Roswell OnDemand
A One-Stop Shop for High Performance Computing

L. Shawn Matott, PhD
Senior Systems Analyst
IT Research Computing

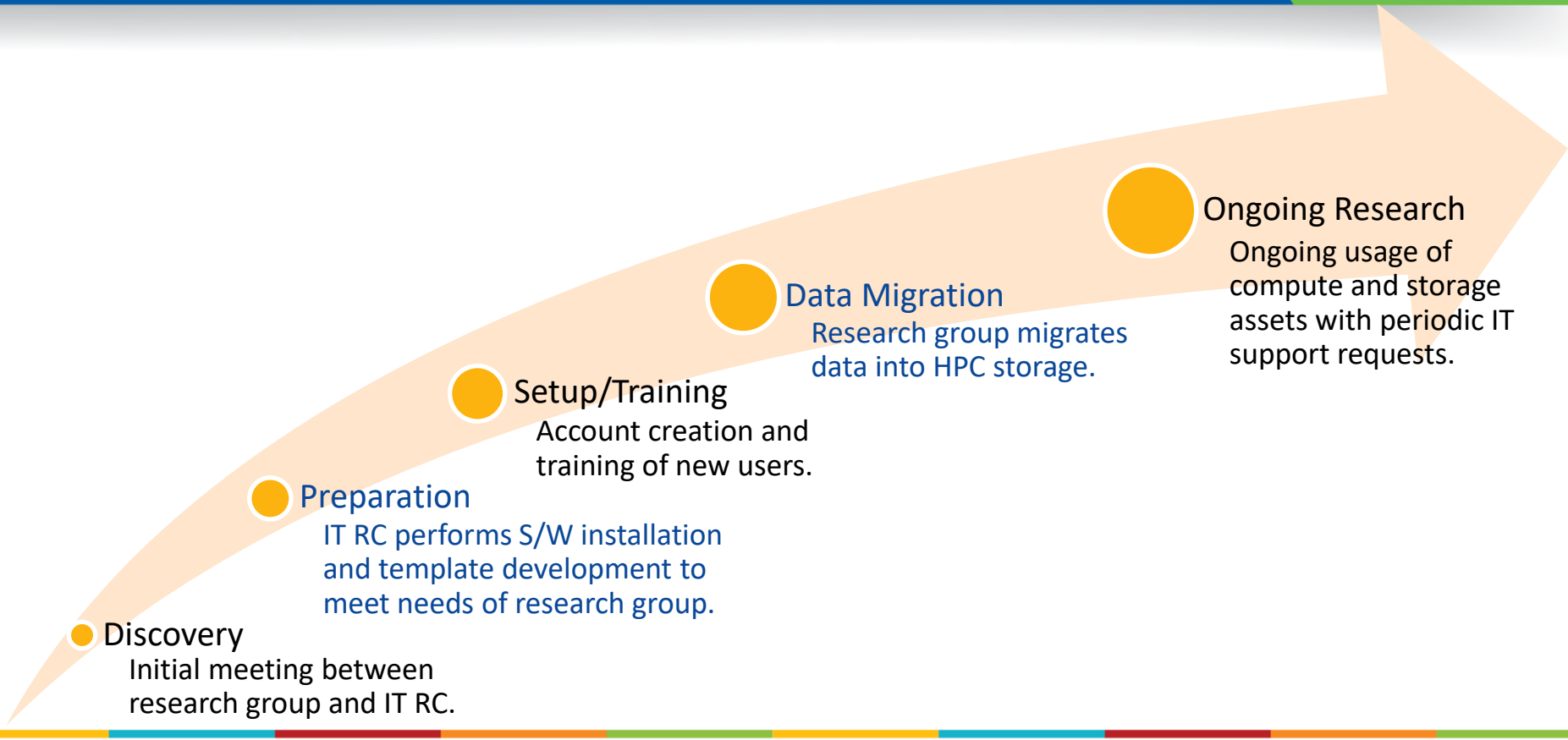
Computing Assets

- Roswell HPCC
  - 1,600 processors, 450 TB Lustre storage
  - Software modules include: Grace, EGSnrc, Keras, and TensorFlow, many bio-informatics packages

A look inside Roswell's HPC data center.

