

사례  
발표

# IBM JT/400을 이용한 미들웨어의 구축

이 병 옥\*, 이 승 호\*\*

● 목 차 ●

1. 서 론
2. IBM JT/400을 이용한 분산 시스템의 설계
3. IBM JT/400을 이용한 분산 시스템의 구현
4. 결 론

## 1. 서 론

소프트웨어의 기술 향상에 가장 큰 요인이 되는 기틀로 하드웨어의 발전을 들 수 있을 것이다. 물론 하드웨어의 발전 자체가 사용자의 편리함에 기인된 보다 큰 요구에 의해서 이루어 지고, 이것이 곧 소프트웨어의 대한 요구라고 해석될 수도 있지만, 소프트웨어 측면만을 본다면 하드웨어는 소프트웨어의 영원한 선발 주자인 셈인 것이다. 불과 수년 전의 386, 486 PC(Personal Computer)에서 구동되던 프로그램들과 오늘날 우리가 사용하고 있는 PC에서 구동 되어지고 있는 프로그램들만 보더라도 확연히 비교가 됨을 우리는 알 수 있다.

이러한 추세와 더불어 최근 수년사이에 소프트웨어의 기술은 물리적 네트워크를 통해 확산되고 있는 인터넷과 함께 가히 혁명이라 할 만큼의 변화를 보이고 있다. 1가구 1PC의 시대에 만족하지 못한 사용자는 원격지의 제2, 제3의 그들의 PC를 요구하고 있는 것이다. 이러한 요구는 분산객체라는 기술적 이해로 실현 가능하게 되었다.

그렇다면 분산객체 기술이란 무엇인가? 컴퓨터 사용자들의 요구는 개인적으로 생산성 있는 응용 프로그램뿐만 아니라 파일이나 데이터베이스처럼 다양한 데이터 저장공간으로부터 정보를 얻고자 하며 게다가 다른 사용자와 자료를 공유하고 서로 통신 하기를 원하고 있다. 이렇듯 다양한 요구 사항을 만족시키는 프로그램을 작성하려면 서로 다른 운영체제와 네트워크 환경 등 이종의 환경에서도 작동하는 Client/Server 소프트웨어가 필요로 하며 이런 목적으로 개발된 소프트웨어는 분산환경에서 서로 다른 시스템과도 쉽게 통합이 가능해야 한다는 데서 제기된 기술적 요구가 탄생 배경이다. 이렇게 다양한 사용자들의 요구를 만족시키기 위해 그리고 이종의 분산환경에서 여러 종류의 응용 프로그램을 통합하기 위해서는 일정한 결합 방식이 필요한데 이런 기술을 분산객체 기술이라 할 수 있다[1].

분산객체 기술들 중에서 최근에 가장 주목 받고 있는 기술이 객체의 생성, 소멸, 저장, Transaction 기능 등을 모두 포함한 Sun Microsystems의 Enterprise JavaBean(이하 EJB) 과 마이크로소프트사의 Distributed Component Object Model(이하 DCOM) 일 것이다. 이 두 기술은 다수의 공통된 점

\* (주)이지시스템 eBusiness 사업부 대리

\*\* (주)이지시스템 대표이사

을 가지면서도 Java의 플랫폼 독립성(Platform Independency)이라는 가장 큰 장점을 포함한 EJB에 반해 윈도우 환경에 의존하는 DCOM의 가장 결정적인 차이를 나타낸다.

이러한 의미에서 IBM은 자사 모델인 AS/400의 시스템 내부 Resource의 외부화를 Java를 통해 실현하였다. IBM의 Java를 이용한 정책은 EJB 환경의 Application Server로 이미 우리에게 잘 알려진 WebSphere를 통해서도 확인할 수 있다. 더불어 IBM Enterprise Toolkit for AS400(이하 JT/400)은 AS400JDBCdriver를 통한 AS/400 내의 DB2 Database 자료의 Query는 물론 AS400 객체를 이용한 Server Program의 Disputed Program Call(원격지 프로그램 호출 - 이하 DPC)과 Database Table의 Record 자료를 access할 수 있는 Record Level Access, Server Command 실행 등 AS/400 내에서 실행 가능한 대부분의 기능을 Java를 통해 외부화 하고있는 편리함을 제공하고 있다.

