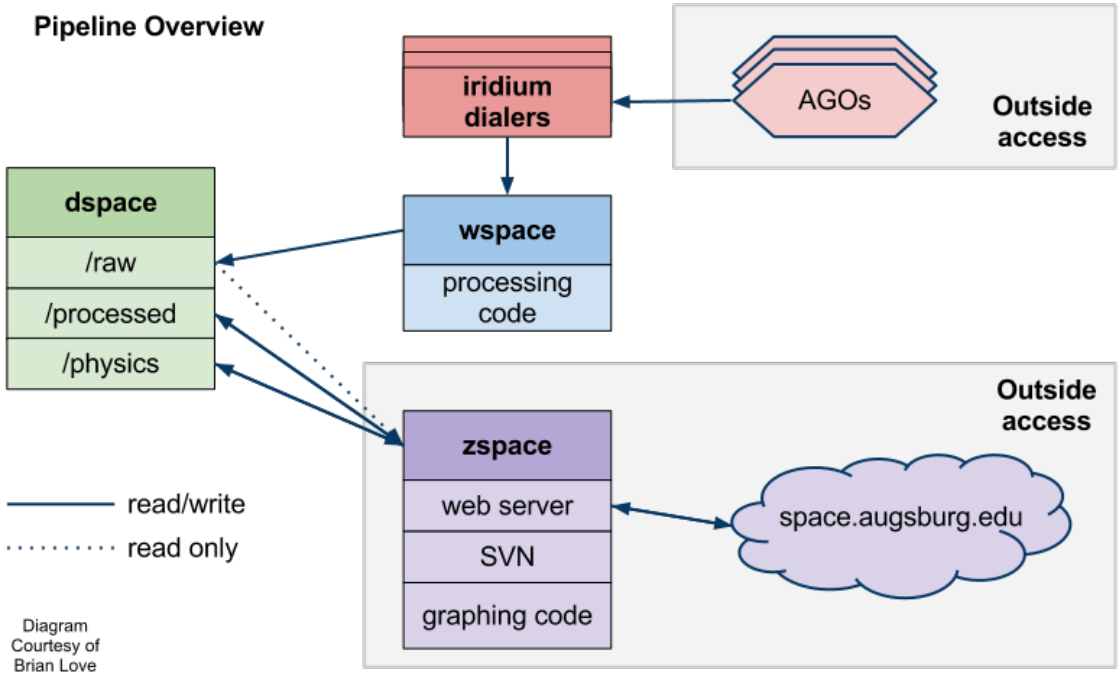

Figure 2: Server setup for data pipeline.

The primary purpose of wspace is to act as a server that pulls data from the iridium dialing machines and then writes the raw data to dspace. Only wspace was allowed to have write access to the raw data and wspace is not accessible from the outside. Limiting write access reduces the possibility of raw data being deleted or corrupted. Once the data have been processed, they are stored in a separate directory onto dspace with read/write access via zspace. In the event that processed data gets compromised, the raw data can simply be reprocessed.

The sharing of data directories (illustrated by arrows in Figure 2) were implemented using Network File System (NFS) mounts. NFS allows the mounting of remote folders as though they were local folders. This is an elegant solution, because moving files around can be accomplished with a standard copy or move command. NFS can also restrict access to only certain clients if desired. This is accomplished in the configuration of dspace.

Additionally, dspace also contains a directory for the space physicists at Augsburg. This provides a temporary place for the large amounts of data that the physicists use in their research.

Data Graphing and Processing

Each individual instrument on the AGOs is assigned a specific channel number. The channel number is used to distinguish the different types of data during collection, processing, graphing, and hosting.

Channel	Data type	Data Rate
0	Housekeeping data	1.0 minute
2	Imaging riometer	
3	X-axis search coil magnetometer	0.1 second
4	Y-axis search coil magnetometer	0.1 second
5	Z-axis search coil magnetometer	0.1 second
6	H-axis fluxgate magnetometer	1.0 second
7	D-axis fluxgate magnetometer	1.0 second
8	Z-axis fluxgate magnetometer	1.0 second
9	0.5-1 kHz N/S VLF Receiver	2.0 second
10	1-2 kHz N/S VLF Receiver	1.0 second
11	2-4 kHz N/S VLF Receiver	1.0 second
12	4-8 kHz N/S VLF Receiver	1.0 second
13	8-16 kHz N/S VLF Receiver	1.0 second
14	16-32 kHz N/S VLF Receiver	1.0 second
15	30-40 kHz N/S VLF Receiver	1.0 second
16	VLF Broadband Spectrum	
17	All-Sky Camera	
18	LF-HF Broadband Receiver	0.1 second
19	1-2 kHz E/W VLF Receiver	1.0 minute
20	24.0 kHz N/S NAA Receiver	1.0 minute

Figure 3: Instruments on an AGO.

As the data are collected during the course of the day, they are written to a specific file depending on the channel. After a full day's worth of data have been collected from the Iridium dialers, the files are moved to the raw data directory on dspace via wspace and is then processed by zspace.

