Introduction to Machine Learning with the Tufts High Performance Research Cluster

Sergey Voronin

Department of Mathematics, Tufts University
In collaboration with Tufts Technology Services

16 May 2017

Thanks to

- Shawn Doughty, Sr. Research Technology Specialist
- Durwood Marshall, Fmr Sr. Research Technology Specialist
- Lionel Zupan, Dir. Research Technology
- David Lapointe, Sr. Bioinformatics Specialist

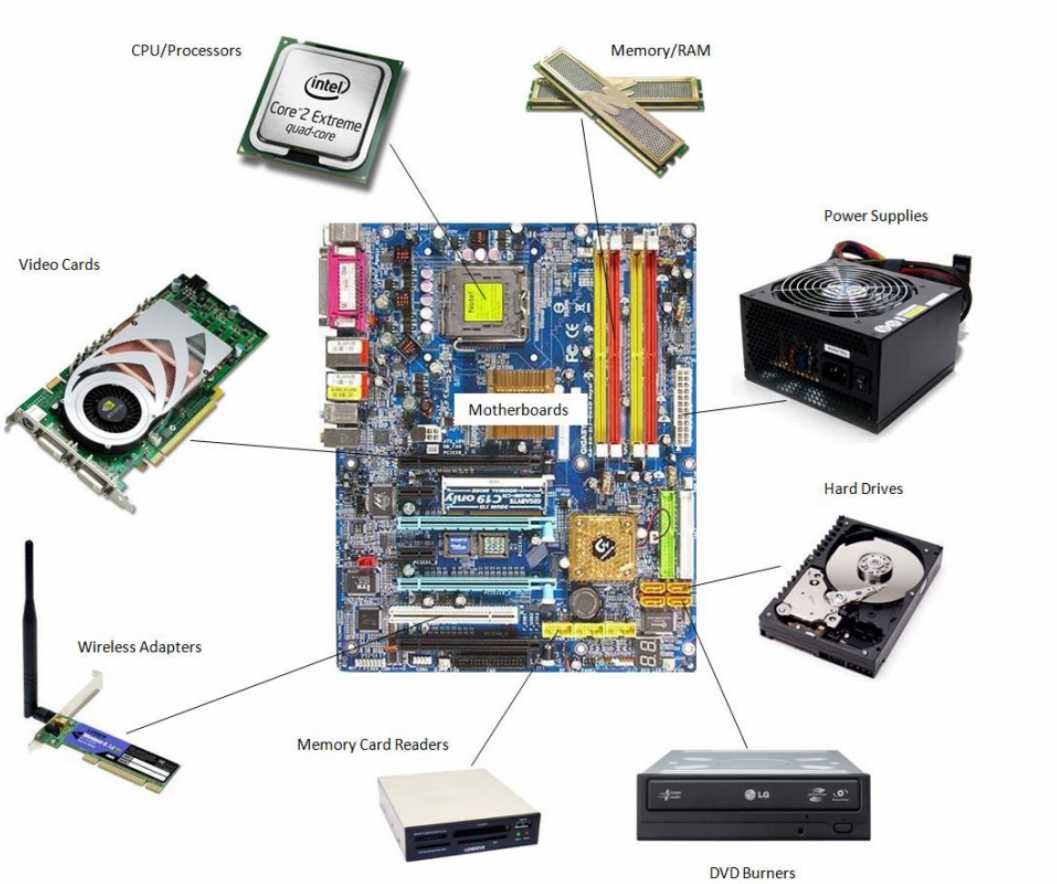
Agenda

- Layout of a cluster.
- Connection and file transfer.
- Slurm resource manager.
- Intro to machine learning.
- Using KNN and SVMs with R and Python.
- Applying deep learning to the ESC data set from scratch with Java and Python.

1. Anatomy of a computer and of a computer cluster

- Fundamental computer parts
- Layout of different nodes in a typical research cluster

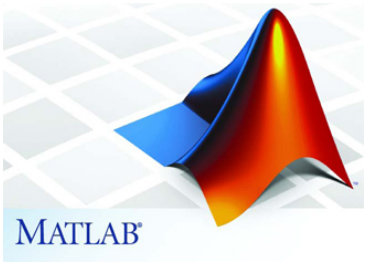
Computer components

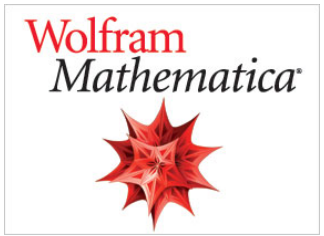


- CPU for computation
- RAM and CPU cache for intermediate process storage
- Hard drive for permanent storage

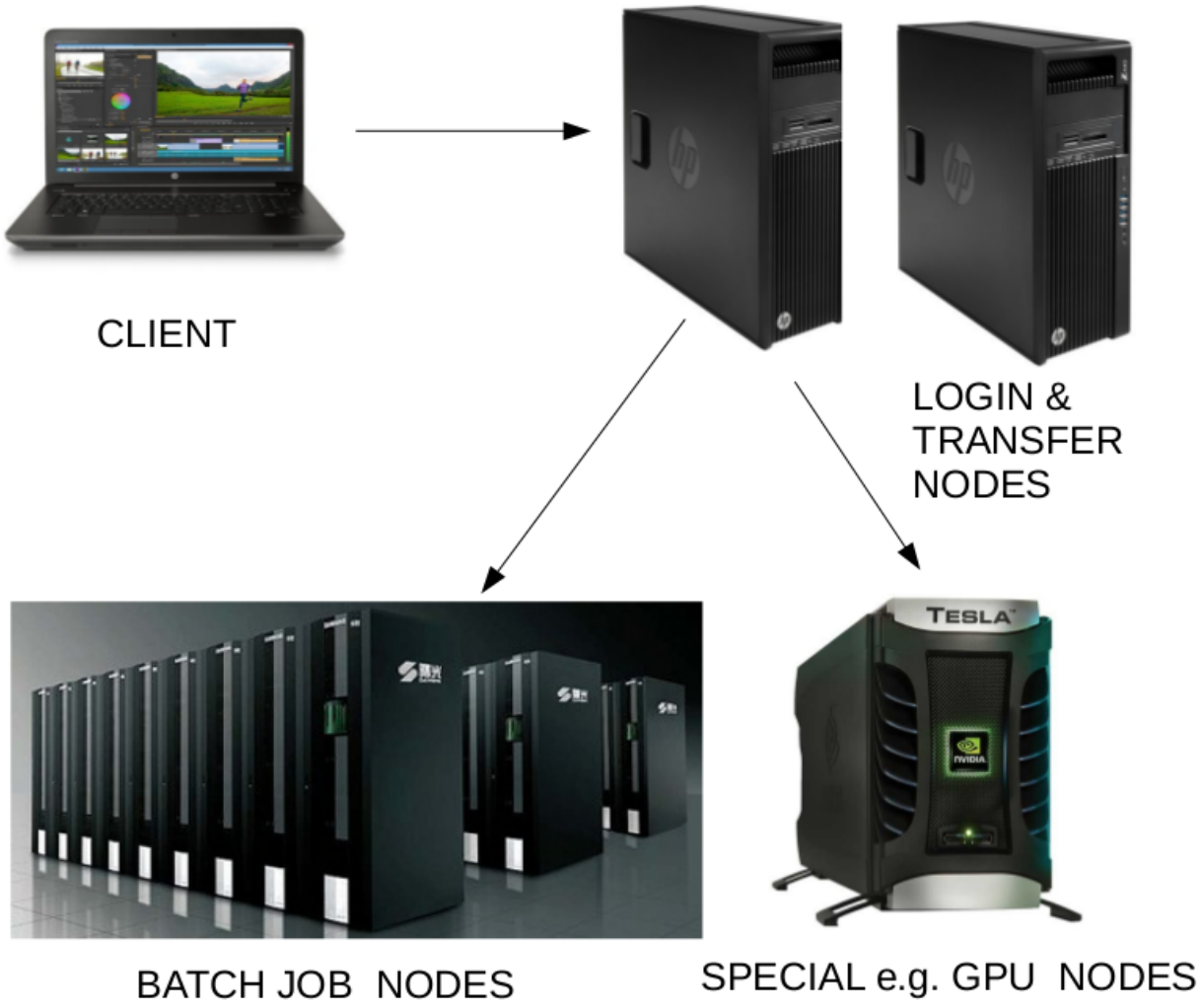
Computing systems and software

- Different problems demand various CPU, RAM, and disk resources.
- Various software maybe useful (some requires hefty licensing fees).





Layout of the computational cluster



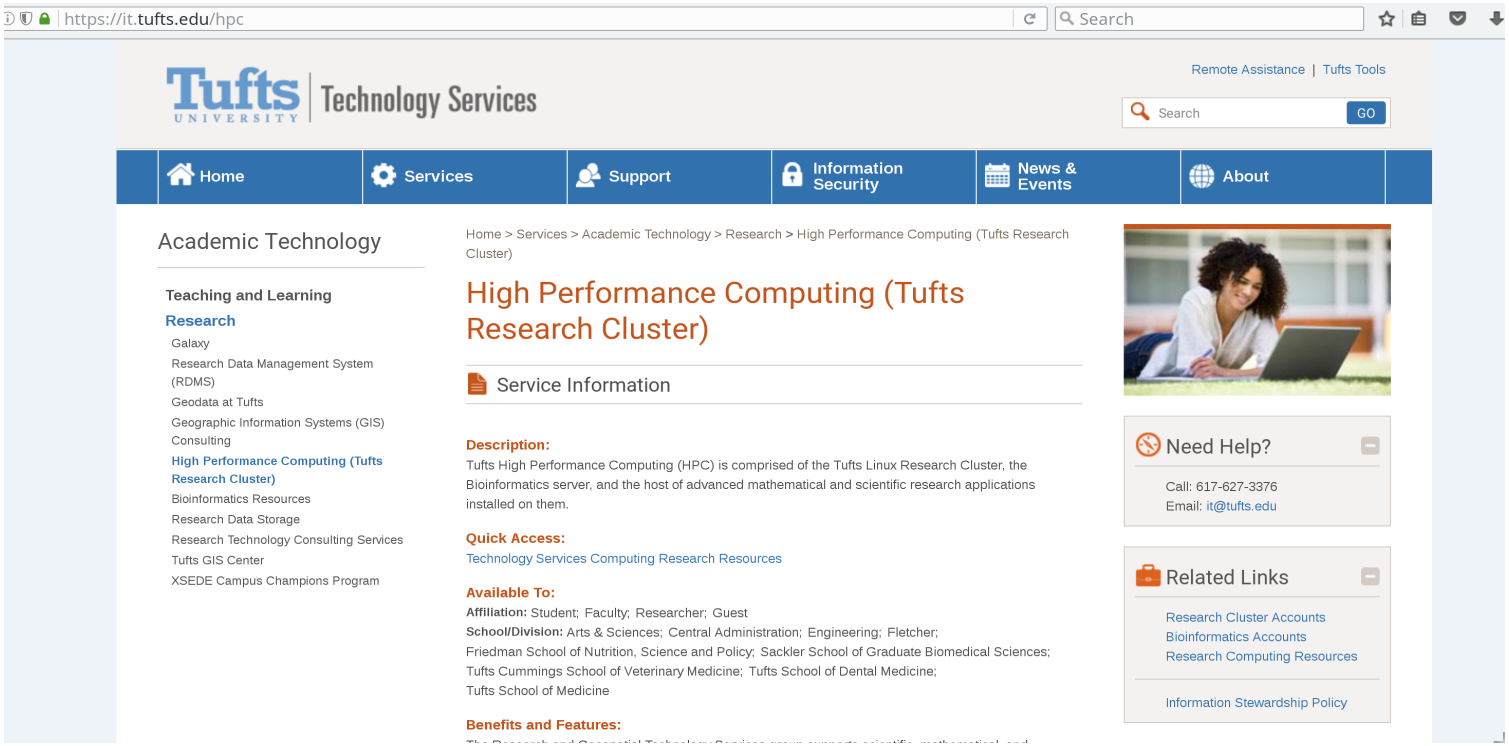
⇒ Don't run computations on login and xfer nodes. These are special nodes used to process logins and for file transfer.

2. Connection and file transfer

- Getting an account
- ssh connection fundamentals
- File transfer via scp and rsync
- X11 server
- Windows and Mac OS specifics

Getting an account on the cluster

- See `https://it.tufts.edu/hpc`.
- Email `tts-research@tufts.edu`.
- Your username and password will be the same as your Tufts credentials.



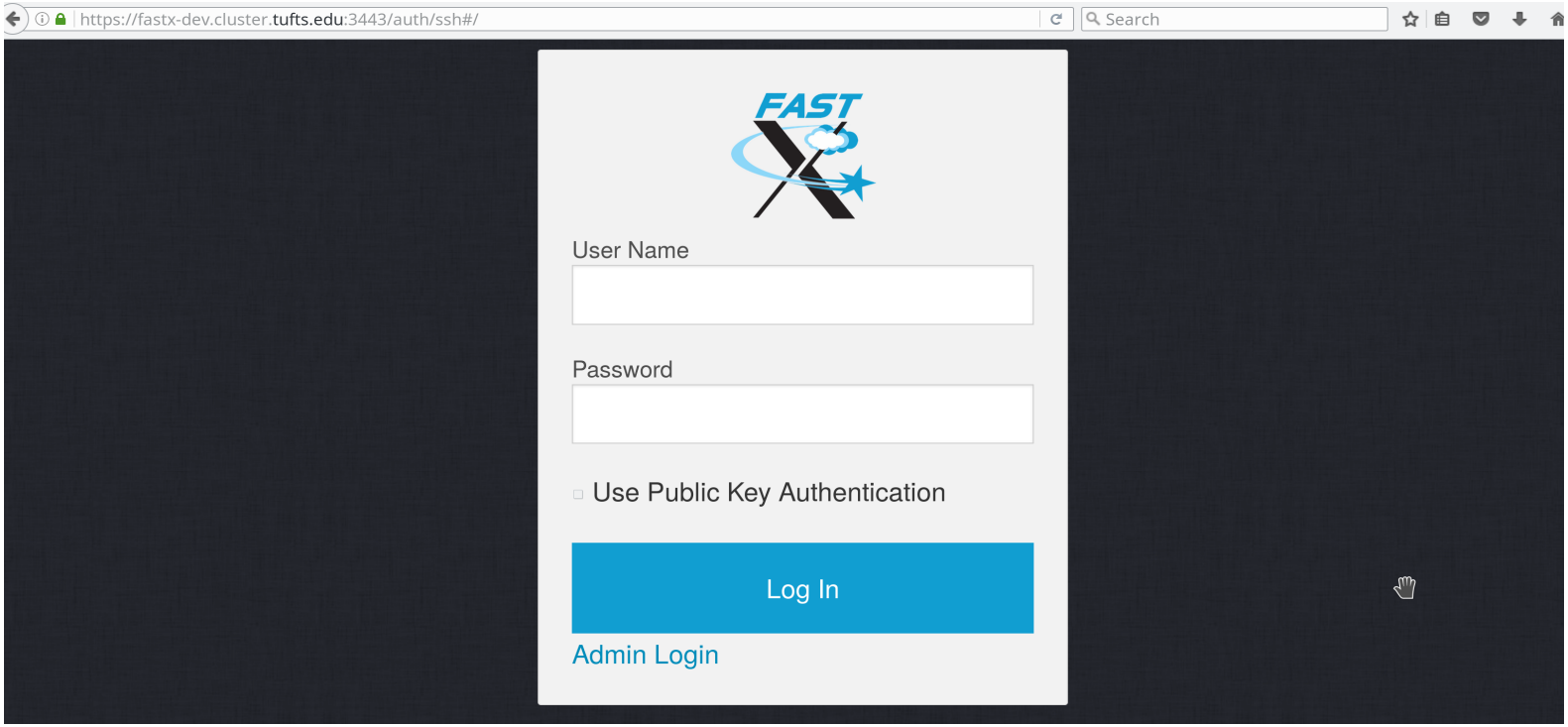
Essential software

- SSH connection software.
- SCP/RSYNC file transfer software.
- X11 Xorg server software for tunneling graphical applications.

