

A CLOUD COMPUTING ARCHITECTURE FOR SUPPLY CHAIN NETWORK SIMULATION

Manuel D. Rossetti
Yaohua Chen

Department of Industrial Engineering
4207 Bell Engineering Center
University of Arkansas
Fayetteville, AR 72701, USA

ABSTRACT

This paper presents a Cloud Computing Architecture For Supply Chain Network Simulation (CCAFSCNS). The structure and elements of the CCAFSCNS as well as the relations among elements are introduced. The purpose of the architecture is to facilitate the distributed simulation of large-scale multi-echelon supply chains with an arborescent structure. The simulator in the CCAFSCNS permits the user to specify the network structure, the inventory stocking policies and demand characteristics so that supply chain performance can be estimated (e.g. average inventory on hand, average fill rates, average backorders, etc.) for each stock-keeping-unit. A prototype system that implements the CCAFSCNS is presented and illustrated with three example use cases. In addition, this paper discusses the time issues associated with the cloud computing solution and shows that the cloud computing solution can significantly shorten the simulation time.

1 INTRODUCTION

The performance of a supply chain impacts the competitive advantage of companies within the chain. How to coordinate the logistics activities in a supply chain, such as inventory control and transportation management, determines performance. Realistic multi-echelon supply chain networks make the analysis by mathematical models problematic for two main reasons: (1) the scale of the problem can be large, and (2) the complicated behaviors of different participants, such as manufacturers, retailers and customers. In addition, decision makers may need system performance during each planning stage to make step-by-step decisions. Therefore, simulation is an excellent tool to assist in supply chain analysis since it can (1) predict supply chain performance, (2) allow users to track the system performance at each time period, (3) perform what-if analysis for specific scenarios, and (4) compare alternative system configurations. One disadvantage of simulation methods is that it may be very time consuming to get statistically accurate results. When the multi-echelon supply chain network contains thousands of stock keep units (SKUs), it might take hours to run a single replication. In order to control the variance or half-width of the statistics generated by the simulation, many replications may be needed. Thus, if a single computer is used to simulate the multi-echelon supply chain network, it may take several hours or days to get the results. The purpose of this paper is to discuss the design of a cloud computing architecture that facilitates the execution of large-scale multi-echelon supply chain network simulations.

In this paper, a background section introduces supply chain modeling and supply chain simulation tools. Then, section 3 illustrates the Cloud Computing Architecture For Supply Chain Network Simulation (CCAFSCNS). The structure and elements of CCAFSCNS as well as the relations among elements will be introduced in detail. A prototype system that implements the CCAFSCNS is then presented while

introducing the elements. Three example use cases are developed to illustrate the functionalities of prototype system. Finally, the time issues associated with cloud computing simulations are discussed.

2 BACKGROUND

Rossetti and Chan (2003) conclude that the definition of supply chain includes common key words such as “logistics network, supplier, end-customer, raw material, information, goods/products, services, and facilities”. A supply chain can be conceptualized as a network within which materials flow from upstream suppliers to end customers. From the upstream to the downstream, the network consists of raw material suppliers, manufacturers, distribution centers (warehouses), retailers and customers. During the flowing processes within the network, the materials transformed from raw materials, to semi-products and finally become products that are ready to sell. Therefore, a supply chain modeling objective is the physical movements of materials in the network from upstream to downstream. Another modeling objective is the information flow from downstream to upstream within the network. The information flow is driven by customers who come to retailers to buy products. When the inventory at the retailer is not enough, the retailer will make orders to its supplier. The manufacturer will monitor the sales status and then replenish the products to its customers and order the raw materials from its vendors.

A comprehensive review of supply chain modeling is available in the paper written by Min and Zhou (2002). They categorize the supply chain problems into three levels of decision hierarchy: (1) competitive strategy, (2) tactical plans, and (3) operational routines. In addition, they classify supply chain models into four major categories: (1) deterministic, (2) stochastic, (3) hybrid, and (4) IT-driven. They also review the papers that relate to each category. This paper focuses on simulation models which belong to the hybrid model according to Min and Zhou’s classification.

Swaminathan, Smith and Sadeh (1998) propose a multi-agent approach to model supply chain dynamics. The different participators within the supply chain are modeled as agents that perform specific activities. The examples of generic agents could be supplier agent, manufacturer agent, distribution center agent, retailer agent, end-customer agent or transportation agent. The behaviors of each agent are specified according to its role within the supply chain. Based on the discrete-event simulator, agents will be activated when certain events occur. The multi-agent modeling approach has several advantages: (1) the modeling of the whole supply chain system is decomposed to the modeling of its participators which reduces the modeling complexity, (2) it allows the performance analysis from different organizational perspectives (Swaminathan, Smith and Sadeh 1998), and (3) it facilitates the distributed simulation structure.

