УДК 608.3

# THE WIKIENCE: COMMUNITY DATA SCIENCE. CONCEPT AND IMPLEMENTATION.

*Rodriges Zalipynis R.A., Zapletin E.A., Averin G.V.*
*Donetsk National Technical University*

*The purpose of this paper is threefold. First, it describes the new approach for doing data science collaboratively. Second, we coin the word "wikience" to mark the new concept. Lastly, for the first time we publicly present the results of our almost yearly research: the cloud system and its client software partially implementing the Wikience concept. The software allows real-time access to all available climate reanalysis data, 3D visualization of isopleths based on it and 3D visualization of cyclone tracks. To our best knowledge, our system is the first in the word capable to perform this kind of data delivery and visualization.*

## Introduction

Along with the enormous increase in data volumes, there is only a slight rise expected for analytical staff [1]. Only a handful of the data will be ever seen by a human eye. The main outcome usually derived from this fact is the need for new sophisticated data mining algorithms and powerful computers for automatic discovery of hidden value in data.

While this is undoubtedly important, there remain uninvolved a huge number of potential human analysts. They can perform much more effectively than a single research team even if equipped with simple tools.

## 1 The Wikience Concept

The word wikience is derived from Wiki + Science which stresses doing science collaboratively by sharing data and ideas. The word is good for international use, as it does not require a translation, just pronunciation. The main slogan behind Wikience is "If we do not have analysts, let's create them". All existing vast amounts of data must be stored and managed within a cloud. As many people as possible regarding their proficiency (researchers, students, scholars) must have access to that data and have simple, intuitive, yet powerful tools for its analysis.

This concept assumes collaboration with geo-referenced discussions, community validations, micro papers (exposing research results to community as soon as possible), easy sharing of research results (charts, datasets), data contribution to the cloud by individuals, new social networks and standards for developing software for Wikience.

## 2 Climate Wikience

To test the concept we chose climate science as one of the most appropriate domains. First, it has vast amounts of retrospective data collected over decades and mature theoretical background. Second, it is very important to society. Lastly, it can be used as a sandbox for development of computer science algorithms that could be generalized onto other disciplines. Previous efforts of collecting large amounts of data and its further automatic analysis have not been very successful. One of the main reasons was the attempt to cover the data from broad number of knowledge domains.

Cyclones are crucial component of atmospheric circulation. Climate changes along with cyclones. Cyclone paths over the last 30 years derived from reanalysis archives may serve as excellent indicators on climate variability and change. However, over 20 years of efforts of automatic tracking algorithms, none gives 100% accuracy.

In context of Climate Wikience we are creating a cyclone reanalysis archive which will contain consistent and verified cyclone tracks. The idea is to use a reanalysis data and cyclone tracking scheme to create initial database of cyclone tracks. After this, it will be available to research community via Internet in 3D. If one visually detects wrong path he/she would be able to correct it using intuitive graphical interface. Different researchers analyze different time periods and regions, thus in a short period of time it is expected to evolve into manually validated cyclone reanalysis database.

With this kind of database we can conduct much more accurate statistics of climate variability and change, improve current tracking algorithms and validate numerical climatic models.

## 3 ChronosServer

One of the mostly used climate data is climate reanalysis archives (AMIP/DOE 2 [2], Era-interim [3], etc). While it is important to have automated processing techniques for large amounts of reanalysis data, a value that OLAP and visualization brings to an expert is hard to overestimate. With visualization a data can be much easily perceived, patterns, outliers, and trends are readily noticed. However, exploration and visualization by analysts requires real-time access to all available data.

Existing solutions are either designed for batch processing (HDFS, HadoopDB [4]), or too excessive and complex (Lustre [5]), or too generic (RDBMS, Rasdaman [6], vertical databases like MonetDB, document oriented like MongoDB, or CloudFS).

We designed and implemented the ChronosServer. We choose “Chronos” (Greek word for time) because it maintains data partitioned over cloud nodes by time. Its architecture is specially designed for real-time data delivery to thousands of simultaneous clients. It assumes intensive reads with absence of writes. All semi-structured or modifiable information has considerably less volume and stored in document-oriented database (not shown).