Client to server connection via SSH protocol

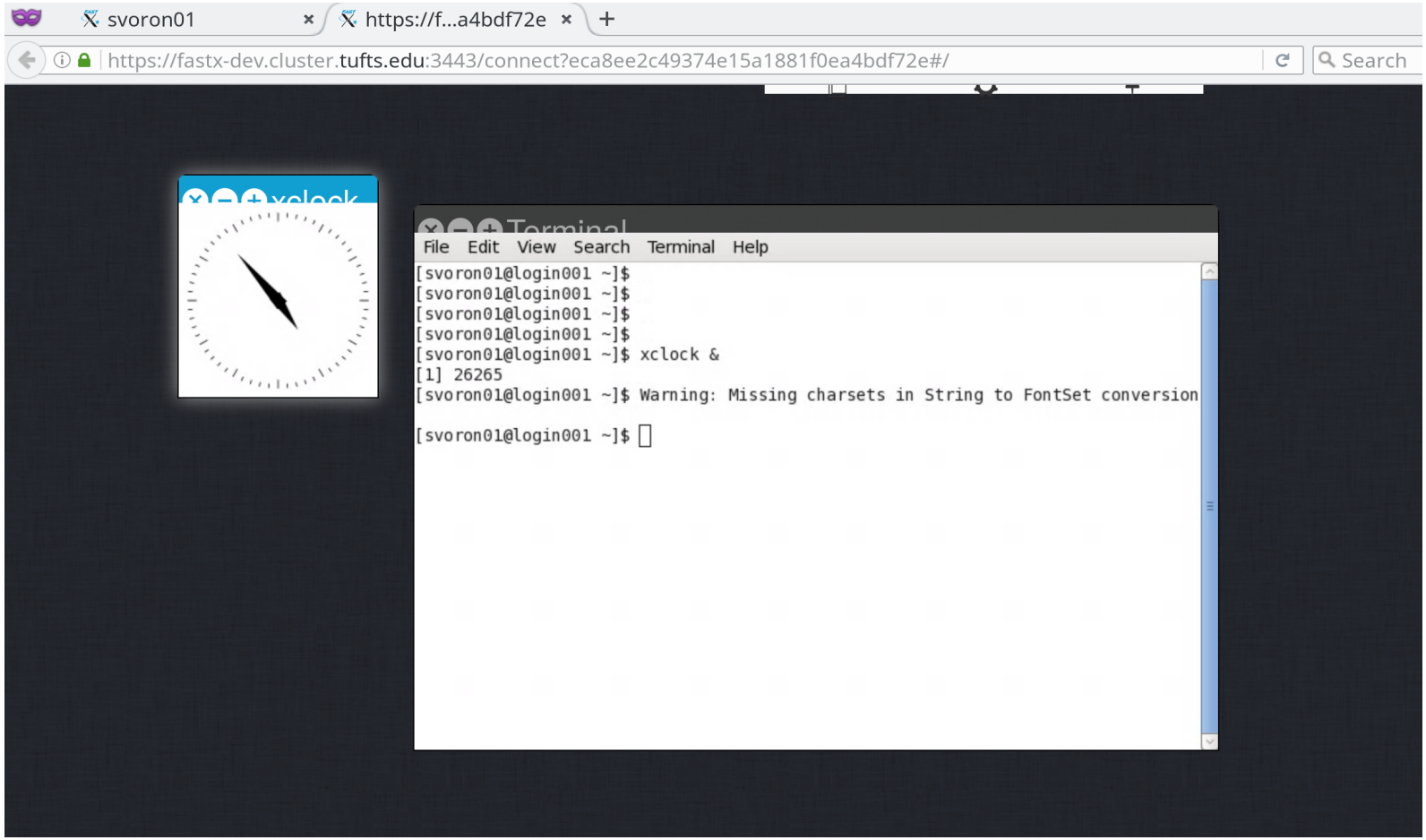
fastX web interface

- Go to `https://fastx-dev.cluster.tufts.edu:3443/`.
- Contact Shawn Doughty `shawn.doughty@tufts.edu`.
- Your username and password will be the same as your Tufts credentials.



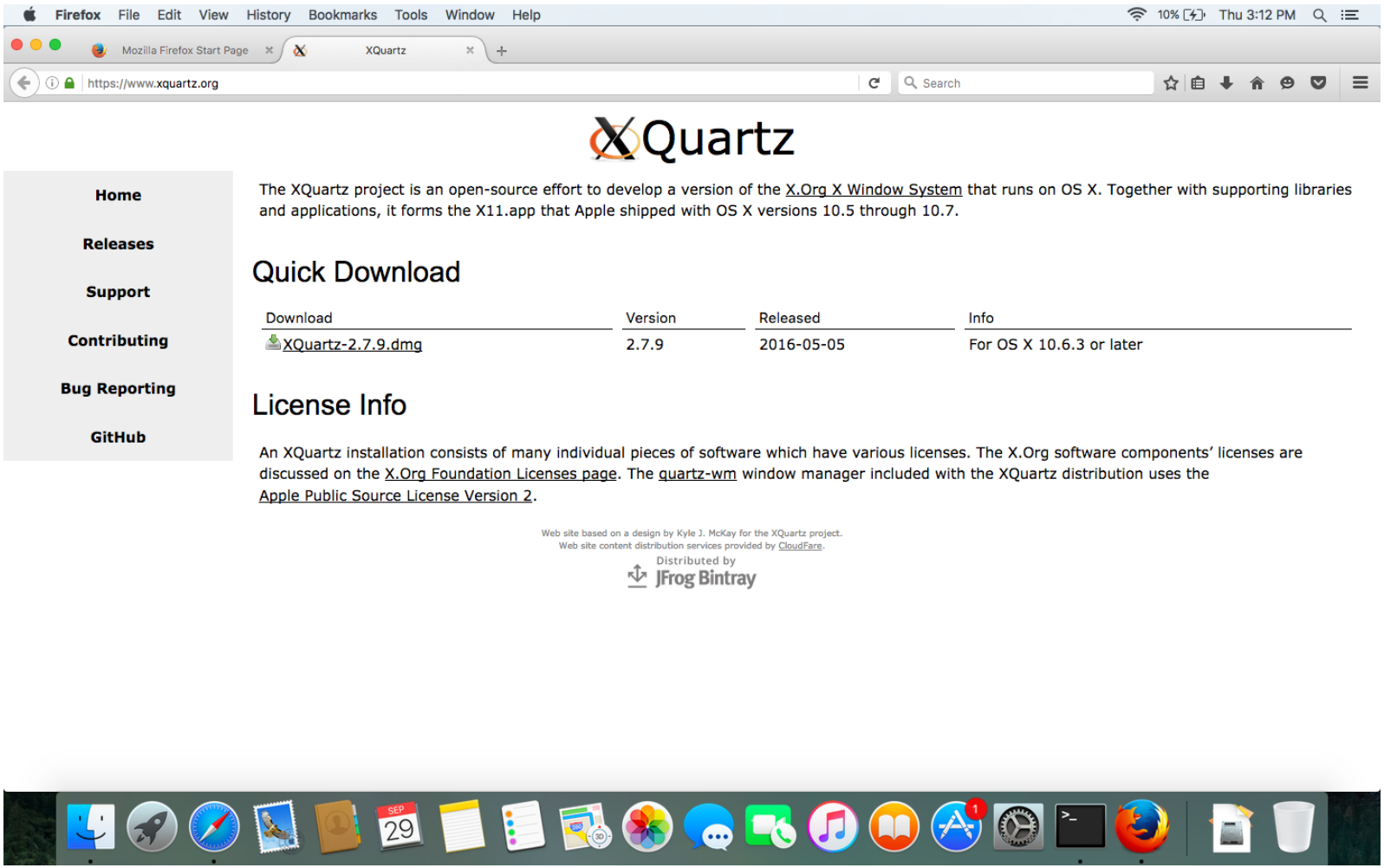
Using SSH with X11 via fastX web interface

Login and run xlock graphical application.

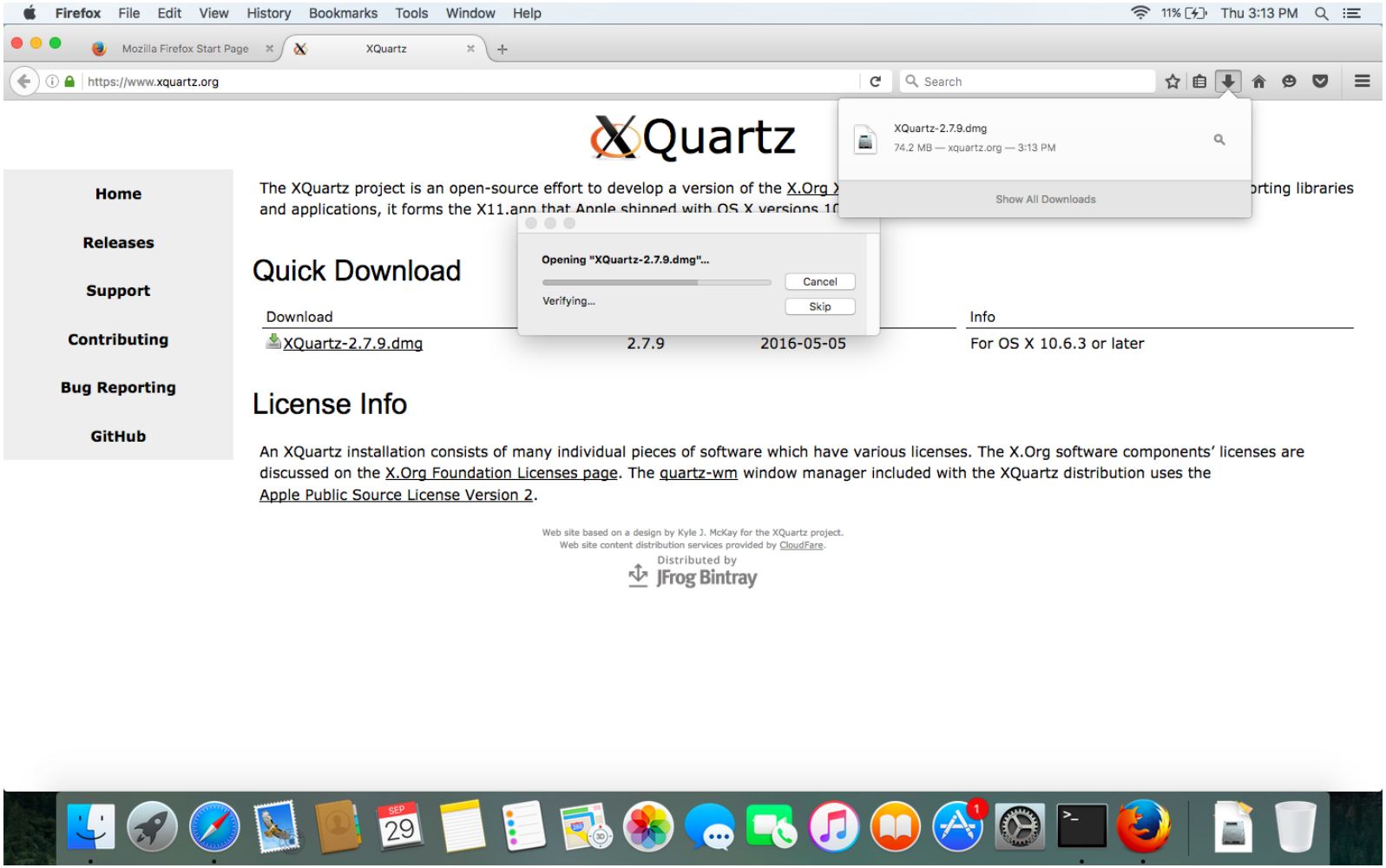


For general cluster use, this is sufficient. But it is useful to know how to do this on Windows, Mac, and Linux.