Terzi and Cavalieri (2004) classify supply chain simulation into two structural paradigms: local simulation paradigm and parallel and distributed simulation (PDS) paradigm. While parallel simulation means the simulation programs are executed on multiple processors simultaneously, distributed simulation refers to running simulation programs on geographically distributed processors. Local simulation paradigm uses only one simulation model and executes on a single computer. PDS paradigm implements more models which execute on more calculation processors, each of which can be regarded as a simulation node. However, PDS does not necessarily mean that the models executing on the different processors are different. The same model could also be implemented in the PDS. Based on their survey of over 80 articles, most of the papers applied the local simulation paradigm in the simulation area of inventory, transportation and distribution. They also conclude that “PDS paradigm has not become a steady applied approach”. However, the PDS paradigm has its own advantages. Terzi and Cavalieri (2004) point out that (1) the PDS paradigm allows each node to have its own simulation model developed with different languages and run on heterogeneous platforms, (2) PDS paradigm can encompass complex simulation models that cross the enterprise boundaries without sharing local models and data, (3) PDS paradigm connects the supply chain nodes that are geographically distributed and have their own simulation model relevant to their business, and (4) sometimes the execution time of a PDS model is shorter than a single complex model.

A discussions of supply chain simulation tools can be found at Terzi and Cavalieri (2004), Kleijnen (2005), Chatfield, Harrison, and Hayya (2006) and Rossetti et al. (2006). Terzi and Cavalieri (2004) conduct a survey over 80 papers and classify them according to three different criteria: (1) scope and objec-

tives, (2) simulation paradigm and technology and (3) development stage. Kleijnen (2005) categorize and summarize supply chain simulation literature into four types (spreadsheet simulation, system dynamics, discrete-event simulation and business games). Rossetti et al. (2006) analyze some commercial simulation software with the focus models and applications and conclude that most cannot easily simulate general purpose supply chains. Chatfield, Harrison, and Hayya (2006) review the supply chain simulators from the perspective of their abilities and “ease of use to create, analyze and share supply chain models”.

The supply chain network simulator (SCNS) used in this paper is developed in Java based on an object-oriented framework for simulating supply systems. The reader who is interested in this framework can refer to Rossetti and Chan (2003), Rossetti et al. (2006), Rossetti and Nangia (2007) and Rossetti, Miman and Varghese (2008). This framework contains the classes that can be used to represent manufacturer, warehouse, retailer and end-customer. Three key layers of abstraction for the supply chain modeling are inventory layer, transport layer and facility layer. The structure and functionality of the inventory layer and transportation layer are discussed in Rossetti et al. (2006) and Rossetti and Nangia (2007) respectively. Rossetti, Miman and Varghese (2008) present an expansion of the work in Rossetti et al. (2006) and introduce the supply chain modeling with two examples. The SCNS is built on top of the Java Simulation Library (JSL), the capabilities of which are described in Rossetti (2008).

The key terms used in the SCNS are listed as follow:

- Item Type: An item type refers to a category of products stored in the network.
- Inventory Holding Point (IHP): The location that can hold inventory is called an IHP. One IHP can hold many different kinds of item types.
- Stock Keeping Units (SKU): A SKU is identified by an item type located at a particular IHP. If the item type or the IHP is different, the SKU is different. A SKU contains the inventory information of an item type stored at a specific IHP, such as the reorder policy.
- Demand Generator: The demand generator represents the customer demand. The time between customer arrivals and the amount of demand can be modeled with probability distributions. The demand generator will generate the demand according to the specified probability distributions.

A supply network consists of SKUs stocked at different locations within the supply chain. Different IHPs may be at different echelons where the IHPs in the higher echelon supply the IHPs in the lower echelon. Figure 1 illustrates a three-echelon supply network.

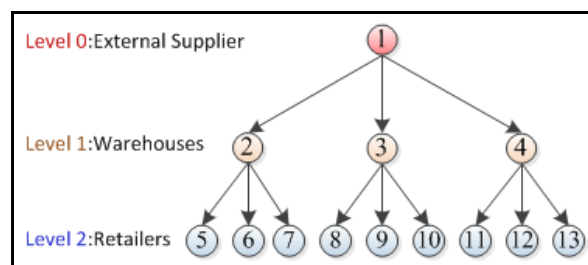


Figure 1: Three-Echelon Supply Network

This paper focuses on the simulation of supply chain networks made up of an arborescent tree of IHPs as shown in Figure 1. The root of the tree represents an external supplier. An IHP at the bottom level may have one or more demand generators attached to it. The demand generator will generate the demand according to a probability distribution, representing the arrivals of the customers. Each SKU has its own reorder policy. Take the continuous (r, q) policy for example, r stands for reorder point, q refers to reorder quantity, and continuous means that inventory position is monitored continuously. If a SKU uses a continuous (r, q) policy, when the inventory position (inventory on hand + inventory on order – backorders) declines to or below r , the related IHP will order q items from its supplier for that SKU. The transporta-

tion time from external supplier or supplier IHPs to customer IHPs is also modeled with probability distributions. The network simulated by the SCNS discussed in the paper uses the following assumptions:

1. The network has one and only one external supplier. The external supplier can supply any item type with unlimited capacity after a pre-specified manufacturing lead-time. Inventory levels at the external supplier are not tracked.
2. In the arborescent tree supply network, each IHP can have only one supplier, but it can have multiple customers.
3. The lead time to get a certain item type at the external supplier is modeled as a random variable.
4. When the order cannot be filled from on hand inventory, it is backlogged.
5. Each order will trigger a transportation request. The shipping cost between two locations is assumed to be a constant.

