
안드로이드 기반의 게임서비스 아키텍처 설계기법에 관한 연구

류창수* · 허창우**

A Study on Design Techniques of Gaming Service Architecture with Android-base

Chang-su Ryu* · Chang-wu Hur**

요 약

스마트폰 게임서비스 산업 규모의 성장속도와 게임서비스 산업이 미치는 영향이 경제적, 문화적으로 꾸준히 증가하는 추세에 있다. 스마트폰 게임같이 동시 플레이수가 많고 축적되는 데이터가 많은 C/S MMO에서 데이터 영속화를 위한 DB 설계는 매우중요한 부분이다.

본 논문에서는 안드로이드 OS에 적합한 운영이나 확장성을 염두에 두고 RDBMS를 활용하여 DB설계에 필요한 매우 짧고 많은 트랜잭션, 트랜잭션의 복잡도, 많은 양의 게임 데이터 처리가 가능하도록 하며, 온라인 게임 업계 이외에서도 널리 사용이 가능한 정규화 MMORPG 게임서비스 아키텍처 설계 기법을 제안한다.

ABSTRACT

The growth speed of the industry of national smart phone game service has become faster and the its economical and cultural effect has increased steadily these days. The DB design for data perpetuation in C/S MMO which has accumulated data a lot and many simultaneous plays like smart phone games is very important.

This paper, mindful of the operation and the scalability suitable for android OS, by utilizing RDBMS, enabling very short, many transactions, their complexity, and a large amount of game data process, suggests fully-qualified MMORPG game service architecture design techniques widely available for more than online game industries.

키워드

게임DB, 게임서비스 아키텍처, MMORPG, C/S MMO

Key word

Game DB, Game Service Architecture, MMORPG, C/S MMO

* 증신회원 : 예원예술대학교 만화게임영상학과(twins70@yahoo.com)

접수일자 : 2012. 10. 05

** 증신회원 : 목원대학교 전자공학과(교신저자)

심사완료일자 : 2012. 10. 25

I. 서 론

컴퓨터 게임의 모든 기본 자료들은 결국 데이터의 집합체, 즉 데이터베이스에서 가져오는 것이며 진행하면서 획득하게 되는 모든 자료들은 데이터베이스에 저장되어 관리가 된다. 광범위한 의미의 데이터베이스에서 조금 범위를 줄여서 컴퓨터 게임이라는 장르에서 살펴본다면 온라인 게임이든 패키지 게임이든 캐릭터, 아이템, 스코어, 랭킹의 모든 것이 데이터베이스이다. 예를 들어 스타크래프트의 맵이 저장되어 있는 것도 데이터베이스이며 디아블로 싱글플레이의 캐릭터와 캐릭터의 모든 능력치와 아이템, 게임상의 모든 아이템의 목록과 정보가 있는 것도 데이터베이스이다[1][2].

스마트폰 게임서비스 산업 규모의 성장속도와 게임서비스 산업이 미치는 영향이 경제적, 문화적으로 꾸준히 증가하는 추세에 있다. 스마트폰 게임같이 동시플레이수가 많고 축적되는 데이터가 많은 C/S MMO에서 데이터 영속화를 위한 DB 설계는 매우중요한 부분이다[3].

본 연구에서는 안드로이드 OS에 적합한 운영이나 확장성을 염두에 두고 RDBMS를 활용하여 DB설계에 필요한 매우 짧고 많은 트랜잭션, 트랜잭션의 복잡도, 많은 양의 게임 데이터 처리가 가능하도록 하며, 온라인 게임 업계 이외에서도 널리 사용이 가능한 정규화 MMORPG 게임서비스 아키텍처 설계 기법을 제안한다.

II. 관련 연구

2.1. 스마트폰 데이터베이스

이동 단말기의 데이터를 효율적으로 저장하거나 검색할 수 있도록 소형 DBMS이다. 무선 통신 기술과 결합하여 스마트폰 장치내에서 운영체제 프로그램, 데이터가 결합되어 제공되는 DBMS이다[11][12].

