

SK텔레콤의 성공적인 다운사이징 사례: U.Key의 탄생

장시영^a, 이상구^b

^a성균관대학교 경영학부 교수
서울 종로구 명륜동3가 53성균관대학교 경영관 325호
Tel: +82-2-760-0458, Fax: +82-2-760-0440, E-mail: syjang@skku.ac.kr

^b서울대학교 컴퓨터공학부 교수
서울 관악구 관악로 599 컴퓨터공학부
Tel: +82-2-880-5357, Fax: +82-2-886-7589, E-mail :sglee@snu.ac.kr

Abstract

SK텔레콤은 급변하는 이동통신시장 환경의 변화에 적절히 대응하기 위하여 지난 10여 년간 사용해 온 메인프레임 기반의 COIS 시스템을 중단하고 Unix 기반의 NGM 시스템으로 다운사이징하기로 결정하였다.

2002년 9월 SK텔레콤은 차세대 IT 인프라 혁신 전략 프로젝트를 완료하고 이에 근거하여 2003년 말 1단계 프로젝트를 추진하게 된다. 그러나 개발 톨의 문제점, Governance의 확보 실패 등 문제점이 계속 누적되어 2005년에 접어들어서는 총체적 난국 상황에 봉착하게 되었다. 2005년 2월 자체 점검 결과 개발 진척도는 50%에 불과한 상황이었다. 결국 2005년 3월, NGM 추진본부는 프로젝트의 추진 중단을 선언하기에 이르렀다. NGM 프로젝트는 처음부터 재 검토하여 Re-planning을 실시한 후 구축을 재개하기로 결정하였다.

2005년 5월 NGM 프로젝트의 Re-plan이 수립되었다. 우선 Turn-key 방식의 문제점을 해소하기 위하여 SK텔레콤이 구축의 총책임을 떠맡고, SK C&C가 시스템 이행에 대한 책임을 지는 Governance 체제를 확립하였다. 또한 2단계 프로젝트에서는 NGM 프로젝트를 기술적으로 원점부터 재 검토하여 이전과는 근본적으로 다른 In-House 구축 접근방식을 채택하기로 하였다.

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. 이 논문에서는 이들 각각의 분야를 구체적으로 검토하였다.

결국 2006년 10월 9일 메인프레임 기반의 COIS는 Shut-down되어 가동을 멈추었고, Unix 기반의 NGM 시스템이 가동을 개시함으로써 차세대 마케팅을 본격 추진하기 위한 다운사이징 프로젝트가 성공적으로 완수되었다.

Keywords: 개발프레임워크, 데이터베이스 아키텍처

I. 서론

SK텔레콤은 고객지원과 요금청구 등 업무 처리를 위하여 지난 10여 년간 COIS 시스템을 활용해 왔다. 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 2006년 10월 Unix 기반의 U.key 시스템을 성공적으로 가동하게 되었다.

NGM 프로젝트는 국내 최대 규모의 시스템 통합 프로젝트일 뿐만 아니라, 세계적으로도 보기 드문 초대형, 초고난이도의 프로젝트이다. 이 프로젝트의 수행을 위하여 3000억원을 초과하는 예산이 투입되었고, 10000MM을 초과하는 개발인력이 투입되었다. 개발의 피크타임 시에는 월 1000명이 넘는 개발자가 동시에 개발작업을 수행하기도 하였다.

NGM 프로젝트는 종전의 일반적인 시스템 통합 프로젝트 진행 방식과는 전혀 다른 새로운 방식으로 접근하여 성공한 독특한 프로젝트이다. 1990년대 중반 ERP가 국내에 처음 소개된 이후 국내에서 수행된 대부분의 대형 프로젝트는 세계 시장에서 검증된 솔루션을 바탕으로, 글로벌 컨설팅 그룹이, Turn-key 계약방식으로 진행해 왔고 NGM 프로젝트도 처음에는 이 방식을 적용하였다. 그러나 Re-plan 이후 NGM 프로젝트는 국내 솔루션을 바탕으로, SK텔레콤의 주도로, 영역별 분리계약 방식으로, 일체형 (one body) 조직 구성과 운영을 통하여 성공적으로 수행되었다(김영곤, 김창완, 2007).