The SCNS is compiled into an executable jar file so that it is easy to be transferred and deployed in the cloud computing environment. Cloud computing is a model of computing which uses outside clouds to run applications. Cloud, in broad terms, refers to any infrastructure, platform or software that serves as a resource for cloud computing. The cloud computing features as well as related technologies are discussed in Vaquero et al. (2009), Wang et al. (2010) and Zhang, Cheng and Boutaba (2010). The SCNS is used in the CCAFSCNS which will be introduced in the next section.

3 SYSTEM ARCHITECTURE

A system architecture is a conceptual model that defines the system elements, the relations among elements, the structure of the system and behaviors within the system. This section presents the system architecture for the CCAFSCNS. Figure 2 shows that CCAFSCNS has 10 elements which are classified into 5 layers. Each layer is represented by a rectangle in the diagram. The connector (arrow or line) in the diagram shows the relations between two elements. Table 1 explains each relation by sequence. The following sections will discuss each layer and its elements in detail.

3.1 Data Layer

The data layer contains two system elements: Input Data and Output Data.

The input data provides the information that describes the supply chain network and specifies the simulation requirements so that the supply chain network simulator can build the network model and run the simulation. The input data is separated into different themes and stored in eight separate tables. Probability distributions table, item type table, location table, shipment table, SKU table, and demand generator table are used to describe the supply chain network. The experiment table is used to specify the requirements of the simulation experiment. The email table stores the user's email address which will be used to send simulation results.

The output data includes enough detail to let users analyze the performance of the network. The output data is generated by the SCNS after the simulation, which includes the performance measures for the whole network. The performance measures are separated and displayed into different tables based on different themes. These themes include the performance of external supplier, SKU, transportation, IHP, level (echelon), and network. The IHP, level and network performance are aggregate performances based on SKU. For each simulation replication, the average values of the performances are stored. The across replication statistics, such as mean and standard deviation, are calculated and stored in the across replication statistics table.

In order to solve the problems associated with redundancy, multiple themes and modification, the rules of relational database are used to design the tables. Seven out of eight input tables satisfy third normal form. Only the SKU table violates the third normal form for the convenience of entering data. All the output tables are designed to satisfy third normal form.

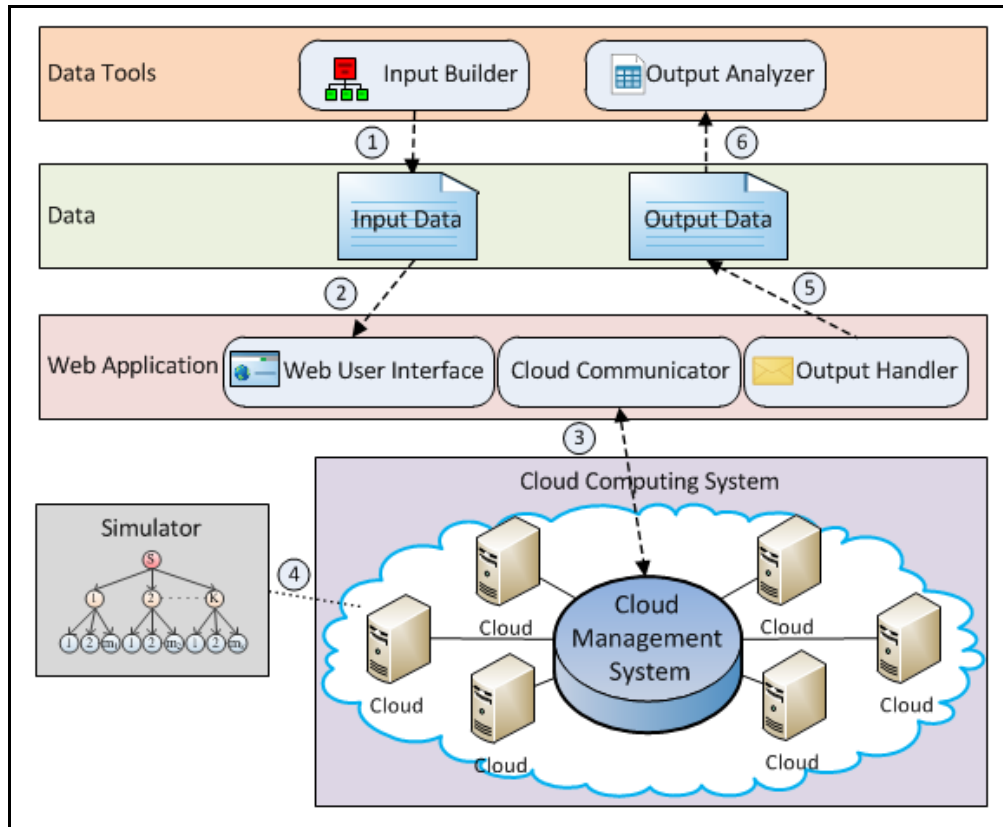


Figure 2: System Architecture

Table 1: Relation Explanation

Relation Number	Explanation
1	Input Builder generates the input data that can be read by the simulator.
2	Input data is uploaded to the web application via the Web User Interface.
3	Cloud Communicator sends the simulation request and associated input data to the Cloud Management System and receives the simulation results from it.
4	The simulator is deployed in the computing cloud.
5	Output Handler generates the output data.
6	Output data is the data source of the Output Analyzer.