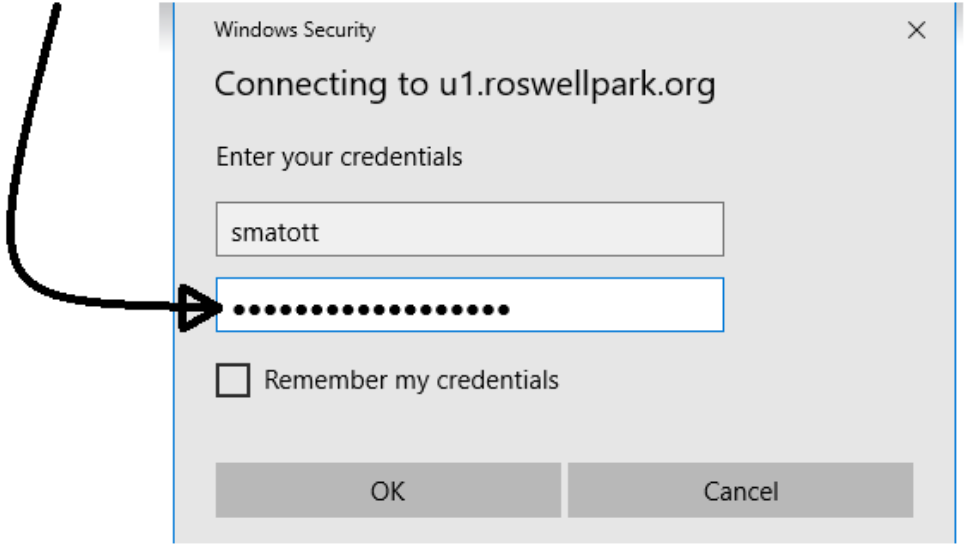
  - OnDemand web interface http://u1.roswellpark.org/pun/sys/dashboard/
  - e-mail ServiceDesk@RoswellPark.org to request access

A typical HPC Engagement



OnDemand Login

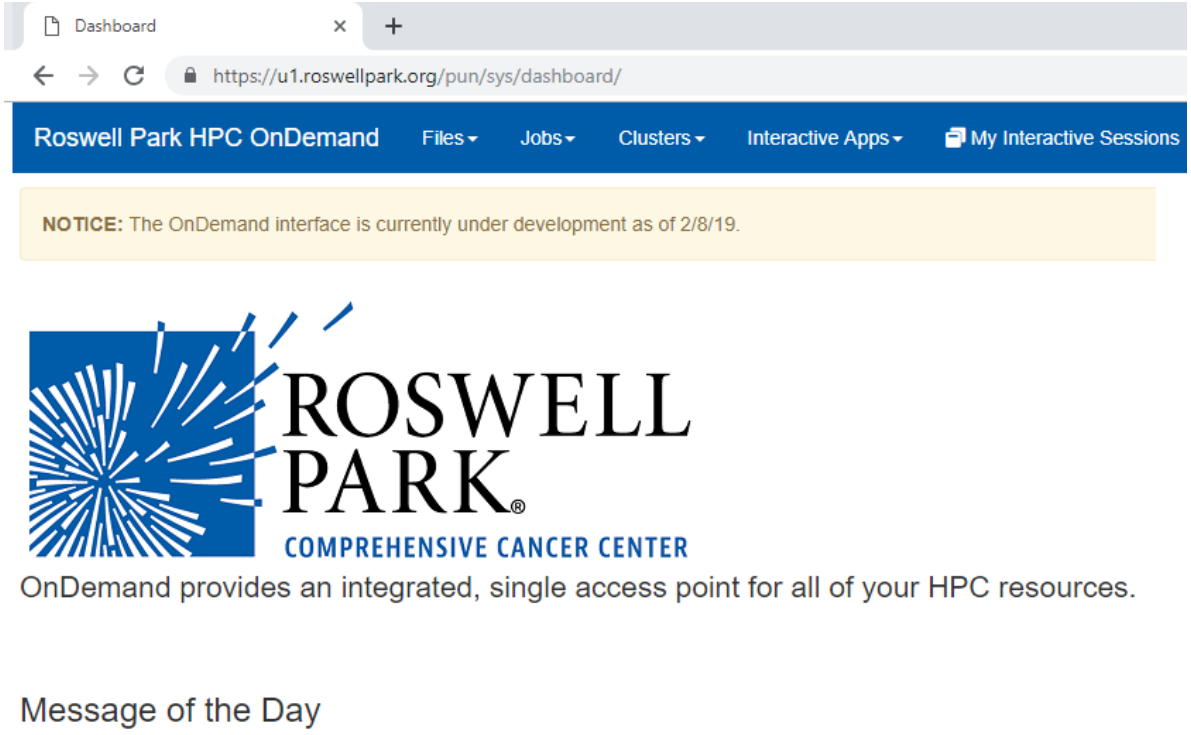
http://u1.roswellpark.org/pun/sys/dashboard/



