

SIT323 Practical Software Development, Trimester 2, 2019

Assessment Task 2

Summary

You need to create some software to help determine a set of allocations of tasks to processors. This set of **allocations must be displayed on your GUI**. These allocations are based on data from within a configuration file.

Your software will use the following configuration files stored on an Azure blob.

- <https://sit323sa.blob.core.windows.net/at2/TestA.txt>
- <https://sit323sa.blob.core.windows.net/at2/TestB.txt>
- <https://sit323sa.blob.core.windows.net/at2/TestC.txt>

The ideal goal is to find an allocation that:

1. consumes a minimum amount of energy and
2. its overall runtime does not exceed the program duration

There might be one or more allocations that meet these constraints. Also, there might be no allocation that meets these constraints because the program duration is too small for all combinations of task allocations.

C#

Your GUI client software will directly read a configuration file that are stored on Azure. That is, your software must open the file using its URL. Do not download these configuration files.

You will design and implement WCF Services to compute a set of allocations based on the imported configuration data.

As your software might not be able to find the ideal allocation within several years, it must attempt to obtain a set of allocations that have the smallest energy consumption that you can find within 5 minutes. That is, after 5 minutes, your software must:

- stop searching for other allocations
- display the low energy consuming allocations that your program found
- display the runtimes and amount of energy consumed next to each allocation

Each WCF Service will remotely run on an AWS virtual machine. This service will be indirectly invoked by your client software. The client software will provide/pass configuration data to this service, and this service will return a set of allocations.

In order to use more compute resources (AWS VMs) to determine a set of allocations, you will deploy several AWS virtual machines, a load balancer, and other AWS resources. Each of these VMs will have your WCF Service installed, i.e., there will be a set of similar WCF Services.

Your software can **send several asynchronous requests** to the set of WCF Services. It will receive responses from the WCF Services. Each response will contain a set of allocations. For example:

- WCF Service 1 returns a set of 8 allocations, where each allocation consumes 100.
- WCF Service 2 returns a set of 4 allocations, where each allocation consumes 150.
- WCF Service 3 returns a set of 12 allocations, where each allocation consumes 90.
- WCF Service 4 returns a set of 5 allocations, where each allocation consumes 123.

Your software needs to determine which response contains allocations that consume the minimum amount of energy. For example, the set of 12 allocations (in the above list) consume a minimum amount of energy.

Your software will display the set of allocations on your GUI client, including runtime and energy consumed for each allocation.

AWS Cloud Architecture (C#)

The cloud architecture that you will implement is depicted in Figure 1. Client requests will be directly sent to the application load balancer which will forward each request to one of the web servers on a VM.

You can have more than one kind of WCF Service. For example, one WCF Service computes an allocation based on a particular algorithm, the second WCF Service computes an allocation based on another kind of algorithm.

