Getting Started with the RCE

Len Wisniewski

IQSS

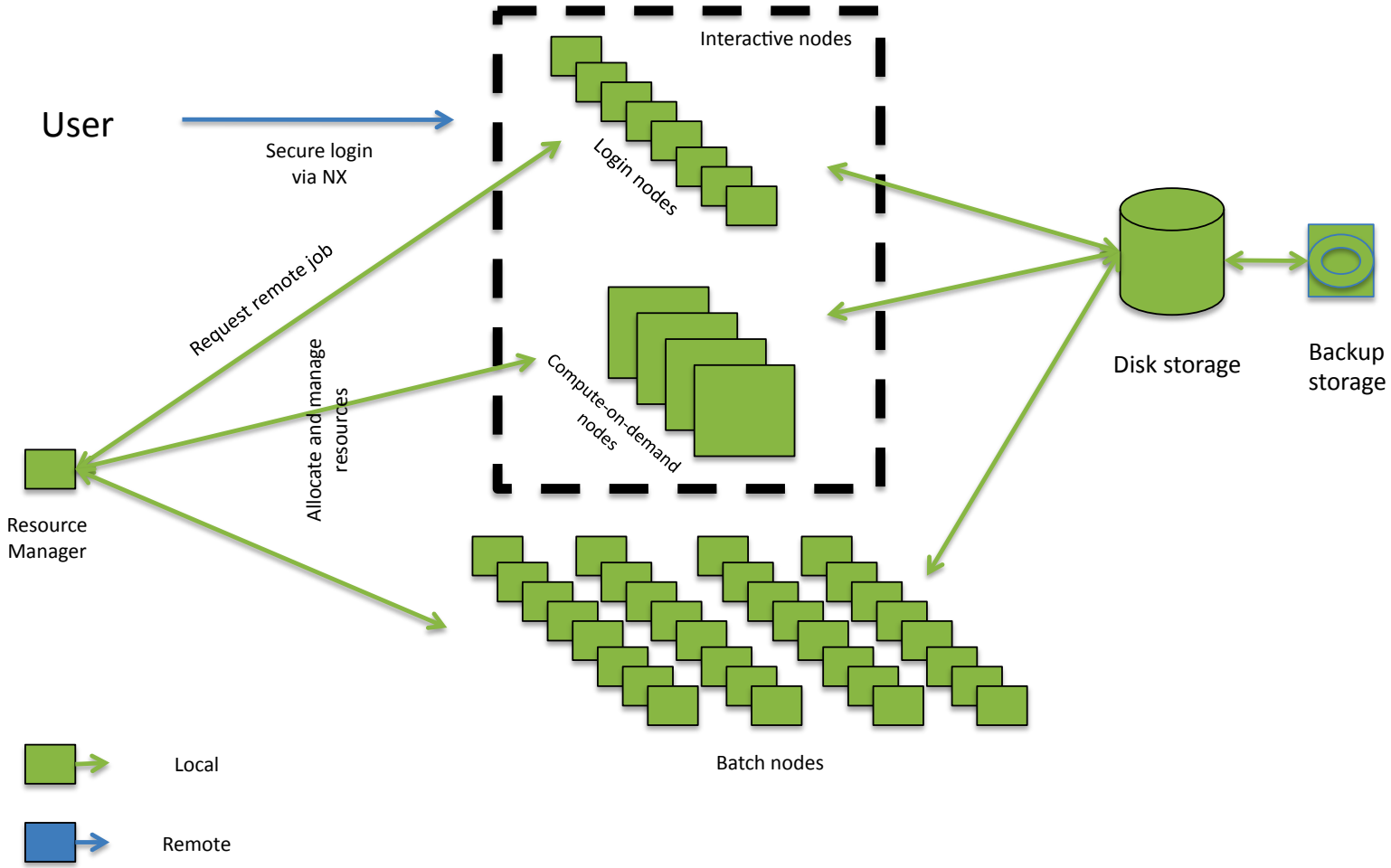
First thing to do

- Sign up for an RCE account
  - Send e-mail to support@help.hmdc.harvard.edu requesting an RCE account
  - Sys admins will send you a short questionnaire asking about your needs and sponsorship
  - If you don't have an IQSS faculty sponsor, contact me
    - lwisniewski@iq.harvard.edu

Research Computing Environment

- RCE = Research Computing Environment, remote infrastructure and scalable computing power for research analysis
- Used to run common statistical applications
  - R, GAUSS, Mathematica, MATLAB, Octave, SAS, S-PLUS, Stata
- RCE has three types of nodes
  - Login nodes
    - User logs in via NX (similar to VNC) and gets a desktop session
    - User can launch an application directly from the desktop
  - Compute-on-demand (COD) nodes
    - 25 compute cores on 9 nodes
    - User has special "RCE Powered Applications" menu to launch applications on machines with large memory resources (up to 64GB)
  - Batch nodes
    - 572 compute cores on 79 nodes
    - Used typically for non-interactive, long-running, scalable jobs
    - Most jobs use R

RCE architecture and configuration



Key Applications

R
GAUSS
Mathematica
MATLAB
Octave
SAS (COD only)
S-PLUS
Stata (SE and MP)

R
GAUSS
Mathematica
MATLAB
Octave
SAS
S-PLUS
Stata (SE and MP)

RCE – Current Metrics

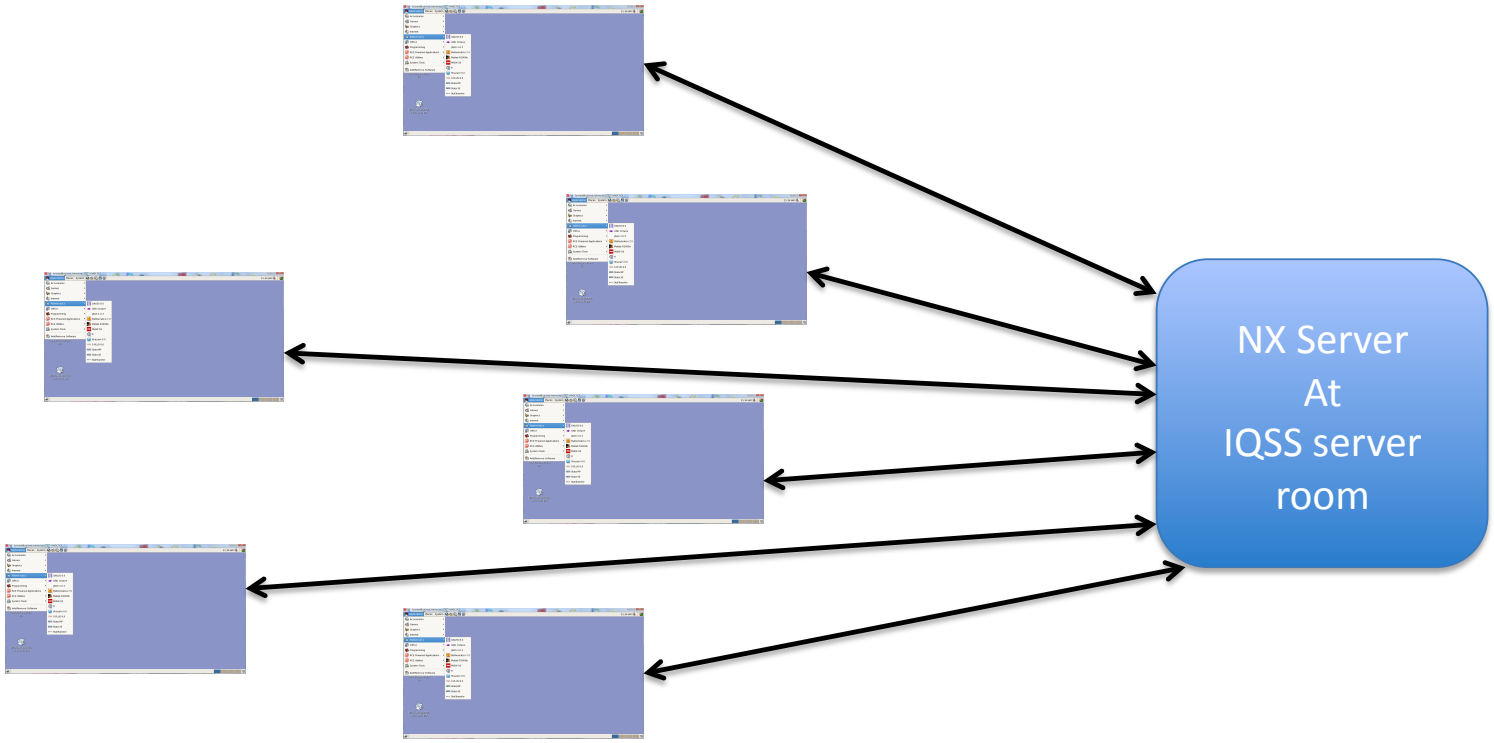
- 471 active user accounts
  - 105 HKS, 70 IQSS, 43 HSPH, 30 Gov, 26 GSE
- 3TB of data over 263 active "project spaces"
- 1.6 million hours of compute time on batch cluster
  - 10-20 active batch users per month
  - Average 50% utilization of batch pool
  - Pool includes 378 processor cores
- 100+ thousand hours of compute time on interactive cluster
  - 35-45 active interactive users per month
  - Average 80% utilization of interactive pool
    - Extended periods of overflow into batch
  - Pool has grown from 12 to 36 processor cores

