CS486C – Senior Capstone Design in Computer Science

Project Description

Project Title: Predictive Storage Tiering during Data Ingest		
Sponsor Information: 	**Daniel Boros** Staff Software Engineer IBM Spectrum Protect dboros@us.ibm.com (520) 799-2216 **Jeff Placer** UI Development and Design IBM Spectrum Protect jrplacer@us.ibm.com (520) 799-2547	**Christopher J. Ruskay** Software Development Manager IBM Spectrum Protect ruskay@us.ibm.com (520) 799-4086

Introduction:

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, tape, and object storage; they need predictive 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 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.

Problem Overview:

Currently, backup administrators manually configure policies to allow for data to demote from hot to cool storage. This is an exploratory project which focuses first on implementing a data processing pipeline that does predictive analysis using machine learning algorithms on, potentially, petabytes of the underlying incoming metadata that Spectrum Protect manages. The core challenge is to create a powerful machine learning module capable of predicting (in real-time) which storage tier a file should be placed into during ingest of the file into the Spectrum Protect environment. Specifically, the envisioned software product would consist of a server-side application running in a Linux environment on the Spectrum protect server. Specific features include:

- Application that runs as a server process to monitor incoming SpectrumProtect metadata streams.

- Uses a metadata feature set developed by the team (with guidance by sponsor) as a basis for driving a machine learning algorithm that classifies incoming ingests as hot or cool storage.
- Gathers data as it runs to assess prediction accuracy by comparing predictions with later information on fate/usage of the ingested data. That is, sees if prediction of hot/cool storage was appropriate, given information on how data was later accessed.
- Ideally would automatically use run-time prediction accuracy results to continually improve learning module.

If successful, this proof of concept could provide the basis for an improved "smart" data management approach to be integrated into the next generation of SpectrumProtect products. Knowing where to store an object during ingest means that we'll promote and demote data to the correct tier the first time, which can save on storage costs by freeing up resources and remove the need for a backup administrator to manually configure policies for data demotion.

In the end, we expect that you will have a fully functional and reproducible end-to-end pipeline for processing and classifying Spectrum Protect data.

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 a programming language that has excellent machine learning and data processing library support, whether it be in the standard library or provided by the open source community.
- Experience with TensorFlow or other machine learning packages.
- Experience or a willingness to learn data processing technologies or processes.
- 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.

Software and other Deliverables:

- Strong as-built report detailing the design and implementation of the solution in a complete, clear, and professional manner.
- Complete professionally-documented and commented codebase, delivered as a repository in GitHub, BitBucket, or other applicable version control solution.
- Complete live code notebook detailing fundamental data analysis and visualizations.
- A user manual covering installation, configuration, and steps to reproduce your results. Your analysis and results should be easily reproducible by us given the same data set!