2015-04-01

A New Global Forecasting Model to Produce High-Resolution Stream Forecasts

Alan Dee Snow
Brigham Young University - Provo

A New Global Forecasting Model to Produce High-Resolution Stream Forecasts

Alan Dee Snow

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

E. James Nelson, Chair
Norman L. Jones
Daniel P. Ames

Department of Civil and Environmental Engineering

Brigham Young University

April 2015

ABSTRACT

A New Global Forecasting Model to Produce High-Resolution Stream Forecasts

Alan Dee Snow
Department of Civil & Environmental Engineering, BYU
Master of Science

Warning systems with the ability to predict floods days in advance can benefit tens of millions of people. Because of these potential impacts there have been efforts to improve prediction systems such as the United States' Advanced Hydrologic Prediction Service and European-developed Global Flood Awareness System. However, these projects are currently limited to relatively coarse resolutions. This thesis presents a method for downscaling and routing global runoff forecasts generated by the European Centre for Medium-Range Weather Forecasts using the Routing Application for Parallel computatIon of Discharge program that make possible orders of magnitude increases in the density of the resolution of stream forecasts. The processing method involves using the Amazon Web Services to distribute execution in a cloud-computing environment to make it possible to solve for large watersheds with high-density stream networks. Using the Amazon Web Services, the number of streams that can be used in the downscaling process in a twelve-hour period is approximated to be close to five million. In addition, an application for visualizing large high-density stream networks has been created using the Tethys Platform of water resources modeling developed as part of the CI-WATER NSF grant. The web application is tested with the HUC-2 Region 12 watershed network with over 67,000 reaches and is able to display analyzed results to the user for each reach.

Keywords: ECMWF, RAPID, Tethys Platform, CI-WATER, Condorpy, flood prediction, forecast, GloFAS, Esri

ACKNOWLEDGEMENTS

First of all, I would like to thank my Heavenly Father for the opportunity to study at Brigham Young University and to work on this research project. I would also like to thank Him for all of the inspiration and guidance I have received while performing this research. I would like to thank my wife Sandra for her support while I spend my days and my thoughts working on this research.

I would like to thank Dr. Nelson for his assistance, guidance, and support throughout this project. I would like to thank Dr. Jones for his support in making CI-WATER and this project possible. I would like to thank Dr. Ames for his support to revise and improve this thesis.

I would like to thank Nathan Swain who was very helpful and patient in teaching me how to use the Tethys Platform as well as Scott Christensen who helped me to find my way around computing in the Amazon Cloud.

I would like to thank Cédric David for guiding me through setting up and running RAPID. I would like to thank Florian Pappenberger and ECMWF for providing the runoff prediction datasets and information about them. I would like to thank the Esri team (Nawajish Noman, Deng Ding, Christine Dartiguenave, Dean Djokic, and Steve Kopp) for developing the RAPID toolbox and providing assistance to use it. Lastly, I would like to thank Ervin Zoster for providing the GloFAS comparison data.

This research is based upon work supported by the National Science Foundation under Grant No. 1135483.

TABLE OF CONTENTS

LIST OF TABLES ..... **vi**

LIST OF FIGURES ..... **vii**

1 Introduction **1**

2 Background and Related Work **4**

2.1 European Centre for Medium-Range Weather Forecasts (ECMWF) 4

2.1.1 Dataset Spatial and Temporal Attributes 4

2.1.2 ECMWF Dataset Production 7

2.1.3 The GloFAS Project 10

2.2 RAPID 11

2.2.1 How RAPID Works 12

2.3 Geoprocessing Tools 13

2.3.1 Preparing Spatial Data 15

2.3.2 Preparing Temporal Data for RAPID 17

2.3.3 Preparing Static Input Files for RAPID 18

2.4 CI-WATER 21

2.4.1 Tethys Platform 21

2.4.2 Cloud Computing 23

3 Methods **25**

3.1 Software Design - Automation of Computations 26

3.1.1 Distributed Computing 28

3.2 Software Design – Tethys Platform Visualization Web App 32

3.2.1 GIS Visualization 33

3.2.2 Computation and Data Store System 33

3.2.3 Forecast Analysis and Visualization 34
3.3 Experimental Use Case – Comparison with USGS, AHPS and GloFAS 35
3.3.1 Magdalena Watershed 35
3.3.2 HUC-4 Region 1209 Modified Watershed 37
3.4 Experimental Use Case – Automation of Computations 38
3.4.1 Dominican Republic Watershed 39
3.4.2 Hobble Creek Watershed 39
3.4.3 HUC-2 Region 12 Watershed 40
3.4.4 HUC-2 Region 3 West Watershed 41
3.5 Experimental Use Case – Tethys Platform Visualization Web App 42
4 Results 43
4.1 Software Implementation - Automation of Computations 43
4.2 Software Implementation - Tethys Platform Visualization Web App 43
4.3 Use Case Results – Comparison with GloFAS, USGS, and AHPS 46
4.3.1 Magdalena Watershed 46
4.3.2 HUC-4 Region 1209 Modified Watershed 47
4.4 Use Case Results - Automation of Computations 52
4.5 Use Case Results - Tethys Platform Visualization Web App 54
5 Conclusion 59
REFERENCES 61

LIST OF TABLES

Table 4-1 Results of Computation Time Based on Area and the Number of Reaches.....52

Table 4-2 The Limits of Computational Methods Extrapolated.....54

Table 4-3 File Sizes for Watershed Streamflow Forecasts.....57

LIST OF FIGURES

Figure 2-1 Low-Resolution ECMWF Runoff NetCDF File Structure 5
Figure 2-2 High-Resolution ECMWF Runoff Forecast Ensemble.....6
Figure 2-3 Example ECMWF Global Cumulative Runoff Prediction Dataset.....7
Figure 2-4 Atmospheric Processes Included in Runoff Prediction (ECMWF 2013) 8
Figure 2-5 Original TESSEL Land-Surface Scheme (Balsamo et al. 2009) 9
Figure 2-6 Additions to HTESSEL Scheme (Balsamo et al. 2009).....10
Figure 2-7 Dominican Republic HydroSHEDS (left) and GloFAS (right) Watersheds.....11
Figure 2-8 HUC-2 12 Drainage Basin Overlaid with Low-Resolution Forecast.....14
Figure 2-9 Required HydroSHEDS Preprocessing.....16
Figure 2-10 Weight Table Geometry for the Yaque del Sur River, Dominican Republic 17
Figure 2-11 Example Weight Table File 18
Figure 2-12 Basic RAPID Static Input Files with Descriptions 19
Figure 2-13 Example River Network.....20
Figure 2-14 Example of Columns in Network Connectivity File.....20
Figure 2-15 Tethys Platform Diagram (Jones et. al. 2014).....22
Figure 2-16 Diagram of How Starcluster, AWS, and HTCondor Interact 23
Figure 3-1 Schematic of ECMWF-RAPID Downscaling Process 26
Figure 3-2 Workflow to Downscale and Route the Runoff Predictions.....27
Figure 3-3 Ideal Distribution to Cores for Downscaling and Routing Forecasts 29
Figure 3-4 Mutliprocessing Computation Diagram.....30
Figure 3-5 Amazon Web Services Computation Diagram 31
Figure 3-6 Diagram of Creating and Using EBS Image.....32
Figure 3-7 Tethys Computation and Data Storage System Diagram.....34

Figure 3-8 Colombia Watershed with ECMWF Runoff Forecast Grid.....36
Figure 3-9 Magdalena Watershed from GloFAS Drainage Grid.....36
Figure 3-10 Modified HUC-4 Region 1209 with ECMWF Runoff Forecast Grid.....37
Figure 3-11 Modified HUC-4 Region 1209 on GloFAS Drainage Grid 38
Figure 3-12 Dominican Republic Watershed with ECMWF Runoff Forecast Grid 39
Figure 3-13 Hobble Creek Watershed with ECMWF Runoff Forecast Grid 40
Figure 3-14 HUC-2 Region 12 Watershed with ECMWF Runoff Forecast Grid 41
Figure 3-15 HUC-2 Region 3 West Watershed with ECMWF Runoff Forecast Grid 42
Figure 4-1 Overview Image of ERFP-Tool App 44
Figure 4-2 Selecting a Reach in the ERFP-Tool App.....44
Figure 4-3 Selecting a Past Forecast in the ERFP-Tool App.....45
Figure 4-4 Comparison of Outlet of Magdalena Watershed with GloFAS 47
Figure 4-5 Comparison of Outlet of HUC-4 1209 Watershed with GloFAS 48
Figure 4-6 ECMWF Forecast to Gage Data at COMID 5781369 49
Figure 4-7 ECMWF Forecast to Gage Data at COMID 5781369 Zoomed In.....50
Figure 4-8 Comparing ECMWF Forecast to Gage Data at COMID 5781919 51
Figure 4-9 Comparing ECMWF Forecast to Gage Data at COMID 5781369 51
Figure 4-10 Computation Time versus Number of Reaches 52
Figure 4-11 Computation Time versus Watershed Area 53
Figure 4-12 Entire Watershed Zoom Level 55
Figure 4-13 Medium Range Zoom Level 56
Figure 4-14 Closest Zoom Level 56
Figure 4-15 File Sizes versus Watershed Streamflow Forecasts Plot.....58

1 INTRODUCTION

Warning systems with the ability to predict floods days in advance can benefit tens of millions of people. From 2000-2009 more than 950 million people were impacted by floods, over $166 billion in damage occurred, and there were approximately 54,000 deaths (EM-DAT 2010). Improvements in flood forecasting and the ability to communicate actionable information to emergency responders has a massive potential benefit (Pappenberger et al. 2015). Because of this possibility there are multiple efforts going on simultaneously to create systems that will be able to predict streamflow days or weeks in advance.

In the United States, the Advanced Hydrologic Prediction Service (AHPS) is a web-based prediction suite produced by the National Weather Service (NWS) to predict streamflow at approximately 3,600 stream gages across the nation. The AHPS produces streamflow predictions that extend from hours to days and months in advance containing the magnitude and uncertainty of the prediction (NOAA 2015). In Europe, the European Centre for Medium-Range Weather Forecasts (ECMWF) is building the Global Flood Awareness System (GloFAS) that is designed through the use of ensemble forecasts to predict floods in large-scale river basins up to 15 days in advance (Alfieri et al. 2013).

The AHPS is limited in its spatial resolution of predictions to 3,600 stations and GloFAS is limited to predicting streamflow for watersheds with areas on the order of 10's of thousands of kilometers squared. As such, it would be beneficial to produce a forecasting system with a higher density stream network anywhere in the world. Currently, the National Flood Interoperability

Experiment (NFIE) led by the Consortium of Universities for the Advancement of Hydrologic Sciences Inc. (CUAHSI) will attempt to produce real-time hydrologic simulations at a high spatial resolution. In effect, NFIE attempts to expand the spatial resolution of the AHPS from 3600 to nearly 2.7 million locations, which will essentially include street level predictions across the United States (CUAHSI 2015).

Two of the main factors limiting hydrologic modeling for scientists and engineers are the amount of time given to run the models and their computing resources (Humphrey et al. 2012). The limitation exists in many scenarios where scientists and engineers are attempting to create hydrologic models to accurately predict what will happen in the future. Due to the sensitivity of the variables, scientists and engineers run stochastic analyses on the models to produce a statistical analysis of the potential output. However, the number of runs that can occur is limited by amount of time and computing resources. Cloud computing can address this limitation by increasing computational power on an as-needed basis, which in turn decreases the computation time.

For water information to be actionable for those responsible to make decisions, including the public at large, it needs to be presented clearly in a visual and dynamic fashion (Jankowski 1995; Pappenberger et al. 2013; Shim et al. 2002). Web GIS mapping technologies have transformed the way information is disseminated broadly including examples such as Google Maps API (Blankenhorn 2005; Boulos 2005). Additionally, there has been an increased use of web applications to give decision makers the information that they can interpret and use without needing the technical background (Swain et al. 2015).

The goal of this research is to implement a method for creating and visualizing high-density streamflow two-week ensemble forecasts in the United States and potentially across the world. To meet this goal the following specific objectives were achieved:

(1) Develop a method for downscaling the global ECMWF runoff forecasts and convert the data so that it can be routed through a stream network of any scale using the Routing Application for Parallel computatIon of Discharge (RAPID) program;

(2) Use multiprocessing methods and the computational power of the cloud-based Amazon Web Services (AWS) to improve computation time;

(3) Create a web- and map-based graphical user interface in Tethys Platform for visualizing the high density stream forecasts.

This thesis is organized as follows. Chapter 2 presents the background of ECMWF, RAPID, cloud computing, and the Tethys Platform. Chapter 3 presents the methods regarding how the datasets were downscaled, how the computation time was improved, and how the web app was developed. In Chapter 4, the results of the implementation are discussed and some conclusions and observations are presented in Chapter 5.

2 BACKGROUND AND RELATED WORK

In this Chapter, the background for (1) the ECMWF runoff prediction dataset used in the GloFAS project, (2) the RAPID program to route the runoff through the reaches, (3) the Amazon Cloud and HTCondor to perform the computations across a distributed network of computers, and (4) the Tethys Platform as the development environment for creating a cloud-based user interface for viewing the high density streamflow forecast data are discussed.

2.1 European Centre for Medium-Range Weather Forecasts (ECMWF)

The ECMWF is a research center that specializes in medium range weather forecasting and was formed as a collaboration of several countries within the World Meteorological Organization's Region V. They produce global datasets for a variety of different meteorological, hydrological, and ocean parameters. For this thesis, the ECMWF global gridded runoff prediction dataset is used, which includes surface and subsurface runoff depth in meters derived at 12-hour intervals from their land-surface model for a 15-day forecast.

2.1.1 Dataset Spatial and Temporal Attributes

Within the global runoff dataset, there are 52 separate ensembles or weather predictions that estimate cumulative runoff depths in meters originally output in the GRIB2 format that is subsequently converted to the NetCDF4 format for this research. The first 51 ensembles are created

at a lower resolution on a ~0.28-degree grid cell (Reduced Gaussian Grid N320) and varying time resolution, with a time step of 3 hours until hour 96 and then 6 hours thereafter for a combined duration of 15 days (apart from Thursdays when a 32 day forecast is provided). In the low-resolution grid, a uniform 6-hour time step for the entire 15 days was generated to simplify the calculations for the datasets used in this research. The variables and dimensions of the low-resolution ensemble are shown in the code of Figure 2-1.

```
dimensions:
  lon = 1280;
  time = UNLIMITED;   // (61 currently
  lat = 640;
variables:
  double lon(lon=1280);
    :long_name = "longitude";
    :units = "degrees_east";
    :standard_name = "longitude";
    :axis = "X";

  float RO(time=61, lat=640, lon=1280);
    :code = 205; // int
    :table = 128; // int
    :grid_type = "gaussian";
    :long_name = "Runoff";
    :units = "m";
    :_ChunkSize = 1, 640, 1280; // int

  double time(time=61);
    :units = "hours since 2014-10-31 00:00:00";
    :calendar = "proleptic_gregorian";
    :_ChunkSize = 1; // int

  double lat(lat=640);
    :long_name = "latitude";
    :units = "degrees_north";
    :standard_name = "latitude";
    :axis = "Y";
```

Figure 2-1 Low-Resolution ECMWF Runoff NetCDF File Structure

The 52nd forecast is produced at a higher resolution with a ~0.14-degree grid cell (Reduced Gaussian Grid N640) and a varying time step over 10 days. For the high-resolution dataset, the

time step for hours 0 to 90 is 1-hour; from hour 90 to hour 150 the data is in 3-hour time steps, and from hour 150 to hour 240 the data is in 6-hour time steps. The variables and dimensions of the high-resolution ensemble are shown in the code of Figure 2-2.

```
dimensions:
  lon = 2560;
  time = UNLIMITED;   // (125 currently
  lat = 1280;
variables:
  double lon(lon=2560);
    :long_name = "longitude";
    :units = "degrees_east";
    :standard_name = "longitude";
    :axis = "X";

  float RO(time=125, lat=1280, lon=2560);
    :code = 205; // int
    :table = 128; // int
    :grid_type = "gaussian";
    :long_name = "Runoff";
    :units = "m";
    :_ChunkSize = 1, 1280, 2560; // int

  double time(time=125);
    :units = "hours since 2014-10-31 00:00:00";
    :calendar = "proleptic_gregorian";
    :_ChunkSize = 1; // int

  double lat(lat=1280);
    :long_name = "latitude";
    :units = "degrees_north";
    :standard_name = "latitude";
    :axis = "Y";
```

Figure 2-2 High-Resolution ECMWF Runoff Forecast Ensemble

An example ECMWF cumulative runoff prediction ensemble at an arbitrary time step is displayed in Figure 2-3.