2.2. 스마트폰 DBMS의 특징

소용량 장비에 탑재 가능한 것으로 제한된 CPU 및 메모리에 적합하도록 최소화 되어 있으며, 디스크 기반 데이터베이스와 복제 및 동기화 기능이 필요하고, 내장형으로 데이터와 응용 프로그램의 결합된 형태로 스마트폰에 탑재한다.

2.2.1. 연결의 비지속성

모바일 기기에 클라이언트가 내장되면, 무선망 내에서 사용자의 이동에 따라 클라이언트도 이동하게 된다. 서버가 이동하는 클라이언트의 위치를 파악하여 지속적인 서비스가 이루어지도록 하는 문제가 요구된다[4][5].

2.2.2. 데이터의 중복성

소형경량이라는 물리적인 제약에 따라 장시간의 통신을 할 만한 충분한 용량의 배터리를 갖추고 있지 않기 때문에 전력 소비를 절감하고 통신 비용을 줄이기 위해 통신 단절이 필요하다. 그로 인한 모바일 컴퓨팅 환경에서 서버와 클라이언트 사이에서 데이터간의 일치성을 어떻게 유지해야 하는가 하는 문제가 요구된다.

2.2.3. 내장형

모바일 컴퓨팅 환경 내에서는 응용 프로그램과 시스템 프로그램이 함께 모바일 기기 내에 내장되며, 수행에 필요한 데이터베이스도 어플리케이션과 결합되어 모바일 기기에 탑재되는 특징을 갖고 있다[8][9].

2.3. 스마트폰 DBMS의 고려 사항

비용 고려 요소 면에서는 애플리케이션을 사용하는 시간 비용과 셀룰러 시스템에서 과금되는 비용 그리고 배터리 전력의 최적화가 필요하다.

단절성과 일관성 면에서는 갱신된 내용이 손실될 수 있을 고려하고 지역적으로 캐시된 데이터가 최근의 내용이 아닐 수 있음을 체크하여야 한다[10].

2.4. 스마트폰 DBMS의 동작 원리

모바일 DBMS는 그림 1과 같이 스마트폰인 클라이언트와 마스터 DB간의 독립적 데이터의 업데이트와 수정된 데이터의 전송 그리고 상호 변경된 데이터 간의 동기화가 이루어져야 한다.

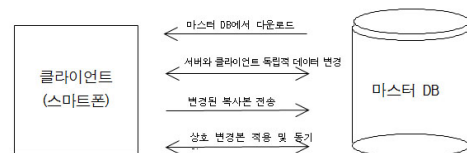


그림 1. 모바일 DBMS 동작원리
Fig. 1 Mobile DBMS operation principle

2.5. 스마트폰 데이터베이스의 요구사항

플랫폼에서의 H/W, S/W에 대한 독립적 이어야 하며, 이동 단말기의 특성을 고려하여야 하고, 서버와 단절된 상태에서의 작업이 빈번하므로 데이터 동기화가 필요하다. 한 명 이상의 동일한 데이터에 대하여 작업이 가능하여야 한다[6][7].

III. 게임 데이터베이스

3.1. 게임 데이터베이스

스마트폰에서의 온라인 게임은 24시간내내 안정적인 서비스가 요구되기 때문에 그림2와 같이 안드로이드 기반의 게임DB는 사용자들의 편의를 위해 빠른 응답속도를 갖추어야 한다.

