

모델 기반 EPCglobal 네트워크의 성능 분석

강용신 · 손경원 · 이용한[†] · 이종태

동국대학교 서울캠퍼스 산업시스템공학과

A Model-based Performance Study of the EPCglobal Network

Yongshin Kang · Kyungwon Son · Yong-Han Lee · Jongtae Rhee

Department of Industrial and Systems Engineering, Dongguk University-Seoul, Seoul 100-715, Korea

The EPCglobal Network is a computer network used to share product data among trading partners. It provides the supply chain with improved visibility and traceability by using Electronic Product Code (EPC), which is stored on an RFID tag. Although this network model is widely accepted as a global standard and the growth of EPCglobal-subscriber base is considerable, the EPC technology adoption process is still in its infancy. This is because some of the critical issues on this model still remain to be verified such as scalability, data management, security, privacy and the economic value of data sharing. In this paper, we focus on scalability issue among the challenges to overcome and we regard performance of the EPCglobal Network only as a track and trace query-processing cost in the network. We developed performance models consisting of three elements of the EPCglobal Network : Discovery Services (DS), EPC Information Services (EPCIS), Object Naming Services (ONS). Then we abstracted out the track and trace query execution model to evaluate performance of the overall EPCglobal Network. Finally using the proposed models, we carried out simulation analysis based on an RFID-based inbound logistics process of automobile parts. This work is an important step towards the EPC technology diffusion and provides guidelines for businesses looking to buy or build the EPCglobal Network-based systems.

Key words: EPCglobal Network, Performance, RFID, Supply Chain

1. 서론

RFID 기술은 무선 주파수를 이용하여 비접촉으로 사물을 인식하는 기술로서, 사물의 위치 파악 및 경로 추적을 통해 기업에게 실시간으로 제품의 상황 정보를 전달할 수 있는 기술이다. 현재 RFID 기술은 산업 전반에 걸쳐 도입 및 적용이 진행되고 있으며 특히 공급망 관리(SCM) 분야에 도입될 경우 수익성과 경쟁력 확보에 크게 기여할 것으로 기대되고 있다. RFID 기술이 공급망 전반에 도입될 경우 개체 정보의 품질이 높아지고 개체 정보의 단절이 줄어들면서 정보 동기화(synchronization)가

향상되고 이로 인해 가시성(visibility)과 추적성(traceability) 제고에 기여하게 된다(Ahn *et al.*, 2007). 이로 인해 채찍 효과(bullwhip effect)가 근본적으로 개선되어 불필요한 주문 및 생산이 감소하고, JIT(Just-In-Time), VMI(Vendor Managed Inventory), CPFR(Collaborative Planning, Forecasting and Replenishment) 등과 연계될 경우 유통 프로세스 최적화에 기여할 수 있을 것이다. 또한, 고객 요구사항 분석이 신속하고 정확하게 이루어질 수 있어 상품화 과정(merchandising cycle)이 줄어들며 고객이 요구하는 높은 품질의 상품을 제공할 수 있을 것으로 기대되고 있다(Joshi, 2000).

본 연구는 지식경제부 전략기술개발사업(10031817)과 농림수산식품부 농업연구센터사업의 지원을 받아 수행되었습니다.

[†] 연락처자 : 이용한 교수, 서울시 중구 필동 3가 26번지 동국대학교 산업시스템공학과,

Fax : 02-2269-2212, Tel : 02-2260-3887, E-mail : yonghan@dgu.edu

투고일(2010년 07년 20일), 심사일(1차 : 2010년 10월 12일), 게재확정일(2011년 02월 01일)

현재 RFID 시스템의 구성요소들을 유기적으로 연계시키기 위한 하드웨어, 소프트웨어, 데이터 포맷, 데이터 콘텐츠, 코드 체계, 통신 프로토콜 등의 표준화가 진행되고 있으며, 표준화 기구 가운데 유통·물류 분야에 특화된 EPCglobal의 표준들이 업계에서 사실(de-facto) 표준으로 받아들여지고 있다. 향후 RFID 관련 데이터는, EPCglobal 아키텍처 프레임워크의 핵심적인 데이터 정보 교환 서비스인 EPCIS(EPC Information Services), ONS(Object Naming Services), DS(Discovery Services)에서 정의하는 데이터 형태로 저장될 가능성이 크다.

하지만 EPC 기술의 산업 적용 과정은 과거 바코드 기술에 비해 아직 초기 단계에 머무르고 있는데 Thiesse *et al.*(2009)는 EPC 기술의 확산을 위해 학계와 산업계에서 극복해야 할 도전 과제로 1) 데이터 교환 인프라스트럭처의 확장성(scalability) 검증 2) 표준의 복잡성·중복성 3) 마스터데이터 관리 4) 데이터 공유의 경제적 가치 평가 5) 보안 6) 사생활 보호 문제 등을 언급하였다. 본 연구에서는 첫 번째 도전과제인 ‘데이터 교환 인프라스트럭처의 확장성 검증’을 위해 EPCglobal 네트워크의 단품 단위 이력추적 성능을 시뮬레이션 분석을 통해 평가하였다. EPC global 네트워크에 저장되는 RFID 이벤트 정보는 공급망의 물류 프로세스에 따라 EPC 이벤트 타입의 종류와 저장량이 결정되고 이러한 물류 프로세스는 기업 및 공급망에 따라 다양한 형태로 구성될 수 있다. 이와 같이 물류 프로세스가 실행될 때 나타나는 정보시스템의 성능 특성들은 수리모형으로 쉽게 예측을 할 수 없는 문제이기 때문에 시뮬레이션을 통한 분석 기법이 요구된다.

본 논문의 구성은 다음과 같다. 먼저 제 2장에서는 EPCglobal 네트워크의 개념 설명과 이력 추적 방법 그리고 웹 서비스 성능 평가 관련 선행 연구를 정리한다. 제 3장에서는 EPCglobal 네트워크의 단품단위 추적 성능 모형을 제안하고 전형적인 공급망과 상용 서버에서의 벤치마크 테스트를 통해 성능 계수를 도출한다. 제 4절에서는 제안된 성능 모형을 실제 RFID 기반 자동차 부품 조달 물류에 적용하여 그 결과를 분석하고 토의 한다. 마지막으로 제 5절에서 향후 연구방향을 포함한 결론으로 마무리한다.

2. 배경지식 및 관련 연구

2.1 EPCglobal 네트워크

EPCglobal 네트워크는 인터넷 상의 EPCglobal 아키텍처 프레임워크를 따르는 모든 서비스 제공자(provider)와 프레임워크에 명시된 서비스를 사용하는 가입자(subscriber)들의 네트워크이다. 즉 RFID 기술을 이용하여 객체를 자동으로 인식하고 인식된 객체 정보를 인터넷을 통해 공유함으로써 상품정보, 이력정보, 이동현황 등을 조회할 수 있는 네트워크 시스템이다. <Figure 1>은 EPCglobal의 네트워크 아키텍처를 나타내고 있으며 본 절에서는 주요 컴포넌트만 설명한다(EPCglobal Inc., 2005).

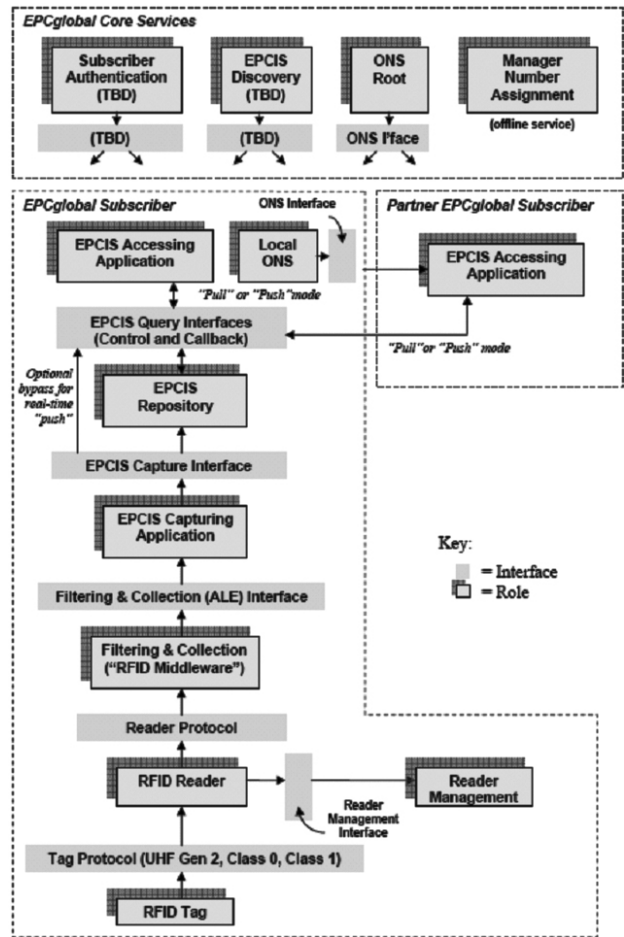


Figure 1. EPCglobal 아키텍처 프레임워크

2.1.1 EPC(Electronic Product Code)

물리적 또는 가상적으로 존재하는 물품 또는 서비스에 고유한 일련번호를 부여하여 식별을 가능하게 해주는 코드이며, 다양한 산업을 지원하기 위해 설계된 메타 코딩 체계로서 바코드와 같은 기존 체계와의 호환성을 보장한다.