– Note: *your OnDemand user ID and password may not the same as your main Roswell ID and password*

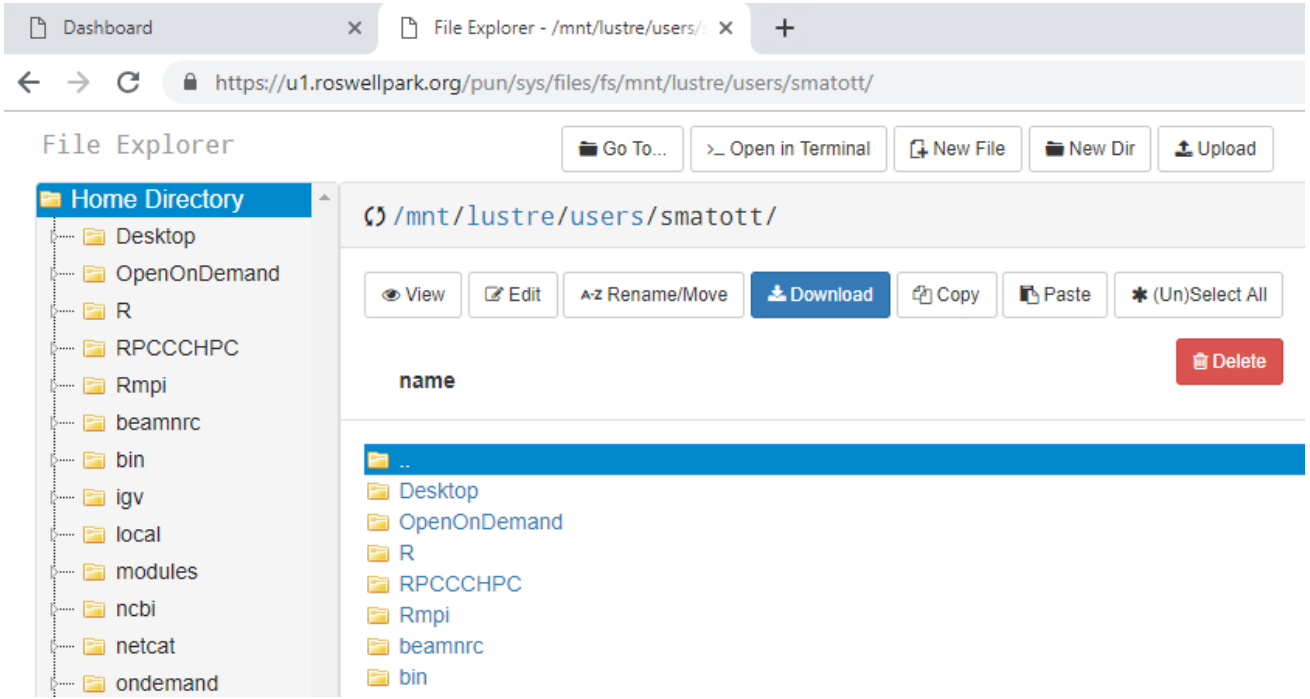
OnDemand Dashboard

- Files
  - Upload, download, edit, move, delete etc.
- Jobs
  - Configure, submit, & monitor jobs
- Clusters
  - Linux CLI
- Interactive Apps
  - Launch programs with GUIs