The rest of this session

- Installing NX
- Logging onto an RCE login node
- Applications and tools on the RCE login nodes
- Running a job on a Compute-on-Demand (COD) node
  - COD = larger memory machines
- Running a job on a Batch node
  - Batch cluster = scalable set of compute resources

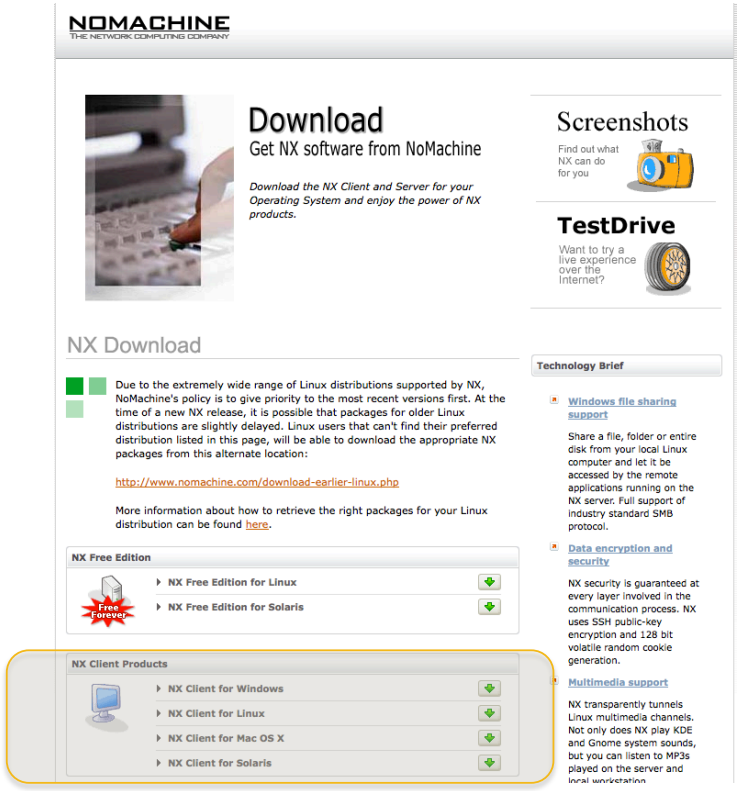
Installing NX

- NX is a computer program that provides secure remote windowing environment connections
  - Security encrypts communication over an SSH session



Step 1: Download NX client (Screen 1)

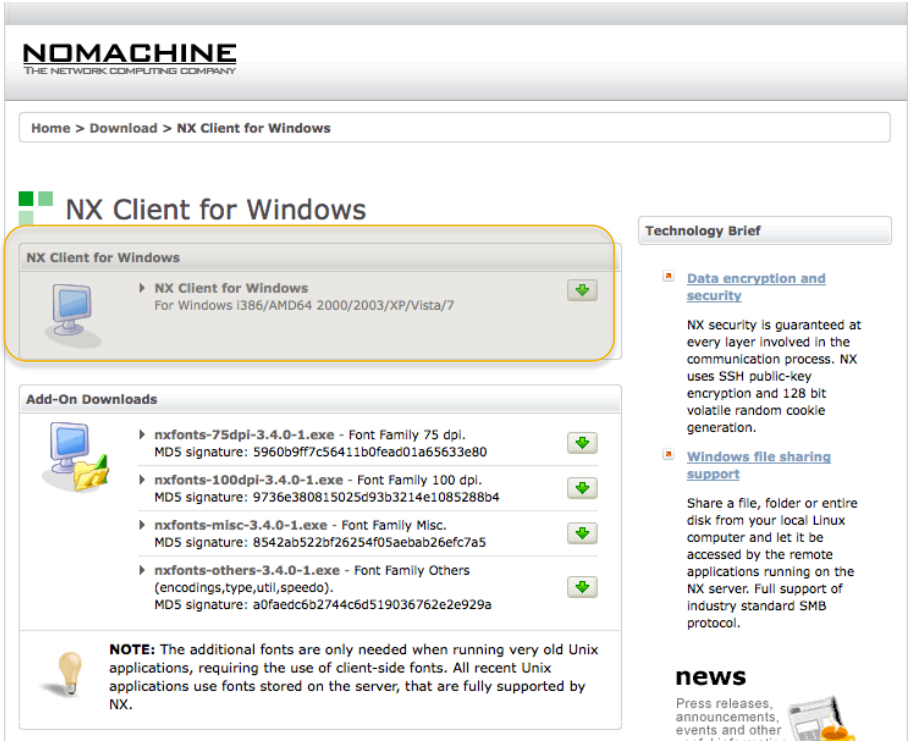
- Screen 1: Select the appropriate client for your OS
  - Windows, Linux, Mac OS X, Solaris
  - http://www.nomachine.com/download.php



Step 1: Download NX client (Screen 2)

- Screen 2: Select which package
  - http://www.nomachine.com/download.php
  - Windows: Download fonts later if you want them

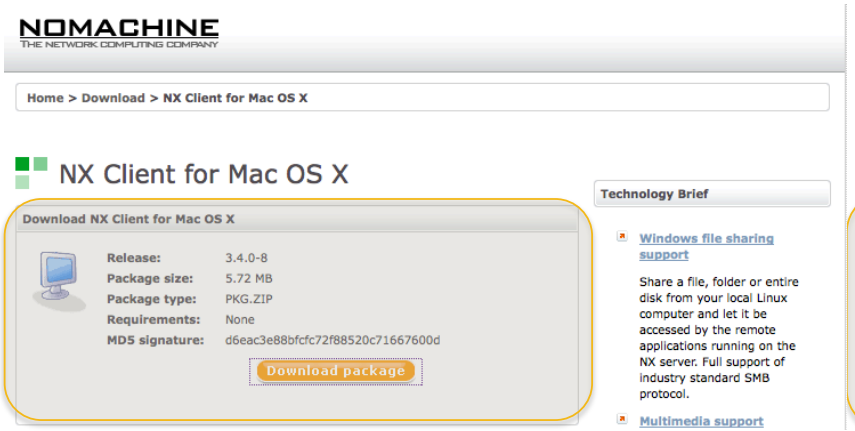
Mac OS

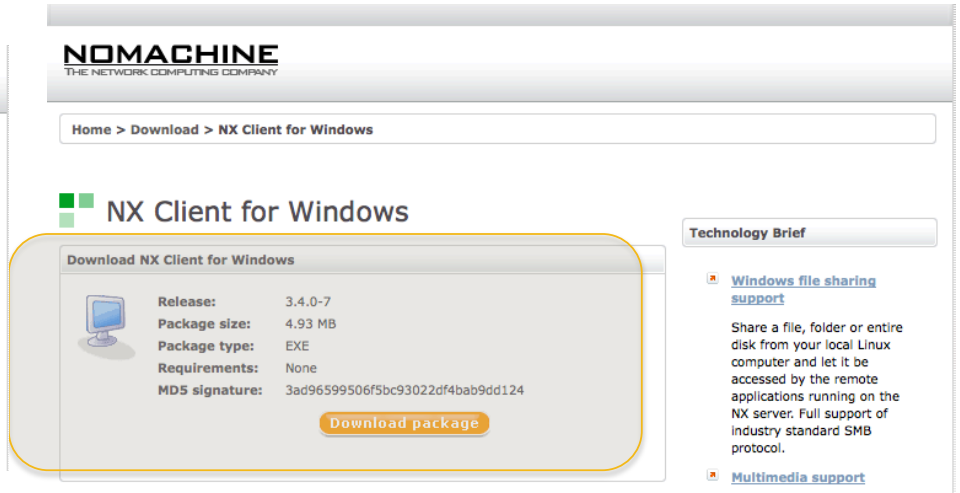


Windows

Step 1: Download NX client (Screen 3)

