

<http://dx.doi.org/10.7236/IIBC.2015.15.6.187>

IIBC 2015-6-26

## 개인화된 건강 데이터의 대량 처리 모니터링을 위한 메시지 모델 및 동적 버퍼 할당 설계

### Design of Dynamic Buffer Assignment and Message model for Large-scale Process Monitoring of Personalized Health Data

전영준\*, 황희정\*\*

Young-Jun Jeon\*, Hee-Joung Hwang\*\*

**요약** ICT 헬싱플랫폼은 만성질환 예방을 목적으로 하며 개인의 생체신호 및 생활습관 등의 정보에 기반을 둔 질환 초기 경보를 목표로 한다. 이를 위한 2-step 개방형 시스템(TOS)에는 헬싱플랫폼과 개인건강데이터 저장소간의 중계가 설계되었으며 데이터 처리과정을 실시간으로 전송(모니터링)하기 위한 대량 커넥션 기반의 publish/subscribe(pub/sub) 서비스가 고려되었다. 그러나 TOS pub/sub의 초기 설계에서는 커넥션 메시지를 deflate 알고리즘으로 인코딩하기 위해, 커넥션의 유휴(idle) 여부 및 메시지의 종류에 상관없이 동일한 버퍼를 할당한다. 본 논문의 동적 버퍼 할당은 다음과 수행된다. 우선 각 커넥션의 메시지 전송 유형을 큐잉하고, 각 큐는 tf-idf를 통해 특징(feature)추출 연산 후 벡터로 변환하여 k-means 클러스터에 입력하여 군집을 생성한다. 특정 군집으로 분류된 커넥션은 해당 군집의 자원 테이블에 따라 자원을 재할당 한다. 이때 각 군집의 센트로이드(centroid)는 해당 군집을 대표하는 큐잉 패턴을 사전에 선택하여 자원참조 테이블(버퍼 크기별 인코딩 효율)로 도출한다. 제안된 설계는 TOS의 인코딩 버퍼 자원을 네트워크 커넥션에 효율적으로 배분하기 위해, 군집 및 특징 연산을 위한 연산 자원과 네트워크 대역폭 간의 trade-off를 수행함으로써 TOS의 tps(단위 시간당 실시간 데이터 처리 모니터링 연결수)를 높이는데 활용할 수 있다.

**Abstract** The ICT healing platform sets a couple of goals including preventing chronic diseases and sending out early disease warnings based on personal information such as bio-signals and life habits. The 2-step open system(TOS) had a relay designed between the healing platform and the storage of personal health data. It also took into account a publish/subscribe(pub/sub) service based on large-scale connections to transmit(monitor) the data processing process in real time. In the early design of TOS pub/sub, however, the same buffers were allocated regardless of connection idling and type of message in order to encode connection messages into a deflate algorithm. Proposed in this study, the dynamic buffer allocation was performed as follows: the message transmission type of each connection was first put to queuing; each queue was extracted for its feature, computed, and converted into vector through tf-idf, then being entered into a k-means cluster and forming a cluster; connections categorized under a certain cluster would re-allocate the resources according to the resource table of the cluster; the centroid of each cluster would select a queuing pattern to represent the cluster in advance and present it as a resource reference table(encoding efficiency by the buffer sizes); and the proposed design would perform trade-off between the calculation resources and the network bandwidth for cluster and feature calculations to efficiently allocate the encoding buffer resources of TOS to the network connections, thus contributing to the increased tps(number of real-time data processing and monitoring connections per unit hour) of TOS.

**Key Words** : ICT Healing platform, dynamic buffer, k-means cluster, vert.x

\*정회원, IDLE CO.,LTD.

