


# CS486C – Senior Capstone Design in Computer Science

## Project Description

<b>Project Title:</b> Space Reclamation from Cloud Object Storage		
<b>Sponsor Information:</b> 	<b>Daniel Boros</b> , Software Development Spectrum Protect Group, IBM dboros@us.ibm.com (520) 799-2216  Jeff Placer, UI Development and Design Spectrum Protect Group, IBM jrplacer@us.ibm.com (520) 799-2547	Christopher J. Ruskay, Software Development Manager Spectrum Protect IBM ruskay@us.ibm.com (520) 799-4086

### Project Overview:

IBM has established itself as a leader in cognitive solutions, infrastructure as a service (IaaS), and storage offerings. IBM's foray into the cloud has been driven by big data analytics and a push to better handle large-scale storage scenarios without the need for manual intervention. Enterprises do not have the time or resources to micro-manage disk and tape; they need the provisions of the cloud and the software solutions that IBM provides to manage their enterprise data for them automatically.

IBM Spectrum Protect is one of the premier products within IBM's suite of data storage management solutions. Spectrum Protect is designed to simplify data protection, whether data is hosted in physical, virtual, software-defined, or **cloud** environments. With Spectrum Protect, administrators can simplify backup administration, improve efficiencies in the backup process, and enable scalability to an entire enterprise of inputs.



One of the central challenges when backing up data is to minimize the size of the archived backups; this is particularly important when backing up to the cloud, where sending massive amounts of data over the wire unnecessarily costs both time and bandwidth. In order to create efficient backups, Spectrum Protect works to de-duplicate data, as well as compressing to the greatest extent possible. Spectrum Protect stores data extents (de-duplicated pieces of the original file) as a logical grouping in an archive container, which lives as an object in the cloud. Over time, these extents naturally become less active and expire, which means that the space in the containers is now eligible to be reclaimed to save on cloud storage costs.

But there's a catch: if we implement a naïve scheme of reclamation where we reclaim after a certain amount of data inside the container has expired, the cost metrics may not make sense... and the cost metrics need to make sense because we're operating at the scale of not a single container, but millions of containers representing billions of objects. The key question, therefore, is: what is the optimal moment for reclamation to be initiated? What would be an "intelligent" trigger for reclamation that optimizes the cost metric equation?

Answering this question is not simple and will require experimentation with various approaches, metrics, and algorithms. The goal of this Capstone project is to provide our team with a simple and effective tool for helping up explore this space management puzzle, as well as some initial insights towards a solution. In particular, you will be implementing a web application that will serve as a storage manager/analyst that IBM can use to visualize space management and allow exploration of various solutions, including the following key features:

- Web application takes the container layout as input.
- Web app will communicate with an asynchronous processing agent to reclaim container space, rewrite the container to object storage, and relay the result back to the front-end.
- The web application should visualize the status of reclamation in a way that allows answering the following questions at any time during the process:
  - Where are we in the process of reclaiming space for any given container?
  - How much space is pending reclamation?
  - What is the cost analysis for each container?
    - Where is the break-even or saving point for reclaiming the space for a given container?
  - How much money am I saving due to reclamation?

In the end, we expect that you will have a fully functional web application and agent to handle the reclamation, processing against a real Amazon S3 cloud.

#### Knowledge, skills, and expertise required for this project:

The proficiencies that will need to be developed to tackle this project are not entirely known in advance and will be clarified as we progress through requirements and design. However, they are likely to include:

- Knowledge of modern web architectures: application frameworks, deploying applications to the cloud, micro-services, et cetera.
- JavaScript, D3.js, and general web application knowledge.
- Micro-service and system programming in Go.
- A willingness to collaborate with your team and us.

#### Equipment Requirements:

- There should be no equipment or software required other than a development platform and software or tools freely available online.
- For all cloud development, you may take advantage of AWS Educate to get free access to cloud object storage and other Amazon tooling.

### Software and other Deliverables:

- The specified web application front-end and functioning back-end, deployed and running in the cloud. While personal accounts can be used for development, final delivered product must be installed on a cloud account provided by the client.
- A personal demo for us of the finished product and all the features by the team. Either in-person or via teleconference.
- A front-end GUI developed with good design principles, along with a user manual covering installation, configuration and operation of the storage analysis tool.
- A strong as-built report detailing the design and implementation of the product in a complete, clear and professional manner. This document should provide a strong basis for future development of the product.
- Complete professionally-documented codebase, delivered both as a repository in GitHub, BitBucket, or some other version control repository; and as a physical archive on a USB drive.