그러나 JT/400의 이러한 기능들은 AS400JDBCdriver를 이용한 Query를 제외한 AS400 객체의 편리한 기능들을 WebSphere를 통해 충분히 지원하지 못하고 있다. 즉, 사용자는 AS400 객체의 기능을 이용하기 위해서 2-Tier 방식을 사용해야 하는 것이다. 이는 WebSphere를 이용한 EJB의 개발 또는 Web 환경 개발에서 객체의 생성, 소멸, 저장, Transaction의 자동화 라는 큰 이점에 반하는 결과를 초래할 수도 있는 것이다.

본 논고에서는 JT/400에서 지원하는 AS400 객체의 관리를 통해 IBM AS/400을 이용한 보다 효율적인 분산 시스템의 구현 방법을 소개한다.

## 2. IBM JT/400을 이용한 미들웨어의 설계

분산 시스템 설계의 기본 구조는 초기 2-Tier 방식에서 3-Tier 혹은 Multi-Tier로 변화 하면서

Business Logic의 수행의 주체가 어느쪽인지에 설계의 초점이 맞추어 지고있다. 최근의 추세로 볼 때 Client의 성능향상 속도는 Server(이하 Server는 분산 시스템의 Sever를 칭함)의 성능향상 속도를 앞지르고 있다고는 하지만, 우리가 원하는 보다 현실적인 네트워크 속도의 지원이 없는 한은 Server의 비중은 절대적일 수 밖에 없을 것이다.

### 2.1 3-Tier 방식의 기본 구조

시스템의 기본구조는 Server의 의존성을 높이는 3-Tier 구조로 이루어 진다. 이는 다양한 Client의 상황에 능동적으로 대처할 수 있을 뿐만 아니라 AS/400 시스템에 미치는 부하도 감소 시킬 수 있다.

### 2.2 AS/400과 JT/400의 특수성 이해

AS/400에서 사용하는 코드체계는 EBCDIC CODE이며, Java에서 사용되는 코드는 UNICODE이다. 두 시스템의 통신에 있어 코드변환은 JT/400의 AS400Text 클래스에서 지원되고 있으며, 사용자는 AS/400과의 통신에 있어 AS/400에서 사용되고 있는 CCSID를 확인해 둘 필요가 있다.

JT/400 분산 시스템 구현의 주체가 될 IBM AS/400 기종은 Data의 처리 및 저장에 있어 2byte 문자를 처리하기 위한 특수 Character(OE/OF)를 사용하고 있다. 이는 기존의 Oracle 또는 Sybase 등의 RDB에 익숙한 사용자에게는 마치 산술연산의 Overflow와 흡사한 예기치 못한 결과를 초래할 가능성이 있어서 처리에 주의를 기울여야 한다. OE/OF는 2byte에서 1byte로의 전환에서 OE가 발생하며, 1byte에서 2byte로의 변화에 OF가 발생한다. 다음은 OE/OF 발생의 간단한 예이다.

<표 1> OE/OF의 처리 방식

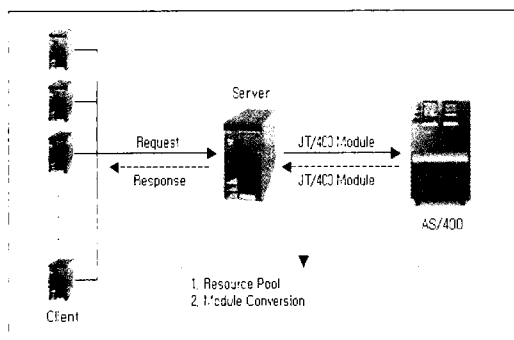
Java에서의 문자열 처리	나는 학교에 간다.	17byte
AS/400에서의 문자열 처리	OE나는OF 학교에OF OE간다OF.	23byte

<표 1> 에서 볼 수 있듯이 "나는 학교에 간다."라는 문장은 17byte의 길이를 가지는 문장이다. 그러나 이 자료가 AS/400에 저장되기 위해서는 23byte의 공간이 필요하다.

