

# SK텔레콤의 차세대 마케팅 시스템 개발사례 연구

(A Case Study on SK Telecom's Next Generation  
Marketing System Development)

이 상 구 †      장 시 영 ††      양 정 연 †††  
(Sang-goo Lee)      (Si-young Jang)      (Jung-yeon Yang)

**요 약** 무한 경쟁체제인 시장에서 변화하는 요구들을 수용하기 위한 해결책으로, SK 텔레콤은 향후 10년간 증가할 대용량의 데이터 및 트랜잭션을 처리하고, 다양한 마케팅활동 지원이 가능한 새로운 마케팅 시스템을 구축하였다. 이 시스템은 기존의 메인프레임 기반의 COIS시스템을 유닉스기반의 클라이언트-서버 구조로 변경한 것이며, 웹브라우저 기반 인터페이스를 제공한다. NGM(Next Generation Marketing) 이라고 명명된 이 프로젝트는 그 규모가 전례가 없이 매우 컸다. 그러나 관리 및 기술적 문제들로 인하여 프로젝트는 위험을 맞게 되었다. 기존의 거대 벤더들이 제공하는 소프트웨어 솔루션에 기반한 어플리케이션 프레임워크가 새로운 시스템의 다양하고 방대한 요구조건을 충분히 소화시키지 못했기 때문이다. 결국 2005년 3월 SK텔레콤은 NGM 프로젝트를 보류하였다.

2005년 5월 종합적인 기획 수정을 통해 프로젝트는 2단계에 착수되었다. 새로운 시스템의 복잡도에 대응하기 위해 더 이상 단일 솔루션을 택하지 않기로 결정하여, 새로운 시스템은 커스텀 빌트의 형태가 되었다. 본 논문에서는 성공적인 NGM 프로젝트 수행을 위해 고려된 3가지 기술적 요소 - 미들웨어 및 어플리케이션 프레임워크, 데이터베이스 아키텍처, 튜닝 및 시스템 퍼포먼스 - 에 대하여 다루고 있다. NGM 구축에 있어서 적용된 이 프로세스 및 접근방법은 텔레커뮤니케이션 사업분야에서의 가장 성공적인 실제 적용사례로 볼 수 있다. 완성된 NGM시스템은 2006년 10월 9일 성공적으로 가동되었고, "U.Key 시스템"으로 명명되었다. 이 새로운 시스템은 가까운 미래에 혁신적이고, 효과적이며 고객지향적인 어플리케이션 등을 제공할 수 있을 것으로 기대된다.

**키워드** : Next Generation System, 다온사이징, 어플리케이션 프레임워크, 데이터베이스 아키텍처, 시스템 퍼포먼스

**Abstract** In response to the changing demands of ever competitive market, SK Telecom has built a new marketing system that can support dynamic marketing campaigns and, at the same time, scale up to the large volumes of data and transactions for the next decade. The system which employs Unix-based client-server (using Web browser interfaces) architecture will replace the current mainframe-based COIS system. The project, named NGM (Next Generation Marketing), is unprecedentedly large in scale. However, both managerial and technical problems led the project into a crisis. The application framework that depended on a software solution from a major global vendor could not support the dynamic functionalities required for the new system. In March 2005, SK telecom declared the suspension of the NGM project.

The second phase of the project started in May 2005 following a comprehensive replanning. It was decided that no single existing solution could cope with the complexity of the new system and hence

† 중신회원 : 서울대학교 컴퓨터공학과 교수  
sglee@europa.snu.ac.kr

†† 정 회 원 : 성균관대학교 경영학과 교수  
syjang@skku.ac.kr

††† 학생회원 : 서울대학교 컴퓨터공학과  
jyyang@europa.snu.ac.kr

논문접수 : 2007년 9월 4일

심사완료 : 2007년 12월 7일

Copyright©2008 한국정보과학회: 개인 목적이거나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 컴퓨팅의 실제 및 레터 제14권 제2호(2008.4)

the new system would be custom-built. As such, a number of technical challenges emerged. In this paper, we report on the three key dimensions of technical challenges - middleware and application framework, database architecture and tuning, and system performance. The processes and approaches, adopted in building NGM system, may be viewed as "best practices" in the telecom industry. The completed NGM system, now called "U.key System," successfully came into operation on the ninth of October, 2006. This new infrastructure is expected to give birth to a series of innovative, fruitful, and customer-oriented applications in the near future.

**Key words** : Next Generation System, Downsizing, Application Framework, Database Architecture, System Performance

## 1. 서론

SK텔레콤은 고객지원과 요금청구 등 업무 처리를 위하여 지난 10여 년간 COIS 시스템을 활용해 왔다.<sup>1)</sup> 메인프레임 기반의 COIS 시스템은 안정성과 보안성은 뛰어나지만 유연성이나 확장성 차원에서 여러 가지 불리한 측면을 노정시키고 있었다.

이에 따라 SK텔레콤은 2002년 9월부터 4년에 걸쳐 차세대 마케팅 활동을 지원하기 위한 NGM 인프라 구축 프로젝트를 추진하여<sup>2)</sup> 2006년 10월 Unix 기반의 U.key 시스템을 성공적으로 가동하게 되었다.<sup>3)</sup>

그러나 SK텔레콤의 NGM 프로젝트의 성공적 완수에 이르기까지에는 여러 가지 기술적 난관이 있었고, 그에 따라 적절한 해결방안이 필요했다. 특히, 프로젝트 초기에 글로벌 벤더(global vendor)의 패키지 솔루션에 기반한 Turn-key 방식의 개발을 추진하였지만, 많은 문제점으로 인하여 맞춤형 시스템 개발을 위한 In-house 개발 방식으로 변경하였으며, 이에 따른 큰 변화와 난제들을 극복하여야 했다.

이 논문의 목적은 NGM 프로젝트의 성공적 완수에 이르기까지의 과정 요소와 성과를 기술적 관점에서 분석하고, 난관의 발생과 극복 등 문제해결 과정을 객관적으로 서술하고자 하는 것이다. 보다 구체적으로는 프로젝트의 수행에 있어서 초미의 관심사가 되었던 미들웨어와 개발프레임워크의 적용, DB 아키텍처의 선정과 슬림화, 그리고 성능 개선의 관점에서 이루어진 기술적 의사결정에 주된 초점을 두고자 한다. 이러한 분석 및 서술 과정에서 초대형 IT 인프라 구축 프로젝트의 기술적 관리에 대한 Best Practice가 소개되고 논의될 수 있는 계기가 마련될 수 있기를 기대한다.

## 2. 배경 및 관련 사례

### 2.1 사례연구의 배경

SK텔레콤은 메인프레임에 기반한 기존의 노후화되고 경직된 마케팅 인프라로는 급변하는 이동통신시장 환경의 변화에 적절히 대응하기 어렵다는 판단에서 Unix 기반 시스템으로의 다중사이드 필요성을 절감하고 있었다. 향후 Convergence 환경에서 시장 선도자의 위치를 확보하기 위해서는 마케팅 프로세스 및 IT 인프라의 혁신이 필요한데, 이를 동시에 지원해 줄 수 있는 '차세대 마케팅 시스템' (즉, NGM 시스템)이 절실히 필요한 시점에 와 있었기 때문이다.

COIS 시스템은 NGM 시스템 이전에 업무용으로 사용되던 시스템으로 메인프레임에 기반한 업무 프로그램으로 구성되어 있다. 따라서 COIS 시스템은 메인프레임이 갖는 문제점을 지니고 있는 단점이 있다. 메인프레임은 독자적인 컴퓨터 구조로 인해 범용 시스템 소프트웨어의 사용이 어렵고, 응용 프로그램 또한 독자적인 버전이 필요하다. 또한 GUI기반의 업무 프로그램 개발이 불가능하여 개발 업무의 효율성을 저하시키며 라이선스, 유지보수, 업그레이드를 위한 비용이 상대적으로 높다는 단점을 가진다. 이와 같은 메인프레임의 단점들 때문에 COIS 시스템의 변경 필요성이 대두되었다.

최근에는 여러 서비스들을 융합하는 'IT 컨버전스'가 화두로 등장하는 등 IT 업계는 변화에 유연하게 대처할 수 있는 시스템을 필요로 하고 있다. SK텔레콤은 비록 오랫동안 COIS 시스템을 사용해왔지만 시스템을 확장하고 유연한 구조를 갖게 하는 관점에서 생각했을 때, 과연 10년 후에도 메인프레임 기반의 COIS 시스템을 계속 사용할 수 있을지에 대한 대답이 결국 NGM 프로젝트의 착수로 이어졌다고 볼 수 있다.

