

# 클라우드 환경에서 멀티 노드들의 최적 경로 탐색을 위한 양자화 데이터 전송

오현창<sup>1</sup> · 김재권<sup>1</sup> · 김태영<sup>1</sup> · 이종식<sup>1\*</sup>

## Quantization Data Transmission for Optimal Path Search of Multi Nodes in cloud Environment

HyungChang Oh · JaeKwon Kim · TaeYoung Kim · JongSik Lee

### ABSTRACT

Cloud environment is one in the field of distributed computing and it consists of physical nodes and virtual nodes. In distributed cloud environment, an optimal path search is that each node to perform a search for an optimal path. Synchronization of each node is required for the optimal path search via fast data transmission because of real-time environment. Therefore, a quantization technique is required in order to guarantee QoS(Quality of Service) and search an optimal path. The quantization technique speeds search data transmission of each node. So a main server can transfer data of real-time environment to each node quickly and the nodes can perform to search optimal paths smoothly. In this paper, we propose the quantization technique to solve the search problem. The quantization technique can reduce the total data transmission. In order to experiment the optimal path search system which applied the quantized data transmission, we construct a simulation of cloud environment. Quantization applied cloud environment reduces the amount of data that transferred, and then QoS of an application for the optimal path search problem is guaranteed.

**Key words** : Cloud Computing, Optimal Path Search, Quantization Method, Data Transmission

### 요약

클라우드 환경은 분산컴퓨팅 분야의 한가지로서, 물리 노드와 가상 노드로 구성이 되어 있다. 분산화 된 클라우드 환경에서의 최적 경로 탐색은 각 노드들이 최적 경로 탐색을 수행하는 것이다. 실시간으로 급변하는 탐색 환경은 빠른 데이터 전송을 통한 각 노드들의 동기화를 요구한다. 따라서 QoS의 보장과 최적 경로 탐색을 위해서 양자화 기법이 필요하다. 양자화 기법을 통해 중앙 서버는 각 노드로 실시간 탐색 환경 데이터를 빠르게 전송가능하며 각 노드들은 원활하게 최적 경로 탐색을 수행할 수 있다. 본 논문에서는 중앙 서버에서 각 노드들의 최적 경로 탐색 문제를 해결하기 위해 데이터의 전송량을 줄일 수 있는 양자화를 적용한다. 최적 경로 생성 시스템에 양자화 데이터 전송을 적용하는 실험을 하기 위해 클라우드 환경의 시뮬레이션을 구성하였다. 양자화 기법의 적용을 통해 클라우드 환경에서 전송 되는 총 데이터를 줄이면서 성능을 높일 수 있으며, 최적 경로 탐색을 위한 어플리케이션의 QoS를 보장할 수 있다.

**주요어** : 클라우드 컴퓨팅, 최적 경로 탐색, 양자화 기법, 데이터 전송

## 1. 서론

\*이 논문은 2102년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업(2012R1A1A2002751) 및 방위사업청과 국방과학연구소의 지원으로 수행되었습니다.(UD110006MD)

접수일(2013년 2월 1일), 심사일(2013년 5월 27일),

게재 확정일(2013년 6월 9일)

<sup>1)</sup> 인하대학교 컴퓨터정보공학과

주 저 자 : 오현창

교신저자 : 이종식

E-mail; jslee@inha.ac.kr

클라우드 컴퓨팅은 분산컴퓨팅의 한 분야로서, 시스템 가상화 및 인터넷 기술의 발전과 함께, 새로운 컴퓨팅 플랫폼으로 부상하고 있다(Armbrust 등, 2010). 클라우드의 인프라 환경은 물리 노드와 가상 노드로 구성되어 있으며, 각 어플리케이션이나 비즈니스 마다 구성방법이 다양하다(김재권, 이종식, 2011). 따라서 클라우드 공급자는 상

호 연결된 여러 멀티 노드로 컴퓨팅 시스템을 구성한다. 클라우드 컴퓨팅의 주요 목표는 QoS(Quality of Service)를 제공하여 사용자를 만족시키고, 클라우드 제공자의 이익을 향상시키는 것이다(Buyya 등, 2009).

클라우드 환경은 멀티 노드들로 구성되어 있어 있는 인프라 환경으로 각각의 최적 경로 탐색을 수행하는 노드와 중앙 시스템으로 구성되어 있다. 중앙 시스템에서 각 노드에게 최적 경로 탐색을 위한 지형 정보 및 여러 데이터를 제공한다. 데이터는 고정적이지 않으며 상황에 따라 실시간으로 변한다. 노드들은 이러한 실시간으로 변하는 지형 정보에 여러 알고리즘을 이용하여 최적 경로 탐색 문제를 해결한다(오현창, 이종식, 2011).

기존의 최적 경로 생성 방법 중 한 가지로 지도 데이터에 단일 프로세서를 이용하여 최적 경로를 생성하는 방법이 있다(이성용 등, 2010). 하지만 단일프로세서에 의한 처리는 다수의 이동객체가 동시에 경로를 요청할 경우 최적 경로 생성에 지연이 발생하며, 이로 인하여 요청의 처리시간이 길어진다. 따라서 실시간으로 변하는 상황에서의 최적 경로 탐색 요청을 정확하게 처리하기 어렵다.

위와 같은 문제를 극복하기 위한 기존의 방법으로는 경로 생성을 다수의 노드를 이용한 분산처리로 하는 방법이 있다(박상혁, 김재훈, 2009). 실시간으로 변화되는 환경에서의 최적 경로 탐색 문제를 서로 연결이 되어 있는 다수의 노드들을 이용하여 분산처리 할 경우, 실시간으로 변하는 탐색 환경에서 많은 수의 최적 경로 요청을 원활하게 처리가 가능하다.

