Design of the Integrated Logistics Information System Based on Cloud Computing

Minjeong An and Hongchul Lee

Abstract—This study presents the practical approach for designing the integrated logistic information system (ILIS) in cloud computing environment. It is intended to optimize resources (vehicles, human resources, oil, etc.) required to logistics management. This system is targeted at the small and medium-sized third-party logistics (3PL) companies which are difficult to establish IT system by providing a remote service environment like SaaS (Software as a Service). In this study, the concepts traveling salesman problem (TSP) and Dynamic Programming (DP) are used for route optimization. We intend to contribute to minimize the operational cost of the logistics system by designing remote service.

Index Terms—Logistics management system, transportation management system, traveling salesman problem, dynamic programming, cloud computing, SaaS, PaaS.

I. Introduction

The integrated logistics information system has contributed to reduce logistics cost and to enhance the efficiency of the logistics business, after the adoption of logistics company in the early 2000s. But the significant parts of the system components are relying on expensive logistics solutions. In addition, the logistic companies wish to integrate the load optimization solution with the route optimization solution within a single system. Therefore, the integrated logistics information system (ILIS) is necessary to the logistics company. But most of third-party logistics companies are small and medium-sized scale, so they are hard to establish and operate IT system in the respect to the cost. In this study, we suggest the design of the main components of ILIS in the shape of SaaS in cloud computing environment.

II. CONCEPTUAL DESIGN

Logistics business is made up of many tasks until complete delivery after receiving order of the customer. Fig. 1 shows the overall logistics business process. After receiving customer's order, a logistics company plans the delivery schedules with consideration for delivery destinations and appointment time. In the case of long-distance transportation such as between the central distribution centers (CDC) and the regional distribution centers (RDC), the load optimization is important because the number of operational vehicles can

Manuscript received February 15, 2014; revised April 15, 2014. This work was supported by the BK 21 plus fund 2014.

Minjeong An and Hongchul Lee are with the Industrial Management Engineering Department, Korea University, Korea (e-mail: ansebby@korea.ac.kr, hclee@korea.ac.kr).

be significantly reduced. In the case of regional delivery, the route optimization is required because it has multiple delivery points from RDC. Warehouse management system makes it more efficient through accurate inventory management and scheduling. The freight charges between shippers and logistic companies should be computed and the path of the vehicle by GPS (Global Positioning System) also should be monitored. By monitoring vehicle movement, route breakaway can be prevented and a sudden accident can be taken action quickly. In order to forecast freight volume and to reflect in the delivery schedule, the analysis of completed delivery data is also necessary.

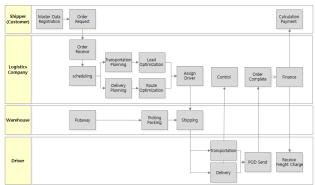


Fig. 1. Overall logistics business process.

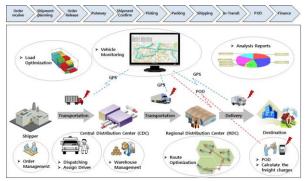


Fig. 2. Overview of the ILIS.

Describing in terms of system, the stakeholders of the ILIS are shippers, logistics companies, warehouse managers and vehicle drivers. The main function of the system consists of master data management, order management, transportation management, warehouse management, freight management, vehicle monitoring, analysis reports, mobile business management, and system management. The core functions are considered to delivery scheduling, dispatching, load optimization for transportations between CDCs and RDCs, route optimization for delivery to customers. Fig. 2 is overview of the ILIS. This shows interaction between main functions of the system. As shown in Fig. 2, the ILIS is essential to the logistics business because real time data

processing between stakeholders is important.

III. MATERIAL AND METHOD

A. Cloud Computing

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [1]. Gartner defines cloud computing as "a style of computing where massively scalable IT-enabled capabilities are delivered 'as a service' to external customers using Internet technologies" [2]. Essential Characteristics of cloud computing are on-demand self-service, broad network access, resource pooling, rapid elasticity and measured service [1], [3]. The cloud computing service is classified to public cloud, private cloud, hybrid cloud and community cloud according to deployment target

[1]. And it is classified to SaaS, PaaS and IaaS according to service model. Table I describes three service models and four deployment models. In this study, we propose design of the ILIS as a shape of SaaS in hybrid Cloud environment. In this study, we propose design of the ILIS as a shape of SaaS in hybrid Cloud environment. And we suggest development and deployment environment of the ILIS to use PaaS. The choice of PaaS should be preceded before designing system because every vendor has some technical constraints and vendor dependency. To choose PaaS for the ILIS, we analyzed the features of PaaS of two vendors. Table II describes comparison of the most famed PaaS. Microsoft Windows Azure is more powerful than Google App Engine in terms of database and function and pricing and supported language. But App Engine provides more useful free services. Development and deploy environment is also importance in terms of development productivity and efficient operation of system. So we need to test sample function at each environment.

