

□ 사례 발표 □

리눅스 운영체제에서의 오라클 데이터베이스와 구축사례

장 세 국[†]

◆ 목 차 ◆

- | | |
|------------------------------|---------|
| 1 서 론 | 3 적용 사례 |
| 2 리눅스에서의 오라클 지원 및 오라클 8i의 특징 | 4 결 론 |

1. 서 론

1991년 리눅스 토발드에 의해 개발이 시작된 Linux O/S는 그동안 국내에서 각종 동호회나 서클등에서 국소적으로 한글작업등 커널작업이 진행되어 오다가 1998년부터 본격적으로 대외적으로 표출하기 시작하였다. 국내 한글화된 배포판이 상용으로 판매 되면서 일반 기업에서도 서서히 눈을 돌리기 시작 하였다.

리눅스 O/S가 확산되고 상용화되기 위해서는 여러가지 활용도나 전제조건등이 있을 수 있다.

그중 몇가지만 고려해보자면 첫번째로 리눅스 O/S 만의 기능으로 클러스터링 기능을 이용한 고 가용성(HA-High Availability) 및 과학계산용으로 서의 활용도이다.

두번째로는 인터넷 컴퓨팅환경이나 클라이언트-서버환경에서의 클라이언트 역할이라고 볼 수 있는데, 즉, 클라이언트 PC용으로 사용하기 위해서 필요한 다양한 OA기능 및 개발 툴과 유틸리티들이라고 볼 수 있다.

이 부분은 현재 가장 더디게 움직이고 있는 부분이라고 할 수 있다. Linux확산에 가장 중요한

요소로 다양한 Word-Processor나 Presentation Tool 및 스프레드 쉬트, 그리고 Game과 같이 클라이언트에서 가장 많이 사용되는 기능과 클라이언트 애플리케이션을 개발하기 위한 GUI 개발 도구들의 등장으로 개발을 위한 플랫폼으로서의 Linux O/S를 사용함으로써의 활용도 확산이 기대되고 있다.

세번째 활용도는 가장 쉽게 적용할 수 있는 인터넷 서버나 데이터베이스 서버로서 사용되는 기반 플랫폼으로 리눅스를 사용하는 경우이다. 이 두형태의 적용 형태의 특징은 사용자가 서버, 즉 Linux O/S를 알필요가 없다는 점이다. 아직까지 Linux O/S에 대한 인지도 및 사용사례가 적기 때문에 기존의 Unix와 실질적으로 별차이가 없는 O/S라고 하더라도, 엔터프라이즈급이 아닌 중소 규모의 인터넷 서버나 데이터베이스 서버로 사용하는 기업 및 단체에서는 그래도 시스템(특히 O/S)전문가가 요구되는 O/S이기 때문에 단지 서버로만 사용하므로써 관리의 부담을 덜 수 있기 때문에 쉽게 적용하고 있다.

인터넷 서버나 데이터베이스 서버로 사용할 경우 요구되는 데이터베이스의 기능을 살펴보면, 우선 데이터베이스 서버로 사용될 경우 필요로 하는 OLTP기능과 OLTP업무에 의해 축적된 데이터를 기업의 의사결정(Decision Making)을 위한 필

[†] 정희원 : 한국오라클(주) 부장

으로 하는 Data Warehousing(DW)구축에 필요한 기능이다.

두번째로, 인터넷 서버로 사용될 경우 요구되는 기능으로는, 최근에 인터넷 구축에 있어 가장 활용도가 Java를 이용한 JDBC, EJB, CORBA, JSP 등의 지원이 되어야 하며, 특히 다양한 MultiMedia 콘텐츠들에 대한 데이터베이스 차원에서의 지원 기능이다.

다음에 Oracle 8i(오라클 인터넷 데이터베이스)에서 위에서 언급되었던 사항중에서 인터넷을 위한 Java지원사항과 단지 DB Server로 사용할 경우 필요로 되는 OLTP/DW특징에 대해 중점적으로 기능을 살펴 보기로 하자.

2. 리눅스에서의 오라클 지원 및 오라클 8i의 특징

2.1 자바 애플리케이션 개발

Oracle8i 플랫폼상에서의 Java프로그래밍은 크게 JDBC, SQLJ, JSP(Java Stored Procedure), EJB, CORBA의 5가지 범주로 나눌 수 있다.