하지만 클라우드 환경에서의 최적 경로 탐색 문제는 경로를 생성해야 하는 멀티 노드들이 동일한 지형 데이터로 동기화가 되어 있어야 한다. 일반적으로 멀티 노드들은 물리 노드 혹은 가상 노드로 구성되어 있으며 지리적으로 떨어져 있기 때문에 네트워크를 통하여 통신을 한다. 지형 정보는 실시간으로 변하기 때문에 짧은 주기로 클라우드를 구성하는 네트워크상의 모든 노드에게 전송되어야 한다.

최적 경로의 정확도를 높이기 위해 지형 정보의 종류를 늘릴 경우 처리해야 하는 데이터의 양은 기하급수적으로 늘어난다. 따라서 기존의 분산처리 환경을 클라우드 시스템에 접목을 시킬 경우 지형 데이터를 전송하는 시간이 오히려 최적 경로 생성에 걸리는 시간보다 더 길어질 뿐만 아니라, 네트워크 상황에 따라 부하가 발생하여 전체 시스템의 성능을 저하시킬 수도 있다.

이러한 문제를 해결하기 위해서는 최적 경로 탐색에 이용하는 지형 정보가 전체 노드에 동기화 되는 시간을

단축시킬 필요가 있다. 각 노드들에게 최적 경로 생성에 쓰이는 정보의 전달을 원활하게 하기 위해서는 지형 정보를 구성하는 데이터의 양을 줄여서 기존 시스템에서의 문제점인 동기화 오버헤드를 줄여야 한다. 또한 줄어든 데이터를 복원하여 신뢰성이 높은 최적 경로를 생성해야 한다.

본 논문에서는 클라우드 환경에서 각 노드들의 최적 경로 탐색 문제를 해결하기 위해 중앙 서버에서 각 노드로 전송되는 지형 데이터에 양자화(Quantization) 기법(Zeigler 등, 2002)을 적용하였다. 최적 경로 생성 시스템에 양자화 데이터 전송을 적용하는 실험을 하기 위해 DEVS(Discrete Event System Specification)(Zeigler 등, 2000)을 이용한 클라우드 환경의 시뮬레이션을 구성하였다.

## 2. 관련 연구

### 2.1 클라우드 환경 기반의 최적 경로 생성

클라우드 환경은 서로 다른 물리적 위치에 존재하는 노드들을 통합하는 가상화(Visualization) 기술을 통해 구현이 된다. 클라우드 환경은 서비스의 목적에 따라 SaaS(Software as a Service), PaaS(Platform as a Service), IaaS(Infrastructure as a Service)로 나누어진다. 또한, 자원 사용량에 따른 과금 체계인 유틸리티 컴퓨팅, 대용량 데이터를 처리하기 위한 그리드 컴퓨팅 등 최근의 기술 동향과 서로 많은 관련이 있다(민영수 등, 2009).

클라우드 환경기반의 최적 경로 생성 시스템은 특성상 많은 수의 서비스 요청을 동시에 처리할 수 있어야 한다. 기존의 단일 프로세스 방식과 중앙처리 방식은 동시다발적인 다수의 최적 경로 생성 요청을 하나의 노드가 처리해야 하기 때문에 부하가 높다. 또한 탐색해야 하는 경로의 신뢰성을 높이기 위해 지형 데이터의 종류를 세밀화 하면 각 요청에 따르는 처리의 부담이 커진다. 그리고 실시간으로 변화하는 지형 상황에 따라 즉각적으로 경로의 수정이 이루어 질 경우 다수의 탐색 요청을 처리하는 데에 부담이 더욱 가중된다.

기존의 클라우드 환경기반의 최적 경로 생성 구조는 사용자로부터 요청 받는 경로 생성 작업을 다수의 가상 노드들이 처리를 하기 때문에 단일 프로세스 방식이나 중앙처리 방식보다 빠른 처리가 가능하다. 하지만 지형 데이터를 각각의 모든 노드에게 전송 하여 동기화 시켜야 한다. 실시간으로 변하는 지형 정보에 맞추어 효율적인 경로를 생성하기 위해서는 정보가 변경되는 즉시 모든 노드에게 데이터를 다시 전송해야 하며 이는 경로를 생성하는 것과 별도로 추가적인 오버헤드를 발생시킨다.

기존의 클라우드 환경기반 최적 경로 생성 시스템은 이러한 데이터 전송에 따른 시간지연의 문제점을 가지고 있다. 지형 데이터의 종류가 적고 경로를 생성하는 노드의 수가 많지 않을 경우 모든 노드의 데이터 동기화 오버헤드는 미미하며 무시가능한 수준이다. 하지만 생성된 경로의 신뢰성을 높이기 위해 지형 데이터의 종류를 늘리고 시스템 전체의 성능을 높이기 위해 노드의 수를 늘릴수록 데이터 동기화에 따른 오버헤드는 점점 커진다. 이러한 오버헤드는 전체 클라우드 시스템의 성능을 저하(정진립 등, 1998)시킨다.

## 2.2. 데이터 양자화 전송 기법

데이터의 양자화는 용량이 많은 데이터를 압축하는 기법으로 대용량 데이터의 처리나 특징 추출, 데이터 축소 등에 많이 사용되며 손실 기법과 무손실 기법으로 분류된다. 무손실 압축방법은 원본 데이터의 내용이 압축과 압축해제 과정을 거친 후에도 완전 동일하지만 압축과정을 통해서 크게 데이터를 줄일 수 없다는 단점이 있다. 반면에 손실 압축방법은 원본 데이터의 내용이 압축과정에서 변화가 일어나 압축을 해제 하면 원본 데이터와 차이를 보이지만 높은 압축률을 얻을 수 있는 장점이 있다. 높은 압축률을 얻기 위해 양자화 기법에서는 주로 손실압축 방법을 이용한다(반승원 등, 2001). 손실 압축기법의 단점인 원본 데이터의 손실을 줄이면서 높은 압축률을 달성하기 위한 연구가 활발하게 진행되고 있다(Eickeler 등, 2000).