- Screen 3: Download the NX client package
  - http://www.nomachine.com/download.php



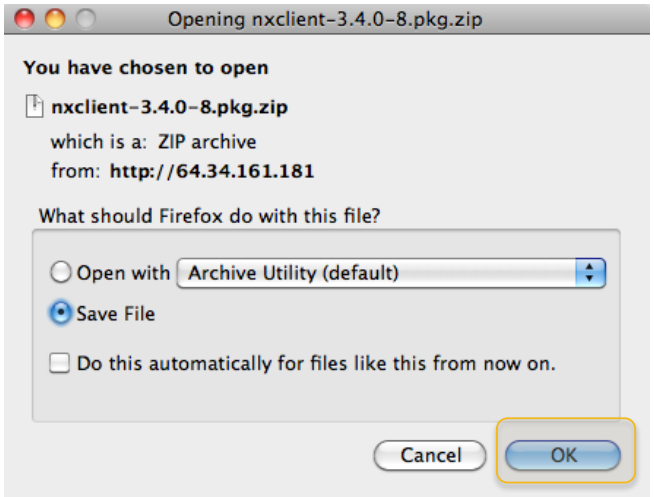


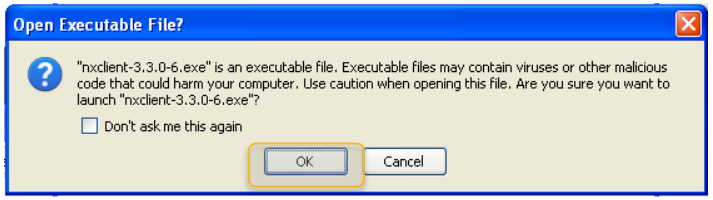
Mac OS

Windows

Step 1: Download NX client (Screen 4)

- Screen 4: Download the NX client package
  - http://www.nomachine.com/download.php



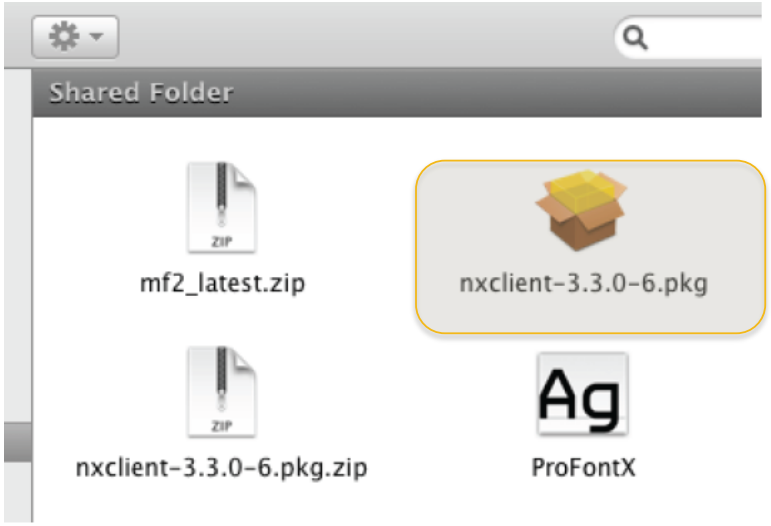


Mac OS

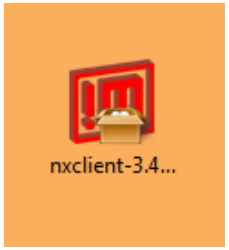
Windows

Step 2: Install NX client (Screen 1)

- Screen 1: Double-click the downloaded client



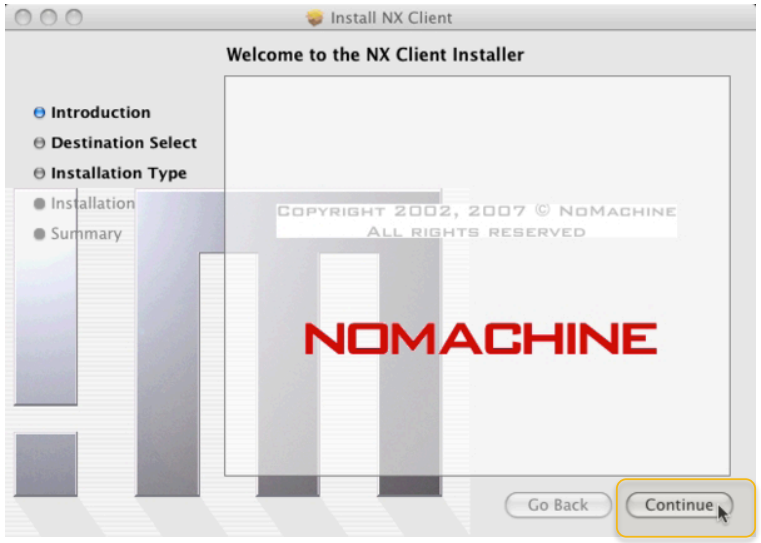
Mac OS



Windows

Step 2: Install NX client (Screen 2)

- Screen 2: Welcome screen

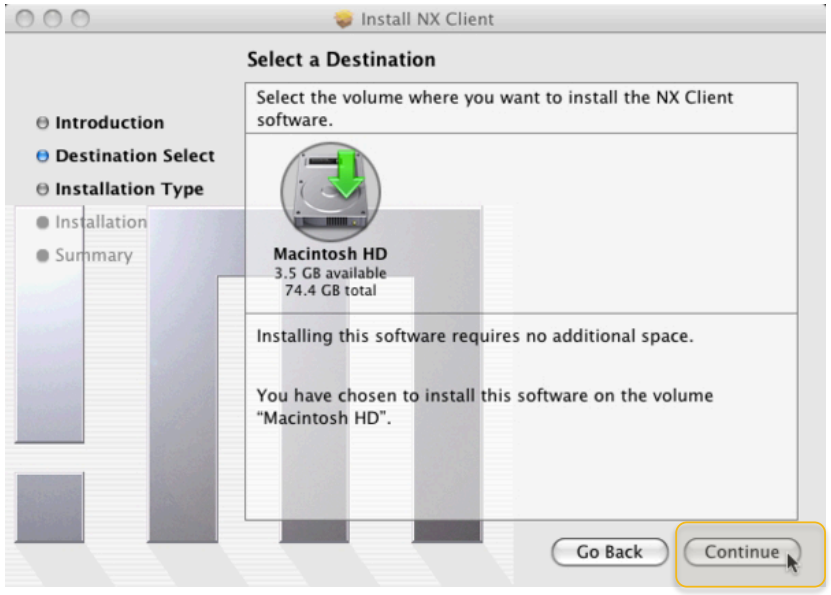


Mac OS

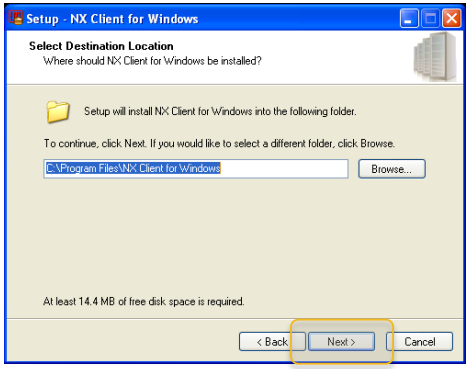
Windows

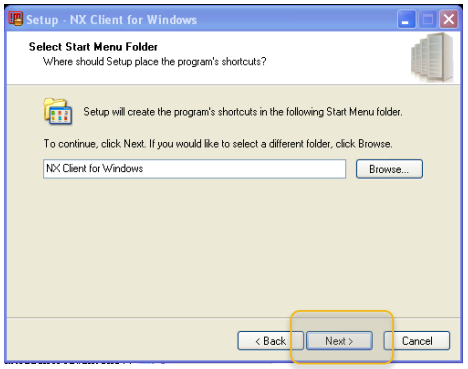
Step 2: Install NX client (Screen 3)