Oracle8i는 3종류의 JDBC드라이버를 제공한다. 첫째는 100% Pure Java 애플리케이션/ 애플릿을 작성하기 위해 JDBC Thin 드라이버이고, 두번째는 Oracle의 Native code library를 사용하는 JDBC OCI(Oracle Call Interface) 드라이버이다. 마지막으로 Oracle8i 서버내에서 실행되는 Java code가 DB에 액세스하기 위해 사용되는 JDBC KPRB 드라이버가 그것이다. Oracle8i가 제공하는 JDBC는 LOB, Object Type, REF Type등 많은 기능이 추가되었다. JServer의 등장으로 필요해진 JDBC KPRB 드라이버에 대해서는 아래 JSP예제에서 간단한 예를 볼 수 있을 것이다.

SQLJ는 일종의 Pre-processor로 C-언어로 데이터베이스를 접근하기 위해 Pro*C라고하는 Pre-processor를 사용했듯이, Java언어로 프로그래밍을 할 때 프로그램 내부에 SQL(Structured Query Language)를 직접 기술하고, 이를 Pre-processor를 이용하여 사전에 한번더 컴파일을 하여 직접적인 SQL문장을 JDBC형태의 문장으로 자동 변환 시켜주는 역할을 한다.

그러므로 개발자는

```
#sql | create table emp(empno number(5), ename, varchar2(10)) ;
```

와 같은 코드를 Java source에 Embed시키고, 이 source를 CreateTable.sqlj라고 하는 이름으로 저장한 후, Oracle8i에서 제공되는 SQLJ translator로 이를 Pre-compile한다. SQL translator는 embedded sql문을 pur JDBC call로 다시 source를 생성해 주며 CreateTable.java가 생성이 된다.

이렇게 생성된 source를 JAVAC를 이용하여 Compile하면 최종 CLASS가 생성이 된다.

SQLJ는 상대적으로 low level이면서 dynamic API인 JDBC에 비해 훨씬 사용하기 쉽고, 개발자들에게 코딩의 량을 현저하게 줄여줄수 있기 때문에 높은 개발 생산성과 유지보수에도 많은 도움을 준다. 단, SQLJ는 static SQL만을 사용할 수있다는 제한을 가지고 있으나, 이는 특별히 단점이라기 보다는 early type checking, iterator등 static SQL에서만 적용이 가능한 기능들을 제공하기 때문이라고 볼 수 있다.

그러므로 개발자는 개발하고자 하는 애플리케이션이 static SQL만을 사용하느냐, 아니면 dynamic SQL을 필요로 하는냐에 따라 적절히 SQLJ와 JDBC를 혼용하여 사용해 개발 생산성과 동시에 실행 성능까지 향상시킬 수 있을 것이다. 이러한 SQLJ는 Oracle뿐만아니라, Tandem, Sybase, Sun등이 협력하여 만든 표준으로 98년이래 ANSI표준으로 채택되어 있다.

- Java Stored Procedure(JSP).

그동안 관계형 데이터베이스의 특징으로 저장 프로시저(Stored Procedure)를 활용하여 Business Flow에 대한 Logic을 DB server측에 주로 PL/SQL 언어를 이용하여 구축되었으나, 인터넷 컴퓨팅 환경에서 Java의 역할이 커짐에 따라 Oracle8i에서는 DB 커널 내부에 Java VM(Virtual Machine)을 내장 시킴으로써 Java언어로 작성한 스토어드 프로시저가 가능하게 되었다. 실제로 기존의 PL/SQL로 작성해 왔던 거의 모든 로직을 Java로 작성하는 것이 가능하며, 또한 기존에 Java로 작성된 거의 모든 프로그램은(JDK 1.1.6호환) RDBMS내에서 소스의 수정없이 저장시킬 수 있으며 또한 실행시킬 수 있다. 간단한 예제와 개발 과정을 통하여 살펴 보도록 하자.

PL/SQL stored procedure와 마찬가지로 Java stored procedure에는 Stored procedures, Stored Functions, Triggers, Object Relation methods와 같은 4가지 유형이 있다. Java stored procedure는 다음과 같은 4가지 맥락에서 호출할 수 있다.