이러한 연구들은 손실된 데이터에서 최대한 특징을 추출하는 방법에 초점이 맞추어져 있다. 손실압축이 적용된 지형 데이터는 실제 지형 정보와 차이를 보이기 때문에 경로 생성 시스템에 요구된 성능에 영향을 끼친다. 이에 따라 지형 데이터의 특징을 해석하여 실제 데이터와의 차이를 최대한 줄이면서 만족할 만한 압축효과를 가질 방법이 필요하다(김태영, 이종식, 2009). 분산 환경에서의 데이터 양자화 기법은 빠른 압축을 기대할 수 있으며, 시스템에서 요구하는 특징적인 부분을 추출하여 전송하기 때문에 높은 신뢰성을 가진다(Kim 등, 2007). 따라서 본 논문에서는 분산 환경에서의 데이터 양자화 기법을 클라우드 환경의 최적 경로 생성에 적용한 시스템을 제안한다.

## 3. 클라우드 환경에서 최적 경로 탐색을 위한 양자화 기법

이 장에서는 클라우드 환경에서 최적 경로 탐색을 위

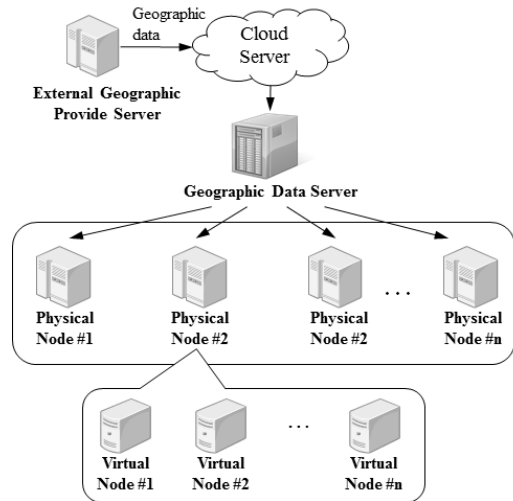


Fig. 1. Structure of Cloud

한 시스템의 구성도를 제공하며, 데이터 양자화 기법에 대해 설명을 한다. 클라우드 환경에서의 최적 경로 탐색을 위한 구조는 Fig. 1과 같다.

클라우드 서버는 지형 데이터 서버와 물리 노드 및 멀티 노드로 구성되어 있다. 외부 지형 정보 제공 서버에서 지형 데이터를 클라우드 서버로 전송을 하면, 지형 데이터 서버는 이 정보를 저장 하게 된다. 그리고 지형 정보가 바뀌게 될 경우를 대비해 주기적으로 갱신을 하게 된다. 물리 노드와 가상 노드는 최적 경로를 탐색하기 위한 프로세서이며, 지형 데이터 서버로부터 지형 정보를 업데이트를 받아 처리하게 된다. 즉, 지형 데이터 서버는 지형 정보만을 저장하는 장소이며, 이 정보를 기반으로 노드들이 작업을 수행하게 된다. 본 논문에서는 멀티 노드의 구조를 1개의 물리 노드에 복수개의 가상 노드가 포함되어 있는 구조로 정의한다.

### 3.1 시스템 구조

최적 경로 탐색 및 데이터를 양자화하기 위해서는 그에 해당하는 모듈에 대한 설명이 필요하며 그에 따라 본 장에서는 클라우드 환경에 적용할 수 있는 모듈에 대해 설명을 한다. 데이터를 양자화가 적용된 데이터로 변환하고 저장 및 전송을 하기 위한 클라우드 시스템 구조는 Fig. 2와 같다.

외부 지형 정보 제공 서버는 실시간으로 업데이트 되는 지형 정보이며, 이 데이터가 클라우드 서버에 입력이 된다. Transmission For Data는 외부로부터 들어오는

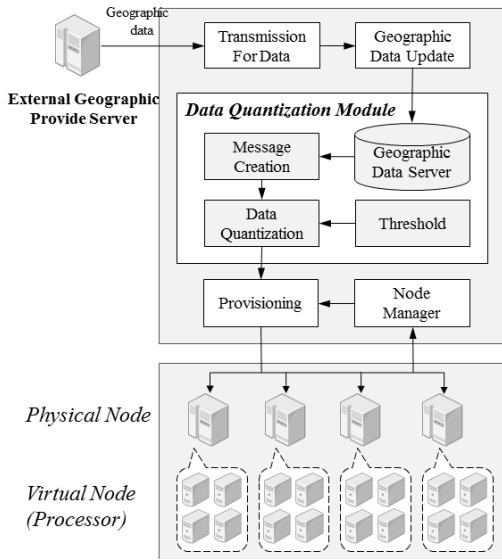


Fig. 2. Structure of Cloud System

XML이나 Binary, Ratio 등의 지형 데이터를 클라우드 서버가 처리할 수 있는 형태로 변환 시키게 된다. Geographic Data Update는 변환된 데이터를 저장 가능한 형태로 쿼리를 생성하여 Geographic Data Server에 전송을 하게 된다.

Data Quantization Module은 지형 데이터를 저장하는 Geographic Data Server와 멀티 노드에 데이터를 전송하기 위한 Message Creation, 데이터를 양자화 시키기 위한 Data Quantization과 지형 데이터의 압축을 위한 임계값을 정의하는 Threshold로 구성이 된다. Geographic Data Server는 실시간의 지형 정보를 저장하는 역할을 한다. Geographic Data Server는 대용량의 지형 데이터를 저장하기 위한 공간이며 멀티 노드들에게 데이터를 전송시키기 위한 데이터베이스이다. Message Creation은 각 노드에게 업데이트된 지형 데이터를 제공해 주기 위한 모듈이다. Data Quantization은 전송할 메시지를 양자화 하는 모듈이며, 임계값을 기반으로 하여 데이터를 양자화 한다.