2.1.2 태그(Tag)와 리더(Reader)

태그는 비 접촉식 RFID 태그를 말하며, EPC를 저장할 수 있는 메모리를 포함하고 있는 장비이다. 리더는 소프트웨어 애플리케이션이 RFID 태그로부터 데이터를 읽거나 쓸 수 있게 해주는 장비로서, 태그로부터 EPC를 판독하고 애플리케이션에 판독한 EPC를 보고하기 위한 장비이다.

2.1.3 미들웨어(Middleware)

미들웨어는 RFID 태그를 인식한 리더로부터 전달 받은 이벤트 데이터를 필터링하고 그룹핑 하여 애플리케이션으로 전달하는 역할을 한다.

2.1.4 EPCIS(EPC Information Service)

EPCIS는 EPCglobal 네트워크에서 게이트웨이 역할을 담당하

는 구성 요소로서, ALE(Application Level Events) 표준 인터페이스를 가지는 미들웨어로부터 태그 이벤트 정보를 받아 이를 이용해 객체의 상태 및 추적 정보를 생성하여 로컬 저장소에 저장하고, 이러한 정보 관리를 통해 주어진 EPC에 대한 정보 취합의 허브 역할을 담당한다. 또한, EPCIS는 비즈니스 파트너 간에 객체에 대한 가시성과 추적성을 제공하기 위해 객체 정보를 공유하는 역할을 한다. 이러한 EPCIS는 개별 사업장에 개별적으로 설치되며 분산 시스템으로 설계되었다. 즉, 각 사업장에서 일어나는 EPC에 대한 이벤트 데이터를 책임 관리하는 형태로 제안되었다(EPC global Inc. 2005; Ahn *et al.*, 2007).

2.1.5 ONS(Object Naming Service)

ONS는 글로벌 검색 서비스를 제공한다. 이는 EPC와 관련된 하나 또는 그 이상의 URI를 제공하고 제공된 URI를 통해 부가적인 정보를 얻을 수 있도록 유도한다. 제공된 URI는 특정 EPC의 부가정보(상품명, 제조일)를 얻을 수 있는 정보시스템(EPCIS, DS 등)을 나타낸다. ONS 네트워크 구조는 최상단에 전체 ONS를 관장하는 최상위 Root ONS가 존재하게 되고, 최상위 Root ONS 하위에 EPC 종류(SGTIN, SSCC 등)별 ONS들이 존재하게 된다. 그리고 EPC별 Root ONS 하위에 개별 EPC 체계에 따른 EPCglobal 네트워크 구성원들의 Local ONS가 존재하게 된다(EPC global Inc. 2005; Ahn *et al.*, 2007).

2.1.6 DS(Discovery Service)

DS는 EPC 추적을 위한 정보를 제공하는 서비스이다. DS는 현재 개념 단계에서 논의되고 있는 상태(standards in development)의 구성 요소로서 EPCglobal 네트워크에서 객체의 추적을 위한

정보를 제공하는 서비스이다. 현재 DS의 데이터 타입, 데이터 유지보수 방법, 쿼리 메소드, 보안 및 운영 문제에 대하여 EPCglobal의 액션 그룹인 Data Discovery JRG(Joint Requirements Group)에서 논의 되고 있다(Metro, 2008).

2.2 EPCglobal Track & Trace 프레임워크

EPCglobal의 액션 그룹인 HLS(Healthcare & Life Science)에서는 <Figure 2>와 같이 EPCglobal 네트워크에서 상품 이력 추적을 위한 track & trace 프레임워크를 제안하였다.

이 프레임워크에 따르면 각 사업장의 상품이 입하 및 출하될 때마다 해당 이벤트 정보와 EPCIS 주소를 중앙 저장소(Core Registry Service), 즉 DS에 등록한다. 상품 추적에 관심이 있는 클라이언트는 우선 DS로부터 사업장 단위의 이력 정보, 즉 EPCIS 리스트를 제공 받고 각 EPCIS에 상품 이력 정보를 질의함으로써 상품의 전체이력을 조회 할 수 있다. 이 프레임워크를 기반으로 학계와 산업에서 효율적인 추적 시스템 구축을 위한 연구가 진행되고 있다. Argawal *et al.*(2006)은 상품 추적을 위한 데이터 모델과 추적 쿼리를 제안하였고 쿼리 처리 실행시간 관점에서 제안된 모델을 평가하였다. Cantero *et al.*(2008)은 BRIDGE 프로젝트를 통해 DS 개발을 위한 내부 아키텍처와 외부 인터페이스를 제안하고 DS 기반 추적 애플리케이션을 제안하였다.

2.3 웹 서비스 성능 평가에 관한 연구

EPCglobal 아키텍처 프레임워크는 Service Oriented Architecture (SOA)를 지향하며 각 주요 서비스들은 웹 서비스 형태로 개발

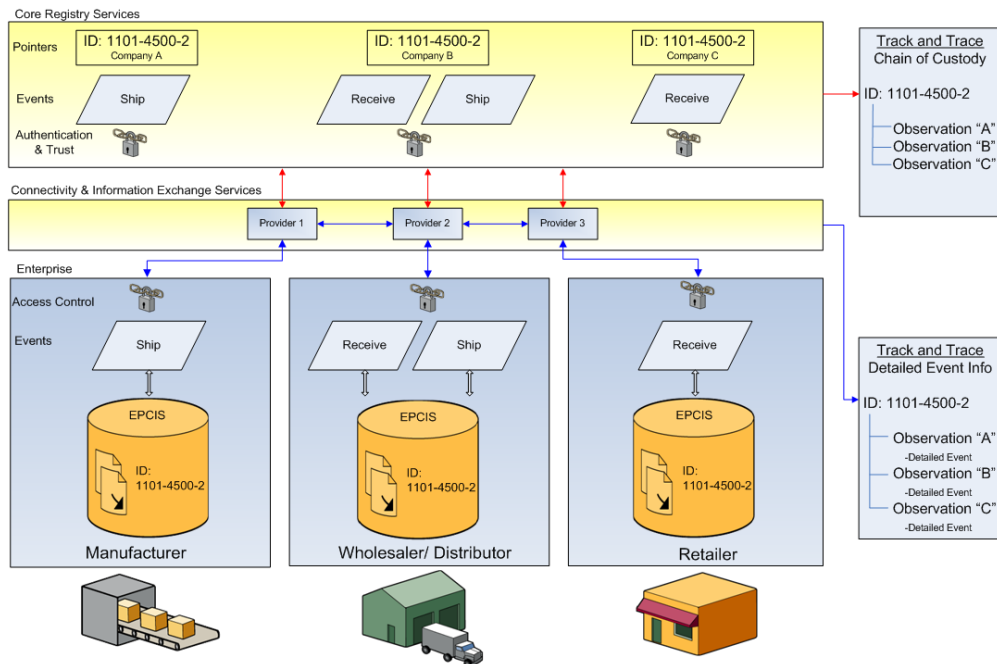


Figure 2. Track and Trace 프레임워크(Verisign and IBM, 2007)

되도록 표준에 명시되어 있으며(EPCglobal Inc., 2005), EPCglobal 네트워크의 성능평가에 대한 기존 연구는 찾아보기 힘들다.