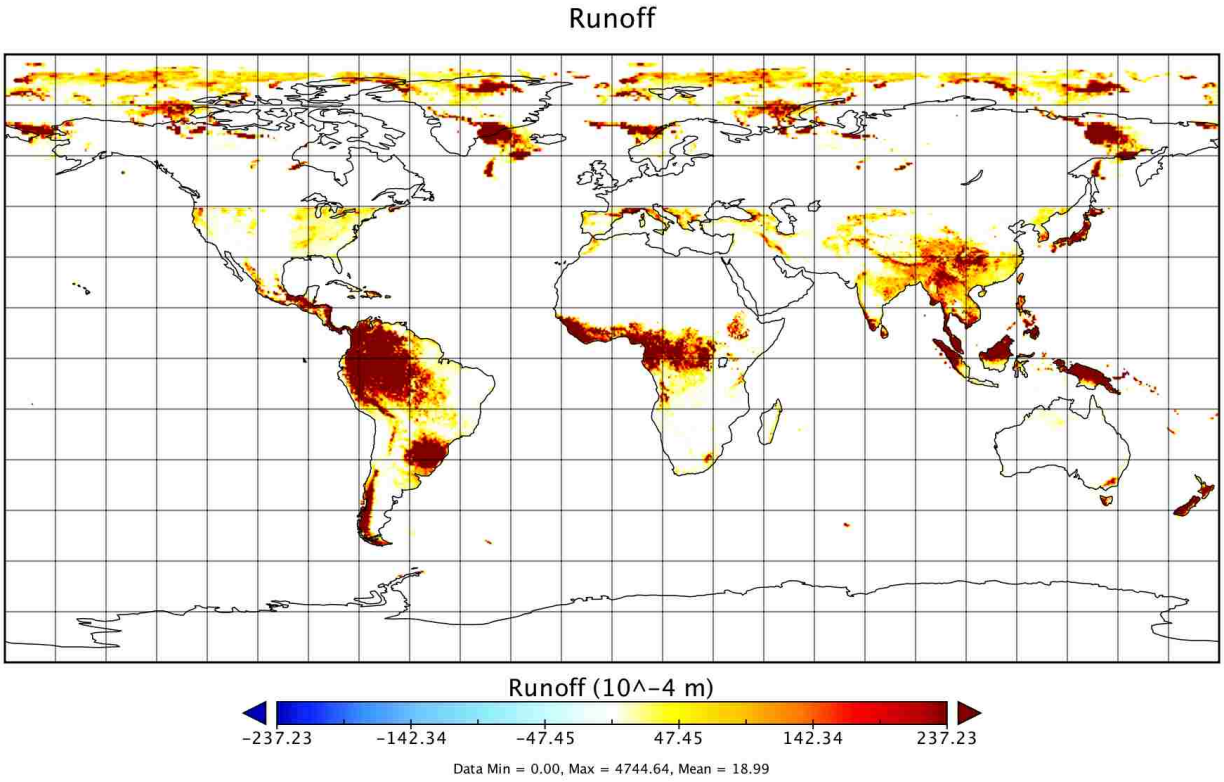

Figure 2-3 Example ECMWF Global Cumulative Runoff Prediction Dataset

2.1.2 ECMWF Dataset Production

The ECMWF global runoff prediction datasets are produced using an atmospheric-land-surface model of physical processes that derive runoff depths from a meteorological forecast. An essential element of the model is the simulation of the flow of moisture within the atmosphere. The atmospheric processes include radiation transfer, turbulent mixing, convection, clouds, surface exchange, subgrid-scale orographic drag and non-orographic gravity wave drag (ECMWF 2013). From these atmospheric processes, forecasts for precipitation, temperature, and cloud cover are determined. The atmospheric processes are illustrated in Figure 2-4.

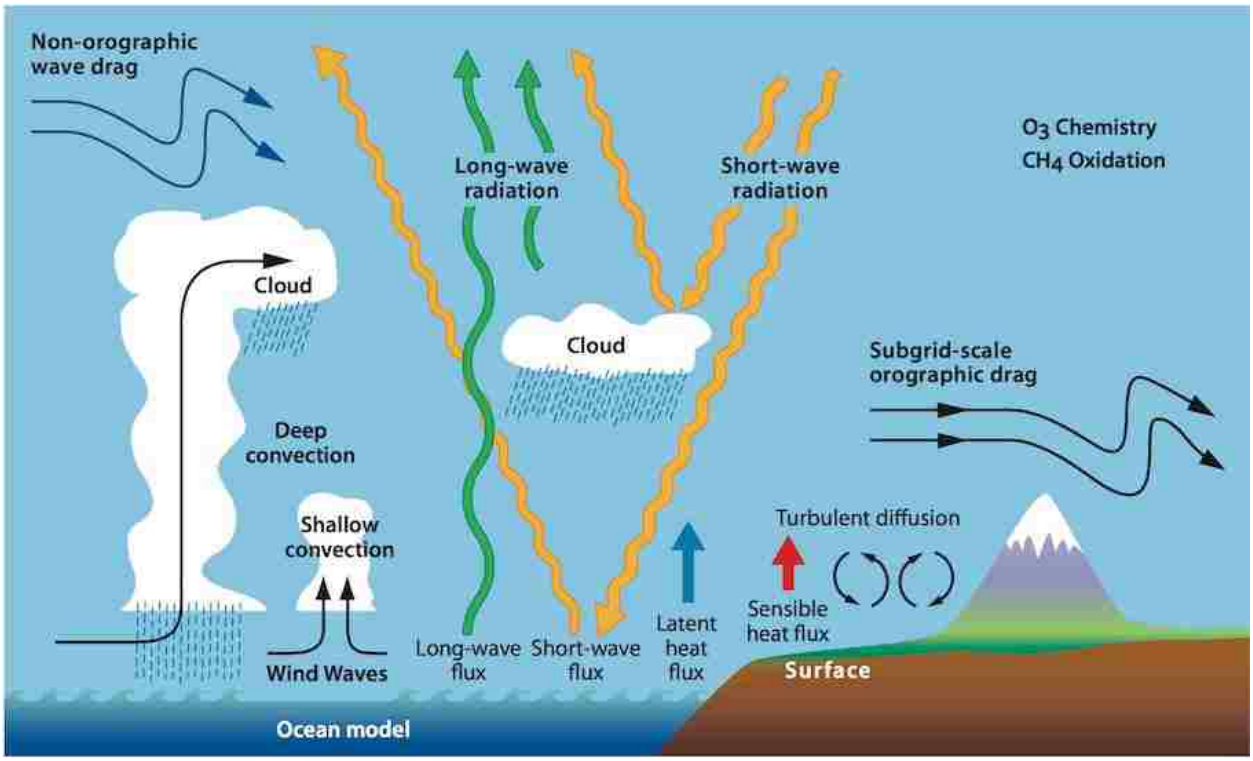

Figure 2-4 Atmospheric Processes Included in Runoff Prediction (ECMWF 2013)

Another essential element of the model is the surface runoff processes. Because the atmosphere and surface processes are interconnected, a scheme was created to provide an interface between the two. The original surface parameterization scheme used to represent the change in water, energy, and sub-surface elements is known as the Tiled ECMWF Scheme for Surface Exchanges over Land (TESSEL). The TESSEL scheme uses a tiled grid system where each grid has up to six classifications which includes bare ground, low and high vegetation, intercepted water, and shaded and exposed snow (ECMWF 2013). Each classification determines how the changes in heat and water occur over the tiled grid. A schematic of the different classifications used in the model is shown in Figure 2-5.

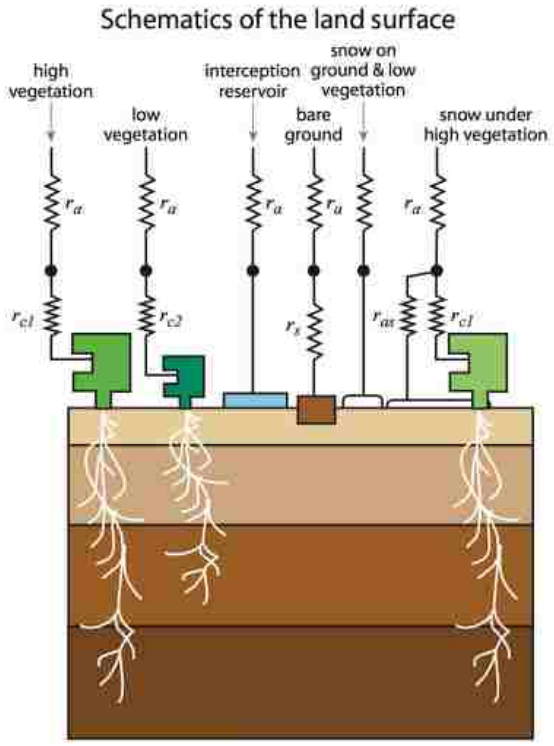

Figure 2-5 Original TESSEL Land-Surface Scheme (Balsamo et al. 2009)

Although the TESSEL scheme includes many different processes to help accurately predict how the atmospheric and land-surface processes interact, it is limited by the assumption of a uniform soil type and surface orography. As such, surface runoff is not included in the model. To accurately predict surface runoff, ECMWF modified the TESSEL scheme to add soil texture on a cell-by-cell basis instead of a global soil type and also incorporated the orography of the land. With the addition of soil texture and orography, runoff and infiltration schemes were then added to the model. The new scheme was renamed HTESSEL because it added more information about the land surface hydrology as depicted conceptually in Figure 2-6 (Balsamo et al. 2009).

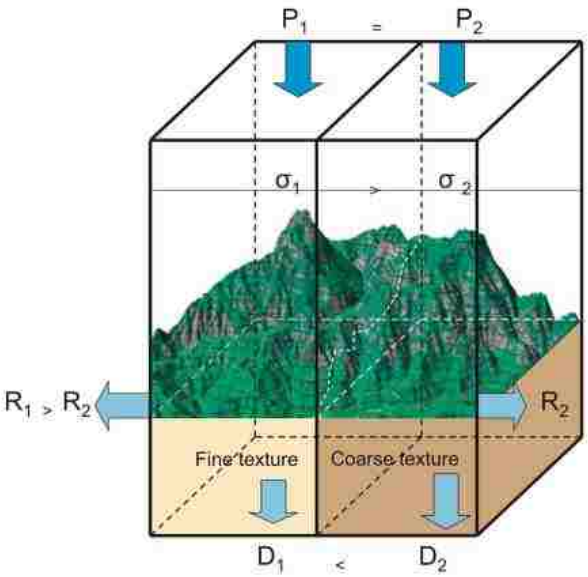

Figure 2-6 Additions to HTESSEL Scheme (Balsamo et al. 2009)

Before creating each prediction dataset, the initial conditions are determined using a data assimilation cycle (ECMWF 2011). To initialize the land surface model, ECMWF used the Land Data Assimilation System (LDAS) to gather information about soil moisture, snow depth, snow temperature, and soil temperature. LDAS combines information from the HTESSEL model, the meteorological situation, and available observations to initialize the next ECMWF runoff forecast model run (De Rosnay et al. 2011).

2.1.3 The GloFAS Project

The Global Flood Awareness System (GloFAS) is designed to predict floods globally for large-scale river basins, and is jointly developed by the European Commission's Joint Research Council (JRC) and ECMWF. Streamflow forecasts are generated using the ECMWF HTESSEL runoff prediction scheme and routed through the river network using the overland flow kinematic wave approach. The river network is generated by upscaling the HydroSHEDS grid to 0.1 degrees

(Alfieri et al. 2013). The 0.1-degree grid cell size ranges from 50 to 115 sq km between the 67° N and 67° S latitude lines. At this resolution, it is difficult to capture streamflow within a local stream or river. For example, Figure 2-7 shows the 5,204 sq km Yaque Del Sur River watershed in the Dominican Republic delineated using the 15 arc-second HydroSHEDS grid on the left versus the 0.1 degree gridded GloFAS drainage network on the right.

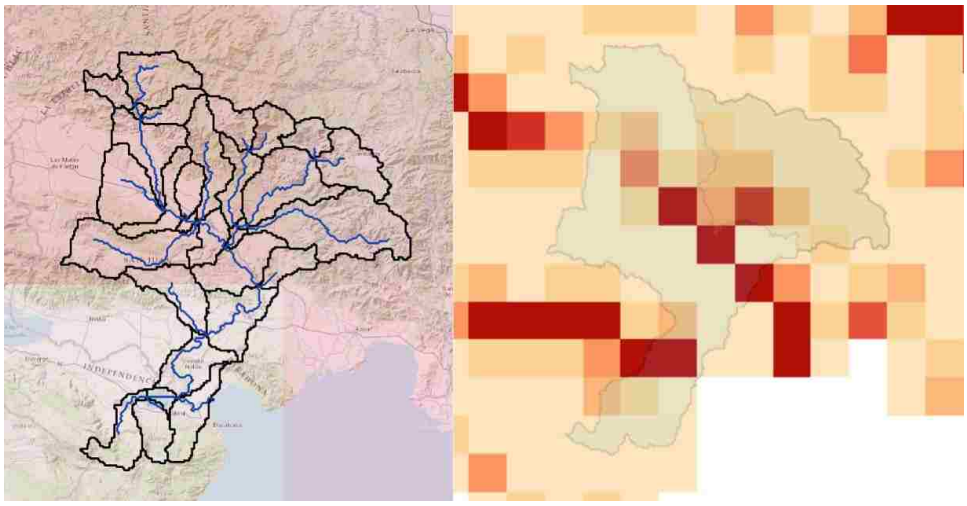

Figure 2-7 Dominican Republic HydroSHEDS (left) and GloFAS (right) Watersheds

Note that the drainage network at such a coarse scale is scarcely apparent, and that only major streams appear in the GloFAS network shown by the darkened grid cells. Due to this limitation, an alternate method needs to be used to apply the ECMWF runoff prediction dataset to smaller watersheds derived from the catchments of the higher density stream networks.

2.2 RAPID

RAPID is an open source model used to route runoff of surface and groundwater inflow to rivers downstream with any density stream network (David et al. 2011), which enables the

ECMWF runoff forecasts to be "downscaled" to smaller watersheds so that predictions are available at a much higher resolution. Additionally, the input for runoff used by RAPID closely aligns with the ECMWF runoff dataset; thus, it is possible to format the time series from the ECMWF runoff dataset as an input for RAPID. However, in order to prepare the input files for RAPID, geoprocessing tools are necessary to downscale the ECMWF gridded global runoff dataset to a higher resolution watershed boundary dataset. These tools will be discussed in Section 2.3.

2.2.1 How RAPID Works

The RAPID program uses the matrix form of the Muskingum routing method to route water in a river network. The traditional Muskingum routing method has two main parameters *k* and *x,* where *k* is the storage constant with a time dimension and *x* characterizes reach properties that contribute to attenuation, is dimensionless, and is stable from 0 to 0.5 (Cunge 1969). The finite difference form of the equation for the Muskingum method is shown in Equation 2-1 (Singh and McCann 1980).

$$Q_n = C_0 I_n + C_1 I_{n-1} + C_2 Q_{n-1} \quad (2\text{-}1)$$

In the equation, $C_0 = (-kx + 0.5\Delta t)/C_3$, $C_1 = (kx + 0.5\Delta t)/C_3$, $C_2 = (k - kx + 0.5\Delta t)/C_3$, and $C_3 = (k - kx - 0.5\Delta t)$ where Δt is the routing time step; Q_n is the rate of outflow from the reach at that time step; I_n is the inflow for the reach at that time step; I_{n-1} is the inflow for the reach at the previous time step; and Q_{n-1} is the rate of outflow from the reach during the previous time step (Singh and McCann 1980). The matrix form of the Muskingum equation used is shown in Equation 2-2 (David et al. 2011).

$$(\mathbf{I} - \mathbf{C}_1 \cdot \mathbf{N}) \cdot \mathbf{Q}(t + \Delta t) = \mathbf{C}_1 \cdot \mathbf{Q}^e(t) + \mathbf{C}_2 \cdot [\mathbf{N} \cdot \mathbf{Q}(t) + \mathbf{Q}^e(t)] + \mathbf{C}_3 \cdot \mathbf{Q}(t) \quad (2\text{-}2)$$

In this equation t is time and Δt is the river routing time step. In the equation: **I** is the identity matrix and **N** is the river network matrix; $\mathbf{C_1}$, $\mathbf{C_2}$, and $\mathbf{C_3}$ are parameter matrices; **Q** is a vector of outflows from each reach and $\mathbf{Q^e}$ is a vector of lateral inflows for each reach (David et al. 2011).

One notable assumption of the model is that it assumes that there is one sub-basin per reach. As such, all of the runoff in that sub-basin will be routed into the stream connected with the sub-basin as denoted by $\mathbf{Q^c}$. Another important factor in the matrix-based Muskingum equation is that each reach j can have its own values of $\mathbf{C_1}$, $\mathbf{C_2}$, and $\mathbf{C_3}$ because each reach can have its own k and x factors (David et al. 2011).

2.3 Geoprocessing Tools

Geoprocessing tools in GIS software, such as Esri's ArcMap, are ideal for performing the geometric data processing required to downscale the gridded ECMWF forecasts to a series of polygon catchments and river networks. To format the data for RAPID, the following geoprocessing tasks are necessary: (1) the watershed spatial geometry and location needs to be determined, (2) how the geometry and location of the watershed relates to the ECMWF runoff forecast needs to be determined, (3) a method for formatting the input from the ECMWF runoff forecast for RAPID needs to be created, (4) creating a workflow to determine what the Muskingum parameters are based off of the geometry of the watershed is required, (5) and the river topology needs to be defined such that RAPID knows how upstream and downstream reaches are connected.

Esri created a toolbox and added tools to the ArcHydro toolbox specifically for formatting data for RAPID. Using the geometry and location of the meteorological global dataset grid and based on the spatial location and geometry of the drainage basins in the watershed, the RAPID toolbox has a tool that determines how much area of the drainage basin is in each grid cell within the ECMWF runoff prediction file. The ECMWF runoff prediction grid overlaid on the HUC-2, Region 12 watershed is shown in Figure 2-8.