데이터 양자화가 완료되면, 각 노드에 메시지를 전송하기 위해서 스케줄링을 해야 하며, Provisioning이 메시지를 각 노드들에게 전송하는 역할을 맡는다. Node Manager는 각 노드들의 상태를 파악하고, 처리할 수 없는 노드는 따로 분류하는 작업을 한다.

멀티 노드환경은 물리 노드와 가상 노드로 구분이 되며, 하나의 물리 노드에 복수개의 가상 노드가 구성되어 있다. 가상 노드는 실제 메시지를 전송 받고 최적 경로 탐색을 수행하는 프로세서이다.

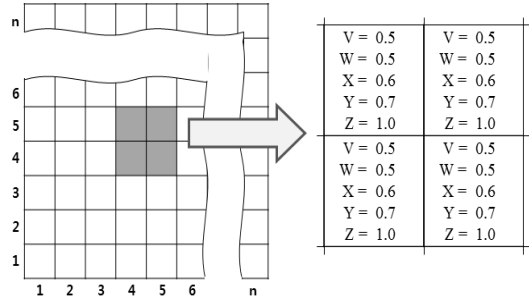


Fig. 3. Map Data

### 3.2 지형 데이터 양자화

지형 데이터는 도메인 혹은 어플리케이션에 따라 다양하게 구성이 될 수 있으며, 각 지형마다 정보의 형태가 다르다. 본 논문에서는 2차원 격자 형태의 지형을 기초로 하며 Fig. 3과 같이 구성된다.

지도를 일정간격의 격자 형태로 나눈 뒤 각 칸에 해당하는 지형이 가진 특성이 지형 데이터로 5가지의 입력 변수로 구성(이성용 등, 2010)이 된다. 각각의 지형이 가지고 있는 정보는 지면의 특성을 반영하는 속도 가중치(V), 기후(W), 이동객체의 밀도(X), 지형의 고도(Y), 해당 지역의 사고유무(Z) 값들이며, 지형의 특성에 따라 서로 다른 값을 가진다.

멀티 노드들은 모두 동일한 지형 데이터를 Geographic Data Server로부터 전송 받아서 그 데이터를 바탕으로 최적 경로를 생성한다. 실제 지형의 상황이나 최적 경로 탐색 문제에 사용되는 데이터는 실시간으로 변한다. 외부 지형 정보 제공 서버로부터 전달받은 최신 지형 데이터와 노드가 현재 가지고 있는 데이터가 차이를 보일 경우, Geographic Data Server는 메시지를 생성한 후, Provisioning을 통해 각 노드에게 전송한다.

지형 데이터를 멀티 노드에 보내기 위한 메시지를 생성하는 과정은 Fig. 4와 같다.

기존의 메시지 생성과정은 모든 지형 데이터를 메시지로 만들어 보내야 하기 때문에, [1,1]부터 [n,n]까지의 모든 데이터를 메시지로 변환하여 전송해야 한다. 하지만 최근에 변화가 일어난 데이터만 갱신을 시키더라도 최적 경로 탐색에는 지장이 없다. 그러므로 메시지의 길이를 줄이기 위해서 양자화를 통하여 값이 변경된 지형 데이터만을 전송한다. 데이터를 양자화 하기 위한 방법은 Table 1과 같다.

초기 값은 Geographic Data Server에 저장되어있는 지형 데이터 Map과 새로운 지형 데이터를 입력 받는

Update\_Map, 각 지형 마다 저장되어있는 정보 V, W, X, Y, Z로 구성이 된다. 그리고 데이터 양자화를 거쳐 최종적으로 데이터를 전송해야 하는 Message로 구성되어있다.

데이터 양자화는 외부로부터 전송 받은 지형 정보가 업데이트 될 때 발생을 한다. 현재의 지형 정보와 업데이트 된 지형의 정보가 다를 경우 이를 감지한 후, 일정한

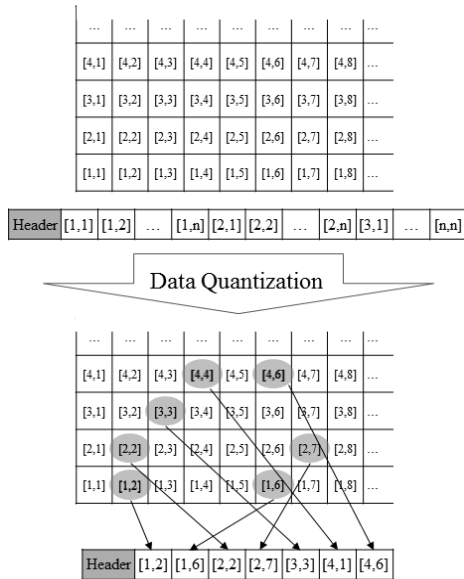


Fig. 4. Message Generation from Map Data

임계값(Threshold)과 비교한다. 만약에 임계값 이상의 차이가 있을 경우, 바뀐 정보만을 메시지에 넣는다. 메시지의 헤더부분은 지도상 어느 위치에 적용을 해야 하는지에 대한 정보가 입력되며, 그 다음으로는 각 지형 데이터의 값이 입력된다. 이후, 새로운 지형 정보를 현재의 지형 정보에 업데이트 시킨 뒤, 완성된 메시지를 제공한다.

양자화 된 메시지의 구조는 Fig. 5와 같다. 양자화 된 메시지는 기존의 메시지와는 다르게 길이가 일정하지 않으며, V~Z까지의 변화된 정보만 메시지에 포함이 된다.

