Google Summer of Code - 2014

High Performance Nearest Neighbor Queries with Hadoop-GIS
libCUDASP – A General Spatial Query Processing Library for GPU
Medical Vocabulary Generating Tool
TCIA Data Exploration and Information Visualization
Data Replication/Synchronization Tools
Auto generate query template from AIM templates (XML)
Web-based UI for temporal query
Web-based source-to-target mapping UI
Automate account creation for new users
Integrate Eureka! with a web-based statistical analysis and data mining platform
Automate QA Process
Data mining algorithms with NoSQL database

High Performance Nearest Neighbor Queries with Hadoop-GIS

Mentor: Fusheng Wang (fusheng dot wang at emory dot edu)
Overview: HadoopGIS (http://confluence.cci.emory.edu:8090/display/HadoopGIS) is a high performance spatial data warehousing system that utilizes MapReduce for processing queries at cloud scale. The system supports some major spatial queries – such as window query, aggregation query and spatial join query, and query execution efficiency is much better than a parallel database implementation. HadoopGIS can easily configured to run on cloud services such as Google Compute Engine, and AWS (https://github.com/EmoryUniversity/libhadoopgis). However, some queries that are used frequently in practice are not implemented at this moment. Specifically, the nearest neighbor query (KNN), and reverse nearest neighbor query (RKNN). Providing query support for the neighbor query would enable users to query large amounts of spatial data on cloud with a very small budget. The main aim of this project is to efficiently implement the above mentioned queries within the HadoopGIS framework.
Programming Languages/Frameworks: Java/C++, Hadoop, MapReduce, Database
Prerequisites: Solid Java/C++ programming skills, familiar with Hadoop internals and MapReduce, experience in spatial query processing
Level of Expertise: Intermediate.

libCUDASP – A General Spatial Query Processing Library for GPU

Mentor: Fusheng Wang (fusheng dot wang at emory dot edu)
Overview: Spatial query processing algorithms involves heavy duty geometry calculations and complex transformation. Utilizing the massive power of modern GPUs to accelerate such CPU intensive operations could increase query performance by several orders of magnitude (http://dl.acm.org/citation.cfm?id=2350268). To systemically support various spatial query operations, a general library is needed and missing from the open source community. In this projects, we aim to develop such a GPU library which can reused for similar application need. Major spatial operations will be implemented in different level of parallelization. The query processing operations will be exposed as simple API which can be extended and reused by other developers and users.
Programming Languages/Frameworks: C++, CUDA/OpenCL
Prerequisites: C++ programming skills, experience in CUDA/OpenCL , experience in computational geometry or spatial query processing
Level of Expertise: Intermediate.

Medical Vocabulary Generating Tool

Mentor: Fusheng Wang (fusheng dot wang at emory dot edu)
Overview: This project is to create a tool that can improve the usage for standardized medical vocabulary such as ICD-10 and HL7. As there are various medical standards and vocabularies in use, to design a system and corresponding protocol that can parse and reformat them to a formal representation, for example: XML, would be very valuable. Moreover, automate the creation for customized vocabulary based on existing domain knowledge is also an important research topic.
Programming Languages/Frameworks: JAVA
Prerequisites: JAVA programming skills, experience in XML, HTML, experience with medical terminology, standard, ontology or vocabulary.
Level of Expertise: Intermediate.

TCIA Data Exploration and Information Visualization

Mentor: Ashish Sharma (ashish dot sharma at emory dot edu)
Overview: The Cancer Imaging Archive provides access to a wealth of biomedical cancer imaging data. It contains over 26 million radiology images, pathology data, and clinical data. The existing web interface for searching the archive is extremely outdated. Recently a REST API for TCIA was implemented to allow programmatic query and download of the data. Using the new REST API this project would seek to create a new search interface to the data as an alternate way to explore the contents of TCIA, create dynamic dashboards that can be extended to support the exploration of TCIA data (similar to "http://nickqizhu.github.io/dc.js/"). In addition to searching the TCIA archive, this project could also include support to intuitively formulate queries that can federate data from other remote archives. Possible strategies could include Microsoft Pivotviewer which provides an interactive data exploration platform.
Programming Languages/Frameworks: Javascript, d3, crossfilter, HTML
Prerequisites: Extensive experience with jQuery. Experience/Coursework in HCI, visual interface development.
Level of Expertise: Intermediate.

