

基于 Ajax 技术和 JAVAEE 的分页查询优化^①

林萍¹, 朱婵²

¹(广东科学技术职业学院 计算机工程技术学院, 珠海 519090)

²(广东科学技术职业学院 广州学院, 广州 510000)

摘要: 本文一方面利用 Ajax 无刷新技术, 无需刷新等待, 另一方面利用 Oracle 11g 数据库服务器中 PL/SQL 编程创建存储过程实现了海量数据分页查询, 把两种技术有机的结合在一起, 只将需要显示的那一部分数据查询或计算出来并传输到 WEB 服务器端, 在客户端采用 Ajax 异步通信技术无刷新的获取 WEB 服务器端的数据, 缩短了用户等待的时间, 同时提高了应用系统执行性能。

关键词: 分页查询; Ajax; 异步; XMLHttpRequest; 存储过程

引用格式: 林萍, 朱婵. 基于 Ajax 技术和 JAVAEE 的分页查询优化. 计算机系统应用, 2017, 26(8): 184-189. <http://www.c-s-a.org.cn/1003-3254/5945.html>

Optimization of Data Paging Based on Ajax and JAVAEE

LIN Ping¹, ZHU Chan²

¹(Computer Engineering Technical College, Guangdong Vocational Institute of Science and Technology, Zhuhai 519090, China)

²(Guangzhou College, Guangdong Vocational Institute of Science and Technology, Guangzhou 519090, China)

Abstract: This paper uses Ajax without refreshing technology to eliminate the waiting time. At the same time, it uses the Oracle 11g database server in PL/SQL programming to create a stored procedure realization of data paging query. With these two technologies combined together, by querying or computing and transmitting the necessary part of the data to the WEB server, the client can use the asynchronous Ajax communication technology to obtain WEB server without refreshing the data, which shortens the waiting time of the users, and improves the application performance.

Key words: paging query; Ajax; asynchronous; XMLHttpRequest; stored procedure

信息社会每天的数据量在以 10 亿 GB 级别增长, 随着大数据应用时代的到来, 数据库中的数据量会越来越大, 数据也会越来越多, 在 WEB 应用软件系统的开发中, 数据分页技术也是 WEB 系统开发中经常使用的数据处理技术, 分页显示速度的快慢, 将直接影响 WEB 应用的网络服务质量和性能。Ajax(Asynchronous JavaScript and XML)技术采用异步通信方式, 不强迫用户等待, 屏幕不会出现一片空白, 只是后台进行数据处理, 用户看到的页面元素基本保持不变, 等后台数据处理完毕后, 只更新和刷新用户请求的数据。这种显示方式缩短了客户端的等待时间, 充分体现了“置于用户控

制之下”的界面设计^[2]。同时, Oracle 11g 数据库是一个技术先进的优秀大型数据库管理系统, 该系统更快、应用广泛, 在数据库领域一直占据者领先地位。随着数据量的增加, 高效地实现分页查询是一个重要的问题^[3]。

现有的分页技术大都是把分页功能封装在 WEB 服务器上, 发送 sql 查询语句到数据库服务器, 数据库服务器把查询的结果全部发回到 WEB 服务器, 再由 WEB 服务器进行数据计算和处理等, 这样造成 WEB 服务器压力过大而数据库服务器闲置的状态。本文基于以上问题的考虑, 提出了利用 AJAX 技术和数据库存储过程两项技术的特点, 将两种技术无缝结合, 一方

① 基金项目: 广东省高职教育一类品牌专业资助项目(2016gzpp007)

收稿时间: 2016-12-26; 采用时间: 2017-01-18

面利用 Ajax 无刷新技术, 无需刷新等待, 同时利用 Oracle 11g 数据库服务器中 PL/SQL 编程创建存储过程实现海量数据分页查询, 为用户提供了一个可共享的、优化了的、通用的查询接口, 将计算和查询操作等集中在数据库服务器端, 而每次只将需要显示的那一部分数据传输到 WEB 服务器端, 最后在客户端采用 Ajax 异步通信技术无刷新的获取 WEB 服务器端的数据, 从而减少了对客户机的资源消耗, 提高了应用系统执行性能。