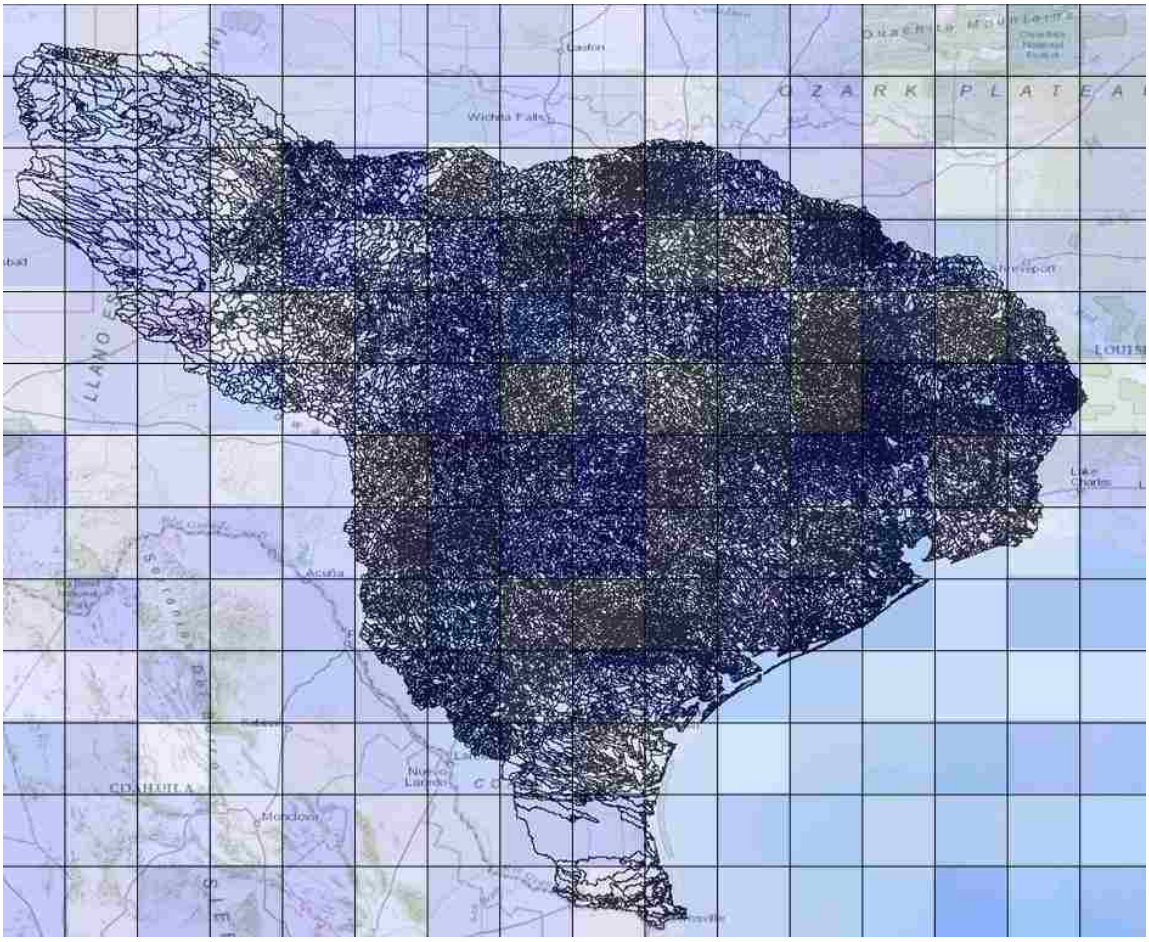

Figure 2-8 HUC-2 12 Drainage Basin Overlaid with Low-Resolution Forecast

The volume of runoff for each catchment is computed as a weighted average of the overlying area within each of the runoff grid cells after converting the cumulative runoff volume at each time step to be incremental. The result is a NetCDF3-Classic formatted file compatible with RAPID that has the incremental runoff defined at each time step for each subbasin in the watershed.

There also need to be tools that can create the necessary input files that are formatted for RAPID based on the geometry and topology of the reaches and subbasins. These files include the Muskingum *k* and *x* values for each reach, as well as the reach connectivity.

2.3.1 Preparing Spatial Data

To create the necessary input for RAPID a drainage network with drainage basins and reaches in a shapefile or similar format is required. Data from the NHDPlus Version 2 and from HydroSHEDS are useful products to obtain the basin and stream network information. However, the necessary drainage network and drainage basins can be obtained with alternate GIS datasets and watershed delineation methods.

The USGS provides GIS shapefiles representing both the drainage network and the drainage basins inside of the United States. The drainage network shapefile dataset is titled the National Hydrography Dataset (NHD) and includes features including streams, rivers, lakes, and ponds. The drainage basin shapefile dataset is titled the Watershed Boundary Dataset (WBD) (USGS 2014). NHDPlus Version 2 combines the NHD, WBD, and National Elevation Datasets (NED) and adds attributes including stream order and attributes that facilitate rapid stream network traversal and query (Corporation 2011). The NHDPlus Version 2 datasets provides the necessary attributes and stream topology to create RAPID input.

The HydroSHEDS datasets were created by the World Wildlife Fund Conservation Science Program. Contributors to the project include the USGS, the International Centre for Tropical

Agriculture, the Nature Conservancy, and the Center for Environmental Systems Research of the University of Kassel, Germany (Lehner et al. 2008). The flow direction and flow accumulation rasters allow for the delineation of drainage basins along with the creation of a drainage network.

When using the HydroSHEDS datasets, an extra preprocessing step to create a delineated drainage network with subbasins and reaches is necessary. The 'Basic Dendritic Terrain Processing – No Fdr & Fac' tool in the 'AHPreprocessing' toolbox can process the flow direction and the flow accumulation rasters to produce a drainage network with drainage basins. A diagram showing the inputs and outputs for this preprocessing step for the Yaque del Sur River watershed in the Dominican Republic is shown in Figure 2-9.

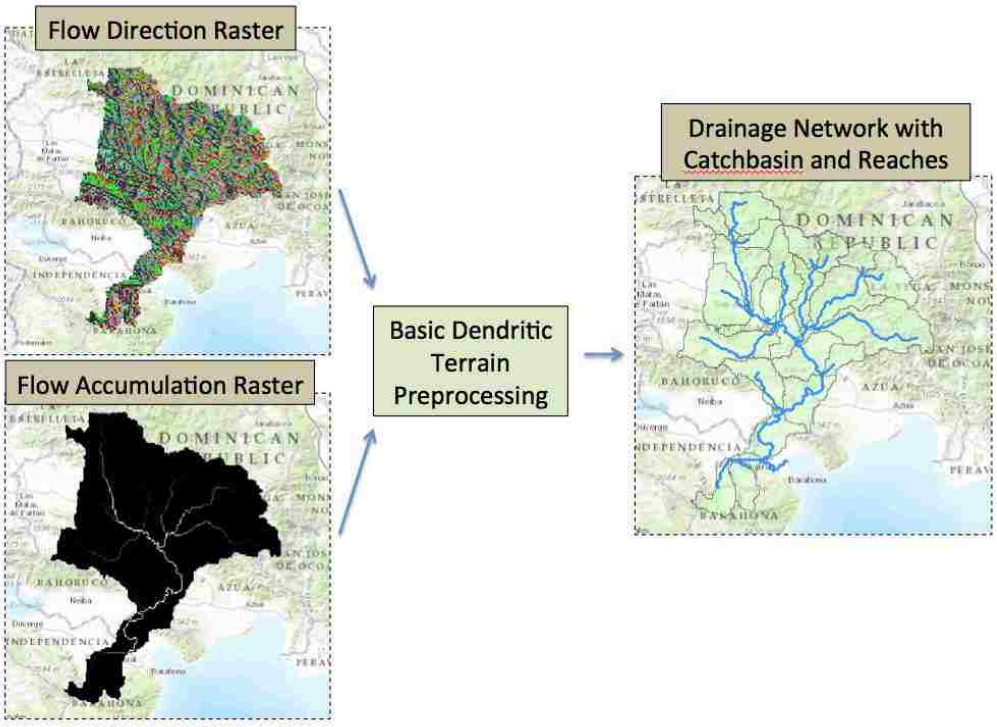

Figure 2-9 Required HydroSHEDS Preprocessing

2.3.2 Preparing Temporal Data for RAPID

The spatial location and geometry of the drainage basins are used by the 'Create Weight Table from ECMWF Runoff' geoprocessing tool in the RAPID toolbox to determine how much area of each catchment is in each ECMWF runoff prediction grid cell.

Figure 2-10 shows the ECMWF grid cells overlaid on the Yaque del Sur River watershed in the Dominican Republic. Here it is evident why it is necessary to create a weighting scheme to apply the prediction values from multiple cells to the appropriate watersheds.

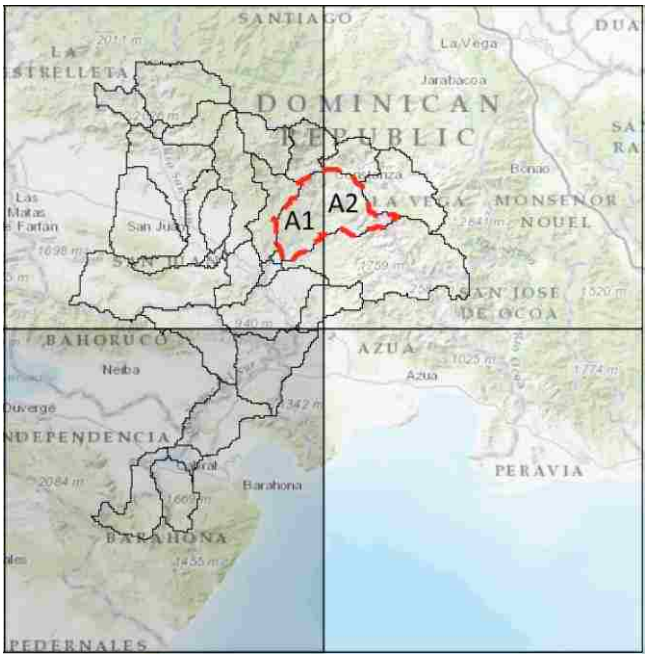

Figure 2-10 Weight Table Geometry for the Yaque del Sur River, Dominican Republic

The output of this tool is a weight table file that is created once and can then be reused each time you convert the global runoff depth prediction dataset into a NetCDF3 Classic file formatted for RAPID containing a time series of volume of runoff into each drainage basin. An example portion of the weight table file is shown in Figure 2-11.

FEATUREID	area_sqm	lon_index	lat_index	npoints	weight	Lon	Lat
101	2066658.85	943	209	2	0.45255957	-94.78125	31.053854
101	2499941.81	944	209	2	0.54744043	-94.5	31.053854
...	...	...	...	...	...	...	...

Figure 2-11 Example Weight Table File

Because the routing computations are performed on the subbasins, reach connectivity information is important to map the inflows from the subbasins to their respective reach in the inflow file. This is accomplished using the 'Add DrainLnID to Catchment' tool in the ArcHydro toolbox.

The tool titled 'Create Inflow File from ECMWF Runoff' calculates the incremental volume of runoff to each reach at each time step and outputs the data into the NetCDF3 format. Runoff is calculated as the sum of the depth of runoff in the grid cell at the lon_index and lat_index multiplied by the area (in square meters) is needed for each FEATUREID.

2.3.3 Preparing Static Input Files for RAPID

For a RAPID simulation to be run, information about the river network needs to be combined into csv files for RAPID to process. These files include the river network connectivity as well as the Muskingum parameters for each reach. The basic static RAPID input files used with descriptions are shown in Figure 2-12.

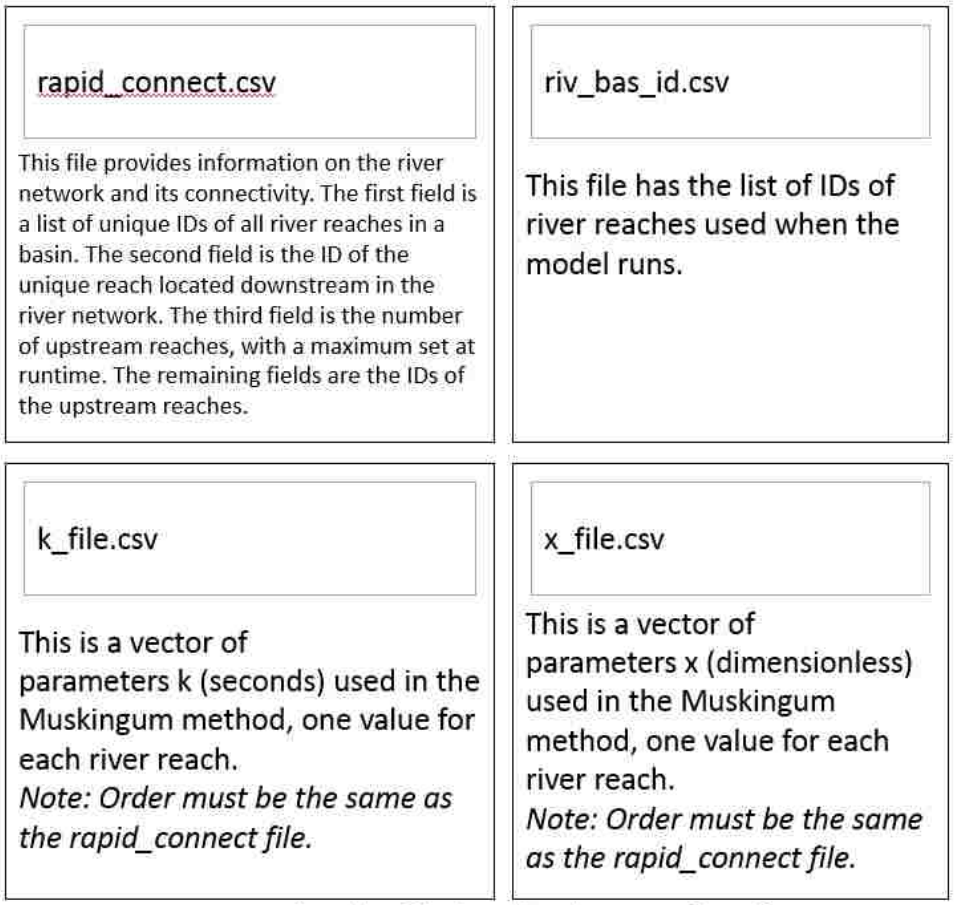

Figure 2-12 Basic RAPID Static Input Files with Descriptions

The ArcHydro toolbox has a tool called 'Calculate the Muskingum Parameters' which calculates the Muskingum k and x values for each stream segment. The value of k is can be associated with the travel time of the reach, where travel time = distance / flow wave velocity. This tool calculates k by weighting the multiplying factor λ_k given by the user (default value of 0.35 and is $1/flow\ wave\ velocity$ with units of seconds/km) by length of the stream segment in kilometers assuming a flow wave velocity of 1 km/hr or 1 km/3600 seconds (e.g. $k = [\lambda_k]$ *[stream segment length (km)] / [1/3600 (km/seconds)]). This tool also assigns the value of x to each stream segment based on the multiplying factor $\lambda_x \in [0,5]$ given by the user (default value of 3) (e.g. x =

λ_x*0.1). The RAPID Muskingum parameter files (k_file.csv and x_file.csv) are created from the calculated *k* and *x* values using the 'Create Muskingum Parameter Files' tool in the RAPID toolbox.

Based on the stream network generated in the spatial datasets, the river connectivity file (rapid_connect.csv) needs to be created. This file tells RAPID which streams are upstream and downstream from each other. A schematic of a river network with IDs assigned is shown in Figure 2-13 and the related portion of the connectivity file is shown in Figure 2-14.

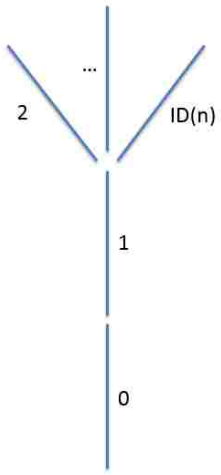

Figure 2-13 Example River Network

Reach ID	Next Downstream Reach ID	Number of Upstream Reaches	Upstream Reach ID 1	...	Upstream Reach ID n
1	0	n	2	...	ID(n)
...	...	...	...	...	...

Figure 2-14 Example of Columns in Network Connectivity File

Based on the stream network generated in the spatial datasets, the river connectivity file (rapid_connect.csv) can be created with the 'Create Network Connectivity File' tool. The last file needed is the 'riv_bas_id.csv' file. This file tells RAPID what subset of reaches to use in the model run. This is created with the 'Create Subset File' tool.

2.4 CI-WATER

The NSF-funded CI-WATER project titled "A Utah-Wyoming Cyberinfrastructure Water Modeling Collaboration", under Grant No. 1135483, has the objectives to: (1) enhance cyberinfrastructure facilities, (2) enhance access to data and computationally intensive modeling, (3) advance high-resolution multi-physics watershed modeling, and (3) provide STEM learning and water science engagement. The BYU research team has focused on the second objective, which has resulted in a web app development platform called Tethys Platform. Tethys Platform is designed for lowering the barrier to creating water resources web apps that can interact with cloud-distributed computing system such as the Amazon Web Services (AWS) (Jones et al. 2014). With Tethys Platform, web apps can be created for decision makers to have access to the results of hydrologic simulations in a comprehensible format.