### 2.2 관련 사례

SK텔레콤이 직면한 문제와 마찬가지로, 다양한 분야의 기업들이 기업 내부의 데이터 및 고객관련 데이터를 관리하고 서비스에 이용하는데 있어서 많은 어려움을 느끼고, 신시장의 환경에 맞는 새로운 시스템을 구축한 사례가 있다.

1) COIS는 IBM 메인프레임 컴퓨터 기반의 '고객운영정보시스템'이다.  
 2) NGM은 Next Generation Marketing의 두문자로서 '차세대 마케팅'을 의미함.  
 3) NGM 프로젝트의 수행 결과 구축된 NGM 시스템의 공식 명칭은 "U.key"이다. U.key는 공모와 심사를 거쳐 선택된 차세대 시스템의 명칭으로서 "Ubiquitous Key"를 의미한다. 이 논문에서는 NGM 시스템과 U.key 시스템을 동일한 의미로 사용한다.

KTF의 경우도 비슷한 상황에서 새로운 시스템을 구축하여, 시스템의 다운타임을 줄이고 정보 리소스를 효율적으로 활용하고자 하였다. 시스템의 신뢰성 및 확장성을 높이고 요금부과 업무의 효율성을 높이기 위함이 주목적이었으며, 데이터베이스 시스템의 변경을 통해 데이터의 효율적인 클러스터링으로 문제를 해결하였다[1].

독일의 주요 모바일 통신사인 E-Plus Mobilfunk GmbH의 경우 고객이 수백만 명에 달하고 매년 급증하게 됨에 따라, 신규고객에 대한 서비스 활성 시간을 단축시키기 위해 신규 시스템을 구축하였다. 불필요한 정보의 최소화 및 매우 높은 가용성, 확장성 등을 고려하여 분산형 데이터베이스를 기반으로 하였으며 고객 서비스에 대한 요구를 수용할 수 있었다[2].

노스웨스트 항공사의 경우 415개의 비행기, 1700여개의 일일노선, 직원, 스케줄 등의 정보를 효율적으로 활용하기 위해 새로운 시스템을 구축하였다. 전 세계에 퍼져있는 로컬 데이터베이스 정보를 활용하기 위해 다양한 데이터베이스 시스템 플랫폼을 모두 활용할 수 있는 모습을 지향하였고, 분산 데이터베이스 시스템 환경에서 효율적인 복제 전략에 대한 연구를 수행하였다[3].

위의 사례들에서 알 수 있듯이, 기업이 얻게 되는 정보는 시간이 흐를수록 방대해지며 그와 함께 고객의 요구도 다양해지고 급증하게 된다. SK 텔레콤의 경우도 이와 같은 기업들이 처한 문제와 같은 상황을 맞이하였다. 뿐만 아니라 처리해야 하는 데이터의 양이 기존에는 찾아볼 수 없을 정도로 거대하다는 문제점이 있었다.

이러한 문제점들을 해결하기 위해서는 기본적으로 분산 데이터베이스 시스템을 구축하게 되는데, SK텔레콤의 경우는 단순히 분산 데이터베이스 구축으로만은 해결하기 힘든 성격의 데이터와 서비스 요구를 가지고 있었다. 기존의 다른 기업들의 사례와 같은 방식으로 적용하기에는 SK텔레콤의 데이터는 훨씬 방대하기 때문에, 그에 맞는 해결책이 필요하였다. 따라서 본 연구를 통해 거대한 기존 데이터, 급증하는 신규데이터 및 고객 서비스 처리 트랜잭션 등을 해결하기 위한 종합적인 시스템 모델을 찾게 되었다.

### 3. 1단계 프로젝트의 중단과 2단계 프로젝트의 시작

#### 3.1 1단계 프로젝트 추진 경과

NGM 시스템의 개발을 위해 글로벌 소프트웨어 패키지를 채택하기로 하고, 시스템 구축에 관한 계약을 Turn-key 방식에 의한 아웃소싱으로 결정한 것은 일반적으로 널리 알려진 원론적인 방법론을 적용한 것이라고 볼 수 있다. 그러나 글로벌 패키지 솔루션은 커스터마이징 하기가 대단히 까다롭고, 프로그램의 source

code를 제공하지 않기 때문에 NGM 시스템과 같이 한국 고유의 업무 특성을 담고 있는 시스템의 개발에 활용하기에는 적지 않은 어려움이 따르게 되었다.

프로젝트 개시 후 몇 달이 지나지 않아 기능 설계 단계에서 개발 틀의 문제가 발생하였다. 2004년 3월~11월 간 9개월에 걸친 기능 설계에서 A사의 개발프레임워크는 관련 기술인력이 부족하였고, 기술지원도 원활하지 않은 문제점을 드러냈다. 또한 개발자가 사용하기에도 불편하여 결국 개발 틀을 변경하기에 이르렀다. 이 과정에서 많은 재 작업이 발생하였고, 개발설계서는 요구 품질에 미달하는 판정을 받게 되었으며, 결국 개발 일정에서 2개월 이상 지연 요인이 발생하였다.

기능 설계에서 나타났던 문제점은 계속 누적되어 2005년에 접어들어서는 심각한 상황에 봉착하게 되었다. 프로젝트 관리 측면에서는 Governance 확보가 미흡했으며, 프로젝트 관리 등의 문제가 노정되어 개발 진척도는 50%에 불과한 상황이었다. 기술적 측면에서는 기본 패키지의 기능적 한계 및 유연성 결여, DB 성능 및 확장성 제약 등의 심각한 문제가 발생하여 프로젝트의 정상적 추진이 어려운 상황에 봉착하게 되었다. 결국 2005년 3월, NGM 추진본부는 프로젝트의 추진 중단을 선언하기에 이르렀다. NGM 프로젝트는 처음부터 재검토하여 Re-planning을 실시한 후 구축을 재개하기로 결정하였다[4].

#### 3.2 2단계 프로젝트의 재 추진

NGM 프로젝트는 세계적으로도 유례를 찾아보기 어려울 만큼 규모가 큰 프로젝트이다. 대규모의 데이터와 트랜잭션을 포함하고 있으며, 업무 프로세스를 틀에 기반하여 일반화하는 작업을 병행하여 수행하였다. 또한 메인프레임에서 불가능했던 추가 요구사항들이 도출되면서 전체 시스템의 트랜잭션 규모가 크게 증가하였고 이에 따라 시스템의 복잡도 또한 기하급수적으로 증가하였다.

시스템의 복잡도가 증가하면 프로젝트를 진행하는 입장에서는 전체 시스템을 조감할 수 있는 global view를 도출해내기 어려운 단점이 생긴다. 1단계의 NGM 프로젝트에서도 이러한 문제점이 발생하였다. 또한 복잡한 시스템의 성능 목표치를 충족시키는 부분에도 어려움이 발생하는데, NGM 프로젝트에서도 선정된 하드웨어로는 성능 목표치를 달성하기가 어려웠다. 성능에 대한 문제점은 Java 기반의 개발프레임워크 적용이 중요한 원인을 제공하였으며, 이에 따라 C 기반의 개발프레임워크 적용 필요성이 제기되었다.

또한 A사의 패키지 솔루션은 DB 서버에서 비즈니스 로직을 처리하는 구조이기 때문에 NGM 프로젝트와 같이 대용량의 데이터에 대량의 트랜잭션을 처리해야 할

경우 DB 서버에 많은 부하를 발생시키는 단점이 존재한다. 또한, NGM에서 요구하는 dynamic 마케팅 캠페인 기능을 추가로 개발하는데 있어서의 유연성이 결여되어 있었다.

프로젝트 관리 관점에서 보면 NGM 프로젝트는 규모, 구조, 기술 등 세 가지 측면의 리스크를 모두 가지고 있는 프로젝트이다.<sup>4)</sup> 이러한 리스크에 대한 고려가 2단계 프로젝트의 Re-plan에 그대로 반영되었다. 우선 Turn-key 방식의 문제점을 해소하기 위하여 SK텔레콤이 구축의 총책임을 맡고, SK C&C가 시스템 이행에 대한 책임을 지는 Governance 체제를 확립하였다[4].

우리나라의 이동통신 시장은 세계에서 가장 치열하고 역동적인 경쟁장이다. 현재 세계에서 NGM에서 요구하는 수준의 기능들을 수행하는 회사는 찾을 수 없으며 이를 지원할 수 있는 패키지 솔루션은 더욱더 있을 수 없는 일이다. 그러므로, 기술적으로는 원점부터 재검토하여 다음과 같이 이전과는 근본적으로 다른 접근방식을 채택하기로 하였다.