- Top level CALL syntax :
SQL3에서 사용하는 표준구문으로 CALL MYPROC(argument)와 같은 형태로 SQL에서 바로 호출할 수 있다.
- SQL DML statements :
select myfunc(ename, empno) from emp와 같은 형태.
- PL/SQL block, subprograms, packages :
JSP와 PL/SQL은 100% 호환 가능하다.
- Implicity by trigger :
PL/SQL 트리거와 마찬가지로 주로 data consistency를 위해 사용된다.

기존에 PL/SQL로 작성되어 왔던 어떤 로직도 Java로 작성해 사용하는 것이 가능하다. 또한, 예

제를 보면 알겠지만 Java stored procedure는 실제로 publish라는 과정을 통해 PL/SQL interface로 wrap된 상태에서 사용되므로 사용자 혹은 애플리케이션 개발자의 입장에서는 PL/SQL stored procedure와 100%동일하게 사용 할 수 있다. JSP를 개발하는 과정은 다음의 4단계로 나뉘어 진다. Java 프로그래밍의 입문단계에서 누구나 한번씩은 작성해 본 hello.java프로그램을 통해 JSP개발 과정을 살펴보자.

단계1: Java stored procedure 작성.

```

Hello.java
Public class java{
Public static void main(String args[]) {
    System.out.println(Hello world!);
}
}
    
```

기존에 Java로 작성된 프로그램은 한 줄의 수 정도 없이 Oracle8i RDBMS에 올릴수 있다. 단, public static으로 선언된 메소드만이 stored procedure로 publish될 수 있다. public이 아닌 메소드는 해당 class내의 다른 메소드에 의해 호출할 수는 있다. 또한 Oracle8i Java VM은 Java stored procedure 호출 시 class를 일일이 생성(construct)하지 않으므로 메소드는 반드시 static이어야 한다. 위의 프로그램은 Sun의 JDK 1.X를 사용하여 다음과 같이 컴파일하고 실행할 수 있다.

```

$ javac hello.java
$ java hello
Hello World!
    
```

단계2 : 작성한 프로그램을 Oracle8i로 로드

JSP는 Oracle과 IBM, Sybase, informix, Sun Microsystems등이 공동으로 제안한 ANSI/ISO표준의 일부이다. 여기에서 CREATE JAVA 등 Java를

SQL에 publish하기 위한 DDL extension과 SQL에서 Java를 호출하기 위해 DML extension등이 포함되어 있다. Java stored procedure는 Oracle8i내에서 스키마객체로 관리되므로 Java를 SQL에 publish하기 위해서는 먼저 1) BLOB이나 BFILE형태로 Java 소스나 클래스를 임시로 저장하고, 2) SQL DDL을 이용하여 이를 라이브러리 유닛으로 로드하는 과정을 거쳐야 한다.

Oracle8i가 제공하는 loadjava 유틸리티를 사용하여 이과정을 간단히 처리할 수 있다. 앞서 compile한 hello.class를 loadjava를 이용하여 Oracle8i에 로드하기 위해서는 다음과 같이 한다.

```
$ loadjava user scott/tiger resolve hello.class
initialization complete
loading : hello
creating : hello
resolver :
resolving : hello
```

JServer는 JDK1.1.6과 완전히 호환되는 Java 컴파일러를 DB내에 내장하고 있다. 따라서 다음과 같이 컴파일하지 않은 source를 로드할 경우 내장된 컴파일러가 이를 수행하게 된다.

```
$loadjava user scott/tiger resolve hello.java
```

클라이언트 혹은 원격 머신에서 작성한 Java객체를 로드하고자 할 경우 다음과 같이 connect string을 적어 주면 된다.

```
$ loadjava -user scott/tiger@dbhost:1521:ORCL
-thin -resolve hollo.java
```

위의 경우는 thin JDBC방식을 사용한 것으로, dbhost는 Java소스(혹은 클래스)를 로드할 DB가 설치되어 있는 머신의 호스트명(혹은 IP주소),

1521DMS DB listener port번호, ORCL은 Oracle SID이다. 클라이언트 혹은 원격머신에 Net8(SQL*NET)이 설치되어 있고 insnames.ora에 위의 connect string이 oradb라고 하는 별칭으로 등록시킬경우는 다음과 같이 별칭만 이용하면 된다.

```
$loadjava user scott/tiger@oradb oci8 resolve hello.java
```

이 경우 loadjava는 JDBC OCI드라이버를 사용하여 Java객체를 Oracle8i에 로드한다. 이외에도 여러가지 다양한 방법과 옵션이 있을 수 있으나 여기서는 이정도로 충분할 것이다. 이제 Java객체를 SQL에 publish할 준비가 되었다.

