TUTORIAL

Cloud Computing for Pharmacometrics: Using AWS, NONMEM, PsN, Grid Engine, and Sonic

S Sanduja*, P Jewell, E Aron and N Pharai

Cloud computing allows pharmacometricians to access advanced hardware, network, and security resources available to expedite analysis and reporting. Cloud-based computing environments are available at a fraction of the time and effort when compared to traditional local datacenter-based solutions. This tutorial explains how to get started with building your own personal cloud computer cluster using Amazon Web Services (AWS), NONMEM, PsN, Grid Engine, and Sonic.

CPT Pharmacometrics Syst. Pharmacol. (2015) 4, 537–546; doi:10.1002/psp4.12016; published online on 22 August 2015.

NON-linear mixed-effect modeling (NONMEM1) is a modeling tool that pharmacometricians use in population pharmacokinetics and pharmacodynamics analysis. NONMEM provides built-in models that can be utilized to solve numerous pharmacokinetic and pharmacodynamic-related problems. It was first programmed in FORTRAN 77 code and requires a FORTRAN 90/95-compliant compiler to run. When dealing with NONMEM executions, it is important that sufficient hardware resources are available to guarantee that a job can be processed as efficiently as possible. Unfortunately, NONMEM runs can take a considerable amount of time, perhaps many hours or even days, depending on the speed of the computer and the size of the problem. This behavior is certainly not desired by pharmacometricians who must often execute analysis jobs quickly. Fortunately, newer versions of NONMEM support parallel analysis in which the computation workload is divided and distributed to multiple CPUs that execute the job in parallel. This ensures a shorter waiting period for a job to complete. The purpose of this tutorial was not to provide guidance on pharmacometric analysis using NONMEM and Perl-speaks-NONMEM (PsN2), but mainly on the usage of AWS3, Grid Engine,4 and Sonic5 applications to support more efficient and faster analysis using NONMEM on a cloud-based compute cluster.

CLOUD COMPUTING BASICS

In this age of the internet, software applications that were traditionally hosted within an organization’s own infrastructure are making the transition to what is known as “the cloud.” But what is the cloud? More generally, what is cloud computing? Cloud computing is an approach to computing that enables organizations to improve their business pipeline efficiently by utilizing shared infrastructures that are outside the scope of the organization. An organization can lease their specific computational resources from the shared infrastructure for the time that they need, ranging from an hour to a year or more. The rise of cloud computing can be attributed to the vast amount of complexity that an entity encounters in the traditional IT environment in which the entity uses assumptions to predict the number of resources it needs to host an application rather than a true accurate measure. Applications that are not hosted on the cloud have a direct impact on multiple areas within a company. From the business perspective, the company must pay some amount upfront to buy and maintain resources without truly knowing how much workload will be directed to the application. For example, simple data storage requires hardware and server maintenance. This can be expensive to the business if the amount of resources purchased is more than what is actually required. On the other hand, if the amount of resources owned by the company is not adequate to support the users who are trying to access the application, the company may face the problem of scalability. This can be problematic, because the goal of a company is to reduce costs and be as efficient as possible while giving more users access to its application. From the developer’s perspective, deploying an application can also be very time consuming. The developer must go through the hassle of setting up the operating system, databases, backups, and any other resources manually that an application might require. This can be inefficient because it is wasteful to spend excess time deploying an application rather than developing it. Eventually the problems of scalability and deployment trickle down to the end user, which can jeopardize the business. The cloud tackles these problems head on. By utilizing the services that the cloud provides, an organization can ensure reliability in all of its applications.