3.2 Data Tools Layer

The data tools layer is made up of the Input Builder and Output Analyzer components.

The purpose of input builder is to help users develop the input data for the whole supply chain network system. A database application (DA) was developed using Access 2010 to perform the duty of the input builder. In the DA, input forms are developed to manage input processes. In order to prevent errors in the input data, the DA is designed to perform following tasks: (1) restrict the value in the table field to meet its domain, (2) build the relationships among the tables and check the referential integrity constraints, (3) guide the users to fill the table by sequence in order to avoid referential integrity problems,

and (4) add instructions in each input form and display them besides the related field to help users fill out the form. After completing the input information, users can export the input data to an Excel file.

The output analyzer is designed to help users view their simulation results and perform some specific analyses. Since all the output data are stored in the tables satisfying third normal form, the database is a good option to develop an output analyzer with its query functions. The DA developed for the input builder is also used as the output analyzer. The DA imports all the data from output Excel file and stores them in tables. Several pre-built analysis forms and charts in the DA can perform best/worst performance analysis for a selected field, ratio analysis for item types, ratio analysis of total costs, etc.

3.3 Cloud Computing System Layer

Cloud Computing System Layer contains the Clouds and the Cloud Management System.

Vaquero et al. (2009) studies more than 20 definitions of cloud and propose a definition that encompasses all related features. In this paper, the cloud refers to the computing resource that can run software applications. Using virtualization technologies, computer hardware can be partitioned into several virtual machines. A virtual machine (VM) is a software container that has its own operating system as well as applications and performs like a physical computer. The clouds selected in this paper are the VMs based on the Grid Appliance, a self-configuring VM appliance used to create pools of computer resources. Detailed descriptions of the Grid Appliance is available in Wolinsky and Figueiredo (2008). The Grid Appliance was customized to facilitate the running of Java programs. The customized VMs are started up and run in the FutureGrid, a NSF funded project that provides researchers with secure clouds. The introduction and the performance characteristics of FutureGrid can be found in Fox, Ho and Chan (2011).

A cloud management system has the responsibility to monitor the performance of the cloud, perform job queueing, schedule jobs to clouds and manage clouds. Condor, a workload management system developed by University of Wisconsin-Madison, is chosen to manage the VMs. It provides the mechanism for job queueing, scheduling, resource monitoring and resource management. A comprehensive overview of Condor is provided in Thain, Tannenbaum, and Livny (2005). A complete Condor system consists of three components: Condor Server, Condor Client and Condor Worker. The Condor management system is installed in the Condor Server, which is used to handle the job requests. The Condor Client can submit jobs to the Condor management system. The VMs run in the FutureGrid are Condor Workers, which are used to run simulation jobs. After receiving the simulation request from Condor Client, the Condor management system will find the available Condor Workers in the FutureGrid network and send the simulation jobs to them. When the simulation jobs are completed at the Condor Workers, the outputs are sent back to a folder specified by the Condor Client.

3.4 Web Application Layer

The web application layer includes the Web User Interface, the Cloud Communicator and the Output Handler.

One advantage of cloud computing solutions is that users can access the service via an Internet browser without installing any software locally. The function of the web user interface is to allow authenticated users to interact with the provided service. In this paper, the web user interface is a set of web pages that allow users submit their input data (input Excel file). Modules from Spring Framework, an open source framework for Java applications, were used to develop the web application. The Spring Security framework was used to design the login web page, and the view component of Spring MVC framework was used to develop other web pages.

The cloud communicator is used to connect the Web Application and the Cloud Management System. The cloud communicator implemented in this paper has two functions: (1) connect the Web Application and Condor Management System and (2) translate the simulation requirements to Condor commands. A Java program was developed to serve as the translator between input data and the description of Condor job. It reads the simulation requirements from input data and writes the Condor job using Condor com-

mands. When this Java program is put in a Condor Client, it becomes the cloud communicator since it has the ability to communicate with the cloud management system.

The output handler provides the mechanisms to handle both valid and invalid input data. In addition, it should report the status of simulation jobs executed in the clouds. A Java program was developed to perform the aforementioned tasks. First, the output handler will summarize the status of the simulation jobs, including the number of successful jobs and the number of errors reported by clouds. Then, the output summary and all the generated files, including simulation output file and error file, will be sent back to user via email.

In this paper, the Web User Interface, the Cloud Communicator and the Output Handler are integrated into a web application project named CloudProject. This web application project was developed using the NetBeans IDE. GlassFish, an open source application server, is used to deploy the CloudProject.

3.5 Simulator Layer

The SCNS is deployed in the cloud in the form of an executable jar file. Given the valid input data, the SCNS will build the supply chain network, run the simulation according to the experiment requirement, and generate the output data.

The simulation results are not truly “random”. In the simulation processes, the random numbers are generated by a random number generator (RNG). The purpose of the RNG is to generate a sequence of numbers that appear random by algorithms. The numbers generated by the RNG can be separated into different streams, each of which is a sequence of numbers on the range (0,1). The JSL has the ability to control the stream of random numbers. Since SCNS is built on top of the JSL, it has the ability to make the streams distinct for different replications. Given the same simulation requirements and the same streams, the simulation results are the same all the time; in this way, the simulation processes can be made repeatable. When the streams are distinct for different replications, the simulation results are different. Therefore, the capacity of controlling the streams ensures that the output from different replications are different, and the whole simulation process is repeatable.