TABLE I: SERVICE MODELS AND DEPLOYMENT MODELS OF CLOUD CIMPUTING

Type	Model	Definition	
Service Models	SaaS	Software as a Service : SaaS provides application software as a shape of online service. (e.g., SalesForce CRM, Google Apps, DeskAway, Impel CRM, Wipro w-SaaS)	
	PaaS	Platform as a Service : PaaS is a broad collection of application infrastructure services (including application platform, integration, business process management and database services) (e.g., Microsoft Windows Azure, Google App Engine, Force.com (SalesForce), OrangeScape, Wolf PaaS)	
	IaaS	Infrastructure as a Service : IaaS delivers computer infrastructure (including servers, software, data-center space and network equipment) as a service.	
		(e.g., Amazon's Elastic Compute Cloud (EC2), Google Cloud Storage)	
Deployment Models	Private cloud	A private cloud is a particular model of cloud computing that involves a distinct and secure cloud based environment in which only the specified client can operate.	
		(e.g., Windows Azure)	
	Community cloud	The cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations) [1].	
	Public cloud	A public cloud is the standard cloud computing model, in which a service provider makes resources, such as applications and storage, available to the general public over the internet.	
		(e.g., Amazon Web Services , IBM SmartCloud, Oracle Cloud, Google App Engine, Windows Azure)	
	Hybrid cloud	The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability [1].	

B. Traveling Salesman Problem (TSP)

The traveling salesman problem (TSP) is to find the minimum cost path passing through all the given n cities exactly once [4]–[8]. It is one of the most widely studied combinatorial optimization problems [5]. It is a well-known NP-hard problem [4]. The exact algorithms for solving TSP, such as integer linear programming (ILP), branch-and-bound (BB) algorithm and dynamic programming (DP) have been proven [5]. These methods could not be practically used due to their exponential time complexity in the worst case [4].

Therefore, one stream of research has consisted of developing heuristic algorithms with a guaranteed worst-case performance [5]. The heuristic algorithms to finding near optimal solution for the TSP are genetic algorithm (GA) [9], ant colony optimization (ACO) [10], simulated annealing

(SA) [11], tabu search (TS) [12], neural network (NN) [13]. But they did not guarantee to find the optimal solution.

The multiple traveling salesman problem (MTSP) is a generalization of the TSP, where more than one salesman is allowed to be used in the solution [14]. MTSP extends the problem to a wide variety of vehicle routing problems (VRPs) by incorporating some additional side constraints such as vehicle capacity limits, delivery time windows, as well as time dependent speeds in metropolitan area [14].

In this study, supposing single driver and the load already was assigned to the vehicle, we adopted dynamic programming to solve TSP for route optimization.

C. Dynamic Programming (DP)

Dynamic programming is a technique for efficiently implementing a recursive algorithm by storing partial results

[15]. In this approach we solve small instances first, store the results, and later, whenever we need a result, look it up instead of recomputing it [4]. In dynamic programming algorithms, we construct a solution from the bottom up in an array (or sequence of arrays) [4]. Because the principle of optimality applies in a Shortest Paths problem, we can develop a recursive property that gives an optimal solution to an instance in terms of optimal solutions to subinstances [4].

TABLE II: COMPARISON OF THE MOST FAMED PAAS

	Google App Engine	MS Windows Azure	
Base Plan Price	\$0.08 per hour	\$0.02 per hour	
Runtimes	Java, Python, PHP, Go	.NET(C#, VB, etc), Java, Node, Perl, PHP, Python, Ruby	
Features	Google Cloud SQL	Auto Scaling	
	Datastore	Cloud Storage	
	Blobstore	Database as a Service	
	Search	Deploy Servers	
	Memcache	Disaster Recovery	
	Dedicated Memcache	Horizontal Scaling	
	Logs	Load Balancing	
	HRD Migration Tool	Vertical Scaling	
		VPN Access	
Native Services	Google Cloud Datastore	-	
	Google Cloud Storage		
Native Databases	MySQL	Microsoft SQL	
		MongoDB	
		MySQL	
Control Interface	Web Based Application/Control Panel		
	API (Application Programming Interface)		

Fig. 3 shows a weighted, directed graph. The circles represent vertices, and a line from one circle to another represents an edge. An arrow shows the direction. If the edges have values associated with them, the values are called weights [4].