### 2.3 Resource Pool

일반적으로 Database의 Query 실행에서 빈번한 Connection의 연결은 처리속도 저하의 가장 큰 이유 중의 하나로 인식되어 Pool 기능의 사용은 이미 일반적인 방법론이 되었다. JT/400을 이용한 분산 시스템의 구현에 있어서도 AS400이라는 일종의 Connection 객체를 사용하게 된다. 이는 Socket 통신을 통하여 AS/400과 통신을 하게 되며, 하나의 AS400 객체는 AS/400 시스템에 하나의 Job을 생성시키게 된다. 즉, 이 현상은 필요 이상의 AS400 객체의 생성은 AS/400 시스템 성능에 영향을 미치게 된다는 설명이다. 또한 AS400 객체의 생성 과정은 AS/400 시스템의 SignOn 과정을 거치게 되면서 그 처리 속도에 영향을 받게 된다.

(그림 1) 은 JT/400을 이용한 분산 시스템의 전체적인 구조를 보여 주고 있다.



(그림1) 시스템 설계의 기본 구조

## 3. IBM JT/400을 이용한 미들웨어의 구현

JT/400을 이용한 분산 시스템을 구현함에 있어서

(그림 1) 에서 보여주듯 Client와 AS/400 사이의 Data 처리 방식을 결정 할 필요가 있다. 우선 Server를 기준으로 AS/400과의 통신은 JT/400을 이용하게 되며, Server와 Client 사이의 통신은 선택에 의하여 Java의 다양한 분산 객체 구조 구현 방법(RMI, CORBA, Socket, EJB 등...)을 이용할 수 있다.

### 3.1 실시간 처리 Server의 구축

분산 시스템의 Server는 특정 Client에 편중된 서비스를 제공해서도 안되며, 또한 이로 인한 대기현상이 발생해서도 안 된다. 이를 위해 Server는 Client의 요청이 받아들여진 이후의 동작은 각각의 Thread를 생성하여 요청을 처리하여야 한다. 물론 이 과정에서의 Resource의 획득과 반환 등의 일련의 작업들은 객체의 직렬화를 통하여 구현하여야 한다.

### 3.2 AS400 객체 Processing

AS400 객체의 Processing은 기본적으로 관리되어야 하는 최대, 최소 개수 설정과 객체의 자동 소멸 시기 이외에 객체의 재사용에 관한 관리가 이루어져야 한다. JT/400은 버전별로 다소 차이가 있으나 V5R1 이전의 버전에서는 AS400 객체의 사용 회수가 30608 번으로 제한되는 현상이 발생한다.

### 3.3 발생 가능한 예외의 처리

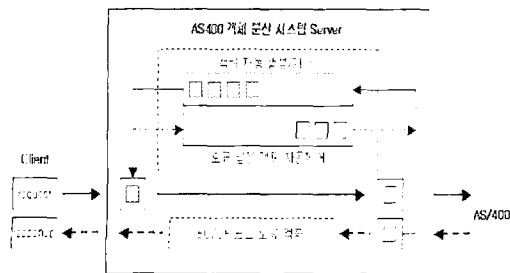
JT/400을 이용한 분산 시스템 개발에는 JT/400 API 자체에서 정의하는 다양한 예외 현상이 발생 가능하다. 그러나 이를 이용한 분산 시스템의 개발에서는 정의된 예외 이외의 별도의 예외가 존재한다. 대표적인 예로 AS400 객체를 이용한 AS/400 Program Call 시에 Client의 정상적인 Call 이후 AS/400 Program의 오류가 발생하였을 경우 Client는 AS/400으로부터 Response가 올 때까지 영원히 대기하는 현상이 발생한다. 이는 Server의 Resource 관리 정책에 오류간주시간을 설정하여 처리되어져

야만 한다. 또 다른 예로 Server와 AS/400간의 물리적인 네트워크 이상현상으로 Pool에서 관리되어지던 AS400 객체가 사용 불가능하게 될 경우 이를 제거하고 재생성해야 한다.