3.6 Prototype System

A prototype system applying the CCAFSCNS has been developed as described within previous sections. It contains six components: (1) the database application (DA), (2) the supply chain network simulator (SCNS), (3) the web application named CloudProject, (4) the Condor Server, (5) the Condor Client, and (6) the Condor Workers. Before using the prototype system, the CloudProject should be deployed in the Condor Client, the SCNS should be put in a directory of the Condor Client and the Condor Worker VMs should be started up in the FutureGrid.

4 USE CASES

This section uses three example use cases to illustrate the functionality of the prototype system.

4.1 Use Case 1: Create the Input Data Using Input Builder

In use case 1, a user creates the input data for a three echelon supply chain network illustrated in Figure 1 using the DA. This use case consists of three steps as shown in Figure 3. With the conceptual model in mind, the user first opens the DA. There is an input management form (IMF) in the DA which manages the input processes. In step two, following the instructions in the IMF, the user will open an input form for each table and enter the related data. After entering all the data, the user exports the input data to an Excel file by clicking a button in the IMF which automates the exporting process using VBA code.

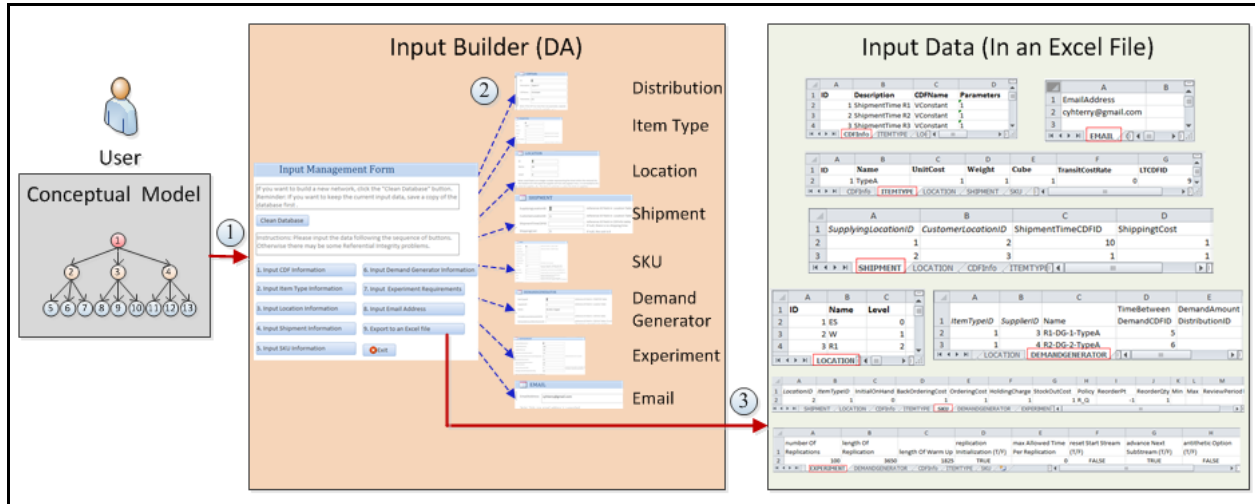


Figure 3: Use Case 1

4.2 Use Case 2: Simulate the Supply Chain Network Using Cloud Computing

The second use case is to simulate the three-echelon supply chain network using the cloud computing technology. The workflow of use case 2 is illustrated in Figure 4.