The cloud offers many services to ensure reliability of an application. The most popular services of the cloud are that of software, platform, and infrastructure or storage. Software as a Service (SaaS), also known as “on-demand software,” is a software delivery method where applications are delivered over the internet rather than having the user manually install the software directly onto his or her machine. Some benefits of SaaS include cost savings, scalability, accessibility, upgradeability, and resiliency. When compared to a traditional IT infrastructure, SaaS can produce cost savings when utilized because additional resources (hardware) are not needed to add new software. In terms of scalability, SaaS allows the user to adjust the monthly SaaS subscription as required when new users are...
added to the organization, eliminating the need for individual software licenses. SaaS also automates software updates whenever a new version is available. This feature removes updating responsibilities of the administrator in the in-house IT department. Another service that the cloud provides is Platform as a Service. Platform as a Service provides the environment and tools needed for creating new online applications. By utilizing Platform as a Service, a company can successfully reduce time and effort and minimize the technical maintenance needed to maintain applications. Finally, we have Infrastructure as a Service (IaaS), where equipment, such as hardware storage, servers, and network components, is owned and hosted by a service provider and is offered to consumers on-demand. In this model, IaaS provides compute power, memory, and storage to the consumer, and is typically priced on a per-hour basis that is dependent on the resource consumption. The consumer will only pay for what is used and the service provides all the capacity that a consumer will need. One of the greatest advantages of IaaS is that it offers a cloud-based data center that does not require the consumer to install new equipment or wait for the hardware procurement process. In turn, this yields to more resources being available that otherwise were not. Currently, there are many vendors in the market that offer IaaS. A few of the vendors that provide IaaS are AWS, Microsoft Azure, Google Cloud, and Rackspace Managed Cloud. There are advantages and limitations associated with each vendor. For example, AWS, although it offers a competitive pricing model, a rich set of services, and integrated monitoring tools, is a complex mixture of services and, thus, it can be difficult to project expenses as workflows become more complex. Similarly, although Azure offers reasonable prices for use, the minimal portal interface may not be appealing to high-end users. The Google Compute Engine is well suited for high performance computing but lacks ease of administration features. It often calls for additional patches and packages to create a cluster. Finally, the Rackspace Cloud also offers minimal costs and an easy-to-use control panel, but does not offer the messaging or specialized services that Amazon does. From a specifications perspective, Amazon EC2 offers 7.5 GB of RAM, while Google Compute Engine offers 3.75 GB, Azure offers 3.5 GB, and Rackspace offers 4 GB. Similarly, Amazon EC2 provides 2 × 420 GB of disk space, whereas Google offers 420 GB, Azure offers 30 GB, and Rackspace offers 160 GB. All in all, IaaS is offered from a number of vendors with varying specifications, but vendor choice depends on how much the consumer intends to use or spend. Typically, the price of the vendor application correlates positively with the number of available features; as the number of features increases, so does the price of the application.

AMAZON WEB SERVICES

AWS uses remote computing services via Amazon.com to construct a cloud computing platform. These services, because they are cloud based, replace IT infrastructure, lower time and effort for most companies, provide agility and instant elasticity that allow for minimizing and maximizing applications, is open and flexible in allowing users to choose what development platform or programming model is used, and maintain a secure network that ensures data privacy and security. AWS offers a number of services and powers thousands of businesses across the globe.

CONFIGURING THE AWS COMMAND LINE INTERFACE

The AWS Command Line Interface is a unified tool used to manage AWS. With this, you may download, configure, and control multiple AWSs from the command line and automate them through scripts. Use the “AWS configure” command to setup the AWS installation. You may be asked to enter an AWS access key ID, AWS secret access key, default region name, and default output format. Refer to the AWS Command Line Interface User Guide to learn more about installing and configuring the tool. “Screen” is a terminal multiplexer shipped with Ubuntu instructions, which allows a user to access multiple separate terminal sessions inside a single window. This increases ease of use when working with SSH tunnels.

ELASTIC COMPUTING CLOUD

The Elastic Computing Cloud (EC2) provides scalable capacity in the AWS cloud and is designed to reduce the efforts in web-scale cloud computing by eliminating investments in hardware and servers. Virtual servers are used to configure security and networking and to manage storage. Users only pay for the capacity used, which aids in building effective applications at reasonable expenses.

Some of the EC2 features include:

- Virtual computing environments (instances);
- Preconfigured templates for instances;
- Various configurations of CPU, memory storage, and network capacity for instances;
- Secure login information for instances using key pairs;
- Storage volumes for temporary data that is deleted when an instance is terminated;
- Persistent storage volumes for data using Amazon Elastic Block Store (Amazon EBS);
- Multiple physical locations for resources (Availability Zones);
- A Security Group that enables specification of protocols, ports, and source IP ranges that reach instances;
- Static IP addresses for dynamic cloud computing (Elastic IP addresses);
- Metadata (tags) used to create and assign Amazon EC2 resources;
- Virtual networks that can be created to isolate from the rest of the AWS cloud; these can be connected to your own network (virtual private clouds (VPCs)).