Data Replication/Synchronization Tools

Mentor: Ashish Sharma (ashish dot sharma at emory dot edu)
Overview: The Cancer Imaging Archive provides access to a wealth of biomedical cancer imaging data. It contains over 26 million radiology images, pathology data, and clinical data. Typically users download images to their local machines before analyzing the downloaded data. Over time, as new studies are uploaded, it becomes difficult to track which imaging studies have been downloaded by users. In this project you will propose and develop a system that can track what has been downloaded by a user, in response to a given query. Think of it as a one-way Google Drive/Dropbox (data always moves from server to client) where each folder is mapped to a particular query, and the contents of that folder are are frequently updated on the server side. Your client side solution would need to track of what has been downloaded and gives users the option of updating their collections. Your proposed solution can include extensions to the Java based web services that are used to create the REST API. Your client side application can be a cross-platform thick client or a desktop-based web application.
Programming Languages/Frameworks: Java, Other tools and languages are highly dependent on your proposed strategy.
Prerequisites: Experience in distributed computing and appropriate languages.
Level of Expertise: Intermediate

Auto generate query template from AIM templates (XML)

Mentor: Pattanasak Mongkolwat (p-mongkolwat at northwestern dot edu)
Overview: The Annotation and Imaging Markup (AIM) model provides a method for capturing structured assessments of biomedical imaging data. AIM-E is software which as been developed to store and query against this structured data. A key component of AIM is the ability to create "templates" which are used to ask a series of questions related to a research hypothesis. The goal of this project would be improve the AIM-E software to to build an automated set of queries which mirrors the questions found in the AIM template.
Programming Languages: Java, XML, XPath, JSON. Experience with OSGI would be a plus
Level of Expertise: Intermediate

Web-based UI for temporal query

Mentor: Himanshu Rathod (himanshu dot rathod at emory dot edu)
Overview: You will investigate novel user interface designs for identifying interesting patient populations for clinical research and healthcare analytics. Eureka! Clinical Analytics is a web-based software system that aims to break down the layers of IT that typically sit between electronic health record data and users of that data such as researchers and healthcare operations personnel. It aims to enable those users to define variables, computed from the source data, that are useful for their analytics or research task, an activity that typically is performed by IT intermediaries. These variables may be computed as patterns in temporal sequences and frequencies of clinical attributes (visit information, vital signs, diagnoses, etc.). These data transformation concepts are challenging to present to research and operations personnel in a web user interface.
This UI is a component of Eureka! Clinical Analytics, a federally funded web application for healthcare analytics. You can learn more about Eureka! at http://aiw.sourceforge.net.
Programming Language: Javascript, HTML, CSS, JQuery, JSP
Prerequisites: Javascript, HTML, CSS. JQuery, JSP a big plus.
Required skills – UI design.
Level of Expertise: Intermediate

Web-based source-to-target mapping UI

Mentor: Michel Mansour (michel dot mansour at emory dot edu)
Overview: You will investigate and prototype novel designs for a user interface for defining mappings from a source data model to a target data model. Source to target mapping is a key component of Eureka! Clinical Analytics, which aims in part to connect to enterprise data warehouses at a medical institution and support straightforward preparation of those data for research and operational analytics. We currently provide the source-to-target mappings functionality only in the form of Java code and some externalized configuration files. To facilitate adoption by hospital and biomedical research IT departments, we need an elegant UI that will allow data modelers (who are not programmers) to easily define source-to-target mappings that will make their enterprise data sources available through Eureka. This mapping UI is a component of Eureka! Clinical Analytics, a federally funded web application for healthcare analytics. You can learn more about Eureka! at http://aiw.sourceforge.net.
Programming Language: Java, JSP, Javascript, HTML, CSS.
Prerequisites: Java, Javascript, HTML, CSS. JSP a plus.
Required skills – UI design.
Level of Expertise: Advanced