웹 서비스의 성능(performance)은 웹 서비스 사용자가 요청한 서비스 요청에 대해서 웹 서비스 제공자가 응답을 하는 속도, 즉 얼마나 빠르게 응답하는가에 대한 속성이다. 이러한 성능 속성은 속도라는 개념으로 표현될 수 있으며, 응답 시간(response time; latency), 최대 처리량(maximum throughput)이라는 세부 품질로 정의될 수 있다. 그 외에 트랜잭션 처리 시간(transaction time), 실행 시간(execution time)이라는 지표도 활용될 수 있으나, 응답 시간과 최대 처리량을 통해 추정할 수 있는 지표로 볼 수 있다. 응답 시간이 짧을수록, 최대 처리량이 높을수록 우수한 성능을 가진 웹 서비스라고 볼 수 있다. 이 외에도 웹 서비스의 품질을 나타내는 지표로서 안정성(stability)은 제공되는 웹 서비스가 어느 정도 안정하게 서비스를 제공하는지에 대한 품질이다. 즉, 처리량의 증가, 폭주, 시스템의 고장, 자연 재해 그리고 사람에 의한 고의적 공격에도 지속적이고 일관성 있으며 원래의 상태로 회복 가능한 서비스를 제공할 수 있는 능력에 대한 품질이다. 이는 이용 가능성(availability), 신뢰성(reliability), 접근성(accessibility)이라는 지표를 활용하여 측정한다(TTA, 2006).

웹 서비스 성능 평가와 관련된 연구들은 대부분 성능에 미치는 요소를 수리적 모형으로 정의하고 사례에 적용한다. Chen *et al.*(2006) 그리고 Xiong(2006)은 웹 서비스 성능에 영향을 미칠 수 있는 요인을 매개 변수와 독립 변수로 정의하고 성능 지표인 응답 시간을 종속 변수로 정의하는 웹 서비스 성능 수식을 제안하였다.

3. EPCglobal 네트워크의 성능모형

본 장에서는 EPCglobal 네트워크의 성능을 Argawal *et al.*(2006)이 제시한 페디그리 쿼리(pedigree query) 실행에 따른 응답 시간 즉, 페디그리 쿼리 처리 시간 관점에서 성능 모형을 개발하고 상용 서버에서의 벤치마크 테스트를 수행하여 성능 모형 계수를 결정 하고자 한다. 페디그리 쿼리는 객체의 완전한 이력을 재구성하는데 사용되는 쿼리를 의미한다.

3.1 페디그리 쿼리 처리 시간

EPCglobal 네트워크 기반 공급망에서의 페디그리 쿼리는 SOA에서 클라이언트가 다수의 서비스를 조합하여 최종 목표를 달성하는 방법과 유사하다. <Figure 3>과 같이 클라이언트는 먼저 ONS에 추적하고자 하는 EPC와 관련된 DS 주소를 요청한다. 이후 클라이언트는 DS에 EPC와 관련된 이력 정보를 요청하여 EPCIS 단위, 즉 사업장 단위 추적을 완료한다. 거점 내부 이력을 추적하기 위하여 클라이언트는 DS로부터 응답받은 레코드에 포함된 EPCIS에 각각 접근하여 내부 이력 추적을 완료하여 상품의 공급망 전체 이력을 완성한다.

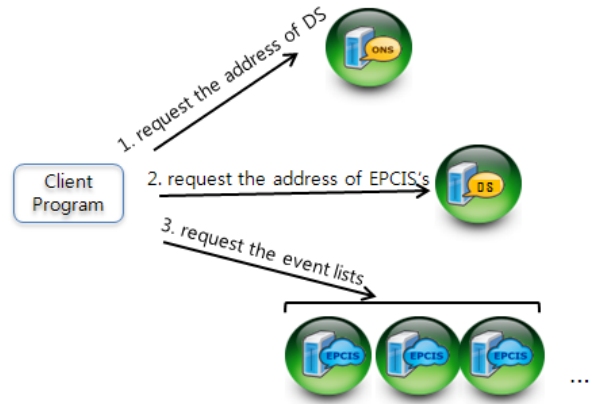


Figure 3. 페디그리 쿼리 실행 절차

따라서 총 페디그리 쿼리 처리 비용은 다음과 같이 추적하고자 하는 EPC와 관련된 ONS, DS, EPCIS에 대한 각각의 쿼리 응답 시간을 합산 한 것으로 표현될 수 있다. 본 연구에서는 쿼리 결과에 대한 정렬 또는 결함을 위한 추가 처리 시간은 고려하지 않는다.

$$C_{Pedigree} = C_{ONS} + C_{DS} + \max_i(C_{EPCIS_i})$$

C_{ONS} , C_{DS} , C_{EPCIS} 는 EPC와 관련된 각 서비스들의 쿼리 응답 시간이다. 특별히 EPCIS의 경우 Kang(2009)에 따르면 실행 가능한 모든 물류 프로세스가 고려된 추적 알고리즘을 사용하면 클라이언트가 모든 EPCIS에 동시 쿼리를 수행할 수 있다. 결과적으로 다수의 EPCIS로부터의 쿼리 응답 시간 중 가장 긴 응답 시간이 전체 EPCIS들의 쿼리 응답 시간을 결정하기 때문에 수식에 $\max()$ 함수가 사용되었다.

(1) C_{ONS} , C_{DS}

ONS 계층 구조에서의 쿼리 처리 절차는 <Figure 4>와 같이 네 단계의 조회(look-up) 작업을 수행함으로써 EPC와 관련된 서버, 즉 DS의 주소를 획득 할 수 있다.

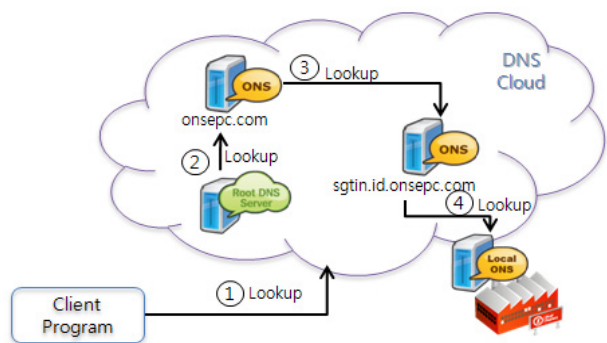


Figure 4. 4 단계 ONS 조회 절차(Ahn *et al.*, 2007)

따라서 ONS 질의 처리 비용(C_{ONS})은 다음과 같이 네 가지 ONS 서버들의 응답 시간(R_{ONS})의 합으로 정의 될 수 있다.

$$C_{ONS} = \sum_{j=1}^4 R_{ONS_j}$$

- j : 각 ONS의 인덱스($j = 1, 2, 3, 4$)
- R_{ONS_j} : ONS 계층구조 중 j 번째 ONS에서의 EPC와 관련된 레코드 응답 시간

DS의 경우 EPCglobal 비표준 상태이며 하나의 EPC와 관련된 DS가 다수 존재할 가능성이 존재하다. 따라서 DS 질의 처리 비용을 다음과 같이 정의 하였다.

$$C_{DS} = \sum_{j=1}^n R_{DS_j}$$

- j : 각 DS의 인덱스($j = 1, 2, \dots, n$)
- n : EPC와 관련된 DS의 전체 수
- R_{DS_j} : j 번째 DS에서의 EPC와 관련된 레코드 응답 시간

(2) C_{EPCIS}

EPCIS의 경우 한 번의 질의로 EPC와 관련된 모든 이력 정보를 획득할 수 없다. 전형적인 물류 프로세스에서는 <Figure 5>과 같이 단품(item)이 케이스에 포장 되고 케이스가 팔레트에 포장된다.

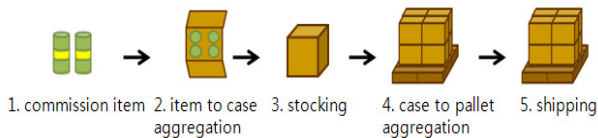


Figure 5. 실제 상황에서의 전형적 포장구조

만약 포장 후 그 구조에서의 가장 바깥쪽 태그만 RFID 리더에 의해 인식되고 EPCIS에 저장된다면 단품의 태그 ID로 EPCIS에 쿼리할 경우 포장(<Figure 5>의 2번 프로세스) 이후의 이력은 응답받지 못한다. <Figure 6>의 상단 표는 EPCIS에 저장된 각 프로세스 별 이벤트를, 저장되는 EPC 관점에서 추상적으로 표현한 것이다. 그림에서 보는 바와 같이 클라이언트는 단품 ID에 대한 EPCIS 이력 조회 후 추가적으로 단품의 부모인 케이스 그리고 케이스의 부모인 팔레트 ID로 EPCIS에 쿼리 해야 해당 EPCIS에서의 아이템 전체 이력 조회를 완성할 수 있다.