Virtual private cloud

The VPC allows the user to arrange a logically isolated section of the AWS Cloud in which the user can define and
launched AWS resources in a virtual network. The user has complete control over the virtual networking environment, including the selection of IP address ranges, creation of subnets, and configuration of route tables and network gateways. The network configuration can easily be customized. For instance, the user can create a public-facing subnet for webservers who have access to the internet and for the backend systems, such as databases or application servers; users can create a private subnet without internet access. This enables the user to leverage multiple layers of security, including security groups and network access control lists to help control access to Amazon EC2 instances in each individual subnet. The user can also create a Hardware Virtual Private Network connection between the corporate datacenter and VPC.

Storage
Amazon Simple Storage Service (Amazon S3) provides developers and IT teams with secure, durable, and highly scalable object storage. Amazon S3 is user-friendly and has a simple web service interface in which the user can store and retrieve any amount of data from anywhere on the web. Amazon S3 is cost-effective because the user only pays for the storage that is actually used and there is no minimum fee or setup cost. In addition, Amazon S3 can be used either alone or in sync with other AWSs, including, but not limited to, Amazon Elastic Compute Cloud (Amazon EC2), Amazon Elastic Block Store (Amazon EBS), and Amazon Glacier.

EC2 Instances
Amazon EC2 Instances provide a wide selection of instance types comprised of varying combinations of CPU, memory, storage, and network capacity to fit different use cases, and offer flexibility to choose a necessary combination of resources for applications. EC2 instances provide features to help deploy, manage, and scale user applications. See Table 1 for the list of select instance types available in EC2, which can provide advanced performance for pharmacometricians.

Amazon Machine Image
An Amazon Machine Image (AMI) is a virtual tool used to create and launch an instance in the cloud. A number of instances or virtual servers can be launched as needed. The lifecycle of an AMI involves creating a template for the root volume for the instance, creating permissions to control the AWS accounts that can use the AMI to launch instances, and providing a block device mapping that specifies the volumes that need to be attached to the instance when launched. Instances can be customized and deregistered as needed.

Security Groups
A Security Group mimics a virtual firewall to control the traffic of a single or number of instances. One or more security groups may be used when an instance is launched. Rules can be added to each security group to control traffic between instances. These can be customized as each rule is added to all instances associated with the security group.

NONMEM
NONMEM, which stands for NON-linear Mixed-Effect Modeling, is a specialized software used to analyze pharmacokinetic and pharmacodynamics data. The software is a regression program that specializes in nonlinear systems by solving differential equations, and is used to solve pharmaceutical statistical problems within a subject or between subjects. The analysis allows pharmaceutical companies to determine necessary dosage strategies for products and helps in understanding how drugs interact with various individuals.

Perl-speaks-NONMEM
PsN, which stands for Perl-speaks-NONMEM, is a collection of Perl modules and programs that assists in the development of nonlinear mixed effect models using NONMEM. Some of the functions include providing solutions to a parameter estimate extraction from output files, ensuring data file sub settings, and resampling. Further, it applies to advanced computer-intensive statistical methods. It allows for the implementation of bootstraps, visual predictive checks, and other useful functionalities. PsN also includes stand-alone tools for end-users and development libraries for method developers.

Grid Engine
Grid Engine is a cluster management software that manages access, reports usage, and enforces business policies for a compute cluster. It is typically used on a computer farm or high-performance computing cluster and is responsible for accepting, scheduling, dispatching, and managing the remote and distributed execution of large numbers of stand-alone, parallel, or interactive user jobs. The allocation of distributed resources, such as processors, memory, disk space, and software licenses is also managed through the Grid Engine. The software allows for more organized usage of the compute hosts. Although it is an integral part of a cloud cluster, deployment is complex, and is beyond the scope of this tutorial. Grid Engine allows users to run applications and administrators to implement resource configurations and policy configurations. Sonic automatically configures Grid Engine so that the user does not need to.

Sonic
Sonic Cloud Computing Platform is a web-based, enterprise-level, and secure software solution for the management of cloud based clusters. It is proprietary software of Rudraya

---

**Table 1 EC2 instance types**

| Model    | Virtual CPUs | Memory (GB) |
|----------|--------------|-------------|
| m3.xlarge| 4            | 15          |
| m3.2xlarge| 8           | 30          |
| c3.2xlarge| 8           | 15          |
| c3.4xlarge| 16          | 30          |
| c3.8xlarge| 32          | 60          |
| r3.2xlarge| 8           | 61          |
| r3.4xlarge| 16          | 122         |
| r3.8xlarge| 32          | 244         |