이 논문의 목적은 NGM 프로젝트의 태동으로부터 성공적 오픈에 이르기까지의 과정 요소와 성과를 주로 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.

II. 사례연구의 배경

SK 텔레콤은 메인프레임에 기반한 기존의 노후화되고 경직된 마케팅 인프라로는 급변하는 이동통신시장 환경의 변화에 적절히 대응하기 어렵다는 판단에서 Unix 기반 시스템으로의 다운사이징 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.

이러한 IT 인프라 혁신의 필요성을 구체화하기 위하여 SK 텔레콤은 컨설팅 업체와 함께 두 차례의 전략 프로젝트를 수행하고 이를 바탕으로 NGM 프로젝트를 태동시키게 된다. NGM 프로젝트가 형성된 과정은 다음과 같다.

먼저 2002년 9월 차세대 IT 인프라 혁신 전략 프로젝트를 완료하였다. 이 프로젝트를 통하여 차세대 IT 플랫폼에서는 기존의 메인프레임을 폐기하고 Unix 시스템을 채택하기로 결정하였다. Unix 시스템은 보안성과 안정성 측면에서는 메인프레임보다 취약하지만 유연성, 확장성, 가용성 면에서 강점이 있으므로 다운사이징을 추진하기로 하였다. 아울러 시스템 구축 방식은 패키지 솔루션 도입 후 환경에 맞게 커스터마이징하는 'Buy and Build' 방식이 적합할 것으로 판단하였다.

2002년 12월 두 번째 전략 프로젝트인 CRM 전략 프로젝트를 완수하였다. 이 프로젝트를 통하여 고객지향적 상품과 서비스의 개발, 비즈니스 파트너와의 관계 강화, 차세대 마케팅의 역량 제고 필요성 등 CRM 관련 과제의 중요성을 인식하게 되었고, 보다 구체적으로는 8대 핵심 영역에 대한 프로세스와 구현방안을 수립하였다.

2003년 7월에는 NGM 시스템의 아키텍처를 설계하였다. 사용자 인터페이스를 통합하여 단일의 GUI를 제공하고, 비즈니스 로직과 데이터를 별도 관리하여 개발과 유지보수가 용이하도록 하였다. 또한, 각종 고객정보를 통합 관리하여 전사적으로 통일된 customer view를 제공할 수 있도록 설계하였다. 또한 IT 인프라 측면에서는 시스템의 가용성을 확보하기 위하여 다중서버의 클러스터링

구조를 기본으로 하도록 설계하였다(SK 텔레콤, 2006).

한 달 후인 2003년 8월에는 개발 솔루션과 개발 방식을 결정하였다. 우선 CRM을 중심으로 A사의 비즈니스 패키지를 적용하여 구현하고, 청구업무 등은 이를 기반으로 하여 개발하기로 하였다(손정희 등, 2004).

2003년 11월에는 NGM 시스템 구축 주관사를 선정하고, NGM 구축의 전 과정을 구축 주관사에 일임하는 Turn-key 방식의 계약을 체결하였다. 2003년 12월, NGM 추진본부는 NGM 시스템의 구축계획 수립을 완료하고 프로젝트 개시에 돌입하였다. 이로써 예산액 2500억원, 개발인력 9800MM, 개발기간 22개월에 달하는 초대형 IT 인프라 구축 프로젝트를 big bang 방식으로 구축하는 사업에 본격 착수하게 된 것이다.

III. 1 단계 프로젝트의 추진과 중단

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 톨의 문제가 터지기 시작하였다. 2004년 3월~11월 간 9개월에 걸친 기능 설계에서 A사의 개발프레임워크는 관련 기술인력이 부족하였고, 기술지원도 원활하지 않은 문제점을 드러냈다. 또한 개발자가 사용하기에도 불편하여 결국 개발 톨을 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. 2005년 2월 자체 점검 결과 NGM 프로젝트는 적색 경고등이 켜진 것으로 판단되었다. 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 재 검토하여 Re-planning 을 실시한 후 구축을 재개하기로 결정하였다. 이에 따라 아키텍처의 재 설계, 추가 일정 및 추가 예산이 배정되었으며, 2005 년 5 월까지 Re-plan 수립, 9 월까지 아키텍처와 기능 설계, 2006 년 7 월 시스템 가동을 결정함으로써 10 개월 간의 시스템 오픈 순연이 발생하게 되었다. 정보시스템 프로젝트의 위험요인 중 대표적인 것으로 볼 수 있는 '일정 및 자원' 상의 위험이 실제로 나타난 것이다(조숙진 등, 2006; Barki, et al., 2001; Schmidt, et al., 2001).

