An implementation of a database management system for real-time large-lake observations

Joeseph P. Smith, Russ J. Miller, Ronald W. Muzzi, Stephen A. Constant, Kyle S. Beadle, Danna A. Palladino, Thomas H. Johengen, Steven A. Ruberg

<<H1>> ABSTRACT

Real-time environmental observations have been typically stored on relatively inaccessible flat text files. In this publication, we present an instance of a PostgreSQL database management system (DBMS) to ingest real-time observations from four buoys in North America's western Lake Erie. Data are transmitted via a cellular data modem, initially archived as text data, and then ingested into the database for further analysis and retrieval. The database utilizes two tables with parallel structure to archive data in a consistent manner. We assign unique keys to instrumentation configurations as they change within and between monitoring seasons. Daily sets of data are linked to their configurations by key, and thus are documented, allowing for efficient browsing of user desired data. Additionally, we quality check data and archive the findings in a corresponding matrix. With assistance from server-side processing, we produce a web interface for the database. We hope the design of this database allows for relatively simple deployment in domains other than Western Lake Erie.

KEY WORDS: Database, management, PostgreSQL, Remote Sensing

<<H1>> INTRODUCTION

For years, researchers in the Laurentian Great Lakes basin have deployed real-time observation platforms across the entirety of the basin, both on the shore and in the water via buoys. Utilizing satellite, cellular, and wireless local area network data transmission technology,

these platforms take measurements of environmentally relevant variables regarding the physical, biological, and chemical state of the ecosystem and transmit data back to their home laboratories for further processing and analysis. A network of stations, the Real-Time Coastal Observation Network (ReCON), has been operated by the National Oceanic and Atmospheric Administration's (NOAA) Great Lakes Environmental Research Laboratory (GLERL) for the past decade (Ruberg et al., 2007). The network, and the data it has produced, have provided numerous benefits, including ground-truthing for comparison with nearshore model outputs and early warning of conditions potentially hazardous to populations living near or accessing the lakes (Ruberg et al., 2008).

Methods of archiving ReCON station data have been limited to variations of flat text files. The accessibility of these high-temporal density data is hindered by this format. Users accessing these data on the internet may not have the skill or knowledge to parse such a file for analysis, and client-side technologies such as Javascript, and assisting packages, have limited methods to parse and display the data as rapidly as possible. Importantly, the specifications of users' computers vary widely, and it is unreasonable to expect that most computers have enough memory to take on high density data, or consistent and sufficient internet service.

Generally, there is a need for fast researcher and stakeholder access to real-time environmental data. Multiple technologies exist to facilitate data management, visualization, and analysis of data in varying spatial and temporal scales (Vitolo et al., 2015 and Smith et al., 2016). The extent to which data technologies are employed, however, is limited by the availability of computational and human resources, along with the scopes of initiatives that employ them.

In an effort to make real-time data collected in the western basin of Lake Erie publicly

viewable in an efficient manner, we developed a relatively simple implementation of a PostgreSQL (postgresql.org) database. Database management systems (DBMS), such as PostgreSQL, Microsoft SQL (MS SQL), and MySQL (www.mysql.com), amongst others, have been developed to handle a wide variety of data. These systems permit connectivity between them and websites, allowing for user interaction with data. PostgreSQL is an industry standard, open-source DBMS with robust specifications that has been employed by organizations such as the United States Department of Defense (Farias 2007; Karels 2003). Through server-side technologies, notably PHP (recursively defined as PHP: Hypertext Processor; see php.net) and the Javascript runtime environment Node.js (nodejs.org), webpages can be connected with data from SQL databases for dissemination. Additionally, desktop applications such as R (R Core Team 2014), with additional libraries, can access the data for further analysis.

Other solutions for connecting DBMSs with web interfaces or accessing them through desktop applications include 52 North's Sensor Observation Service (http://52north.org/communities/sensorweb/sos/), which is relatively agnostic as to what DBMS is used, and Water Information Systems by KISTERS (or WISKI, see https://www.kisters.net/NA/products/wiski/), which utilizes MS SQL and Oracle DBMSs. We find, however, that the capabilities of the PostgreSQL database, PHP, and R are sufficient for secure storage, and later analysis, either numerical or visual, of real-time Lake Erie data.