\*\*정회원, 가천대학교 IT대학 컴퓨터공학과(교신저자)

접수일자: 2015년 10월 13일, 수정완료: 2015년 11월 13일

게재확정일자: 2015년 12월 11일

Received: 13 October, 2015 / Revised: 13 November, 2015 /

Accepted: 11 December, 2015

\*\*Corresponding Author: hwanghj@gachon.ac.kr

Dept. of Computer Engineering College of Information Technology,  
Gachon University, Korea

## I. 서 론

ICT 헬싱플랫폼은 만성질환 예방을 목적으로 하며 개인의 생체신호 및 생활습관 등의 정보에 기반을 둔 질환 조기 경보를 목표로 한다<sup>[1, 2, 3]</sup>. ICT 헬싱플랫폼 상에서의 데이터 증계는 2-Step운용을 전제로 한다. Step1은 병원과 같은 데이터 공급자가 사용자 민감 정보를 제외한 개인의 건강정보나 라이프로그를 사용자의 모바일 플랫폼을 통해 사용 동의를 받은 상태에서 개인 저장소(클라우드 스토리지)<sup>[4, 5]</sup>에 저장하는 것을 의미한다<sup>[3, 6]</sup>. Step 2는 사용자가 접근을 허용한 서비스 공급자가 모바일 플랫폼을 통해 개인 저장소로부터 건강데이터에 접근하는 것을 의미한다<sup>[7]</sup>. 선행 연구<sup>[8]</sup>인 개인화된 건강 데이터 접근을 위한 2-Step 개방형 시스템은 헬싱플랫폼과 개인건강데이터 저장소간의 증계를 설계하였으며 2-Step 데이터 처리과정을 실시간으로 전송(모니터링)하기 위한 대량 커넥션 기반의 pub/sub 서비스를 고려하였다. TOS 초기 설계에서 각 커넥션은 메시지를 deflate 알고리즘으로 인코딩하기 위해 커넥션의 유휴 여부 및 메시지의 종류에 상관없이 동일한 버퍼를 할당한다. 그러나 헬싱플랫폼과 I/O바운드를 수행하는 TOS에서 네트워크 대역폭은 여타 하드웨어 장비에 비교해 손쉽게 확장하기 힘든 자원이므로, TOS의 대역폭사용 효율을 높이기 위해서는 인코딩 효율을 높이기 위한 버퍼 재할당이 선행되어야 한다. 이를 위해 각 커넥션 별로 수행한 메시지 전송 이력을 큐잉하여 특징점을 추출 및 군집하고, 해당 군집특성에 맞는 버퍼 자원 할당을 지속적으로 수행할 필요가 있다. 본 논문에서 제안하는 개인화된 건강 데이터의 대량 처리 모니터링을 위한 동적 버퍼 할당 설계는 TOS의 인코딩 버퍼 자원을 네트워크 커넥션에 효율적으로 배분하기 위해, 군집 및 특징 연산을 위한 연산 자원과 네트워크 대역폭 간의 trade-off를 수행하여 자원을 재배치함으로써 시스템의 단위 시간당 실시간 데이터 처리 모니터링 가능한 커넥션의 수를 높여 효율적인 네트워크 대역폭 사용을 초기 목표로 한다. 최종 목표는 TOS의 가용성을 높이는 것이다.

## II. 관련 연구

### 1. Vert.x

Vert.x<sup>[9]</sup>는 서버 네트워크 환경을 구축하고 운영하는

서버 프레임워크로써 이벤트 기반 프로그래밍 모델을 제공하며 비동기 형태의 API를 제공한다. Vert.x의 기본 실행 단위는 Vertical이다. Event-Loop 스레드에 의해 하나의 vert.x 인스턴스에서 동시에 여러개의 Vertical이 실행될 수 있다. 하나의 호스트 또는 다른 호스트에서 vert.x 인스턴스가 실행될 수 있으며, Vertical 이나 Module 간에는 Event Bus를 통해 통신이 가능하다.

