AdWare Server Design

We need to the following projects to the document

Ad submission
Click
User registration
Audit

Play List Servlet

Description

This section describes the play list servlet (PLS) for phase I. The PLS is a server side program that services HTTP requests and return HTTP responses. Each request lunches a different thread.
Note : italic fonts describe phase II tasks.

Packages

The PLS is using the following Java packages:

XP

Description

XP is an XML 1.0 parser written in Java. The parser checks a given XML document for well formedness and validity.
URL: http://www.jclark.com/xml/xp/

Usage

The PLS use the XP for
1. parse the client request and make sure that its valid[1]
2. *parse the play list response and make sure its valid*

SAX

Description

SAX (Simple API for XML) is a standard interface for event-based XML parsing, the parser reads the XML document line by line and fires events that contain information about the line that was just read. The PLS listens to particular events of interest and extract the data from the XML document in that way.

URL: http://www.megginson.com/SAX/

[1] The use of document type definition (DTD) will raise the performance of this task

Usage

The PLS use the sax interface both in the XML request and the XML response. In the request the PLS 'looks' for specific tags to build the request object. In the response the PLS sends events to generate the play list XML response.

MM.MySQL

Description

MM.MySQL is a Type-4 JDBC driver. A type 4 driver is an all-Java driver that issues requests directly to the database. This is the most efficient method of accessing the database.
The JDBC API is made up of classes and interfaces found in Java.sql and Java.text packages.
URL: http://www.worldserver.com/mm.mysql/

Usage

The PLS use the JDBC methods to:

- Establish connection – to communicate with the database using JDBC. PLS first establish a connection through the appropriate JDBC driver. The connection object can be used to perform all operations on the given database. *In the next phase we will create pool of connection objects during the servlet initialization.*
- Execute SQL statements and retrieve results – the PLS perform SQL query to the database using both Statement and Prepared Statement objects.

Tasks Flow

The following section explains the task flow when the servlet doPost method is invoked.

Parse request

The PLS parse the XML request and build objects that represents the client Update request.
The data access is done using the SAX API.

Logging the client request

The PLS store the client request information in ClientUpdate table (see document table.sql)
When issuing the same SQL statement repeatedly, it is more efficient to use the Prepared Statement rather then a Statement to execute a query. In the logging we will use the following semantic:

```
PreparedStatement ps = conn.prepareStatement("INSERT INTO ClientUpdate (date, userAgent, playListId,
…) values (?, ?, ?, ?,..)");
```

Generating new Play List

For generating play list the servlet is using both SQL queries and programming filtering. This process is synchronic in order to prevent conflicts when accessing the database.

Following is a pseudo code:

- The list of available ads is build from the following query:

```
ads  =  dbCon.prepareStatement("SELECT * FROM ads WHERE StartDate <= today AND
endDate >= today + 30 AND AdType = "I" AND AdStatus = "A" AND ImpressionsServed <
Impressions ORDERD BY ImpressionsServed ASC);
```

run out ads = dbCon.prepareStatement("SELECT * FROM ads WHERE StartDate <= today AND endDate >= today + 30 AND **AdType = "R"** AND AdStatus = "A" AND ImpressionsServed < Impressions ORDERD BY ImpressionsServed ASC);

Ads list holds all the image ads that are active and can be delivered within the time frame

- calculate the number of seconds need to be delivered back

face time left for today [seconds] = faceTime[today] – faceTimeUsedToday

face time left for today is the number of secondes we can use to deliver special ads today.

predict face time [seconds] = SUM(faceTime[tomorrow] , faceTime[tomorrow + 1] , … faceTime[tomorrow + reqInterval])

predict face time is the number of seconds we predict the user is going to have

goal show time left [seconds] = predict face time – faceTimeLeft

goal show time left is the number of sequined we need to fill with ads.

- Targeting

```
while (face time left for today ) {
        if ad is not in the history {
                select ad [according to target =  today]
                face time left for today -= ad.showFor
        }
        next ad
}

while (Goal show time left ) {
        if ad is not in the history {
                select ad [according to target]
                goal show time left -= ad.showFor
        }
        next ad
}
```

- If we have more time fill it with run out ads
  Find run out ad that is not in the ads history that fits into the Goal show time left.

Default values:

reqInterval = 1 day.
facetime = 30 minutes
faceTimeQuota is ?
histLength = 31 days

Play List Response

When generating XML, it often is useful to generate comments, processing instructions, and so on. XP Writer provides a set of methods for creating specific kinds of nodes in the output.

