A Cloud-Based Flood Warning System For Forecasting Impacts to Transportation Infrastructure Systems

Jon Goodall
Associate Professor, Civil and Environmental Engineering
Associate Director, Link Lab
Research Motivation

• Predicting **street scale flooding**, especially for low relief terrains, is a challenge.

• For these regions, we need **high resolution datasets** and **2-D hydrodynamic models**.

• Flood predictions are needed **in near real-time** in a form that **aids decision makers**.

Chloe Riffle, 7, watches as she is surrounded by water on Sunday, Sept. 4, 2016 in the Ocean View section of Norfolk, Virginia.

Need for an Automated, Accurate Flood Prediction System
Objectives

- **Objective 1:** Build a *flood warning system* based on modern cyberinfrastructure using an *automated workflow* for obtaining the *real-time rainfall forecast data*.

- **Objective 2:** *Execute* the model using *cloud computing resources* to identify the flooded roadways and bridges location in *short-time duration* for warning and emergency management purposes.

- **Objective 3:** Generate *online inundation map* with the location of the *flooded roadways* and *bridges* with the ability to send automated warning email messages.
Study Areas

**VDOT’s Hampton Roads District**
(Rural Area)

**Norfolk Area**
(Urban Area)
Modeling Work - VDOT’s Hampton Roads District

• Hampton Roads district, Virginia which is 11x10^3 km^2 of combined watersheds

• This Study area includes about 500 VDOT bridges and culverts

• 11 subwatersheds that surround the study area are incorporated as boundary conditions
Regional River Severe Storm Model (R²S²)

R²S² is a diverse, multi-function digital platform that offers various applications to VDOT’s Districts and Residencies, regional Emergency Services, and Environmental Agencies.
Enhancing the Model

Model main Component

Streamflow line Shapfile Generated from the DEM

Road Shapfile files that include the Road Crossing and Elevation with streamflow lines

DEM Rater with 10m Resolution

Manning’s Coefficient Shapefile Based on the Landuse

TUFLOW Model
High resolution LiDAR data with resolution from 0.76m to 1.52m is available in most part of the study area.
DEM Comparison

Cross Section

1m DEM

10m DEM
Streamline

Old Version Flowline

New Version Flowline
Infiltration Implementation

**Soil Type**

**Imperviousness**

Very small percentage of area has an imperviousness ratio greater than zero.
Prepare Soil Moisture from SMAP to TUFLOW

HDF Data Format

Request GeoTiff Format Data Through Api

SAMP Soil Moisture (m3/m3)
There are 10 USGS station located in the study. However, we believe more stations are necessary to better calibrating and verifying the model due to the complex and flat topographic.
Observation Gridded Rainfall Data Preparation

• Three sources we explored to obtain the gridded rainfall data:
  • TRMM data
    • A code have been developed to pull and process the TRMM data for any given shapefile in the USA.
  • HRRR archived data
    • A code have been developed to pull and process the archived HRRR data from the University of Utah repository for any given shapefile in the USA.
  • NEXRAD data
    • However this is the best gridded rainfall data available but sometime the rainfall data was not recorded for critical storm events period. Also it is time consuming to order, download, and preprocess the data manual using NOAA Weather and Climate Tool (WCT).
• All the codes are available through the project Phase II GitHub repository.
### Rainfall Forecast Data Automation and Preparation for the Flood Warning System

#### High-Resolution Rapid Refresh (HRRR)

- NOAA/NCEP operational weather prediction system
- Forecasts hourly surface total precipitation at a **spatial resolution of 3-km**
- Comprised of a numerical forecast model and initialized by an analysis/assimilation system
- HRRR is run every hour of the day and forecasts out 18 hours on an hour time-step.
- We automated HRRR data access using Python and OPeNDAP