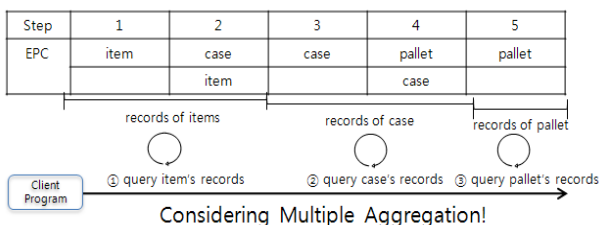


Figure 6. 페디그리 쿼리를 위한 EPCIS 조회 프로세스

이러한 다중 결합(multiple aggregation)을 고려했을 때 EPCIS 쿼리 처리 비용은 다음과 같이 정의될 수 있다.

$$C_{EPCIS} = \sum_{j=1}^n R_{EPCIS_j}$$

- j : 각 다중 결합의 인덱스($j = 1, 2, \dots, n$)
- n : 총 다중 결합 횟수
- R_{EPCIS_j} : EPCIS로부터 다중 결합 j 수준에서의 EPC 레코드 응답 시간

3.2 벤치마크 테스트

제안된 성능 모델을 완성하기 위해서는 R_{ONS} , R_{DS} , R_{EPCIS} 가 결정되어야 한다. 본 연구에서는 두 가지 상용 서버(low-end 서버와 high-end 서버)에서의 EPCIS 및 DS의 응답 속도를 측정 하였다. ONS의 경우 아직 Root ONS가 서비스화 되지 않아 테스트를 생략하였다.

3.2.1 테스트 시나리오

각 서비스에 저장되는 이벤트 데이터는 EPCglobal 액션 그룹인 HLS에서 제시한 의약품 물류 시나리오(VeriSign and IBM, 2007)에 따라 물류 프로세스가 수행될 때마다 발생하는 이벤트를 EPCIS와 DS에 저장하였다. <Figure 7>은 전체 HLS 의약품 물류 프로세스와 'Manufacturer Plant'의 내부 물류 프로세스의 구성을 보여준다.

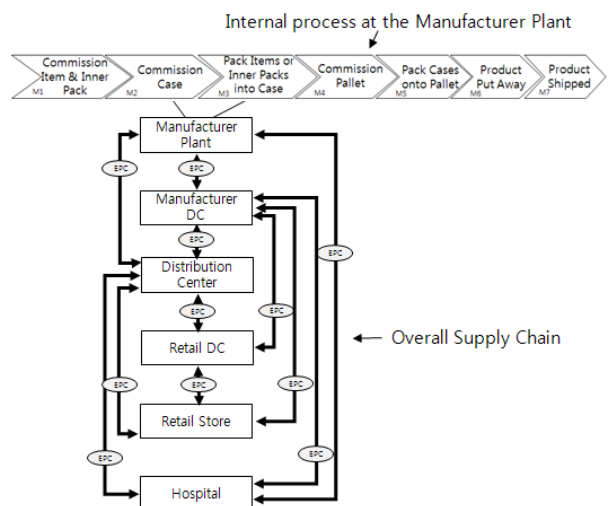


Figure 7. HLS 의약품 물류 프로세스

3.2.2 테스트 환경 및 테스트 솔루션

벤치마크 테스트를 위해 low-end, high-end 급 서버에서 각 서비스들을 운영하여 응답 속도를 비교하였고 클라이언트 프로그램은 노트북에서 구동하였다. EPCIS 솔루션은 EPCglobal에서 인증 받은 제품을 사용하였고 DS 솔루션은 VeriSign(2004)에

서 제시한 DS 구현 가이드를 준용한 제품을 사용하였다. 각 서비스가 설치된 서버는 클라이언트의 외부 네트워크에 존재하도록 설치하였고 유선 연결 환경에서 테스트하였다. <Table 1>은 테스트 환경을 요약한다.

Table 1. 테스트 환경

항목	서버		클라이언트
	low-end	high-end	
CPU	Intel Xeon 3G	Intel Xeon Quad Core 2.66GHz	Intel Core 2 Duo 2.66GHz
메모리	2G	12G	2G
OS	Windows Server 2008	Windows Server 2008	Windows XP
웹 서버	Apache Tomcat 5	Apache Tomcat 5	-
DB	Oracle 10g	Oracle 10g	-
역할	서비스 운영	서비스 운영	캡처 및 쿼리 메시지 요청

3.2.3 테스트 수행

벤치마크 테스트의 목적은 1) 초 당 데이터 캡처(capture) 및 쿼리 요청 메시지의 수가 쿼리 응답 시간에 미치는 영향평가와, 2) 서버의 이벤트 데이터 저장량이 고정된 상태에서 동시 쿼리 처리수가 쿼리 응답 시간에 미치는 영향 평가에 있다. 이를 위해 아래의 두 가지 테스트를 수행하였다.

- 테스트 A-캡처/쿼리 메시지 도착 빈도가 쿼리 응답시간에 미치는 영향에 대한 테스트
- 테스트 B-서버 이벤트 저장량과 쿼리의 동시 요청건수가 쿼리 응답시간에 미치는 영향에 대한 테스트

(1) 테스트 A

일종의 부하 테스트이며 <Figure 8>과 같이 초 당 데이터 캡처 및 쿼리 요청 메시지의 수가 쿼리 응답 시간에 미치는 영향과 한계를 보고자 함이다. 서로 다른 두 종류의 메시지(캡처 및 쿼리)의 생성(generation)을 위해서 그림에 보이는 바와 같이 두 개의 client 프로그램을 개발하여 적용하였다.

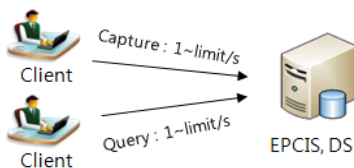


Figure 8. 테스트 A의 개념

캡처 메시지가 초당 1개씩 발생하는 상황에서 쿼리 메시지의 초당 발생 빈도를 1씩 증가시켜 가면서 쿼리 메시지의 응답 시간을 측정하였다. 응답 시간이 timeout되는 시점(30초로 설정)에 테스트 수행을 중지하고 캡처 메시지의 초당 발생 빈도

를 1 증가 시킨 후에 동일한 테스트를 반복하였으며 같은 조건에서 50번 반복 수행하여 그 평균값을 기록하였다. 테스트 결과를 요약한 <Table 2>을 살펴보면 low-end 서버에서 EPCIS를 운영한 경우 캡처 메시지가 초당 1개씩 요청될 때 초당 22개의 쿼리 메시지 요청을 처리할 수 있었으며 쿼리 메시지가 초당 1개씩 요청될 때 캡처 메시지가 초당 약 80개까지 처리될 수 있었다. high-end 서버의 경우 같은 조건에서 쿼리 메시지의 처리 한계가 55개, 캡처 메시지는 1,000개 이상도 처리 가능한 것으로 나타났다. 이처럼 서버의 성능이 향상 되는 것에 따라 전체적인 성능이 향상된다는 것은, 서비스의 성능이 하드웨어의 조건에 따라 결정되며 서비스가 확장성(scalability)을 가지고 있다고 할 수 있다.

Table 2. 처리 한계치 결과(단위 : 건/초)

	low-end 서버		high-end 서버	
	Capture Arrival Rate	Query Arrival Rate	Capture Arrival Rate	Query Arrival Rate
EPCIS	1	22	1	55
	82	1	> 1,000	1
DS	1	16	1	50
	33	1	> 420	1

마찬가지로 DS가 low-end 서버에서 운영되는 경우 캡처 메시지 요청이 초당 1개씩 발생할 때 쿼리 메시지는 초당 약 16개까지 처리될 수 있는 것으로 나타났다. 그리고 캡처 메시지의 경우 쿼리 메시지가 초당 1개씩 발생할 때 초당 33개 처리에서 한계에 도달하는 것으로 나타났다. high-end 서버에서 운영되는 경우 캡처 메시지가 초당 1개씩 발생할 때 초당 약 50개에서 쿼리 메시지 처리 한계에 도달하는 것으로 나타났다. 또한, 쿼리 메시지가 초당 1개씩 발생할 때 캡처 메시지는 초당 420개 이상도 처리가 가능한 것으로 나타났다.