- Screen 3: Choose the location for the executable
- On Windows, there will be several extra screens



Mac OS

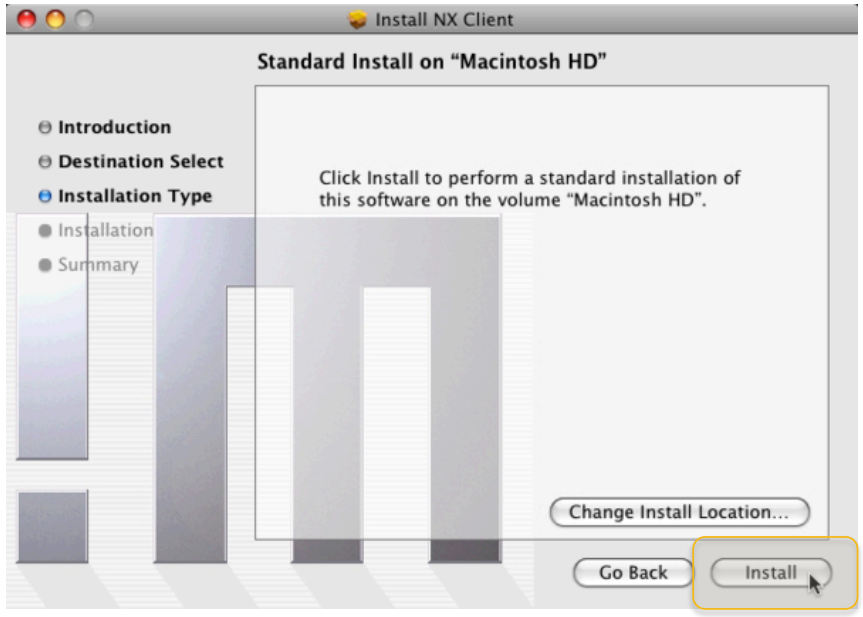




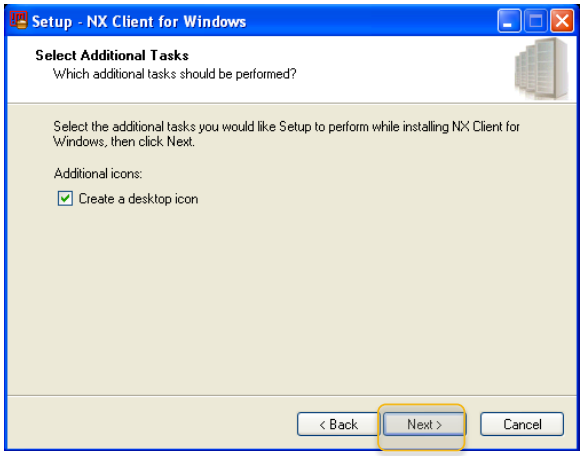
Windows

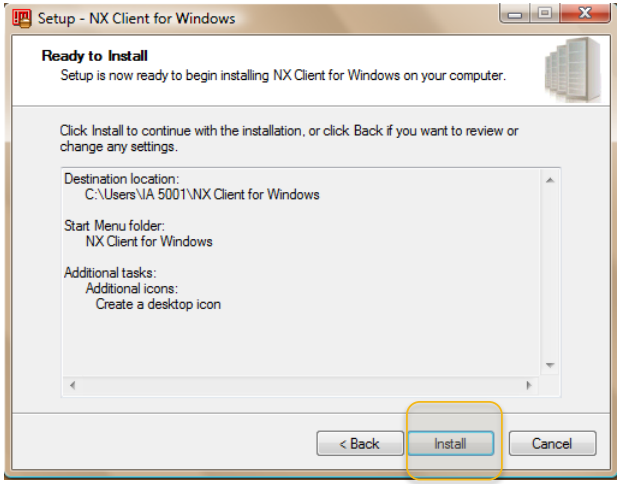
Step 2: Install NX client (Screen 4)

- Screen 4: Install NX



Mac OS

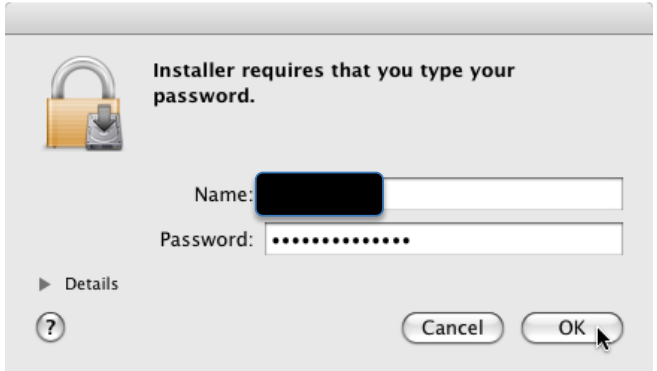




Windows

Step 2: Install NX client (Screen 5)

- Screen 5: Enter a local account and password, if necessary

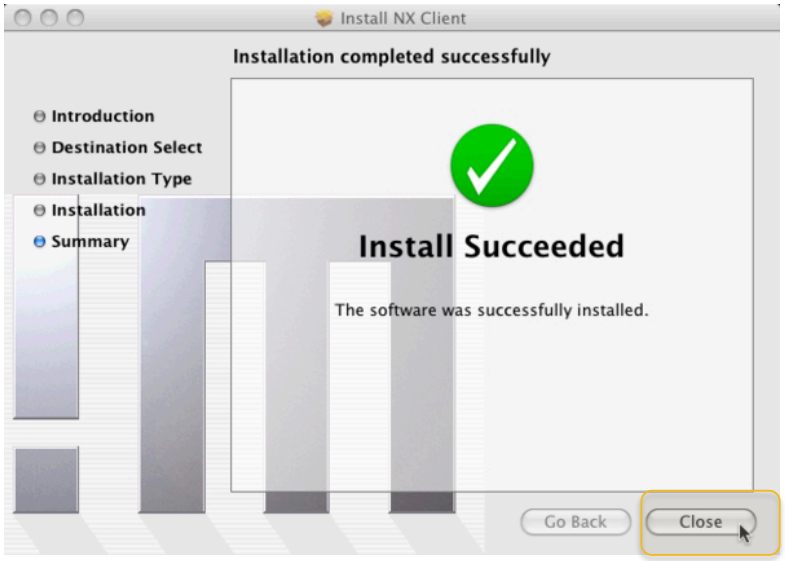


Mac OS

Windows

Step 2: Install NX client (Screen 6)

- Screen 1: Installation complete



Mac OS

Windows

Step 3: Configuring NX client

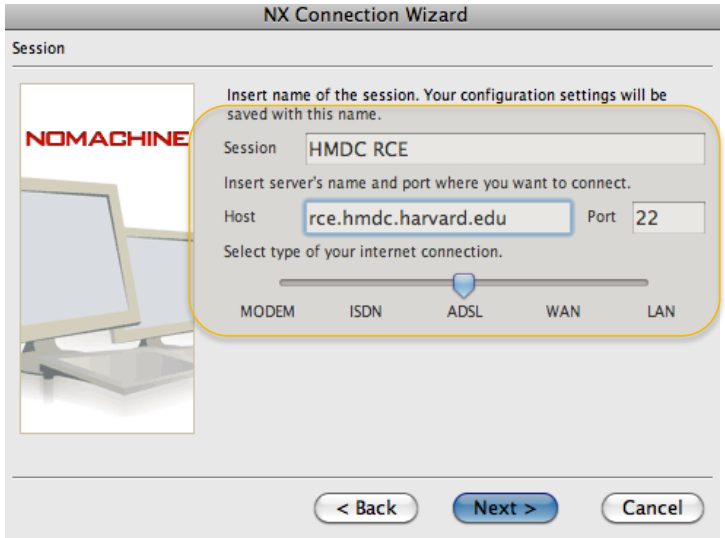
- Click on the NX Connection Wizard icon (on Mac) or NX Client for Windows icon (on Windows)