---

Cloud Computing for Pharmacometrics: Using AWS, NONMEM, PsN, Grid Engine, and Sonic
Sanduja et al.
Corporation and the cost associated depends on the size of implementation. This allows the user to ensure that the tool is maintained, supported, and validated. Sonic provides resizable compute capacity in the cloud. The cloud-based computing platform provides an economical, agile, and flexible environment for pharmacometrics, genomics, biostatistics, and translational medicine needs. Cloud computing involves a combination of software and computing delivered on demand as a service. There are options apart from Sonic that allow the user to set up a cloud cluster, including products and services by Metrum, Mango Solutions, Certara, and Cycle Computing, however, this tutorial focuses on Sonic as one example of a software tool that provides an easy-to-use environment for cloud computing. One of the key benefits of cloud computing is the opportunity to replace up-front capital infrastructure expenses with less time and effort, increasing efficiency and productivity for a business.

**CREATING PERSONAL CLOUD COMPUTE CLUSTER**

Create AWS account

To create a new AWS account, proceed to http://aws.amazon.com/ and click the “sign up” button in the top-right corner. Select the “I am a new user” checkbox and enter your email address. Fill out the email address and password boxes and click to continue. You may get a confirmation on the next screen which says “you have not yet signed up for AWS: click here to sign up for AWS,” and do so (click the link). Fill out all of the residential information and continue.

The next section will ask you for basic credit card payment information, and after this you will be asked to verify your telephone number. This is an automated system, which works well, and we have never had an issue with it. After clicking “call me now,” you will get a telephone call in which a computer voice will state the pin number you are to enter on the page. In the final segment, you will be asked to choose what type of support plan you desire. Because you are most likely creating this account as a test for the first time, choosing a “basic support plan” is fine at this time. After this is complete, you will return back to the AWS website homepage and you will need to log in once more. This is accomplished by clicking the small dropdown menu next to the location where you clicked “Sign up” previously. Choose “my account” and log in with your email and password. You should be logged in successfully to the AWS control panel and be able to see the “services” menu in the extreme top left that you can expand (see Figure 1).

Create head node and compute node instances or use existing

After setting up the AWS account initially, we can use it to create a standalone server for running NONMEM analysis or PsN execute and bootstrap jobs as described later. Go to the top-left corner and click the services dropdown menu and choose “EC2.” In the left-hand panel, choose instances. In the upper left, choose “Launch instance.” Select “Ubuntu 14.04” or whatever the most recent LTS release of Ubuntu is. Next, select an instance size. For a basic testing system, using a micro instance is fine, and you always have the option to change to a more powerful system later.

At the time of writing, AWS micro instances only come in the form “t2.micro,” and “t2” instances cannot be launched into a standalone network (EC2-Classic). This means we will have to do a little work setting up a VPC to connect to an instance.

Click the “create new VPC” button to the right of the VPC selection box. This will open a new tab in the web browser that we will use to set up the VPC that our machine will reside in. Again click “create VPC” in the upper left corner. In the box, choose any name you would like; for this guide, we will be using the convention “Sample vpc,” “Sample
choose the “elastic ip’s” section to the left and click the

The machine will start to boot up. While it is working,
select “create new” and download the key; then launch.

This will create a box allowing you to choose a key pair. You
choose “new security group” is selected and that only port
22 (SSH) is listed below. Hit “next” and then hit “launch;
“create a new security group” is selected and that only port

name you would like and insert it into the value box. Hit
“next: tag instance.” Choose any
name you would like and insert it into the value box. Hit
“next: add storage.” Keep the
subnet,” etc. CIDR block should be “10.0.0.0/16” and Ten-

in the “subnets” section. In here, click “create subnet” and fill out the name. VPC
should have a VPC named the same as previously available in the dropdown menu; choose it. For the Availability
zone, pick “No Preference,” and for the CIDR block, input “10.0.0.0/24.” After this, choose “internet gateways” on the
left, and create a new internet gateway with any name you
like. Right click on the gateway and choose “attach to vpc,”
making sure to choose the VPC you created earlier. Next
choose “Route tables” on the left. In the list, there should exist a route table, which has already been created for our
VPC, and has your VPC name listed. Click on it and some options will appear below. Under the “subnet associations”
tab, click “edit” and select your previously created subnet;
click save. Under the “Routes” tab, add another route with
destination being “0.0.0.0/0” and target being the inter-

Figure 2 PuTTY configuration window.

Creating a new key pair
When you start a new instance, PuTTY will ask you if you
want to create a new key pair. If you choose to do so,
you will be prompted to enter a key pair name and
choose a location to save the files. The key pair name
will be used to reference the key file in future steps.