2.4.1 Tethys Platform

Tethys Platform makes it easier for water resource developers to create web apps by addressing unique needs such as handling geospatial data and computational power. With Tethys, developers can compose web apps that can use the computational power of a distributed computing cluster to run hydrologic programs to process and produce large quantities of data and present them on maps in a meaningful way to decision makers.

Tethys Platform provides a suite of software, built on the Django web framework, to address the specialized needs of water resources web apps and a Python software development kit (SDK) for incorporating the functionality of the software into the web apps. The software suite includes web GIS data storage via PostGIS, geoprocessing with 52 North WPS, and GIS data publishing with GeoServer. There are also visualization tools such as Google Maps™ and OpenLayers for displaying GIS data and Highcharts for creating plots. The software suite also includes a computing component with HTCondor and TethysCluster to manage distributing computations. Figure 2-15 shows the overall software architecture for the Tethys Platform.

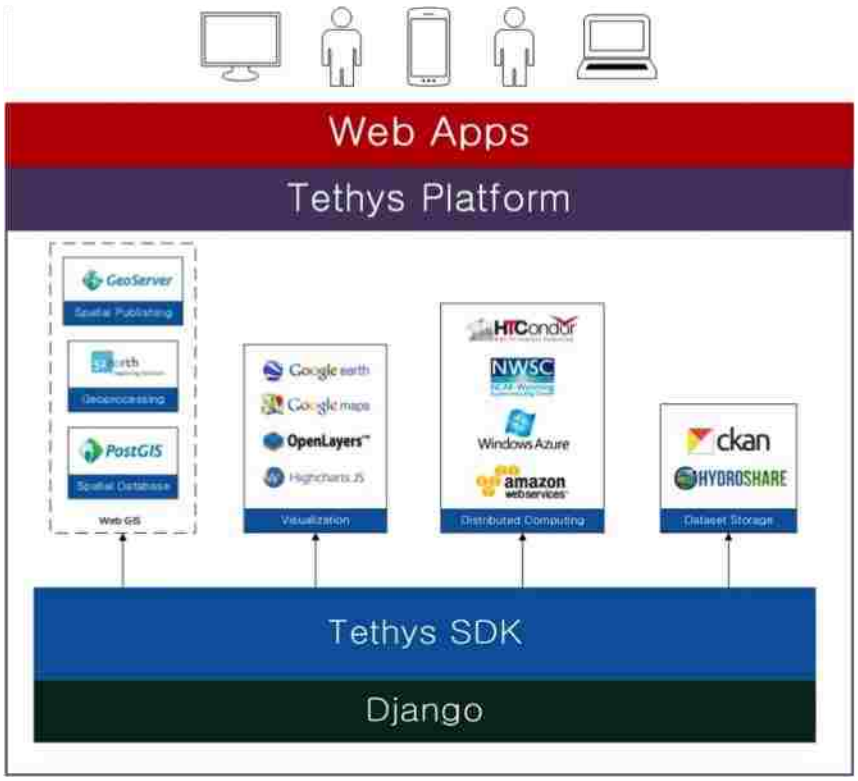

Figure 2-15 Tethys Platform Diagram (Jones et. al. 2014)

In short, Tethys Platform provides the tools needed to create a web application to visualize and interact with large GIS data sets.

2.4.2 Cloud Computing

An essential component of the Tethys Platform is cloud computing. The main components that will enable the distribution of computations are AWS, Starcluster, and HTCondor. AWS provides a pool of computing resources in the cloud where jobs can be created and distributed between the nodes of computers. Starcluster is software created by Justin Riley at the Massachusetts Institute of Technology (MIT) for AWS to create and manage clusters of computers. It enables provisioning a cluster of Virtual Machines (VMs) that are connected with HTCondor, a job and computing resource management software. A diagram of how Starcluster, AWS, and HTCondor interact is shown in Figure 2-16.

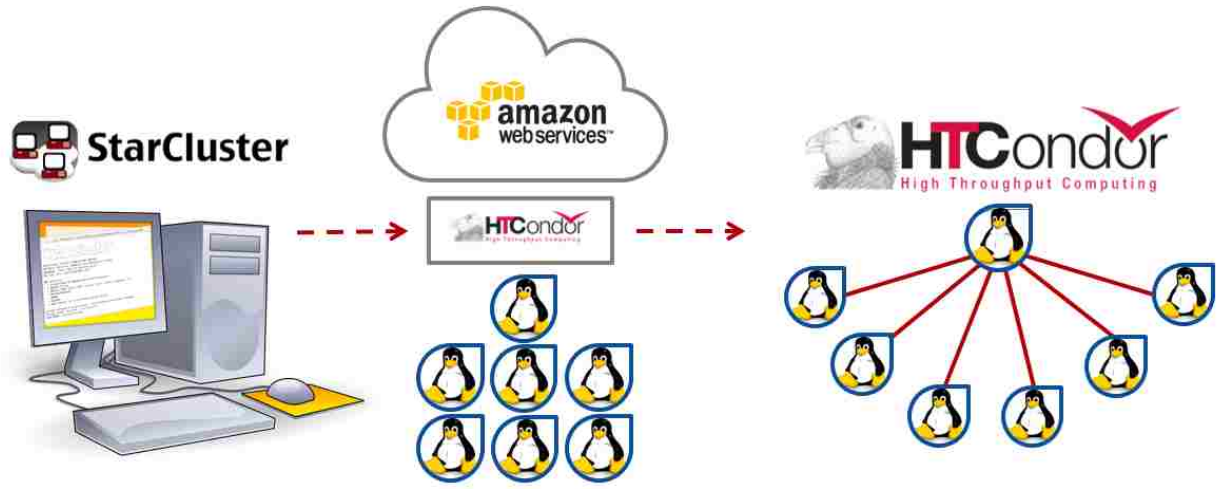

Figure 2-16 Diagram of How Starcluster, AWS, and HTCondor Interact

Customized clusters of computing resources are managed for programmatic creation and modification in AWS using Starcluster's Python library.

HTCondor is a commonly used batch-scheduling and resource management software that distributes jobs to computing resources based on their availability (Buyya et al. 2013). HTCondor

can be added to AWS as a plugin using Starcluster and manages when and where the jobs are submitted. There is a Python library named Condorpy (https://pypi.python.org/pypi/condorpy) that facilitates programmatic HTCondor job submittal.

3 METHODS

The goal of this thesis is to produce higher spatial resolution streamflow predictions using the ECMWF runoff prediction datasets complete with a method for viewing the predictions. To achieve this goal the ECMWF runoff prediction dataset needs to be downscaled to a watershed; the runoff needs to be routed through the reaches within the watershed using an in-stream routing model (in this case, RAPID); and a method for presenting high resolution stream forecasts to decision makers in a standardized, intuitive format needs to be developed.

The overall process for downscaling and routing the ECMWF datasets using RAPID is shown in Figure 3-1. In step (1) the forecast ensembles are downloaded to the server. In step (2) the computations for (2a) using geoprocessing to downscale the ECMWF runoff forecast and (2b) routing the forecasted runoff through the reaches using RAPID. In step (3) the in-stream forecasts are sent to a data store such as CKAN (http://docs.ckan.org/en/ckan-2.2/datastore.html) or HydroShare (Tarboton et al. 2014). Finally, in step (4) the Tethys Platform web app downloads the recent forecasts to display them to the user.

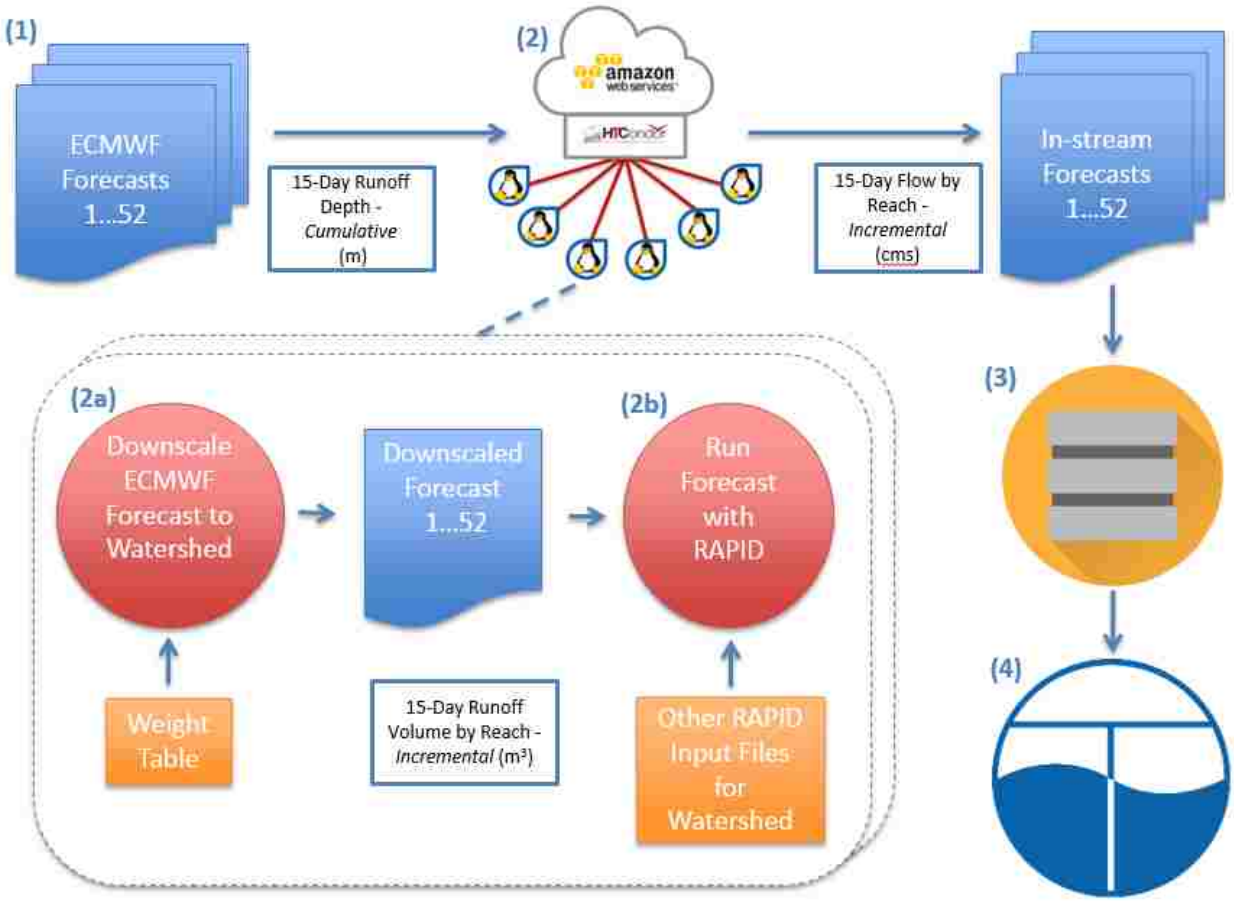

Figure 3-1 Schematic of ECMWF-RAPID Downscaling Process

This Chapter presents a method for automating the downscaling of ECMWF runoff prediction datasets and routing the runoff with RAPID as well as a web application created using Tethys Platform for viewing the output. Additionally, this Chapter presents methods for testing and validating the methods presented herein.

3.1 Software Design - Automation of Computations

The 52 separate ensembles produced in the ECMWF runoff each need to be downscaled and the volume of runoff in each drainage basin computed at each time step. The main steps of the process are (1) download and extract the ECMWF runoff forecast; (2) loop through forecasts and

(2a) downscale ECMWF runoff forecast to produce RAPID runoff input, then (2b) run RAPID with static input files and the runoff forecast; and (3) upload the stream forecasts results from RAPID to a data store. A workflow diagram of the main elements of the loop in step two is shown in Figure 3-2.

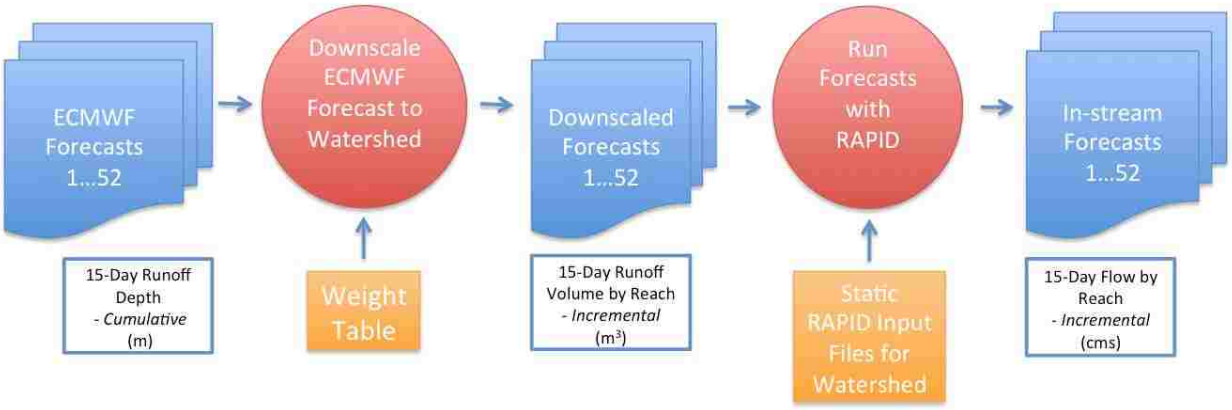

Figure 3-2 Workflow to Downscale and Route the Runoff Predictions

The computations required to downscale the watershed's runoff for RAPID and route the water into the streams could take over 12 hours if the number of subbasins in the watershed is extremely large, as is the case for a dense stream network. The main problem with this extended duration of computation is that new prediction datasets are produced every 12 hours. Thus, in the absence of efficient methods, the computations from the last dataset will not be completed before a new set of computations commences. When this happens, a cycle of computing lag will occur. In order to prevent this situation, alternative methods need to be incorporated in the computation process in order to downscale the watersheds in a timely manner and enable the computing system to be ready to process the next dataset. We did not encounter this in our research due to the limited size and stream density of the watersheds used in this study. However, because the potential for

this to occur, methods were researched to enable the setup for larger number of subbasins derived from dense stream networks.

3.1.1 Distributed Computing

One method to improve time is to distribute the computations between computer processors or computing nodes. The distribution method could be applied to a single server with multiple processing cores or to a cluster of computers. If the watershed is small and the stream network is sparse, the requirement for computing power is not as demanding as it would be for a large watershed with a dense stream network.

There are several ways to improve the time of computations. In this project I used multiprocessing on a single computer and Amazon Web Services (AWS). Using the multiprocessing Python library, the jobs can be distributed between the processing cores on a single computer. If AWS is available, the jobs can be distributed between computing nodes. The ideal setup to minimize the computation time would be to have a single computational run per computing node. However, depending on the number of nodes available, this may be impossible, and in such cases distributing the jobs as evenly as possible will minimize computation times. A diagram for distributing the jobs between computing nodes is shown in Figure 3-3.

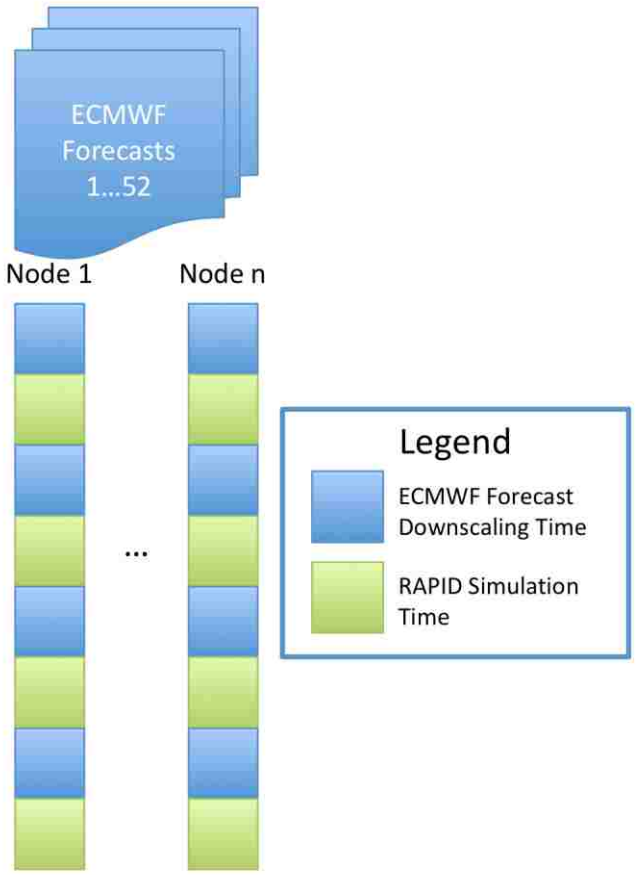

Figure 3-3 Ideal Distribution to Cores for Downscaling and Routing Forecasts

Several instances of RAPID can be running simultaneously, however they cannot start at the same time. The reason several instances cannot start at the same time is because RAPID needs to read the namelist file, which provides information about what files to read in and other important parameters. If other instances are attempting to modify the file before one instance of RAPID finishes reading the file, errors can occur. One method I used to get around this problem uses locks from the multiprocessing library. When each instance of RAPID needs to set up the namelist file and read it in to start that instance, it waits for a lock to be released from another instance trying to start at the same time. Once the lock is acquired, it can then read the necessary information from

the namelist file, initialize, and then release the lock to allow the next instance of RAPID to start. Another method I used was to create separate execute directories and create a symbolic link to the RAPID program in each directory. This method works when using HTCondor as there are separate working directories created automatically for each job submitted.