Part of the project was graphing summary data and then developing a website where a user could view a quick snapshot of a particular day. If something interesting appeared they could investigate the binary data more closely. The first step was to develop a method of graphing this binary data.

The graphing was accomplished using the Java library jFreeChart. This open source library provides many useful features for graph and chart making. Code to convert the raw data into a form that was usable by the jFreeChart library was written by the Brian Love, and is out of the scope of this paper.

Figure 4 shows a summary plot generated by the system.

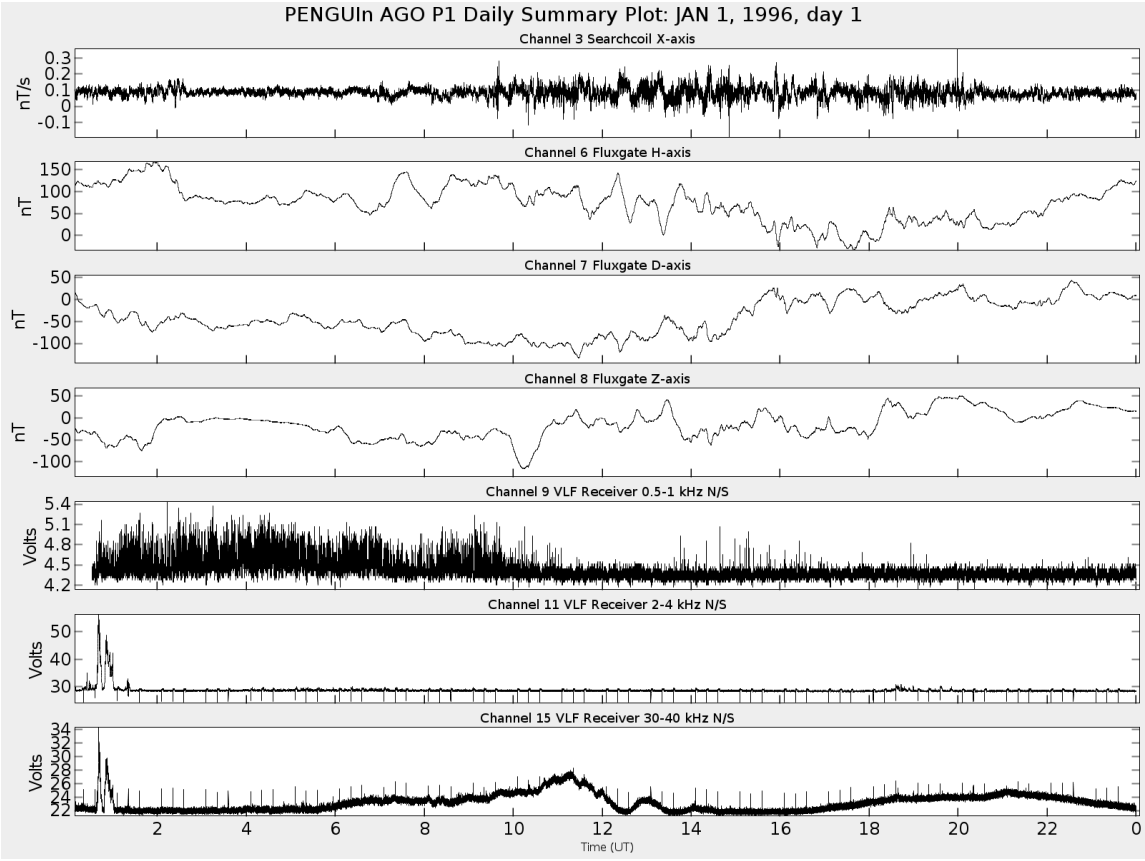

Figure 4: Sample summary plot generated by graphing software.

The next step in the process is tying it all together. Scripts which invoke all the different bits of processing software and data retrieval from the dialers needed to be automated. Every day at UTC 1:00 data needs to be pulled from the dialers and processed. To accomplish this cron was configured to run the data processing and moving scripts.

Future Work

As of right now there are many instruments which are not processed and graphed. It is left up to researchers to process these data themselves. In the future we hope to have most if not all of the AGO channels being graphed in real time.

An additional future project is to generate ASCII data from the data. Many researchers prefer to have the ASCII data.

Acknowledgements

This work was supported by National Science Foundation grant ANT0840158 to Augsburg College Physics Department. We thank Brian Love for his work on the project and proofreading this paper, and Erik Steinmetz for designing the styles and providing a framework for the jFreeChart library.

References

[1] *AGO Field Photos: Dec 2003 – Jan 2004.* (n.d.). Retrieved March 6, 2012, from Augsburg College Space Physics: http://yspace.augsburg.edu/ago/photos03.html

[2] *Antarctic Project History.* (n.d.). Retrieved March 15, 2011, from Augsburg College Physics Department: http://www.augsburg.edu/home/physics/abs.htm

[3] *Engebretson, M. J., et al. (1997)*. The United States automatic geophysical observatory (AGO) program in Antarctica. (M. Lockwood, M. N. Wild, and H. J. Opgenoorth, Ed.), Satellite – Ground Based Coordination Sourcebook, ESA-SP-1198, 65-99.