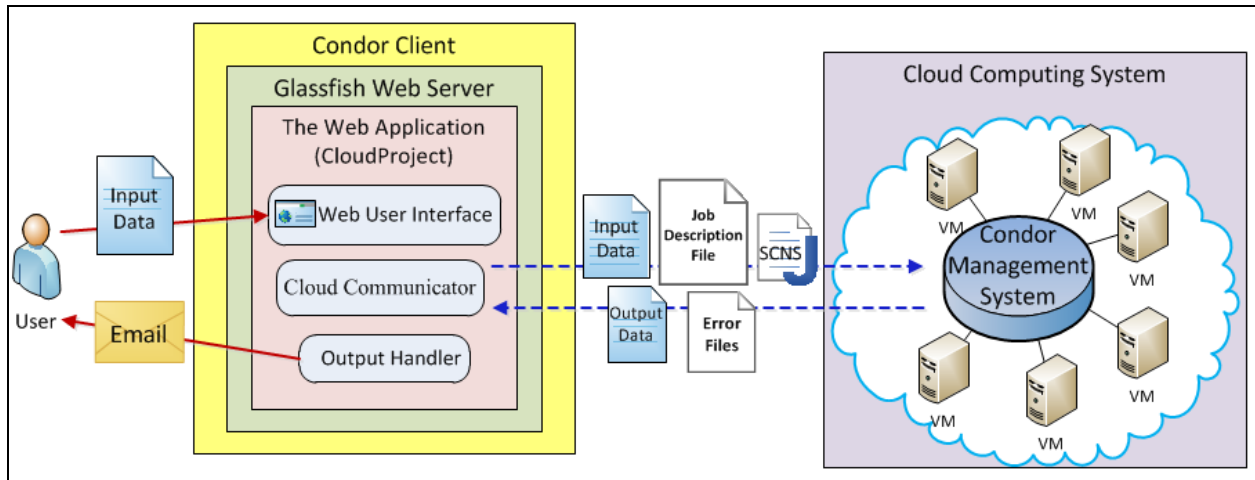


Figure 4: Use Case 2

The user only needs to log into a web page with the user name and password and upload the input Excel file generated in use case 1. The web application CloudProject is deployed in the Glassfish web server that is installed in a Condor Client. The compiled SCNS jar file is also placed in the Condor Client. When the web application receives the input Excel file from the upload page, it generates the Condor job description file based on the experiment requirements in the input Excel file. Then, the Cloud Communicator sends the input data, the job description file and the SCNS jar file to the Condor Management System. When the Condor Management System receives the files, it reads the job description file and then schedules the jobs to the available clouds (Condor Worker VMs). Each time the cloud receives a simulation job, it will use the SCNS jar file to run the simulation with input data and generate a within replication output file. If there are some Java run time errors during the simulation, the errors will be recorded in an error file. All the within replication output files and error files are transferred back to the destination

folder in the Condor Client. After all the simulation jobs are completed, the output handler performs following tasks: (1) make a summary of the job status, including the number of successful jobs and the number of error files, (2) combine all the error files into a text summary file (3) combine all the within replication output files into one Excel file, (4) calculate the mean and standard deviation of across replication statistics and store them in the same Excel file, and (5) send the summary of the job status, an Excel file with all the output data, and a text file with all the error information to the user by an email.

4.3 Use Case 3: Analyze the Output Data with Output Analyzer

In use case 3, the user wants to use the output analyzer to answer following three questions: (1) Which IHP has the highest inventory holding cost? (2) In the IHP that holds the highest inventory holding cost, which item has the highest inventory holding cost? (3) What is the total inventory cost and transportation cost for the whole supply chain network? Use case 3 is illustrated in Figure 5.

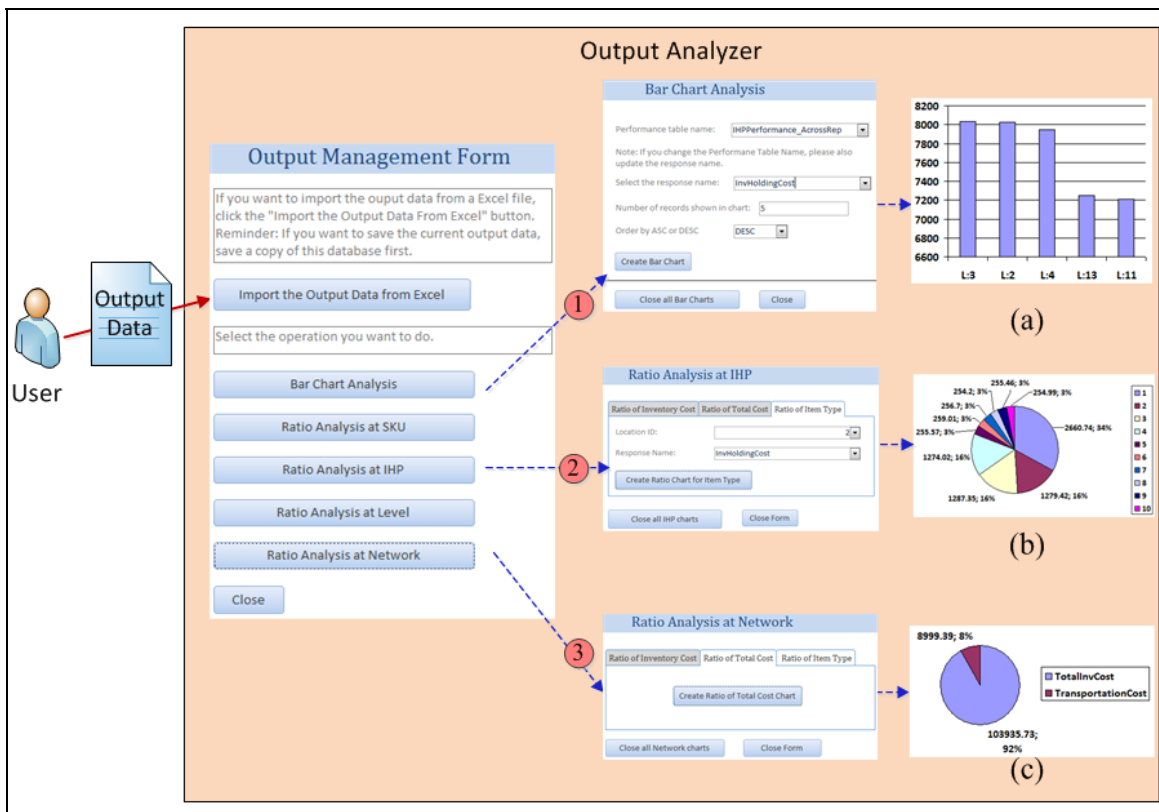


Figure 5: Use Case 3

In use case 2, the user receives the output Excel file as an attachment of the email. This output Excel file is the data source of the output analyzer. The user opens the output management form (OMF) to import output data from Excel file to the database. In the OMF, the user selects different buttons to perform different kinds of analyses in the associated forms. The user first chooses the “Bar Chart Analysis” button in the OMF and will be directed to the Bar Chart Analysis Form, which is used to find the IHP that holds the highest inventory holding cost. After entering the query criteria, the Bar Chart Analysis Form creates a bar chart, the (a) chart in Figure 5, to show the query result. The (a) chart shows that the location 3 is the answer to the first question. Then, the user selects the “Ratio Analysis at IHP” button in the OMF and will be directed to the IHP Analysis Form, which is used to perform ratio analysis for item type at location 3. In the IHP Analysis Form, choose “Ratio of Item Type” analysis and enter the query criteria. A pie chart,