3.1.1.1 Multiprocessing Method

On a computer with four processing cores I installed the RAPID program, Python, and netCDF4-python library. I used the Python multiprocessing library to manage to jobs distributed between the computing cores. Then, I set up the code such that the downscaling and routing processes were combined and sent out to a computational node. The reason for separating the code in this method is because each process in looping through the forecasts is independent of the other iterations. A diagram of the workflow for the computations using the multiprocessing method is shown in Figure 3-4.

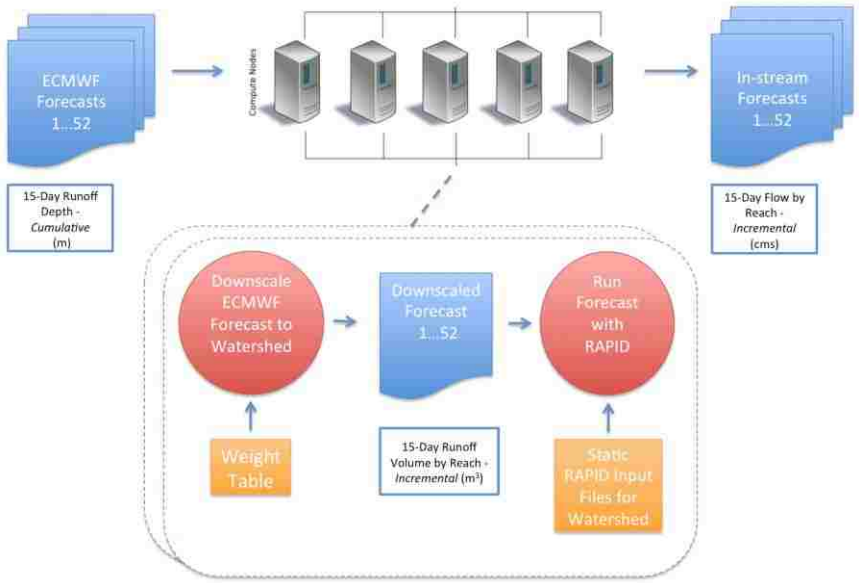

Figure 3-4 Mutliprocessing Computation Diagram

3.1.1.2 Amazon Web Services Distribution Method

Using AWS, I created a cluster of computers managed by the HTCondor software. Similar to the multiprocessing method, I created a setup with AWS computing nodes with everything else held constant. A diagram for the methodology for computations in the AWS computer clusters is shown in Figure 3-5.

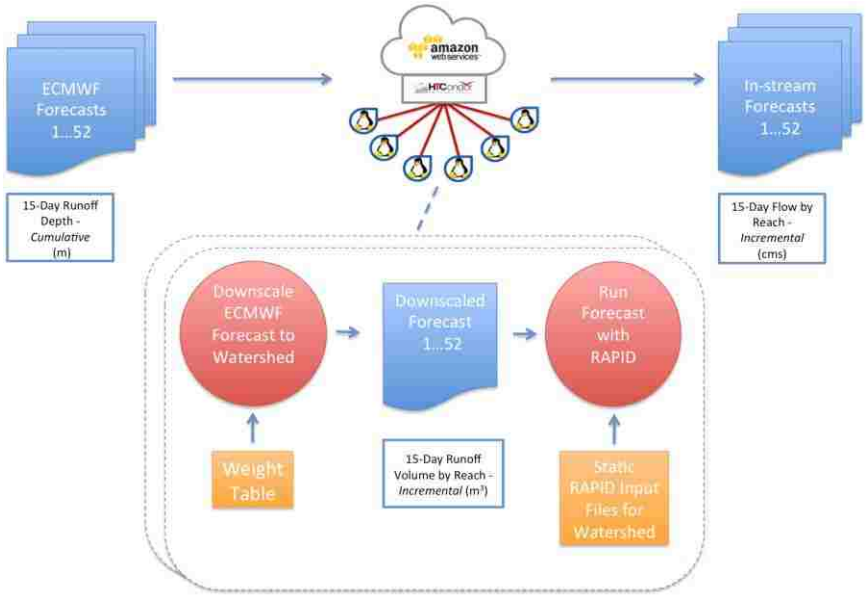

Figure 3-5 Amazon Web Services Computation Diagram

To ensure each node could perform the computations, I created an Amazon Elastic Block Store (Amazon EBS) backed image of a node with Python, the netCDF4-python library and RAPID installed (AWS 2015). Then, I initialized three computing nodes with the same image so that they would also have the Python netCDF4 library and RAPID installed. A diagram illustrating the method I used for creating and deploying computing nodes with EBS images in the AWS is shown in Figure 3-6.

Figure 3-6 Diagram of Creating and Using EBS Image

The master node was a c8x.large, another node was a c3.4xlarge, and another node was a c3.xlarge (http://aws.amazon.com/ec2/instance-types). The number of processors totaled to be 52. To simplify the distribution of jobs between nodes, I used Condorpy in my script on the master node to call up jobs in HTCondor. Using this method, I was able to create a job for each ensemble that needed to be downscaled such that an ideal computation time was achieved.

3.2 Software Design – Tethys Platform Visualization Web App

To make the output easily accessible to anyone, I created an online application that can be viewed from anywhere that you have access to the Internet. To create this application, I need a framework that provides a Geographical Information System (GIS) graphical interface; is able to connect to other data servers; has the tools to analyze the data; and can present the results in a meaningful way. The framework I chose is a Django-based platform called Tethys Platform because it provides all of these features. Within this platform, I created an app to visualize the output from the downscaling process.

3.2.1 GIS Visualization

The GIS visualization technique for displaying the data is a coupling of the GeoServer 2.6.2 GIS data publishing and OpenLayers 3.2.1 GIS mapping systems. When the OpenLayers map loads, it adds the layers to the map by querying the GeoServer.

There are two main methods used in the app for loading the data into the OpenLayers map. The first method uses the Open Geospatial Consortium (OGC) Web Feature Service (WFS) system and loads the vector data into the map from the GeoJSON format. The WFS method is used for loading in the stream segments into the map so that when the user clicks on the stream segment, the attributes of the stream segments are readily available. The next method is the OGC Web Map Service (WMS) where it retrieves a georeferenced map image from the server. The WMS loading method is applied to the subbasins and the gauge layers as they are only there for the user's reference and the data loads faster in this format.

3.2.2 Computation and Data Store System

The Tethys Platform web app was set up such that the computations and the actual web service are separate entities. The computations to be performed on another server or cloud-based computing system twice daily as the ECMWF runoff prediction datasets become available. After the computations are finished, they are then uploaded to a data store server such as HydroShare or CKAN. In a separate process, the Tethys Platform web app then downloads up to a week of predictions and displays it to the user. The basic configuration between the computing nodes (such as AWS), the data store server, and the Tethys Platform web app is shown in Figure 3-7. In this configuration AWS manages the computations and sends the data to a data store. Then, the Tethys Platform web app downloads a subset of all the data from the data store to store a week's worth of forecasts.

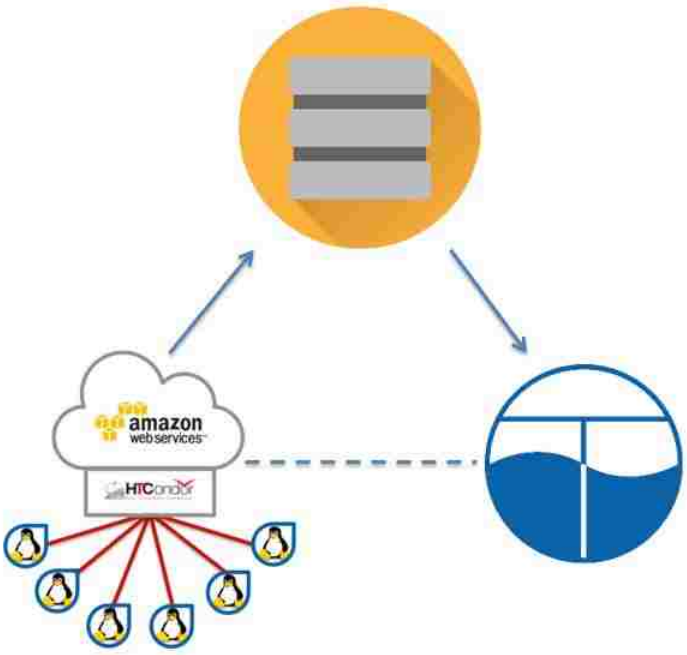

Figure 3-7 Tethys Computation and Data Storage System Diagram

3.2.3 Forecast Analysis and Visualization

In order to display the results, I used the netCDF4-python and NumPy Python libraries to extract and analyze the stream forecast datasets. In this process I used the netCDF4-python library to loop through each ensemble and extract the stream forecast time series for the reach by id. Then, using the NumPy library, I analyzed the output by taking the mean, standard deviation, maximum, and minimum of all of the forecasted time series. I then returned the analyzed results as well as the high resolution forecast in the JSON format. When a reach is selected, I take the JSON output from the server and add it to a Highcharts chart (www.highcharts.com).

3.3 Experimental Use Case – Comparison with USGS, AHPS and GloFAS

To test the accuracy of the geoprocessing tools and routing with RAPID, two watersheds were chosen to test the system. One of the watersheds has a lower density stream network that closely matches the GloFAS river network. The other watershed has a high-density stream network to test the accuracy of increased density. When setting up the watersheds, I calculated the Muskingum parameters using the default multiplying factors $\lambda_k = 0.35$ and $\lambda_x = 3$ in the 'Calculate the Muskingum Parameters' tool. The output from RAPID will be compared to USGS gage station data and AHPS predictions, where available, and to the output from GloFAS. Due to the fact that the watersheds in the GloFAS system were designed for larger watersheds, the watersheds used for comparison will be larger than 2,000 sq km.

3.3.1 Magdalena Watershed

One example watershed I set up with a low-density stream network is located in Colombia, along the Magdalena River with an outlet near the city of El Banco. The stream network used has 29 drainage lines and a total area of 139,844 sq km and was set up using the 15 arc-second HydroSHEDS flow direction and flow accumulation rasters. The Magdalena watershed with the ECMWF low-resolution grid is shown in Figure 3-8.

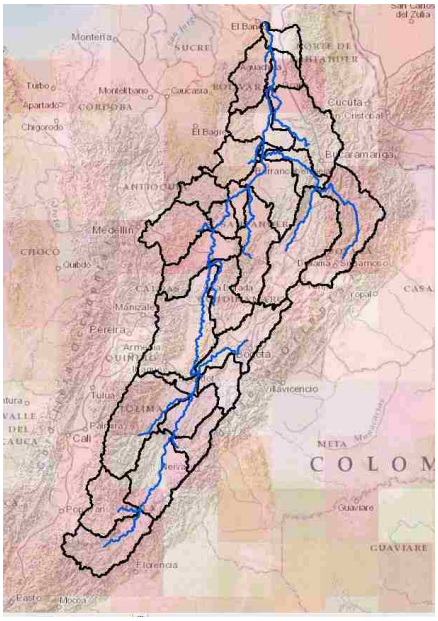

Figure 3-8 Colombia Watershed with ECMWF Runoff Forecast Grid

The GloFAS watershed used for comparison is shown in Figure 3-9.

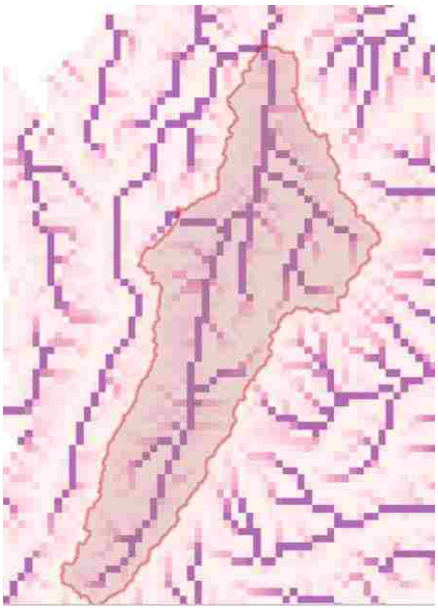

Figure 3-9 Magdalena Watershed from GloFAS Drainage Grid

3.3.2 HUC-4 Region 1209 Modified Watershed

An example of a medium sized watershed with a high-density stream network that I set up was a modified version of HUC-4 Region 1209 in the United States. The watershed included the 120800, 120901, 120902, and 120903 HUC-6 regions and was created from the USGS NHDPlus dataset. This watershed network has 11,212 drainage lines with a total approximate area of 110,008 sq km. The HUC-4 Region 1209 watershed with the ECMWF low-resolution grid is shown in Figure 3-10.

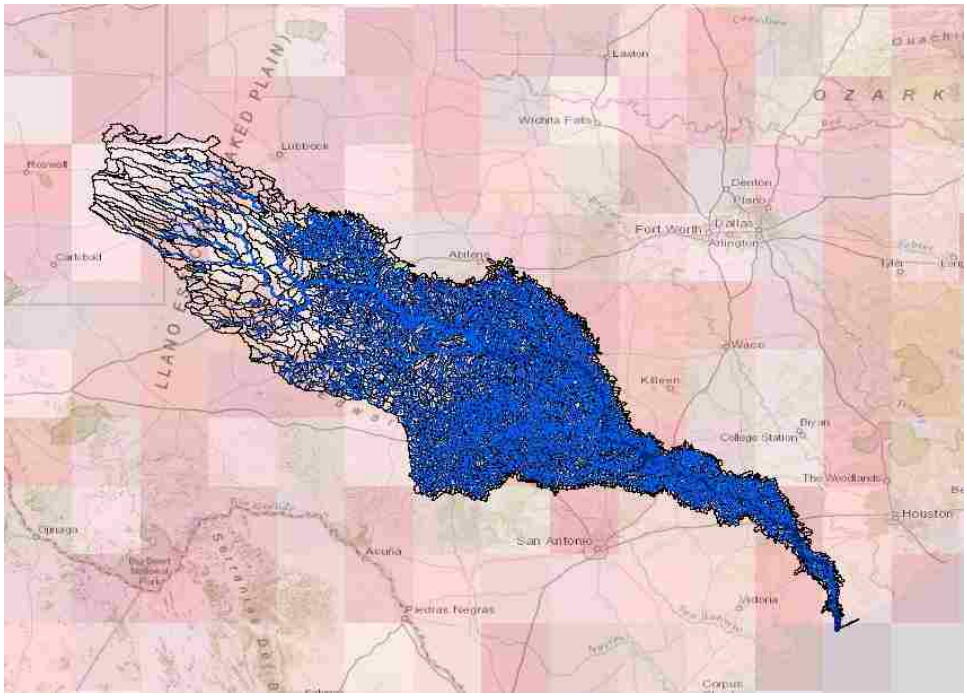

Figure 3-10 Modified HUC-4 Region 1209 with ECMWF Runoff Forecast Grid

The GloFAS watershed used for comparison is shown in Figure 3-11.

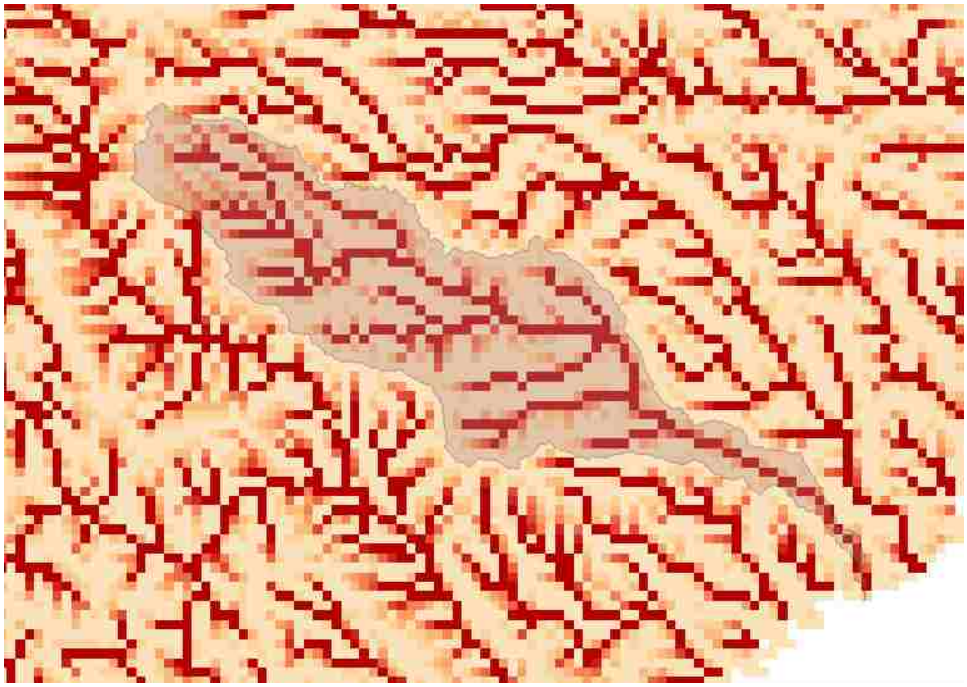

Figure 3-11 Modified HUC-4 Region 1209 on GloFAS Drainage Grid

3.4 Experimental Use Case – Automation of Computations

To determine how much of an improvement took place with the computational methods, I ran tests using a variety of watershed sizes and stream network densities. When setting up the watersheds, I calculated the Muskingum parameters using the default multiplying factors $\lambda_k = 0.35$ and $\lambda_x = 3$ in the 'Calculate the Muskingum Parameters' tool. Both watersheds from Section 3.4 will be included in these tests. Additional test watersheds will also be described in this section. Timing tests will be performed using a linear processing method, a multiprocessing method, and a method using the Amazon Web Service computing cloud.