Mac OS

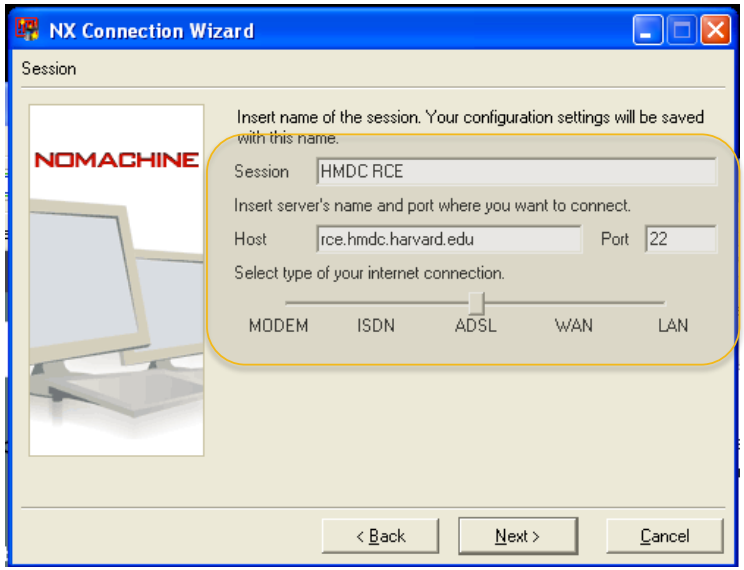
Windows

Step 3: Configuring NX client (Screen 1)

- After an initial welcome screen
- Screen 1: Enter a Session name (your choice), Host Internet address (rce.hmdc.harvard.edu), Port number (22, for SSL), and Internet connection type (ADSL)



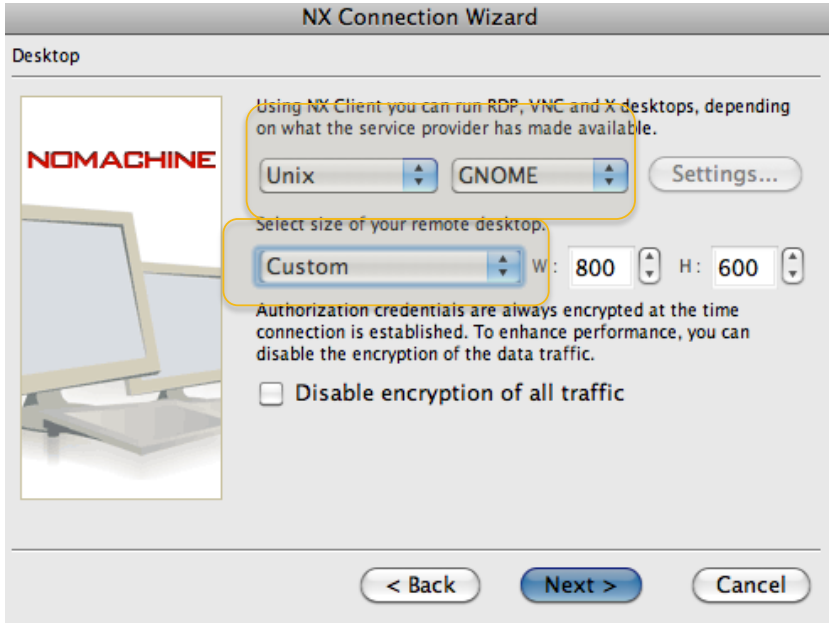
Mac OS



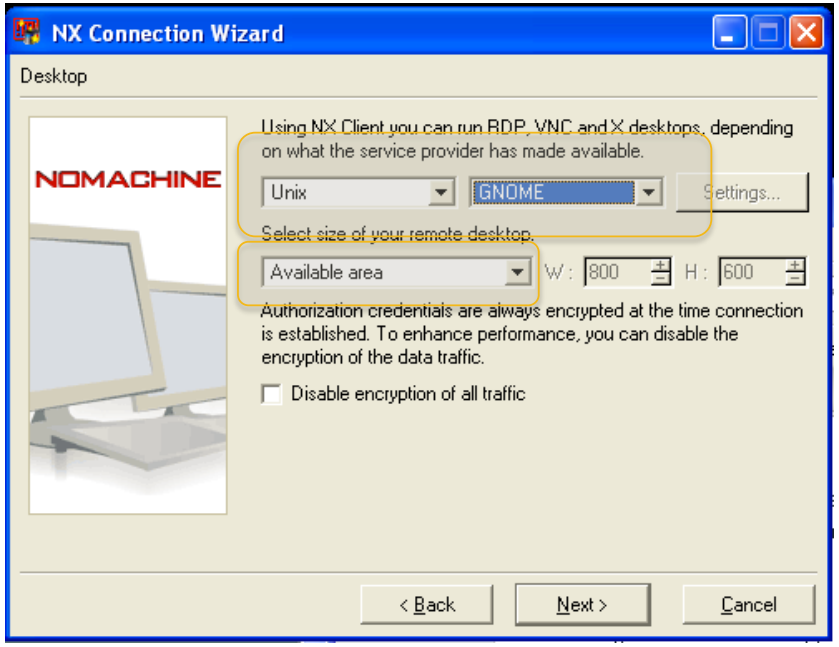
Windows

Step 3: Configuring NX client (Screen 2)

- Screen 2: Enter an OS type (Unix, because RCE runs on Linux), a desktop type (GNOME, the type supported by RCE's NX server), and a size (Custom for Mac, Available area for Windows)



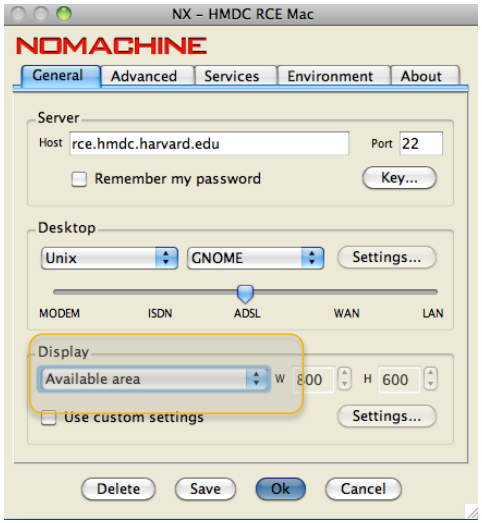
Mac OS



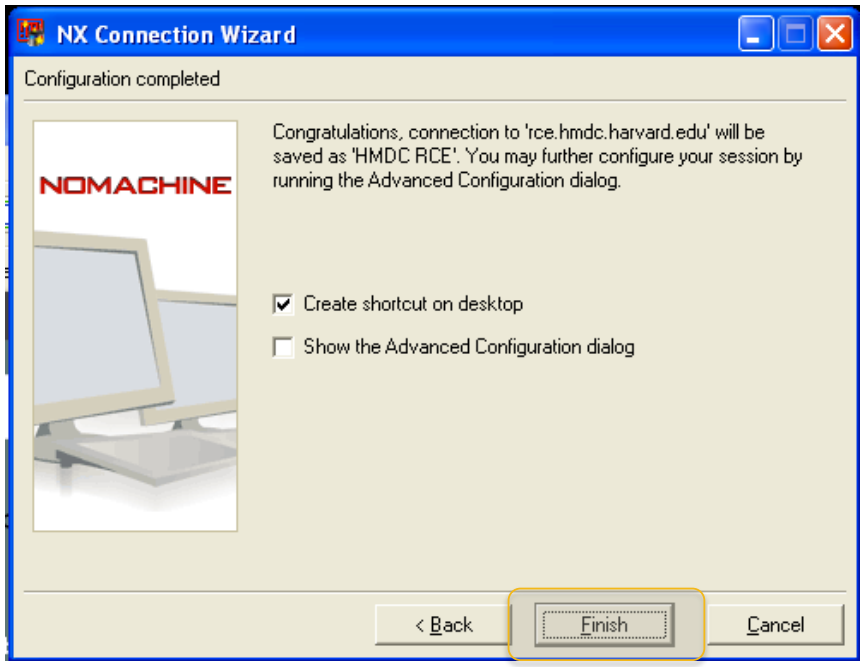
Windows

Step 3: Configuring NX client (Screen 3)