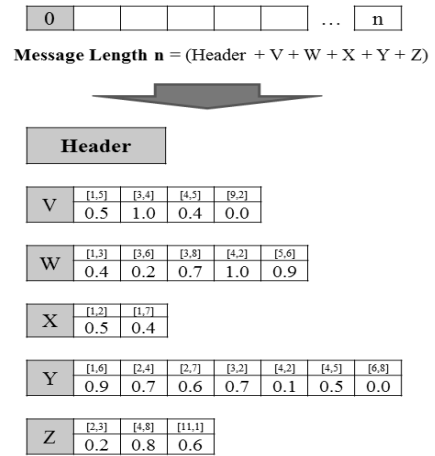


Fig. 5. Structure of Quantized Message

Table 1. Data Quantization Algorithm

**Initial State**

Map[X\_distance][Y\_distance][Value]  $\leftarrow$  [1,1] to [n,n] Current Map Data and Value Information  
 Update\_Map[X\_distance][Y\_distance][Value]  $\leftarrow$  [1,1] to [n,n] Update New Data and Update Value Information  
 Value[V, W, X, Y, Z][Value]  $\leftarrow$  each Map Information  
 [V, W, X, Y, Z]  $\leftarrow$  Detail Map Information  
 Message[]  $\leftarrow$  Message Information

**Function** Data Quantization(Map, Value) **return** Message[]

**If** Update to Map Information

**for** Map [1,1] to [i, j] **do**

**while** Compare Map to Update\_Map **do**

**If** Map[i,j]  $\neq$  Update\_Map[i, j]

**If** ( | Map[i,j] - Update\_Map[i,j] | > Threshold)

Message[0]  $\leftarrow$  (const) Map[i][j] // Header

Message[1]  $\leftarrow$  (const) Value[V][Value] to Value[Z][Value] // each Map Information

**end If**

Map[i, j][Value]  $\leftarrow$  Update Map[i, j][Value]

**return** Message[]

**end If**

**end for**

**end If**

그렇기 때문에 헤더부분은 전체 길이에 대한 정보를 추가로 가지고 있어야 하며 각 최적 경로 생성에 사용되는 각 지형 데이터(V, W, X, Y, Z)의 헤더 역시 길이 정보를 가지고 있어야 한다. 데이터의 경계를 알려주는 이 두 가지 헤더는 메시지 전체를 놓고 봤을 때 미미한 수준이며 크기가 고정되어 있다.

반면 양자화 된 각각의 데이터 값 바로 앞에 있는 좌표 정보는 변화가 일어난 전장 데이터에 비례하여 증가하기 때문에 양자화에 따른 가장 큰 오버헤드로 작용을 한다. 일반적인 상황에서 지형 데이터의 변화는 양자화 오버헤드를 감안하더라도 기존의 시스템 보다 효율적이지만 양자화를 통한 시스템의 성능 향상을 최대한 보장하기 위해 전체 메시지의 크기는 작을수록 좋다(오현창, 이종식, 2011).

### 3.3 임계값 설정

데이터 양자화는 지형 데이터를 메시지로 구성할 때 변화가 일어난 정보를 대상으로 최적 경로 탐색에 영향을 미칠 수 있는 수준의 변화가 일어난 영역을 선별하여 메시지를 구성함으로써, 각 노드로 전송되는 데이터의 크기를 줄여준다. 지형 데이터는 특성상 빈번하게 변하지만 값 자체의 크기는 미세하게 변화는 양상을 보인다. 일반적으로 이러한 작은 변화는 최적 경로에 큰 영향을 미치지 못한다.

최적 경로에 영향을 주지 않는 미세한 변화를 필터링

하기 위해 임계값을 지정하였다. 변화된 지형 데이터가 임계값을 넘지 않는 한도 내에서 변했을 경우, 노드로 전송될 메시지에 포함 시키지 않는다.

임계값을 지정하기 위해서는 도메인 어플리케이션마다 Heuristic을 적용하는 것이 일반적이지만, 도메인 지식이 없는 경우에는 계산적인 결과를 바탕으로 추정한다(Zeigler 등, 2002). 따라서 본 논문에서도 특정한 어플리케이션을 적용하지 않기 때문에 계산적인 결과에 따라 임계값을 지정한다.

## 4. 실험 및 결과

본 논문에서는 데이터 양자화를 적용한 클라우드 환경에서 최적 경로 탐색 시스템에 대한 실험환경을 구성하고, 그 성능을 측정하기 위해 근접 비율(Proximity Ratio), 데이터 전송량(Data Transmission Rating), 프로세싱 시간(Processing Time)을 비교한다.

### 4.1 실험 환경

최적 경로 탐색 시스템을 위한 클라우드 환경의 구성을 위해 DEVS형식론(Zeigler 등, 2000)을 사용하였으며, Fig. 6과 같이 구성하였다.

실험환경은 Geographic Data Provider와 SaaS and PaaS Environment, IaaS Environment로 구성하였다.