단계3 : Java프로그램을 SQL로 publish

Java메소드를 SQL에서 호출하기 위해서는 이 메소드를 publish해야 한다. 여기서 publish는 Java 메소도에 대한 SQL interface를 제공함을 의미한다. 실제로는 Java 메소드에 대한 일종의 Proxy역할을 하는 PL/SQL프로그램을 작성하게 되며, SQL은 이 Wrapper를 호출하여 Java 메소드를 호출하게 된다.

이 Wrapper는 call spec이라고도 불리는데, 이 단계에서 Java data타입과 SQL data타입간의 매핑을 지정해 주어야 하기 때문이다.

앞서 로드한 hello클래스의 main() 메소드를 hello라는 이름으로 publish해 보자.

SQL*plus나 svrmgrl등을 사용하여 Oracle8i에 접속하고 다음의 DDL을 실행한다.

```
SQL> CREATE OR REPLACE PROCEDURE hello
AS LANGUAGE JAVA
NAME hello main(java lang.String{});
/
Procedure created.
```

단계 4 : SQL에서 Java프로그램을 호출

이제 간단한 Java stored procedure가 완성되었으므로 SQL에서 이 메소드를 호출하는 일만 남았다.

먼저, System.out이 SQL*Plus의 스트리밍 버퍼, 즉, SQL 프롬프트상에서 출력하도록 Re-direction을 설정한 후, hello()를 call하도록 한다.

```
SQL> SET SERVEROUTPUT ON
SQL> call dbms_java.sct_output(2000);
SQL> call hello();
Hello!!
Call completed.
```

Call구문은 앞서 말한 바와 같이 SQL3스펙에 지정되어 있는 구문이다.

단계적으로 설명하다 보니, 거처은 과정에 비해 결과가 너무 간단해 실망할 수도 있겠지만 이는 어디까지나 이해를 위해 보여준 간단한 예일 뿐이다.

얼마든지 복잡한 로직, 즉, 엔터프라이즈 수준의 비즈니스 로직까지도 구현이 가능하다. 더욱이 Oracle8i가 제공하는 EJB/CORBA를 이용한다면 이들은 분산객체로도 별다른 어려움이 없이 구할 수가 있다.

- EJB/CORBA

EJB와 CORBA는 각각 Sun Microsystems와 OMG (Object Management Group)에서 내놓은 컴포넌트 기반 애플리케이션 개발 플랫폼 모델이다. Oracle8i JSer는 Inprise의 자회사인 Visigenic의 Visibroker를 내장하고 있고 EJB container, ORB등의 기능을 수행하여 완벽한 분산 객체 환경을 제공한다. EJB와 CORBA를 이해하기 위해서는 먼저 분산 객체 기술에 대한 기본적인 이해가 필요하다.

2.2 최상의 데이터웨어하우스 및 OLTP 시스템

Oracle8i는 데이터웨어하우스 사용 환경에 있어

다수 사용자의 시스템 자원을 효율적으로 관리할 수 있는 새로운 관리 기법을 제공하고, 보다 다양한 데이터 분할 방법을 제시할 뿐만 아니라 기존의 OLAP 툴을 효과적으로 사용할 수 있도록 다양한 기능들을 제공한다. 또한 동일한 오라클 데이터베이스 간의 손쉽고 신속한 데이터 이동 방법을 제시 함으로서 효과적으로 데이터웨어하우스를 구축할 수 있게 하였다.