Following are the method PLS is using to generate the output:

- Starts an element – starts - tag
- Ends an element - end-tag or close the currents start-tag as an empty element.
- Attribute – add attributes to a tag name value pair format
- Comments – writes a comment.

Logging the response

The PLS store the response information in two tables:

1. play list general response table – holds the client info section and the play list general information
2. play list specific response – holds the entry section

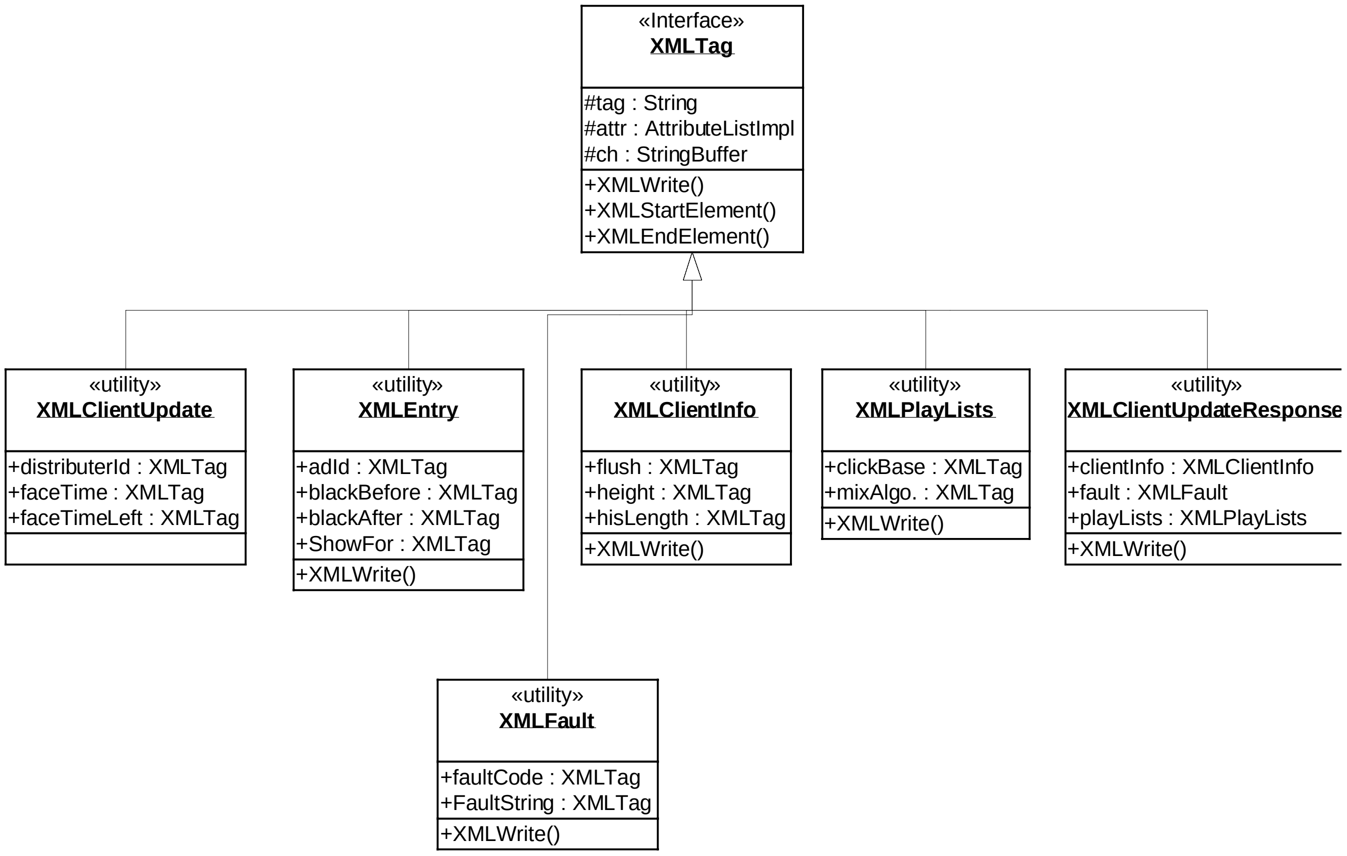
(See document table.sql for tables design)

In this case the PLS use the prepare staminate API to optimize the performance of the query.

Class Diagram:

Mapping XML to Objects:

The following class diagram describes the representation of the play list request and response. The important method is the XML Write this method writes the XML tags.