### 2. tf-idf(Term Frequency-Inverse Document Frequency)

tf-idf은 텍스트 마이닝에서 중요한 단어를 추출할 때 쓰이는 대표적인 방법이며 한 텍스트 내 특정 단어의 빈도를 그 단어의 전체 텍스트 출현 빈도로 나누어 준다<sup>[10, 11]</sup>. 각각 단어에 대하여 tf-idf 값은 식(1)을 통해 계산한다.

$$(tf-idf)_{w,d} = f_{w,d} \cdot \log\left(\frac{|D|}{f_{w,D}}\right) \quad (1)$$

단, w는 특정단어, d는 특정 텍스트, D는 전체 텍스트 데이터이며,  $f_{w,d}$ 는 d에 나타나는 w의 빈도,  $f_{w,D}$ 는 w가 등장한 텍스트 수, |D|는 전체 텍스트 데이터의 수를 의미한다. 즉,  $(tf-idf)_{w,d}$ 값이 감소하고, 소수의 텍스트에 출현하면 증가한다. 특정단어의 빈도를 추출하고 적절한 단어를 선별하여 가중치를 확보하고 텍스트 분류에 적절한 단어를 선별하는 과정이 필요하다.

### 3. k-means clustering

k-means clustering 알고리즘은 가장 일반적으로 사용되는 분할 클러스터링 알고리즘이다<sup>[12]</sup>. 데이터가 갖는 분포에 따라 유사한 데이터 포인트들을 군집으로 묶어 주어진 d차원 포인트 n개를 각각 k개의 군집에 분배하고 같은 군집에 포함된 포인트는 갖은 속성을 갖게하고 서로 다른 군집에 속하는 포인트는 상이한 속성을 갖게 한다. 할당-재계산 과정에서 중심이 이동하면서 적절한 위치로 이동한다. 패턴들과 그 패턴에 속하는 클러스터의 중심과의 평균 Euclidean 거리를 최소화 한다. 클러스터의 중심은 그 클러스터에 속한 패턴의 평균 또는 중심이라 하고 식(2)와 같다.

$$\vec{v}(w) = \frac{1}{|w|} \sum_{x \in w} \vec{x} \quad (2)$$

단,  $w$ 는 클러스터에 속한 패턴집합이며,  $\vec{x}$ 는 클러스터에 속한 특정 패턴이다. 패턴은 실수 값을 갖는 벡터로 표현한다. 패턴 수가  $n$ 일 때 시간 복잡도는  $O(n)$ 이다.

### III. 메시지 모델 및 동적 버퍼 설계

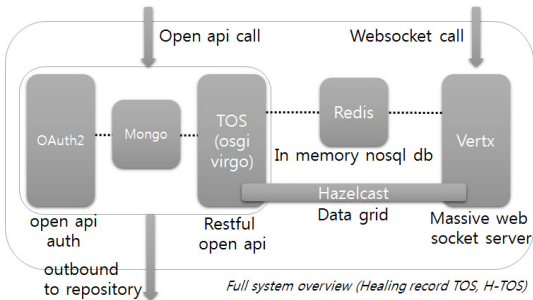


그림 1. TOS 개요 블록 다이어그램  
 Fig. 1. TOS overview block diagram

기존 연구인 Two-Step Open System(TOS)에서는, 사용자별 개인 저장소와의 중계상황을 모니터링 하기 위해 연속적인 대량 커넥션에 사용할 비동기 소켓프레임워크인 Vert.x 기술을 바탕으로 하여 개인화된 건강데이터 처리의 모니터링을 적용을 향후 연구로 명시하였다. 그림 1 은 이를 위해 본 논문에서 모니터링 메시지를 publish/subscribe 하기 위한 초기 구조이며 다음과 같은 프로세스로 진행된다. 1.subscribe를 요청(topic수신 관계)하기 위해서는, vert.x 서버에 접속해야 한다. 2.Vert.x 서버(연결서버)에 접속하기 위해서는 접속 인증에 해당하는 Token이 필요하다. 3.Token은 TOS api를 통해 발급받는다. 4.발급받은 Token으로 vertx 에 접속한다. 5.수신하고자 하는 토픽 변경시 TOS API 혹은 Vert.x서버를 통해 토픽 변경 등의 관계 명령을 전송한다.