효율적인 자원 관리 기능

데이터웨어하우스 또는 OLTP 시스템에 있어 CPU 및 디스크 자원의 효율적인 관리 기능은 매우 중요한 요소로 부각 되고 있다. 제한된 자원에서 다수의 사용자에게 대한 효율적인 서비스를 위해, 오라클 8i는 각각의 사용자에게 할당된 시스템 자원을 효과적으로 제어할 수 있는 새로운 자원 관리 방법을 제공 한다. 사용자에게 자원 사용에 대한 우선 순위를 부여하고, 사용자간에 자원 사용에 대한 경합이 발생할 경우 보다 높은 우선 순위를 지닌 사용자에게 시스템 자원을 우선적으로 할당 함으로서 비중 있는 업무를 우선 처리할 수 있도록 한다.

테이블과 색인에 대한 다양한 분할 기법

데이터 분할 기능은 대량의 데이터를 지닌 테이블과 인덱스를 관리하기 쉬운 단위로 분할 함으로서, 대량의 데이터에 대한 관리의 편의성 및 가용성 그리고 확장성 있는 성능을 제공 한다.

Oracle8i는 Oracle8i의 Range Partitioning 기법 이외에 Hash Partitioning 및 Composite Partitioning 기법을 제공함으로서 데이터베이스 관리자로서 하여금 용도에 따른 다양한 형태의 데이터 분할을 구현할 수 있도록 하였다.

Hash Partitioning은 row의 특정 컬럼 값에 Hash Function을 수행시킴으로써 다수의 저장 장치에 대량의 데이터를 골고루 분할 할 수 있도록 한다.

Range Partitioning에 비해 관리가 쉬우며 특정 파티션에 데이터가 집중되어 분포 되어질 염려가 없으므로 병렬DML 수행 및 병렬 질의 시 빠른 수행 성능을 나타낸다. Composite Partitioning은 Range Partitioning과 Hash Partitioning 기법이 혼합된 형태로, 일단 Range Partitioning에 의해 분할된 데이터의 특정 컬럼에 대해 Hash Function을 수행시킴으로써 하나의 파티션이 다수의 저장 영역에 골고루 분포 될 수 있도록 한다.

요약 정보 관리(Summary Management)

Oracle® Express® Analyzer과 같은 OLAP툴들은 데이터에 대한 다양한 형태의 정보조회가 가능하도록 기본적인 유연성을 제공 한다. 일반적으로 이러한 툴들은 특정 Dimension의 변경 시에도 빠른 질의 성능을 획득할 수 있도록, 매우 잘 정제된 형태의 다차원 데이터 모델을 요구한다. Oracle8i는 테이블에 대한 요약 정보 관리 기능을 부여하여 다차원 데이터 모델이 효과적으로 구현 되도록 함으로서 강력한 데이터웨어하우스 시스템을 구축할 수 있도록 하였다. 실질적으로, 질의 수행 시 테이블 내의 각각의 레코드를 조회하는 것이 아니라, 사용자는 이미 연산 된 요약 정보 만을 조회 함으로서 매우 빠른 질의 수행 성능을 획득할 수 있다. 또한 Oracle8i의 Query Rewrite 기능은 요약 정보의 기저 테이블에 대한 사용자 질의를 투명하게 요약 정보에 대한 질의로 변형 함으로서 매우 빠른 수행 성능을 보장 한다. Oracle8i의 summary advisory의 기능을 이용하면 분석된 Workload 및 통계 정보를 통해 요약 정보의 생성 및 삭제 그리고 관리에 대한 자세한 정보를 추출할 수 있다. Oracle8i는 Cube와 Rollup operator를 제공하는 데, 이는 전체 통계 정보에 대해 Union같은 Set operator를 사용하지 않고도 중간 통계나 Crosstab 형태의 리포트를 구현 할 수 있도록 한다.

전송 가능한 테이블 스페이스 (Transportable Tablespace)

일반적으로 데이터웨어하우스 시스템의 데이터를 데이터마트로 이동하던가, OLTP 시스템의 데이터를 데이터웨어하우스 시스템으로 이동하는 등의 작업은 빈번히 발생 하면서도 쉽게 처리 되지 않는 지루하고 고된 작업이다.

Oracle8i 는 SQL*Loader를 사용한 Direct Path Loading 또는 병렬 DML의 수행을 통한 데이터의 이동 방법 외에, 특정 테이블 스페이스의 데이터 파일을 하나의 데이터베이스에서 목적 데이터베이스로의 복사(file copy)만으로 데이터의 이동 가능한 Transportable Tablespace 기능을 추가 하였다. 이 기능을 사용할 경우 데이터의 이동 시간은 서버간의 파일 복사에 소요되는 시간과 동일 함으로 데이터베이스 관리자는 데이터 이동 작업을 과거에 비해 보다 신속하게 수행 할 수 있다.