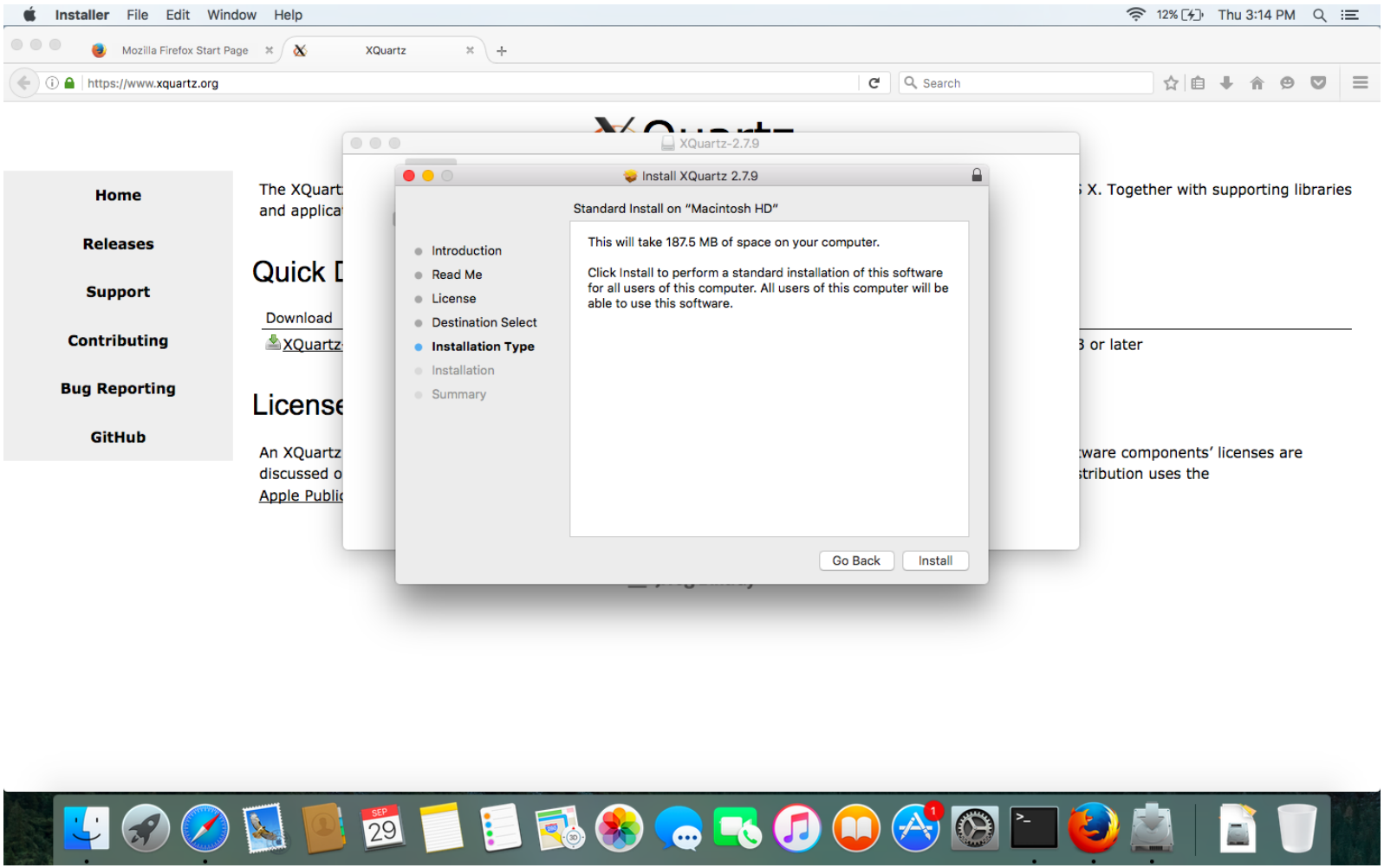
Using SSH with X11 on Mac OS X



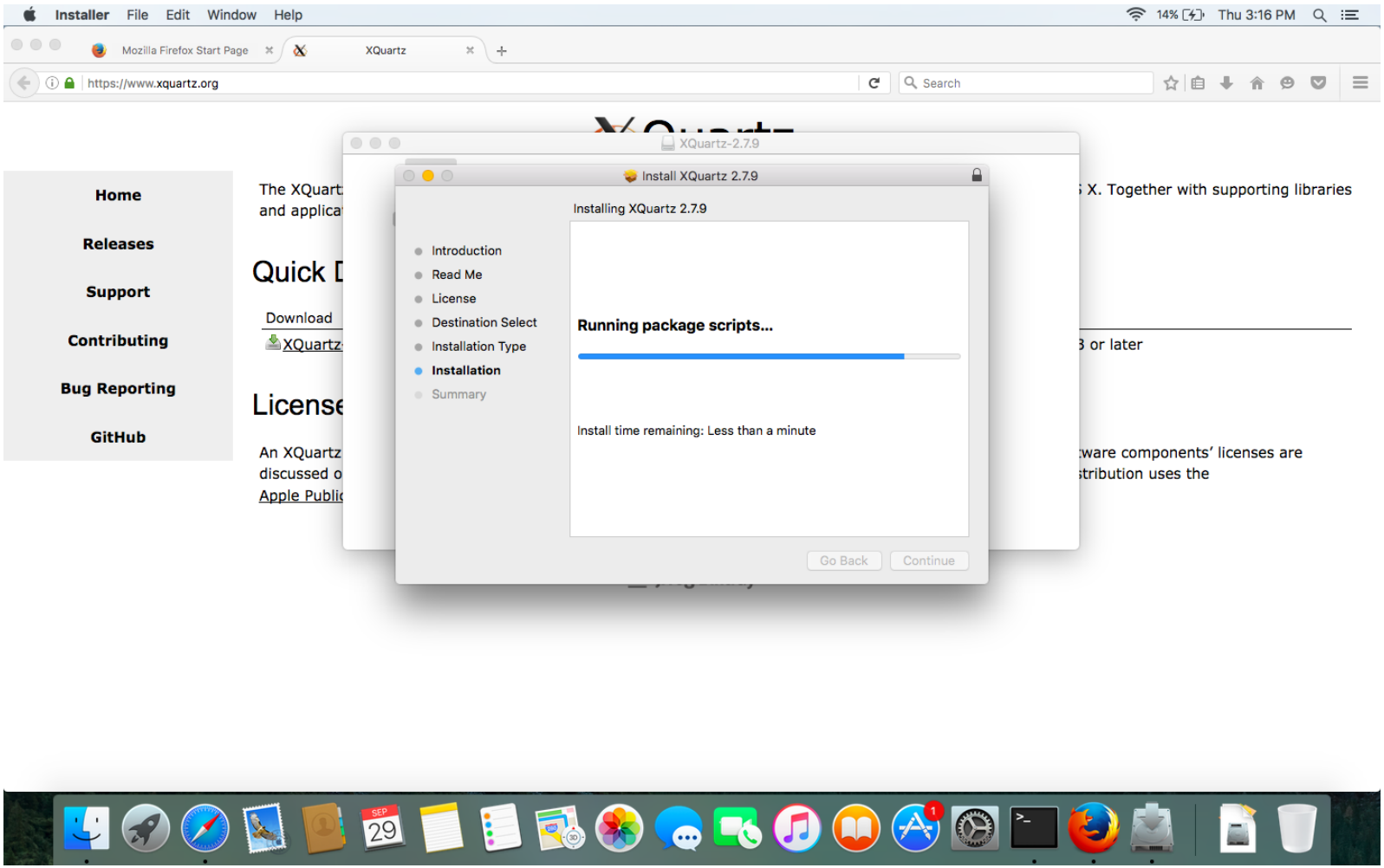
Using SSH with X11 on Mac OS X



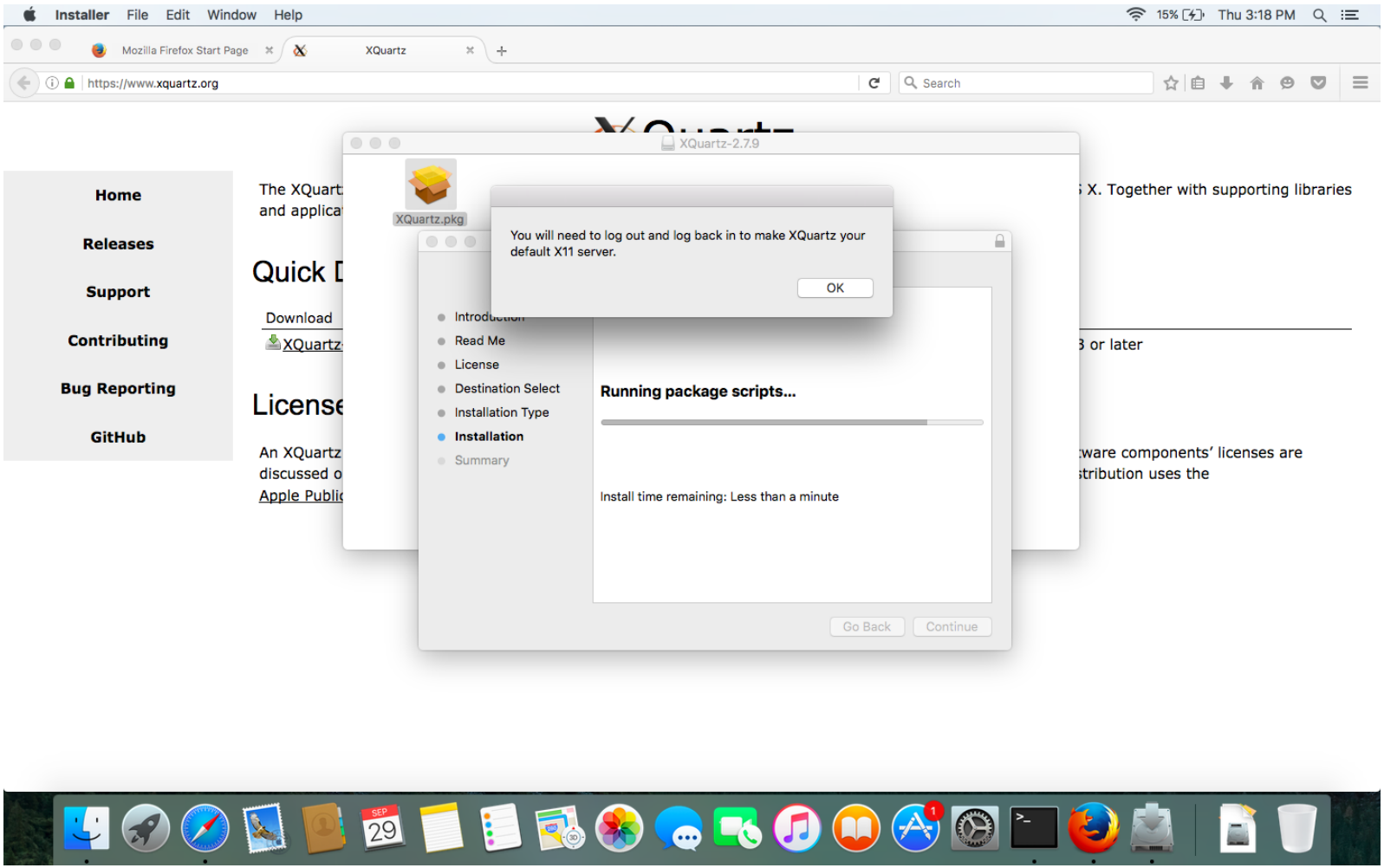
Using SSH with X11 on Mac OS X



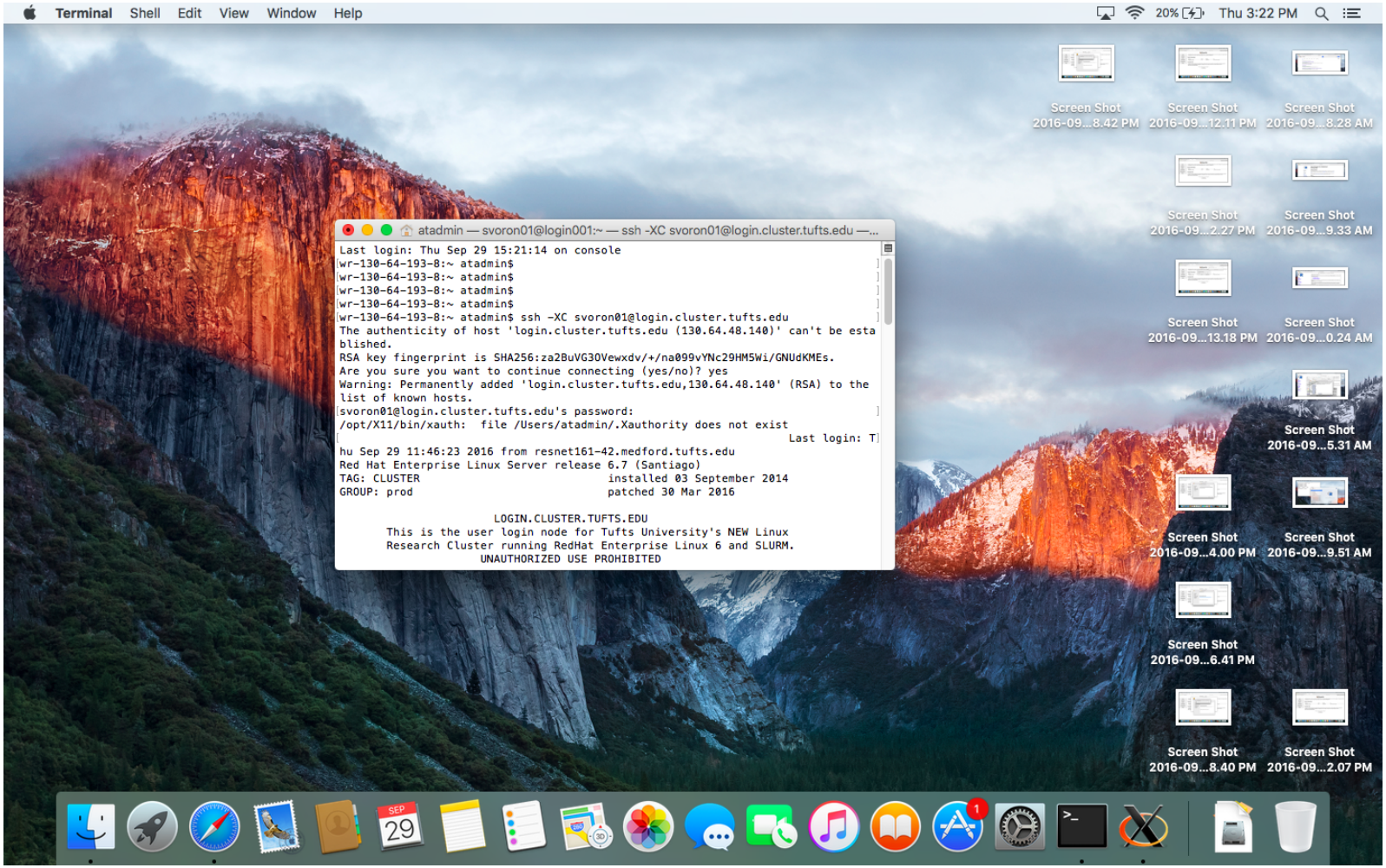
Using SSH with X11 on Mac OS X



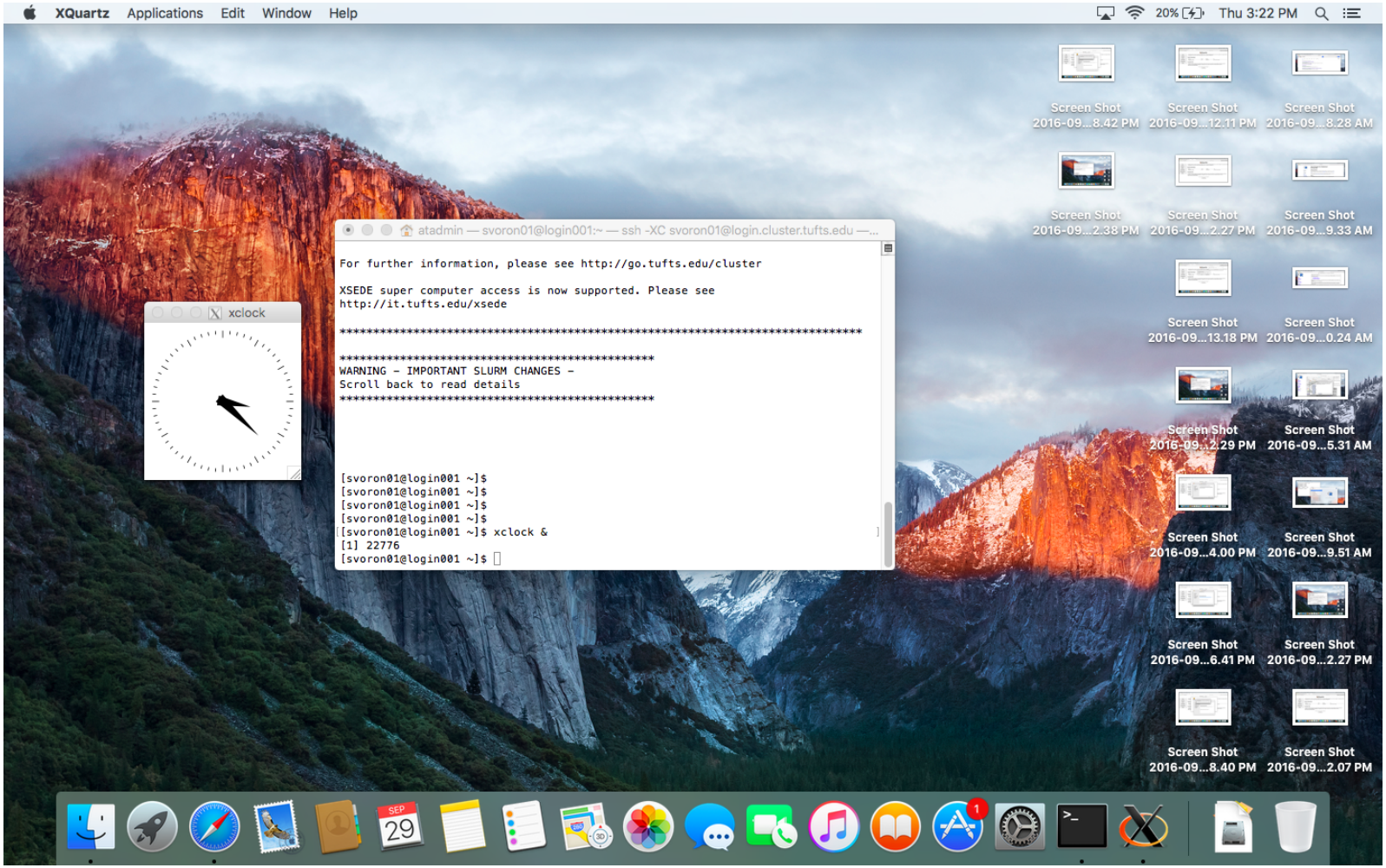
Using SSH with X11 on Mac OS X



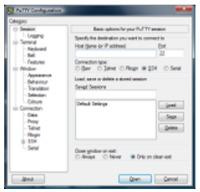