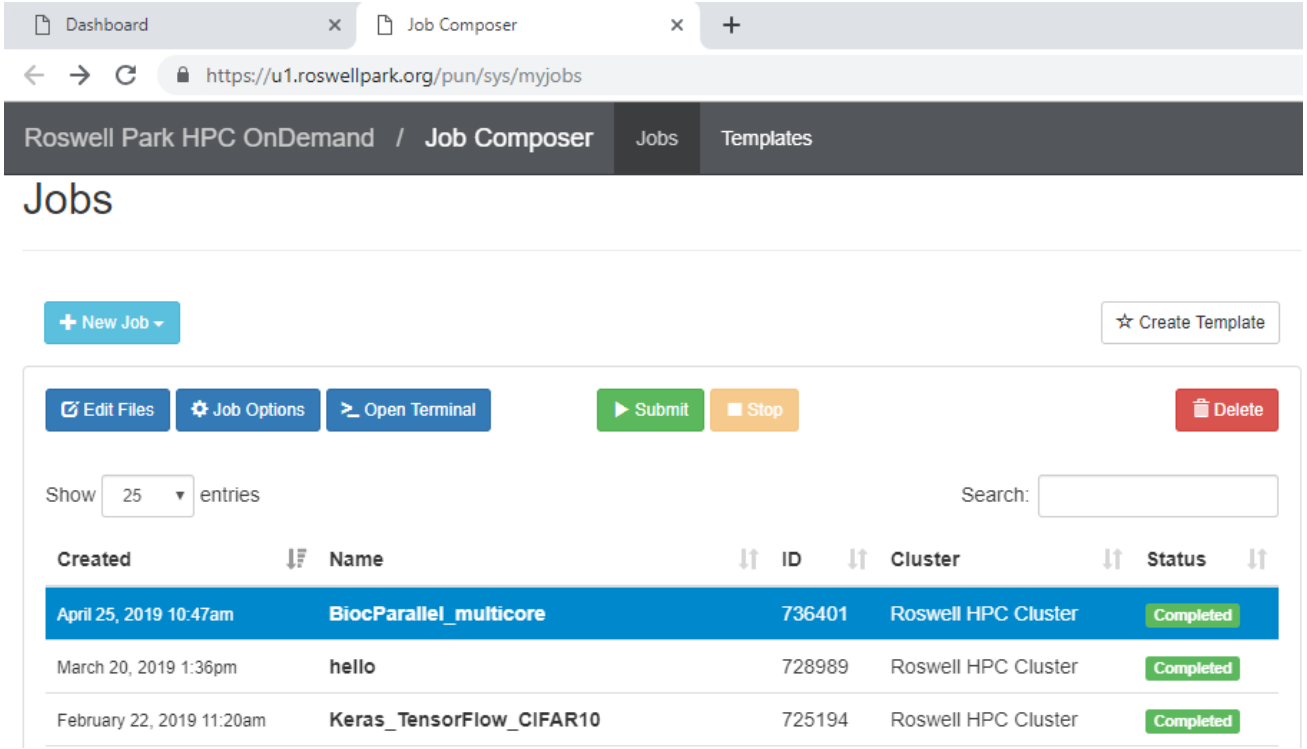
OnDemand - Files

- File Explorer
  - Transfers via upload/down-load or drag and drop
- Editor
  - Language aware coloring (e.g. Python comments, etc.)