첫째, 검증된 상용 솔루션을 기반으로 한 기술 아키텍처를 자체적으로 수립한다. 하나의 중심 패키지 적용을 지양하고 부분별 솔루션을 이용하여 맞춤형 시스템을 개발한다.

둘째, C 기반의 커스터마이징된 개발프레임워크를 활용하여 자체 개발의 생산성을 획기적으로 제고한다.

셋째, 독립적인 사용자 인터페이스를 도입하여 적용한다.

넷째, DB 아키텍처를 재검토하여 새로운 기술 아키텍처에 가장 적합한 방안을 모색한다.

1단계 프로젝트의 중단에 따른 교훈을 거울삼아 Re-plan을 수립하고, 이에 따라 새로운 아키텍처와 개발 방식을 적용하여 2단계 프로젝트를 추진하는 과정에서 SK텔레콤은 수많은 기술적 난제를 헤쳐 나가며 중요한 의사결정을 수행하여야 했다.

기본적으로 NGM 프로젝트는 메인프레임 기반의 시스템을 Unix 시스템으로 다운사이징하면서 시스템의 유연성을 확보하고 신규 서비스의 확장성을 도모하기 위하여 수행되었다. 그러나 기존 시스템의 규모에 새로이 추가된 기능들은 엄청난 양의 데이터와 트랜잭션을 필요로 하여 시스템 구축과 도입을 어렵게 하는 요인이 되었다. 이에 따라 프로젝트의 성공을 위해서 유연한 시스템 구조와 성능을 보장하는 미들웨어 및 개발프레임워크, 대용량의 데이터를 효과적으로 다룰 수 있는 DB 아키텍처, 그리고 주어진 자원과 아키텍처에서 최선의 성능을 확보하기 위한 성능 튜닝이라는 기술적인 이슈

를 도출하여 이를 해결하기 위해 노력하였다.

따라서 2단계 프로젝트에서 관심을 집중한 기술적 의사결정 영역은 크게 ① 미들웨어 및 개발프레임워크의 적용, ② DB 아키텍처의 결정 및 슬림화, ③ 시스템 성능 개선 등의 세 분야로 집약할 수 있다. 본 논문의 나머지 장에서 이들 각각의 분야를 구체적으로 살펴보고자 한다.

## 4. 미들웨어 및 개발프레임워크의 적용

### 4.1 미들웨어 및 개발프레임워크의 필요성

NGM 프로젝트의 중요한 화두 중 하나는 '오픈 아키텍처'이다. 이것은 Unix로 다운사이징 하면서 표준을 따르는 여러 미들웨어를 도입하는 것을 의미한다. 미들웨어를 도입하여 얻을 수 있는 장점은 다음과 같다.

첫째, 표준을 따르는 검증된 모듈을 사용할 수 있다. NGM 시스템은 여러 모듈로 나누어져 있는데, 각각의 모듈은 그에 맞는 기능적 요구사항을 표준으로 정의하고 있으며 각 미들웨어는 이에 따라 표준에 맞는 하나의 모듈만을 담당하게 된다. 따라서 시스템의 구성이 변경되었을 때도 변경된 이외의 부분에 미치는 영향은 최소화되는 시스템을 구축할 수 있다.

둘째, 시스템의 성능을 최적화할 수 있다. 전체 시스템이 독립적으로 구성되지 않는다면 해당 시스템의 복잡도는 매우 높아진다. 그러나 각 모듈을 담당하는 미들웨어가 존재할 경우, 미들웨어가 해당 모듈의 표준에만 집중할 수 있으므로 오류를 최소화하고 성능을 높일 수 있다. 또한 각각의 모듈로 전체 시스템을 구성하면 시스템 복잡도를 낮출 수 있으므로 Global view를 고려하는데 용이한 장점이 존재한다.

미들웨어와 마찬가지로 개발프레임워크 또한 전체 시스템의 유연성 및 성능에 많은 영향을 미친다. 전체 시스템이 여러 모듈로 나누어지는 경우 각 모듈들은 서로 간의 기능을 호출하는 과정을 필요로 하게 된다. 만약 공통으로 사용되는 기능들을 잘 구성된 개발프레임워크로 만든다면 다음과 같은 장점을 가질 수 있다.

첫째, 개발 생산성을 향상시킨다. 시스템에서 공통으로 사용하는 기능은 많은 반복적인 작업을 필요로 한다. 예를 들어, DB로부터 데이터를 가져올 경우 커넥션을 맺고, 질의문을 전송하는 등 상당한 작업이 반복된다. 이러한 부분을 하나로 묶음으로써 반복 작업을 줄일 수 있게 되고 개발 생산성이 높아진다.

둘째, 성능 향상 및 에러 검출에 용이하다. 반복적으로 사용되는 기능을 하나로 묶어 놓기 때문에 시스템 성능을 튜닝하는데 있어서 여러 부분을 수정할 필요가 없어진다. 즉, 공통적인 한 부분에 대한 수정이 전체 시스템으로 확장되어 반영된다. 마찬가지로 오류가 존재할

4) 개발 프로젝트는 규모가 클수록, 구조가 복잡할수록, 그리고 적용기술이 새로운 기술일수록 급격히 리스크가 증대된다[5,6].

표 1 TPM과 WAS 후보 비교

제품	Tmax/JEUS	Tuxedo/WebLogic
J2EE 표준 인증	O	O
OSI, X/Open DTP 표준	O	O
TPM 구조	Peer-to-Peer 모든 노드에서 Admin 기능 수행 가능	Master-Slave 마스터 노드 다운 시 대응 어려움
TPM 통신방식	Stream Pipe	IPC Queue
구축사례	적음	많음
Customizing	국내업체, 빠른 대응	외국업체, 느린 대응

경우에도 개발프레임워크의 수정은 전체 시스템으로 확장될 수 있다. 만약 개발프레임워크가 존재하지 않는다면 반복 작업된 부분을 모두 수정하는 비효율적인 작업이 필요하게 된다.

4.2 미들웨어 및 개발프레임워크 선정 주요 요소

미들웨어와 개발프레임워크는 잘 짜여진 아키텍처에 필수적인 요소이다. 그러나 여러 layer로 나누어진 시스템은 항상 성능에 대한 위험요소를 안고 있다. 또한 전체 시스템이 특정 미들웨어에 의존적이 될 가능성도 존재한다. 따라서 미들웨어와 개발프레임워크를 선정하는데 있어서는 여러 가지 요소들을 고려해야 한다. 그림 1은 미들웨어 및 개발프레임워크를 선정하는 데 있어서 고려해야 할 주요 요소들을 보인 것이다.

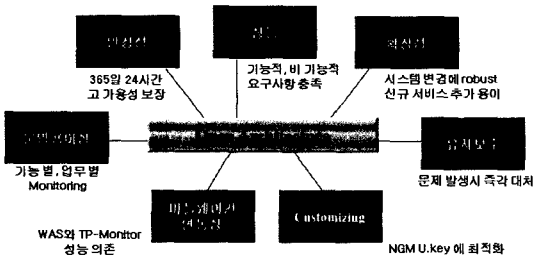


그림 1 미들웨어 및 개발프레임워크 선정 주요 요소

오픈 아키텍처의 흐름에 따라 여러 요소를 고려할 수 있지만 특히 NGM 프로젝트에서는 커스터마이징에 대한 요구사항을 추가하였다. 세계 최대 규모의 프로젝트인 만큼 기존의 사례가 없는 경우도 많았고, 이에 따라 기존의 솔루션들을 NGM 프로젝트에 적합하게 변경하는 과정이 필요한 부분이 많았다. 따라서 NGM 프로젝트를 위한 미들웨어 및 개발프레임워크 선정에 위한 다차원적이고 거시적인 시스템 구조 재구성 계획을 수행하게 되었다.

4.3 미들웨어의 선정

TPM과 WAS<sup>5)</sup> 미들웨어의 선정은 시스템 안정성과 성능에 큰 영향을 미치게 된다. 일반적으로 TPM과

WAS를 선정할 때에는 성능 최적화와 미들웨어 간의 연동성을 위해 하나의 벤더를 선택하는 경우가 많다. 이에 대한 후보로 NGM 프로젝트에서는 TmaxSoft사의 'Tmax/JEUS' 조합과 BEA사의 'Tuxedo/WebLogic' 조합을 고려하였다[7,8].

표 1에서 보는 바와 같이 후보 제품들은 모두 표준에 따라 개발되었기 때문에 어느 후보를 선택하더라도 시스템을 구성하는 것은 가능하다. NGM과 같은 대형 프로젝트에 구축사례가 상대적으로 적고 회사의 규모도 훨씬 작은 국내업체의 제품을 채택하는 것은 커다란 리스크가 동반되는 일이지만, 유례없는 대용량의 데이터와 트랜잭션 처리가 필요했기 때문에 C 기반이면서 커스터마이징에 유리한 TmaxSoft사의 제품을 도입하기로 결정하였다. 결과적으로, 수많은 튜닝과 커스터마이징을 통해 최적화된 미들웨어 성능을 얻을 수 있었으며 NGM의 성공에 중요한 요인이 되었다.