Automate account creation for new users

Mentor: Himanshu Rathod (himanshu dot rathod at emory dot edu)
Overview: Eureka! Clinical Analytics is an open source web-based analytics application that provides user interfaces for new users to register for an account. Account creation is currently a manual, tedious, time-consuming, and error-prone activity. In this project, you will implement automated account creation that will run whenever a new user signs up. This code will hook into Eureka!'s existing infrastructure. It will require interaction with Eureka!'s own databases as well as with 3rd party software. New user registration is a component of Eureka! Clinical Analytics, a federally funded web application for healthcare analytics. You can learn more about Eureka! at http://aiw.sourceforge.net.
Programming Language: Java
Prerequisites: Java, XML. JDBC/JPA experience a plus.
Level of Expertise: Intermediate

Integrate Eureka! with a web-based statistical analysis and data mining platform

Mentor: Michel Mansour (michel dot mansour at emory dot edu)
Overview: You will extend healthcare data processing software to support straightforward analysis of its output using the R programming language (http://www.r-project.org/). Eureka! Clinical Analytics, our web-based clinical data processing software, provides sophisticated functionality for preparing electronic health record data for use in research and analytics. R is one of the most popular languages for statistical analysis and data analytics. We aim to create a web-based data analysis platform using a combination of Eureka and R. While Eureka supports outputting prepared data in various formats that can be consumed by R, it has no intrinsic integration with R or any other statistical analysis or data mining tool. You will make transferring prepared data from Eureka into R as easy as possible via selection of a web-based R solution, backend integration of Eureka and the selected R solution, and minor user interface extensions to invoke R on a prepared dataset. This project is an extension of Eureka! Clinical Analytics, a federally funded web application for healthcare analytics. You can learn more about Eureka! at http://aiw.sourceforge.net.
Programming Language: Java mostly, with some JSP, Javascript, HTML and CSS.
Prerequisites: Java, Javascript, HTML, CSS. JSP a plus.
Level of Expertise: Intermediate

Automate QA Process

Mentor: Michel Mansour (michel dot mansour at emory dot edu)

Overview: You will examine methods for and implement automated quality assurance of data for our software, Eureka! Clinical Analytics. Eureka performs complex transformations on large volumes of clinical data. We have reference datasets that we use to test the system. During every release cycle, we spend a lot of time verifying that the transformations' output are correct for each dataset. Since the data and transformations are well-defined, so is the output. We want you to build an automated system that, given a source dataset and expected output, computes whether the expected output and actual output are the same. If not, it should provide lots of detail about where the differences lie and what transformations may be producing incorrect output. This project is an extension of Eureka! Clinical Analytics, a federally funded web application for healthcare analytics. You can learn more about Eureka! at http://aiw.sourceforge.net.
Programming Language: Java, and a scripting language like Python, Ruby, etc.
Prerequisites: Java, database experience, comfortable with complex algorithms
Level of Expertise: Advanced

Data mining algorithms with NoSQL database

Mentor: Himanshu Rathod (himanshu dot rathod at emory dot edu)
Overview: We are currently working to export data from our application to NoSQL databases, to take advantage of the flexible schema. The nature of the exported data makes it suitable for a NoSQL graph database, such as Neo4J. You will explore and implement various graph algorithms within a NoSQL environment, to analyze the exported data. The aim of this effort is to provide researchers with an easy-to-use set of tools that can be used to gain a deeper understanding of the relationship between the data. The data to be analyzed is inserted into the graph database by the Eureka! Clinical Analytics application, a federally funded web application for healthcare analytics. You can learn more about Eureka! at http://aiw.sourceforge.net.
Programming Languages: Java
Prerequisites: Java. Familiarity with NoSQL databases and graph algorithms a plus.
Level of expertise: Intermediate