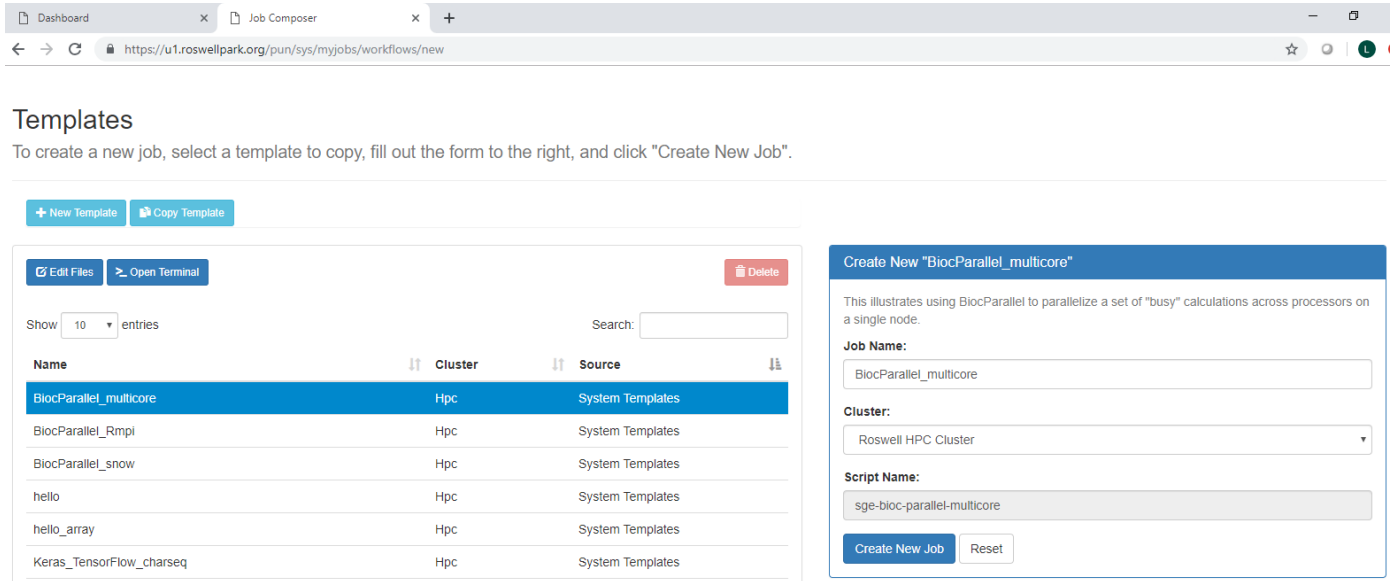
OnDemand – Compute Jobs

- Composer
  - Create and configure a new job
  - Use templates to minimize effort
  - Monitor or alter current jobs
- Active Jobs
  - Check job queues