Using SSH with X11 on Mac OS X



Using SSH with X11 on Mac OS X



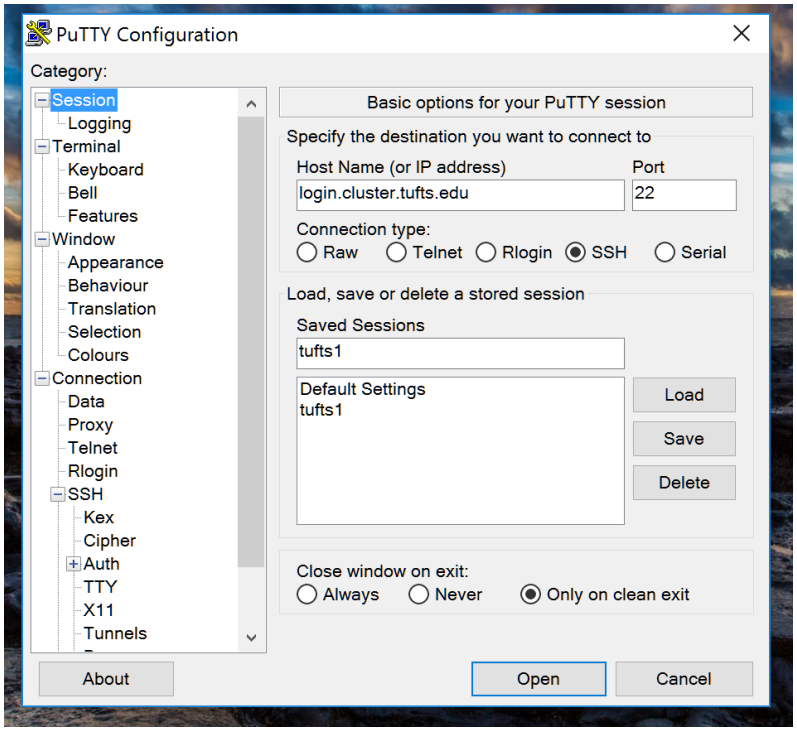
Using SSH with X11 on Windows 10



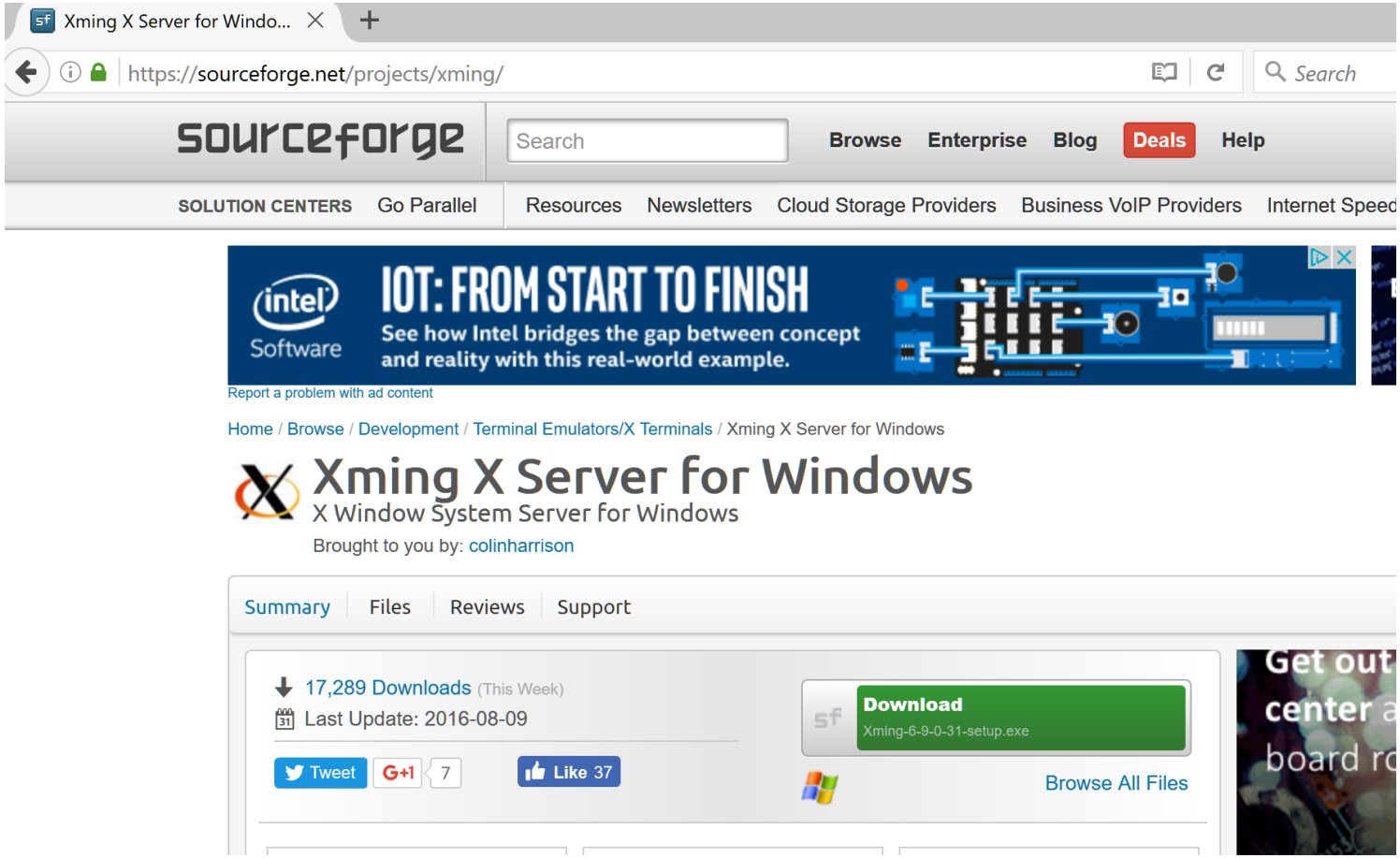
Download PuTTY

PuTTY is an SSH and telnet client, developed originally by Simon T
source software that is available with source code and is developec

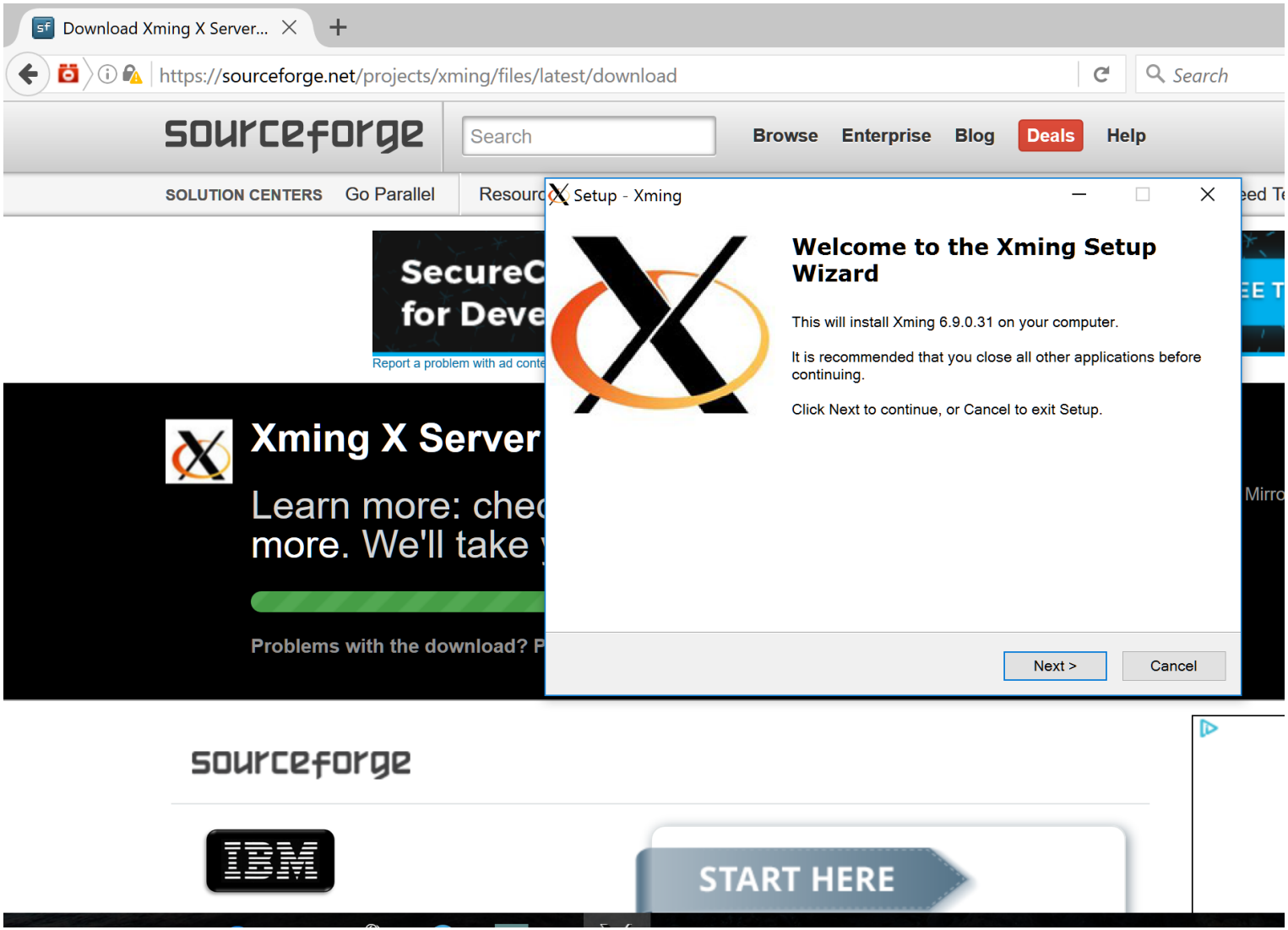
You can download PuTTY here.