In this publication, we outline the data to be ingested into the PostgreSQL database, then describe the database structure. Finally, we discuss a web interface to the real-time data. This database implementation may be used as a model to develop databases of real-time data from other locations.

<<H1>> THE REAL-TIME DATABASE

<<H2>> TARGET DATA FOR INGESTION

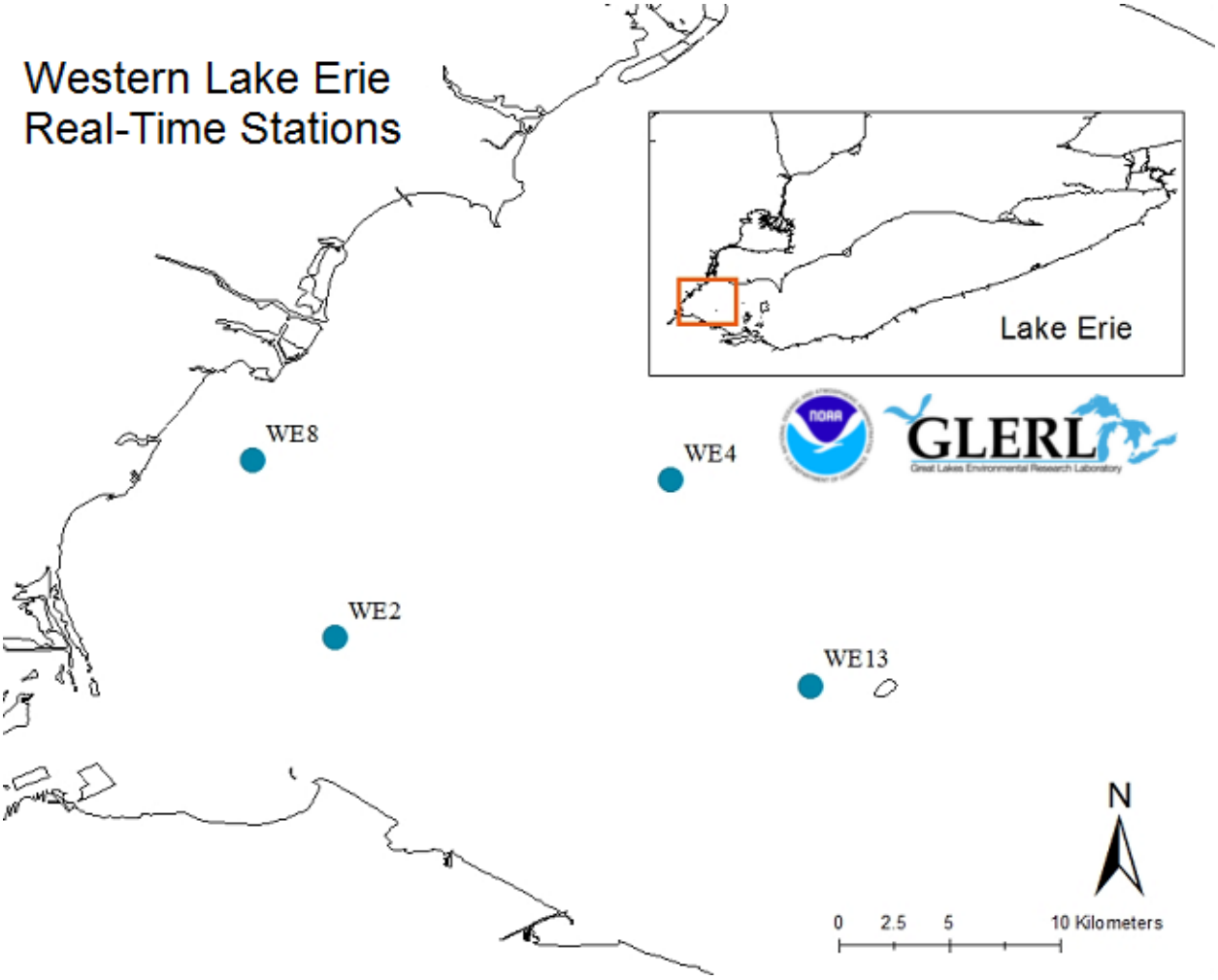

Figure 1: Map of western Lake Erie real-time buoy locations. Lake Erie is located on the southern end of the Canadian province of Ontario, and is surrounded by Michigan, Ohio, Pennsylvania, and New York in the United States.