종합적으로 캡처 메시지 처리의 경우 서비스 입장에서 메시지를 수신 후 저장소에 저장만 하는 반면 쿼리 메시지는 쿼리를 분석하여 저장소를 검색하고 결과를 응답해야 하기 때문에 처리 시간이 많이 소요된다. 또한 EPCIS가 DS보다 응답 시간에서 우위를 보이고 있는데 이는 VeriSign(2004)에서 제시한 DS 구현 가이드를 기반으로 구현된 DS 제품이 기본적으로 메시지를 수신 후 접근 권한을 확인하기 때문에 처리 시간이 EPCIS 보다 긴 것으로 볼 수 있다. 또한 이 결과를 통해 캡처 메시지 처리가 쿼리 처리 시간에 유의미한 수준의 영향을 주지 않는다는 것을 알 수 있었다. <Figure 9>는 테스트 A 결과를 캡처와 쿼리 메시지 도착률 증가에 따른 각 서비스의 허용 가능한 성능 한계를 보여주는 그래프이다. 영역의 색깔이 짙어질수록 쿼리 메시지 응답 시간(millisecond 단위)이 증가하는 것을 의미하며 검색 영역은 응답 시간이 허용 한계(즉 timeout 시간)를 넘어서게 되는 영역이다.

이 테스트는 제안된 성능모형을 구성하는 R_{DS} , R_{EPCIS} 성능식과 그 계수를 도출하기 위한 실험이며 실험의 절차는 다음과 같다.

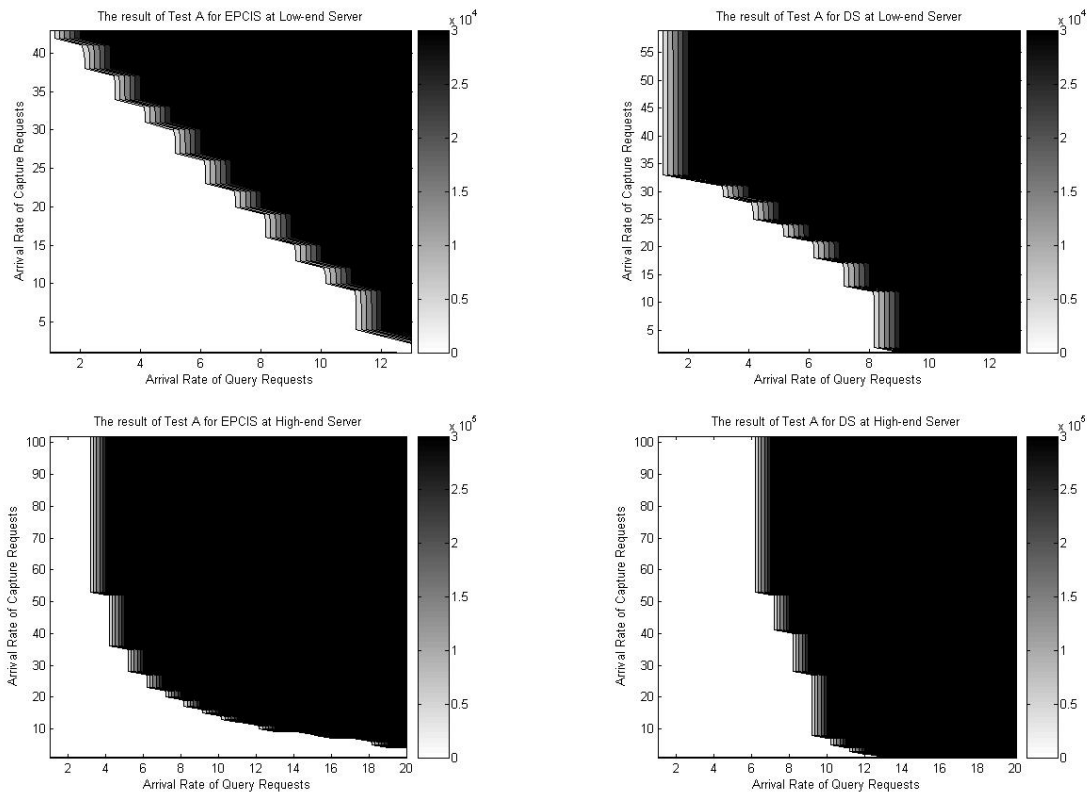


Figure 9. 테스트 A의 결과-메시지 도착 빈도에 따른 한계성능 분포

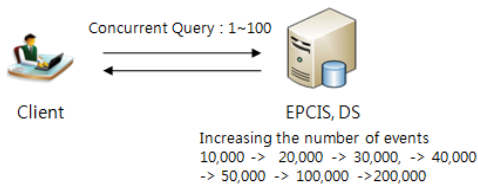


Figure 10. 응답 도착률 테스트의 개념

1. HLS 의약품 물류 프로세스 진행에 따른 이벤트를 발생시켜 각 서비스에 저장시킨다.
2. 1에서 발생한 EPC 100개를 랜덤하게 선택한다.
3. 2의 EPC를 1~100개까지 동시에 각 서비스에 질의한다.
4. 저장된 이벤트 수를 증가시켜 1번부터 반복한다.

4번은 10,000개, 20,000개, 30,000개, 40,000개, 50,000개, 100,000개, 200,000개의 이벤트 데이터를 생성시키고 각 사이클마다 50번씩 실험하여 개별 질의에 대한 평균값을 측정된 결과는 <Figure 11>과 같다.

<Figure 11>을 보면 저장된 이벤트 량이 많아질수록 점점 응답 시간이 증가하는 것을 알 수 있다. 그리고 서버에 저장된 이벤트 량이 증가할 때 동시 쿼리 처리에 따른 응답 시간에 대한 추세선 간의 간격도 어느 정도 선형적으로 증가하는 것으로 추정된다. 또한 테스트 A와 마찬가지로 같은 서버일 경우 EPCIS가 DS보다 우수한 성능을 보였다.

3.3 서비스 응답속도 계수 도출(R_{DS} , R_{EPCIS})

테스트 B의 결과에서 서비스의 동시 쿼리 처리 수와 서버의 이벤트 데이터 저장량이 쿼리 응답 시간에 선형적인 영향을 미치는 것으로 추정하였던바, 이에 대한 유의한 모형의 도출을 위해, 동시 접속자 수와 데이터 저장량을 독립 변수로, 서비스 응답 시간을 종속 변수로 설정한 다중 회귀 방정식을 다음과 같이 정의하였다.

$$Response\ Time = \beta_0 + \beta_1 X_1 + \beta_2 X_2$$

X_1 : 서비스의 쿼리 동시 처리 량

X_2 : 서버에 저장된 데이터 량

테스트 B의 결과를 토대로 다중 회귀 분석을 수행한 결과 <Table 3>과 같이 도출할 수 있었다. 계수가 모두 유의 수준에 들어왔지만 교호작용이 있으며, 도출된 계수를 검증하기 위해 실제 벤치마크 테스트 데이터와 회귀 식의 예측 값과 비교한 결과 차이가 최소 31ms에서 최대 169,381ms로 크게 나타나서 위의 식을 서비스 응답 속도에 대한 수식으로 사용할 수 없었다. 하지만 <Figure 11>을 통해 서버의 이벤트 데이터 저장량이 동시 쿼리 처리 수의 영향력(동시 쿼리 처리 수의 계수)에 영향을 미치고 있는 것을 확인하여, 우선 서버의 이벤트 데이터량을 증가 시켜가면서 동시 쿼리 처리수와 서비스 응답 시간의 관계를 단순 회귀분석을 통해 분석하고(1차 회귀분석),

동시 쿼리 처리수의 계수와 서버의 이벤트 데이터 저장량의 영향력을 단순 회귀 분석(2차 회귀분석)한 결과 <Table 4>, <Table 5>와 같은 결과를 도출할 수 있었다.