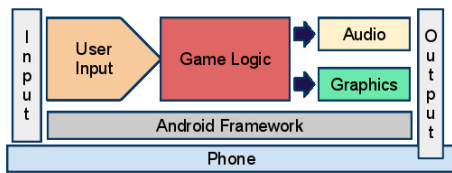


그림 2. 안드로이드기반의 스마트폰 게임 아키텍처
Fig. 2 Android-based smartphone game architecture

3.1.1. 코드성 데이터의 유연성

게임서버나 DB서버는 트랜잭션 처리량은 데이터를 최대한으로 처리되거나 인출되기도 할 수 있다. 게임서버는 클라이언트가 게임을 진행할 수 있도록 코드성 데이터이다. 그런 코드성 데이터는 저장위치에 관계없이 쉽게 변환할 수 있는 성질이 요구된다. 게임 서버가 한번 로딩한 데이터는 캐시의 역할을 할 수 있어 게임 서버를 서비스할 때 성능에 전혀 영향을 받지 않는다.

3.1.2. 다량의 게임 데이터 처리

스마트폰에서 게임데이터 처리는 빠른 처리속도와 안전성을 유지하여야 한다. DBMS에 따른 매우 짧고 많은 트랜잭션 처리, 트랜잭션은 매우 높은 복잡도를 가지며, 캐릭터 또는 회원별 트랜잭션이 발생하며, 많은 양의 게임 데이터와 그 이력 데이터를 처리 할 수 있도록 DB가 설계되어야 한다.

3.1.3. 온라인게임 물리적 아키텍처

데이터센터에서 전용서버를 두어 그 서버를 경유하여 게임 프로그램 간에 데이터를 송수신한다.

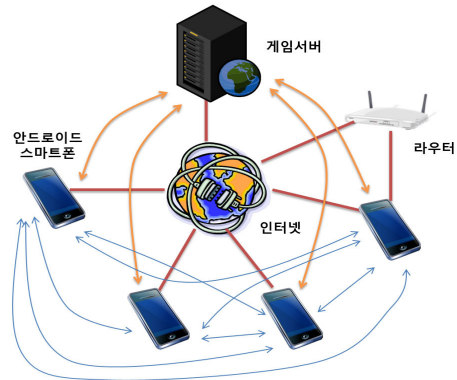


그림 3. 온라인게임 물리적 아키텍처
Fig. 3 Online games physical architecture

그림3과 같이 동기식과 비동기식을 모두 사용하며 동기식 중 폴 메시형으로 단말이 방화벽의 기능을 가진 라우터를 내부에 배치한다.

3.2. 안드로이드 SQLite 데이터베이스

SQLite는 안드로이드에 포함되고 있는 오픈 소스 데이터베이스로 기존의 SQL 구문, 트랜잭션, 선언문 같은 표준 관계형 데이터베이스 기능을 지원한다.

그림4의 android.database.sqlite.SQLiteDatabase와 같이 다른 형태의 2가지 메소드를 참조한다.

```
// API : android.database.sqlite.SQLiteDatabase
public Cursor query (String table, String[] columns, String selection, String[]
selectionArgs, String groupBy, String having, String orderBy, String limit);

// 모든 레코드를 반환하는 쿼리를 실행합니다.
Cursor all = myDB.query("data", null, null, null, null, null, null, null);

// 이름이 google인 레코드를 반환하는 쿼리를 실행합니다.
Cursor sel = myDB.query("data", "name = google", null, null, null, null,
null, null);
```

그림 4. 쿼리 실행과 반환
Fig. 4 Query execution and return

3.2.1. 안드로이드기반 SQLite

모든 안드로이드 기기에서 사용이 가능하며 SQLite DB를 사용하여 모든 데이터베이스 설정이나 관리는 필요로 하지 않으며 안드로이드 플랫폼에 의해 관리한다. 응용프로그램이 데이터베이스를 만드는 경우,

DATA/data/APP_NAME/databases/FILENAME

데이터 경로 Environment.getDataDirectory()는 메소드를 반환한다. APP_NAME은 응용 프로그램 이름이고 FILENAME은 데이터베이스 응용 프로그램 코드에서 지정한 이름이다.