4.4 구현 아키텍처 및 개발프레임워크

NGM 프로젝트에서 미들웨어 및 개발프레임워크의 선정은 엄격한 검증 절차를 거쳐서 결정되었으며, 그림 2는 이러한 과정을 거쳐서 최종 선정된 미들웨어 및 개발프레임워크를 나타낸 것이다.

특히 TmaxSoft사의 ProFrame의 경우 DB 접근과 애플리케이션의 공통 로직을 담당하는 개발프레임워크로서 C 언어 기반으로 커스터마이징 되었다. ProFrame은 핵심적인 공통 처리 모듈을 포함하여 전체 시스템의 성능 향상 및 개발 생산성 향상에 많은 역할을 담당하였다.

그림 3은 NGM 프로젝트에서 ProFrame이 담당한 역할을 개략적으로 나타낸 것이다. 개발프레임워크의 역할은 메모리 관리, 캐싱, 트랜잭션 처리, 로깅, 리커버리 등 크게 다섯 개의 영역으로 나누어진다. 이 기능들은 전체 시스템 및 비즈니스 로직을 처리하는 데 있어서 모든 모듈이 사용해야 하는 공통적인 기능이며, 시스템의 성능에 가장 큰 영향을 미치는 DB 접근에 대한 부분을 포함한다. 따라서 비즈니스 로직을 수행하는 모듈에서는 많은 양의 반복 작업을 줄일 수 있으며 성능에 대한 보장을 받을 수 있다.

5) TPM=TP Monitor, WAS=Web Application Server

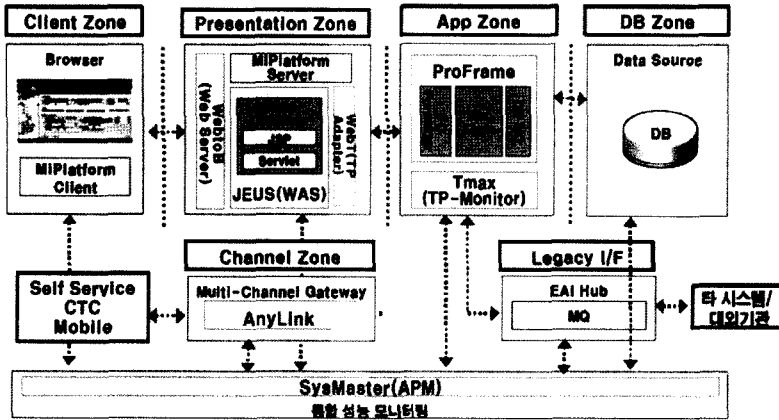


그림 2 전체 시스템 아키텍처

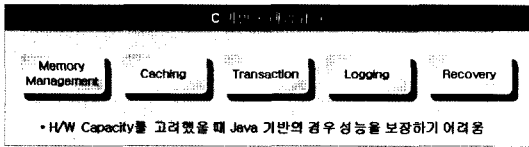


그림 3 개발프레임워크의 역할

특히 NGM 프로젝트와 같이 세계 최대 규모의 데이터와 트랜잭션을 처리해야 하는 시스템의 경우 Java 기반으로 개발된 개발프레임워크의 성공 사례는 찾아볼 수 없으며, 이에 따라 C 언어 기반으로 커스터마이징된 개발프레임워크를 사용함으로써 프로젝트의 성능 향상을 가져왔다고 볼 수 있다.

일반적으로 개발프레임워크의 도입은 그림 4의 개발프레임워크 학습 곡선에서 보는 바와 같이 초기에 도입기와 적응기를 필요로 한다. 더욱이 NGM 프로젝트의 경우 이미 구축된 개발프레임워크라기 보다는 C 언어 기반의 새로운 개발프레임워크를 구축했기 때문에 초기에는 개발 생산성의 향상이 눈에 띄지 않거나 오히려 감소하는 경향을 나타내었다. 그러나 구축과 적응의 기간이 지난 이후에는 급격한 개발 생산성 향상의 효과를

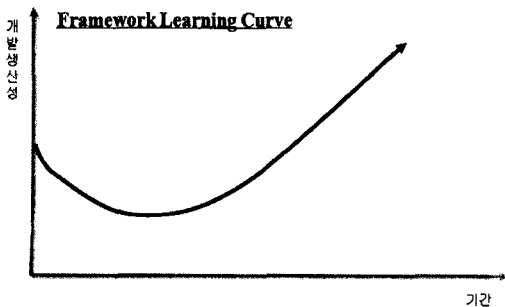


그림 4 개발프레임워크 학습 곡선

보일 수 있었으며, 이외에도 성능 및 안정성 측면에서도 좋은 결과를 기대할 수 있게 되었다.

#### 4.5 미들웨어 및 개발프레임워크 도입 결과

미들웨어 및 개발프레임워크의 도입은 다음과 같은 측면에서 전체 시스템에 많은 이득을 가져다주었으며, 결국 이와 같은 장점들이 NGM 프로젝트의 성공적인 수행 결과로 나타났다고 볼 수 있다.

첫째, 시스템 변경 및 신규 서비스 접목이 용이한 유연한 시스템 구조를 갖추게 되었다. 메인프레임 기반의 단일 시스템에서 전체 서비스가 동작하는 구조에 비하여 multi-layer를 갖는 구조는 표준에 따른 미들웨어를 채택하기 때문에 새로운 서비스를 도입하거나 시스템을 변경하는 것이 용이하다는 장점을 가진다. 또한 layer 간 느슨한 연결 구조를 갖도록 표준을 따르기 때문에 특정 미들웨어에 의존하게 되는 문제점을 피할 수 있었다.

둘째, 터미널 기반 업무 프로그램을 웹 기반의 GUI로 변환하였다. 사용자 측면에서 터미널 기반의 업무 프로그램은 업무 숙련에 매우 많은 시간을 필요로 하지만, GUI 기반의 사용자 인터페이스는 새로운 직원도 쉽게 업무에 숙련할 수 있는 구조로서 업무 처리 능력을 증대시키는 결과를 가져올 수 있었다.

셋째, 각 모듈의 독립성에 의하여 안정적인 서비스를 가능하게 했다. 전체 시스템에서 각각의 모듈이 독립적으로 동작하기 때문에 혹시 문제가 생길 경우 해당 모듈에 대한 수정만으로 전체 시스템을 복구할 수 있었다. 또한 모듈 별 유지보수 및 업그레이드, 지속적인 테스트와 소스 레벨의 분석으로 미들웨어에 대한 철저한 검증이 가능했다.

### 5. DB 아키텍처의 결정 및 슬림화

#### 5.1 분산 DB 아키텍처

NGM 프로젝트는 대량의 데이터와 트랜잭션을 처리

해야 하는 시간적, 공간적 제약을 모두 가지고 있는 대형 프로젝트이다. DBMS로는 Oracle의 RAC<sup>6)</sup> 기술을 탑재한분산형 DB 서버가 선정되었다. Oracle RAC은 분산환경에서 원격 데이터의 접속이 일어날 때 해당 노드의 cache를 먼저 참조하여 원하는 데이터가 있을 경우 디스크 접근 없이 바로 데이터를 전송받는 방식을 이용하여 원격 데이터 접속 비용을 최소화하는 기술이다(Floss, 2004). 그러나 현실적으로 DB 서버 확장은 scale-up 요소(서버 2배 증설 시 성능이 2배 증가하는 것을 가능하는 척도)가 1이하로 나타나기 때문에 수평적인 DB 서버 노드를 추가하는 것만으로는 NGM 프로젝트에서 요구하는 대용량의 데이터와 트랜잭션을 효과적으로 처리할 수 없게 된다.

NGM 프로젝트의 트랜잭션은 업무 유형에 따라 Online 트랜잭션과 Batch 트랜잭션으로 나눌 수 있다. Online 트랜잭션은 업무 프로그램이나 웹을 통한 사용자 인터페이스에 의하여 발생하는 트랜잭션을 의미하고, Batch 트랜잭션은 사용자 요금 정산 등 일괄적이고 정기적인 처리를 필요로 하는 트랜잭션을 의미한다. Online 트랜잭션은 처리 시간이 짧지만 자주 일어나며, Batch 트랜잭션은 자주 발생하지는 않지만 매우 긴 처리 시간을 요구하는 특성이 있다. 두 종류의 트랜잭션은 특성은 다르지만 공통으로 사용하는 데이터가 존재하기 때문에 상호 간섭에 의하여 성능 저하를 발생시키는 원인이 되기도 한다[9].

