IBM

SETI INSTITUTE

Mining Massive Datasets from the Allen Telescope Array

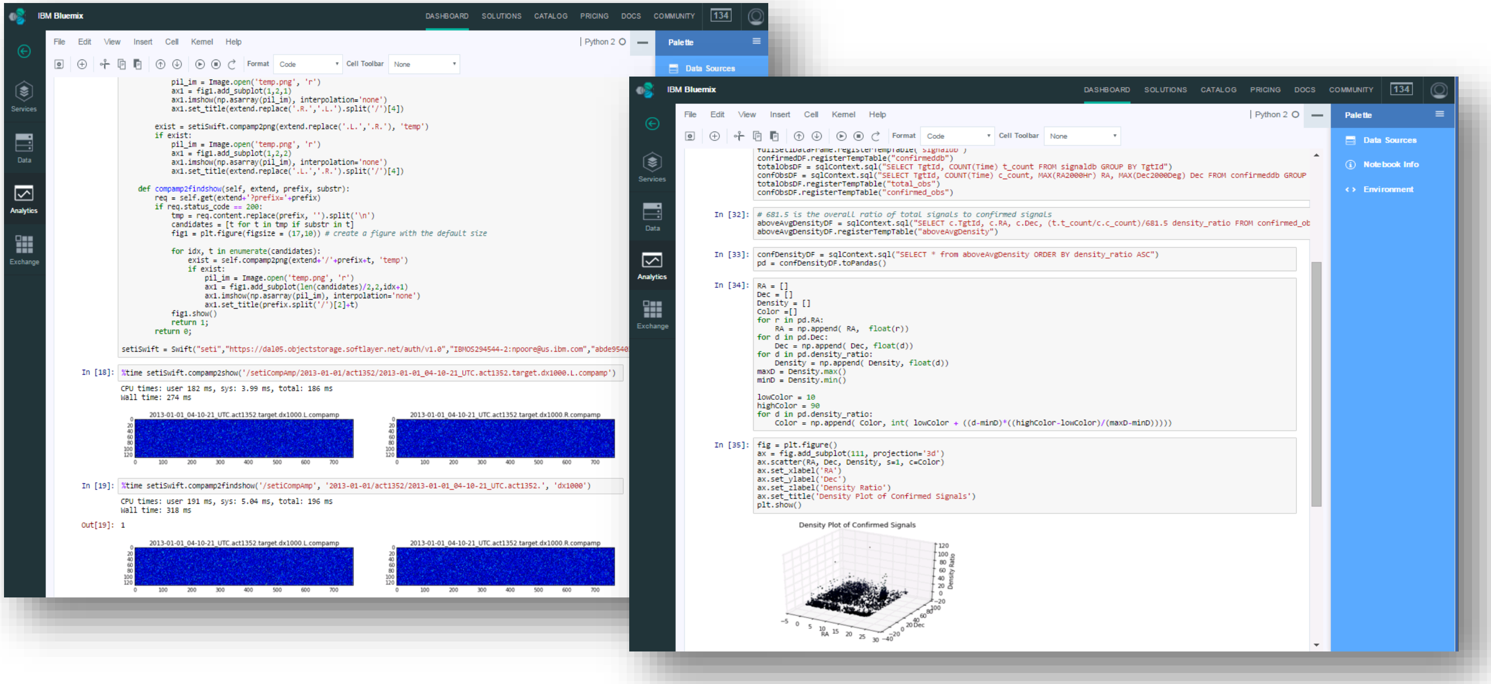
Proposal for Stanford CS341 students to use IBM Cloud services to analyze massive datasets from the Allen Telescope Array at Hat Creek Radio Observatory

IBM Spark@SETI Backgrounder

- The SETI Institute operates the Allen Telescope Array (ATA) to observe star systems for radio signals which may provide evidence of extraterrestrial intelligence.
- IBM collaborating with the SETI Institute to use IBM Apache Spark services to analyze radio signal data from the ATA.
- Astronomers and data scientists from around the world are using the Spark@SETI environment to analyze millions of signal events collected from the ATA over 10 years.
- Apache Spark with Jupyter Notebooks are proving to be highly effective in enabling experimental approaches to the analysis of the SETI Institute’s archive of signal data.
- IBM is offering to make the entire Spark@SETI environment available to Stanford CS341 students during the Spring 2016 quarter.