NGM 프로젝트는 세계적으로도 유례를 찾아보기 어려울 만큼 규모가 큰 프로젝트이다. 대규모의 데이터와 트랜잭션을 포함하고 있으며, 업무 프로세스를 룰에 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 view 를 도출해내기 어려운 단점이 생긴다. 1 단계의 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 A 사의 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.

IV. 2 단계 프로젝트의 재 추진과 성공적 수행

2005 년 5 월 NGM 프로젝트의 Re-plan 이 수립되었다. 프로젝트 관리 관점에서 보면 NGM 프로젝트는 규모, 구조, 기술 등 세 가지 측면의 리스크를 모두 가지고 있는 프로젝트이다 (Applegate 등, 2003; Laudon and Laudon, 2006). 이러한 리스크에 대한 고려가 2 단계 프로젝트의 Re-plan 에 그대로 반영되었다. 우선 Turn-key 방식의 문제점을 해소하기 위하여 SK 텔레콤이

구축의 총책임을 맡고, SK C&C 가 시스템 이행에 대한 책임을 지는 Governance 체제를 확립하였다.

우리나라의 이동통신 시장은 세계에서 가장 치열하고 역동적인 경쟁장이다. 현재 세계에서 NGM 에서 요구하는 수준의 기능들을 수행하는 회사는 찾을 수 없으며 이를 지원할 수 있는 패키지 솔루션은 더더욱 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 부문별 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.

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

기본적으로 NGM 프로젝트는 메인프레임 기반의 시스템을 Unix 로 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 이슈를 도출하여 이를 해결하기 위해 노력하였다.

따라서 2 단계 프로젝트에서 관심을 집중한 기술적 의사결정 영역은 크게 ① 미들웨어 및 개발프레임워크의 적용, ② DB 아키텍처의 결정 및 슬립화, ③ 시스템 성능 개선 등의 세 분야로 집약할 수 있다. 이들 각각의 분야를 구체적으로 살펴보면 다음과 같다.

4.1 미들웨어 및 개발프레임워크의 적용

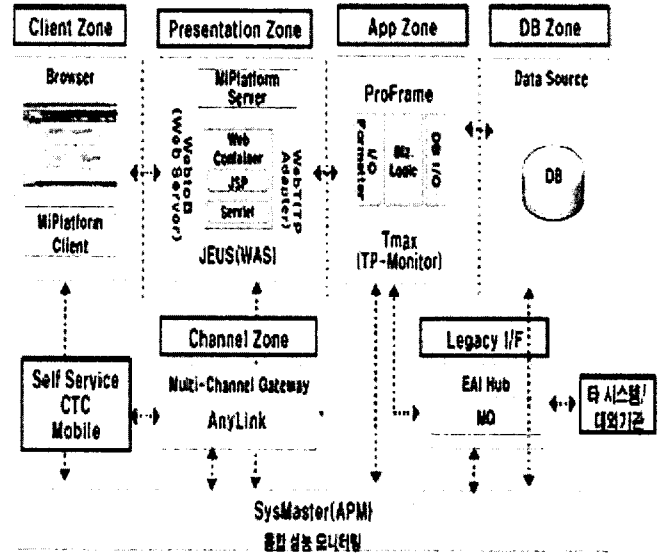
NGM 프로젝트의 중요한 화두 중 하나는 '오픈 아키텍처' 이다. 이것은 Unix 로 다운사이징

하면서 표준을 따르는 여러 미들웨어를 도입하는 것을 의미한다. 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 여러 가지 장점을 가질 수 있다.