이와 같은 문제점들은 결국 전체 시스템의 확장성 및 성능에 영향을 미치게 된다. 특히 DB 서버의 부하는 성능 저하가 발생하는 대부분의 경우 중 가장 치명적인 부분이기 때문에 이에 대한 해결방법이 제시되어야만 한다. 따라서 NGM 프로젝트에서도 이를 해결하고 최적화된 데이터베이스를 구성하기 위한 다각적인 방안을 검토할 필요가 있었다.

**5.2 DB 구성의 기본 구조 : 중앙 집중 방식과 완전 분산 방식**

NGM 프로젝트에서 성능에 대한 문제점 없이 대용량 데이터베이스의 확장성을 고려하기 위하여 가장 먼저 두 가지의 간단한 구조를 생각해 볼 수 있다[10]. 첫 번째는 Online과 Batch 트랜잭션이 하나의 RAC에서 이루어지는 중앙 집중 방식이다. 그림 5는 중앙 집중 방식의 DB 구성 형태를 보여준다. 이 경우 시스템이 간단해지기 때문에 프로그램 개발 및 유지보수 작업이 용이해지는 장점을 갖는다. 그러나 확장성에 문제가 있으며, Online 트랜잭션과 Batch 트랜잭션의 상호 간섭이 빈번하게 일어나는 단점을 갖는다[9].

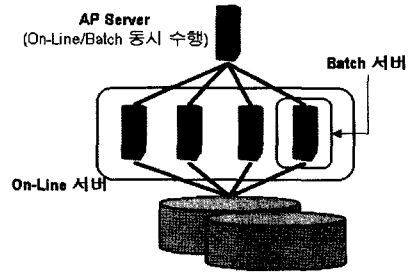


그림 5 중앙 집중 방식 DB 구성

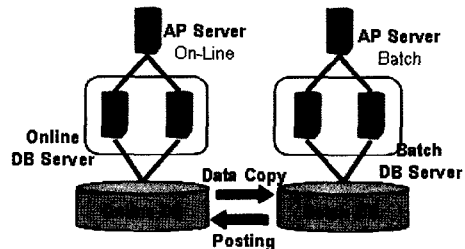


그림 6 완전 분산 방식 DB 구성

두 번째 구조는 Online 업무와 Batch 업무를 분리한 완전 분산 방식의 모델이다. 그림 6과 같은 구성을 보이며, 이 경우 중앙 집중 방식의 문제점이었던 확장성에 대한 제약을 해결할 수 있는 장점이 있다. 그러나 완전 분산 방식의 경우 공통으로 사용되는 데이터의 중복과 복사로 인한 성능 저하라는 문제점이 발생한다.

표 2는 중앙 집중 방식과 완전 분산 방식을 비교한 것이다. 확장성 측면에서 완전 분산형 구조는 노드 추가에 따른 scale-up 요소가 중앙 집중 방식에 비하여 상대적으로 크게 나타난다. 이것은 확장성 측면에서 완전 분산 방식이 더 좋은 성능을 보임을 나타내고 있다. 시스템 성능은 중앙 집중 방식이 Online 트랜잭션과 Batch 트랜잭션의 상호 간섭이 존재한다 하더라도 분산 방식의 데이터 일관성 유지를 위한 부가적인 작업에 소모되는 오버헤드에 비하여 양호한 성능을 보인다. 또한 가용성 측면에서도 중앙 집중 방식은 같은 일을 하는 여러 노드들이 작업을 분담하여 처리하게 됨으로써 완전 분산 방식보다 상대적으로 높은 가용성을 보장한다. 마지막으로 응용 프로그램 개발의 효율성 측면에서도 완전 분산 방식은 개발자에게 시스템의 투명성을 보장해 주기 어려운 구조이기 때문에 중앙 집중 방식이 더 적합하다고 볼 수 있다. 결론적으로 두 가지의 기본적인 DB 구성 방식을 비교해 보면 확장성의 측면을 제외하면 분산 구조 방식을 선호할 이유가 없다는 결론을 내릴 수 있다.

**5.3 NGM 프로젝트를 위한 융합 모델**

앞에서 언급한 바와 같이 중앙 집중 방식의 single

6) RAC = Real Application Clusters

표 2 중앙 집중 방식과 완전 분산 방식 비교

비교사항	중앙 집중형	완전 분산형	비고
확장성	- 4노드 적용 시, scale-up 0.825 - 6노드 적용 시, scale-up 0.783 (6노드 이상 검증된 사례 없음) - Oracle RAC 기술향상 기대 (Oracle 9i 대비 10g의 성능 28% 향상)	- 2x2 RAC 분산 환경 시, scale-up 0.9 - 차후 4x4 RAC으로 확장 가능 - Online/Batch 부하에 따라 노드 추가 용이	완전 분산형이 더 좋음
성능	- Online과 Batch간 간섭 문제 잠재	- 2PC와 DB link로 인한 overhead - Online과 Batch간 간섭 문제 잠재	중앙 집중형이 더 좋음
가용성 및 안정성	- DB장에서 전체 서비스 중단 - 하나의 DB노드 장애 시, 분산DB보다 가용성이 떨어짐	- DB장애 시 특정 서비스 중단 - 하나의 DB노드 장애 시, 가용성이 떨어짐 - 2PC, DB link에 의한 Join으로 인해 안정성 저하 (In doubt 트랜잭션, 2PC 장애처리 문제 등)	중앙 집중형이 더 좋음
어플리케이션 개발 효율성	- 어플리케이션에 DB Transparency가 보장됨	- 어플리케이션에서 Transparency 문제 발생 - 2PC를 위한 추가 개발 부하 발생	중앙 집중형이 더 좋음
결론	확장성 외에는 분산 DB를 선호할 이유가 없음		

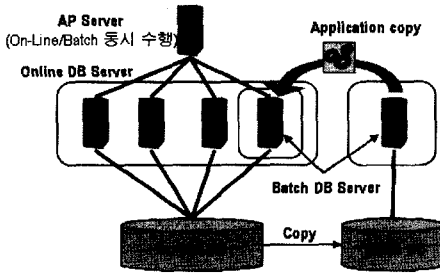


그림 7 Hybrid Model

DB는 확장성이 부족하고 Online 업무와 Batch 업무 사이의 간섭으로 인하여 성능 문제가 발생할 가능성이 존재한다. 한편 분산 DB를 도입하면 트랜잭션 간 공유 데이터가 존재하고, 2 Phase Commit 및 DB Link 부하 때문에 성능 문제를 해결하기 어렵다. 따라서 두 기본 형태를 융합한 하이브리드 방식의 모델을 생각할 수 있다. 그림 7은 NGM 프로젝트를 위한 하이브리드 DB 구성을 나타낸 것이다.

제시한 하이브리드 모델은 중앙 집중 방식으로 연결된 서버에 Batch 업무 전용 서버를 도입하고, Batch 업무와 관련한 매출 추정 및 검증을 수행하는 서버를 추가하는 형태로 구성된다. 이를 위하여 부분적인 데이터의 복제가 이루어지는 부분은 존재하지만 Batch 업무 전용 서버에 의하여 확장성을 증대시키고 Online 트랜잭션과 Batch 트랜잭션간의 상호 간섭을 최소화할 수 있는 장점이 존재한다.