Connecting via SSH/SFTP and PuTTY Configuration

The key file that you download is a type of file known as a
“private key” file. It is designed so that it will cryptographi-
cally match with another file, called a “public key” file, which
is already installed on the cloud server. The standard for-
mat of the key files has a “.pem” file extension. This should
be familiar to anyone who had used linux/ssh before. If you
use a windows system and connect with the PuTTY SSH cli-

The key file that you download is a type of file known as a
“private key” file. It is designed so that it will cryptographi-
cally match with another file, called a “public key” file, which
is already installed on the cloud server. The standard for-
mat of the key files has a “.pem” file extension. This should
be familiar to anyone who had used linux/ssh before. If you
use a windows system and connect with the PuTTY SSH cli-

Using the previously obtained IP address; you may con-
nect to the new node/instance in order to execute com-
mands or download files. For executing commands on
Windows, we will focus on using the “PuTTY” SSH client
tool. To connect to the instance, type or paste the IP
address into the “Host Name (or IP address)” field in the
top center (see Figure 2).

Next, expand the “SSH” section under the “Connection”
section in the left pane. Under it, choose the “Auth” button.
In the bottommost field will be a section labeled “Private
key file for authentication.” Click “browse” and select the
location where you have stored the “.ppk” file previously
(see Figure 3).

Finally, click back to the uppermost button in the left
pane: “session.” Type a name under the “saved sessions”
section and click the nearby “save” button to make an eas-
ily reusable profile for this instance. Now you may connect
by clicking “Open” (at the bottom). A new window will open
which states: “login as:...”. Type in the user name “<your
username>” and press the enter key. The session will begin
and you will now be able to execute SSH commands.

In order to transfer files, we can use a similar process with
another program: Filezilla. Open up a Filezilla client and
type in the IP address once more into the “host” field and the
name “<your username>” into the “username” field near the

www.wileyonlinelibrary.com
top. Now, from the top toolbar, choose "edit > settings..." Under "Connection > SFTP" should be a button which reads: "Add keyfile..." Click it, and this time, choose the "PEM" file instead of the "PPK" file. Choose the "ok" button to exit settings and choose "Quickconnect" (to the right of the host and username fields). After a few seconds, you should see a listing of the files on the remote server.

Install the analysis software and create an image

In order to prepare for the installation, we will need to install some supporting libraries and tools. In Ubuntu, this is fairly easy because it only requires inputting one command for each library. By default, instances are set up with Ubuntu user account with sudo privileges. The same account can be used to install analysis software. Note that for the remainder of this tutorial, the "$" symbol followed by text indicates that that line of text is meant to be entered into the PuTTy window when connected to your machine. You enter the text after the "$," but not the "$" itself. Enter the following commands, in order, one at a time:

```bash
$ sudo add-apt-repository ppa:webupd8team/java
$ sudo apt-get update
$ sudo apt-get install oracle-java7-installer
$ sudo apt-get install gcc-multilib
$ sudo apt-get install r-recommended
$ sudo apt-get upgrade
```

Next, we need to install the two commercial pieces of software in the setup: Intel Fortran composer27 and NONMEM. A noncommercial version of the Fortran compiler can be used, however, optimal performance is obtained with the Intel compiler. Each should have either come with a physical installation disk or a digital download. For the Fortran composer, we are looking for the folder containing a file called "install.sh." Upload this entire folder to the machine using the Filezilla SFTP client we opened earlier. It is fine to place it in your user home directory, for example "home/<your user>/fortran." Assuming this is where you placed it, enter the directory on the PuTTy window and run the file:

```bash
$ cd ~/fortran
$ sudo chmod +x install.sh
$ sudo ./install.sh
```

The script is good in the readability department and you should have no issue following along. After entering the license key, you should choose option 2 to change which components to install. Choose 3, 5, and 7 to disable 32 bit components (the vast majority of AWS instances will be 64 bit only). Choose option 1 and continue with the installation. This can take a very long time but it will finish eventually. To complete the installation on Ubuntu, you will need to add a failsafe shell script to add the license file to the environment if it is not automatically found. To do this, create a new file:

```bash
$ sudo nano /etc/profile.d/ifort.sh
```

Copy and paste the following text into the file (in PuTTy, copy the text here as you usually would, and right click anywhere in the PuTTy window to paste).

```bash
if [x$INTEL_LICENSE_FILE = "x"]
then source /opt/intel/bin/compilervars.sh intel64
fi
```