TPM 과 WAS 미들웨어의 선정은 시스템 안정성과 성능에 큰 영향을 미치게 된다. 일반적으로 TPM 과 WAS 를 선정할 때에는 성능 최적화와 미들웨어 간의 연동성을 위해 하나의 벤더를 선택하는 경우가 많다. 이에 대한 후보로 NGM 프로젝트에서는 TmaxSoft 사의 'Tmax/JEUS' 조합과 BEA 사의 'Tuxedo/WebLogic' 조합을 고려하였다.

<표 1>에서 보는 바와 같이 후보 제품들은 모두 표준에 따라 개발되었기 때문에 어느 후보를 선택하더라도 시스템을 구성하는 것은 가능하다. NGM 과 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 TmaxSoft 사의 제품을 도입하기로 결정하였다. 결과적으로, 수많은 튜닝과 커스터마이징을 통해 최적화된 미들웨어 성능을 얻을 수 있었으며 NGM 의 성공에 중요한 요인이 되었다.

NGM 프로젝트에서 미들웨어 및 개발프레임워크의 선정은 엄격한 검증 절차를 거쳐서 결정되었으며, <그림 1>은 이러한 과정을 거쳐서 최종 선정된 미들웨어 및 개발프레임워크를 나타낸 것이다.



<그림 1> 전체 시스템 아키텍처

특히 T 사의 ProFrame 의 경우 DB 접근과 애플리케이션의 공통 로직을 담당하는 개발프레임워크로서 C 언어 기반으로 커스터마이징 되었다. ProFrame 은 핵심적인 공통 처리 모듈을 포함하여 전체 시스템의 성능 향상 및 개발 생산성 향상에 많은 역할을 담당하였다.

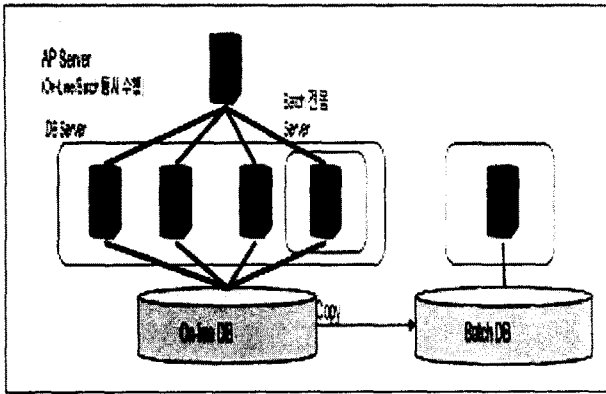
<표 1> TPM 과 WAS 후보 비교

제품	Tmax/JEUS	Tuxedo/WebLogic
J2EE 표준 인증	○	○
OSI, X/Open DTP 표준	○	○
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

4.2 DB 아키텍처의 결정

중앙 집중 방식의 single DB 는 확장성이 부족하고 Online 업무와 Batch 업무 사이의 간섭으로 인하여 성능 문제가 발생할 가능성이 존재한다. 한편 분산 DB 를 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 2>는 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.



<그림 2> NGM 프로젝트를 위한 Hybrid Model

SKT의 경우 대용량의 통화기록 정보 및 서비스 내역들을 기초로 하여 각종 매출 추정 및 검증작업이 수행되어야 하는데, 이러한 대용량 Batch 작업을 수행하는 경우 Online 트랜잭션의 간섭문제와 더불어 데이터 및 처리 결과의 copy 문제 등이 가장 큰 문제이다. 또한, Batch 작업을 수행하는 어플리케이션이 다양한 요구조건에 따라 자주 변경되는 특성을 가지고 있으므로, Batch 작업에 대한 노드가 상당히 큰 문제점이 있다. 따라서 하이브리드 DB 구성 모델에서는 별도의 Batch 작업 전용 서버와 DB를 구성하고, 매출 추정 및 검증 등의 어플리케이션을 실행 및 보정하는 작업을 처리한다. 모든 Batch 작업에 대한 작업이 완료되면, 완성된 버전의 어플리케이션을 Online 서버군쪽의 Batch 작업 담당 서버로 copy 하고, 최종 Batch 작업을 단시간에 처리한다. 이러한 방식으로 Online 트랜잭션과의 간섭으로 인한 성능저하를 최소화 하고 Batch 작업의 확장성을 보장하는 한편, 기존 시스템의 가장 큰 문제점이었던 시간적 제약을 크게 완화시킬 수 있었다.