- Screen 3: Completion screen (Check Advanced Configuration on Mac)



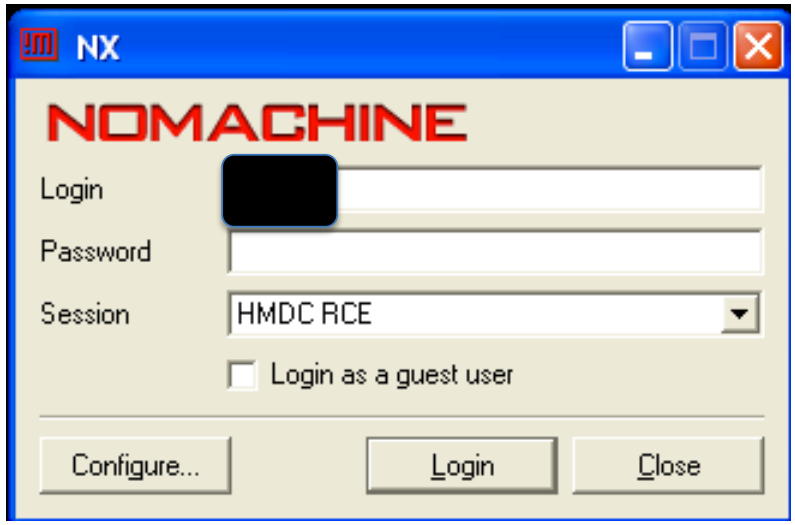
Mac OS



Windows

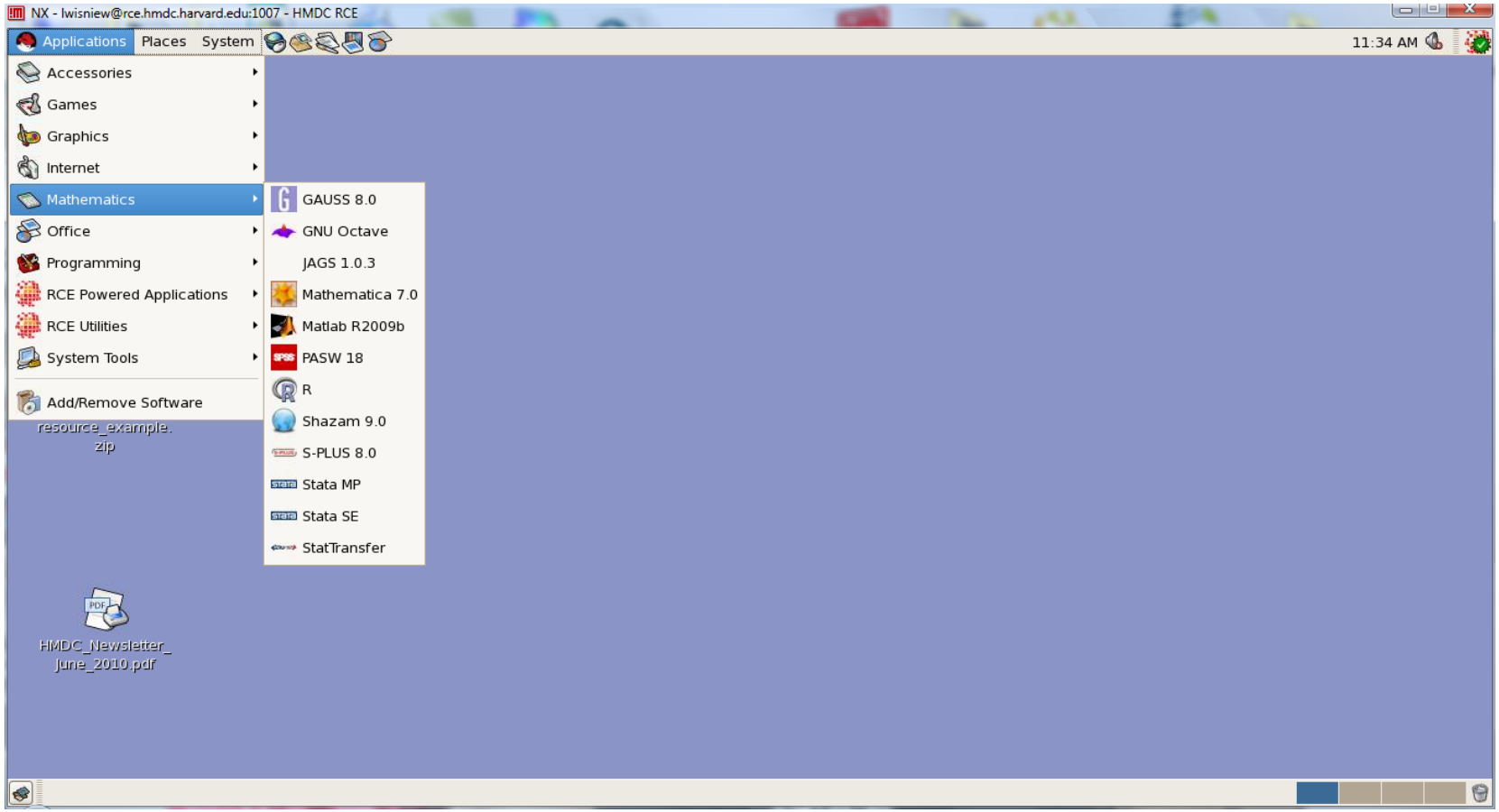
Logging onto an RCE login node

- Click on the desktop icon created when configuring NX
- Use your login username and password obtained when signing up for an RCE account



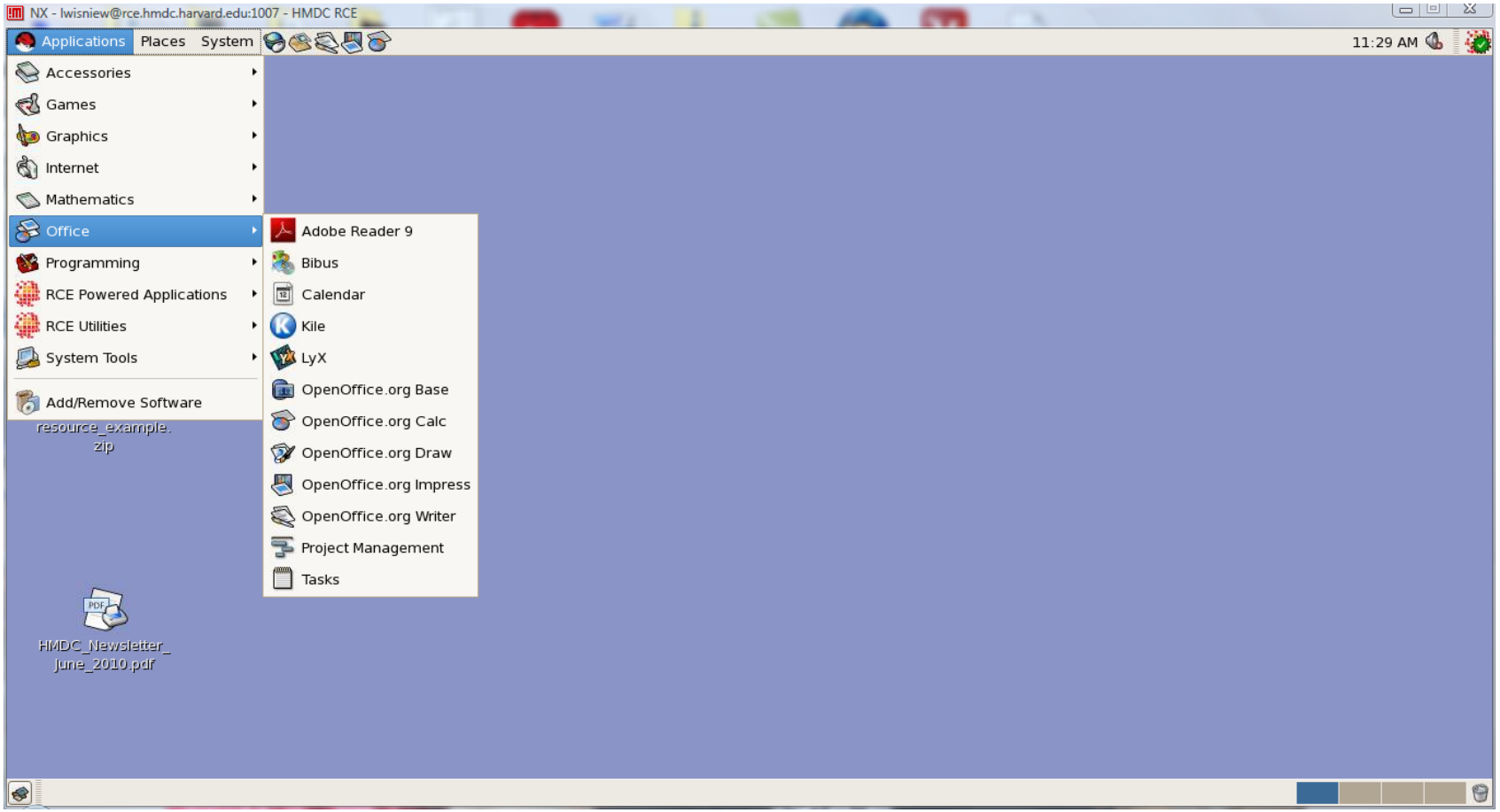
Applications and tools on the RCE login nodes