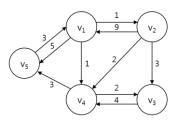


Fig. 3. A weighted, directed graph [4].

We represent a weighted graph containing n vertices by an array W where

$$\mathbf{W}[i][j] = \begin{cases} \text{weight on edge if there is an edge from } v_i \text{ to } v_j \\ \infty & \text{if there is an edge from } v_i \text{ to } v_j \\ 0 & \text{if } i = j. \end{cases}$$

The graph in Fig. 3 is represented by adjacency matrix like below [4].

$$W = \begin{bmatrix} 0 & 1 & \infty & 1 & 5 \\ 9 & 0 & 3 & 2 & \infty \\ \infty & \infty & 0 & 4 & \infty \\ \infty & \infty & 2 & 0 & 3 \\ 3 & \infty & \infty & \infty & 0 \end{bmatrix} \quad D = \begin{bmatrix} 0 & 1 & 3 & 1 & 4 \\ 8 & 0 & 3 & 2 & 5 \\ 10 & 11 & 0 & 4 & 7 \\ 6 & 7 & 2 & 0 & 3 \\ 3 & 4 & 6 & 4 & 0 \end{bmatrix}$$

The matrix D is the distance matrix. It contains the lengths of the shortest paths in the graph [4]. We accomplish this by creating a sequence of n+1 arrays $D^{(k)}$, where $0 \le k \le n$ and where $D^{(k)}[i][j]$ is the length of a shortest path from v_i to v_j using only vertices in the set $\{v_1, v_2, \dots, v_k\}$ as intermediate vertices [4]. We can determine $D^{(k)}$ from $D^{(k-1)}$ as follows [4]:

$$D^{(k)}\left[i\right]\!\left[j\right] = minimum\!\left(D^{(k-1)}\!\left[i\right]\!\left[j\right], D^{(k-1)}\!\left[i\right]\!\left[k\right] + D^{(k-1)}\!\left[k\right]\!\left[j\right]\right)$$

The array P is the optimal path matrix [4]. When at least one intermediate vertex exists, P[i][j] is the highest index of an intermediate vertex on the shortest path from v_i to v_j [4]. If no intermediate vertex exists, it is zero [4]. The graph in Fig. 3 is represented by the optimal path matrix like below [4].

$$P = \begin{bmatrix} 0 & 0 & 4 & 0 & 4 \\ 5 & 0 & 0 & 0 & 4 \\ 5 & 5 & 0 & 0 & 4 \\ 5 & 5 & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 \end{bmatrix}$$

Using the above dynamic programming, we can get the optimal solution, which the minimum distance is 14 and the shortest paths are {1-2-3-4-5-1}.

In the case of a straightforward enumerative of paths, the time complexity is is O(n!) [4]. In the case of dynamic programming, the time complexity can be reduced to $O(n^22^n)$ [4]. The memory complexity is $O(n2^n)$ [4], [16]. Although the complexity was reduced, this method is feasible when n is small [4].

In logistics area, the vertices can be represented to destinations for delivery and the edge is distance between each destination. Because we supposed the load already was assigned to the vehicle, the number of vertices are not large. Moreover our system is going to be run in cloud computing environment, we can get more efficient performance than single local computing environment.

IV. SYSTEM DESIGN

We designed the architecture of the ILIS based on cloud computing shown as Fig. 4. Infrastructure layer is a physical layer which is running ILIS. Application layer is the environment which services are developed and deployed and testing. This layer has vendor dependency. Therefore we need detailed investigation about technology constraint according to the vendor of PaaS. Service Layer consists of our services for logistics business. We designed service components based on service oriented architecture (SOA).

Master data management component performs management about base data for running ILIS such as customer, product, vehicle, driver, destination, zone, and so on. Order management component consists of functions that receive, modify and complete order from customer. Transportation management component consists of functions such as delivery scheduling, the route optimization and the load optimization.

These components are shape of remote service. Therefore these are independent and interoperable between each component.

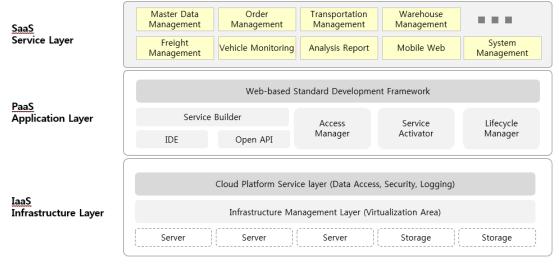


Fig. 4. The architecture of the ILIS based on cloud computing.