### 3.4 발생 가능한 오류의 처리

JT/400을 이용한 자료의 처리과정에는 Code Conversion 시에 발생하는 RunTime 오류가 발생할 수 있다. Server가 AS/400으로부터 전송 받은 자료는 EBCDIC CODE에서 UNICODE로의 변환과정에서 Code Table에 존재하지 않는 code를 내포할 가능성이 있으며, 또한 OE/OF의 쌍이 일치하지 않는 경우가 발생할 수 있다. 이는 Server가 AS/400으로부터 전송 받은 최초의 자료인 Byte 형태의 Stream에서 OE/OF를 검색해 Code Conversion 이전에 복구 가능하다.

(그림 2)에서 AS400 객체지향 미들웨어 구조도를 보여준다. Client의 request이 들어오게 되면 미들웨어는 즉시 사용 가능한 AS400 객체를 할당해 준다. 할당되는 AS400 객체는 Pool을 통하여 생성, 공급, 회수, 제거의 Normal한 Life Cycle을 가지게 되며, 더불어 오류 발생 간주 및 검증의 위하여 사용 중인 Pool의 AS400 객체에 대한 정보가 별도로 Monitoring 된다. 시스템에서 AS400 객체는 AS/400과의 연결 시점에 한해서만 Client에게 공급이 되며 이후는 즉각 회수되어 재사용에 대비한다. AS/400으로부터 확인된 결과는 UNICODE로 Conversion되



(그림 2) AS400 객체지향 미들웨어 구조도

기 이전인 16진수 코드값을 분석해 코드를 검증 후 Client에 전송되어 진다. 시스템은 일련의 과정에서 발생하는 예외에 대해서는 즉시 Flow를 멈추고 결과를 Client에 전송한다.

다음 (그림 3)은 JT/400을 이용한 분산 시스템의 결과를 그래프로 표시한 것이다. 그래프는 모 인터넷 제품 판매 회사에서 사용된 결과이며, 5분 단위로 Access되는 Client의 request(그래프 하단)와 미들웨어가 AS/400과 유지하고 있는 AS400 객체의 현황을 보여준다. 그래프에서 확인할 수 있듯이 Client의 요청에 반해 Server에서 관리되어 생성되는 AS400 객체는 확연히 비교되는 수치를 나타내었다. 이는 곧 AS/400의 부하를 줄이고 아울러 즉시 사용 가능한 AS400 객체를 Client에 할당해 줌으로써 처리속도의 향상을 가져옴을 확인할 수 있었다.

(그림 3) Client 호출에 따른 AS400 객체 상태 비교

## 4. 결론

본 보고서는 IBM JT/400을 이용한 AS/400 System의 분산 시스템의 구축에 대해서 살펴 보았다. 인터넷과 함께 발전하고 있는 Client/Server 환경의 구현에서 현재 상태로 Application Server에서 지원하고 있지 않는 AS400 객체의 활용을 통해 얻을 수 있는 이점과 이를 위한 고려 사항 및 해결점들을 강조하였다. Java와 JT/400을 이용한 분산 시스템의 구축은 분산 시스템 구축으로 얻어지는 다

양한 접근성과 함께 AS/400을 이용해 오던 많은 CS Application User에게 보다 적은 노력으로 보다 큰 안정성과 친근감으로 외부화를 유도 해 줄 수 있을 것으로 본다.

### 참고문헌

- [1] <http://sh.hanarotel.co.kr/~ggola/telecom/분산객체개요.htm>
- [2] 실시간 JAVA 1998, 7 정보처리학회지
- [3] AS/400 강좌, Dr Frank G.Soltis 저, 정혜성 역
- [4] Enterprise JAVABEANS 김동균, 우미영, 강승우, 한빛미디어
- [5] IBM WebSphere Solution Guide 조광진, 이-인티그레이션
- [6] Enterprise Toolkit for AS/400, IBM
- [7] Java for AS/400, SAM Seminar, 2000.6 한국 IBM

### 저자약력



**이 병 옥**

2000년 동국대학교 전자계산학과 (학사)  
1999년-현재 ㈜이지시스템 eBusiness사업부 대리  
관심분야: 프로그래밍 언어, Java 분산객체



**이 승 호**

1984년 한국전자통신연구원  
1990년 (주)에이아이소프트 차장  
1992년 (주)씨앤아이 개발담당 이사  
2001년 (주)이지시스템 대표이사  
(주)이지시스템 부설연구소 소장  
관심분야: EIP, e-Business, CRM, JAVA, Linux