특히 이 경우 Online 업무를 처리하는 DB로부터 일부 데이터를 Batch 업무를 처리하는 DB로 copy 하는 작업이 시스템 성능에 중요한 영향을 미치게 되는데, 이 때 사용하는 기술로서 크게 Disk Copy 방식과 log 기반 방식이 존재한다. SKT의 경우 Disk Copy 방식을 기존부터 사용하고 있던 상황이었다. 기존 방식의 변경 가능성을 타진해보기 위해 Redo log 기반 기술인 Oracle Streams를 고려하였다. 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 copy 되어야 하는 모든 데이터에 대한 예측이 아닌 실질적 검증이 필요하다는 문제점이 발생하였다. 따라서, SKT는 기존의 Disk Copy 방식을 그대로 유지하기로 하였고 copy 효율성을 높이기 위하여 스토리지 구성을 달리하는 방식을 취하였다.

4.3 시스템 성능 개선

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 묶어내어 최적화 하는 과정을 거쳤다. 다양한 검색 조건을 Grouping 한 후 이를 최적화하였고, 이를 최종적으로 Merge 하여 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.

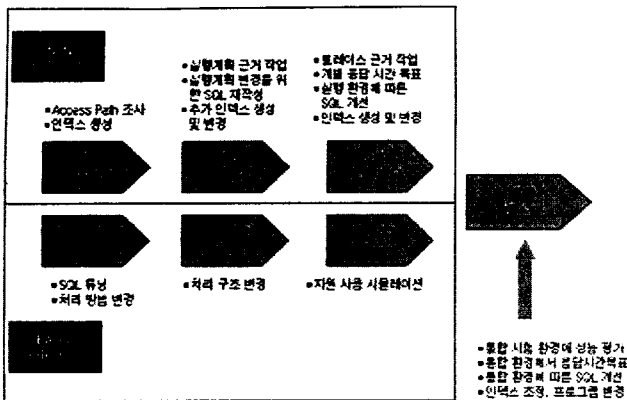
마지막으로 응답 시간 개선을 위한 성능 개선 작업에서는 작업 특성 별로 전략을 수립하고 이후에 통합 테스트를 수행하였다. Online 작업은 SQL 튜닝 등의 전략적 측면에 초점을 맞추고, Batch 작업은 작업 프로세스 전반에 걸친 최적화에 초점을 맞추었다. 이후 통합 테스트를 통해 전체 작업의 응답 시간 개선을 모색하였다.

NGM 프로젝트의 성능 개선 절차는 <그림 3>에서 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(Millsap, 2003).

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.



<그림 3> 성능 개선 절차

V. 결론 및 논의

2006년 10월 9일 메인프레임 기반의 COIS는 Shut-down 되어 가동을 멈추었고, Unix 기반의 NGM 시스템이 가동을 개시함으로써 차세대 마케팅을 본격 추진하기 위한 다운사이징 프로젝트는 성공적으로 완수되었다. 이 프로젝트를 성공적으로 완수함으로써 SK 텔레콤은 통신업계 다운사이징의 선진 사례를 남길 수 있게 되었고, 향후 오픈 시스템을 활용한 혁신을 기대할 수 있게 되었다.

NGM 프로젝트의 성공요인은 다음과 같이 요약할 수 있다. 첫째, 관리적, 조직적 요인으로서, 기본적으로 발주사의 Governance 확립, 참여 구성원의 One body 화를 통한 팀워크 제고, 프로젝트의 진전에 따른 유연한 조직 운용 등 관리적, 조직적 지원을 들 수 있다(김준우 등, 2005; 이성호, 안중호, 2005). 즉, Global view를 조망하고 프로젝트 전체를 아우르는 기능조직을 설치하여 운용한 점이다. 비록 철저한 검증에 의하여 미들웨어 및 개발프레임워크를 선정하고 도입하였으나 각 모듈 간에 발생하는 정보 교환을 파악하지 못한다면 전체 시스템을 구성하는 것은 어렵다. 결국 전체 시스템은 Global view를 고려해야만 구성 가능한 것이므로 NGM 프로젝트에서는 Global view를 유지하기 위하여 DAB(Design Advisory Board)를 구성하였다. DAB는 각 모듈을 담당하는 담당자의 모임으로서 각 모듈의 공통 사용기능을 도출하고 이를 개발프레임워크에 구현하는 역할을 담당하였다. 주 2회씩 6개월 간 계속된 협의를 통하여 Global view를 중단없이 고려할 수 있었으며 이를 통하여 구현된 개발프레임워크는 프로젝트 성공에 큰 밑거름이 되었다.