#### 가. 개요- 대량 모니터링 메시지 모델 설계

본 논문을 통해 정의한 메시지 모델은 다음의 두 종류이며 하나는 subscribe을 관계하는 topic command 메시지와, publish에 해당하는 topic content 메시지이다. 표 1.은 topic관련한 두 메시지의 구조를 나타낸다. 그림 2.는 그

표 1. Publish/Subscribe 메시지 BNF

Table 1. Publish/Subscribe message BNF

Rule 명	메시지 BNF
Topic contents 구조 (publish)	$\langle \text{message} \rangle ::= \{ \text{"timestamp"} \langle \text{timestamp} \rangle, \text{"message"} \langle \text{messagesn} \rangle, \text{"payload"} \langle \text{payload} \rangle \}$ $\langle \text{payload} \rangle ::= \{ \text{"invokedapi"} \langle \text{invokedapi} \rangle, \text{"uid"} \langle \text{uid} \rangle, \text{"requestprovider"} \langle \text{requestprovider} \rangle, \text{"responsetime"} \langle \text{responsetime} \rangle, \text{"exception"} \langle \text{exception} \rangle, \text{"data"} \langle \text{data} \rangle \}$
Topic command 구독/해제 관계 (subscribe)	$\langle \text{operation} \rangle ::= \{ \text{"uid"} : \langle \text{uid} \rangle, \text{"op"} : \langle \text{op} \rangle \}$ $\langle \text{uid} \rangle ::= \langle \text{str} \rangle \# \text{ 제약사항은 위 표 참조}$ $\langle \text{op} \rangle ::= [ \text{"GA"} ] \# \text{ get available topic list}$ $[ \text{"GL"} ] \# \text{ get all subscribe topic list}$ $[ \text{"S"} , \langle \text{list} \rangle ] \# \text{ subscribe topic}$ $[ \text{"U"} , \langle \text{list} \rangle ] \# \text{ unsubscribe topic}$ $[ \text{"SA"} ] \# \text{ subscribe all}$ $[ \text{"CA"} ] \# \text{ clear all subscribed topic list}$ $\langle \text{list} \rangle ::= [ \langle \text{str} \rangle , \dots ]$ $\langle \text{str} \rangle ::= \# \text{ standard JSON string}$

림 1.의 기본 구조에서 표 1.의 메시지를 전송하기 위해 연결서버와 in-memory db(redis)간의 순차다이어그램을 나타내고 있다.

그림 2.에서 연결서버인 vert.x는 비동기 기반의 프로세스로 수행하기 때문에 TOS와 데이터 그리드인 Hazelcast<sup>[13]</sup>로 event를 주고받고, imdb인 redis에서 topic 리스트를 가져오거나 publish 할 topic contents를 가져오기 위해서는 (1) send job~ (2) return result 사이의 과정처럼 non-block 기반의 프로세스로 설계되었으며. 이는 연결서버 내에서 blocking-queue 등을 사용중에 thread hold 시 single thread 기반인 vert.x 플랫폼에 freezing이 발생할 수 있음을 반영한 결과이다.

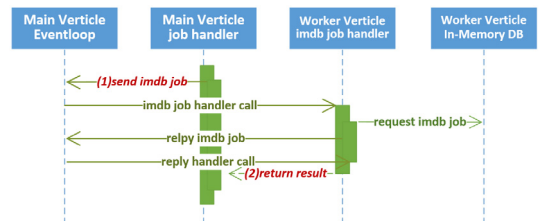


그림 2. 연결서버와 in-memory db 간의 프로세스 순차 다이어그램

Fig. 2. The Process sequence diagram between connection server(vertx) and imdb

그림 3.은 그림 2.의 프로세스를 바탕으로 표 1.의 메시지를 전송하는 300초간의 네트워크 대역폭 변화도 이다. 커넥션 수 별 tps지표는 표 3.과 같다.