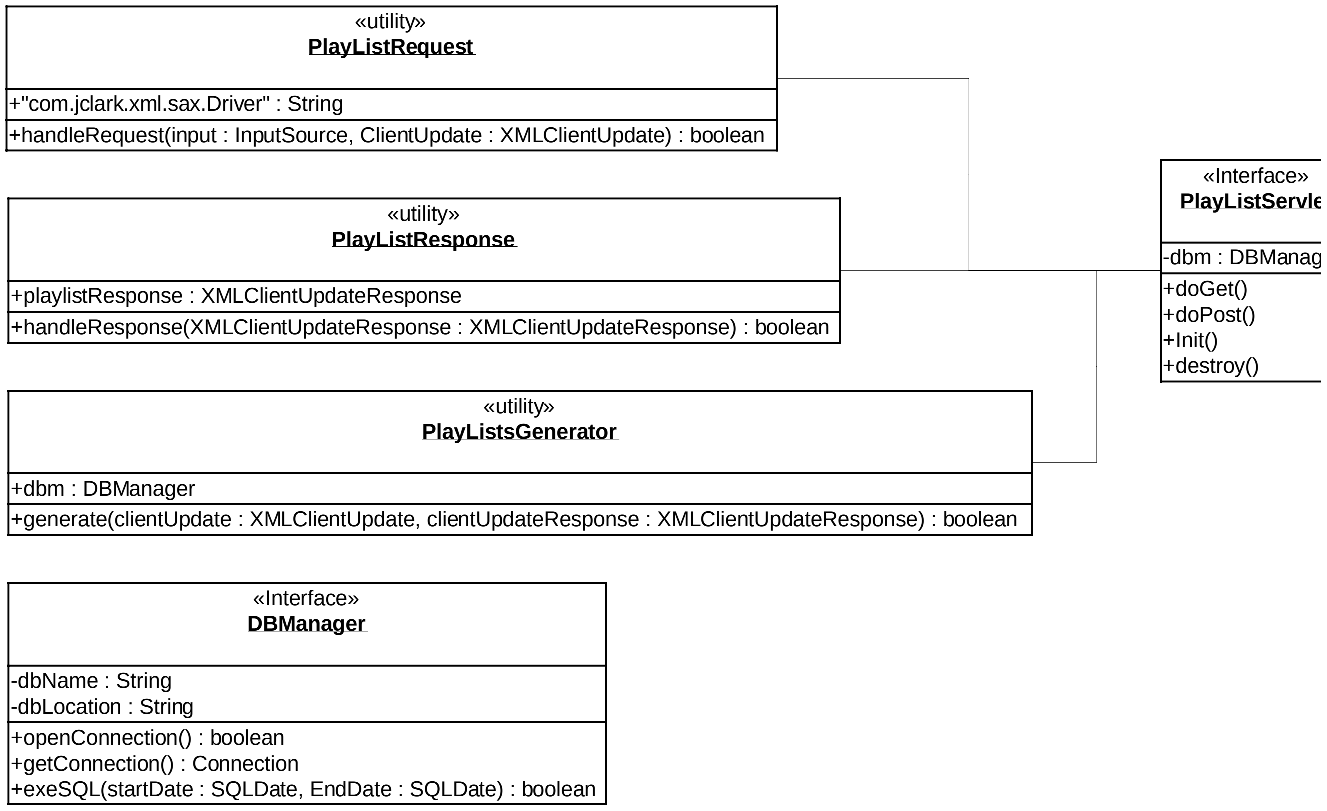


Play List Servlet Classes

The following class diagram describes the play list servlet.