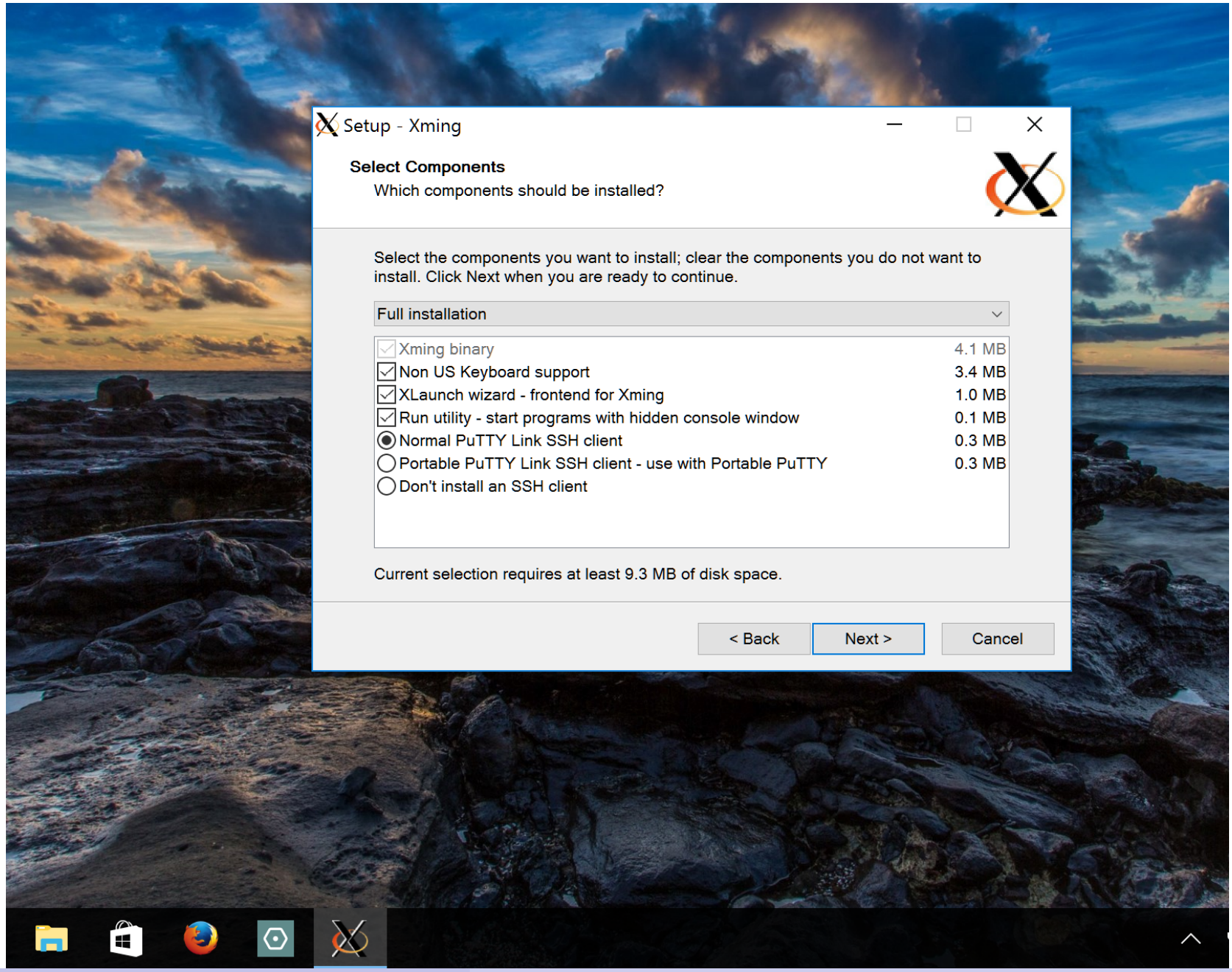
Using SSH with X11 on Windows 10



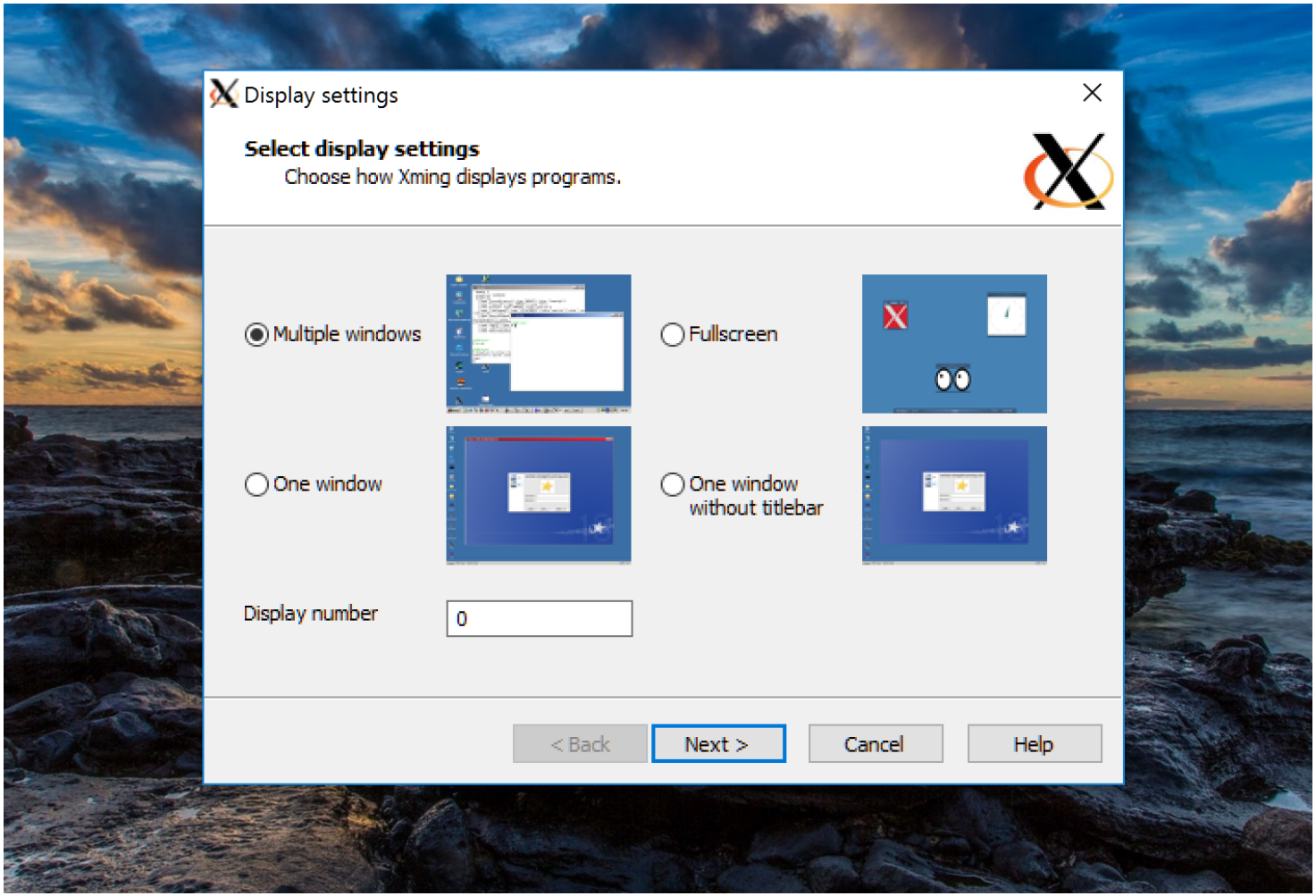
Using SSH with X11 on Windows 10



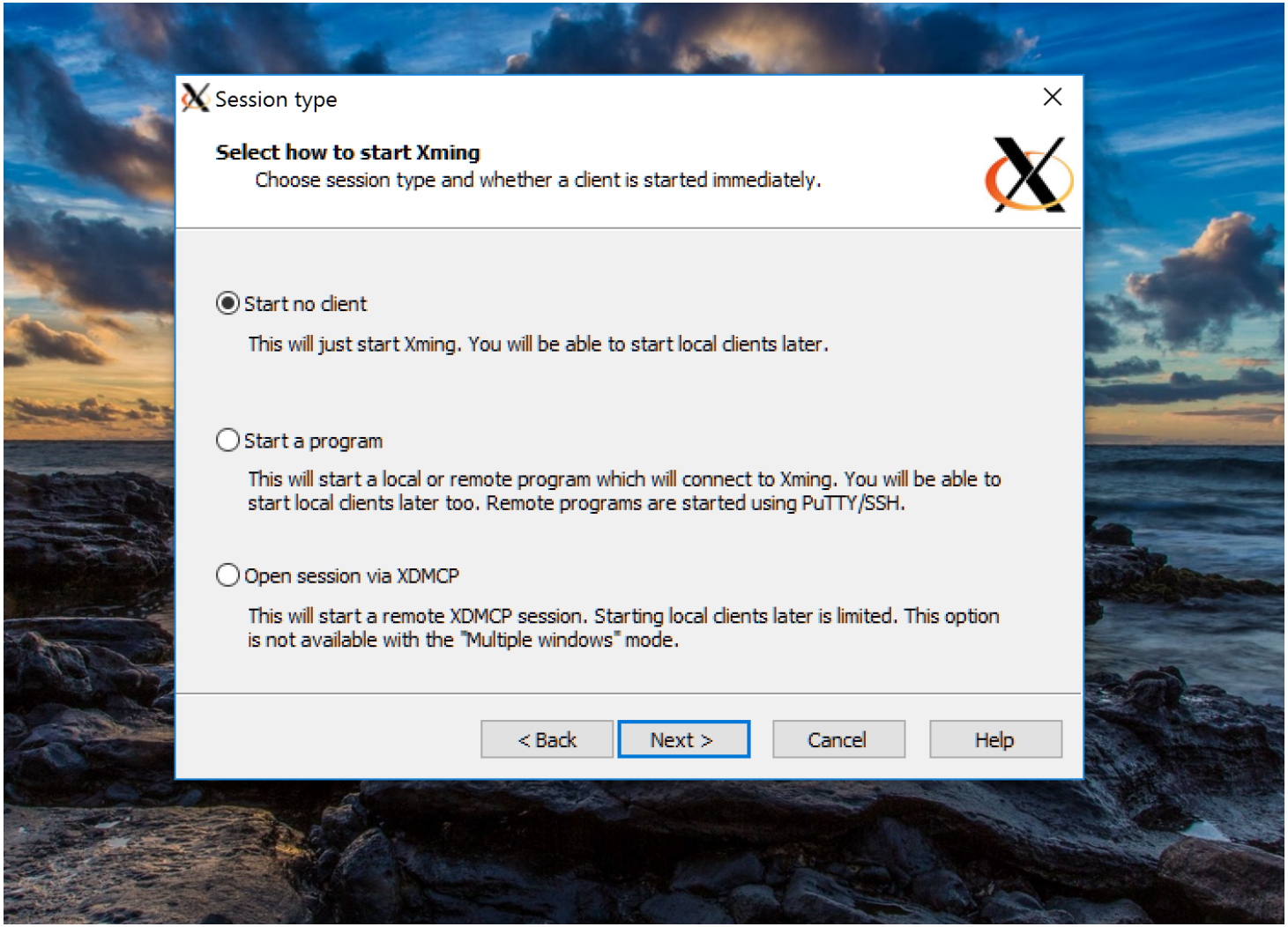
Using SSH with X11 on Windows 10



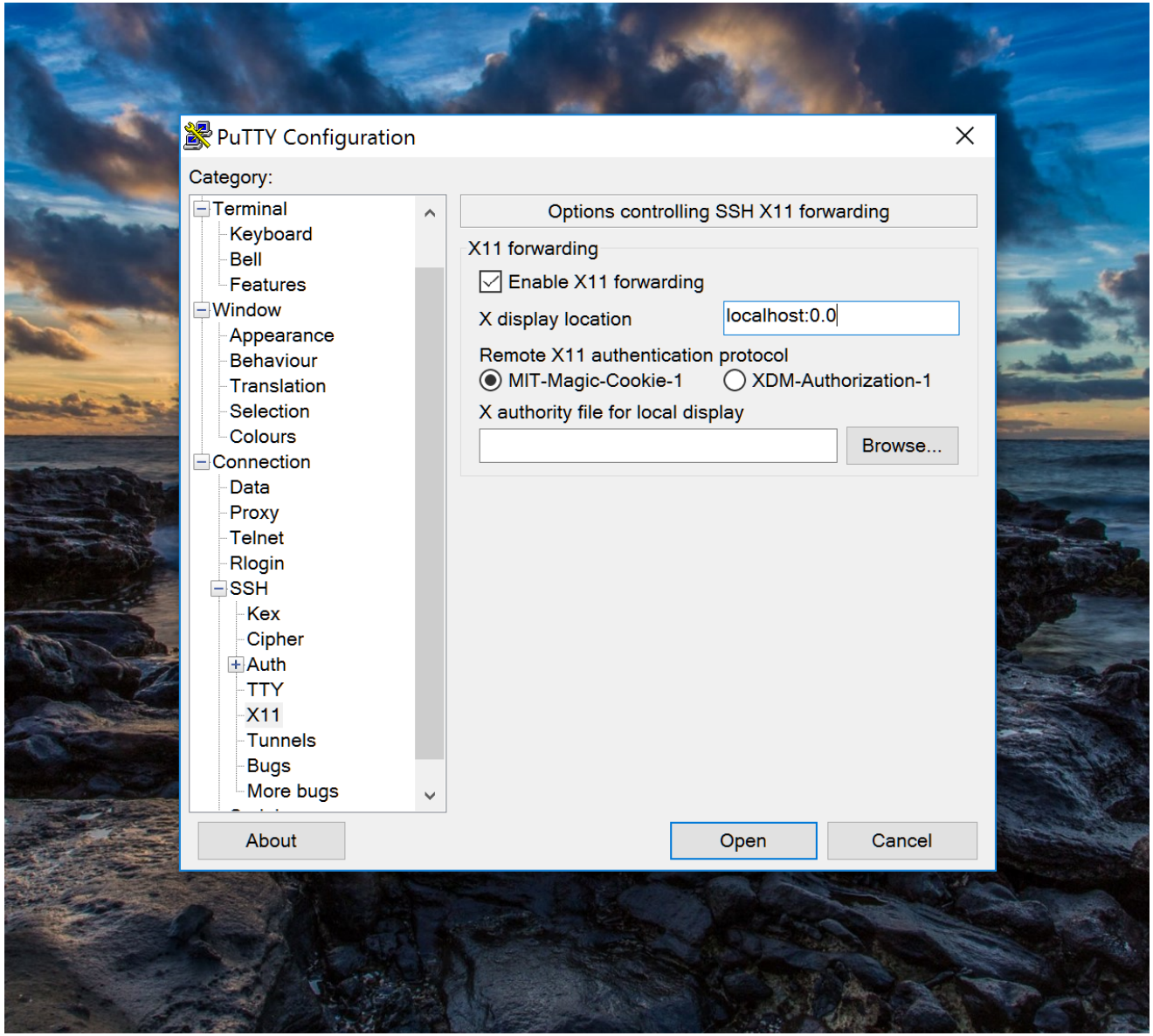
Using SSH with X11 on Windows 10



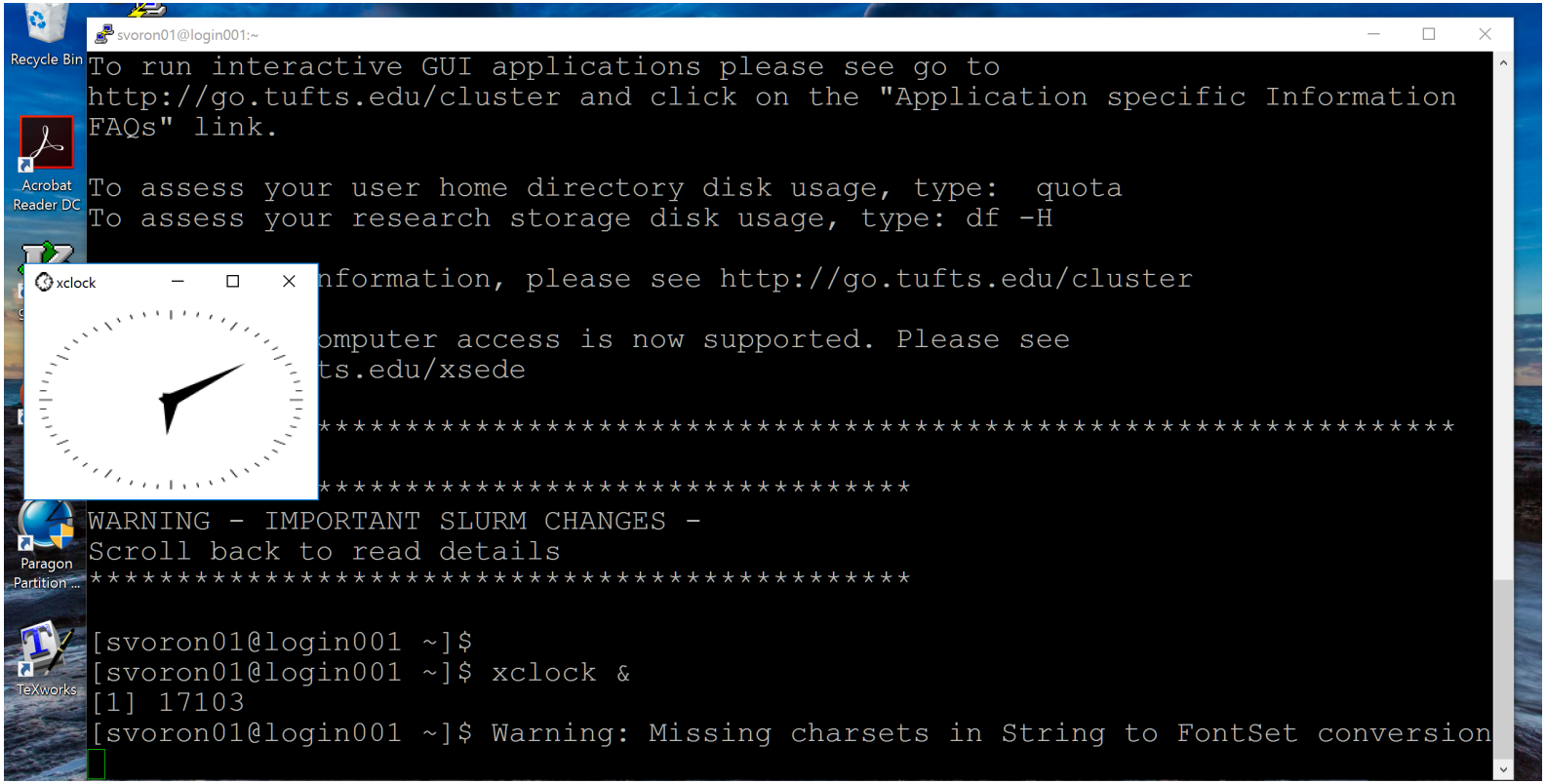
Using SSH with X11 on Windows 10



Using SSH with X11 on Windows 10

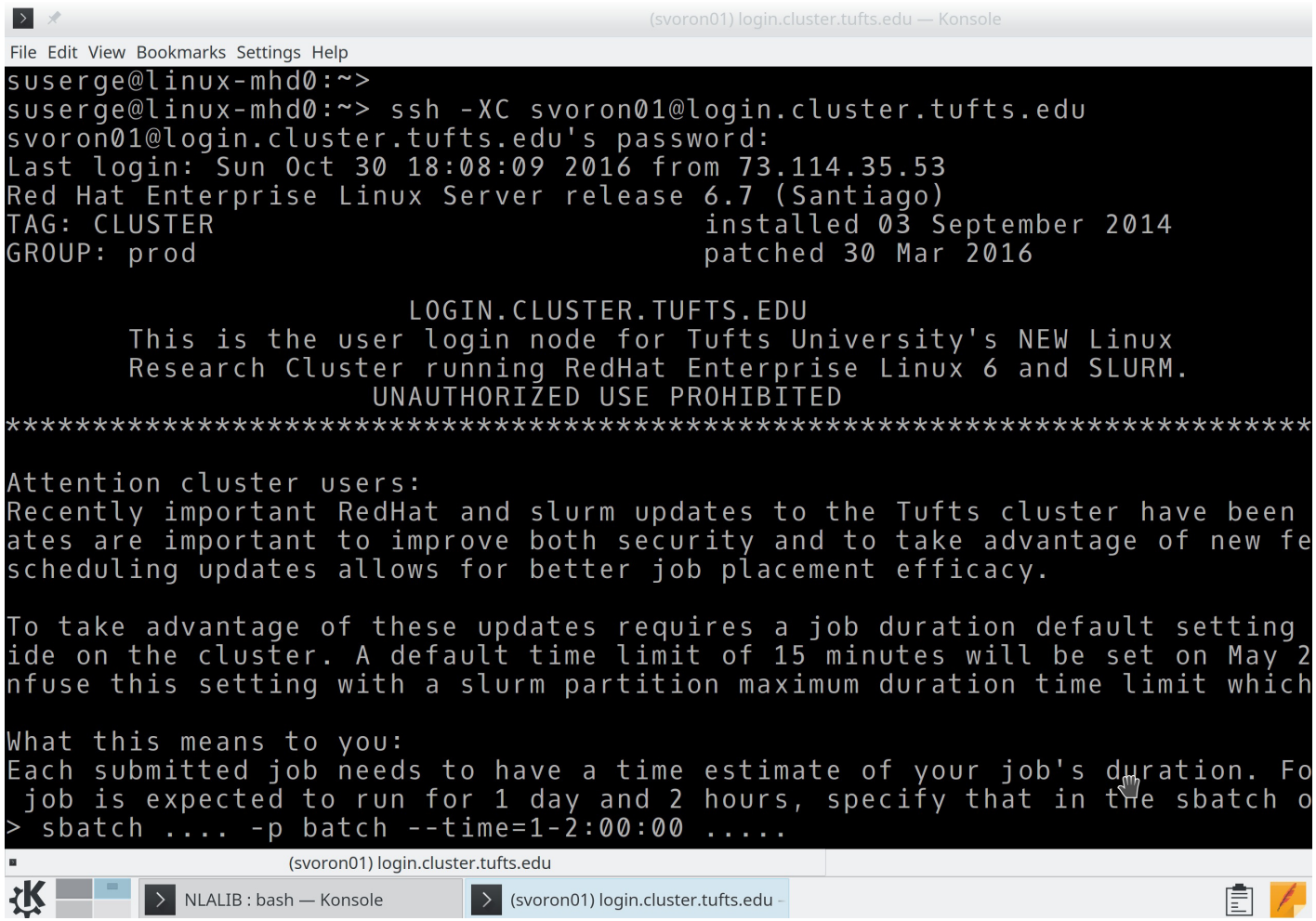


Using SSH with X11 on Windows 10



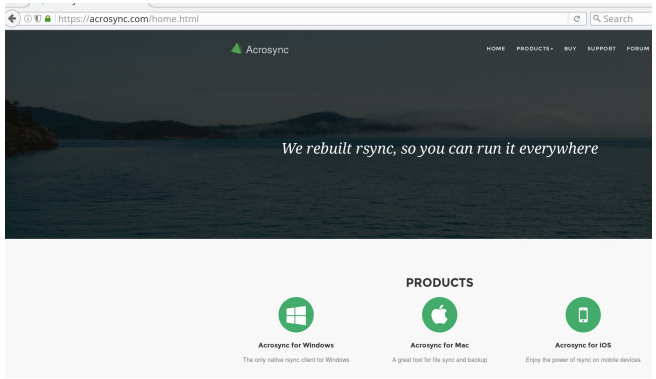
Linux and Mac

- In terminal: ssh -CX username @ server address
- '-C' option for compression
- '-X' option for X11 server forwarding



scp and rsync for file transfer

- Two related protocols to transfer data between client and server.
- scp to transfer large files (all at once).
- rsync to transfer incremental changes or for backups.



scp and rsync for file transfer

- Similar syntax to ssh, transfer file via ssh protocol.
- scp username@server:/path/to/file/on/server /local/path
- The '-r' (recursive) option is used to transfer directories.
- rsync can be used to transfer incremental changes (only transfers what has changed, useful for backups and code syncs). This can save a lot of time over scp transfer!
- Use **xfer node** for file transfers. Avoid the use of the login node for this.
- Command line examples on Linux and Mac:

```
svoronin@linux-mhd0:~> scp svoron01@xfer.cluster.tufts.edu:~/scripts/socket_script.pl .

svoronin@linux-mhd0:~> scp -r svoron01@xfer.cluster.tufts.edu:~/scripts/ .

svoronin@linux-mhd0:~> rsync -avp scripts/ svoron01@xfer.cluster.tufts.edu:~/scripts/
```

3. Slurm resource manager and examples

- Resource requests
- Interactive work and batch scripts
- Different partitions (interactive, batch, cpu)
- Examples with image denoising, matrix multiply.

SLURM

Simple Linux Utility for Resource Management (getting cluster resources).
(Command table credit: thanks to David Lapointe).