표 2. connection tps (publish)  
Table 2. connection tps (publish)

connec tion	run time(s)	avg.respo nse(ms)	message count	transaction per second(tps)
6,000	140.134	0.088	1,584,865	16,963
18,000	140.962	0.078	1,812,298	14,034
36,000	139.698	0.051	2,75,8441	32,573

표 2.의 connection은 동시 접속중인 클라이언트 커넥션 수이다. run time (단위 초)는 실험 시작후 경과한 시간이다. average response은 요청 하나를 처리하는데 소요하는 응답시간이다. received message count는 해당 run Time 시점에 누적된 총 발송수이다. transaction per second는 단위 초당 요청처리 능력이다. 표 3. 및 그림 3.의 수행은 topic에 대한 인코딩을 적용하지 않았을 때의 결과이다. 그림 3.을 고려할때 대역폭 사용을 감안해 효율적으로 인코딩을 적용할 제반 방안이 필요함을 알 수 있다.

나. 커넥션 히스토리 큐의 tf-idf

커넥션의 메시지 큐에는 그림 4.와같이 관계, 유희 및 content에 해당하는 이력이 입력된다. t1 시점에서의 큐 윈도우가 사용되었다면 다음 버퍼 재할당 시점인 t2에는 윈도우가 슬라이딩하여 t2 윈도우에서의 큐 리스트를 생성한다. 이후 클러스터링을 위해, 메시지 큐는 tf-idf를 통해 벡터 공간 모델로 벡터로 변환된다. 메시지 큐에서의 이력별 가중치  $W_r$ 은 식 (3)과 같이 부여 된다.

$$W_r = TF_r \log \frac{N}{DF_r} \tag{3}$$

$TF_r$ 은 이력 빈도이며, 비교대상인 커넥션 이력 리스트의 수는  $N$  이고  $DF_r$ 는 이력별 문서 빈도(Document Frequency)이다.

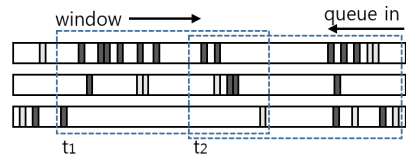


그림 4. 커넥션 메시지 큐  
Fig. 4. connection message queue

다. 클러스터링 시뮬레이션

tf-idf를 통해 벡터화된 결과 리스트는 클러스터링에 입력하여 센트로이드를 중심으로한 군집을 생성하도록 한다. 본 논문에서는 k-means 클러스터링을 군집 알고리즘으로 사용하도록 한다. 초기 군집은 세 개의 군집으로 나누도록 센트로이드를 생성하고, 세 군집이 계산의 편의상 유사한 수로 군집되는 것을 목표로 한다. k-means 적용시 각 군집은 독점군집(exclusive clustering)으로 아이템이 하나의 군집에만 속하도록 하고 선형거리(linear distance)를 사용하여 군집을 나누도록 한다. 본 논문에서 퍼지 k-means등의 여러 군집 방법 및 tf-idf 이외에 lda(Latent Dirichlet Allocation) 방법을 사용하거나 중첩 군집등, 여러 방법중 최적해를 자동 선택하는 것은 설계의 범위를 넘어가므로 해당 내용은 향후 연구의 구현 과제로 남기도록 한다. 아래의 그림은 사진에 생성한 768 건의 메시지에 대해 tf-idf 한 결과를 k-means로 간략히 시뮬레이션 한 결과이다. 그림 5.에서 군집 생성시 적절한 센트로이드에 수렴하도록 초기 조건 및 모델을 최적화 하는 것 또 향후 구현 연구의 과제이다.