SK텔레콤의 경우 대용량의 통화기록 정보 및 서비스 내역들을 기초로 하여 각종 매출 추정 및 검증작업이 수행되어야 하는데, 이러한 대용량 Batch작업을 수행하는 경우 Online 트랜잭션의 간섭문제와 더불어 데이터 및 처리 결과의 copy문제 등이 가장 큰 문제이다. 또한, Batch작업을 수행하는 어플리케이션이 다양한 요구조건

에 따라 자주 변경되는 특성을 가지고 있으므로, Batch 작업에 대한 노드가 상당히 큰 문제점이 있다. 따라서 하이브리드 DB 구성 모델에서는 별도의 Batch작업 전용 서버와 DB를 구성하고, 매출 추정 및 검증 등의 어플리케이션을 실행 및 보장하는 작업을 처리한다. 모든 Batch작업에 대한 작업이 완료되면, 완성된 버전의 어플리케이션을 Online 서버군쪽의 Batch 작업 담당 서버로 copy하고, 최종 Batch작업을 단시간에 처리한다. 이러한 방식으로 Online 트랜잭션과의 간섭으로 인한 성능저하를 최소화하고 Batch작업의 확장성을 보장하는 한편, 기존 시스템의 가장 큰 문제점이었던 시간적 제약을 크게 완화시킬 수 있었다.<sup>7)</sup>

특히 이 경우 Online 업무를 처리하는 DB로부터 일부 데이터를 Batch 업무를 처리하는 DB로 copy하는 작업이 시스템 성능에 중요한 영향을 미치게 되는데, 이때 사용하는 기술로서 크게 Disk Copy방식과 log기반 방식이 존재한다. SK텔레콤의 경우 Disk Copy방식을 기존부터 사용하고 있던 상황이었다. 기존 방식의 변경 가능성을 타진해보기 위해 Redo log 기반 기술인 Oracle Streams를 고려하였다[9]. Oracle RAC을 사용하여 대용량의 데이터 및 트랜잭션을 처리하는 기업 중 하나인 P사와 SK텔레콤의 NGM 시스템의 상황을 비교하여 Oracle Streams의 적용 가능성을 살펴보았으며, 테이블 수, 트랜잭션 양, copy 대상 데이터 양, 자원 사용률 등의 요소를 살펴본 결과, NGM 시스템에 일부 원부 데이터를 대상으로 Streams 기술을 적용하여 data copy 작업을 수행할 수 있는 가능성을 엿볼 수 있었다. 하지만, Oracle Streams를 직접 적용하기 위해서는 Batch 업무를 처리하는 DB로 copy되어야 하는 모든 데이터에

7) 실제로 이러한 모델을 통하여 정기청구 업무시 청구업무에 제약이 존재하던 문제를 해결할 수 있었으며, Online 트랜잭션에 대한 시간을 1/3 수준으로 감감시키는 효과를 얻을 수 있었다.



대한 예측이 아닌 실질적 검증이 필요하다는 문제점이 발생하였다. 따라서, SK텔레콤은 기존의 Disk Copy 방식을 그대로 유지하기로 하였고 copy 효율성을 높이기 위하여 스토리지 구성을 달리하는 방식을 취하였다.

**5.4 DB 슬림화**

NGM 시스템은 부분적인 분산을 통하여 확장성 및 성능 측면의 향상을 가져올 수 있었지만 이러한 문제점에 대한 보다 근본적인 접근이 추가로 필요하였다. 확장성 및 성능의 문제는 결국 DB 시스템 자체에서 중복되고 사용 빈도가 낮은 데이터의 존재로 인하여 전체적의 크기가 증가했기 때문에 발생한 것으로 볼 수 있기 때문이다. 따라서 NGM 프로젝트에서도 이러한 부분에 대한 해결을 시도했으며, 이를 DB 슬림화로 명명하였다.

그림 8은 DB 슬림화의 목적을 도식화한 것이다. 결국 DB 슬림화란 데이터 중복을 제거함으로써 최소한의 용량으로 데이터를 유지하고 사용 빈도에 따라 핵심 영역과 기타 영역을 분리, 분산 배치함으로써 핵심 영역의 성능을 향상시키고 유지보수를 용이하게 함을 의미한다. NGM 프로젝트에서는 DB 슬림화를 수행하기 위해서 다음과 같은 원칙을 고려하였다.

- ① 전체 크기의 90% 이상을 차지하는 테이블을 분석 대상으로 한다.
- ② 데이터의 생명주기를 고려하여 최소한의 DB 보관주기를 원칙으로 한다.
- ③ 별도의 DB 영역을 설정하여 조회 빈도, 읽기 전용 등의 특성을 부여한다.
- ④ DB 영역에 대한 Batch 기준을 수립한다.
- ⑤ 업무 처리를 위한 최소기간의 데이터, 그리고 빈번한 조회를 필요로 하는 데이터는 핵심 영역에 배치한다.
- ⑥ 화면 조회 시 각각의 DB를 나누어서 가져올 수 있도록 설계한다.

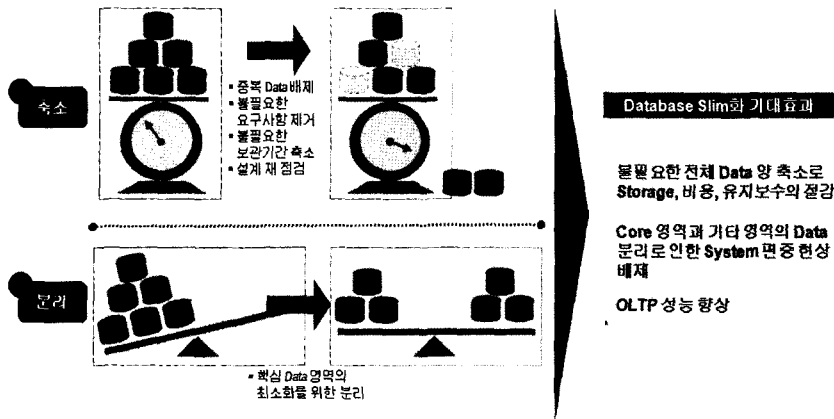


그림 8 DB 슬림화의 목적

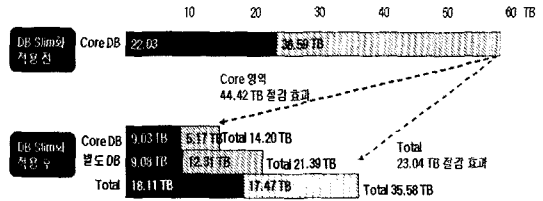


그림 9 DB 슬림화로 인한 10년 후의 예상 DB 크기

이러한 원칙 아래 응용 소프트웨어 개발에 재 작업이 일어나지 않도록 하는 방향으로 대상을 선정하고, 데이터를 분리했을 경우 사용자 화면에 나타나는 속도 및 조회 기간 만족에 대한 사항을 고려하여 슬림화를 적용하였다.

DB 슬림화로 인하여 DB를 핵심 영역과 별도 DB로 분리한 후 10년 후의 데이터베이스 크기를 예상했을 때 그림 9와 같은 결과를 얻을 수 있었다. 여기에서는 10년 후의 크기를 테이블 별 연간 증가량 \* 보존기간(최대 10년)으로 계산하였다. 그림 9에서 볼 수 있듯이 DB 슬림화를 적용한 후에 약 23TB의 저장 공간 절약 효과를 낼 수 있다는 결론을 얻었다.

**6. 시스템 성능 개선**

**6.1 성능 향상의 이유**

NGM 시스템의 성능 향상에는 여러 도전적인 문제가 존재하였다. NGM 시스템이 다루는 데이터베이스는 세계 최대 수준의 크기로, 거시적으로 접근한 구조개선 뿐만 아니라 미시적 관점에서 세밀한 관리가 필요하였다 [11]. 또한 NGM 시스템의 서비스가 매우 방대하고 다양하기 때문에 발생하는 시스템의 복잡도를 제대로 관리하여야만 시스템 개선으로 인한 성능 향상을 온전히 이끌어낼 수 있는 것이다. 또한 NGM 시스템이 실현하

고자 하는 서비스 품질을 확보하기 위해서는 각 응답 시간 요구 조건을 만족시키도록 시스템에 대한 추가 튜닝이 필요하였다. 이와 같은 이유로 NGM 프로젝트에서는 아키텍처 개선 이후 성능 향상을 위해 많은 노력을 기울이게 되었다.

**6.2 성능 향상을 위한 접근 방법**

NGM 시스템에서는 성능 개선의 세 가지 이슈인 복잡도 극복, DB 성능 향상, 응답 시간 개선을 위해 각각에 적합한 접근 방법을 선택하였다.

가장 먼저 복잡도를 극복하기 위해서는 여러 서비스의 다양한 검색 요구 조건을 최대한 공통 요소로 묶어 내어 최적화 하는 과정을 거쳤다. 다양한 검색 조건을 Grouping 한 후 이를 최적화하였고, 이를 최종적으로 Merge하여 이용하였다.

DB 성능 향상을 위해서는 DB와 Application에 걸친 최적화 작업을 수행하였다. DB 접근 자체를 최소화하고 최적화했으며, DB와 Application 간의 통신을 최적화하였다. 또한 병렬 수행을 극대화하여 전반적인 성능 향상을 이끌어내었다.

마지막으로 응답 시간 개선을 위한 성능 개선 작업에서는 작업 특성 별로 전략을 수립하고 이후에 통합 테스트를 수행하였다. Online 작업은 SQL 튜닝 등의 전략적 측면에 초점을 맞추고, Batch 작업은 작업 프로세스 전반에 걸친 최적화에 초점을 맞추었다. 이후 통합 테스트를 통해 전체 작업의 응답 시간 개선을 모색하였다.