Next, save and exit by pressing CTRL+O, <Return key>, CTRL+X. Finally, verify that the file is accessible by all users:

```bash
$ sudo chmod 755 /etc/profile.d/ifort.sh
```

This concludes the Fortran composer installation. Next, we will install NONMEM. Upload the installation files in a similar manner, with the folder having a file with the name “SETUP73.” Supposing you have uploaded the installation files to the folder called “NONMEM,” run the following command to start the installer:

```bash
$ sudo /bin/bash SETUP32/home/<your username>/NONMEM7/opt/nm73/opt/intel/bin/ifort y ar same rec l
```

Accept the defaults for the settings by entering "y" at each of the prompts. As before, the installation may take a very long time. Next, you will have to install the license. Using Filezilla, upload the license file to your home directory and execute the following commands:

```bash
$ cd ~/
$ cp nonmem.lic /opt/nm73/license/nonmem.lic
```

Next, you can add the NONMEM binary to the system path so that you can run it from anywhere easily.

```bash
$ sudo nano/etc/profile
```

---

**Figure 3** PuTTy configuration window.
Scroll to the bottom of the file with the arrow keys and paste in the following text as before:

PATH=$PATH:/opt/nm72/run
export PATH

Save the file and exit as before. You will need to log out of PuTTY and log back in for the path to update, but after that, the NONMEM installation should be complete. Now that the underlying components are installed, we can install the automation software PsN to greatly assist in running NONMEM jobs.

To install PsN, begin by setting up the required Perl libraries:

```
$ sudo apt-get install libmoose-perl
$ sudo cpan Moose::Random
$ sudo cpan Statistics::Distributions
$ sudo cpan Archive::Zip
$ sudo cpan File::Copy::Recursive
$ sudo cpan Storable
$ sudo cpan Moose
$ sudo cpan MooseX::Params::Validate

(If you are prompted to configure as much as possible automatically, select “yes” and for “approach,” select “sudo.”)
```

As of this writing, the main PsN installation file can be downloaded for free at http://sourceforge.net/projects/psn/files/PsN-4.2.0.tar.gz/download. Obtain the file and upload it using Filezilla.

The following installation commands assume that you placed the file in the folder “PsN” under your home directory:

```
$ cd ~/psn
$ tar -zxf PsN-3.5.3.tar.gz
$ cd PsN-Source
$ sudo perl -w setup.pl
```

Choose the default option for every step and the installation will be completed.

At this point, you have the option to create an image (AMI) on AWS so that you will not have to go through the installation again. Keep in mind that if you terminate an instance, unless you create an AMI, any machine configuration as well as all stored data on the machine will be lost forever. Back on your AWS account, under the EC2 service, select “instances” on the left-hand side. Right click on the sample instance you made and select “create image.” Choose any name you like; note that the instance will shut down during imaging, but they will not be deleted. You can start them from where you left off once the image creation process finishes. After this process is complete, you can click on “images” to the side, right click on your image, and choose “launch” to create a ready-to-use server.

**Perform NONMEM analysis using PsN execute command**

You now have everything setup to run your jobs and retrieve the output. First, you need to place the data on the server.

```bash
Model run time: 2m53.486s 3m2.563s
Memory: 8053828kB 7693824kB
Disk space: 27.8 GB 8 GB
CPU cores: 4 2
CPU speed: 2.53 GHz 2.50 GHz
CPU usage: 100% 100%
Server location: Local, New Jersey, USA Virginia, USA
Server type: VMWare Instance Amazon Instance
```

You may create a new folder if you wish by right clicking on the right-hand pane in Filezilla and choosing “create directory.” Once you have opened the folder you wish to use, drag and drop or right click > upload any control and data files that are required to the folder on the server (in the right pane of Filezilla). Note that in order for the job to run correctly, the folder that it runs from must be under /home on the head node. Once the files are uploaded, switch to the PuTTY/SSH window and navigate to the directory that you uploaded to. If you are not sure how to do this, the easiest guaranteed method is to choose the path listed in the right pane of Filezilla above the files listing, and paste it into the PuTTY window after the command “cd.” For example, suppose you created and entered a new folder in Filezilla under your home directory called “test01.” The path displayed above the files in Filezilla should read: “/home/<your username>/test01.” Take this and paste or type it into the terminal window so the command in PuTTY should read: “cd/home/<your username>/test01.” Next, you may begin execution. In the terminal window, enter the command “execute test01.ctl,” replacing “test01.ctl” with the name of the control file that you had uploaded. The terminal window will show a change indicating that the model is running.