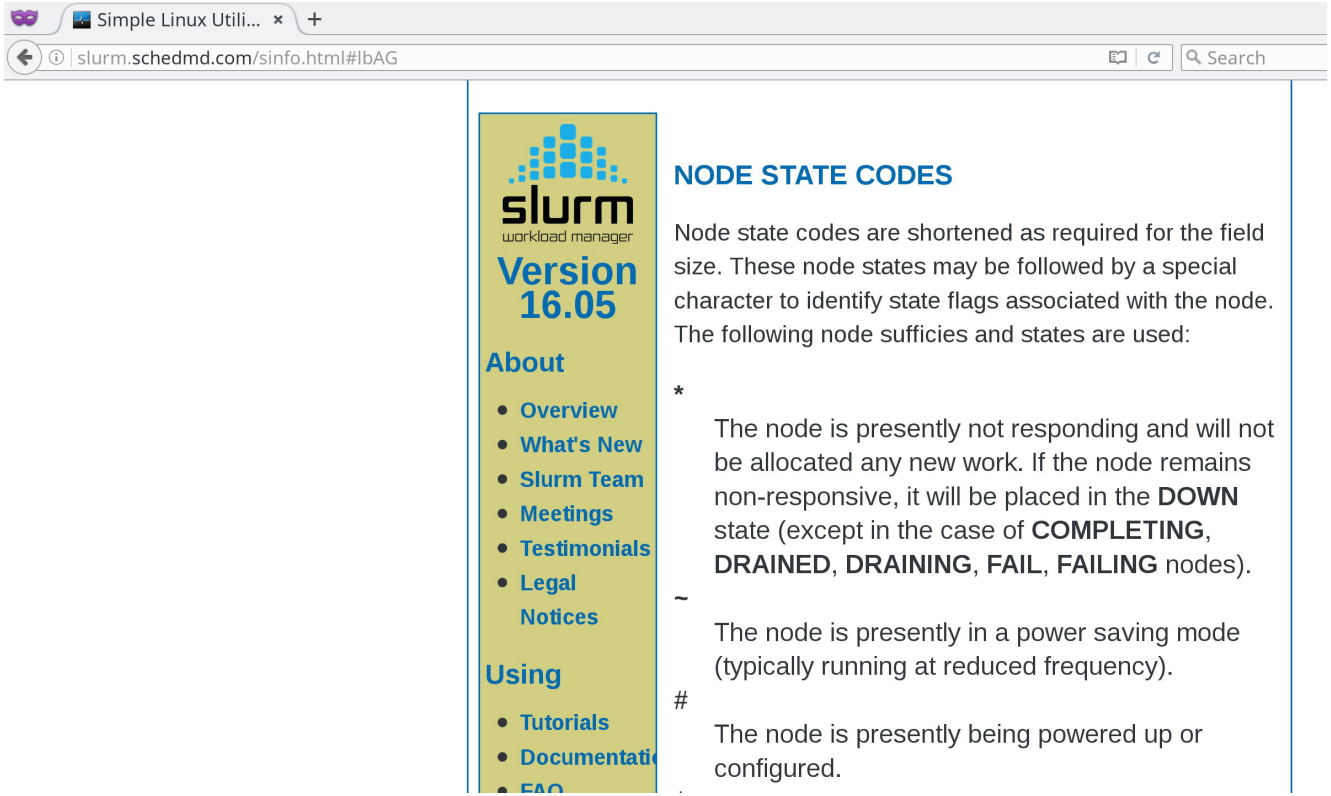
Action	Slurm
Show running jobs	squeue
Submit a batch job	sbatch
Submit with allocations	salloc
Start an interactive session	srun
List queues/partitions	squeue
List nodes	sinfo
Control running jobs	scontrol
Kill a job	scancel
User accounting	sreport or sacct
Other functions	srun

Available Partitions

```
[svoron01@login001 ~]$ sinfo
PARTITION   AVAIL  TIMELIMIT  NODES  STATE NODELIST
gpu            up 3-00:00:00      2   idle alpha025,omega025
largemem       up 7-00:00:00      4    mix alpha[022-023],omega[022-023]
interactive    up    4:00:00      1   comp alpha001
interactive    up    4:00:00      1    mix omega001
interactive    up    4:00:00      2   idle alpha025,omega025
batch*         up 3-00:00:00      1  down* alpha027
batch*         up 3-00:00:00      2  drain m3n[52,54]
batch*         up 3-00:00:00     31    mix alpha[002-008,018,022-024,028-030],m3n[22-23,43-45
,49-51],m4lmem01,omega[012,022-023,026-030]
batch*         up 3-00:00:00     26  alloc alpha[009-017,019-021,026,031],omega[002-011,024,0
31]
batch*         up 3-00:00:00     52   idle m3n[01-21,24-38,40-42,46-48,53],omega[013-021]
mpi            up 7-00:00:00      1  down* alpha027
mpi            up 7-00:00:00     17    mix alpha[003-008,018,024,028-030],omega[012,026-030]
mpi            up 7-00:00:00     25  alloc alpha[009-017,019-021,026,031],omega[003-011,024,0
31]
mpi            up 7-00:00:00      9   idle omega[013-021]
m4             up 7-00:00:00     59  alloc m4c[01-59]
m4             up 7-00:00:00      1   idle m4c60
[svoron01@login001 ~]$
```

The largemem nodes have 396 GB of memory.
The gpu nodes have an integrated gpu computational card.
The mpi nodes have fast interconnections.

Slurm node states



ALLOCATED: The node has been allocated to one or more jobs.
DRAINED: The node is unavailable for use per system administrator request.
DRAINING: The node is currently executing a job, but will not be allocated to additional jobs.

Node details

```
[svoron01@login001 ~]$ sinfo -N -l
Mon May  8 00:19:37 2017
NODELIST   NODES   PARTITION       STATE CPUS    S:C:T MEMORY

alpha001       1 interactive        idle   40   2:10:2 129054

alpha002       1     testing     drained   40   2:10:2 129054

alpha003       1         mpi       mixed   40   2:10:2 129054

alpha003       1      batch*       mixed   40   2:10:2 129054

alpha004       1         mpi       mixed   40   2:10:2 129054

alpha004       1      batch*       mixed   40   2:10:2 129054
```

```
[svoron01@login001 ~]$ sinfo -o "%15N %10c %10m  %25f %10G"
NODELIST        CPUS       MEMORY     FEATURES                  GRES
alpha025,omega0 40         128915     usnic,gpu                 gpu
alpha[001-021,0 40         128858+    usnic                     (null)
m3n[01-38,40-51 12         15937+     M3                        (null)
alpha[022-023,0 40         258130+    usnic,bigmem              (null)
m4lmem01        16         387581     (null)                    (null)
m4c[01-60]      16         32079+     M4                        (null)
[svoron01@login001 ~]$
```

Available software

The cluster has many different installed libraries and software packages. Run command: *module avail* to see available packages. The command *module load [package]* sets up the environmental variables for the use of package.

```
[svoron01@login001 ~]$ module avail

openmpi/1.8.6
OpenSees/r4985(default)
OpenSees/r4985-MP
paml/4.8
pandaseq/2.5(default)
paraview/3.8.1
paraview/4.4

python/2.7.6
python/3.5.0

R/3.2.2
R/3.2.5
R/3.3.2


tophat/2.0.9(default)
transdecoder/2.0
trinity/10.15.15
trinotate/2.0.2
ufc/2.3.0(default)
umfpack/5.5.1
usearch/7.0.1001(default)
VASP/4.6
VASP/5.4.1
vcftools/0.1.12b(default)
velvet/1.0.19
velvet/1.2.10
ViennaRNA/2.1.6(default)
visit/2.6.2
vmd/1.9
Wavelab/850
weka/3-6-10(default)
wpp/1.1-2008.0/optimize
wpp/1.1-2008.0/optimize-min
wpp/2.0-2010.0/optimize
wpp/2.0-2010.0/optimize-min
wpp/2.1.5
yap/5.1.3
[svoron01@login001 ~]$
```

Using Maple

Maple is full featured package and CAS for analytical and numerical analysis. We will use it for modeling pendulum motion.

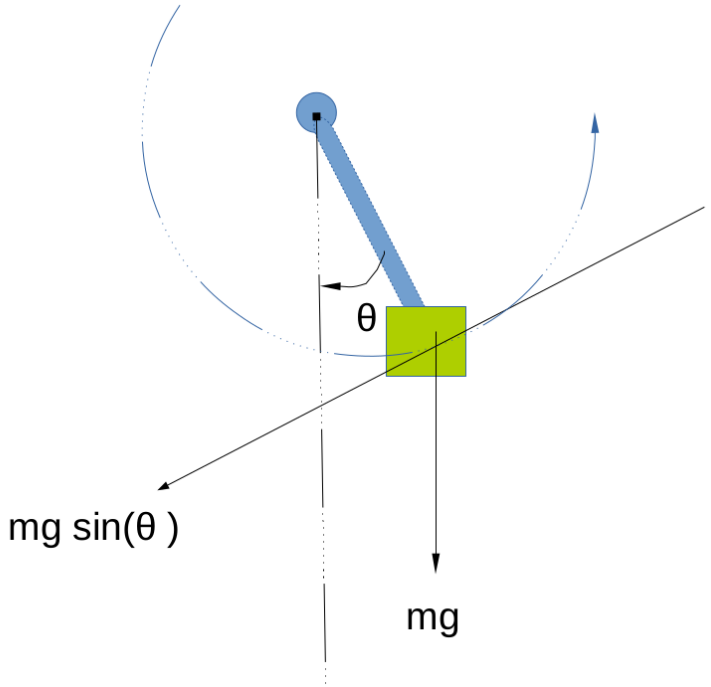
(1) salloc -N1 -c4 -t 100 -p interactive

(2) squeue -u username; ssh -XC granted_node

(3) module load maple

(4) $ xmaple &

Modeling pendulum motion



$$F = ma \implies -\alpha v - mg\sin(\theta) = ma \implies a = -\frac{\alpha}{m}v - g\sin(\theta) \qquad (1)$$

It remains to express the acceleration a in terms of l and θ. Let us take s to be the arc length along the circular direction of motion. Then we have that:

$$\frac{s}{2\pi l} = \frac{\theta}{2\pi} \implies s = l\theta \implies v = \frac{ds}{dt} = l\frac{d\theta}{dt} \implies a = \frac{d^2 s}{dt^2} = l\frac{d^2\theta}{dt^2} \qquad (2)$$

Plugging in for v and a in terms of θ, we obtain:

$$l\frac{d^2\theta}{dt^2} = -\frac{\alpha}{m}l\frac{d\theta}{dt} - g\sin(\theta) \implies \frac{d^2\theta}{dt^2} = -\frac{\alpha}{m}\frac{d\theta}{dt} - \frac{g}{l}\sin(\theta)$$

and the IVP:

$$\frac{d^2\theta}{dt^2} + \frac{\alpha}{m}\frac{d\theta}{dt} + \frac{g}{l}\sin(\theta) = 0 \quad ; \quad \theta(0) = \theta_0 \quad \text{and} \quad \frac{d\theta}{dt}(0) = v_0 \tag{3}$$

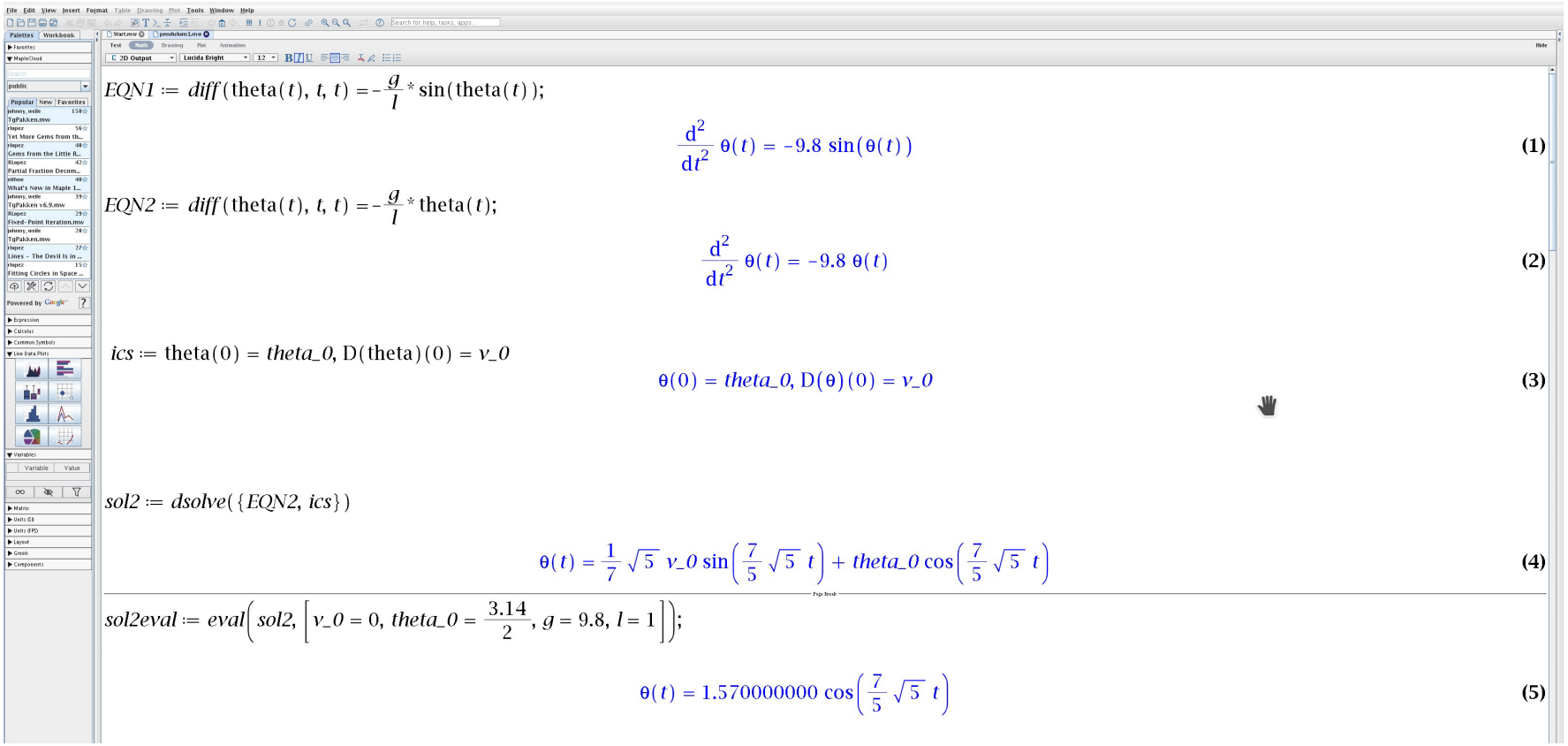
We first rewrite the IVP as a first order system, by defining $p = \theta$ and $q = \theta'$:

$$\begin{aligned} \frac{dp}{dt} &= q \\ \frac{dq}{dt} &= -\frac{\alpha}{m}q - \frac{g}{l}\sin(p) \\ p(0) &= \theta_0 \quad \text{and} \quad q(0) = v_0 \end{aligned}$$

Next, we rewrite the system as a single first order ODE of a vector valued function. Letting:

$$u(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} = \begin{bmatrix} p(t) \\ q(t) \end{bmatrix} \quad \text{and} \quad f(t) = \begin{bmatrix} u_2(t) \\ -\frac{\alpha}{m}u_2(t) - \frac{g}{l}\sin\left(u_1(t)\right) \end{bmatrix}$$

Modeling with maple



$$EQN1 := diff(\theta(t), t, t) = -\frac{g}{l} * \sin(\theta(t));$$

$$\frac{d^2}{dt^2}\theta(t) = -9.8\,\sin(\theta(t)) \quad (1)$$

$$EQN2 := diff(\theta(t), t, t) = -\frac{g}{l} * \theta(t);$$

$$\frac{d^2}{dt^2}\theta(t) = -9.8\,\theta(t) \quad (2)$$

$$ics := \theta(0) = theta\_0, D(\theta)(0) = v\_0$$

$$\theta(0) = theta\_0, D(\theta)(0) = v\_0 \quad (3)$$

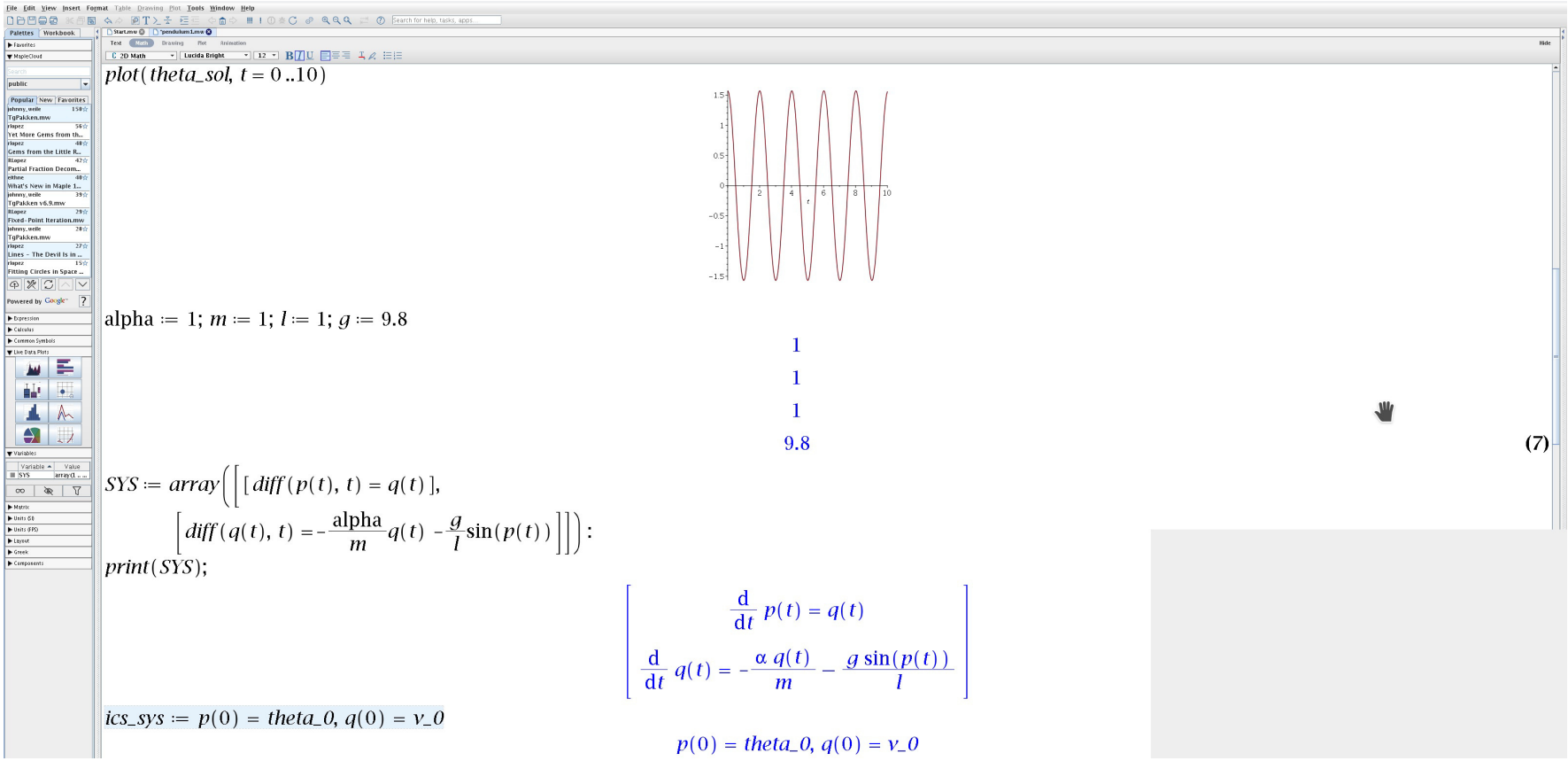
$$sol2 := dsolve(\{EQN2, ics\})$$

$$\theta(t) = \frac{1}{7}\sqrt{5}\ v\_0 \sin\left(\frac{7}{5}\sqrt{5}\ t\right) + theta\_0 \cos\left(\frac{7}{5}\sqrt{5}\ t\right) \quad (4)$$