<Table 4>, <Table 5>의 상단(1차 회귀분석 결과)를 통해 각 서비스에서 동시 쿼리 처리 수의 계수 값(β_1)은 서버의 저장된 이벤트 데이터 량에 따라 증가함을 알 수 있다. 이는 서버에 저장된 이벤트 데이터 량이 일정하게 동시 쿼리 처리 수의 계수에 영향을 주고 있음을 의미하며 <Table 4>, <Table 5>의 하단(2차 회귀분석 결과)을 통해 서버에 저장된 이벤트 량의 계수(β_1)와 y 절편(β_2) 모두 신뢰수준 95%에서 유의하므로 이벤트 량과 동시 쿼리 처리 수의 계수 간에 선형 관계가 있음이 증명되었다. 따라서 두 차례의 회귀분석을 통해 도출된 서비스의 응답 시간에 대한 수식을 다음과 정의하였고 각 서비스에

대한 응답 시간에 대한 계수를 low-end, high-end 서버로 구분하여 <Table 6>과 같이 도출하였다.

$$R_{EPCIS, DS} = \beta_0 + (\beta_2 + \beta_3 X_{events}) X_{concurrent}$$

- X_{events} : 서비스의 쿼리 동시 처리 수
- $X_{concurrent}$: 서버에 저장된 데이터 량
- β_0 : 동시 쿼리 처리 수와 서비스 응답 시간의 관계 회귀식의 초기값(y 절편)
- β_2 : 서버의 이벤트 데이터 저장량과 동시 쿼리 처리 수의 계수와와의 관계 회귀식의 초기값(y 절편)
- β_3 : 서버의 이벤트 데이터 저장량이 동시 쿼리 처리 수의 계수에 대한 영향을 나타내는 계수

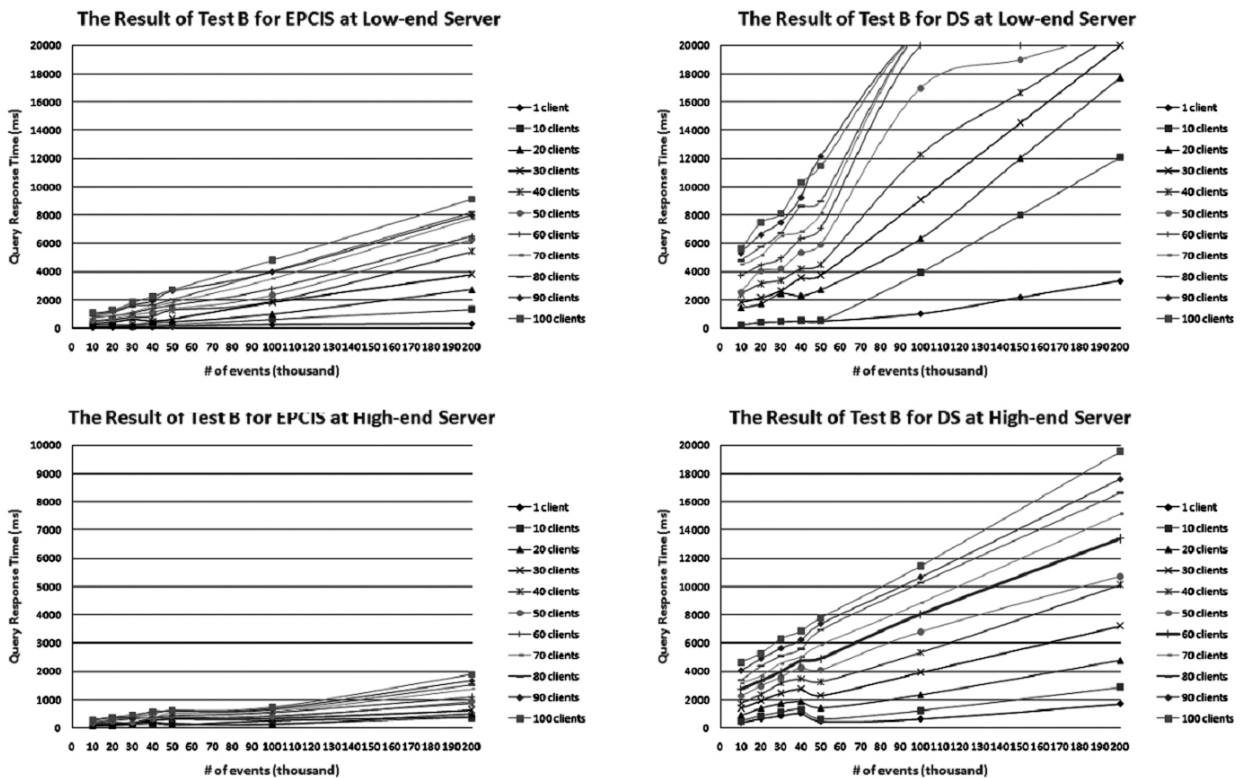


Figure 11. 테스트 B 결과-서버 이벤트 저장량이 쿼리성능에 미치는 영향

Table 3. 다중 회귀 분석 결과

		비표준화 계수			Sig.		상관계수		
		β_0	β_1	β_2			0차	편	부분
E P C I S	low-end 서버	-1453.817	31.813	0.026	.000	X_1	.487	.765	.487
						X_2	.770	.882	.770
	high-end 서버	-144.009	5.761	0.004	.000	X_1	.516	.756	.516
						X_2	.731	.853	.731
D S	low-end 서버	-54297.554	739.974	0.860	.000	X_1	.319	.463	.319
						X_2	.725	.765	.725
	high-end 서버	-2232.109	84.320	0.044	.000	X_1	.647	.867	.647
						X_2	.666	.873	.666

Table 4. EPCIS에 대한 2단계 단순 회귀 분석 결과

1차 회귀분석		$E(y X_{concurrent}) = \beta_0 + \beta_1 X_{concurrent}$						
이벤트량		10000	20000	30000	40000	50000	100000	200000
항목								
low-end 서버	y절편(β_0)	51.82643	77.54984	106.48943	130.49806	158.19816	260.49741	247.46717
	계수(β_1)	10.51793	13.06615	18.18824	21.80688	26.73953	44.17848	89.81825
	표준오차	0.20244	0.21993	0.29159	0.27741	0.34733	0.95473	1.82357
	t-통계량	51.95569	59.41030	62.37508	78.60833	76.98511	46.27314	49.25401
	p-값	3.9×10^{-73}	1.1×10^{-78}	1.1×10^{-80}	2.4×10^{-90}	1.8×10^{-89}	2.1×10^{-68}	6.0×10^{-71}
high-end 서버	y절편(β_0)	54.78216	58.44767	77.36304	127.3162	82.71077	84.42275	168.4062
	계수(β_1)	2.54329	3.24807	3.72224	4.30104	5.82205	7.08558	16.93822
	표준오차	0.06644	0.07186	0.09257	0.13783	0.11736	0.17438	0.27706
	t-통계량	38.28165	45.19872	40.21054	31.20595	49.60711	40.63250	61.13496
	p-값	9.51×10^{-61}	1.93×10^{-67}	1.024×10^{-62}	1.047×10^{-52}	3.085×10^{-71}	3.898×10^{-63}	7.475×10^{-80}
2차 회귀분석		$E(\beta_1 X_{events}) = \beta_2 + \beta_3 X_{events}$						
항목		계수(β_3)	표준오차	t-통계량	p-값			
low-end 서버	y절편(β_2)	5.23228	0.815075	6.419381	0.001362			
	계수(β_1)	0.000417	9.15×10^{-06}	45.56457	9.61×10^{-08}			
high-end 서버	y절편(β_2)	1.50819	0.50466	2.98854	0.02150			
	계수(β_1)	0.00007356	5.668×10^{-06}	12.979442	4.839×10^{-05}			