See Table 2 for an example of a NONMEM analysis run on the local cluster vs. a run on the cloud cluster. Note that local or cloud-based virtual machines with similar specifications are able to process a resource intensive NONMEM execution in a similar amount of time. This illustrates that it is possible to obtain similar productivity using an Amazon cloud image compared to a local virtual image on a dedicated hardware, although in either case the quality (server hardware or Amazon instance type) of the resource that you purchase will have a direct impact on the speed at which runs are processed.

**Perform NONMEM analysis using PsN bootstrap command**

The process for initiating a bootstrap job is very similar to that of the above step, starting an execute job. The main difference is that you may choose the amount of threads per core and the amount of samples (separate runs) for bootstrapping. All of the setup (creating the folders and uploading the data and control files) is exactly the same. Suppose we have another control file, called test02.ctl, which we intend to run as a bootstrap job. Assuming that we have already navigated to the correct folder, we can run the command “bootstrap test02.ctl –threads=4 –samples=8,” replacing the “4” with the number of threads to use (multiple threads will...
run on only one machine at a time) and replacing “8” with the number of samples to repeat (each sample can run on a separate compute node). The terminal window will show the status of each sample run, adding a number to the count after each run is submitted, and after each run is completed.

Launching a new cloud cluster
To use the Sonic interface, proceed to http://sonic.mashframe.com and log in using the boxes in the top-right corner. After logging in, click the “Sonic clusters” dropdown and choose “Clusters Manager” (see Figure 4).

Choose “New Cluster...” in the top right corner and fill out the short information form (see Figure 5) that is presented. The screen will return to an overview of any clusters that you have previously created and will keep track of the status of preparing the new cluster. This may take some time to complete, but usually no more than 15 minutes.

When it is finished, a button reading “Get keys and information” will become available. This will pop-up in a screen giving the login username and IP address, as well as a button that will download the key file to authenticate to the head node. Note that after clicking download for the first time, you will not be able to re-download the key again for this cluster. Also, be sure to make a note of the IP address before closing the dialog box. By using this IP address and key file, you will be able to log into the system.

Tracking of jobs on a cloud cluster
You may often need to run an analysis on a very large dataset in which the execution will take a very long time, and, often, you will be sharing the available resources with other users. Because of this, it is helpful to be able to monitor the status of the current running jobs on the grid. Commands for this are provided for simple viewing using the SSH connection. On the PuTTY window, type the command “qstat –f.” This shows a listing of all of the compute nodes and the head node as well as any jobs currently running and jobs that are still waiting for a free slot. Note that you cannot run this command in the same PuTTY window as you submitted your own jobs in because PsN will keep control of the window until it is finished. Instead, you will have
to open a second instance of the PuTTY program and log in again.

Also note that a few useful commands are available to modify jobs you have started if something comes up. Type the command “qmod -sj 5” or “qmod -sj "***" to suspend a running job. If the argument is numerical, as in the former, you are indicating the job number/id as read from qstat. If the argument is "***" (note that the quotes are required), all of your jobs will be suspended. Note that this will stop the job from consuming CPU time but will not remove the job from the queue. Two more commands can be used with similar syntax: “qmod -usj” and “qdel” (both of which still require the "***" or job number). The former will unsuspend a previously suspended job, and qdel will completely remove/delete a job (see Table 3).

Audit trail of jobs running on a cloud cluster
The Sonic audit trail extension will make audit data from a Linux cluster available to view on the Sonic Platform. This provides a level of increased convenience as well as retention of the audit data after the cluster machine is terminated. The audit trail must be activated when the cluster is created and cannot be added after the fact. When using the Sonic Platform to create a cluster, as described above, just select the check-box called “enable auditing” to set up the auditing in the background. Please note that it will create slightly more network overhead, which is why it is not enabled by default. Currently, the audit viewer on the Sonic Platform website is broken down into two sections: “TTY” and “Low Level.” TTY mode will show the exact commands that each user typed in the Bash session. It is useful when a user may have made a mistake and you can look back to see why the problem might have arisen. “Low Level” really means: everything else. These are system events that can be configured to your liking using the audited configuration options, which are well-documented on the internet. Anything that appears in the audit log file will appear here. The automatically generated Sonic instances are moderately paranoid by default, but you may wish to manually add more monitoring. Please note that for the TTY log, the log will only be updated after the user ends their Bash session. So if you are imagining that a command was entered that you cannot find in the log, you can force the user to log out first and then check the log.