라. 동적 버퍼 할당 프로세스 설계

그림 1.를 기반으로 버퍼 재할당을 고려한 블록다이어그램은 그림 6.과 같다. 그림 5.의 왼쪽 블록으로, 버퍼 재할당을 수행하는 분석서버는 기계학습용 프레임워크인 머하우트<sup>[14]</sup>을 고려하여 하둡 기반의 분산 분석처리를 수행하도록 한다. 또한 벡터연산 및 군집연산은 머하우트 라이브러리에 기반의 하위 모듈을 통해 메타성격의 중간데이터를 전처리 하도록 한다. 오른쪽 블록인 연결 서버에서

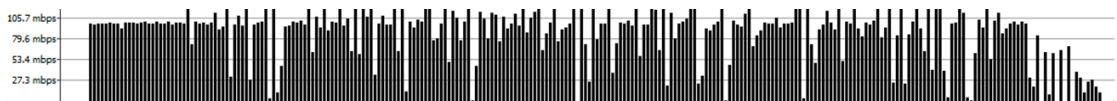


그림 3. 18,000 커넥션시의 TOS 네트워크 대역폭 그래프 (run time: 300sec). win 7 64bit, i7- 2570, 16GB Ram, 100MB local Lan JDK 1.7 -xms512m -xmx8g, vertx 2.0.2final  
Fig. 3. TOS network bandwidth graph at 18,000 connections

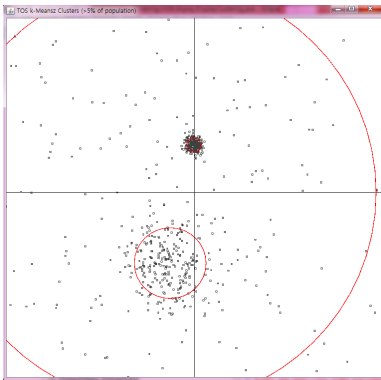


그림 5. 768 건의 메시지에 대한 클러스터링 실험 결과  
 Fig. 5. Clustering test results for the 768 cases of messages

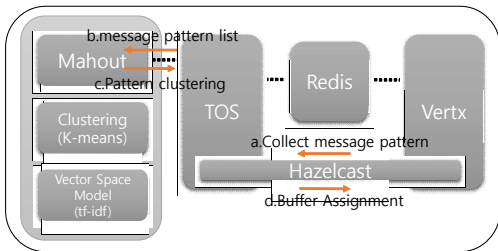


그림 6. 버퍼 할당을 수행하는 TOS 블록다이어그램  
 Fig. 6. TOS block diagram for a buffer assignment

는 커넥션당 메시지 이력을 수집하여 TOS 에 전송하고 TOS는 버퍼 재할당을 위한 분석 서버에 커넥션별 패턴 리스트를 전달하여 버퍼 재할당 작업을 요청한다. 분석된 결과는 군집으로 분류되어 군집별 버퍼를 차등 할당한다. 위 과정의 순차다이어그램은 그림 7.과 같다.

그림 7. 다이어그램에서 연결 서버에서는 수시로 현재의 대역폭 임계치 상황을 리포팅하여 TOS 버퍼 재할당이 필요할지 판단하도록 한다. TOS는 사전에 정한 버퍼 할당조건이 충족하면 연결서버에 현재까지의 각 커넥션별 메시지 유형 큐를 요청한 후, 분석서버에 수집한 커넥션별 패턴리스트를 전달하여 분석을 요청한다. 분석 서버는 그림 8에서 수행한 결과로 군집 결과를 리턴한다. TOS는 커넥션 별 군집리스트를 확인하여 센트로이드를 선택한다. TOS는 분석서버에 센트로이드를 대표값으로 한 버퍼할당 참조테이블 생성을 요청한다. 마지막으로 TOS는 각 커넥션별 버퍼 재할당을 기술하는 명령서를 작성하여 연결서버에 전송한다. 연결서버는 수신한 버퍼 재할당 명령을 통해 커넥션의 버퍼를 할당하도록 한다.