the (b) chart in Figure 5, shows the query result. The (b) chart shows that item 1 has the highest inventory holding cost at location 3. Finally, the user clicks the “Ratio Analysis at Network” button in the OMF and will be led to the Network Analysis Form. The Network Analysis Form will create a Network chart, the (c) chart in Figure 5, which shows the total inventory cost and transportation cost for the whole supply chain network.

5 COMPUTATIONAL TIME STUDY

One of the main purposes of this research is to utilize cloud computing to shorten the simulation time. Therefore, we need a way to compare the time to simulate on the local computer and the time spent on the cloud computing solution. This section will first analyze the time components of the cloud computing solution and then compare the time spent on the traditional solution and the cloud computing solution.

5.1 Time Study on Cloud Computing Solution

The time spent on the cloud computing solution can be divided into four components: (1) startup time of VMs (clouds), (2) scheduling time, (3) execution time and (4) file transfer time. Each time component will be discussed in detail.

- **The Startup Time of VMs** -The startup time of VMs is the time to start up Condor Worker VMs in FutureGrid. The startup time can be calculated by the interval between the time executing the startup commands and the time when all the VMs are started up.
- **Scheduling Time** - The scheduling time is the interval between the time to submit a job and the time when the job is executed. The log file of the Condor Client records the information about events, including job submit events, job execute events, and job terminate events. The event time for each event type can be found in the log file. Within one simulation request, the time interval between the submit event and the first job execute event is calculated as the scheduling time.
- **Execution time** - The execution time is the time to run the simulation in the Condor Worker. The execution time can be found in the log file of the Condor Client.
- **File Transfer time** - File transfer time is the time to transfer the files within the prototype system. As shown in Figure 4, file transfers occur in four places. In time sequence, the file transfers happen at the place when: (1) uploading the input Excel file to the web application, (2) submitting the input Excel file, Condor job description file and other related materials from the Cloud Communicator to the Condor Management System, (3) transferring the output Excel files and error files from the Condor Management System to the Cloud Communicator, and (4) sending the email with attachments from the Output Handler to the user. The file transfer time can be estimated by the size of the total transferred files over the file transfer speed.

5.2 Comparison between the Tradition Solution and the Cloud Computing Solution

If all the simulation tasks are run in the local computer, only the execution time is needed. Thus, if the execution time on the local computer is longer than the time spent on the cloud computing solution, the purpose of shortening the simulation time by cloud computing technologies is justified. A test case was developed to compare the local simulation time and the time spent on the cloud computing solution. The structure of the network is shown in Figure 1. There are 10 items stored in this network. These items can be categorized into three types: A, B and C. 10% of items belong to Item type A that has an average demand of 1000/day/retailer. 30% of items belong to Item Type B that has an average demand of 500/day/retailer. The remaining 60% of items are Item Type C that has an average demand of 100/day/retailer. Except for the external supplier, all the locations hold all the item types; therefore, there are 120 SKUs in this network. Assuming that 30 replications are needed to satisfy the requirement of half-width. The comparison between the traditional solution and the cloud computing solution is summarized in Table 2.

In Table 2, the cloud computing solution takes 49 minutes and 20 seconds to simulate 30 replications. This value includes all four of the time components discussed in the previous section. The total time spent on the cloud computing solution for 30 replications is 35% less than the time spent on the traditional solution. During a real world cloud computing application, the startup time of VMs does not need to be included in the cloud computing solution since it only occurs once at the beginning; the VMs keep running all the time except during maintenance periods. Twenty-six minutes and thirty seconds are used to start up the VMs in FutureGrid. Without the time to start up the VMs, the cloud computing solution can save 70% of the simulation time.

Table 2: Comparison between Traditional Solution and Cloud Computing Solution

	Traditional Solution	Cloud Computing Solution
Operating System	Window 7 Enterprise	Linux
Number of Cores	2	1 for each VM; 15 in total
RAM	2 GB	1251 MB for each VM
Simulation time for 1 replication	2 minutes and 41 seconds	Around 2 minutes and 35 seconds on a VM
Total time spent on the simulation for 30 replications	75 minutes and 52 seconds	49 minutes and 20 seconds

6 CONCLUSION AND FUTURE WORK

A CCAFSCNS with 10 components is proposed in this paper to facilitate large-scale multi-echelon supply chain network simulation. A prototype system that implements the CCAFSCNS was developed with Excel, Access, Spring Framework, SCNS and the Condor System, FutureGrid, and the Grid Appliance. With the power of cloud computing technology, the simulation time of large-scale multi-echelon supply chain networks is shown to be significantly shortened.