- Statistical applications



Applications and tools on the RCE login nodes

- Office productivity applications

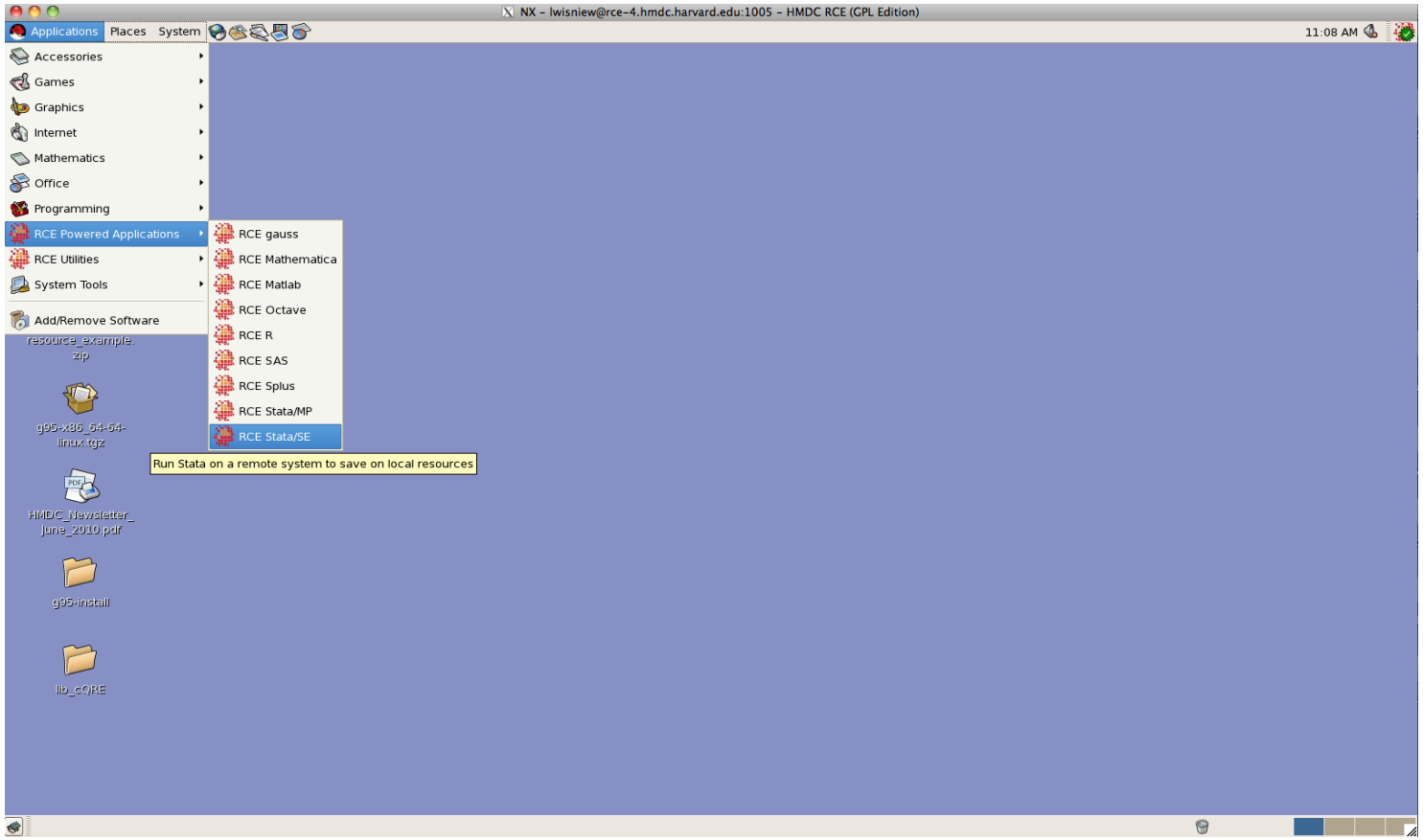


As your data set grows

- As your data set grows…
  - Your data set will require more memory
    - Run on machines with larger memory (e.g., COD)
  - Your data set may take much longer to compute
    - Take High Performance Computing in the Social Sciences class (Nov 30)
    - Attend RCE for Advanced Users sessions (starting in Spring)
      - Run on more lightly-loaded resources
      - Use more efficient algorithms
      - Parallelize your code

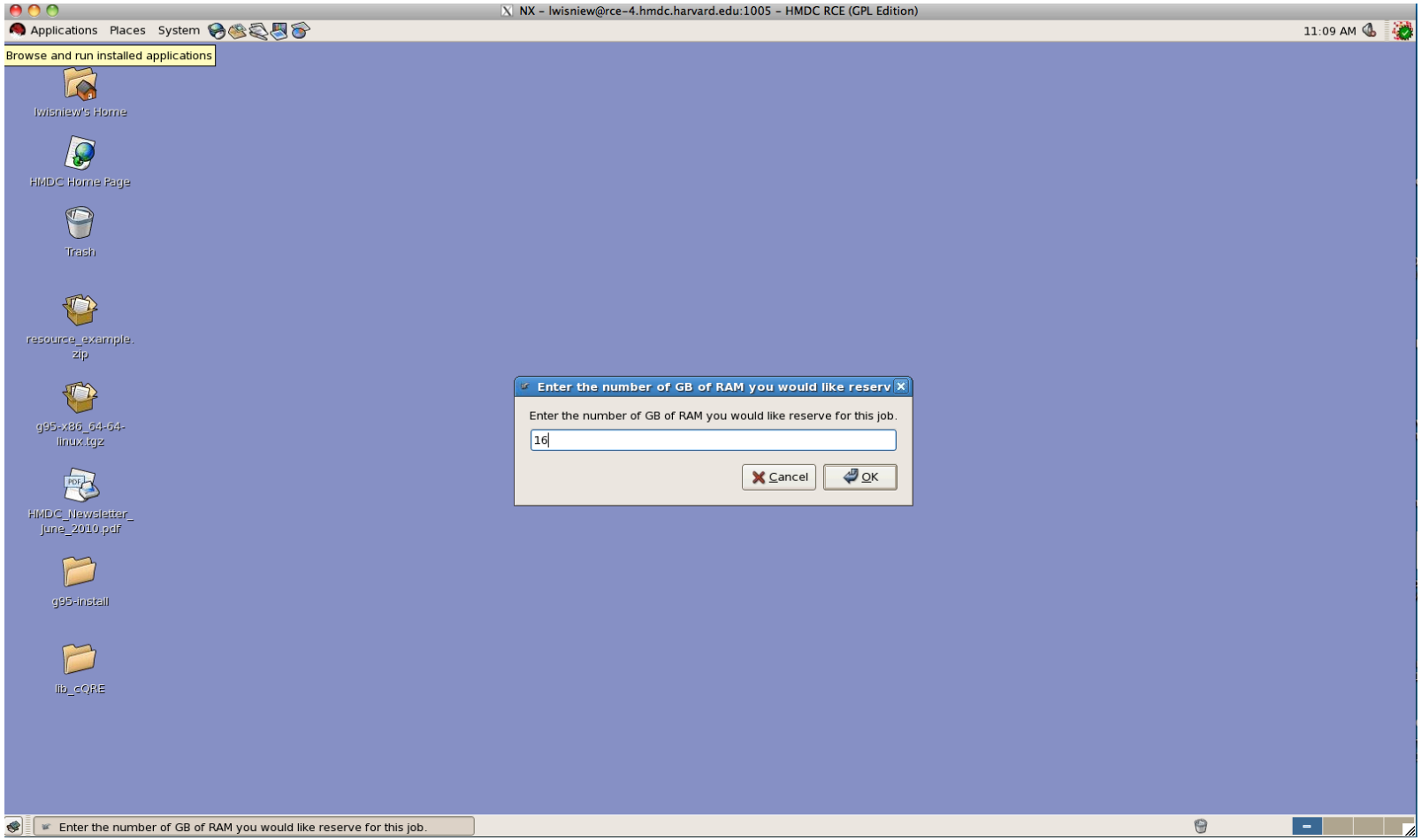
Running a job on a Compute-on-Demand (COD) node