Direct Path Load API

Oracle8i는 direct path load 메커니즘의 구현이 가능한 API을 제공한다. 이 API를 사용할 경우 ISV 와 시스템 관리 툴 개발자는 오라클의 SQL*Loader에 머무르는 고성능의 데이터 Loading 툴을 손쉽게 개발할 수 있게 했다.

기타 DW 기능의 향상

데이터웨어하우스 시스템의 관리 및 성능의 향상을 위해 Oracle8i 는 다양한 다수의 부가적인 기능을 제공 한다. 데이터베이스 관리자는 인덱스의 재생성 작업을 다른 사용자의 자료 입력 및 수정, 삭제 작업에 영향을 주지 않고 수행 할 수 있게 하였으며, Function-Based Index의 사용을 통해 이미 연산 된 컬럼 값에 대한 색인을 생성 함으로서 질의 시 소요될 연산 시간을 단축할 수 있도록 하였다. 또한 내림차순의 색인 생성 기능은 내림차순으로 정렬된 Row를 결과 값으로 리

턴 받고자 할 경우 높은 수행 성능을 획득할 수 있다. 이외에도 Oracle8i에서 새롭게 제공되는 Sample Function을 사용 할 경우 테이블의 Row 또는 Block에 대해서 Function에 정의된 바울 만큼의 데이터에 대해서만 질의를 수행 하게 된다. 이는 평균값을 구하거나 총합을 구하는 등의 수치 연산을 할 경우 해당 연산의 근사치를 얻고자 할 때 효과적으로 사용 될 수 있다.

Oracle8i의 객체에 대한 통계(statistics) 정보에 대한 복사 기능은, 해당 통계 정보를 하나의 데이터베이스에서 다른 데이터베이스로 복사 가능하도록 하여, 데이터의 이동 작업 시 다시금 이동된 객체에 대해 통계 정보를 생성해야 하는 필요성을 없앴다.

자동화된 Standby Database

Oracle7 release 7.3.에서 처음으로 소개된 Standby Database의 기능이 Oracle8i에서는 보다 자동화 되었다. Standby Database에 대한 리두로그 파일의 전송 및 적용이 자동으로 이루어 지게 되었으며, 보다 단순해진 Standby Database의 관리 형태는 현재 운영 중인 데이터베이스 시스템에 오류가 발생할 경우 Standby Database로의 신속한 서비스의 전이가 가능 하도록 하였다.

향상된 큐잉

Oracle8i의 advanced queuing 기능은 기 등록된 클라이언트에 대한 메시지의 자동 전송을 위해 규칙 기반의 publish/subscribe model를 구현 하였으며, 이를 통해 사용자의 애플리케이션은 해당 애플리케이션과 관련된 (즉 다수의 메시지 큐 중 관심 있는) 메시지 만을 정의된 규칙에 의거하여 수령 받을 것임을 예약 할 수 있도록 하였다. 또한 데이터베이스의 구동 및 사용자의 로그인 등 시스템에서 발생 가능한 다양한 시스템 이벤트에 대해 Trigger의 발생을 가능토록 하여 advanced

queue를 통해 해당 이벤트의 발생을 타 애플리케이션에게 전달 가능 하도록 하였다.

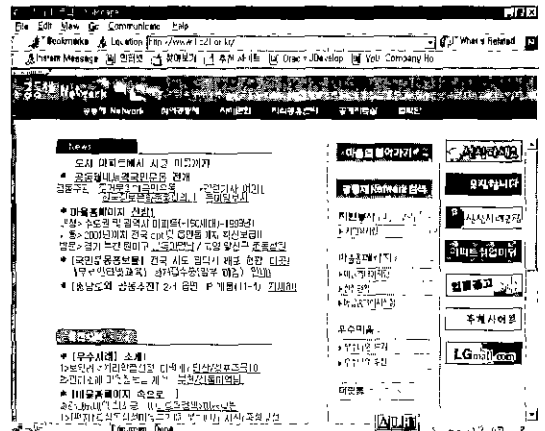
3. 적용 사례

3.1 공동체네트워크 국민운동(<http://www.hc21.or.kr>).