1 关键技术介绍

1.1 Ajax 技术

Ajax 是异步 JavaScript 和 XML 的结合, 是一种支持异步请求, 结合 Java 技术、JavaScript 技术、XHTML 和 CSS、DOM、servlet、XMLHttpRequest 和 XML 等技术组合而成^[1]。

大部分 WEB 网页操作时, 往往只是局部数据改变, 大部分内容是不变的, 如果每次都全部重新执行一遍, 会浪费很多资源, 如果执行局部的操作而其他内容保持不变, 消耗资源减少, 就能大大提高软件的操作性能。在传统的 WEB 同步交互方式中, 当客户端发送 HTTP 请求时, 用户开始等待, WEB 服务器端收到数据后, 处理并响应客户端的请求, 用户接受到需要的数据后显示在浏览器中, 整个过程用户浏览器处于停滞等待状态, 一旦数据量过多过大, 网络负载过重, 那么, 用户就陷入了无限的等待中, 浏览器一片空白, 直至数据完全返回, 这种用户体验时不连贯的。而引入 Ajax 的 WEB 应用模式, 数据传输采用异步交互方式, 能够消除数据交互过程中的处理-等待-处理等待的缺点, 它在用户和 WEB 服务器中引入了一个中间媒介, 当客户端浏览器发送 HTTP 请求时, 通过 JavaScript 方法调用 Ajax 引擎, Ajax 引擎再向 WEB 服务器发送 HTTP 请求, 通过调用 XMLHttpRequest 对象与服务器进行数据交换, WEB 服务器处理完毕并准备好数据后, 通过 XML 文档发回给客户端, 客户端通过 Ajax 引擎获知 WEB 服务器端数据已发送到客户端, 客户端通过 javascript 和 Ajax 引擎解析收到的数据并进行局部更新, 由于客户端与服务器之间的所有数据通信在后台运行, 用户感觉不到, 这样, 用户发出 HTTP 请求后, 屏幕没有太多变化, 不会一片空白, WEB 服务器在后台进行处理请求, 待请求的数据准备好后发送给客户端, 客户端此时

接收需要的数据, 在此过程中客户端根本感觉不到有所等待, 还可以继续和服务器端有其他交互请求, 这种应用模式, 不仅减轻了服务器的负担, 最大限度的减少了用户等待的时间, 是用户体验得到良好的升级, 图 1 所示为传统 WEB 应用模式, 图 2 所示为基于 Ajax 的 WEB 应用模式^[2]。

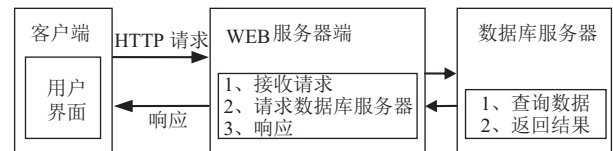


图 1 传统的 WEB 模式

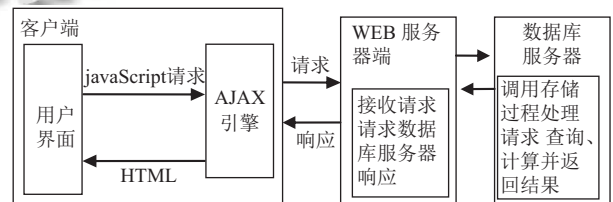


图 2 基于 Ajax 的 WEB 模式

1.2 分页技术

传统 WEB 请求模式中, 数据会全部显示到客户端, 造成客户端等待时间长, 浏览器一片空白的现象。为了解决这个问题, 引入分页显示技术, 即 WEB 服务器只查询需要的数据, 发送给 Ajax 引擎, 再有 Ajax 引擎在适当的时候把数据显示在客户端浏览器中, 这样能够解决不用重载整个页面, 即可更新局部数据的问题, 在整个过程中, 用户感觉不到自己在等待, 还可以继续和 WEB 服务器进行其他交互请求, 大大减少了用户的等待时间, 提高了数据显示速度, 提升用户体验^[2]。