3.4.1 Dominican Republic Watershed

An example of a small watershed with a low-density stream network that I set up is in the Dominican Republic including the Yaque Del Sur River. This watershed network has 31 drainage lines with a total area of approximately 5,204 sq km and was set up using the 15 arc-second HydroSHEDS flow direction and flow accumulation rasters. The Dominican Republic watershed with the ECMWF low-resolution grid is shown in Figure 3-12.

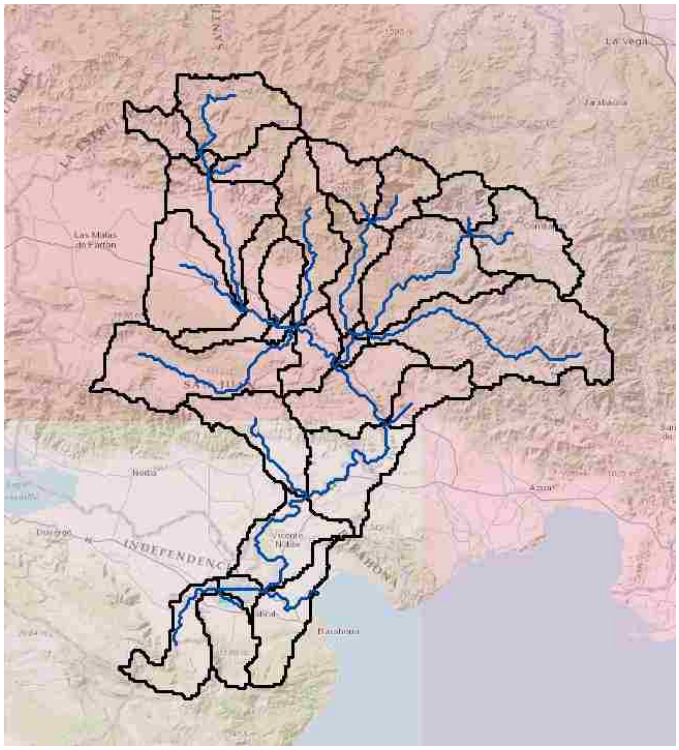

Figure 3-12 Dominican Republic Watershed with ECMWF Runoff Forecast Grid

3.4.2 Hobble Creek Watershed

The Hobble Creek watershed is an example of a small watershed with a high-density stream network that I set up and is located near Provo, Utah, USA. This watershed network has 182

drainage lines with a total area of approximately 323 sq km and was created from the NHDPlus dataset. The Hobble Creek watershed with the ECMWF low-resolution grid is shown in Figure 3-13.

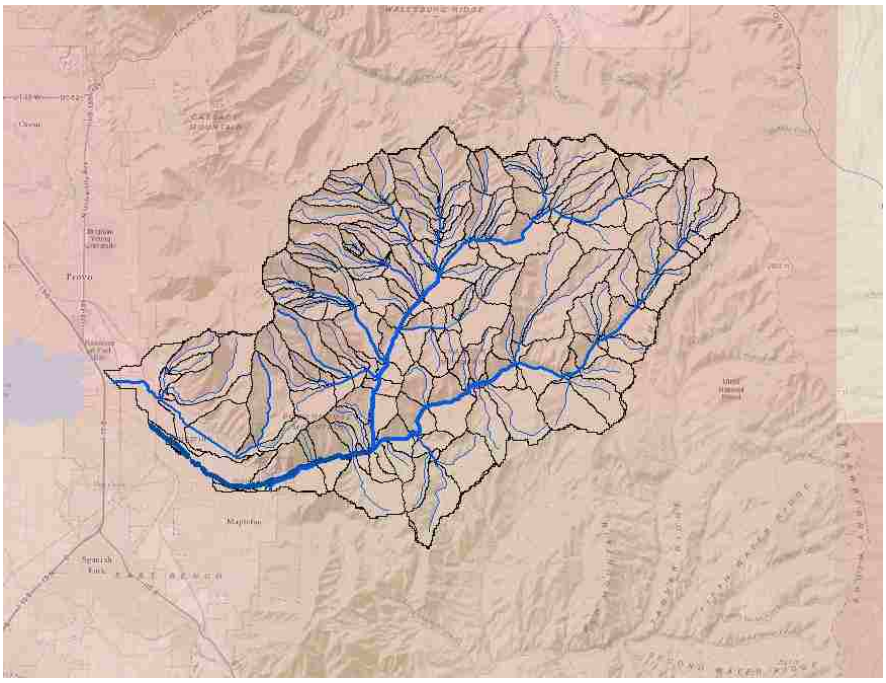

Figure 3-13 Hobble Creek Watershed with ECMWF Runoff Forecast Grid

3.4.3 HUC-2 Region 12 Watershed

An example of a large watershed with a high-density stream network that I set up is the Texas Gulf Region also identified as the HUC-2 Region 12 watershed. This watershed network has 67,313 drainage lines with a total area of approximately 464,493 sq km and was created from the NHDPlus dataset. The HUC-2 Region 12 watershed with the ECMWF low-resolution grid is shown in Figure 3-14.

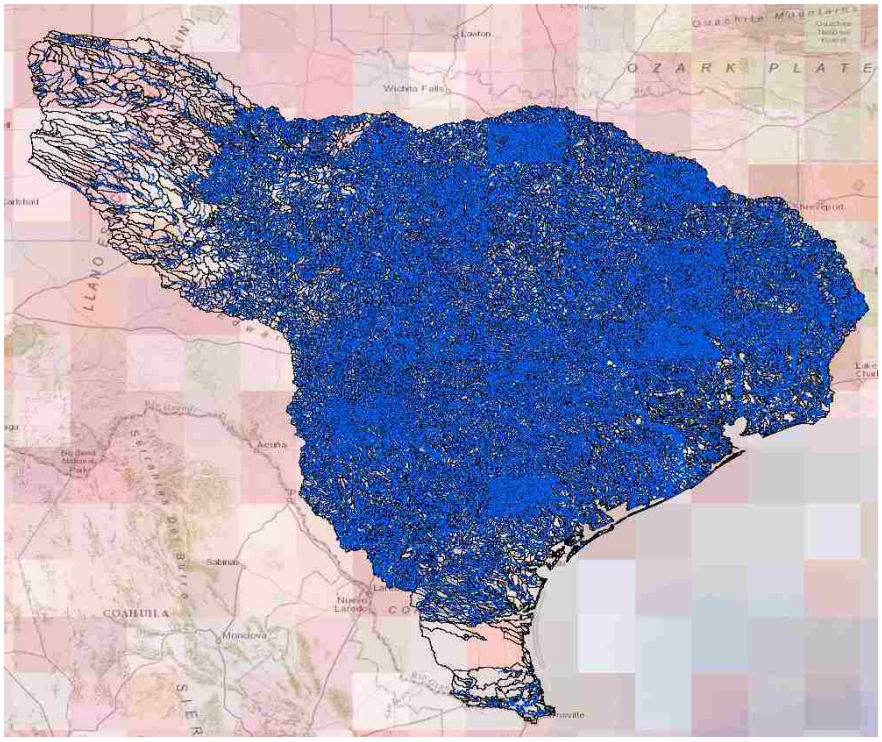

Figure 3-14 HUC-2 Region 12 Watershed with ECMWF Runoff Forecast Grid

3.4.4 HUC-2 Region 3 West Watershed

Another example of a large watershed with a high-density stream network that I set up is in the South Atlantic Region and is the HUC-2 Region 3 west watershed. This watershed network has 136,083 drainage lines with a total area of approximately 242,239 sq km and was created from the NHDPlus dataset. The HUC-2 Region 3 west watershed with the ECMWF low-resolution grid is shown in Figure 3-15.

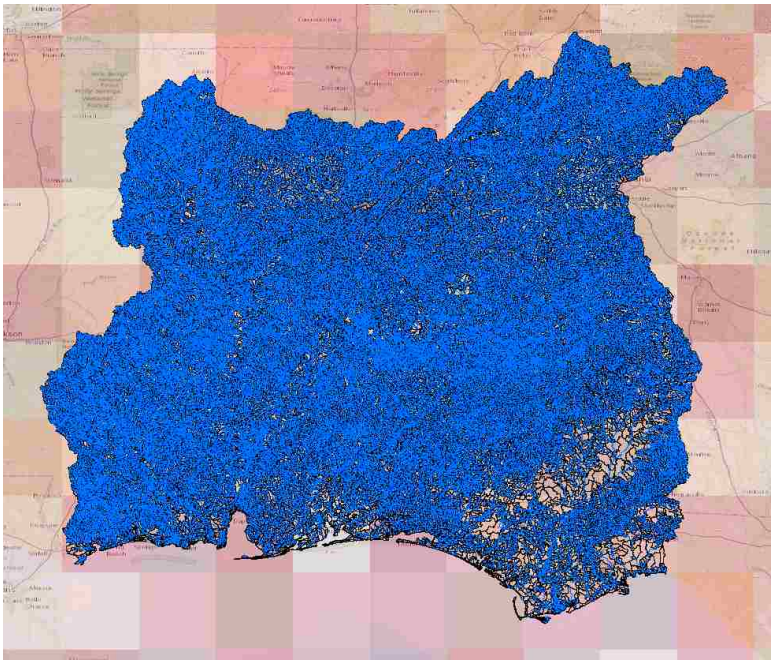

Figure 3-15 HUC-2 Region 3 West Watershed with ECMWF Runoff Forecast Grid

3.5 Experimental Use Case – Tethys Platform Visualization Web App

To test the GIS visualization capabilities, the speed of information being passed to the user, and the user friendly interface in the Tethys Platform Visualization Web App, the HUC-2 Region 12 watershed described in Section 3.5 will be used because it is large and has a high-density of reaches. To test the amount of storage required for the streamflow prediction data, the watersheds described in Section 3.3 and Section 3.4 will be used.

4 RESULTS

4.1 Software Implementation - Automation of Computations

I successfully implemented a method for automating the downscaling the ECMWF runoff forecasts using Esri's RAPID tools and then routing the runoff using RAPID that can be used for reaches of high and low density for any watershed in the world. The method can be applied to a single Linux computer, or in a Linux cloud environment. The code for automating the process using multiprocessing on a single Linux computer is located at https://github.com/CI-WATER/erfp_data_process_ubuntu. The code for automating the computations in an AWS Linux environment is located at https://github.com/CI-WATER/erfp_data_process_ubuntu_aws.

4.2 Software Implementation - Tethys Platform Visualization Web App

The full name of the Tethys Platform Web App is the ECMWF-RAPID Flood Prediction Tool (ERFP Tool). I was able to create the ERFP Tool in Tethys that shows the reaches on a map and allows the user to interactively select a reach and see a statistical hydrograph of the ensemble flow predictions of the reach. The app can be downloaded from https://github.com/CI-WATER/tethysapp-erfp_tool.

The app is capable of displaying information for multiple watersheds around the world as shown in the example view of the app in Figure 4-1.

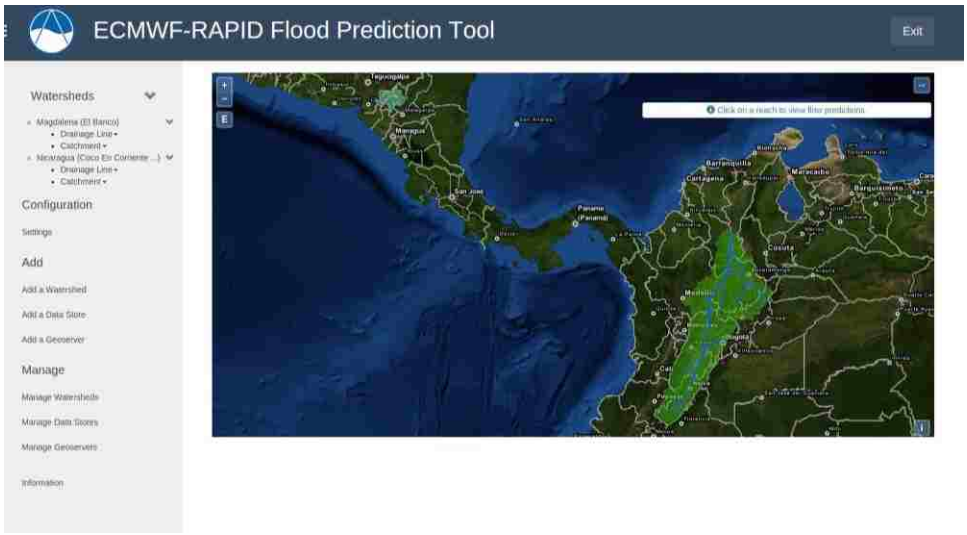

Figure 4-1 Overview Image of ERFP-Tool App

As shown in Figure 4-1, the tool provides a web GIS graphical interface for the user to view drainage basins (catchments) as well as reaches (drainage lines) for a watershed. When the user selects a stream segment, a graph pops up with a statistical summary of the output hydrographs from all 52 ensembles as shown in Figure 4-2.

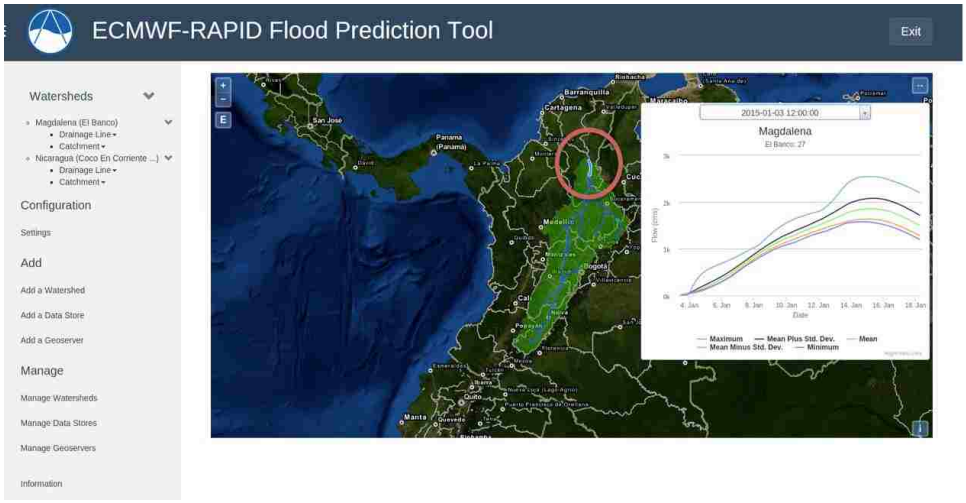

Figure 4-2 Selecting a Reach in the ERFP-Tool App

Once a reach is selected, the user is able to select past forecasts and view the statistical hydrograph for the selected reach as shown in Figure 4-3.

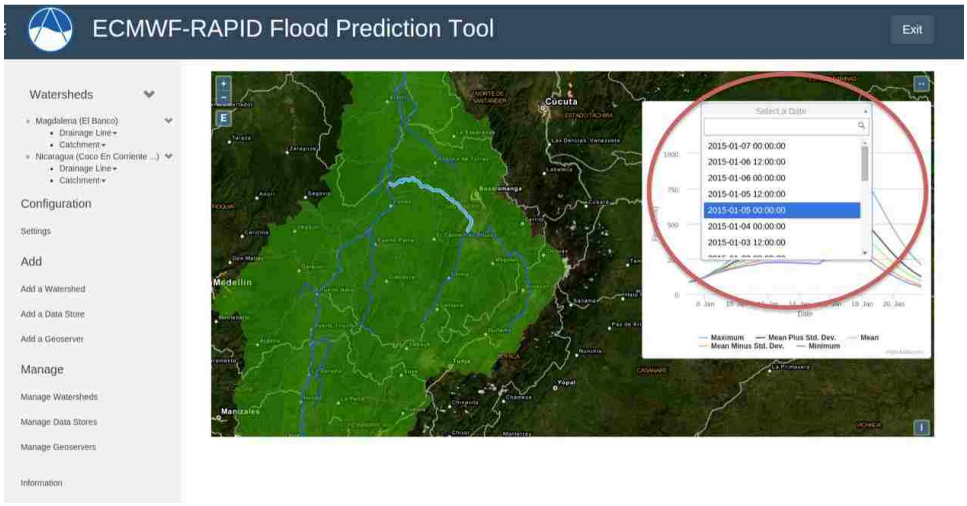

Figure 4-3 Selecting a Past Forecast in the ERFP-Tool App

Because of how the ERFP Tool App was designed, it is very flexible in how it can be used. The set up for the tool allows for computations to be performed on different computers or AWS clusters and then submitted to the same database or different databases. Additionally, several instances of the ERFP Tool App can be created on different servers that can access one or more data store servers or GeoServers and download data.

With the flexibility the ERFP Tool App, an institution can set up a central computing location that handles everything by itself. Another configuration could include setting up the computations for watersheds in local regions and then store the data on a central server.

The ERFP Tool was presented in both the NFIE Technical Review Webinar as well as in the NFIE Student Presentations Webinar. In the presentations, the ERFP Tool App was presented as

an example for what other students may work towards as well as a resource that they may use to further their projects (CUAHSI 2015).