Since 2014, NOAA-GLERL, in partnership with the University of Michigan's Cooperative Institute for Great Lakes Research (CIGLR), has deployed a series of buoys in the western basin of North America's Lake Erie (Figure 1) for the monitoring of seasonal Harmful Algal Blooms (HABs). Instruments aboard each buoy (summarized in Table 1) estimate

meteorological conditions (wind speed and direction, barometric pressure, air and water temperature) and water quality variables (turbidity, conductivity, concentrations of phosphorus, chlorophyll, blue-green algae, and nitrogen). Data generated from the instruments are formatted into a CR1000 data acquisition system table, transmitted via a cellular data modem, and received by Campbell Scientific's LOGGERNET data acquisition system. The flat-text data tables are then archived on a server at NOAA-GLERL.

Instrument	Years Implemented	Locations deployed	Variables
SeaBird Satlantic Suna V2	2016	WE2, WE4	Nitrogen
AIRMAR WS-200WX	2016, 2015	WE2, WE4, WE8, WE13	Meteorological
YSI EXO2	2016, 2014 (WE8)	WE2, WE4, WE8, WE13	Water Quality
SeaBird/Wetlabs CycleP	2016, 2015, 2014	WE2, WE4, WE8, WE13	Phosphate
Turner C6	2015, 2014	WE2, WE4, WE8	Water Quality
YSI 6600V2-4	2014	WE8	Water Quality

Table 1: Instruments deployed with buoys in western Lake Erie

<<H2>> DATABASE STRUCTURE

Table	Column	Postgres Data Type	Description
Configurations	configid	Character(40)	Unique identification code for the configuration
	startdate	Timestamp without time zone	Date the configuration was instantiated
	enddate	Timestamp without time zone	Date the configuration was terminated
	notes	Character(300)	Additional notes
	gmtoffset	Smallint	Time offset for time zone
	configtable	Text array	Matrix of

			configurations
Data and Quality Assessments	datalabel	Character(40)	Unique identification code for the day's set of data
	configid	Character(40)	The corresponding configuration for the day's data
	measuredate	Date	The day the data were collected
	dtable	Text array	Matrix of data from measuredate
	flagtable	Text array	Matrix of quality check flags for data from measuredate

Table 2: Structure of database for ingestion of real-time observations from western Lake Erie.

The PostgreSQL database consists of two (2) tables: one for buoy configurations, the other for the daily data and quality assessments (see Table 2). Key to the database's functionality is the parallel relationship between the configuration matrix and data matrix, allowing for algorithmic processing of any day's data.

Each configuration is given its own entry in the database, with a unique identifier corresponding to the specific buoy and the date the configuration took effect (e.g. cWEXXYYYYMMDD). The configuration matrix itself consists of seven columns: a variable name universal across data sets, the variable name as provided in the data, units of data (as applicable), statistic of data (i.e. minimum, maximum, average), datatype, and two columns describing the expected range of the data for the prescribed required climatological range checks from the U.S. Integrated Ocean Observing System (IOOS) Quality Assurance/Quality Control of Real Time Oceanographic Data (QARTOD) project (Bushnell 2016).

Configuration matrices have the same number of rows as columns in the daily data matrices, each of which has its own entry in the database. These entries also have a unique

identifier, corresponding to the specific buoy, the date of observations, and a single character indicating configuration (e.g. dWEXXYYYYMMDDa, where ‘a’ indicates this is the first configuration for that day). Rarely does a day’s data have more than one configuration, but the system has been designed to handle multiple configurations in a single day. To map the data matrices’ columns to their variables in the configuration, each matrix’s corresponding configuration ID is included in each entry. This allows for consistent data retrieval of variables of interest throughout the database. Lastly, a matrix of QARTOD standard data quality flags for their climatological range checks is included in each entry.