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Data Provider</th>
<th>Relevant Data Product</th>
<th>Resolution Spatial (km)</th>
<th>Forecast (hrs)</th>
<th>Model Cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>HRRR</td>
<td>NCEP</td>
<td>Surface total precipitation</td>
<td>3</td>
<td>18</td>
<td>24/day</td>
</tr>
<tr>
<td>RAP</td>
<td>NCEP</td>
<td>Surface total precipitation</td>
<td>13</td>
<td>18</td>
<td>24/day</td>
</tr>
<tr>
<td>NDFD</td>
<td>NWS</td>
<td>Quantitative precipitation forecast</td>
<td>5</td>
<td>72</td>
<td>8/day</td>
</tr>
<tr>
<td>NAM</td>
<td>NCEP</td>
<td>Surface total precipitation</td>
<td>12</td>
<td>36</td>
<td>4/day</td>
</tr>
</tbody>
</table>

http://ruc.noaa.gov/hrrr/displayImage.cgi?image=hrrrcrefimage&width=859&model=hrrr&title=HRRR
The National Water Model simulation will be used as boundary input for flood forecasting system.

**Historical Runs**

The HEC-HMS model was built to simulate the outlet flow rate of each sub-watershed to be applied as boundary input of the 2D model.

**Flood Forecasting Runs**

The National Water Model simulation will be used as boundary input for flood forecasting system.

http://water.noaa.gov/about/nwm
Hurricane Nicole, 2016
Hurricane Nicole, 2016
Preliminary Calibration for Hurricane Nicole, 2016
Preliminary Calibration for Hurricane Nicole, 2016
Modeling Work – Norfolk Area

A 2D hydrodynamic model needs to be developed to predict inundated roads in Norfolk Region. The ODU storm surge model will provide the tide level boundary condition.
Storm Surge and Urban Flood models
Coupling Workflow

- **Delft3D**
  - Hydrodynamic Storm Surge Model using Delft3D

- **NOAA Observation Tidal Gauges**

- **Generated Tidal Data at the Boundary Conditions**

- **TUFLOW**
  - Urban Flood Model using TUFLOW-HPC

- **NOAA Rainfall Forecast Data**

- **Flooding Zones**

- **Risk Assessment**

Storm surge model interaction with the urban flood model
Model main Component

Storm Surge
Precipitation

Forces

Impervious Ratio
Soil Type
Manning’s Coefficient Shapefile

Flowline Shapfile
Road Network
1m DEM and bathymetry

TUFLOW Model
Buildings

Topography Mesh
Integrating High Resolution DEM

1m Lidar

VB Bathymetry

Integrated DEM

NOAA Bathymetry

(https://www.ngdc.noaa.gov/mgg/coastal/crm.html)
Imperviousness and Building Footprint

Surface Imperviousness

Building Footprint for Hague Community

Building Footprint
Hurricane Irene, 2011

The storm surge simulation we have so far is from 26-Aug-2011 to 28-Aug-2011. We generated a hotstart file at the start time of the storm surge model by running the TUFLOW model with NOAA tidal level observation. Then, the hotstart file is feed into the model to let it run from wet condition.
Hurricane Irene, 2011

Flood Reports
- Flood Reports

Maximum Water Depth (m)

Value
- High: 16.668
- Low: 0

Maximum Water Depth (m)
Hurricane Irene, 2011
Modeling the Hague Community

A. Using the current model domain.
B. Create a small domain for the Hague community from the current model domain.
C. Using the small domain along with the storm drainage system for the Hague community.
Hurricane Irene, 2011
Modeling the Hague Community
Cloud-based Flood Warning System

https://vfis.uvahydroinformatics.org
Architecture of the Flood Warning System Through GCP

- **GPU Windows OS Model components**
  - **Start Model Run Manually**
  - **Power User**
  - **Send Alerts Via Email**
  - **Access Current and Archived Flooded Locations Information**
  - **Trigger/Visualization**
  - **Linux OS**
  - **Access DB**
  - **Access Current Flooded Locations Information**
  - **Send Alerts Via Email**

- **Storage Bucket**
  - **Request and Retrieve Rainfall Data**
  - **Store Input/Output**