이 공동체네트워크는 현재 대표적인 주거형태로 자리잡고 있는 공동주택은 주거생활의 편리성을 제공하고는 있지만 고밀도 집합형,폐쇄성이라는 특성으로 인하여 바로 옆집이 누구인지로 모르고 살아가는 공동체문화 해체라는 중대한 문제를 초래하고 있다. 이러한 문제점들을 사이버 공간으로 해결하기 위하여 주거문화21이라는 시민단체의 주도로 주거공간내 주민들을 대상으로 정보문화 운동과 공동체문화 운동을 병행, 전개목적으로 공동체 네트워크이 구성되었다.

이 공동체네트워크 홈페이지는 Linux 운영체제를 O/S로 선정하여 아파치 웹서버를 사용하고 있다. 주로 마을(아파트단지)홈페이지를 신청받아 홈페이지를 만들어 주고, 자원봉사 신청 및 우수마을을 선정하여 알림으로써, 마을이나 아파트단지의 자치운영과 참여의 기회를 제공하고 있다.

이러한 기능들을 서비스하기 위해 PHP를 이용



(그림 1)

하여 오라클(Oracle)데이터베이스와 연동된 마을 별 데이터베이스가 구축되고 있는 것이다.

그 이외의 서비스로는 지식정보 공유와 공개 자료실을 운영하여 각종 마을 정보와 정보화를 위한 자료들을 제공하고 있다.

3.2 Ntime.co.kr(http://dwc.net)

ntime은 컴퓨터 & 인터넷 포탈 사이트로 “컴퓨터로 할수 있는 모든것”을 표방 하고 있다.

포탈 에이전트 소프트웨어를 세계최초로 개발하여 지난 11월8일 홈페이지를 통해 공개 하였으며, 서버에서의 지원은 최근 붐을 이루고 있는 리눅스 / 오라클 기반에서의 솔루션을 근간으로 엔진이 개발되어 비용대비 효과를 극대화하고 있다.

인터넷 사업의 성패는 DB를 이용한 엔진 개발이라고 해도 과언은 아닐듯, ntime의 모든 페이지는 오라클의 고성능 엔진을 이용하여 홈페이지 자동 업데이트는 물론 통계까지 일괄 처리하고 있다.

ntime의 주요내용은

1. 북마크가 더이상 필요치 않은 ntime 버전 1.0 을 다운 받을수 있다.

ntime 버전 1.0은 인터넷의 실시간 정보를 사용자에게가장 빨리 전달하여 마치 [내일의 신문]을 연상케하는 폭발적인 정보 전달력이 가장큰 매력이다.

2. 국내 최대의 경품정보 서비스

각종 매체에서 발생하는 경품정보를 조목조목 분류, DB화 하여, 방문자로 하여금 원하는 정보를 빠르게 전달할 수 있도록 구축되어 있다.

3. 채널 서비스

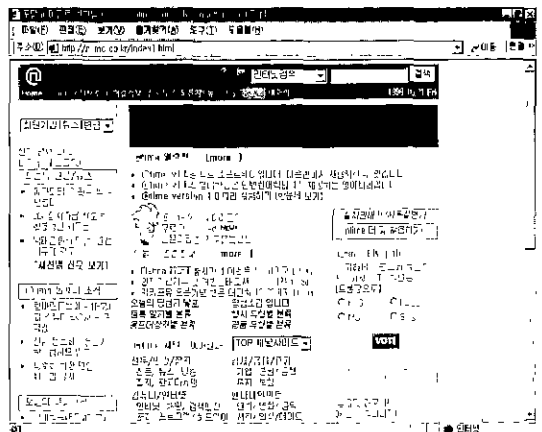
미니 야후 라고 불리는 카테고리 검색엔진을 자체 개발하여 신속한 검색및 편리한 기능을 제공 한다.

통상적인 검색엔진의 경우 “20,000건의 자료를 찾았습니다”와 같이 정작 자료는 무수히 많으나 찾고자 하는 자료를 찾기 힘들었던것을 생각한다면, ntime의 채널 서비스는1일 방문자수 1000명 이상인 사이트만을 데이터베이스화하여 정보의 품질을 높이고 있다.