The ChronosServer is based on the following observations. When dealing with very long period of time the data is usually already partitioned on time slices. Data for each time slice is stored in a separate file. That files could be viewed as chunks with variable size. In addition, each data is best stored in the format specially designed for that type of data. Data is usually shipped compressed and it is preferably to keep its size close to the original one.

Our cloud stores reanalysis data unmodified and uncompressed in its source files. The ChronosServer reads data directly from NetCDF/HDF files. This allows a number of benefits. First, we preserve original metadata intact. Second, with existing extent-based file systems and optimized libraries we achieve very fast read times measured in several milliseconds. Third, new data addition to the cloud becomes as simple as plug-and-play: download the file and copy it onto any cluster node. Forth, we operate on compressed data. Fifth, it enables simple replication and removal by using conventional copy and delete operations of a file system.

The overall Chronos architecture has three main components: ChronosWorker which is run on every data node, ChronosAPI which is used by client software to remotely access the data, and ChronosGate which connects clients using ChronosAPI with ChronosWorkers since ChronosWorkers do not have direct access to the Internet (later simply worker, client and gate).

The data files are simply located on local disks of workers. The gate does not have any information on the number of nodes in the cluster or data locations. Once a worker starts up, it recursively scans a given data directory on local file system to find out which datasets, their geographic coverage and time intervals are available. After that, the worker connects to the gate and sends the collected information via the network. The gate registers the worker and its information in worker pool.

When a worker crashes, it stops sending heartbeats to the gate. The information on its datasets is removed from the worker pool. The fully fresh collect-send-remove phase repeats each time a worker joins/leaves the cluster. This allows for better fault tolerance, flexibility and easy data addition/removal from the cloud.

The gate keeps the full contents of each worker storage namespace in operating memory for speedup. The same file may be replicated across several workers for reliability.

The real-time data delivery is organized as follows (fig. 1).

When GUI Client starts up, ChronosAPI connects to ChronosGate and establishes a session. Each

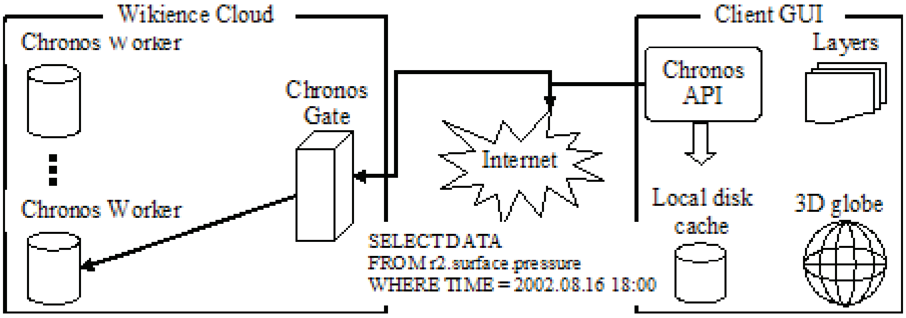


Figure 1. Climate Wikience query processing

GUI client has a unique session which distinguishes it from other clients. When user changes time, zooms or rotates the globe each layer receives notification to display new data for the new view setup. If the data is already in memory for the given new time/resolution/area it is simply rendered. Otherwise, a layer checks the local disk cache first to accelerate retrieval of frequently used data.

In case of cache miss, a layer generates an SQL-like query to retrieve data remotely from ChronosServer. For example, “SELECT DATA FROM r2.surface.pressure WHERE TIME = 2002-08-18 18:00” which will initiate retrieval of the single regular grid of surface pressure of AMIP/DOE Reanalysis 2.

The gate parses the query and chooses a worker which has requested data files for given time and issues internal query to the chosen worker. The worker is permanently connected to the gate for speedup. Once a worker receives the query it reads data directly from reanalysis file and sends the data back to the gate as a response. The gate retransmits the data to the client.