Spark@SETI for CS341

- ❑ Benefits of this proposal include:
  - ❑ Simple access to the Apache Spark platform to help expand the scope of CS341 projects beyond MapReduce computational methologies.
  - ❑ Use of the IBM Spark service at no charge, accessible on the IBM Cloud using a standard web browser.
  - ❑ Unlimited access to the ATA data archives, stored on the IBM cloud.



```
            pil_im = Image.open('temp.png', 'r')
            ax1 = fig1.add_subplot(1,2,1)
            ax1.imshow(np.asarray(pil_im), interpolation='none')
            ax1.set_title(extend.replace('.R.','.L.').split('/')[4])

        exist = setiSwift.compamp2png(extend.replace('.L.','.R.'), 'temp')
        if exist:
            pil_im = Image.open('temp.png', 'r')
            ax1 = fig1.add_subplot(1,2,2)
            ax1.imshow(np.asarray(pil_im), interpolation='none')
            ax1.set_title(extend.replace('.L.','.R.').split('/')[4])

    def compamp2findshow(self, extend, prefix, substr):
        req = self.get(extend+'?prefix='+prefix)
        if req.status_code == 200:
            tmp = req.content.replace(prefix, '').split('\n')
            candidates = [t for t in tmp if substr in t]
            fig1 = plt.figure(figsize = (17,10)) # create a figure with the default size

            for idx, t in enumerate(candidates):
                exist = self.compamp2png(extend+'/'+prefix+t, 'temp')
                if exist:
                    pil_im = Image.open('temp.png', 'r')
                    ax1 = fig1.add_subplot(len(candidates)/2,2,idx+1)
                    ax1.imshow(np.asarray(pil_im), interpolation='none')
                    ax1.set_title(prefix.split('/')[2]+t)
            fig1.show()
            return 1;
        return 0;

setiSwift = Swift("seti","https://dal05.objectstorage.softlayer.net/auth/v1.0","IBMOS294544-2:npoore@us.ibm.com","abde9540

In [18]: %time setiSwift.compamp2show('/setiCompAmp/2013-01-01/act1352/2013-01-01_04-10-21_UTC.act1352.target.dx1000.L.compamp')

CPU times: user 182 ms, sys: 3.99 ms, total: 186 ms
Wall time: 274 ms

In [19]: %time setiSwift.compamp2findshow('/setiCompAmp', '2013-01-01/act1352/2013-01-01_04-10-21_UTC.act1352.', 'dx1000')

CPU times: user 191 ms, sys: 5.04 ms, total: 196 ms
Wall time: 318 ms

Out[19]: 1
```

```
fullSetiDataFrame.registerTempTable("signaldb")
confirmedDF.registerTempTable("confirmeddb")
totalObsDF = sqlContext.sql("SELECT TgtId, COUNT(Time) t_count FROM signaldb GROUP BY TgtId")
confObsDF = sqlContext.sql("SELECT TgtId, COUNT(Time) c_count, MAX(RA2000Hr) RA, MAX(Dec2000Deg) Dec FROM confirmeddb GROUP
totalObsDF.registerTempTable("total_obs")
confObsDF.registerTempTable("confirmed_obs")

In [32]: # 681.5 is the overall ratio of total signals to confirmed signals
aboveAvgDensityDF = sqlContext.sql("SELECT c.TgtId, c.RA, c.Dec, (t.t_count/c.c_count)/681.5 density_ratio FROM confirmed_ob
aboveAvgDensityDF.registerTempTable("aboveAvgDensity")

In [33]: confDensityDF = sqlContext.sql("SELECT * from aboveAvgDensity ORDER BY density_ratio ASC")
pd = confDensityDF.toPandas()

In [34]: RA = []
Dec = []
Density = []
Color =[]
for r in pd.RA:
    RA = np.append( RA,  float(r))
for d in pd.Dec:
    Dec = np.append( Dec, float(d))
for d in pd.density_ratio:
    Density = np.append( Density, float(d))
maxD = Density.max()
minD = Density.min()

lowColor = 10
highColor = 90
for d in pd.density_ratio:
    Color = np.append( Color, int( lowColor + ((d-minD)*((highColor-lowColor)/(maxD-minD)))))

In [35]: fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.scatter(RA, Dec, Density, s=1, c=Color)
ax.set_xlabel('RA')
ax.set_ylabel('Dec')
ax.set_zlabel('Density Ratio')
ax.set_title('Density Plot of Confirmed Signals')
plt.show()
```

CS341 Students and Spark@SETI

Students using Spark@SETI for their CS341 projects will receive:

- ❑ A free account on the IBM BlueMix preconfigured with Spark services and direct connections to the ATA data repository.
- ❑ A “kick start” portfolio of Jupyter Notebooks that provide all the foundational Python code to query the ATA databases and read the recorded binary signals for analysis.
- ❑ Access to a 200 million row relational database of ATA signal events
- ❑ Access to approximately 15 million binary “complex amplitude” files that store the raw signal data from the ATA at the moment that a signal was detected
- ❑ Support from the Spark@SETI team for issues relating to platform usage (e.g. dropped Python kernels)

SETI INSTITUTE

CS341 Project Goals

- ❑ CS341 projects using Spark@SETI should not restrict project goals strictly to identifying a signal of interest.
- ❑ Much of the ongoing work in the Spark@SETI initiative is focused on the inverse problem: that of improving signal classification methods to eliminate signals that are human radio frequency interference (RFI).
- ❑ CS341 students may produce novel and valuable analytic results, which will be added to the Spark@SETI repository of Spark notebooks for use by astronomers and data scientist from around the world.
- ❑ Example: Igor Nikitin at the Fraunhofer Institute for Algorithms and Scientific Computing used ATA data to demonstrate how to apply the Radon transform to greatly improve the signal-to-noise ratio of narrow-band signals, thereby permitting the detection and classification of many new signals.

CS341 Spark@SETI - Examples

Examples of potential CS341 student projects include:

❑ Supervised machine learning of spectrogram images to classify signals according to known categories (e.g. radar interference, narrow-band zero drift, aircraft RFI).

❑ Analysis of the 200 million signal event database for targets that show unusual consistency of signal features (e.g. consistent power) over long periods of time. Outliers could be reviewed by the radio telescope operations staff for potential re-observation.

❑ Extracting scalar features from signal recordings with PCA and/or MDA results that indicate these features will help to spread and segment signals in useful ways for further analysis by Spark@SETI scientists.