Using R, configuration matrices are generated, and archived text data are processed into the database, as well as quality checked. Automated processing is achieved through shell scripts on a UNIX machine run via the native crontab task scheduler.

<<H1>> WEB INTERFACE TO DATA

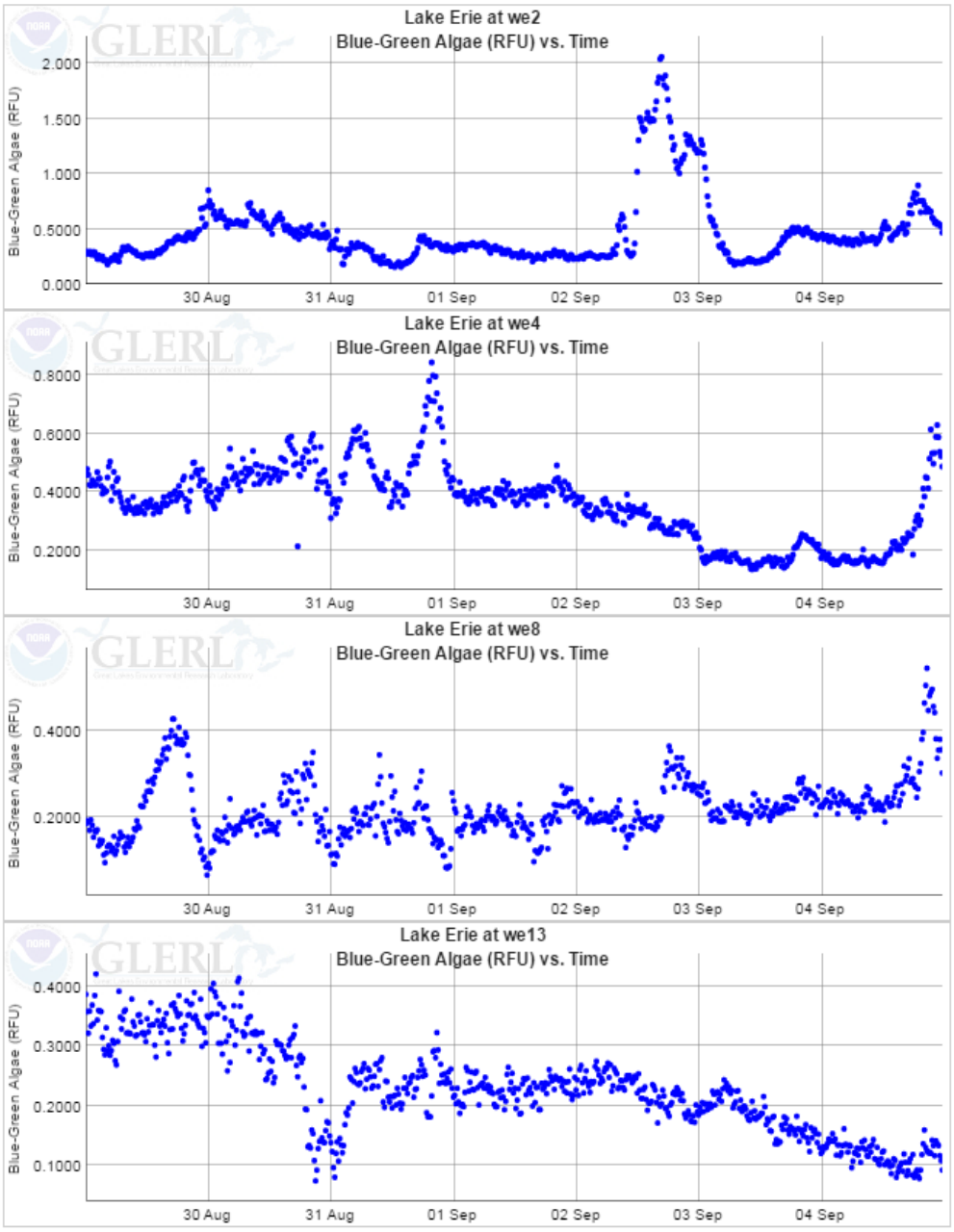

Figure 2: Screenshot of blue-green algae plots across the 4 real-time stations from the 2016 HAB season.