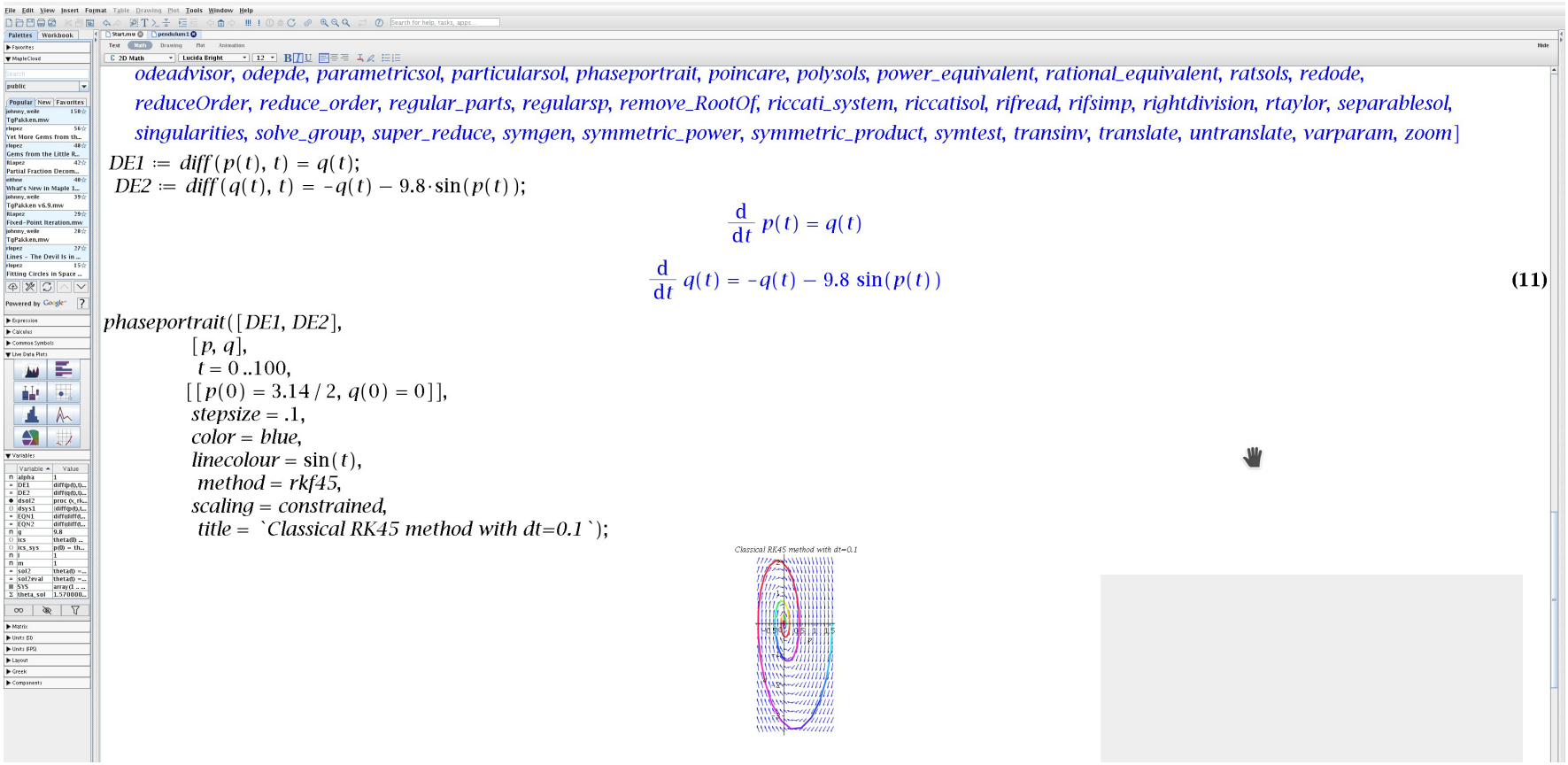
$$sol2eval := eval\left(sol2, \left[v\_0 = 0, theta\_0 = \frac{3.14}{2}, g = 9.8, l = 1\right]\right);$$

$$\theta(t) = 1.570000000 \cos\left(\frac{7}{5}\sqrt{5}\ t\right) \quad (5)$$

Modeling with maple



Modeling with maple



Using R, Python, Java

R is an open source package useful for statistical processing and data analysis. By installing some packages for R locally, we can use it for linear and non-linear model fitting. Example use:

(1) salloc -N1 -c4 –mem 30G -t 100 -p interactive (4 cores, 30 GB total mem, for 100 min on interactive partition)

(2) squeue -u username; ssh -XC granted_node

(3) module load R/3.3.2

(4) R

(5) $ > install.packages('class')

(6) module load python

(7) python2 test1.py

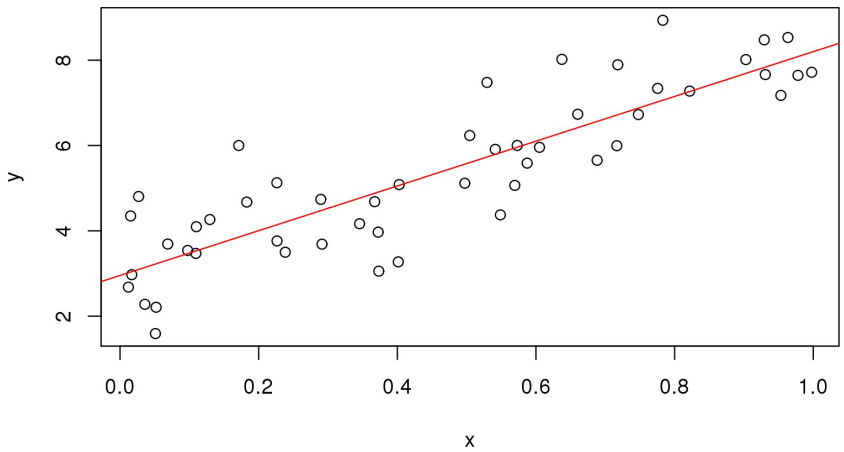
(8) module load java/1.8.0_60

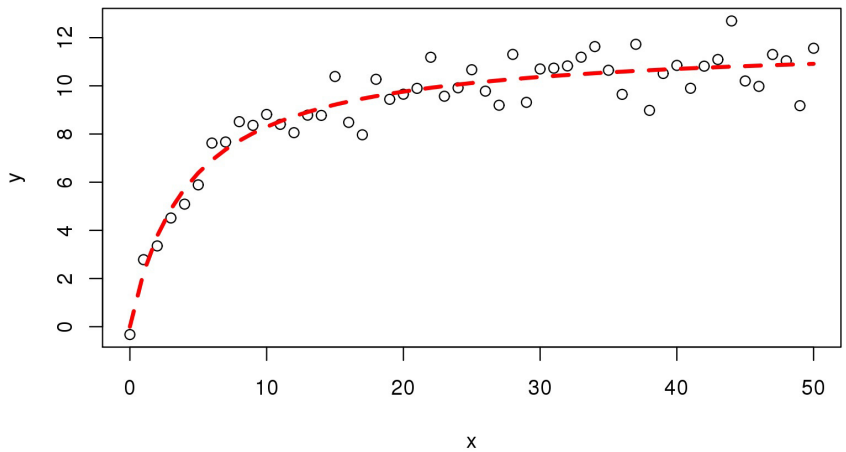
(9) javac runAlg.java ; java runAlg

Using R

Can use without or with X11 output (e.g. plotting).

```
[svoron01@alpha001 ~]$ module load R/3.3.2
[svoron01@alpha001 ~]$
[svoron01@alpha001 ~]$
[svoron01@alpha001 ~]$
[svoron01@alpha001 ~]$ R

R version 3.3.2 (2016-10-31) -- "Sincere Pumpkin Patch"
Copyright (C) 2016 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```





Using iPython

iPython is an interactive environment for testing and debugging your Python codes (similar to a Matlab/Octave prompt). First, run:

$ pip2 install –user ipython

Then, call from .local subdirectory:

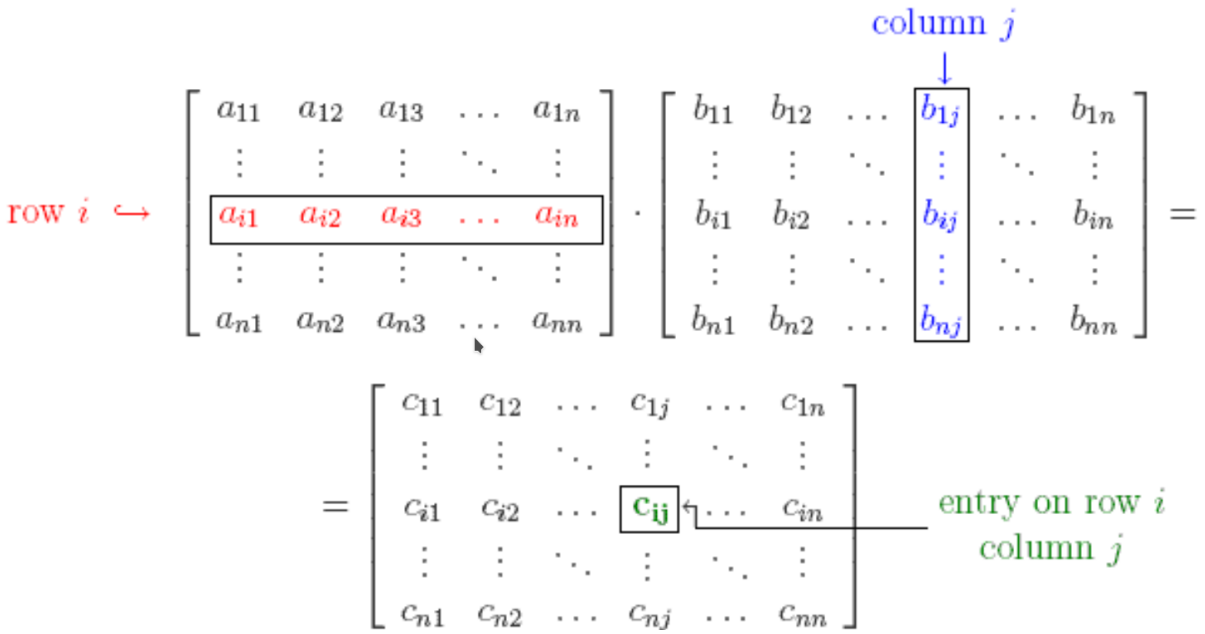
```
[svoron01@alpha001 ipython-5.3.0]$ ~/.local/bin/ipython2
Python 2.7.6 (default, Jan 14 2014, 09:37:27)
Type "copyright", "credits" or "license" for more information.

IPython 5.3.0 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.

In [1]:

In [1]: import numpy as np

In [2]:

In [2]: np.random.uniform(-1,1,4)
Out[2]: array([-0.22273188, -0.04487004,  0.93704455,  0.90914968])

In [3]:
```

Using Java

Can use Java with source code or (sometimes with GUI interface)

```
[svoron01@login001 ~]$
[svoron01@login001 ~]$
[svoron01@login001 ~]$
[svoron01@login001 ~]$
[svoron01@login001 ~]$ ssh -XC alpha001
Last login: Mon May  8 12:03:56 2017 from login001.lux.tufts.edu
[svoron01@alpha001 ~]$
[svoron01@alpha001 ~]$ cd storage/machine_learning_workshop/weka-3-8-1
[svoron01@alpha001 weka-3-8-1]$
[svoron01@alpha001 weka-3-8-1]$ module load java/1.8.0_60
[svoron01@alpha001 weka-3-8-1]$
[svoron01@alpha001 weka-3-8-1]$ java -jar weka.jar &
[1] 40795
[svoron01@alpha001 weka-3-8-1]$
```

Matrix multiplication in batch mode



Using sbatch

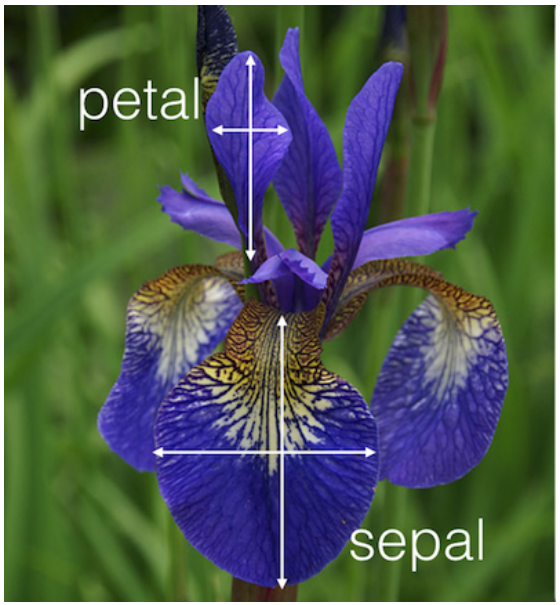
Often, (time-limited) interactive use is not necessary. This applies especially to developed code one wishes to run on the cluster. In this case, the batch interface is used.

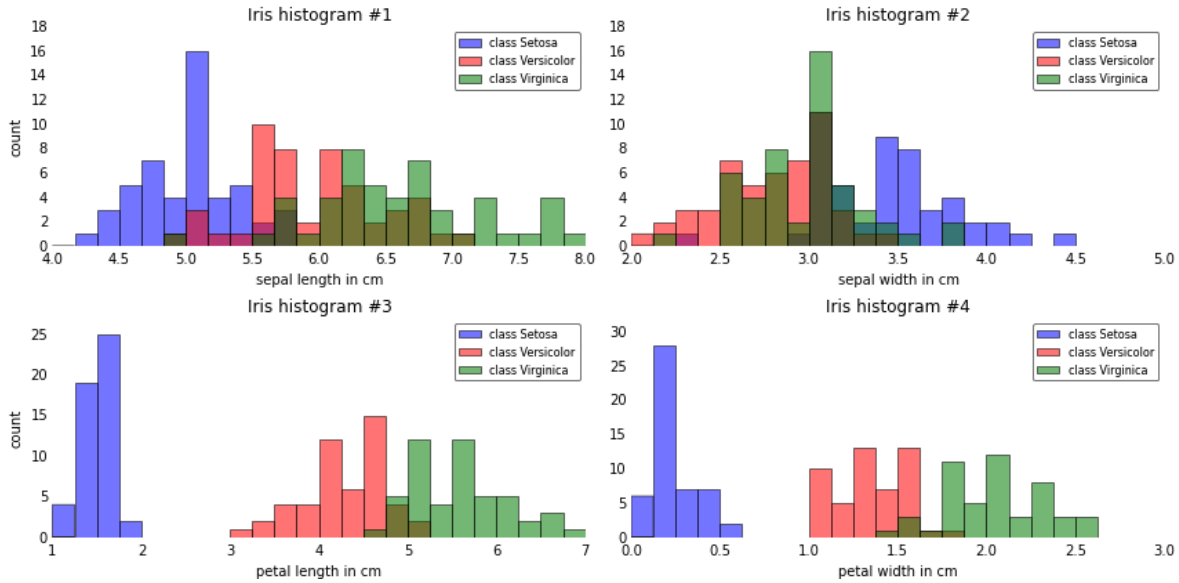
```
#!/bin/bash
#SBATCH --partition=gpu
#SBATCH -c 4
#SBATCH --mem-per-cpu=4000
#SBATCH --time 00:30:00
#SBATCH --output=mm_gpu.%N.%j.out
#SBATCH --error=mm_gpu.%N.%j.err
module load matlab
matlab -nodesktop -r "run('matrix_mult_gpu.m'); exit;"
```

Submit the above script using *sbatch batch_script.sh*. It will run the specified Matlab script for at most 30 minutes on a gpu parition, having access to 4 cores and 4 GB per core. Reported errors and output will go to the specified text files.

Using machine learning for classification

One of the main uses of machine learning is for clustering. Given, some training data, we would like to classify new instances.
Classic example: Iris flower data set





Clustering with R