Table 5. DS에 대한 2단계 단순 회귀 분석 결과

1차 회귀분석		$E(y X_{concurrent}) = \beta_0 + \beta_1 X_{concurrent}$						
이벤트량		10000	20000	30000	40000	50000	100000	200000
항목								
low-end 서버	y절편(β_0)	182.13453	278.57548	452.37125	396.81446	427.92061	620.80196	1019.15343
	계수(β_1)	60.03162	69.88559	79.55739	98.93024	110.60989	296.75796	871.53721
	표준오차	0.73872	0.61441	1.02304	0.82528	0.84219	5.96077	34.41160
	t-통계량	81.26398	113.74489	77.76535	119.87456	131.33603	49.78515	25.32684
	p-값	9.88×10^{-92}	6.8×10^{-106}	6.91×10^{-90}	4.1×10^{-108}	5.7×10^{-112}	5.65×10^{-43}	1.58×10^{-15}
high-end 서버	y절편(β_0)	26.63772	433.02481	661.20673	823.47774	-166.88443	277.15241	1581.75917
	계수(β_1)	44.80799	48.60515	56.38799	62.16304	84.59432	122.06642	188.65900
	표준오차	0.20537	0.41099	0.63028	0.88476	0.95723	1.66342	2.36165
	t-통계량	218.18001	118.26386	89.46536	70.25992	88.37431	73.38270	79.88441
	p-값	2.0×10^{-133}	2.0×10^{-107}	9.1×10^{-96}	1.2×10^{-85}	3.0×10^{-95}	1.8×10^{-87}	5.2×10^{-91}
2차 회귀분석		$E(\beta_1 X_{events}) = \beta_2 + \beta_3 X_{events}$						
항목		계수(β_3)	표준오차	t-통계량	p-값			
low-end 서버	y절편(β_2)	44.74259	3.261203	13.71966	0.000838			
	계수(β_1)	0.001302	9.83×10^{-05}	13.24138	0.000931			
high-end 서버	y절편(β_2)	36.60931	3.54083	10.33920	0.00015			
	계수(β_1)	0.00078	0.00004	19.61600	6.4×10^{-06}			

Table 6. 최종 서비스 응답 시간 모형의 계수

		β_0	β_2	β_3
EPCIS	low-end 서버	58.180	5.232	0.000417
	high-end 서버	7.361	5.761	0.004
DS	low-end 서버	221.729	44.742	0.0013026
	high-end 서버	46.831	36.609	0.00078

위 계수를 통해 예측한 응답 시간과 실제 벤치마크 테스트를 통해 얻은 응답 시간은 최소 3ms에서 최대 341ms의 차이를 보이나, 다중회귀 분석을 통해 얻은 수식보다 우수한 예측력을 보였다.

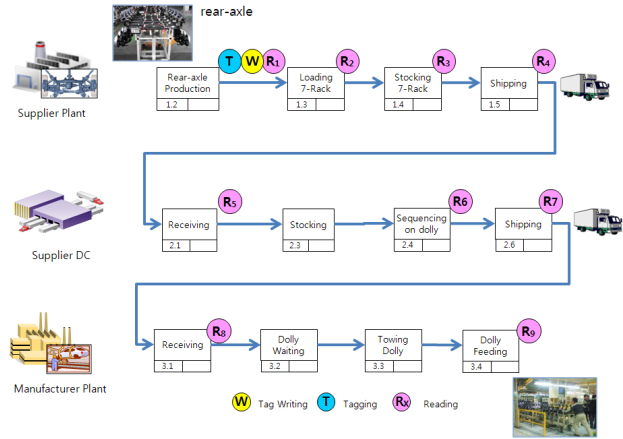


Figure 12. RFID 기반 자동차 부품 조달 물류

4. 시뮬레이션을 통한 사례 분석

본 연구에서 제안된 성능 모형은 전형적인 물류 프로세스에서의 페디그리 퀴리 절차를 기준으로 개발되었고 RFID 정보 시스템 구축에 사용될 수 있는 상용 서버 및 솔루션을 기반으로 성능 지표를 도출하였다. 본 장에서는 제안된 모형을 국내 완성차 제조업체인 G사의 실제 RFID 기반 자동차 부품 조달 물류에 적용하여, 물류가 지속됨에 따라 단품 단위의 페디그리 퀴리 성능 변화 추세와, 서버 데이터 백업이나 압축 같은 유지 보수 없이 서버가 운영될 수 있는 기간을 분석하였다. 본 절의 물류 프로세스 시뮬레이션은 범용 시뮬레이션 소프트웨어인 ARENA(Version 12.0)를 이용하여 수행하였다.

4.1 프로세스 모델 및 시뮬레이션 변수

전체 공급망은 <Figure 12>에 보이는 바와 같이 부품 생산 공장, 부품 집하장, 완성차 조립공장의 세 사업장으로 구성되어 있다. 첫 번째 사업장인 부품 생산 공장은 생산, 포장, 저장, 출고의 총 4개 RFID 인식 지점(read point)이 구성된다. 포장 단계에서는 부품 7개가 한 개의 용기('7-rack')에 적재된다. 두 번째 사업장인 부품 집하장은 입하, 저장, 서열작업, 출하로 구성되어 있으며 저장단계를 제외한 세 단계에 모두 RFID 인식이 이루어진다. 서열작업 단계에서는 총 5개의 부품이 한 개의 이동형 용기(dolly)에 적재된다. 마지막으로 완성차 제조 공장은 입하, 대기, 이동, 조립라인 투입의 네 단계 프로세스로 구성되며 입하 및 조립라인 투입 단계에서만 RFID 인식이 이루어진다.

제품의 생산률과 프로세스를 구성하는 작업 처리 시간 및 운송 시간은 현장 인터뷰와 실제 기록 데이터 분석을 통해 도출하였고 <Table 7>과 같이 정리하여 시뮬레이션 모델에 적용하였다. 본 시뮬레이션에서의 모든 EPC는 한 개의 DS와 관련되며 ONS퀴리 처리 시간은 평균 0.5초, 분산 0.2초의 정규분포를 따르는 것으로, 모든 EPCIS와 DS에는 각각 25%의 확률로 1~4명이 동시에 퀴리하고 있는 것으로 가정된 바, 이는 현업 인터뷰를 통해 도출된 현실적 모수들이다.

Table 7. 시뮬레이션 변수

	변수	값/분포
부품 생산 공장	production rate	1/min
	production time	N(50, 25) (unit:sec)
	7-rack loading time	N(10, 1) (unit:sec)
	7-rack stocking time	N(20, 2) (unit:sec)
	7-rack shipping time	N(20, 2) (unit:sec)
	full truck load	16 racks/truck
부품 집하장	receiving time	N(30, 3) (unit:sec)
	stocking time	N(10, 1) (unit:sec)
	sequencing time	N(60, 3) (unit:sec)
	shipping time	N(10, 1) (unit:sec)
	full truck load	6 dollies/truck
	transportation time	exp(2) (unit:min)
완성차 조립 공장	part feeding schedule	1440/day
	receiving time	N(10, 5)
	feeding time	N(60, 10)

4.2 시뮬레이션 결과

먼저 물류 프로세스가 진행됨에 따라 각 거점 별 EPCIS와 DS에 저장되는 이벤트 저장량의 변화를 살펴보면 <Figure 13>과 같다. 물류 프로세스가 실행됨에 따라 저장되는 이벤트의 수는 DS가 가장 많았고 그 다음으로 부품 생산 공장 EPCIS, 부품 집하장 EPCIS, 완성차 조립공장 EPCIS 순으로 나타났다. 이는 RFID 인식 지점 개수에 비례한 결과로 볼 수 있다.

본 연구의 주요 관심사인 페디그리 퀴리에 대한 응답속도의 변화는 <Figure 14>, <Figure 15>에서 보여준다. 사용자가 페디그리 퀴리를 실행한 후 응답 시간을 30초까지 기다린다고 가정했을 때 low-end 서버의 경우 별도의 백업절차 없이 약 3년까지 운영가능 한 것으로 예측되었고 high-end 서버의 경우 low-end서버의 2배에 달하는 약 6년 정도까지 운영 가능한 것으로 예측되었다. 또한 응답 시간의 변동 폭 역시 high-end 서버가 low-end 서버보다 상대적으로 낮음을 알 수 있다. 이러한 시뮬

레이션 결과는 단품 단위 이력 추적을 위해 사용되는 페디그리 쿼리의 처리 한계를 넘어서는 각각 3년, 6년 이전에 서버의 이벤트 데이터에 대한 백업이나 장비 업그레이드 등의 대비책을 강구할 수 있도록 하드웨어 용량계획(capacity planning)에 도움을 줄 수 있었다.