The data is read from files packed and transmitted also packed. While unpacking a single grid takes only a split of a second, with the presence of thousands of simultaneous clients this will put significant load onto the cloud. Thus, a client receives packed data along with instructions on how to unpack it. This additional work is not noticeable for a single client but significantly reduces load onto workers. We called this “split data delivery” in contrast to existing cloud services that fully rely on performing complete cycle of computations on the internal servers delivering only ready to show results.

## 4 Graphical user interface

The GUI client to ChronosServer is a desktop application the main purpose of which to visualize data in 3D. The source reanalysis data is shown by colored circles located in the nodes of latitude-longitude grid. The size and color of a circle is proportional to the value.

We developed algorithm for building closed isopleths and their visualization in 3D. The main features of our 3D contours are the following (fig. 2). They are all closed objects, not just a picture on a screen. Thus, they can be used for GIS operations (inside, covers, intersect, etc.). Our algorithm for their building is much faster compared to existing tools (IDV). They are visualized as terrain following. A user can easily switch time by clear and intuitive time sliders making isopleths 4D. What is more important, terabytes of data on which isopleths are built is stored in the cloud and accessed real-time on demand.

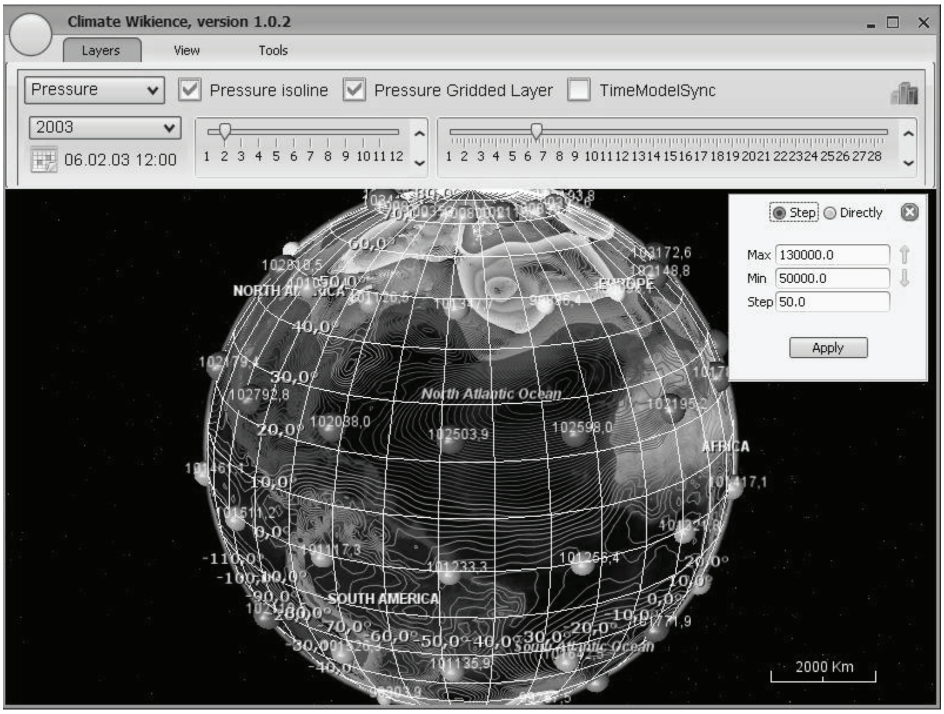
To build the isopleths we use CONREC, R-tree, and our merge algorithm that is not described here due to space constraints.

We also performed cyclone identification and tracking for reanalysis data with the University of Melbourne Automatic Cyclone Tracking Scheme [7]. The software was kindly provided by Kevin Keay and Ian Simmonds. We store cyclone centers, their tracks and characteristics in MongoDB. We visualize cyclone tracks in 3D (fig. 3).

We also developed parsers of data files from IBTrACS (the most complete archive to date of tropical cyclone tracks) [8] and we are also able to visualize tropical cyclone tracks in 3D (not shown due to space constraints).

We also able to visualize in 3D MODIS L3 data (global 1x1 regular grid).