```
> data(iris)
>
>
> names(iris)
[1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"  "Species"
```

```
> iris
    Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
1            5.1         3.5          1.4         0.2     setosa
2            4.9         3.0          1.4         0.2     setosa
3            4.7         3.2          1.3         0.2     setosa
4            4.6         3.1          1.5         0.2     setosa
5            5.0         3.6          1.4         0.2     setosa
6            5.4         3.9          1.7         0.4     setosa
7            4.6         3.4          1.4         0.3     setosa
8            5.0         3.4          1.5         0.2     setosa
9            4.4         2.9          1.4         0.2     setosa
10           4.9         3.1          1.5         0.1     setosa
11           5.4         3.7          1.5         0.2     setosa
12           4.8         3.4          1.6         0.2     setosa
```

```
> n <- nrow(iris);
> train <- sort(sample(1:n, floor(n/2)))
>
> train
 [1]   5  12  15  16  17  19  21  22  23  24  25  27  28  29  30  36  38  41  42
[20]  43  46  47  49  50  53  55  57  58  59  61  62  63  65  67  68  70  71  82
[39]  83  84  86  87  88  91  92  94  97  98  99 103 104 105 110 111 114 115 117
[58] 118 119 120 121 122 123 124 126 128 130 132 133 135 136 140 141 142 146
>
> n
[1] 150
>
> length(train)
[1] 75
>
> iris.train <- iris[train,]
> iris.test <- iris[-train,]
```

Using KNN

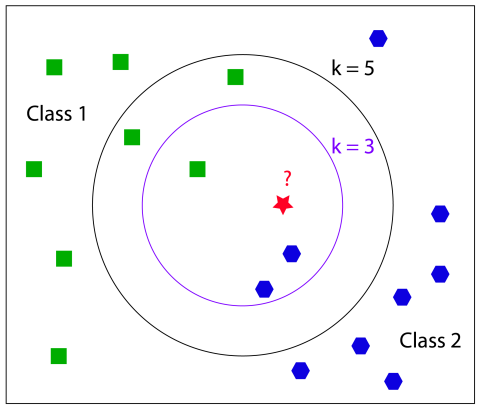
The K nearest neighbor algorithm uses the minimum distance between the test instance (the one we would like to classify) to the training samples to determine the K nearest neighbors. After the K nearest neighbors are determined, one takes the simple majority of these K nearest neighbors to be the prediction of the test instance.

Distance functions

Euclidean $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$

Manhattan $\sum_{i=1}^{k}|x_i - y_i|$

Minkowski $\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$




```
test_pred <- knn(train = iris.train,
    test = iris.test, cl = train_labels, k = 2);
```

```
> source('test_iris1.R')
[1] 75
[1] "results:\n"
[1] "classifcation accuracy:"
[1] 0.9733333
>
```

Disease detection

Important to scale features to a similar range. Goodness of high accuracy is problem dependent. Look at confusion matrix.

```
> names(wbcd)
 [1] "diagnosis"          "radius_mean"        "texture_mean"
 [4] "perimeter_mean"     "area_mean"          "smoothness_mean"
 [7] "compactness_mean"   "concavity_mean"     "points_mean"
[10] "symmetry_mean"      "dimension_mean"     "radius_se"
[13] "texture_se"         "perimeter_se"       "area_se"
[16] "smoothness_se"      "compactness_se"     "concavity_se"
[19] "points_se"          "symmetry_se"        "dimension_se"
[22] "radius_worst"       "texture_worst"      "perimeter_worst"
[25] "area_worst"         "smoothness_worst"   "compactness_worst"
[28] "concavity_worst"    "points_worst"       "symmetry_worst"
[31] "dimension_worst"
>
```

```
 $ perimeter_worst  : num  87 78.3 79.9 76.5 104.5 ...
 $ area_worst       : num  549 425 471 434 819 ...
 $ smoothness_worst : num  0.139 0.121 0.137 0.137 0.113 ...
 $ compactness_worst: num  0.127 0.252 0.148 0.182 0.174 ...
 $ concavity_worst  : num  0.1242 0.1916 0.1067 0.0867 0.1362 ...
 $ points_worst     : num  0.0939 0.0793 0.0743 0.0861 0.0818 ...
 $ symmetry_worst   : num  0.283 0.294 0.3 0.21 0.249 ...
 $ dimension_worst  : num  0.0677 0.0759 0.0788 0.0678 0.0677 ...
[1] "confusion matrix:"
    obs
pred  0  1
   0 61  2
   1  0 37
```

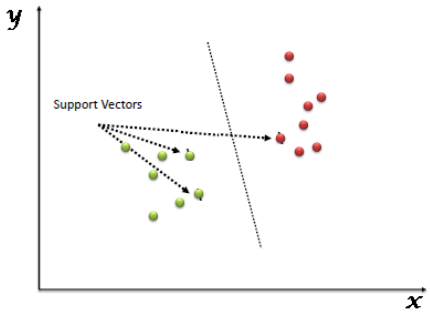
Analyzing the confusion matrix at different k

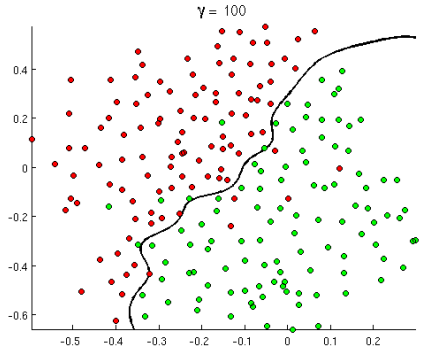
To see false positives and most importantly false negatives.

k value	False negatives	False positives	Percent classified incorrectly
1	1	3	4 percent
5	2	0	2 percent
11	3	0	3 percent
15	3	0	3 percent
21	2	0	2 percent
27	4	0	4 percent

Gender classification with support vector machines

If each point is plotted in n-dimensional space (where n is the number of features), then classification can be performed by finding a hyper-plane that differentiates two classes with respect to some cost function.






```
In [29]: run run_in_python.py
Python version: 2.7.6 (default, Jan 14 2014, 09:37:27)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-3)]
pandas version: 0.18.1
NumPy version: 1.12.1
SciPy version: 0.19.0
IPython version: 5.3.0
scikit-learn version: 0.18.1
Loading data..

classification accuracy: 88.62

confusion matrix:

(P)    0   1
(A) 0| 461  38
(A) 1| 81   466
```

Environmental sound data set

Ten classes of sound files (we will start with the first 5).

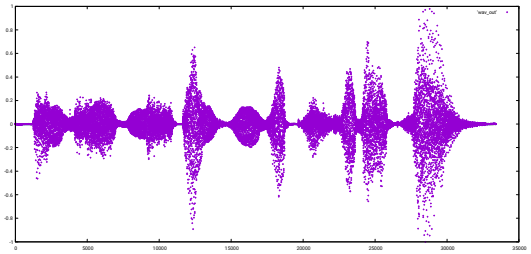
ESC-10: Dataset for Environmental Sound Classification

11 commits | **1** branch | **0** releases | **1** contributor

Branch: **master** | New pull request | Find file | Clone or download

karoldvl Update README.md — Latest commit 553c8f1 on Oct 8, 2015

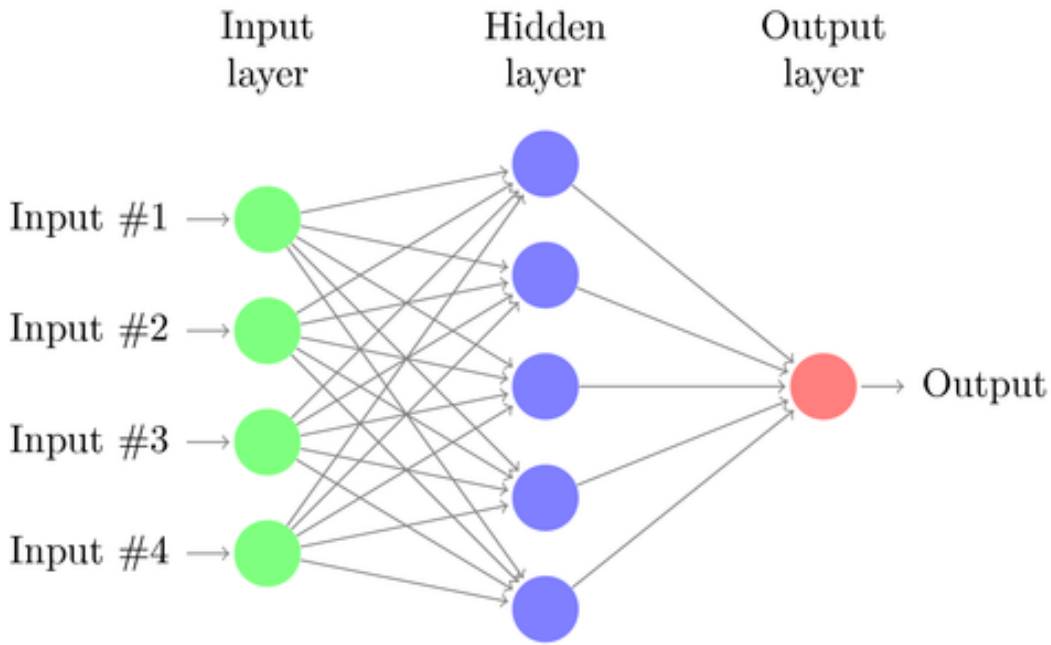
001 - Dog bark	Fix time alignment issues by exporting to Vorbis instead of MP3 (FFmp…	2 years ago
002 - Rain	Fix time alignment issues by exporting to Vorbis instead of MP3 (FFmp…	2 years ago
003 - Sea waves	Fix time alignment issues by exporting to Vorbis instead of MP3 (FFmp…	2 years ago
004 - Baby cry	Fix time alignment issues by exporting to Vorbis instead of MP3 (FFmp…	2 years ago
005 - Clock tick	Fix time alignment issues by exporting to Vorbis instead of MP3 (FFmp…	2 years ago
006 - Person sneeze	Fix time alignment issues by exporting to Vorbis instead of MP3 (FFmp…	2 years ago
007 - Helicopter	Fix time alignment issues by exporting to Vorbis instead of MP3 (FFmp…	2 years ago
008 - Chainsaw	Fix time alignment issues by exporting to Vorbis instead of MP3 (FFmp…	2 years ago
009 - Rooster	Fix time alignment issues by exporting to Vorbis instead of MP3 (FFmp…	2 years ago
010 - Fire crackling	Fix time alignment issues by exporting to Vorbis instead of MP3 (FFmp…	2 years ago



Extract features; normalize columns; split data:
extract_features.py normalize_data.py process_test_train_split.pl

Multi Layer Perceptron

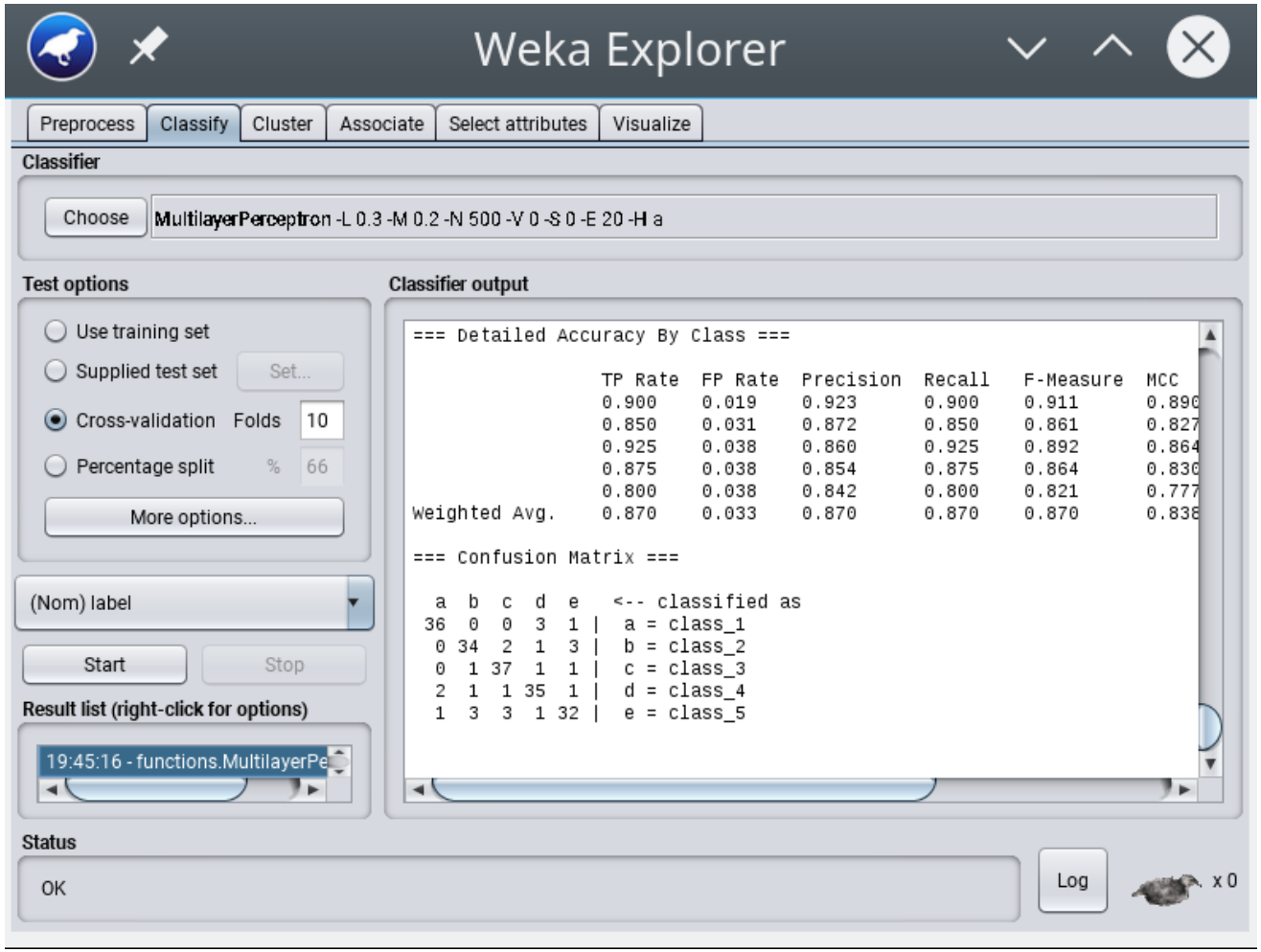
Each node in the Hidden layer is a function of the nodes in the previous layer, and the Output node is a function of the nodes in the Hidden layer.



We can test the algorithm with Java Weka engine (using either code or GUI). For this, we must convert our csv files to arff format.

Using Java Weka

```
module load java
java -jar weka-3-8-1/weka.jar &
```



Using Python

Scikit-learn and Tensorflow have a large collection of machine learning algorithms. To install the scikit-learn library use:

```
module load python/2.7.6
pip2 install --user scikit-learn
```

Usage from Python:

```
from sklearn.neural_network import MLPClassifier
mlp = MLPClassifier(solver='lbfgs',random_state=0,hidden_layer_sizes=[125]);
mlp.fit(X_train, y_train);
y_pred = mlp.predict(X_test)
print("mlp test set score: {:.2f}".format(np.mean(y_pred == y_test)))
```