- **NOAA NOMADS**
  - **Monitor Rainfall Data**
  - **Access Current Flooded Locations Information**

- **Database Instance**
  - **Access DB**
  - **Start If Rainfall Recorded**

- **Regular User**
  - **Access Current and Archived Flooded Locations Information**

- **University of Virginia**
Model Running Speed on the GCP GPU Instance

- Google Cloud has completely customizable instances with up to 8 GPUs, either 8 GPUs of NVIDIA Tesla K80 (each $0.45/hr) or 4 GPUs of NVIDIA P100 (each $1.46/hr) (the maximum number of GPU on the current TUFLOW license is 8 GPU).

- Currently, we are using 4 GPUs of NVIDIA Tesla K80, 8 Gb of memory, and 50 Gb of storage as the basic configuration with cost of $2.078/hr ($0.278/hr without any GPUs). P100 is roughly 2x K80 in performance.

<table>
<thead>
<tr>
<th>Grid Cell Resolution (m)</th>
<th>Running Time (hrs) for 19 hours modeling</th>
<th>CPUs</th>
<th>RAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>0.17</td>
<td>4</td>
<td>8 GB</td>
</tr>
<tr>
<td>40</td>
<td>0.27</td>
<td>4</td>
<td>8 GB</td>
</tr>
<tr>
<td>30</td>
<td>0.6</td>
<td>4</td>
<td>10 GB (+$0.009/hr)</td>
</tr>
<tr>
<td>20</td>
<td>1.70</td>
<td>4</td>
<td>20 GB (+$0.06/hr)</td>
</tr>
<tr>
<td>10</td>
<td>8.75</td>
<td>10 (+$0.2/hr)</td>
<td>65 GB (+$0.26/hr)</td>
</tr>
</tbody>
</table>
Model Running Speed on the GCP GPU Instance

• Focusing on 30m resolution grid cell size, the following chart shows the different in the running time using K80 vs P100 GPUs.

• using 4 P100 GPUs, the 20m resolution grid cell size runs in 0.91 hrs
Website SQL Database

• The database interaction is done by using SQLAlchemy

```python
class User(db.Model):
    __tablename__ = 'users'
    id = db.Column(db.Integer, primary_key=True)
    email = db.Column(db.String(120), unique=True)
    password_hash = db.Column(db.String(128))
    first_name = db.Column(db.String(120))
    middle_name = db.Column(db.String(120))
    last_name = db.Column(db.String(120))
    organization = db.Column(db.String(120))
    title = db.Column(db.String(120))
    role = db.Column(db.String(120))
    country = db.Column(db.String(120))
    state_province = db.Column(db.String(120))
    phone_number = db.Column(db.String(120))
    website = db.Column(db.String(120))

    def hash_password(self, password):
        self.password_hash = pwd_context.encrypt(password)

    def verify_password(self, password):
        return pwd_context.verify(password, self.password_hash)

    def as_dict(self):
        return {c.name: getattr(self, c.name) for c in self.__table__.columns}
```

```python
class Constructions(db.Model):
    __tablename__ = 'constructions'
    fedid = db.Column(db.Integer, primary_key=True)
    road_name = db.Column(db.String)
    xcord = db.Column(db.Float)
    ycord = db.Column(db.Float)
    stream = db.Column(db.String)
    roadlevel = db.Column(db.Float)
    forecast = db.relationship(Forecast)
```

```python
class Forecast(db.Model):
    __tablename__ = 'forecasts'
    id = db.Column(db.Integer, primary_key=True)
    start_date = db.Column(db.String)
    maxw1 = db.Column(db.Float)
    floodedby = db.Column(db.String)
    end_date = db.Column(db.String)
    construction_fee_id = db.Column(db.Integer, db.ForeignKey('constructions.fedid'))
    run_date_time = db.Column(db.String)
```

Visualization Website

https://vfis.uvahydroinformatics.org
Thank You

Jonathan L. Goodall
goodall@virginia.edu