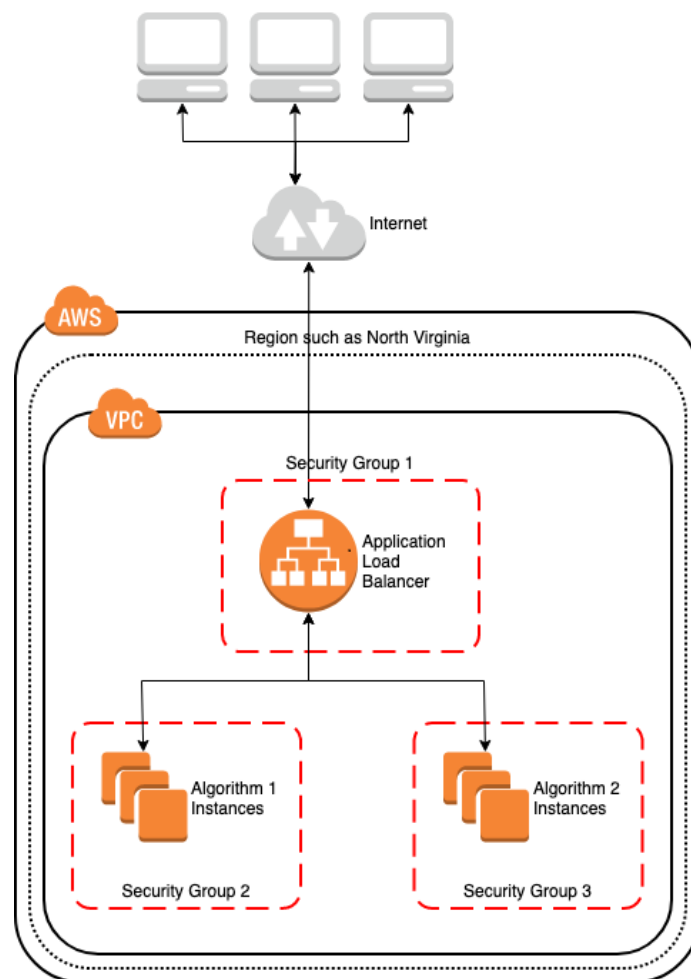


Figure 1.

Excel (for students enrolled in the old Cloud Computing major)

Your Excel software will directly import a configuration file that are stored on Azure. That is, your software must open the file using its URL. Do not download these configuration files. The following menu item will be useful to import data from a web page:

Data > Get External Data > From Web

You will design and implement one or more techniques to determine a set of allocations based on the imported configuration data. This can be a mixture of manual and automation techniques.

As your (manual or automatic) techniques might not be able to find the ideal allocation within several years, it must attempt to obtain a set of allocations that have the smallest energy consumption that you can find within 5 minutes. That is, after 5 minutes, you must:

- stop searching for other allocations
- display the low energy consuming allocations that your program found
- display the runtimes and amount of energy consumed next to each allocation

Your software will display this set of allocations on a worksheet, including runtime and energy consumed for each allocation.

AWS Cloud Architecture (Excel)

The cloud architecture that you will implement is depicted in Figure 2. Client requests will be directly sent to the application load balancer which will forward each request to one of the web servers on a VM.

Responses will be based on static web pages containing the set of allocations that you produced in Excel. That is,

1. you need to produce your allocation(s) in Excel
2. create a HTML page(s) containing allocations, run times, and energy values
3. upload web pages to your web servers

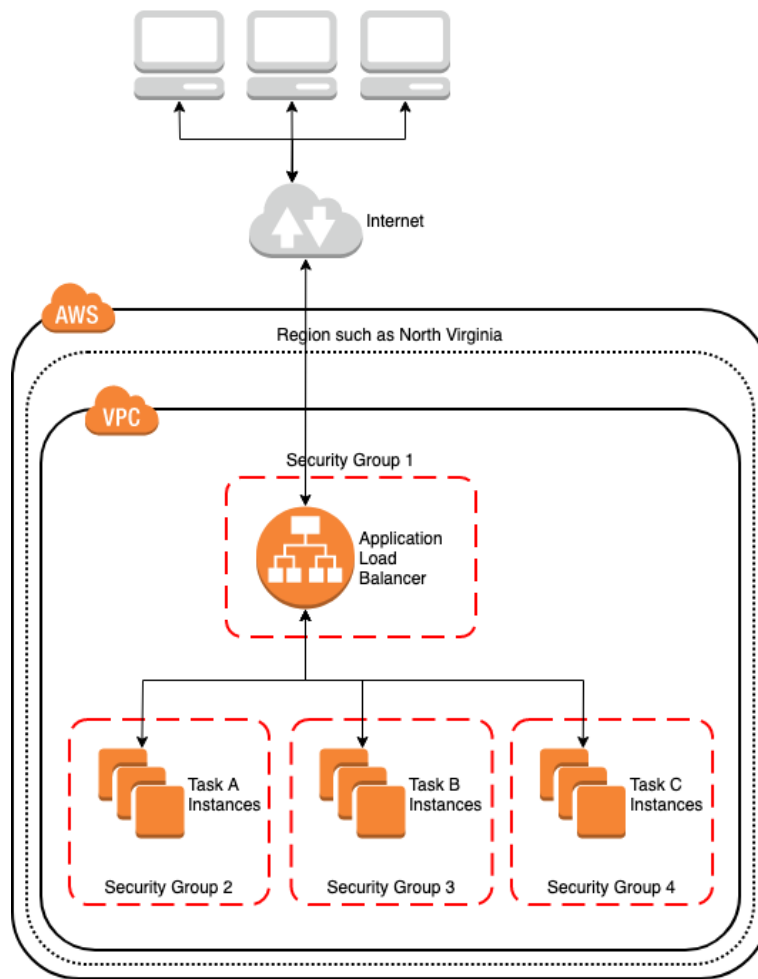


Figure 2.