**6.3 성능 개선 절차 및 사례**

NGM 프로젝트의 성능 개선 절차는 그림 10에서 보는 바와 같다. 개선 작업은 크게 Online 작업과 Batch 작업으로 나누어 튜닝한 후, 통합 테스트로 성능 평가를 하는 과정으로 이루어졌다.

Online 작업은 기초 인덱스 디자인부터 시작하여, 실행 계획을 기반으로 한 1차 SQL 튜닝, 그리고 트레이스를 근거로 한 2차 SQL 튜닝을 통해 응답 시간을 최소화하였다[12].

Batch 작업은 단위 프로그램 별로 튜닝을 하고, 전체 처리 구조를 변경하는 기능 처리 튜닝을 거친 후, 자원 전반에 걸친 작업 주기 튜닝을 거쳐, 처리 시간을 최소화하였다.

통합 테스트에서는 통합 시험 환경에서 전체 성능 평가를 하였고, 이 과정에서 다시 SQL을 개선하고 인덱스를 조정하는 작업이 추가로 이루어졌다. 또한 프로그램을 변경하는 작업을 거쳐 전체 성능 최적화를 수행하였다.

특히 프로그램 하단에서 발생하는 SQL 처리에 있어서 많은 튜닝작업이 필요했는데[14], 이는 대부분의 프로그램에서 여러 개의 테이블을 다양한 조건에 의한 조인연산을 통해 결과를 도출하게 되는데, 단순한 튜닝원리를 적용하지 않았기 때문이었다. 그 유형을 살펴보면 다음과 같다.

- Table Full Scan으로 인한 성능저하 발생
- Join 방식의 잘못된 사용
- 적절한 Hint의 사용을 하지 않음
- 병렬처리의 미사용
- 반복적인 loop성 작업에 대한 처리 미약
- Insert 향상을 위한 고려 필요
- 불필요한 In-line view 제거

실제로 현업에서 이루어지는 많은 개발 작업에서 위와 같은 튜닝에서 반드시 필요한 고려사항 및 작업들을 간과하여 수행하지 않아 시스템 성능이 매우 낮은 경우가 많았고, 이를 수정하여 NGM 시스템에서는 큰 성능

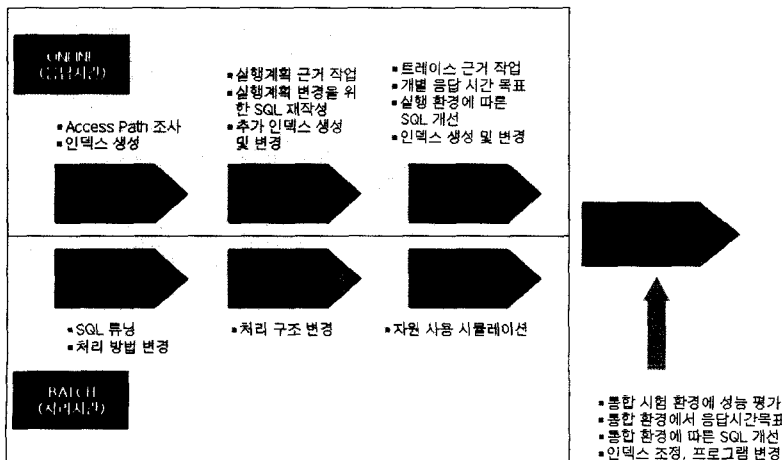


그림 10 성능 개선 절차

향상을 이룰 수 있었다. 이와 관련하여 표 3에 대표적인 튜닝 사례를 나타내었다.

**6.4 전반적인 성능 향상 결과 및 성능 개선의 성공요인**

이러한 성능 개선 노력을 통하여, 개별 작업의 성능 향상뿐만 아니라 전체적인 성능 수치도 향상되었다. 그림 11에서 보는 바와 같이 성능 테스트에서 Online Peak TPS, Batch Average TPS 등 여러 기준에서 성능 개선의 기준치(목표치)를 모두 초과 달성했음을 알 수 있다. 개별적으로는 표 4에서 보는 바와 같이 대표적인 Online 및 Batch 업무 모두에서 적게는 2배, 많게는 132배에 이르는 성능 향상을 달성하였다.

성능 개선 작업이 성공적으로 끝날 수 있었던 요인은 여러 가지가 있다. 첫째, 문제가 되는 핵심 부분을 제대로 파악하고, 각각에 맞는 접근 방법을 택해 문제를 해결했다는 점이다. 이로써 대용량 DB, 복잡도, 엄격한 응

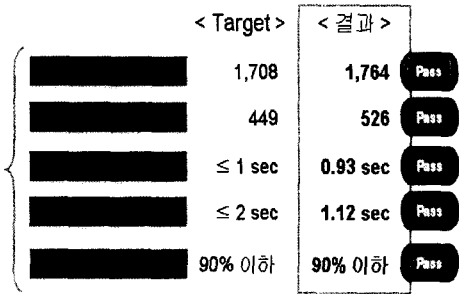


그림 11 NGM 성능 테스트 결과

답성 요구 등의 핵심 문제를 해결할 수 있었다.

둘째, 다양한 작업 원칙을 세우고 엄격히 준수하였다는 점이다. 거대한 시스템을 구축하는 협업 과정 중에서 발생할 수 있는 문제를 최소화하기 위해서 다양한 원칙

표 3 성능 개선 대표 사례

성능 문제 요인	상황 및 문제점	해결 방안
Table Full Scan	- 어플리케이션에서 2가지 조건이 입력으로 들어옴 - 조건1(1~4개), 조건2(1~2개) - 4x2개의 case를 OR연산으로 처리 - 참조되는 모든 Table에 Full scan이 발생함	- 대이타건수의 증가에 따라 실행계획을 분리 - OR연결 조건들에 대하여, 검색조건이 그룹별로 UNION ALL 연결 - 필수검색항목의 대상테이블이 Driving 테이블인지 확인[12] - 관련 Index검토 및 추가
Index Full Scan	- Index Full Scan은 불필요한 경우 발생하면 Bad Plan의 성격을 지니므로 SQL 수정 필요 - 수백만 건의 row를 대상으로 Index Full scan이 일어나는 경우 비효율적인 데이터 액세스 발생	- LEADING, INDEX, USE_NL 등의 Hint를 사용하여 실행계획을 제어
Loop 작업	- 여러 종류의 데이터 셋을 하나의 커서로 선언 - Loop 내에서 Fetch하며 처리함으로써 과도한 SQL호출 및 Random access 발생 - Parallel Query Option(PQO)을 사용하지 않아 단일 프로세스로 처리되고 있으며, CPU, Disk등의 자원 활용이 좋지 못함	- 데이터 셋을 로직 형태별로 분리하여 따로 한꺼번에 Update처리 - Array처리를 통해 Fetch성능 향상[13] - Merge, Hash Join, Full Table Scan, PQO 등을 적절히 활용 - parallel hint를 사용하여 단일 SQL이 CPU 및 Disk자원을 최대한으로 활용하도록 함

표 4 세부 성능 개선 결과

작업 형태	대상 작업	개선 전	개선 후	향상 (배)
Online	자동 납부 신청 등록	0.1초	0.002초	50
	법인 대표회선 조회	0.82초	0.31초	2.65
	법인 회선 조회	74초	1초	74
Batch	SKT자동납부신청 및 해지자료 추출	20분	10분	2
	인출요청 통계 생성 CMS	90분	10분	9
	자동납부 해지자 파일 생성	90분	10분	9
	은행인출결과 통계 생성 CMS	180분	5분	36
	신규고객 인출을 통계 생성 CMS	180분	10분	18
	외부정산 청구구분자료 생성	180분	30분	6
	3개월 미인출 대상 추출	240분	10분	24
	인출요청 제외대상자 처리	360분	20분	18
	통계 생성 공통적용 작업	5.674초	0.043초	132
	JUNE관리수수료 대상 추출	9시간 이상	643.13초	-
	자동납부할인 계산	10시간 이상	1시간 이내	-

을 세우고 이를 준수하였다.

셋째, 성능 개선 작업 중에 발생하는 요구들을 능동적으로 처리했다는 점이다. SQL 트레이싱 툴을 자체 개발해서 활용하는 등, 기술적인 도전들을 적극적으로 해결해 나갔다.

넷째, 성능 개선 작업이 프로젝트 전반에 걸쳐 진행되었다는 점이다. 튜닝 팀의 활동은 한 모듈에 국한되지 않고, 수평적으로 시스템 전반에 걸쳐 작업을 수행하였다. 모든 SQL에 ID를 부여하고 개발자와 직접 소통하는 적극성을 통해 전체적인 성능 개선을 성공적으로 이끌어낼 수 있게 된 것이다.