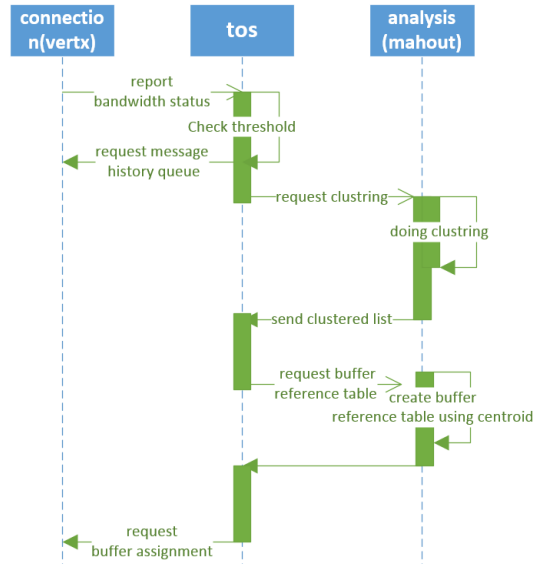


그림 7. TOS, 커넥션서버, 분석서버 간의 버퍼 할당 순차 다이어그램  
 Fig. 7. The sequence diagram between TOS, connection server and analysis server

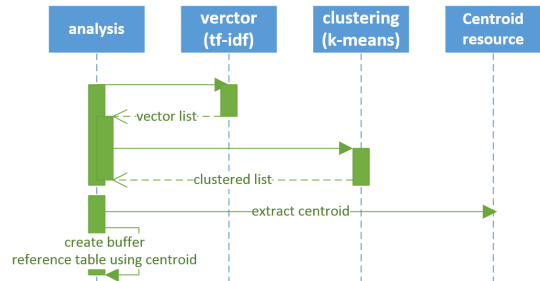


그림 8. 클러스터링을 수행하는 분석 서버의 순차 다이어그램  
 Fig. 8. Sequential diagram of the analysis server to perform clustering.

그림 8.은 TOS의 군집 생성 요청을 수행하는 분석 서버의 실행 순차도를 나타낸다. 분석 모듈은 TOS로부터 분석요청시 수신한 패턴리스트를 전처리하여 tf-idf 알고리즘에 입력한 후 초기 연산결과를 받고, 해당 결과는 벡터화 하여 k-means 군집 알고리즘에 입력한다. 만들어진 군집 리스트는 TOS에 전송된다. TOS는 수신한 군집리스트를 바탕으로 센트로이드 생성을 분석서버에 요청한다. 분석서버는 생성된 센트로이드 id 와 초기 패턴 리스트를 비교하여 커넥션별 버퍼 테이블을 생성한다. 생성된 버퍼할당 테이블은 TOS가 커넥션 별 버퍼할당 명령서 생성시 참조 활용한다.

## IV. 결론

본 논문을 통해 설계한 동적 버퍼 할당의 최종 목표는 데이터 처리 모니터링을 수행하는 단위 시간당 커넥션의 수를, 버퍼자원의 재할당을 통해 향상함으로써 효율적인 네트워크 대역폭 사용하고 결과적으로 시스템의 가용성을 높이는 것이다. 본 연구의 설계는 TOS의 인코딩 버퍼 자원을 네트워크 커넥션에 효율적으로 배분하기 위한 방안을 다음의 3 단계로 나누어 진행하였다. 1. 커넥션별 메시지를 큐잉하고 tf-idf 알고리즘으로 큐의 특징을 연산하여 백터리스트를 생성한다. 2. 생성된 백터들을 k-means 클러스터링 알고리즘을 통해 리스트간의 거리를 판별하여 군집을 생성한다. 3. 생성된 군집의 샌트로이드를 통해 자원 할당 테이블을 생성하여, 군집에 소속한 백터는 해당 테이블을 참조하여 커넥션에 자원을 재할당한다. 이를 위해 메시지 설계 및 대량 연결시의 부하 전송 실험을 통해 대역폭 효율적인 인코딩이 필요함을 나타내었다. 차후의 후속 연구 목표는 amazon ec2 상에서의 실제 성능 평가 및 tf-idf 및 lda간의 특이점 추출 차이와, k-means 및 퍼지 k-means간의 비교등을 실 구현을 통해 TOS의 tps가 향상 가능성을 확인하는 것이다.