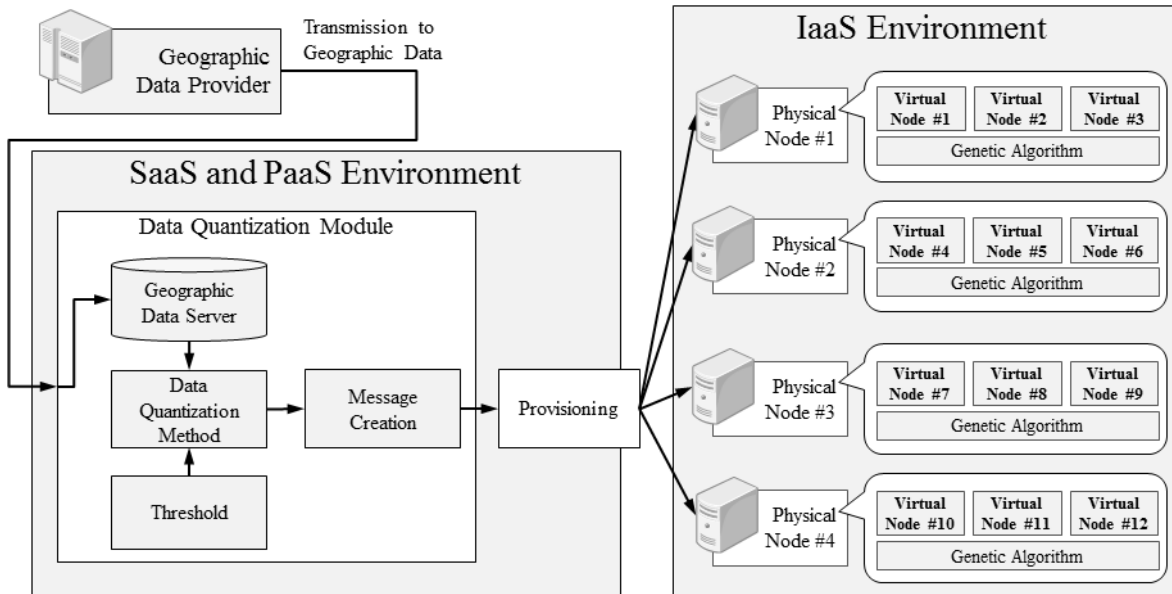


Fig. 6. Experimental Environment

Geographic Data Provider는 실시간으로 변하는 지형 데이터를 클라우드 서버에 전송을 하게 된다. SaaS and PaaS Environment는 지형 데이터를 받으면 데이터 저장 및 양자화를 통해 메시지를 생성한다. 그리고 Provisioning을 통해 각 노드들에게 전송을 한다. IaaS Environment는 멀티 노드를 의미하며, 각 물리 노드당 3개의 가상 노드를 가지고 있다. 총 4개의 물리 노드로 구성했으며, 총 12개의 가상 노드로 구성되어 있다. 그리고 가상 노드별 최적의 경로를 탐색하기 위한 메소드는 Genetic Algorithm (이성용 등, 2010)을 사용하였다.

지형 데이터를 위한 지도의 구성은 가로 16, 세로 16의 크기를 가진 지도로 구성하였다. 각 지도는 5개의 정보 변수 V, W, X, Y, Z를 가지고 있으며, Geographic Data Provider에서는 해당 변수들의 변경된 내용을 전송하게 된다. Provisioning은 생성된 메시지를 각 노드에게 보내는 역할을 하며, 모든 노드에 동일한 메시지를 전송한다. 멀티 노드는 모두 동일한 성능을 가지고 있으며 물리 노드는 경로 생성 작업을 하지 않고 가상 노드가 데이터를 처리하게 된다.

지형 데이터의 변화는 각각의 변수의 특성에 맞게 변화도록 설정하였다. 지면의 특성을 반영하는 속도 가중치(V)의 경우에는 거의 변하지 않지만 기후(W)에 따라 순간 변할 수 있으며 기후는 확률적으로 급변할 수 있다. 이 동객체의 밀도(X)는 특정 구간 특정 시간대에 점진적으로 증가하다가 다시 감소하며 사고유무(Z)에 영향을 받는다. 지형의 고도(Y)는 거의 변하지 않지만 기후에 따라 사고유무의 확률을 높이며 밀도가 높은 경우 역시 사고의 확률은 높아진다.

실험은 처음에는 지형 데이터를 가상 노드에 전송을 하며 최적 경로의 탐색이 시작된다. 일정한 시간 마다 Geographic Data Provider는 지형 정보를 전송한다. 업데이트가 된 지형 정보는 다시 가상 노드로 전달이 되고 변경된 지형 데이터에 맞추어 최적 경로의 재탐색을 진행한다. 실험의 종료는 모든 노드가 최종 목적지까지 최적 경로의 탐색을 완료하였을 경우로 한다.

각각의 이동객체는 출발지부터 목적지까지 생성된 최적경로를 따라 이동한다. 실험의 복잡성을 줄이기 위해 객체의 이동방향은 2개 방향(북쪽, 동쪽)으로 제한하였다. 이에 따라 객체는 목적지까지 이동하는데 총 30번의 이동을 하게 된다. 객체가 한 칸을 이동한 시점에서 다시 목적지로의 최적경로를 탐색하며 모든 객체가 목적지에 도달하면 시스템을 종료한다.

실험은 클라우드 환경에서 양자화를 적용시키지 않은

최적 경로 탐색 모델(Non-Quantization Model)과 본 논문에서 제시하는 양자화를 적용한 탐색 모델(Quantization Model)을 비교 분석하였다. 양자화를 적용한 탐색 모델의 경우 각 임계값에 따른 실험 결과의 변화 역시 분석하기 위해 실험을 통해 계산된 최적의 양자화 임계값인 0.2 이외에 0.3, 0.4 도 함께 실험을 하였다.

#### 4.2 실험 결과

제안하는 모델의 성능을 측정하기 위해 정확도 확인을 위한 근접 비율(Proximity Ratio), 데이터 전송량(Data Transmission Rating), 프로세싱 시간(Processing Time)을 비교한다.

첫 번째로, 근접 비율(Proximity Ratio)은 탐색이 된 최적 경로가 얼마나 정확하냐 나타내는 지표이다. 근접비율은 실제 알려진 최적 경로와 제안하는 모델이 탐색한 경로를 비교하는 것으로 식 (1)과 같으며, 근접 비율에 대한 결과는 Fig. 7과 같다.

$$Proximity(\%) = \left( \frac{T - |t - T|}{T} \right) * 100 \quad (1)$$

$T$  = 실제 최적 경로 이용시 소요시간

$t$  = 양자화 모델 이용시 소요시간

근접비율은 양자화를 거친 데이터가 최적 경로 생성에 얼마나 영향을 주는지를 보여주는 그래프이다. 가로축은 경로를 요청한 객체의 이동 횟수를 나타내며, 전송 받은 지형 데이터를 이용하여 생성된 경로가 실제 최적 경로에 얼마나 근접하는지 세로축에 표시한다. 양자화 임계값 0.2를 사용한 모델은 뒤에 나올 Fig. 8의 그래프에 보이는 데로 총 메시지의 양이 50%수준으로 줄어들었음에도 불구하고 실제 최적 경로에 거의 근접하다.