- Running a job on a larger memory Compute-on-Demand (COD) node
- Step 1: Launch an RCE Powered Application



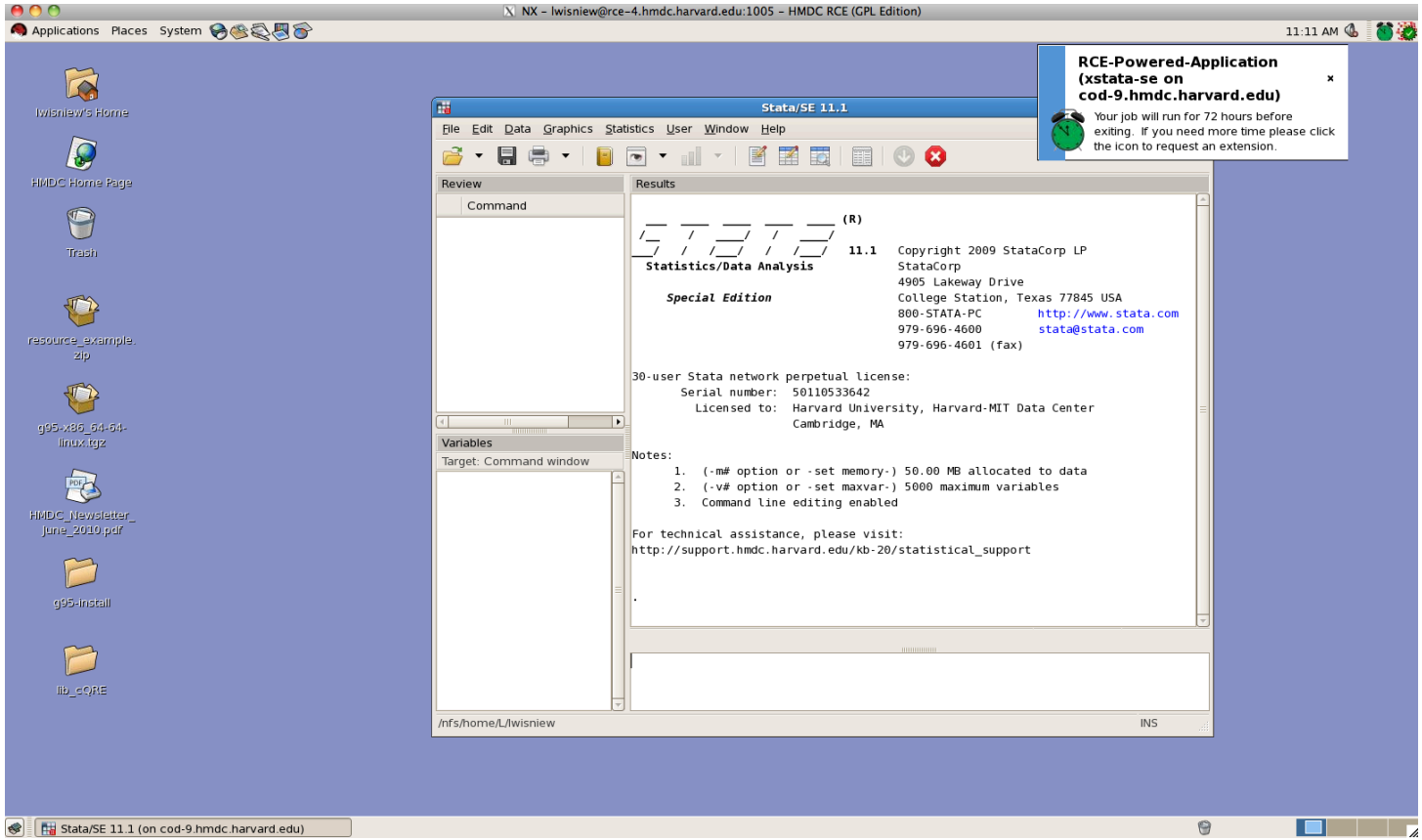
Running a job on a Compute-on-Demand (COD) node

- Step 2: Enter the memory requirement for the job



Running a job on a Compute-on-Demand (COD) node

- Step 3: The interactive job is running, a message appears about your time limit for completing your job



Running a job on a Batch node

- Batch environment
  - Batch system – jobs are submitted to a queue to be executed when resources matching the job's resource requirements are available
  - Batch cluster = scalable set of compute resources
  - Jobs are scheduled on the queue in a fair manner
  - Condor is the resource manager that manages the resources (and queues for those resources)
    - Open-source solution developed by University of Wisconsin
    - http://www.cs.wisc.edu/condor/
    - http://www.cs.wisc.edu/condor/manual/
  - Jobs submitted to a batch queue should not be interactive

Running a job on a Batch node

- Step 1: Create a directory with all necessary files to submit a job
  - e.g. input files, output files, log files, error files
- Step 2: Create a program or script that does not require user interaction
- Step 3: Create a condor_submit file, containing paths of input files, output files, executable file, and any other information needed to match your job to a resource and execute it successfully
  - Manually
  - Using the condor_submit_util script
- Step 4: Submit a job with the condor_submit command (or condor_submit_util)
  - When you submit a job, you are assigned a tag
    - <cluster number>.<process number>
  - Each submission by a user generates a cluster number
  - If you submit multiple runs of the same executable, then each of those runs is differentiated by a process number
  - Use this tag to track the progress of your job (see upcoming condor_q description)

Example of a condor_submit file

Submits 10 instances of an R job running <program>.R

```
> cat hosttest1.submit

    Universe = vanilla
    Executable = /usr/bin/R
    Arguments = --no-save --vanilla
    when_to_transfer_output = ON_EXIT_OR_EVICT
    transfer_output_files = out.$(PROCESS)

    input = <program>.R
    output = out.$(Process)
    error = error.$(Process)
    Log = log.$(Process)
    Queue 10
```

Other useful Condor commands

- Found in /usr/local/condor/bin and /usr/local/condor/sbin
  - condor_q
  - condor_status
  - condor_rm
  - condor_history

- If things don't run as planned, check out the error and log files in the submission directory for information about what happened.

Documentation

- http://support.hmdc.harvard.edu/kb-14/research_computing_environment
  - Accessing the RCE
    - Connecting to the RCE (via NX)
    - Working in the RCE (productivity tools)
  - Developing Software (CVS and R packaging tools)
  - Using Statistical Programs
    - Working with {R, GAUSS, Mathematica, MATLAB, Octave, SAS, S-PLUS, Stata, RCE Powered}
  - Recent Updates
  - RCE FAQ

- http://support.hmdc.harvard.edu/kb-13/cluster_computing
  - Working with Batch Servers
  - Working with Interactive Servers
  - Cluster Computing FAQ

Future directions

- Become more adaptive to resource needs
  - Single flexible pool of resources for batch-interactive jobs
- Leverage scalable resources
  - Understand how users problems scale
  - Understand how users are solving their scalable problems
  - Migrate more users to take advantage of scalable resources
- Make account renewals and job extensions easier
- Use cloud resources for elastically launching large R jobs

Questions?

Any questions, e-mail support@help.hmdc.harvard.edu

Web sites:

RT (Request Tracker): help.hmdc.harvard.edu

IQSS IT support: support.hmdc.harvard.edu