A web interface is available via the NOAA-GLERL website utilizing PHP and the Javascript package Dygraphs (see

https://www.glerl.noaa.gov/res/HABs_and_Hypoxia/rtMonSQL.php). Users may select a specific date, and how many days surrounding the date to view data from (maximum of 30 days worth of data due to server memory constraints). After selecting a time range, users may view data from an individual buoy, or a subset of variables across all buoys.

<<H1>> DISCUSSION AND FUTURE WORK

We have developed an instance of a PostgreSQL database for real-time environmental observations in western Lake Erie. After the data are transmitted from a buoy, and stored as flat text files on a server, they are processed into the database and the configuration of the data is noted. Also, QARTOD recommended quality checks are performed and results are stored in the database. Lastly, with assistance from server-side processing, an efficient and accessible website is available to visually analyze the real-time data.

The database, as presented, could be expanded to include data from other real-time buoys and stations from outside the western basin of Lake Erie. Expansion can either be a) done within the current database, or b) two new PostgreSQL tables can be created with the same functionality. We plan on expanding the database to include, initially, ReCON data from Muskegon (on Lake Michigan) and Saginaw Bay (in Lake Huron). With sufficient performance, we will expand out to include all real-time stations from ReCON buoys and stations located on offshore and near-shore structures.

<<H1>> ACKNOWLEDGEMENTS

Funding for this project comes from the Great Lakes Restoration Initiative, administered by the Environmental Protection Agency, and NOAA's High Performance Computing and Communications Program. The authors would like to thank the NOAA-GLERL IT group for

system setup and physical maintenance, CIGLR's Thomas Johengen, Mary Ogdahl, Dack Stuart, Danna Palladino, and co-authors for supporting and contributing to this project.

The use of product names, commercial and otherwise, in this paper does not imply endorsement by NOAA, NOAA-GLERL, CIGLR, or any other contributing agency or organization.

This is NOAA-GLERL contribution XXXX and CIGLR contribution ZZZZ. Funding was provided to the Cooperative Institute for Great Lakes Research through the NOAA Cooperative Agreement with the University of Michigan (NA12OAR4320071).

<<H1>> REFERENCES

Bushnell, M. 2016. Quality Assurance / Quality Control of Real-Time Oceanographic Data. In: OCEANS 2016 MTS/IEEE Monterey. pp. 1-4. , Monterey, CA. IEEE

Farias, M.A. 2007. Extending DoD modeling and simulation with Web 2.0, Ajax and X3D. Doctoral dissertation. Naval Postgraduate School. 228 pp.

Karels, M. J. 2003. Commercializing open source software. ACM Queue, 1(5), 47-55.

R Core Team. 2014. R: A language and environment for statistical computing (Version 3.0. 2). Vienna, Austria.

Ruberg, S.A., E. Guasp, N. Hawley, R.W. Muzzi, S.B. Brandt, H.A. Vanderploeg, J.C. Lane, T.C. Miller, And S.A. Constant. 2008. Societal benefits of the real-time coastal observation network (ReCON): Implications for municipal drinking water quality. MAR TECHNOL SOC J. 42(3):103-109.

Ruberg, S.A., S.B. Brandt, R.W. Muzzi, N. Hawley, T. Bridgeman, G.A. Leshkevich, J.C. Lane, And T.C. Miller. 2007. A wireless real-time coastal observation network. EOS Transactions 88(28):285-286.

Smith, J.P., A.H. Clites, C.A. Stow, T.S. Hunter, A.D. Gronewold, T. Slawecki, and G. Muhr. 2016. An expandable world wide web based platform for visually analyzing multiple data series related to the Laurentian Great Lakes. Environmental Modelling and Software 78:97 – 105

Vitolo, C., Elkhatib, Y., Reusser, D., Macleod, C. J., & Buytaert, W. 2015. Web technologies for environmental Big Data. Environmental Modelling & Software 63:185-198.