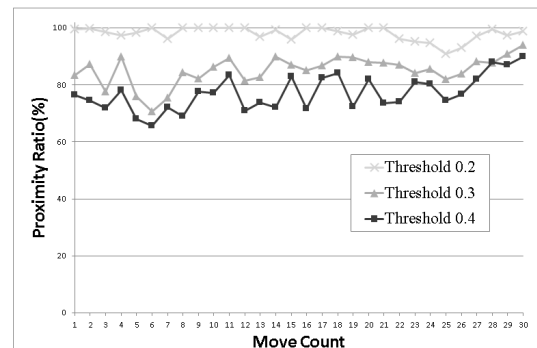


Fig. 7. Proximity Ratio

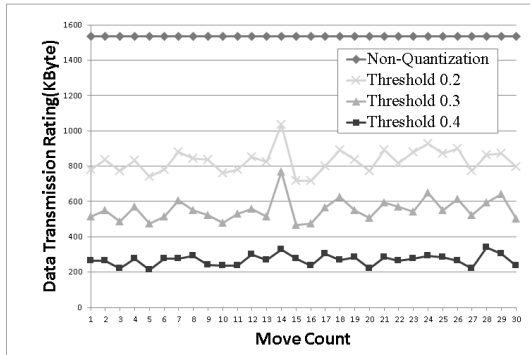


Fig. 8. Data Transmission Rating

실험 전체에 걸쳐 이동객체는 30번의 이동을 하였으며 이에 따라 30번 최적 경로가 생성 되었다. 평균 1.89%의 오차가 발생하였다. 30번의 경로 생성 중 과반수가 넘는 18번의 경로가 평균보다 높은 근접비율을 보였고 최악의 상황인 25번째 생성된 최적 경로 역시 9.03%의 경로차이만이 발생하였다. 실험 결과로 볼 때 양자화 임계값 0.2를 이용해 생성된 최적 경로의 신뢰성은 높다고 할 수 있다. 반면에 양자화 임계값 0.3, 0.4를 이용하여 생성된 경로들은 각각 14.36%, 22.14%의 비교적 큰 평균 오차율을 가졌다.

두 번째로, 데이터 전송량(Data Transmission Rating)은 양자화를 통해 데이터의 총 전송량이 얼마나 줄었는지를 확인할 수 있는 지표이며, KByte 단위의 데이터의 전송량을 측정한다. 데이터 전송량의 결과는 Fig. 8과 같다.

가로축은 경로를 요청한 객체의 이동 횟수를 나타내며 세로축은 전송된 메시지 크기를 나타낸다. 양자화를 적용하지 않는 모델의 경우, 매번 정확히 같은 크기의 지형 데이터를 노드가 받기 때문에 일정한 형태를 보이며 약 46,000 KByte를 전송하였다. 제안하는 모델은 변화가 일어난 지형 데이터 중에서 임계값에 못 미치는 변화는 제외한 메시지를 전송하기 때문에 매번 일정하지 않은 크기의 메시지가 생성이 된다. 앞에서 신뢰성이 높은 것으로 평가되었던 양자화 임계값 0.2를 사용한 모델은 총 24,500 KByte를 전송하였다. 결과적으로 양자화를 적용한 모델은 기존 모델에 비해 약 53% 수준의 메시지를 생성하게 된다. 양자화 임계값 0.3, 0.4는 비록 신뢰도는 떨어지지만 각각 데이터 전송량을 약 36%, 17% 수준으로 크게 줄일 수 있다.

마지막으로, 프로세싱 시간(Processing Time)은 Geographic Data Server로부터 멀티 노드가 메시지를 받은 시간부터

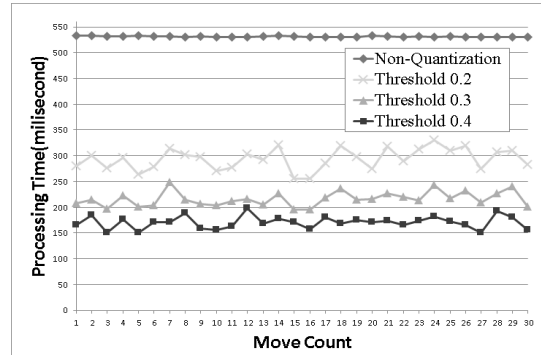


Fig. 9. Processing Time

최적 경로의 생성이 완료될 때까지의 시간을 millisecond 단위로 측정한다. 프로세싱 타임의 결과는 Fig. 9와 같다.

각 모델의 프로세싱 타임 즉, 경로 생성 시간 그래프의 차이는 네트워크계층으로부터 받은 메시지를 멀티 노드가 최적 경로를 생성 하는데 쓰일 수 있는 데이터로 변환하는 과정에서 대부분 발생하였다. 양자화를 적용하지 않은 모델은 항상 같은 크기의 메시지를 받아 지형 데이터로 변환을 하기 때문에 최적 경로 생성 시간 역시 매번 큰 폭의 변화가 없는 약 530ms를 보여주고 있다.

반면 양자화 임계값 0.2를 사용한 모델은 경로 생성에 평균적으로 294ms가 걸린다. 하지만 양자화의 영향으로 메시지의 크기가 가변적이기 때문에 매 경로 생성 시간의 차이가 커서 최고 약 75ms까지의 차이가 보인다. 프로세싱 타임은 양자화를 적용하지 않은 모델에 비해서 약 44% 정도 단축되었다.

## 5. 결론

본 논문은 클라우드 환경에서 멀티 노드를 통하여 최적 경로를 탐색하기 위해 지형 데이터의 양자화를 제안한다. 제안된 모델은 다수의 최적 경로의 요청을 멀티 노드를 이용하여 처리 하며, 경로를 생성하는 모든 노드들의 지형 데이터의 동기화를 위해 양자화 기법을 적용하였다.

양자화 기법은 최적 경로 생성에 이용되는 지형 정보 중 변화가 일어난 데이터만을 골라 미리 정해 놓은 임계값보다 큰 변화가 일어난 데이터를 메시지로 구성한다.

