INTRODUCTION

Background

- Scientific climate data plays an increasingly important role in the research on unusual climate events caused by global warming and climate change
- Accordingly, increasing number of studies on climate data visualization
- However, unlike general data without location information, the data should be plotted with map elements such as projection, contour line, color bar, spatial data (i.e. land, coastline)
- Also, there are not the appropriate libraries that can be used in any environments (e.g. GUI program, web service, server system) and any program languages (e.g., C#, JAVA) to plot the scientific climate data on the map
- In general, script-based open source software (i.e. NCL, GrADS) has been used in the field of climate science
- Therefore, it is necessary to develop the climate data visualization service that can be used in any environments considering a low-bandwidth network environment

Objectives

- To develop OGC WPS-based climate data visualization service that can be used in any environments and any program languages
- To design a lighter climate data exchange format in order to provide to the countries that have a low-bandwidth network environment and a plot option format

SERVICE DESIGN

- OGC WPS-based Climate Data Visualization Service

  ![Image](image1)

  - Development Environment: CentOS 6.5 64bit, Python, PyWPS and Matplotlib
  - OGC WPS operation name: OgammaWPS_CV_VasakulaNorthExtrn

- gClimateJSON

  ![Image](image2)

  - JSON-based data structure
  - Grid data notation by Base64 encoding algorithm (BinaryBase64, BinaryGzipBase64)
  - Sequence data (e.g. latitude, longitude) notation (individual, sequence)
  - CRS (Coordinate Reference System) attribute

  ![Image](image3)

  (1) JSON-based data structure
  (2) Grid data notation by Base64 encoding algorithm (BinaryBase64, BinaryGzipBase64)
  (3) Sequence data (e.g. latitude, longitude) notation (individual, sequence)
  (4) CRS (Coordinate Reference System) attribute

- Plot option based on Matplotlib functions

  ![Image](image4)

  - gClimateJSON
  - Plot option based on Matplotlib functions

RELATED WORK

- Map elements for visualizing climate data

  ![Image](image5)

  - Title:
  - Axis label
  - Contour line
  - Country boundary
  - LatLon grid
  - Color bar
  - Color bar label

- Case of Web-based Climate Information Service (KNMI – Climate Explorer)

  - Supports various map-based visualization options
  - Various map projections
  - Shadering and contours, shading, contours, grid boxes
  - Contour color
  - Blue-grey-red, red-grey-blues, grey-red, grey-blue, red-grey, etc.
  - No color bar, no title, no grid, no label options

  ![Image](image6)

  - Background projection
  - Robinson
  - Shading and contours
  - Grid boxes

RESULTS

- Comparison of data compression rates by grid data notation approaches

  ![Image](image7)

  - Data size is decided by integer parts and decimal place value
  - Data Size: Approx. 95 KB (4 decimals)
  - Regardless of integer parts and decimal place value, data size is only decided by data type (float, double, etc.)
  - Data Size: Approx. 55 KB
  - Data size is smallest than above approaches because of data compressed by GZIP algorithm using binary data of data type (float, double, etc.)
  - Data Size: Approx. 59 KB

- Result of OGC WPS-based visualization service

  ![Image](image8)

  - Data size: Approx. 47 KB
  - Original data size: 51 KB (two compression)

- Example of Windows Application using OGC WPS-based visualization service

  ![Image](image9)

CONCLUSION & REMARKS

- OGC WPS-based climate data visualization service was implemented using PyWPS and Matplotlib libraries for users (e.g. program developers) who are unfamiliar with climate data visualization
- The gClimateJSON exchange format with three grid data notation approaches (None, BinaryBase64, BinaryGzipBase64) was designed and suggested using BASE64 encoding and GZIP compression algorithm
- The suggested BinaryGzipBase64 notation approach was reduced by about 40 percent than None (written in pure text) approach
- The service can be provided on a low-bandwidth environment faster
- Also, users are able to take advantage of the data visualization, using scientific climate data, via the internet more easily and conveniently, making the service accessible to a larger range of users

* Acknowledge: This research was supported by APEC Climate Center