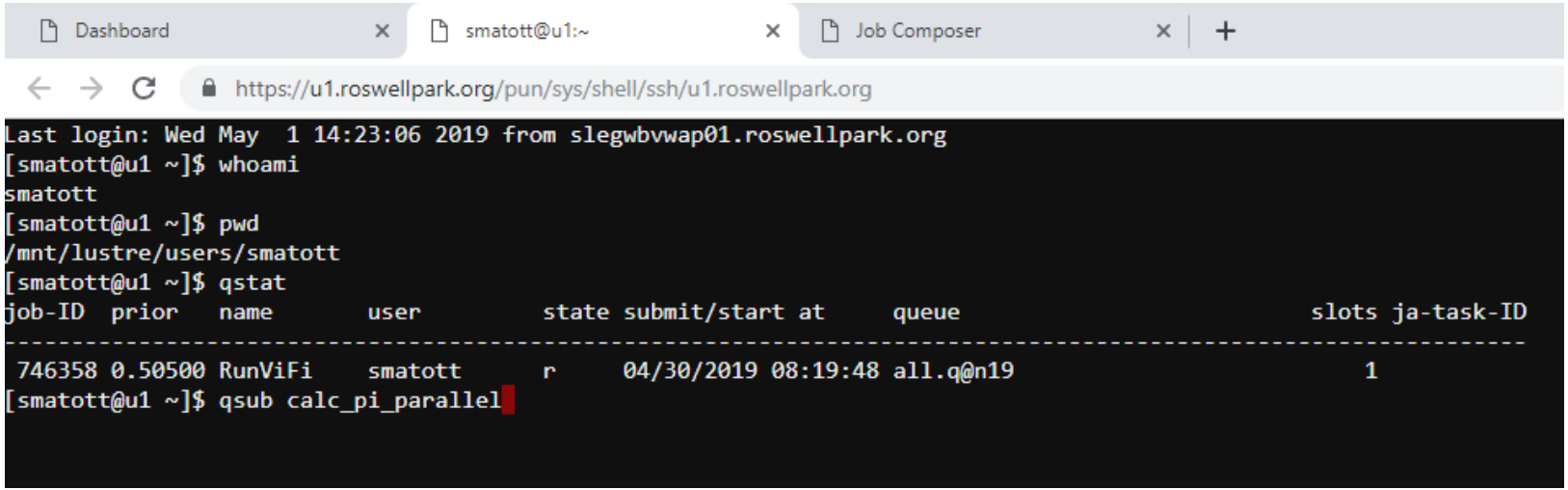
OnDemand – Job Templates

- Templates
  - Searchable
  - IT RC (i.e. Shawn) can create new templates on request
  - Users can also create and manage their own templates



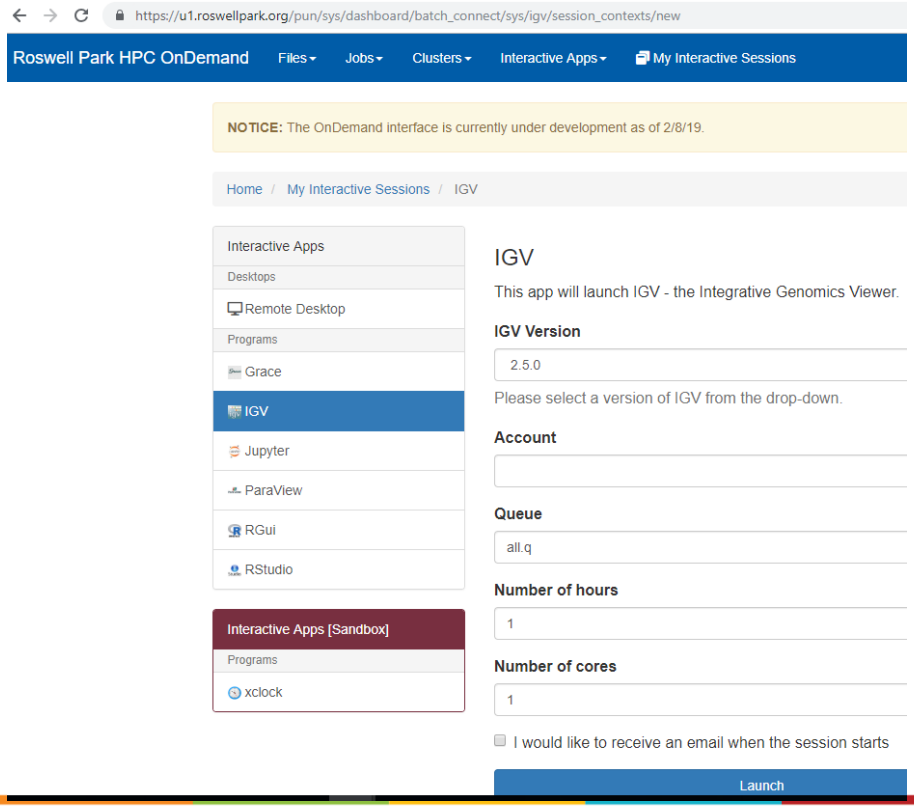
OnDemand – Cluster CLI

- Provides a bash command line interface on front-end node (u1)
- Handy for those who are familiar with conventional HPC access paradigm (e.g. putty, ssh)