Although the prototype system realizes the simulation in the cloud, it's far from a perfect cloud computing solution. In the prototype system, the simulation of supply chain network has been provided as a web service, but the input builder and output analyzer are used on a local computer. In addition, the input data and output data are transferred between users and the servers. A complete cloud computing solution should provide all the services from the server side; the only tool used by the user from the local computer should be a web browser and the Internet. In order to achieve a complete cloud computing solution, several challenges need to be overcome: (1) develop a GUI-based tool to build the simulation model and deploy it in the cloud, (2) store the data, both input and output, in a data server, and (3) develop the output analyzer in the cloud and analyze the data from the data server.

ACKNOWLEDGEMENTS

This material is based upon work supported in part by the National Science Foundation (NSF) under Grant No. 0910812 and No. 0437408. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. We would like to thank Dr. Renato Figueiredo and Kyungyong Lee from the ACIS Lab at the University of Florida for their extremely helpful assistance customizing the GridAppliance.

REFERENCES

Chatfield, D. C., T. P. Harrison, and J. C. Hayya. 2006. "SISCO: An Object-Oriented Supply Chain Simulation System." *Decision Support Systems* 42:422-434.

- Fox, G. C., A. Ho, and E. Chan. 2011. "Measured Characteristics of Futuregrid Clouds for Scalable Collaborative Sensor-Centric Grid Applications." In *Proceedings of the 2011 International Conference on Collaboration Technologies and Systems*, 151-160.
- Kleijnen, J. P. C. 2005. "Supply Chain Simulation Tools and Techniques: A Survey." *International Journal of Simulation and Process Modelling* 1:82-89.
- Min, H., and G. Zhou. 2002. "Supply Chain Modeling: Past, Present and Future." *Computers & Industrial Engineering* 43: 231-249.
- Rossetti, M. D., and H. T. Chan. 2003. "A Prototype Object-Oriented Supply Chain Simulation Framework." In *Proceedings of the 2003 Winter Simulation Conference*, Edited by S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, 1612-1620. Piscataway, New Jersey: IEEE, Inc.
- Rossetti, M. D., M. Miman, V. Varghese, and Y. Xiang. 2006. "An Object-Oriented Framework for Simulating Multi-Echelon Inventory Systems". In *Proceedings of the 2006 Winter Simulation Conference*, Edited by L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, 1452-1461. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Rossetti, M. D., and S. Nangia. 2007. "An Object-Oriented Framework for Simulating Full Truckload Transportation Networks." In *Proceedings of the 2007 Winter Simulation Conference*, Edited by S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, 1869-1877. Piscataway, New Jersey: IEEE, Inc.
- Rossetti, M. D. 2008. "Java Simulation Library (JSL): An Open-Source Object-Oriented Library for Discrete-Event Simulation in Java." *International Journal of Simulation and Process Modelling* 4:69-87
- Rossetti, M. D., M. Miman, and V. Varghese. 2008. "An Object-Oriented Framework for Simulating Supply Systems." *Journal of Simulation* 2:103-116.
- Swaminathan, J. M., S. F. Smith, and N. M. Sadeh. 1998. "Modeling Supply Chain Dynamics: A Multi-Agent Approach." *Decision Sciences* 29:607-632.
- Terzi, S., and S. Cavalieri. 2004. "Simulation in the Supply Chain Context: A Survey." *Computers in Industry* 53:3-16.
- Thain, D., T. Tannenbaum, and M. Livny. 2005. "Distributed Computing in Practice: the Condor Experience." *Concurrency and Computation: Practice and Experience* 17:323-356.
- Wolinsky, D. I., and R. J. Figueiredo. 2008. "Simplifying Resource Sharing in Voluntary Grid Computing with the Grid Appliance." In *Proceedings of the 2008 IEEE International Parallel & Distributed Processing Symposium*, 1-8.
- Vaquero, L. M., L. Rodero-Merino, J. Caceres, and M. Lindner. 2008. "A Break in the Clouds: Towards a Cloud Definition." *Computer Communication Review* 39:50-55.
- Wang, L., G. Von Laszewski, A. Younge, X. He, M. Kunze, J. Tao, and C. Fu. 2010. "Cloud Computing: a Perspective Study." *New Generation Computing* 28:137-146.
- Zhang, Q., L. Cheng, and R. Boutaba. 2010. "Cloud Computing: State-of-the-Art and Research Challenges." *Journal of Internet Services and Applications* 1:7-18.

AUTHOR BIOGRAPHIES

MANUEL D. ROSSETTI is a Professor in the Industrial Engineering Department at the University of Arkansas. He received his Ph.D. in Industrial and Systems Engineering from The Ohio State University. His research and teaching interests are in the areas of simulation modeling, logistics optimization, and inventory analysis applied to manufacturing, distribution, and health-care systems. He serves as an Associate Editor for the International Journal of Modeling and Simulation and is active in IIE, INFORMS, and ASEE. He served as co-editor for the WSC 2004 and 2009 conference. His email and web addresses are rossetti@uark.edu and <http://www.uark.edu/~rossetti/>.

YAOHUA CHEN is a M.S. Candidate in Industrial Engineering at the University of Arkansas. His research interests include supply chain modeling, inventory control and transportation management.