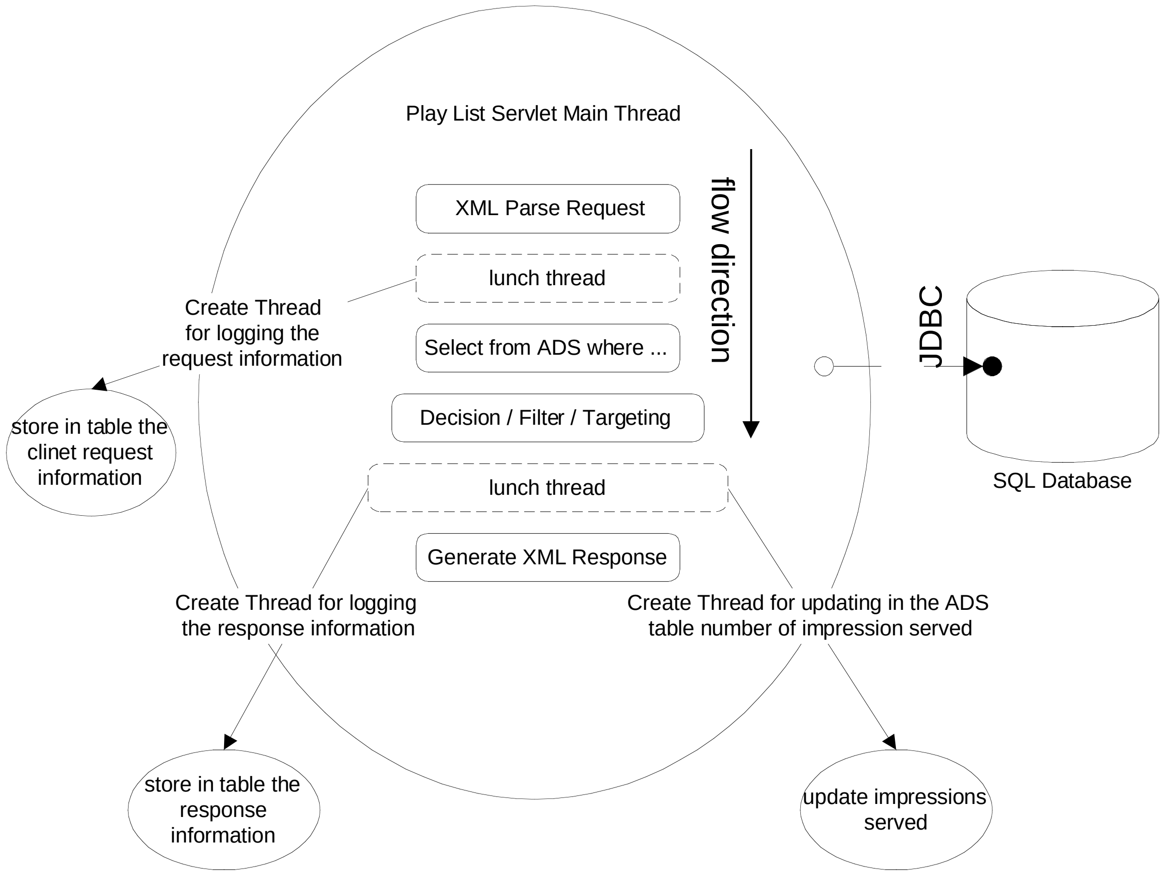
- PlayListRequest class handle the request and map the XML request to the clientUpdate object
- PlayListResponse class handle the response and write the clientUpdateResponse back to the client
- PlayListsGenerate class generates the play lists.
- DBManager class handle the Data Base connection pool

- PlayListServlet class is the servlet interface.


«utility»
PlayListRequest
+"com.jclark.xml.sax.Driver" : String
+handleRequest(input : InputSource, ClientUpdate : XMLClientUpdate) : boolean
«utility»
PlayListResponse
+playlistResponse : XMLClientUpdateResponse
+handleResponse(XMLClientUpdateResponse : XMLClientUpdateResponse) : boolean
«utility»
PlayListsGenerator
+dbm : DBManager
+generate(clientUpdate : XMLClientUpdate, clientUpdateResponse : XMLClientUpdateResponse) : boolean
«Interface»
DBManager
-dbName : String
-dbLocation : String
+openConnection() : boolean
+getConnection() : Connection
+exeSQL(startDate : SQLDate, EndDate : SQLDate) : boolean
«Interface»
PlayListServle
-dbm : DBManag
+doGet()
+doPost()
+Init()
+destroy()

Servlet Threads

All the database storing actions can be threaded. The following scheme illustrates these tasks.


Play List Servlet Main Thread
XML Parse Request
lunch thread
Select from ADS where ...
Decision / Filter / Targeting
lunch thread
Generate XML Response
flow direction
JDBC
SQL Database
Create Thread for logging the request information
store in table the clinet request information
Create Thread for logging the response information
store in table the response information
Create Thread for updating in the ADS table number of impression served
update impressions served

References

Additional documents and books

Name	description
Audit Data Collection And Report Preliminary design	Describes the audit task for gadding client information
Tables.sql	Describes the SQL tables format
Professional Java server programming	Java servlet book
MySQL mSQL	SQL book