4. 실시간 여론조사및 토론을 실시하는 토론장 사회 각 분야의 이슈를 모두 모아 분류를 나누고 이 분류를 다시 방문자 각자가 의견을 개진할 수 있도록 개발하여, 1일 10,000명 이상의 참여를 끌어내는 등, 인터넷의 또다른 커뮤니티를 형성하고 있다.

5. 기타 대화방은 기본, 인터넷의 또다른 서비스 “유즈넷”까지도 웹으로 구현하여, 사용자의 편리함을 극대화 하고 있다.

구, 컴퓨터의춤을(http://dwc.net) 사이트(당시 1일 방문자수35,000명, 회원 16만명)에서 시작하여 ntime포탈 소프트웨어 발표를 계기로 연내 1,000만명 이상의 회원가입을 목표로 하고 있다.



(그림 2)

위에서 언급되었던 구축사례이외에도 리눅스를 기반으로 오라클 데이터베이스를 적용하고 있는 곳은 조선 인터넷TV(<http://www.tvchosun.com>)에서의 조선인터넷TV 정보서비스로 기본적으로 PC기반이 아닌 웹터미널을 통한 고객들을 대상으로 하는 정보서비스이다. PC를 모르는 중장년층들을 대상으로 생활에 필요한 고급정보들을 쉽고 편리하게 TV를 통하여 이용하게 하지는 목적이다. 인터넷TV는 콘소시엄형태로 출발하였으며 단말기 개발, 컨텐츠, 통신망, 홈쇼핑, 원격보안, 원격진료 등에서 각 참여업체 각자가 갖고 있는 장점을 최대한 살려 인터넷TV 서비스를 제공하며, 이에 인터넷TV 단말기 판매를 맡은 조선인터넷TV는 인터넷TV 대리점을 대폭 확충하여, 컴퓨터 없이도 TV를 통해 인터넷의 모든 정보를 이용할 수 있게 해주는 것으로 궁극적으로 조선인터넷TV의 정보서비스는 사용자층을 확대하여 종합적인 PORTAL 서비스를 지향하고 있다.

그리고 제일모직의 생산공정 관리 시스템(c-AMICS)은 OS는 Linux, DBMS는 Oracle, WEB은 Apache + PHP를 조합으로 하는 솔루션을 선택하여 타 Unix나 NT에 비해 여러 개발 환경을 저렴한 비용으로 얻을 수 있다는 점이 큰 매력으로 작용하였으며, Web으로 구축된 생산진행 현황을 통해서 생산진행/외주 진행/생산 종결/QR생산 진행/분임 조별 생산진행 현황등을 관련 부서에서 적극 활용할 수 있게 되었다.

4. 결 론

이와 같이 다양한 형태에 이미 리눅스를 채택하여 업무를 서비스하고 있으며, 안정적인 시스템 운용의 검증에 의해 확산일로에 있다고 해도 과언은 아니다. 그러나 여러가지 측면에서 보강되어야 할 점도 없지는 않다.

무엇보다도 중요한 점은 리눅스의 채택이 이러한 특정 부분의 서버로서의 역할이 아니라 엔터프라이즈 서버(Enterprise Server)로서의 역할을 하기 위해서는 SMP 또는 MPP로서의 성능 향상과 이에 따른 클러스터링 기술, 그리고 이러한 환경에서 운용될 수 있는 다양한 애플리케이션의 개발과 보급이 필요하다. 현재는 기초적인 접근 방법으로 단지 MS-Windows에서 제공하는 기능을 Linux에서도 똑같이 제공하므로써 MS-Windows에 익숙해져 있는 사용자들을 Linux로 전환하기 위한 접근 방법을 취하고는 있지만 커널차원에서의 이러한 노력도 필요한 부분이다. 전사적차원에서의 Linux활용을 위하여 오라클에서는 데이터베이스 서버이외에도 윈도우개발툴(Motif용)과 ERP등을 현재 개발 진행 중에 있다.



장 세 국

1981년 고려대학교 통계학과 졸업
 1987년 한국 전자 계산 주식회사
 입사
 1990년 한국 오라클 주식회사 입사
 현재 한국 오라클㈜ 인터넷기술
 팀/Linux지원팀 팀장/부장