## 7. 논의 및 향후 과제

2006년 10월 9일 메인프레임 기반의 COIS는 Shut-down되어 가동을 멈추었고, Unix 기반의 NGM 시스템이 가동을 개시함으로써 차세대 마케팅을 본격 추진하기 위한 다운사이징 프로젝트는 성공적으로 완수되었다. 이 프로젝트를 성공적으로 완수함으로써 SK텔레콤은 통신업계 다운사이징의 선진 사례를 남길 수 있게 되었고, 향후 오픈 시스템을 활용한 혁신을 기대할 수 있게 되었다. 이 시점에서 지난 4년에 걸친 프로젝트 기간 동안의 경험에서 얻어진 교훈으로부터 프로젝트의 성공요인을 도출해 보고 향후의 과제를 전망해 보는 것은 매우 의미 있는 일이 될 것이다.

### 7.1 NGM 프로젝트의 성공요인

NGM 프로젝트의 성공요인은 다음과 같이 요약할 수 있다.

첫째, 꼼꼼한 검증과 논리적 분석을 통해 최적의 시스템 프레임워크와 DB 아키텍처를 설계하였으며, 그 안의 수많은 HW와 SW를 묶어서 실지로 연동시켰다. 개발 플랫폼을 도입하여 개발 생산성을 높였으며, DB 슬립화와 세밀한 성능튜닝을 통하여 시스템 성능을 확보할 수 있었다.

둘째, 원활한 기술지원 체계를 확보하여 벤더로부터 글로벌한 수준의 지원을 획득하였다. 2단계 프로젝트 초기에 HW 및 SW 납품업체인 Oracle (DBMS), HP (서버), Hitachi (스토리지) 등의 본사와 Global Virtual 협력계약을 체결하고 TF팀을 구성하여 필요시 조속한 기술지원을 제공받았다. 한편 미들웨어와 개발프레임워크 부문은 과감하게 국산 소프트웨어를 채택하고, 개발사인 TmaxSoft와 비즈니스 파트너 관계를 맺음으로써 상시적으로 기술과 인력 지원을 이끌어 낼 수 있었다.

셋째, 충분한 테스트와 사용자 교육이 프로젝트의 성공에 기여한 바가 크다. 시스템 오픈 일정을 무리하게 앞당기지 않고 통합 테스트와 사용자 교육에 충분한 시간과 자원을 할당하였다. 5개월 간 8차에 걸쳐 통합테스

트를 실시하였고 두 차례의 이행 리허설을 거쳤다. 또한 사용자의 시스템 숙련도를 보장하기 위하여 1만 명에 대하여 집합교육을 실시하였고, 추가로 8천 명에 대하여 전과교육 및 자가학습을 지원하였다.

마지막으로 조직적 요인으로서, 기본적으로 발주사의 Governance 확립, 참여 구성원의 One body화를 통한 팀워크 제고, 프로젝트의 진전에 따른 유연한 조직 운용 등 관리적, 조직적 지원을 들 수 있다.<sup>8)</sup> 즉, Global view를 조망하고 프로젝트 전체를 아우르는 기능조직을 설치하여 운용한 점이다. 이 기능조직을 통하여 전체 모듈 간에 발생하는 정보 교환을 파악 및 조율하고, 쟁점 및 위험관리를 함으로써 프로젝트의 리스크를 크게 감소시킬 수 있었다.

### 7.2 향후 과제

미들웨어 및 개발프레임워크와 관련해서는, 고수준의 커스터마이징으로 인한 시스템의 지속적 업그레이드 보장 문제가 있다. 이러한 문제점을 어떻게 해결할 것인가에 대한 대책을 수립하고, 사용하고 있는 미들웨어의 벤더에 대한 안정성 확보 및 기술지원에 대한 확실한 체계를 정립하기 위하여 SK텔레콤은 개발에 참여한 해당 업체와 협의를 통한 조치를 취하고 있다.

본산 데이터베이스 아키텍처와 관련해서는, 시스템이 업그레이드됨에 따라 요구사항 및 트랜잭션의 변화에 맞추어 데이터의 변화, 데이터베이스 구조의 변화 등을 능동적으로 반영하고 해결할 수 있는 DB Migration 전략이 수립되어야 하기 때문에, Migration 담당 모듈에서 시스템 요구사항 변경에 능동적으로 대처할 수 있는 DB Migration 전략수립을 담당하고 있다.

시스템 성능 개선에 있어서 향후 발생할 수 있는 변화에도 성능 저하가 일어나지 않도록 하는 것이 매우 중요한 과제일 것이다. 현재의 고성능을 유지하기 위해서는 추후에 발생할 수 있는 DB 구조 변경과 액세스 패턴 변화에도 대처할 수 있도록 지속적으로 가이드라인을 수립하고 배포 및 교육을 실행하고 있다. 또한, 튜닝 작업 중에 발생한 일부분의 성능 저하를 지속적으로 관리하여 전체적인 성능의 균형을 맞춰나가고 이러한 여러 원칙에 대한 준수 여부를 지속적으로 관리 감독도 수행하고 있다.

NGM U.key 시스템은 마케팅 및 고객지향적인 시스템으로 출발하였다. 따라서 향후 선도적인 CRM의 적용, 보다 고객에게 다가가는 상품 개발 등 본래의 목적

8) Applegate 등(5)은 구조가 복잡하고 신기술을 적용하는 프로젝트는 리스크가 매우 높으므로 외적 통합(external integration)과 내적 통합(internal integration) 노력이 모두 필요함을 강조하였다. 외적 통합은 발주사에 의한 프로젝트 관리와 조정위원회 등의 장치를 포함하며, 내적 통합은 기술적 문제를 해결하기 위한 검토절차와 외부 기술지원 등을 포함한다(p.587, p.592).

을 달성하기 위하여 조만간 시스템 고도화 사업을 추가로 수행할 요구가 나타날 수도 있다.

### 참 고 문 헌

- [1] Oracle, Oracle Customer Case Study: KTF Offers Reliable, Real-time Customer Service, 2005.
- [2] Microsoft Corporation, E-Plus Mobilfunk GmbH, 1999.
- [3] IBM Corporation, DB2 Replication Flies Data to its Destination at Northwest Airlines, 1998.
- [4] SK텔레콤, NGM Storybook, SK텔레콤 NGM 본부, 2006.
- [5] Applegate, L.M., Austin, R.D. and McFarlan, F.W., Corporate Information Strategy and Management, McGraw-Hill, 2003.
- [6] Laudon, K.C. and Laudon, J.P., Management Information Systems: Managing the Digital Firm (10th Edition), Prentice Hall, 2006.
- [7] BEA, <http://www.bea.com>
- [8] TmaxSoft, <http://www.tmax.co.kr>
- [9] 이상구, 이익훈, 양정연, SK텔레콤 NGM의 데이터베이스 구성을 위한 연구보고서, 서울대학교, 2005.
- [10] Yang, J.Y., Lee, I.H., Jeong, O.R., Song, J.Y., Lee, C.M. and Lee, S.G., "An Architecture for Supporting Batch Query and Online Service in Very Large Database Systems," IEEE International Conference on e-Business Engineering, pp. 549-553, 2006.
- [11] Shasha, D. and Bonne, P., Database Tuning: Principles, Experiments, and Troubleshooting Techniques, Morgan Kaufmann, 2002.
- [12] Millsap, C., Optimizing Oracle Performance, O'Reilly Media, 2003.
- [13] Gurry, M. and Corrigan, P., Oracle Performance Tuning (2nd Edition), O'Reilly Media, 2005.
- [14] Floss, K., Oracle SQL Tuning and CBO Internals, Rampant Tech Press, 2004.



장 시 영

현재 성균관대학교 경영학부 교수로 재직 중이다. 서울대학교 산업공학과에서 학사, 대학원 경영학과에서 석사, 그리고 미국 Pittsburgh 대학교에서 MIS 전공으로 경영학 박사학위를 취득하였다. 주요 관심분야는 MIS의 계획 및 통제, 정보시스템 관리, 전자상거래 등이다.



양 정 연

2002년 충남대학교 컴퓨터공학과 학사  
2003년~현재 서울대학교 컴퓨터공학부 석박통합과정 재학중. e-Business 기술 연구센터 연구원. 관심분야는 e-Business Technology, Databases, Semantic Technology



이 상 구

1985년 서울대학교 계산 통계학과 학사  
1987년 M.S. Computer Science, Northwestern University, Evanston, Illinois.  
1990년 Ph.D. Computer Science, Northwestern University, Evanston, Illinois.  
~현재 서울대 컴퓨터공학부 교수, e-

Business 기술 연구 센터장. 관심분야는 e-Business Technology, Databases, Mobile Database