3.2.2. SQLite 데이터베이스

SQLiteDatabase는 안드로이드에서 SQLite 데이터베이스 작업에 대한 기본 클래스이며, 쿼리를 열고 업데이트하고 데이터베이스를 종료 할 방법을 제공한다. SQLiteDatabase이 제공하는 insert(), update() 그리고 delete() 방법이다. execSQL() 직접 SQL문을 실행할 수 방법을 객체 ContentValues는 키와 값을 정의할 수 있다. 키는 테이블 식별자를 나타내고 값은 테이블 레코드에 대한 내용을 나타낸다. ContentValues는 데이터베이스 항목의 삽입 및 업데이트에 사용한다. 검색어를 통해 생성 할 수.rawQuery()와 query() 메서드나 통해 SQLiteQueryBuilder의 클래스가 된다.rawQuery()는 직접 입력으로 SQL SELECT 문을 사용하고 query()는 SQL쿼리를 지정하는 구조화 된 인터페이스를 제공한다.

SQLiteQueryBuilder는 SQL쿼리를 구축 할 수 있다.

3.3 데이터베이스와 데이터 모델

데이터베이스를 만들기 위해 onUpdate()는 기존의 모든 데이터 수정하거나 테이블을 삭제하는 메소드이다. 또한 테이블 이름과 테이블 컬럼에 대한 몇 가지 상수를 정의한다.

```
// Database creation sql statement
private static final String DATABASE_CREATE = "create table "
+ TABLE_COMMENTS + "(" + COLUMN_ID
+ " integer primary key autoincrement, " + COLUMN_COMMENT
+ " text not null);";

public MySQLiteHelper(Context context) {
    super(context, DATABASE_NAME, null, DATABASE_VERSION);
}

@Override
public void onCreate(SQLiteDatabase database) {
    database.execSQL(DATABASE_CREATE);
}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    Log.w(MySQLiteHelper.class.getName(),
        "Upgrading database from version " + oldVersion + " to "
        + newVersion + ", which will destroy all old data");
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_COMMENTS);
    onCreate(db);
}
```

그림 5. 데이터베이스 생성
Fig. 5 Creating a database

그림5와 같이 데이터베이스를 생성하고 클래스의 데이터모델과 데이터베이스를 저장하고 사용자 인터페이스를 표시한다.

```
package de.vogella.android.sqlite.first;

public class Comment {
    private long id;
    private String comment;

    public long getId() {
        return id;
    }
    public void setId(long id) {
        this.id = id;
    }
    public String getComment() {
        return comment;
    }
    public void setComment(String comment) {
        this.comment = comment;
    }

    // Will be used by the ArrayAdapter in the ListView
    @Override
    public String toString() {
        return comment;
    }
}
```

그림 6. comment class 생성
Fig. 6 Create the comment class

그림6의 클래스는 데이터베이스에 저장하고 사용자 인터페이스에 표시하며, 데이터를 포함한다.

3.4.MMORPG 서비스 아키텍처

한 게임월드에서 동시 접속 인원 제한을 두지 않으며, 더 많은 사용자를 위한 복제된 게임월드를 추가하고, 월드간의 커뮤니티케이션과 DB의 복제를 통한 양방향 게임 서비스를 한다.

3.4.1. 게임 서버존 분할

게임의 기획 단계부터 게임 콘텐츠를 구분하여 다양한 요구에 대응할 수 있는 서버를 구성하며 확장이 용이한 DB를 구축한다.

3.4.2. 통합 SNS 서비스

백만 명 이상의 동시접속이 가능하며 게임 월드이 SNS서비스가 2대의 서버로 20만 이상의 채널로 양방향 커뮤니케이션 서비스를 한다.

3.4.3. C/S MMO 데이터베이스 사용

MMO 개발에서는 구체적으로 데이터를 규칙을 가지고 저장하지만 그 규칙이 보다 엄격하여 나중에 사용하기 쉽게 구성하며 KVS(Key-Value Store)은 스키마로 구조 정의되는 테이블의 행에 데이터를 저장하는 DB가 아니라 키와 값의 조합으로 정보를 저장하는 타입의 데이터베이스를 구축하여야 한다.



