Our involvement in developing a CyberGIS curriculum began when we co-taught the course Mapping and Analysis in the Cloud as CyberGIS Fellows in the fall of 2014. In the course, we were able to introduce students to HTML, JavaScript, and the use of the Google Maps API to create a range of web mapping applications. The textbook that we used for the course, Mapping in the Cloud (Peterson, 2014), provided a good overview of basic cartographic and GIS principles and also provided numerous example code for developing various web mapping applications. This course also took advantage of our separate skills, Professor Slocum’s in cartography and visualization, and Professor Li’s in spatial analysis and programming. It also gave us a chance to be involved in team-teaching, something that we had not done together before.

Our biggest challenge in teaching the course was the limited programming capability of the students as several of the students enrolled in the class had little or no programming background. Although we had stressed that some programming background was essential prior to offering the class, we could not afford to have too many students drop the class. As a result, we had to move slower through the material than we originally had anticipated and could not cover as much material as we had hoped. We were particularly disappointed that we could not conduct analysis in the cloud. We were only able to briefly introduce Google Earth Engine (GEE), which has strong cloud analysis capabilities, and we didn’t have time for students to complete an associated exercise. Another challenge that we faced was trying to fit our conception of what should be covered with Peterson’s textbook. Peterson’s book alternates chapters of cartography/GIS material with chapters of coding approaches. At an introductory level, we expect that this approach could be quite effective, but many of our students already had considerable background in cartography and GIS; as such, our students would have benefitted by more of a programming emphasis in the book. Another difficulty in using the textbook is that it does not emphasize mapping and analysis of big data.

Realizing the lack of programming skills for most geography and earth sciences students, in the fall of 2015 Professor Li offered the introductory programming course, Computer Programming for Mapping and Spatial Analysis – this course assumes no previous programming experience and teaches programming and computational thinking using spatial data and analysis as applications. The course had been offered previously using the Python programming language, and dealt with only desktop spatial datasets and analysis. In the fall of 2015, Professor Li made a significant change by integrating the emerging areas of cloud computing and CyberGIS. Instead of Python, the course used the JavaScript programming language and taught students to develop web mapping applications and perform cloud-based spatial data analysis using GEE. In one assignment, students were asked to explore land cover/use change at their favorite place on the Earth’s surface using images at two different times from the Landsat 5 image archive (between 1984 and 2012) available in GEE to show significant land cover/use change caused by natural events or human activities. With the few lines of JavaScript code they learned from the class, students were able to see the shrinking Aral Sea, the impact of hurricane Katrina on New Orleans, and the extent of the 2011 Missouri River flood. In another assignment, students were able to visualize, explore, and analyze a quasi-global trend using the CHIRPS precipitation time series data between 1981 and 2015 (Funk et al. 2014). Both
assignments would be nearly impossible to complete in a single assignment if the data, visualization, and analysis capabilities were not available in the Cloud.

Building on the experience of the above courses, Professor Li also is offering the course *GIS Application Programming* in the spring of 2016. The emphasis of this course is to teach students to develop spatial and temporal data analysis tools and applications on desktop computers and in the Cloud. Students are developing desktop solar radiation models for horizontal, inclined planes, and on DEM where the shading effect from surrounding terrain is considered using desktop GIS. Students then scale the models to the Cloud with global DEM using GEE. Another planned topic and related assignments are to develop the Mann-Kendal non-parametric trend analysis tool on GEE and use it to analyze the trend of annual snow cover frequency with 15 years of MODIS daily snow cover data, and eventually to serve the analysis on the Google App Engine cloud platform as shown by the Trendy Snow cloud application (https://litrendylights.appspot.com/).

In summary, our teaching experience with CyberGIS, and especially GEE, has opened up and demonstrated the potential and capabilities of cloud computing to students with examples from spatial and temporal data analysis. Ultimately, future CyberGIS might provide a standard GUI that would be transparent to the user, and thus hide the difference between local and cloud data and analysis. Current users and future power users, who need to develop custom computing tools, of geospatial cloud computing platforms, however, still need to access cloud computing programmingly. There are still gaps in teaching spatial programming with current transition from traditional programming concepts and constructs to the programming paradigm for CyberGIS and cloud computing. We need to identify and introduce new programming concepts, such as asynchronous programming, the map reduce programming mode, and programming patterns for spatial and temporal analysis, for the era of big geospatial data analytics.

**Acknowledgements**

This position paper is supported by the CyberGIS Fellows Program and a Google Earth Engine Research Award.

**References:**