양자화 기법을 통해 네트워크를 이용하여 각 노드들에게 전해지는 메시지의 총 양을 양자화가 적용되지 않은 시스템에 비해 약 53% 수준으로 줄여 지형 데이터의 동기화에 따른 시스템의 성능저하와 네트워크의 과부하를



방지하였다. 크기가 줄어든 메시지에 의해서 각 노드당 최적 경로의 생성에 걸리는 시간이 약 44% 감소하였다. 줄어든 총 메시지의 양에도 불구하고 클라우드 시스템에서 생성된 최적 경로는 실제 최적 경로들 보다 평균 1.89%의 낮은 최적 경로와의 오차를 보여주고 있기 때문에 신뢰성이 높다.

따라서 본 논문에서 제안한 지형 데이터 양자화 기법이 클라우드 환경에서 멀티 노드를 이용한 최적 경로 생성 시스템의 성능을 크게 향상 시키면서 생성되는 최적 경로의 오차를 합리적인 수준으로 유지 시킬 수 있다는 것을 증명하였다. 또한 클라우드 환경의 탐색 문제 해결을 위한 어플리케이션에 적용이 가능할 것이라고 생각한다.

## References

1. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., "A View of Cloud Computing", *Communication ACM*, Vol. 53, No. 4, pp. 50-58, 2010.
2. Ban, S.W., Kim, B.J., Suk, J.Y., Kwon, S.G., Kwon, K.G., Kim, Y.C., Lee, K.I., "Efficient Multispectral Image Compression Using Variable Block Size Vector Quantization", *Journal of the Institute of Electronics Engineers of Korea*, Vol 38-SP, No. 6, pp. 105-113, 2001.
3. Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J., Brandic, I., "Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility", *Future Generation Computer Systems*, Vol. 25, Issue 6, pp. 599-616, 2009.
4. Eickeler, S., Muller, S., Rigoll, G., "Recognition of JPEG Compressed Face Images Based on Statistical Methods", *Image and Vision Computing*, Vol. 18, Issue 4, pp. 279-287, 2000.
5. Jung, J.L., Woo, Y.J., Jung, C.S., "Distributed Simulation Environment Using WWW", *Korea Society for Simulation, Proceedings of the 1998 Spring Conference*, pp. 96-100, 1998.
6. Kim, I.K., Jang, S.H., Lee, J.S., "Adaptive and Mobility-predictive Quantization-based Communication Data Management for High Performance Distributed Computing", *Simulation*, Vol. 83, No. 7, pp. 529-547, 2007.
7. Kim, J.K., Lee, J.S., "Virtual Machine Provisioning Scheduling with Conditional Probability Inference for Transport Information Service in Cloud Environment", *Journal of the Korea Society for Simulation*, Vol. 20, No. 4, pp. 139-147, 2011.
8. Kim, T.Y., Noh, C.H., Lee, J.S., "Fuzzy Logic-based Bit Compression Method for Distributed Face Recognition", *Journal of the Korea Society for Simulation*, Vol. 18, No. 2, pp. 9-17, 2009.
9. Lee, S.Y., Jang, S.H., Lee J.S., "Modeling and Simulation of Optimal Path Considering Battlefield-situation in the War-game Simulation", *Journal of the Korea Society for Simulation*, Vol. 19, No. 3, pp. 27-35, 2010.
10. Min, Y.S., Kim, H.Y., Kim, Y.K., "Distributed File System for Cloud Computing", *Communication of the Korea Information Science Society*, Vol 27, No. 5, pp. 86-94, 2009.
11. Oh, H., Lee, J.S., "Optimal Path Generation System Using Distributed Simulation for War-game Moving Objects", *Korea Society for Simulation, Proceedings of the 2011 Spring Conference*, pp. 52-57, 2011.
12. Park, S.H., Kim, J.H., "Real-Time NCW Systems using Distributed Processing", *Korean Institute of Information Scientists and Engineers, Proceeding of the Korea Computer Congress 2009*, Vol. 36, No. 1(B), pp. 245-250, 2009.
13. Zeigler, B.P., Cho, H.J., Kim, J.G., Sarjoughian, H.S., Lee, J.S., "Quantization-based Filtering in Distributed Discrete Event Simulation", *Journal of Parallel and Distributed Computing*, Vol. 62, No. 11, pp. 1629-1647, 2002.
14. Zeigler, B.P., Praehofer, H., Kim, T.G., *Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems*, 2nd Edition, Academic Press, pp. 76-96, 2000.



**오 현 창** (windbust@hotmail.com)

2011 인하대학교 컴퓨터공학부 학사  
2011~현재 인하대학교 컴퓨터정보과 석사과정

관심분야 : 클라우드 컴퓨팅, 분산처리, 모델링&시뮬레이션



**김 재 권** (jaekwonkorea@naver.com)

2011 가천의과대학교 정보처리과 학사  
2011~현재 인하대학교 컴퓨터정보공학과 석사과정

관심분야 : 클라우드 컴퓨팅, 인공지능, 모델링&시뮬레이션



**김 태 영** (silverwild@gmail.com)

2007 인하대학교 컴퓨터공학부 학사  
2009 인하대학교 정보공학과 석사  
2009~현재 인하대학교 컴퓨터정보공학과 박사과정

관심분야 : 시스템 모델링 % 시뮬레이션, 분산처리



**이 종 식** (jslee@inha.ac.kr)

1993 인하대학교 전자공학과 학사  
1995 인하대학교 전자공학과 석사  
2001 미국 애리조나대 전기·컴퓨터공학과 박사  
2001~2002 캘리포니아 주립대학교 전기·컴퓨터공학과 전임강사  
2002~2003 클리블랜드 주립대학교 전기·컴퓨터공학과 조교수  
2003~현재 인하대학교 컴퓨터정보공학부 교수

관심분야 : 소프트웨어공학, 모델링&시뮬레이션