## References

- [1] JeMin Song, MyungSic Kim, KyeongJa Jeong and MoonSun Shin, "RBAC-based health care service platform for individual recommended health information service", Journal of the Korea Academia-Industrial cooperation Society(JKAIS) Vol.15, No.3, pp.1740-1748, 2014.
- [2] Seokjin Im, Hee-Joung Hwang, "Development of Smart Health Client based on Real-Time Health Information Sharing Framework", The Journal of The Institute of Internet, Broadcasting and Communication (JIIBC) Vol.14, No.3, pp.131-137, 2014.6.
- [3] Kim, Il Kon. "Health Innovation for Aging Society." Healthcare informatics research Vol.21, No.2 pp.63-66. 2015.
- [4] Frank E, Gillett "The Personal Cloud" Forrester Research, 2009.
- [5] Dropbox Developers, <https://www.dropbox.com/developers>
- [6] International Organization for Standardization. Health informatics - Person-owned document repository for PHR applications and health information exchange. Geneva: International Organization for Standardization; 2015. (ISO/AWI TR 20055).
- [7] Roy Fielding, "Architectural Styles and the Design of Network-based Software Architectures", Dissertation of Doctor of Philosophy in Information and Computer Science, University of California, IRVINE, 2000.
- [8] YoungJun Jeon, HeeJoung Hwang "Design of Two-Step Open System for Personalized Health Data Access", The Journal of The Institute of Internet, Broadcasting and Communication (JIIBC) Vol.15, No.4, pp.177-183, 2015.8.
- [9] Vertx framework <http://vertx.io/>
- [10] Rajaraman, Anand, and Jeffrey D. Ullman. "Mining of massive datasets", Vol. 77. Cambridge: Cambridge University Press, 2012.
- [11] Luhn, Hans Peter. "A statistical approach to mechanized encoding and searching of literary information", IBM Journal of research and development Vol.1. No.4, pp.309-317, 1957.
- [12] MacQueen, James. "Some methods for classification and analysis of multivariate observations", Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. Vol.1. No.14. 1967.
- [13] Hazelcast: In-Memory Data Grid <https://hazelcast.com/>
- [14] Mahout framework <http://mahout.apache.org/>

## 저자 소개

### 전 영 준(정회원)



- 2003년 2월 : 인천대학교 컴퓨터공학과(공학사)
- 2005년 8월 : 인천대학교 컴퓨터공학과(공학석사)
- 2010년 2월 : 인천대학교 컴퓨터공학과(공학박사)
- 2013년 3월 ~ 현재 : IDLE CO.,LTD.

<주관심분야 : Software Engineering, u-Health, Big Data, Medical Informatics, OSGi>

### 황 희 정(정회원)



- 2000년 9월 : 인하대학교 컴퓨터공학과(공학석사)
- 2008년 2월 : 인천대학교 컴퓨터공학과(공학박사)
- 2000년 10월 ~ 현재 : 가천대학교 IT 대학 컴퓨터공학과

<주관심분야 : Software Engineering, u-Health, Big Data, Medical Informatics, Ubiquitous Computing.>

※ 이 논문은 2015년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임  
[B0101-15-247 개인 건강정보 기반 개방형 ICT 힐링 플랫폼 기술 개발]