둘째, 쟁점 및 위험관리의 고도화를 들 수 있다. 쟁점관리와 위험관리는 통상적인 프로젝트 관리

활동의 일부이지만, NGM 프로젝트에서는 발생 가능한 쟁점과 위험을 식별하고 관리하기 위하여 별도의 IRM(Issue and Risk Management) 조직을 설치하고 전담자를 두어 가능한 모든 쟁점과 위험을 본격 관리하였다. 쟁점이나 위험에게는 고유 식별번호를 부여하고 이들이 소멸될 때까지 추적 관리함으로써 프로젝트의 리스크를 크게 감소시킬 수 있었다.

셋째, 원활한 기술지원 체제를 확보하여 벤더로부터 글로벌한 수준의 지원을 획득하였다. 2 단계 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.

넷째, 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 슬림화와 세밀한 성능튜닝을 통하여 시스템 성능을 확보할 수 있었다.

참고문헌

- SK 텔레콤, NGM Storybook, SK 텔레콤 NGM 본부, 2006.
- 김영곤, 김창완, “SK 텔레콤의 NGM 시스템 구축,” Working Paper, 2007.4.
- 김준우, 김승기, 전동진, “군 정보시스템 구현의 중요성공요인 분석 연구,” 경영정보학연구, 제 15 권 제 1 호, 2005년 3월, pp. 81-113.
- 손정희, 장윤희, 이재범, “정보시스템 구축환경에 따른 ERP 관리와 성과에 관한 사례 연구,” Information Systems Review, 제 6 권 제 1 호, 2004년 6월, pp. 1-18.
- 이상구, 이익훈, 양정연, SK 텔레콤 NGM의 데이터베이스 구성을 위한 연구보고서, 서울대학교, 2005.
- 이성호, 안중호, “고속철도에 있어서의 정보시스템 구현 사례에 대한 연구,” Information Systems Review, 제 7 권 제 2 호, 2005년 12월, pp. 1-21.

조속진, 이석준, 함유근, “ 정보시스템 프로젝트의 위험요인에 관한 실증 연구,” 경영정보학연구, 제 16 권 제 3 호, 2006 년 9 월, pp. 143-158.

Applegate, L.M., R.D. Austin and F.W. McFarlan, *Corporate Information Strategy and Management*, McGraw-Hill, 2003.

Barki, H, S. Rivard, and J. Talbot, "An Integrative Contingency Model of Software Project Risk Management," *Journal of Management Information Systems*, Vol. 17, No. 4, 2001, pp. 37-69.

Floss, K., *Oracle SQL Tuning and CBO Internals*, Rampant Tech Press, 2004.

Gurry, M. and P. Corrigan, *Oracle Performance Tuning (2nd Edition)*, O'Reilly Media, 2005.

Laudon, K.C. and J.P. Laudon, *Management Information Systems: Managing the Digital Firm (10th Edition)*, Prentice Hall, 2006.

Millsap, C., *Optimizing Oracle Performance*, O'Reilly Media, 2003.

Schmidt, R., K. Lytinen, M. Keil, and P. Cule, "Identifying Software Project Risks: An International Delphi Study," *Journal of Management Information Systems*, Vol. 17, No. 4, 2001, pp. 5-36.

Shasha, D. and P. Bonne, *Database Tuning: Principles, Experiments, and Troubleshooting Techniques*, Morgan Kaufmann, 2002.

Yang, J. Y., I. H Lee, O. R. Jeong, J. Y. Song, C. M Lee, and S. G. Lee, "An Architecture for Supporting Batch Query and Online Service in Very Large Database Systems," IEEE International Conference on e-Business Engineering, 2006, pp. 549-553

BEA, <http://www.bea.com>

TmaxSoft, <http://www.tmax.co.kr>