V. CONCLUSIONS AND FUTURE WORKS

The ILIS based on cloud computing designed in this study can contribute to the optimization of logistics operations through offer to the small and medium sized logistic company by an economical cost. The function of the load optimization can reduce a number of operation vehicles through maximize vehicle capacity. The function of the route optimization can reduce the driving distances through suggestion about shortest paths before starting delivery. If the core functions are serviced by mobile web, logistics work can be more efficiently performed by real time. The systems of solving optimization problem which need to large amount of hardware resources such as CPU and memory can be run more efficiently in cloud computing environment.

Our future works are consideration about security and data collision because this system is sharing by many logistics company. And after developing the route optimization, we are going to do quantitative measurement about performance in cloud computing environment.

ACKNOWLEDGMENT

This work was supported by the BK21 Plus (Big Data in Manufacturing and Logistics Systems, Korea University).

REFERENCES

- P. Mell and T. Grance, The NIST Definition of Cloud Computing, NIST Special Publication SP800-145-2011.
- [2] Cloud Computing: Defining and Describing an Emerging Phenomenon, Stamford, CT: Artner, Inc., 2008.
- [3] G. Reese, Cloud Application Architectures, Sebastopol, CA: O'Reilly, 2009, pp. 1-46.
- [4] R. E. Neapolitan and K. Naimipour, Foundations of Algorithms using C++ Pseudocode, 3rd ed., New York, U.S.A.: Jones and Bartlett, 2004, ch. 3, pp. 91-133.
- [5] L. Gilbert, "The traveling salesman problem: An overview of exact and approximate algorithms," *European Journal of Operational Research*, vol. 52, no. 2, pp. 570-578, July 1993.
- [6] Y. Dumas, J. Desrosiers, E. Gelinas, and M. M. Solomon, "An optimal algorithm for the traveling salesman problem with time windows," *Operational Research*, vol. 43, no. 2, pp. 367-371, Apr. 1995.

- [7] A. Langevin, F. Soumis, and J. Desrosiers, "Classification of travelling salesman problem formulations," *Operations Research Letters*, vol. 9, no. 2, pp. 127-132, Mar. 1990.
- [8] G. B. Dantzig, D. R. Fulkerson, and S. M. Johnson, "On a linear-programming, combinatorial approach to the traveling-salesman problem," *Operations Research*, vol. 7, no. 1, pp. 58-66, Feb. 1959.
- [9] S. Chatterjee, C. Carrera, and L. A. Lynch, "Genetic algorithms and traveling salesman problems," *European journal of operational* research, vol. 93, no. 3, pp. 490-510, Sep. 1996.
- [10] M. Dorigo and L. M. Gambardella, "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53-66, Apr. 1997.
- [11] P. P. Yip and Y. H. Pao, "Combinatorial optimization with use of guided evolutionary simulated annealing," *IEEE Transactions on Neural Networks*, vol. 6, no. 2, pp. 290-295, Mar. 1995.
- [12] J. Knox, "Tabu search performance on the symmetric traveling salesman problem," *Computers & Operations Research*, vol. 21, no. 8, pp. 867-876, Oct. 1994.
- [13] C. K. Looi, "Neural network methods in combinatorial optimization. Computers & Operations Research," *Computers & Operations Research*, vol. 19, no. 3, pp. 191-208, May 1992.
- [14] T. Bektas, "The multiple traveling salesman problem: an overview of formulations and solution procedures," *Omega*, vol. 34, no. 3, pp. 209-219, June 2006.
- [15] S. S. Skiena, *Dynamic Programming*, 2nd ed., London, U.K.: Springer-Verlag, 2008, ch. 8, pp. 273-315.
- [16] R. Bellman, "Dynamic programming treatment of the travelling salesman problem," *Journal of the ACM*, vol. 9, no. 1, pp. 61-63, Jan. 1962.



Hongchul Lee was born in Seoul, Korea. He obtained the Ph.D. in the Department of Industrial Engineering at Texas A&M University, USA in 1993. He is currently working as a professor in the Department of Industrial Management Engineering at Korea University, Korea since 1996.

His research interests are production information system and logistics information system and Supply Chain Management (SCM)



Minjeong An was born in Seoul, Korea. She obtained the master degree in the Department of Industrial Management Engineering at Korea University, Korea in 2008. She is currently a Research Assistant of the Department of Industrial Management Engineering at Korea University besides working on her Ph.D. degree.

Her research interests are logistics information system and cloud computing architecture