| Table 3 Common grid engine commands |
|-------------------------------------|
| **Command** | **What it is used for:** |
| qstat -f | Display a listing of all nodes and their load/slots as well as all current running jobs and their status, such as what node they are running on and their job number. |
| qmod -sj 15 | Suspend (temporarily stop) a specific job. |
| qmod -sj "***" | Suspend (temporarily stop) all jobs. |
| qmod -usj 15 | Unsuspend (start after having been stopped) a specific job. |
| qmod -usj "***" | Unsuspend (start after having been stopped) all jobs. |
| qdel 15 | Stop and remove a specific job. |
| qdel "***" | Stop and remove all jobs. |

Backup of analysis data on a cloud cluster
The Sonic backup program is a simple interface to AWS Glacier features. AWS Glacier is a cost-efficient product of AWS, which allows extremely large storage volumes for a very low cost. A downside to using Amazon Glacier is that obtaining the data from storage may take a number of hours to complete. Amazon Glacier is suited for long-term backups, while S3, an alternative backup program, is secure, efficient, and cost-effective for backing up and retrieving information immediately. The Glacier service is preferred for this application because you may or may not actually use the stored data, and so, the low price tag—usually less than $0.01 per gigabyte per month—is highly desirable. Glacier services are available in the same regions as the other AWS so you can keep latency low. Amazon Glacier synchronously stores your data across multiple facilities before returning SUCCESS on uploading archives, so there are no worries of lost data.

The included program is meant as a set-up-and-forget, which will keep a backup sustained at a defined interval. Before launching the program, you must edit the configuration file to reflect your preferences. Enter the command “sudo nano /etc/sonic/sonicBackup.ini.” You will be able to change the name of the vault that the cluster uses, the AWS credentials, and the AWS region.

You may then launch the setup script by running the command “sudo sonicBackup” on any node created with Sonic. The script will present a small list of options, the primary one being the first “create backup.” You navigate this program by typing a number or letter for an option and pressing the enter key. Choose option 1—create backup—to begin. Follow the on-screen instructions. The only step that might not make immediate sense to you is entering the path, as you must manually type the path. Usually typing “/home” will obtain all user files and get everything that you need, but you may specify another path if you desire. After following the setup instructions, the job will continue to run in the background at the specified interval.

If you would like to set up email notifications for verifications, you may use option 4 and enter the desired email address.

Managing user access on a cloud cluster
The Sonic administrative program is another helper program that provides automation of common Linux administrative actions across all of the compute nodes in parallel. It does not require any configuration when it is launched as part of a Sonic cluster. You may run the program by typing “sudo sonicAdmin.”

Options include adding users and groups, suspending or unsuspending users, and adding/removing users to/from groups. All of these actions should be self-evident. When creating a new group, an option to create a study folder is presented, which will set up automatic folder permissions for users of a specific study group so that when they run a job from the study folder the group may be able to read the results.

It is recommended that you only add new users with the script instead of manually if you plan to allow the user to
run jobs on the grid because it is required for the users to exist on all of the nodes in the cluster.

**Security in the cloud**
Amazon provides support for encryption of one or more storage devices that may be attached to an Amazon image. This makes it easy to encrypt large amounts of data; a private key is used to log into the machines, streamlining access, whereas an encryption key works like a password to provide heightened security for the server. Amazon encrypted EBS is generally very helpful, but it is limited in scope because it can only encrypt specific locations and not the entire machine (for example, system boot files). If analyses are to be stored in a specific folder, where the encrypted volume may be mounted, then encrypted EBS may be a viable option to encrypt results.

**CONCLUSION**

In this tutorial, we presented how to get started with building your own personal cloud compute cluster using AWS, NONMEM, PsN, Grid Engine, and Sonic. In our view, a cloud-based compute cluster can provide a powerful yet very economical computing platform for performing NONMEM analysis. Tools like Sonic can provide a very simple-to-use web-based interface for the management of cloud-based clusters, so that you only actively run the amount of resources that you need during analysis.

**Conflict of Interest.** S.S. is CEO of Rudraya Corporation, which provides commercial licensing of Sonic. The other authors declared no conflict of interest.

1. Beal, S., Sheiner, L.B., Boekmann, A. & Bauer, R.J. NONMEM’s user’s guides (ICON Development Solutions, Elliott City, MD, 2009).
2. Nyhoff, L.R. & Leestma, S.C. Fortran 90 for engineers (Prentice-Hall, Upper Saddle River, NJ, 1997).