4.3 Use Case Results – Comparison with GloFAS, USGS, and AHPS

In this Section, I will discuss the results from downscaling the ECMWF global runoff prediction datasets and routing with RAPID to the data from GloFAS, USGS, and AHPS where available.

4.3.1 Magdalena Watershed

I only compared the Magdalena watershed with GloFAS as that was the only data available. The comparison with GloFAS in the Magdalena watershed highlights the need for an initialization method. In the plot comparing ERFP with GloFAS, the plots as derived from the RAPID simulation as defined in the legend begin with 'ERFP' while the plots from the GloFAS model begins with 'GloFAS' as shown in Figure 4-4.

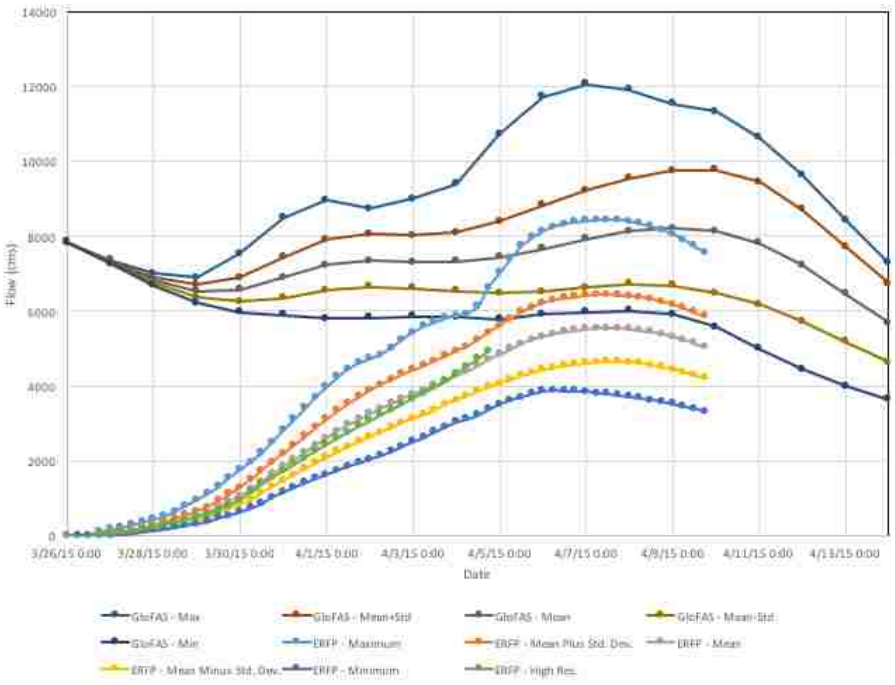

Figure 4-4 Comparison of Outlet of Magdalena Watershed with GloFAS

In Figure 4-4 the difference between the initialization points is shown to be approximately 8,000 cms. The ERFP model is limited as the simulation always begins with zero flow in the streams. Because flows begin at zero and a "negative" flow is not possible, the ERFP model does not catch the dip in the flow where the flows continue to decline but at a decreasing rate. If the model were initialized at a higher flow, the streamflow could have exhibited this decreasing rate of decline and captured the dip in the resulting hydrograph. Nevertheless, both of the models have similar rises and drops in flow at the same time periods.

4.3.2 HUC-4 Region 1209 Modified Watershed

For the HUC-4 Region 1209 modified watershed, I compared the ERFP results at a reach near the outlet (COMID 3766334) to the GloFAS results and USGS data (Station 08162500).

Additionally, at locations where the AHPS streamflow predictions were available I compared the AHPS predictions with ERFP predicted streamflow.

The comparison with GloFAS results at the outlet is shown in Figure 4-5 where the plot legend distinguishes the ERFP simulations with 'ERFP' and GloFAS simulations with 'GloFAS'.

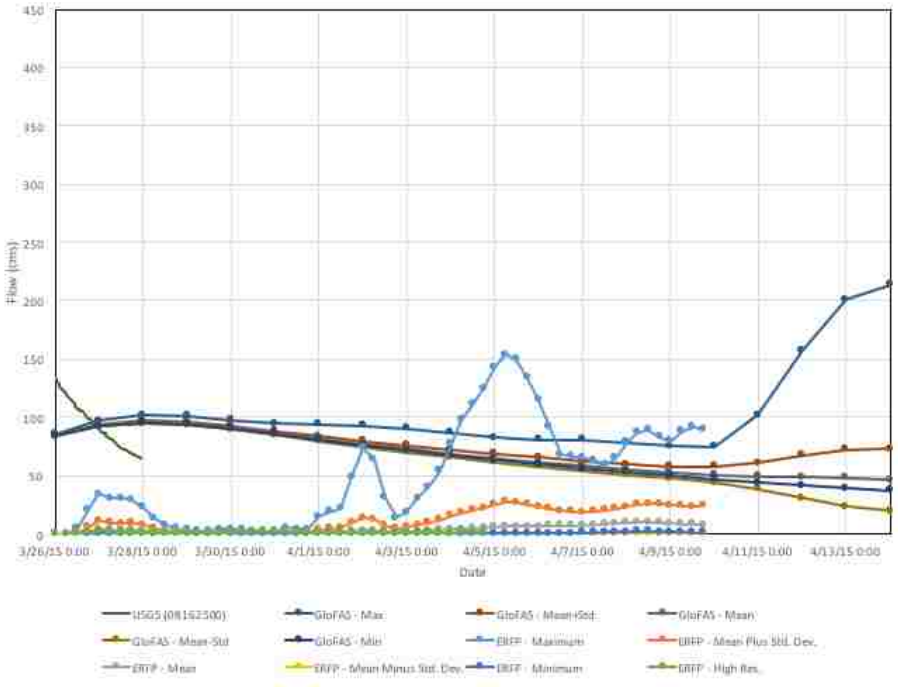

Figure 4-5 Comparison of Outlet of HUC-4 1209 Watershed with GloFAS

Again, it is apparent that the ERFP simulation lacks initialization as the flow begins at zero. The GloFAS simulation appears to initialize the flow close to what actually occurred, with the likely difference due to the fact that the gage station was not directly at the outlet. The ERFP model appears to show a higher variation in the flows and predicts more of a peak close to 4/5/2015 whereas the GloFAS model seems pretty constant with a steady decrease. The increase in variation in ERFP could be attributed to the increase in the density of reaches in the model.

From the limited comparisons made with USGS and AHPS, a gap has typically been noted between the ERFP forecasted streamflow and the USGS and AHPS streamflow at the beginning of the forecast. The gap arises from the fact that the flows are currently not being initialized in the RAPID simulation. As such, all flows from the forecast start at zero as shown in Figure 4-6 and Figure 4-7.

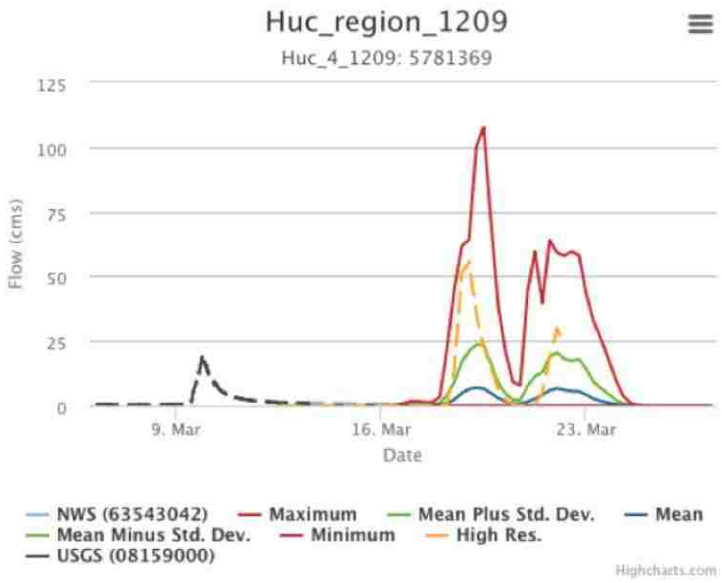

Figure 4-6 ECMWF Forecast to Gage Data at COMID 5781369

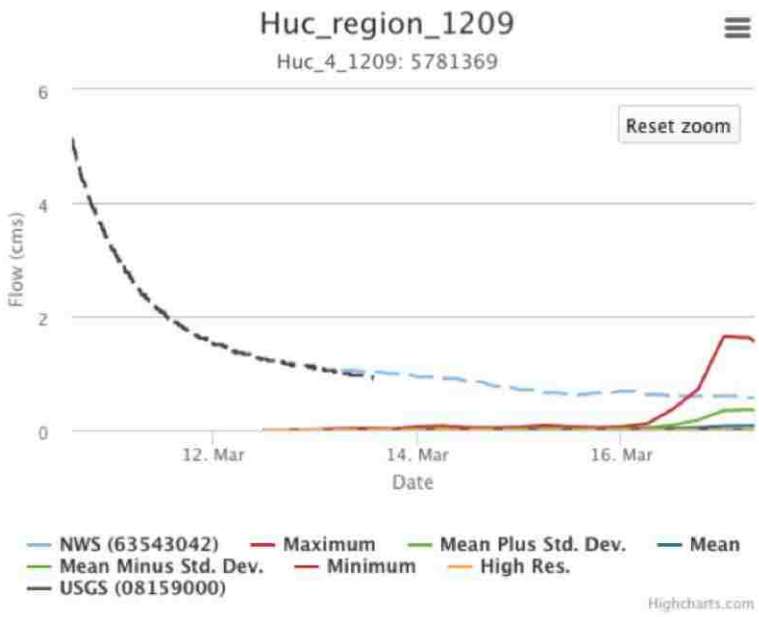

Figure 4-7 ECMWF Forecast to Gage Data at COMID 5781369 Zoomed In

For this reason, the results currently will not be able to accurately predict the streamflows that are going to happen. In spite of this the ERFP model is predicting events to occur with a relative magnitude, and when comparing past predictions from ECMWF to USGS gage data, some of the results have been within the band of uncertainty as seen in Figure 4-8 and Figure 4-9.

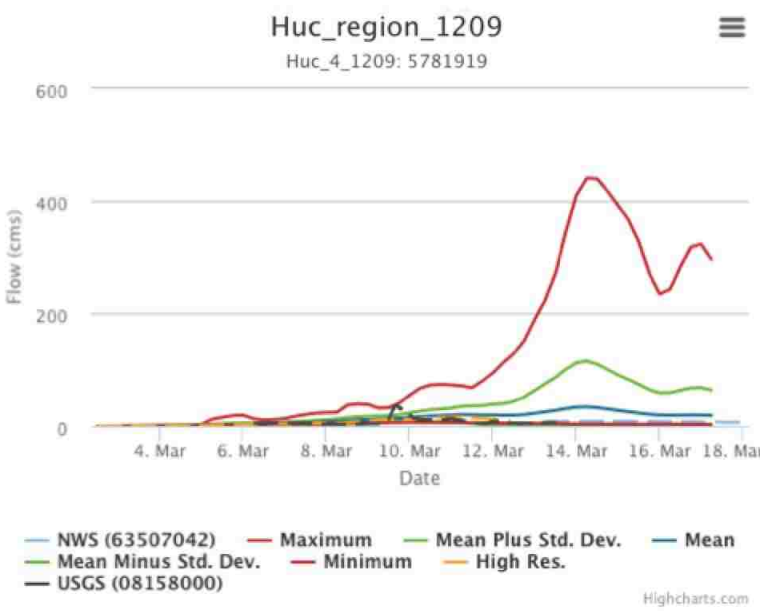

Figure 4-8 Comparing ECMWF Forecast to Gage Data at COMID 5781919

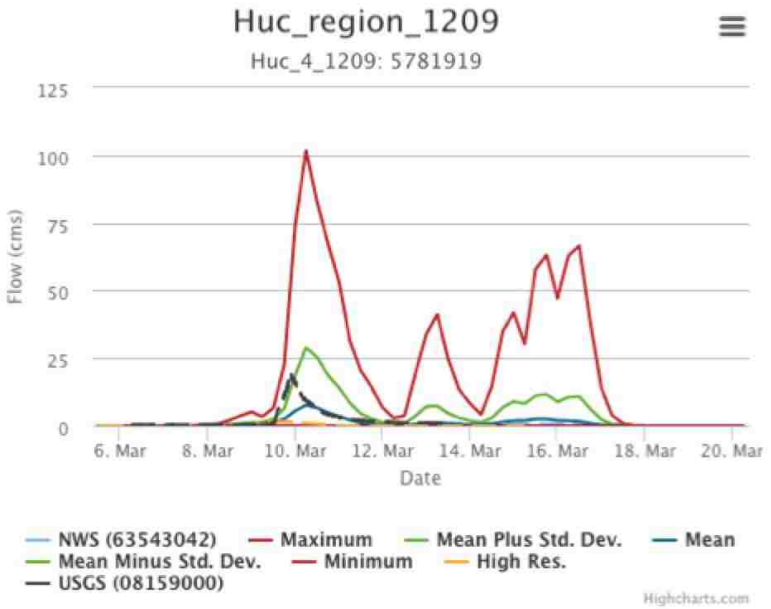

Figure 4-9 Comparing ECMWF Forecast to Gage Data at COMID 5781369

4.4 Use Case Results - Automation of Computations

I compared the computational performance of computing using the multiprocessing method, and the Amazon Web Services method and estimated the linear computational time results based off of the individual computation times. The results are shown in Table 4-1.

Table 4-1 Results of Computation Time Based on Area and the Number of Reaches

			Compute Time (seconds)		
Watershed Name	Area (sq km)	Number of Reaches	Linear	Multiprocessing	Amazon Web Services
HUC 2 Region 03	242,239	136,083	51,700	11,899	1,225
HUC-2 Region 12	464,493	66,373	12,346	2,745	304
HUC 4 Region 1209	110,008	11,212	238	110	28
Hobble Creek	324	182	215	107	28
Dominican Republic	5,204	31	215	106	29
Magdalena	139,844	29	214	106	28

In Figure 4-10 the computation time versus the number of reaches from Table 4-1 is shown with trend lines. The data matches the trend line quite well with R-squared values above 0.93.

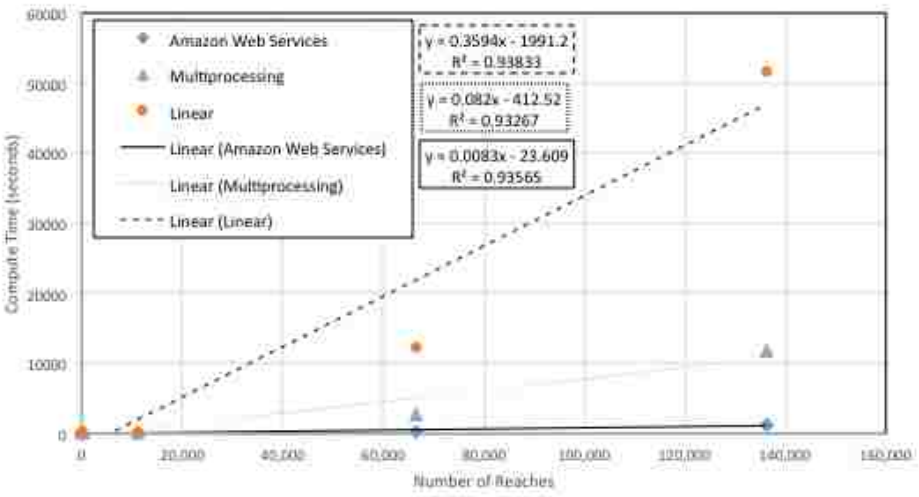

Figure 4-10 Computation Time versus Number of Reaches

In Figure 4-11 the computation time versus the watershed area from Table 4-1 is shown with trend lines. It appears that the data does not match the line of best fit when comparing computation time and watershed area as all of the R-squared values are below 0.2.

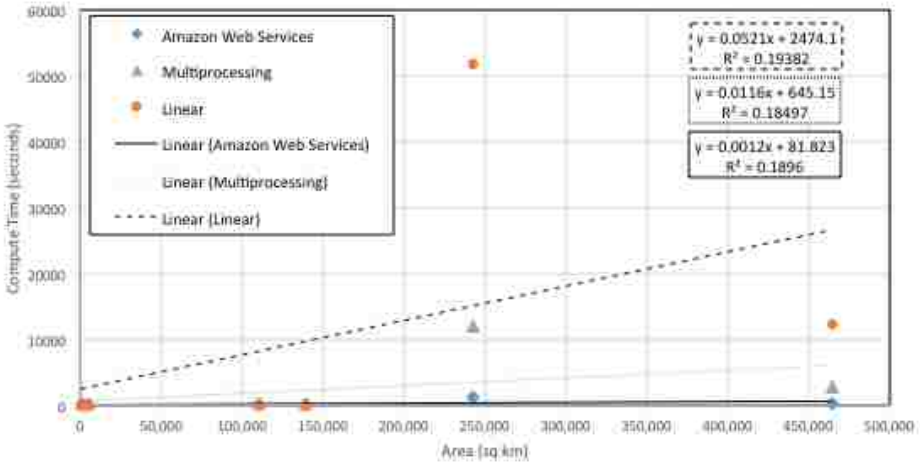

Figure 4-11 Computation Time versus Watershed Area

Using the number of reaches to compare, the computation time improvement when using the multiprocessing and AWS versus linear computations is more prominent as the number of reaches enters the tens of thousands. The time of computation for each method falls within the required 12-hour window between forecasts needed to prevent a computation lag. I examined the number of reaches to extrapolate the limits of the processes because the data had R-squared values closer to one. Using the equations in Figure 4-10, I extrapolated the maximum number of reaches in a watershed that can be downscaled within the 12-hour window. The results of the extrapolation are shown in Table 4-2.