OnDemand – Interactive Apps

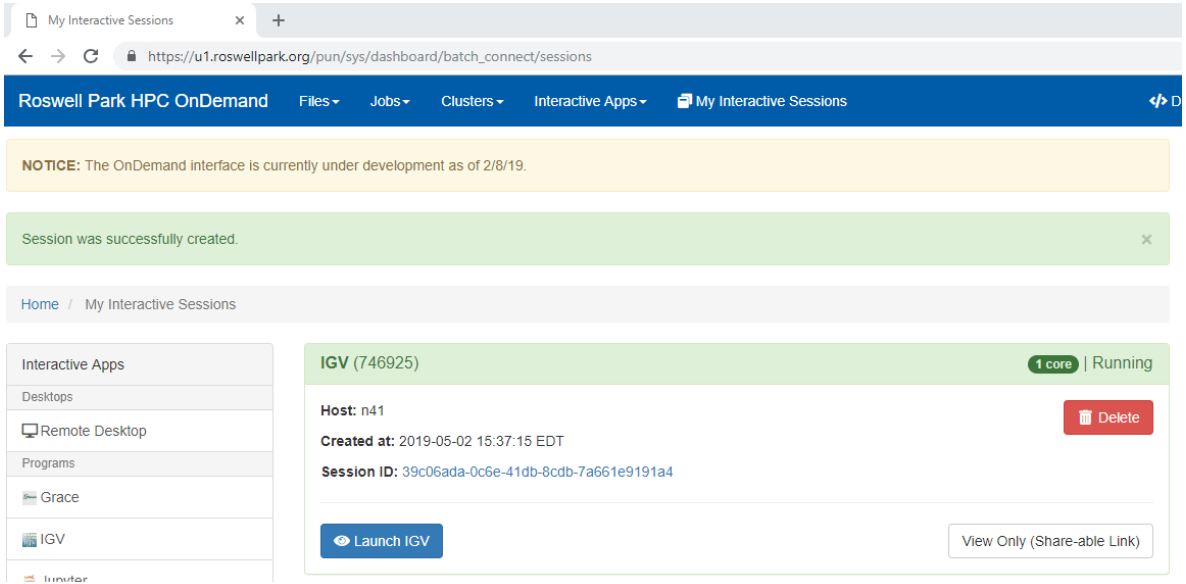
- Special compute jobs for launching interactive GUI-based applications
- Remote Desktop + Individual Apps
- OpenGL codes can benefit from GPU hardware acceleration (viz nodes)



Use the app window at left to select an app and submit request via “Launch” button.

OnDemand – Interactive Sessions

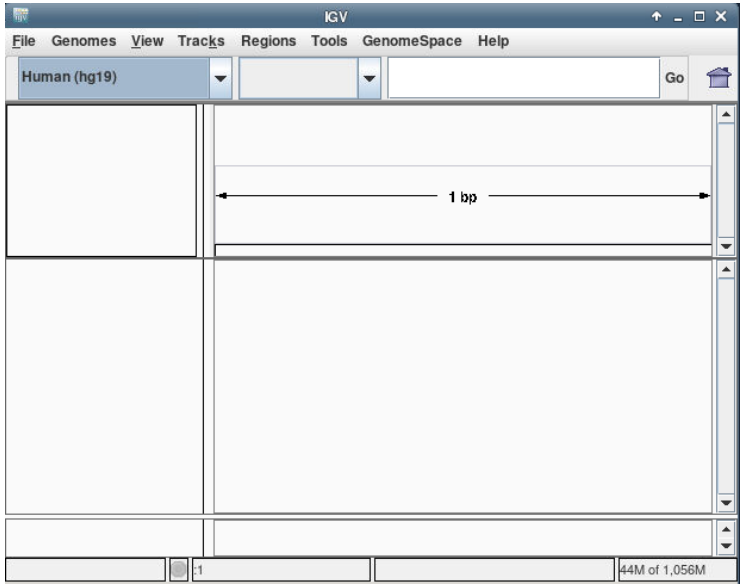
- Interactive apps are compute jobs!
- Can close browser window and reconnect without loss of work.