그림 7. 최적화된 게임서비스 아키텍처
Fig. 7 Game service architecture optimized

그림7의 최적화된 게임서비스 아키텍처를 구축하고 콘텐츠 속성에 따른 DB 데이터를 구분하여 동기식과 비동기식을 사용하고 웹과 연동하여 지원하며 통합 정보관리시스템으로 게임플레이 정보, SNS정보를 제어하는 게임 서버용 데이터베이스에 맞는 RDBMS를 구축한다.

IV. 결 론

안드로이드 OS에 적합한 운영이나 확장성을 염두에 두고 RDBMS를 활용하여 DB설계에 필요한 매우 짧고 많은 트랜잭션, 트랜잭션의 복잡도, 많은 양의 게임 데이터 처리가 가능하도록 하며, 안드로이드 SQLite 데이터베이스를 이용한 온라인 게임 업계 이외에서도 널리 사용이 가능한 정규화 MMORPG 게임서비스 아키텍처 설계 기법을 제안하였다.

향후 스마트폰 환경의 Bluehole의 MMORPG와 NC의 MMORPG 그리고 XL games의 MMORPG를 분석하여 읽기보다 쓰기가 빈번한 데이터 저장에 적합한 데이터베이스 모델링에 대한 연구하고자 한다.

참고문헌

- [1] 나카시마 켄코, “온라인 게임을 지탱하는 기술”, 위키북스, 2012.06.
- [2] 박승제, “안드로이드 게임 개발편”, 제이펍, 2011.03.
- [3] C. S. Ryu, C. W. Hur, “A Study on Youth Ethics Education Using Location-Based Services of Smartphone”, Journal of the Korea Institute of Information and Communication Engineering, vol.16, no.8, pp.1805-1810, 2012.
- [4] 김호광, 이재학, “MMORPG 게임 데이터베이스 설계”, 마이크로 소프트웨어, pp.122-129, 2006.06.
- [5] 김종원, “차세대 MMORPG 서비스 아키텍처”, KGC2010, 2010.05.
- [6] 홍강운, 윤장우, 류원, “게임 가상화 기술 동향”, 정보통신산업진흥원, 2012.08.
- [7] 김상형, “안드로이드 프로그래밍 정복”, 한빛미디어, 2011.

- [8] 임충규, 김성수, 김경일, 원종호, 박창준, “클라우드 컴퓨팅 기반의 게임 스트리밍 기술 동향”, ETRI, 전자통신동향분석, 제26권 제1호, 2001.02.
- [9] 김양훈, 강종구, “신뢰성 있는 게임 환경을 제공하기 위한 게임 DB 보안 플랫폼 개발”, 한국정보기술융합학회논문지, 제1권 제1호, pp.15-29, 2008.03.
- [10] VOGELLA, <http://www.vogella.com/articles/AndroidQLite/article.html>
- [11] 데이터베이스 구축 운영 종합정보, <http://www.dbguide.net>
- [12] 데이터베이스진흥원, <http://www.kdb.or.kr>

저자소개



류창수(Changsu Ryu)

2006년 목원대학교 컴퓨터교육과
(교육학석사)

2010년 숭실대학교 수학교육과
(교육학석사)

2010년 8월~현재 목원대학교 IT공학과 박사과정

2011년 3월~현재 예원예술대학교 만화게임영상학과
교수

※ 관심분야: 애니메이션, 모바일 3D 게임, 컴퓨터 그래픽, 3D 모델링, 모바일 프로그래밍, Web 디자인, 풍력에너지, 인터넷윤리



허창우(Changwu Hur)

1991년 연세대학교 전자공학과
공학박사

1986년~1994년 LG 중앙 연구소

1994년 3월~현재 목원대학교
IT공학부 교수

※ 관심분야: 반도체공학 및 VLSI 설계