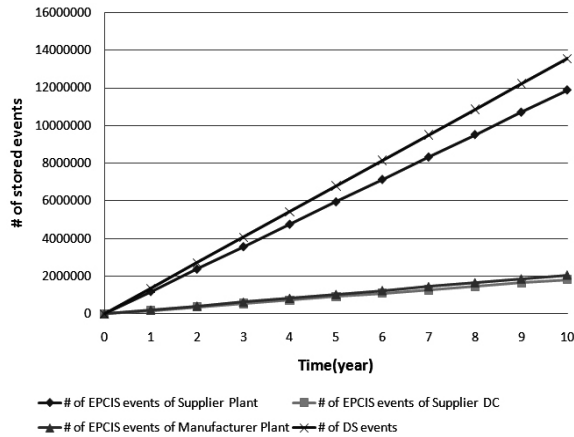


Figure 13. 서비스 별 이벤트 개수 변화

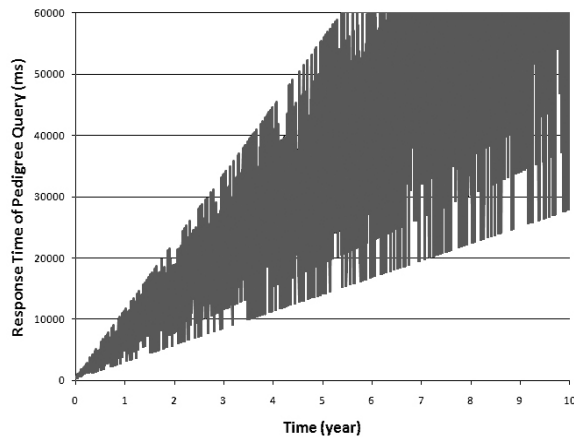


Figure 14. low-end 서버에서의 페디그리 쿼리 성능

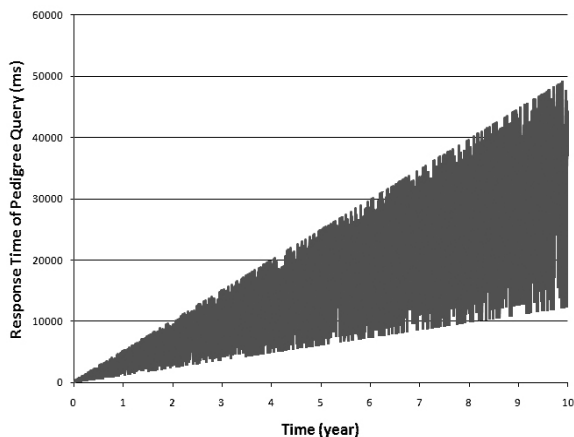


Figure 15. high-end 서버에서의 페디그리 쿼리 성능

5. 결론

본 논문에서는 EPCglobal 네트워크의 성능을 EPCglobal track and trace 프레임워크를 기반으로 단품 단위 이력 추적을 위해 사용되는 페디그리 쿼리 처리의 비용 관점에서 모형을 개발하였고 성능 계수를 전형적인 물류 프로세스와 상용 서버를 통해 도출하였다. 도출된 성능모형과 계수를 실제 RFID 기반 물류 프로세스 구축 프로젝트에 적용하여 단품 이력 추적이 안정적으로 이루어질 수 있는 한계 시점을 예측할 수 있었으며, 이는 RFID 시스템 도입 프로젝트의 하드웨어 용량계획에 활용되었다.

본 연구결과는 RFID 기반 물류 정보시스템의 사실 표준으로 인식되고 있는 EPCglobal 네트워크의 현실적인 성능모형을 제시함으로써, EPC 표준 관련 기술 및 이에 따른 RFID 기반 시스템의 확산에 기여할 것으로 기대된다. 특히 EPCglobal 네트워크 기반 시스템을 구축 또는 구매하고자 하는 기업들에게는 현실적인 가이드라인을 제공할 수 있을 것이다.

향후 연구과제로서 아직 현장에서 서비스화 되지 않은 ONS에 대한 응답 시간의 성능 모형을 도출하는 것과 RFID 이벤트 데이터 뿐 아니라 EPCglobal 네트워크에 저장 될 수 있는 센서 데이터, 위치 데이터 그리고 마스터 데이터를 고려한 다양한 쿼리 관점에서의 성능 연구가 필요할 것이다.

참고문헌

Agrawal, R., Cheung, A., Kailing, K., and Schonauer, S. (2006), Towards Traceability across Sovereign, Distributed RFID Databases, IDEAS '06. 10th International Database Engineering and Applications Symposium, 174-184.

Ahn, J.-M., Rhee, J.-T., and Oh, T.-S. (2007), EPCglobal Network 기반의 RFID 기술 및 활용, Global, Seoul, Korea.

Cantero, J. J., Guijarro, M. A., Arrebola, G., Garcia, E., Baos, J., Harrison, M., and Kelepouris, T. (2008), Traceability applications based on discovery services, ETFA 2008, IEEE International Conference on Emerging Technologies and Factory Automation, 1332-1337.

Chen, S.-P., Yan, B., Zic, J., Liu, R., and Ng, A. (2006), Evaluation and Modeling of Web services Performance, Proceedings of International Conference on Web Services, 437-444.

EPCglobal Inc. (2005), The EPCglobal Architecture Framework, Version 1.3.

EPCglobal Inc. (2007), EPC Information Service(EPCIS) Version 1.0 Specification, Ratified Standard.

EPCglobal Inc. (2005), Object Naming Service(ONS) Version 1.0 Specification, Ratified Standard.

EPCglobal Inc. (2007), Business Requirements & Process Flows, HLS Track & Trace Interest Group.

Joshi, Y. V. (2000), Information Visibility And Its Effect On Supply Chain Dynamics, Massachusetts Institute of Technology, Master's Thesis.

Kang, Y.-S., Jin, H.-J. and Lee, Y. H. (2009), A Development of Traceability Services in EPCglobal Network Environment, Proceedings of the 10th APIEMS Conference.

Metro (2008), MEAT TRACEABILITY USE CASE JRG Data Discovery Documents, EPCglobal JRG Meeting.

Song, S.-H., Shim, T.-K., and Park, J.-H. (2006), Proxy based EPC Track & Trace Service, IEEE International Conference on e-Business Engineering, 528-531.

Telecommunications Technology Association (TTA) (2006), *Web Service Quality Model*, TTA Standard, TTAS.KO-10.0204.

Thiesse, F., Floerkemeier, C., Harrison, M., Michahelles, F., and Roduner, C. (2009), Technology, Standards, and Real-World Deployments, *Internet Computing*, 13(2), 36-43.

VeriSign. (2004), VeriSign Discovery Services-Discovery Service Implementation Guide Version 1.0.

VeriSign and IBM (2007), Track And Trace Framework, EPCglobal HLS Track

and Trace Industry Action Group.

Xiong, K. (2006), Web services performance modeling and analysis, International Symposium on High Capacity Optical Networks and Enabling Technologies, 1-6.

Zhang, J.-F., Feng, P.-F., Wu, Z.-J., and Yu, D.-W. (2008), Automatic Identification-Enabled Traceability in Supply Chain Management, WiCOM '08. 4th International Conference on Wireless Communications, Networking and Mobile Computing, 1-4.



강용신

동국대학교 산업시스템공학 학사
동국대학교 산업시스템공학 석사
현재 : 동국대학교 산업시스템공학과
박사과정

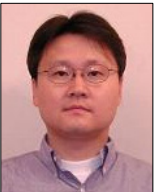
관심분야 : SCM, BPM, RFID, SOA, Web Services



손경원

동국대학교 산업시스템공학 학사
동국대학교 산업시스템공학 석사
현재 : 동국대학교 산업시스템공학과
박사과정

관심분야 : 의사결정, Risk Management, 프로젝트 관리, Healthcare Management



이용한

서울대학교 산업공학 학사
KAIST 산업공학 석사
Pennsylvania State University, I&ME 박사
대우자동차 기술연구소 선임연구원
현재 : 동국대학교 산업시스템공학과 부교수
관심분야 : RFID, Sensor Network, Multi-Agent Systems, Agent-Based Modeling and Simulation, SOA



이종태

서울대학교 산업공학 학사
KAIST 산업공학 석사
UC Berkely IE&OR 박사
현재 : 동국대학교 산업시스템공학과 교수
관심분야 : SCM, RFID 응용, 신경회로망 응용