Once the job starts the window will refresh and offer a "Launch [app]" button. Click it to open the app in a browser window.

OnDemand – Interactive Sessions

- Interactive apps are compute jobs!
- Can close browser window and reconnect without loss of work.



In this example, a remote desktop is opened and pre-populated with an instance of the IGV genome viewer.

OnDemand – Machine Learning example

- Dashboard → Jobs→Job Composer → Templates
- Search for “Keras”
- Select the “wordseq” template
  - Generate HPL-style prose!
- Click “Create New Job”
  - Drops you into the “Jobs” interface
- Explore the job details area
- Click on the “.py” file to open in editor
  - contains the Keras code
- Find the section marked “define model”
  - Feel free to add, delete, modify layers

```
170     # define model
171     model = Sequential()
172     model.add(Embedding(vocab_size, 50, input_length=seq_length))
173     model.add(LSTM(100, return_sequences=True))
174     model.add(LSTM(100))
175     model.add(Dense(100, activation='relu'))
176     model.add(Dense(vocab_size, activation='softmax'))
177     print(model.summary())
178
179     # train the model
180     model.compile(loss='categorical_crossentropy', optimizer='ada
181     model.fit(X, Y, batch_size=128, epochs=100, verbose=2)
182
183     # calculate and report performance of the model
184     print("")
185     print("---------------------------------")
186     print_confusion_matrix(model, X, Y, inv_mapping)
187     print("---------------------------------")
188
189     # save model
190     model.save('model.h5')
```

OnDemand – Machine Learning example

- Find the section marked “procedure == “generate”
  - Feel free to adjust the generator sequence and see what sort of Lovecraftian text you can get the AI to come up with!
  - The sequence must be at least 10 words and include a space at the end.

```
322     elif(procedure == "generate"):
323         passage = generate_sequence("the horror of it all was quite something to behold ", 500)
324         passage = " ".join(str(x) for x in passage)
325         passage = passage.replace('!', '\n')
326         print("")
327         print("*"*20)
328         print(passage)
329         print("*"*20)
330         print("")
```

OnDemand – Machine Learning example

- Open the sge-keras-wordseq script in the editor
- Add lines at the bottom to run test and generate portions of the model:

```
23  python keras_wordseq_dunwich.py $NSLOTS train
24  python keras_wordseq_dunwich.py $NSLOTS test
25  python keras_wordseq_dunwich.py $NSLOTS generate
```

- Save changes
- Job Composer → Submit
- Wait for job to complete
- Open wordseq.out →

```
4130  ********************
4131  the horror of it all was quite something to behold the panic bearing in household careful the copy
4132   muscular hellish elder and boarded non users
4133   and menaces and demoniac
4134   having was instruct of settled business in the fetor in all and grinned were this time wilbur afterward
4135   bits of no direction like night and only a restraint and very stigmata of loading of the neighboring
4136   and drag it and foul and a chance of putting ye the men then those fragments and and a hastily
4137   had less road there was
4138   crumpled to the
4139
4140   edition died came his use it seemed to comment in the wildest green allowed
4141   blasted albino yog sothoth he was in any use
4142   rate and way to certain availabl the singularly state of crushed lantern sounds as haow whose heads
4143   grow ahead
4144   seemed to let to seth is no frye then against any books fer
4145   aout over what was armitage walk from the hill indoors as the doors of growing void which even earl
4146   though pull he feared
4147   disappeared near down
4148   dogs he shuddered were tarlike ancient to let in wilbur was tell it was creaters other distaste no
4149   only the bending beside the smell are
4150   why partly
4151   but in august he caved
```