Table 4-2 The Limits of Computational Methods Extrapolated

Number of Reaches	Compute Time (hours)		
	Linear	Multiprocessing	Amazon Web Services
125,741	12	2.7	0.3
531,860	53	12.0	1.2
5,207,664	519	118.5	12.0

As shown in Figure 4-2, the linear and multiprocessing methods work well for watersheds with low-density stream networks but cannot finish in time with larger, high-density stream networks. To be able to handle watersheds with more than 500,000 reaches, AWS is required. However, the Amazon Web Services computation method has its limit as well, which is apparently close to five million reaches using the applications developed in this research for the ERFP model. Nonetheless, this is all theoretical, and testing actual watersheds is required to determine the limit to the number of reaches that can be computed in a 12-hour period. Of course other optimizations to the processes, especially the RAPID computations, which require most of the time could also improve and increase the number of reaches possible.

4.5 Use Case Results - Tethys Platform Visualization Web App

The GIS visualization capabilities of the ERFP Tool app are able to handle displaying the HUC-2 Region 12 watershed by coupling GeoServer and OpenLayers. The app works best in a browser on a recent computer and currently does not have any recognized issues. The loading time is relatively fast, averaging around seven seconds to load. Displaying large high-density stream networks is currently limited to stream networks defined by the NHDPlus dataset as it has the stream order defined, which makes dynamic display of stream networks possible as you zoom in and out on the map.

The ERFP Tool App is able to display the watershed because the reaches are divided into three levels of density. When the map is zoomed out to the full watershed, only the main reaches are shown as seen in Figure 4-12.

Figure 4-12 Entire Watershed Zoom Level

When sufficiently zoomed in, the next level of stream density (order) shows the main reaches along with some of the medium sized reaches as seen in Figure 4-13.

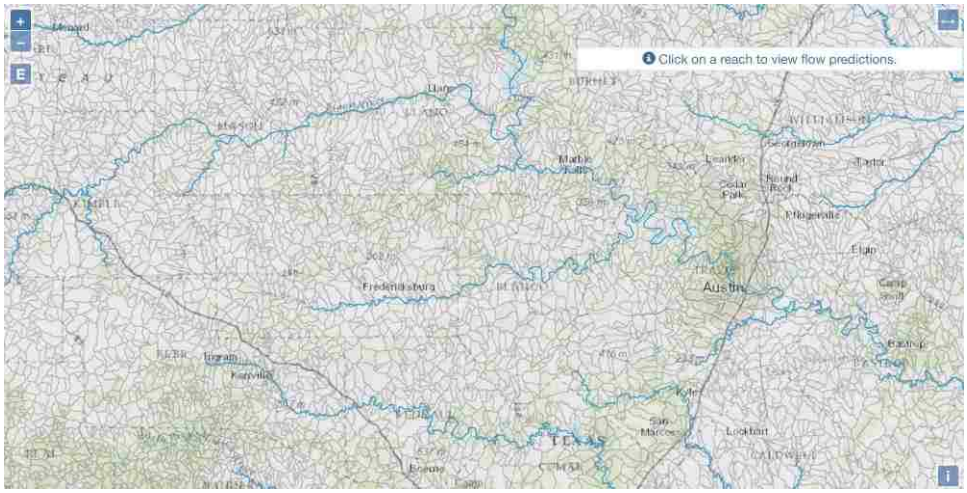

Figure 4-13 Medium Range Zoom Level

When the user is zoomed in closer to the catchments, all of the reaches in the users view are shown as seen in Figure 4-14.

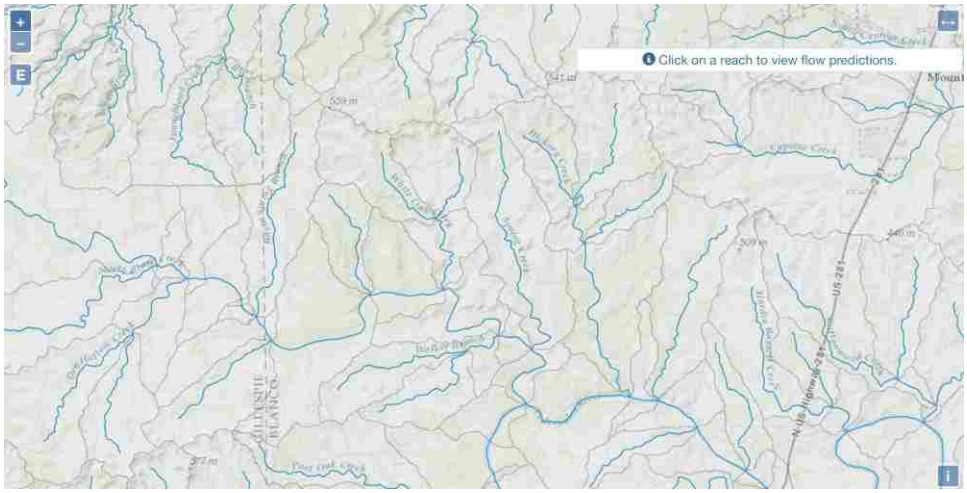

Figure 4-14 Closest Zoom Level

However, with mobile devices or older computers, the visualization may still be slow and use up available memory on the device.

The speed of information being passed to the user is instantaneous in most cases. However, there are times when the information can take up to five seconds to load, such as if there are too many requests being sent to the server. During the research no testing on simultaneous access limits was performed.

Owing to its map-based interface, it is easy to use with minimal explanation. All the end user needs to do is select a watershed and then select a reach of interest. However, currently it is difficult to select a reach and several users have indicated that it can take around three clicks to get the reach they are interested in.

One problem with the setup of storing all of the prediction results on a data store server is the memory required. The results of the memory required for each watershed per forecast is shown in Table 4-3. Note that the total reported is the total after the files are compresses in the tar.gz format.

Table 4-3 File Sizes for Watershed Streamflow Forecasts

		Total Size (MB)		
Name	Number of Reaches	Low Resolution	High Resolution	Total (tar.gz)
HUC 2 Region 03	136,083	33.2	22.3	1400
HUC 2 Region 12	66,373	16.4	11	721
HUC 4 Region 1209	11,212	2.7	1.8	122.5
Hobble Creek	182	0.0445	0.03	2
Dominican Republic	31	0.0072	0.0049	0.3283
Magdalena	29	0.0072	0.0049	0.3282

As shown in Table 4-3, the file size corresponds almost exactly with the number of reaches divided by one thousand. This is confirmed when performing a linear regression analysis of the number of reaches and the file size shown in Figure 4-15.

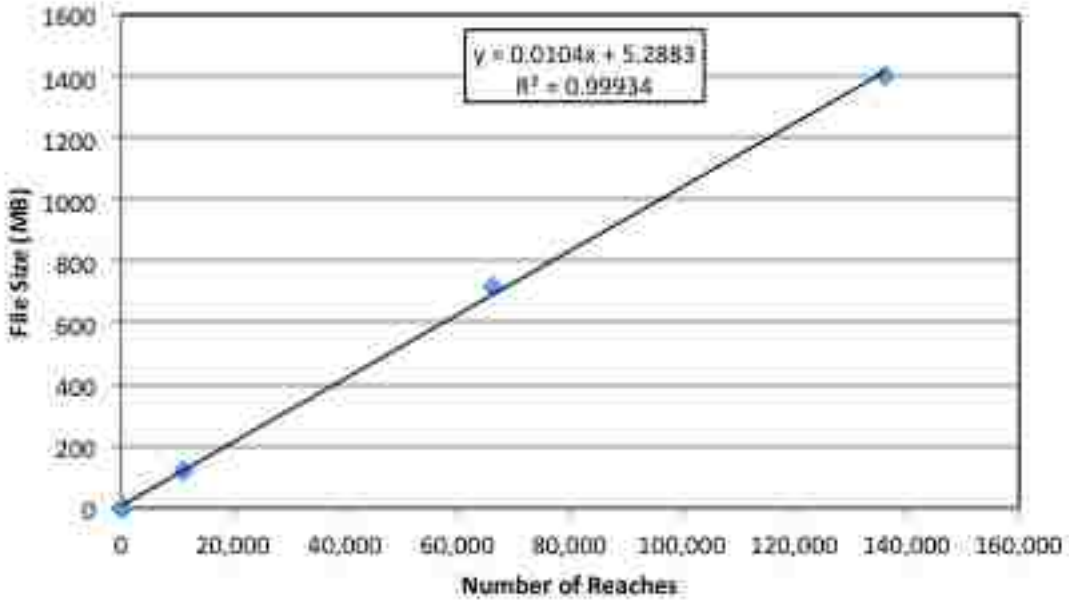

Figure 4-15 File Sizes versus Watershed Streamflow Forecasts Plot

The correlation between the number of reaches and the file size is very good with an R-squared value of almost one. Accordingly, storing forecasts for the entire 2.6 million reaches of the United States NHDPlus would require approximately 27 GB per forecast. Since the forecasts come twice daily, approximately 20 TB of storage per year would be required to store all forecasts.

5 CONCLUSION

The creation of a flood warning system that can predict floods days and even weeks in advance at a high spatial resolution is possible. By downscaling runoff forecasts generated by the ECMWF using Esri's RAPID toolbox and routing the runoff using the RAPID program, I was able to produce high-density stream forecasts using the NHDPlus datasets in the HUC-2 Region 12 and the HUC-2 Region 03 West watersheds. This method could be used to create a forecast density at the level of the 2.67 million reaches nationwide from the NHDPlus datasets, which is a huge increase from the 3,600 AHPS forecasts currently generated.

With the computational resources of the AWS, setting up this system for the entire United States is possible. With an estimated limit close to five million reaches, each of the 21 HUC-2 regions could be set up to run the hydrologic computations centrally. Alternatively, each HUC-2 region could be set locally and could be harvested by the NWS into a central data store. Using either method, a national high spatial resolution hydrologic forecasting network could be produced.

The generation of high resolution stream forecasts is not limited to the United States. With global geospatial datasets, such as HydroSHEDS, higher resolution stream networks can be generated almost anywhere in the world. With these drainage networks the same methods can be applied to generate forecasts in places such as the Yaque Del Sur River in the Dominican Republic

or the Magdalena River in Colombia as shown in this research. As such, the spatial resolution of the GloFAS forecasts could be expanded to millions or billions of locations globally.

With the technology to display the results in a web environment, Tethys Platform, I developed an interface to display the high-density streamflow forecasts to decision makers. This tool gives decision makers analyzed information from NetCDF datasets containing stream forecast information for hundreds of thousands of reaches, it shows them where the stream segment is located, and it presents the a statistical summary of the potential streamflow up to fifteen days in advance all with the click of a button. As such, it simplifies the data access and interpretation for decision makers substantially.

Using the method and tools developed by this research, the creation of additional tools and improvements to the current system are both possible and needed. As shown in the results, the forecast flow begins at zero currently. As such, the most pressing of the improvements is a method for initializing streamflow in the rivers. The next improvements needed are items such as calibrating models, and a method for including reservoirs in the modeling process. In addition to improvements, new tools can and should be created based off of the high resolution stream forecasts. These new tools can involve ideas such as predictive flood index maps based off of the data from the in stream forecasts. With the improvement of this method and the creation of new tools, decision makers will be better equipped to plan and prepare for water related natural disasters.

REFERENCES

Alfieri, L., Burek, P., Dutra, E., Krzeminski, B., Muraro, D., Thielen, J., and Pappenberger, F. (2013). "GloFAS–global ensemble streamflow forecasting and flood early warning." *Hydrol. Earth Syst. Sci*, 17(3), 1161-1175.

AWS (2015). "Creating an Amazon EBS-Backed Linux AMI." *awsdocumentation*, <http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/creating-an-ami-ebs.html>. (Feb. 11, 2015).

Balsamo, G., Beljaars, A., Scipal, K., Viterbo, P., van den Hurk, B., Hirschi, M., and Betts, A. K. (2009). "A revised hydrology for the ECMWF model: Verification from field site to terrestrial water storage and impact in the Integrated Forecast System." *Journal of hydrometeorology*, 10(3), 623-643.

Blankenhorn, D. (2005). "Google map API transforms the Web." *ZDNet.com Open Source Blog*, <http://www.zdnet.com/article/google-map-api-transforms-the-web>. (Mar. 4, 2015).

Boulos, M. N. (2005). "Web GIS in practice III: creating a simple interactive map of England's strategic Health Authorities using Google Maps API, Google Earth KML, and MSN Virtual Earth Map Control." *International Journal of Health Geographics*, 4(1), 22.

Buyya, R., Vecchiola, C., and Selvi, S. T. (2013). *Mastering cloud computing: foundations and applications programming*, Newnes.

Corporation, H. S. (2011). "Weaving the National Hydrologic Geospatial Fabric." *NHDPlus Version 2*. (April 6, 2015).

CUAHSI (2015). "National Flood Interoperability Experiment." *The Consortium of Universities for the Advancement of Hydrologic Science, Inc. (CUAHSI)*, <https://www.cuahsi.org/NFIE>. (Feb. 11, 2015).

Cunge, J. A. (1969). "On the subject of a flood propagation computation method (Musklngum method)." *Journal of Hydraulic Research*, 7(2), 205-230.

David, C. H., Maidment, D. R., Niu, G.-Y., Yang, Z.-L., Habets, F., and Eijkhout, V. (2011). "River network routing on the NHDPlus dataset." *Journal of Hydrometeorology*, 12(5), 913-934.

De Rosnay, P., Drusch, M., Balsamo, G., Isaksen, L., and Albergel, C. (2011). "Extended Kalman Filter soil moisture analysis in the IFS." *ECMWF Spring Newsletter*, 127, 12-16.

ECMWF (2011). "IFS Documentation – Cy37r2 Operational implementation 18 May 2011 - Part II: Data Assimilation." *European Centre for Medium Range Weather Forecasts (ECMWF)*.

ECMWF (2013). "IFS Documentation – Cy40r1 Operational implementation 22 November 2013 - Part IV: Physical processes." *European Centre for Medium Range Weather Forecasts (ECMWF)*.

EM-DAT, C. (2010). "The OFDA/CRED international disaster database." *Université catholique*.

Humphrey, M., Beekwilder, N., Goodall, J. L., and Ercan, M. B. "Calibration of watershed models using cloud computing." *Proc., E-Science (e-Science), 2012 IEEE 8th International Conference on*, IEEE, 1-8.

Jankowski, P. (1995). "Integrating geographical information systems and multiple criteria decision-making methods." *International journal of geographical information systems*, 9(3), 251-273.

Jones, N., Nelson, J., Swain, N., Christensen, S., Tarboton, D., and Dash, P. (2014). "Tethys: A Software Framework for Web-Based Modeling and Decision Support Applications." *International Environmental Modelling and Software Society (iEMSs) 7th Intl. Congress on Env. Modelling and Software*.

Lehner, B., Verdin, K., and Jarvis, A. (2008). "New global hydrography derived from spaceborne elevation data." *EOS, Transactions American Geophysical Union*, 89(10), 93-94.

NOAA (2015). "Advanced Hydrologic Prediction Service." *National Oceanic and Atmospheric Administration's (NOAA) National Weather Service (NWS)*, <http://water.weather.gov/ahps/about/about.php>. (Feb. 17, 2015).

Pappenberger, F., Cloke, H. L., Parker, D. J., Wetterhall, F., Richardson, D. S., and Thielen, J. (2015). "The monetary benefit of early flood warnings in Europe." *Environmental Science & Policy*

Pappenberger, F., Stephens, E., Thielen, J., Salamon, P., Demeritt, D., Andel, S. J., Wetterhall, F., and Alfieri, L. (2013). "Visualizing probabilistic flood forecast information: expert preferences and perceptions of best practice in uncertainty communication." *Hydrological Processes*, 27(1), 132-146.

Shim, J. P., Warkentin, M., Courtney, J. F., Power, D. J., Sharda, R., and Carlsson, C. (2002). "Past, present, and future of decision support technology." *Decision support systems*, 33(2), 111-126.

Singh, V. P., and McCann, R. C. (1980). "Some notes on Muskingum method of flood routing." *Journal of Hydrology*, 48(3), 343-361.

Swain, N. R., Latu, K., Christensen, S. D., Jones, N. L., Nelson, E. J., Ames, D. P., and Williams, G. P. (2015). "A review of open source software solutions for developing water resources web applications." *Environmental Modelling & Software*, 67, 108-117.

Tarboton, D., Idaszak, R., Horsburgh, J., Heard, J., Ames, D., Goodall, J., Band, L., Merwade, V., Couch, A., and Arrigo, J. "HydroShare: Advancing Collaboration through Hydrologic Data and Model Sharing." *Proc., International Environmental Modelling and Software Society (iEMSs) 7th International Congress on Environmental Modelling and Software San Diego, California, USA, DP Ames, N. Quinn(Eds.)* http://www.iemss.org/society/index.php/iemss-2014-proceedings.

USGS (2014). "Hydrography." *U. S. Geological Survey*, <http://nhd.usgs.gov/index.html>.