In addition, we developed installer for the GUI client, automatic updater via Internet, visual latitude-



Climate Wikience, version 1.0.2

Pressure

☑ Pressure isoline ☑ Pressure Gridded Layer ☐ TimeModelSync

2003

06.02.03 12:00

● Step ○ Directly

Max 130000.0

Min 50000.0

Step 50.0

Figure 2. Source reanalysis data (circles; mean sea level pressure)
and 3D isobars built on it

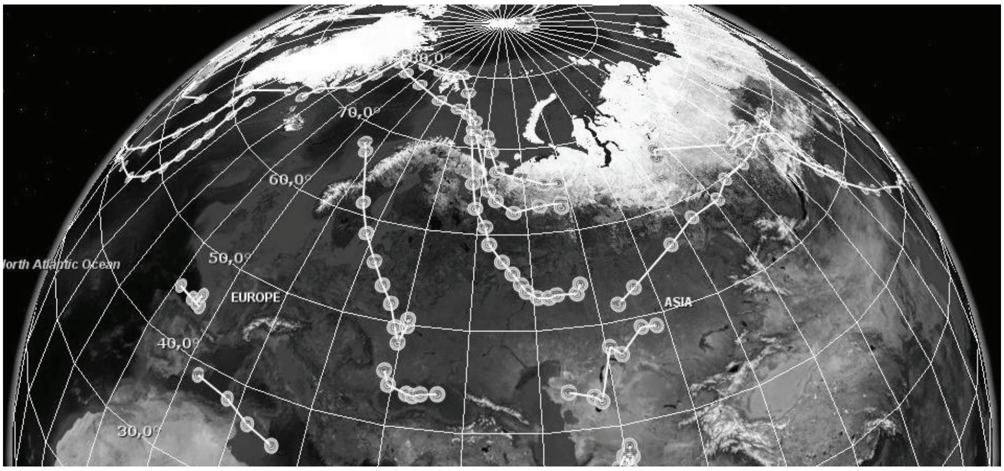


Figure 3. 3D visualization of cyclone tracks

longitude grid with configurable step, panel with layers list which is loaded remotely from the cloud, tools for calculating descriptive statistics (user visually selects a region for which a window with a Tukey boxplot is shown), and the ability to save views (current camera, time, layers, etc. state).

### Conclusions

One of the major accomplishments in climate research was the release of climate reanalysis archives. We performed the next step making them active and on-line. We plan to extend the system with more types of data and visual tools. We hope this system will be useful for the broad audience including decision support managers and policy makers. We plan to make it a platform for analyzing and visualizing vast amounts of geo-referenced data.

The Climate Wikience is freely available at wikience.donntu.edu.ua.

**Acknowledgements**

This work was supported by Award No. UKM1-2973-DO-09 of the U.S. Civilian Research & Development Foundation (CRDF). Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of CRDF.

**References**

[1] The 2011 IDC Digital Universe Study. Electronic recourse: http://www.emc.com/collateral/about/news/infographic.pdf

[2] NCEP Reanalysis 2 data provided by the NOAA/OAR/ESRL PSD, Boulder, Colorado, USA. El. resource. Access method: http://www.esrl.noaa.gov/psd/

[3] ERA-Interim Reanalysis data provided by ECMWF. El. resource. Access method: http://data-portal.ecmwf.int/data/d/license/interim/

[4] A Abouzeid et. al, An Architectural Hybrid of MapReduce and DBMS Technologies for Analytical Workloads, VLDB ‘09, August 24–28, 2009.

[5] Lustre (file system). El. resource. Access method: http://en.wikipedia.org/wiki/Lustre_(file_system)

[6] Raster data manager. El. resource. Access method: http://www.rasdaman.com/

[7] University of Melbourne Automatic Cyclone Tracking Home Page. El. resource. Access method: http://www.earthsci.unimelb.edu.au/tracks/cychome.htm

[8] International Best Track Archive for Climate Stewardship. El. resource. Access method: http://www.ncdc.noaa.gov/oa/ibtracs/index.php