1.3 查询优化技术

分页查询时, 首先进行排序, 然后筛选需要显示的数据。例如, 从科研项目信息表(project_info)中查询年度(i_year)排序(从高到底)第 rownumStart 条到第 rownumEnd 条之间的所有科研项目信息(rownumStart < rownumEnd), SQL 语句可以表示为:

```

SELECT * FROM
(SELECT ROWNUM rn, T1.* FROM
(SELECT * FROM project_info WHERE i_year IS
NOT NULL ORDER BY i_year desc)T1
)T2
  
```

WHERE rn BETWEEN rownumStart AND rownumEnd

该语句由三层嵌套查询组成,最内层首先查询所有项目并按年进行降序排序,然后通过第二层查询增加序号,最外层查询通过 between 和 and 来筛选出需要的结果,这种采用 between 和 and 来筛选的方式效率很低,花费时间长,把筛选的压力全部集中在第三层,如果能在第二层就筛选掉一些记录,会大大提高查询的效率.因此,在 CBO(Cost-Based Optimization)优化模式下,将第二层查询就筛掉大部分记录,从而可以显著提高查询的执行效率.优化语句为:

```
SELECT * FROM
(SELECT T1.*, ROWNUM r FROM
(SELECT * FROM project_info WHERE i_year IS
NOT NULL ORDER BY i_year desc)T1
WHERE ROWNUM <= rownumEnd)T2
WHERE r > rownumStart
```

在高效科研管理系统中,随着查询数据量的增加,上述两种查询语句需要花费的时间如下表所示,从表中可以看出,经过优化的查询语句效率显著提高.表 1 显示了随着表中数据的增加,两种查询方法在 oracle11g 的 SQL DEVELOP 中的效率比较.

表 1 优化与非优化的查询方法效率比较

数据记录数	Rownum Start	Rownum End	非优化查询时间 (s)	优化查询时间 (s)
1250	100	115	0.53	0.056
12500	100	115	0.52	0.051
125000	100	115	0.54	0.05
1250000	100	115	1.06	0.53
11069500	100	115	13.41	0.54

1.4 存储过程与动态 SQL 结合进一步提高分页查询效率

为了进一步提高系统的查询效率,分解 WEB 服务器的工作,可以把查询放在 oracle 服务器中进行,在 oracle 中使用 PL/SQL 程序设计存储过程,通过 IN 模式参数接收外界输入的参数,同时通过 OUT 模式参数把计算后的参数输出到外界,存储过程是由 oracle 的 pl/sql 引擎编译和调用,常驻内存中,当 WEB 服务器请求调用存储过程时,pl/sql 引擎接收 WEB 提交的 IN 参数,然后运行事先编译好的存储过程,执行完成后,通过 out 模式参数把 WEB 服务器需要的数据传递给

WEB 服务器,从而得到一个高效的分页优化查询.在编写查询 PL/SQL 过程时,为了提供一个通用的分页查询存储过程,设计三个 in 参数,分别时每页显示的记录数、当前的页码和动态 sql 语句,再设计了三个 out 参数,分别是可以显示的总页数,查询到的记录数和记录,下面代码是使用动态 SQL 进行分页查询的存储过程的原型^[3].

```
CREATE OR REPLACE PROCEDURE
sp_pageQuery (pageSize number(5, 0), -- 每页显示的记录数
curPageNo number(5, 0), -- 当前页号
sqlSelect VARCHAR2(500), -- 动态 SQL 语句
totalPages OUT number(5, 0), -- 总页数
totalRecords OUT number(5, 0), -- 记录总数
outCursor OUT REFCURSOR -- 采用优化查询返回当前页的数据集
)
```

传统的数据分页查询采用 DAO 记录集分页法,也就是先将所有数据存放到服务器端内存中,然后利用 DAO 层程序实现分页,适合于少量数据的数据分页查询.传统的分页查询与本文提出的优化分页查询,在高校科研管理系统中,随着记录数的增加,效率得进一步提高.表 2 对两种方法的执行效率进行了比较.

表 2 传统与优化的分页查询方法执行效率比较

数据记录数	查询页码	非优化查询时间(s)	优化查询时间(s)
1250	50	1.225	1.407
12500	100	12.901	0.15
125000	1000	11.797	0.137
11069500	10000	17.226	0.16

2 基于 Ajax 技术和 JAVAEE 以及优化的 SQL 实现数据分页的设计

通过存储过程优化方案和 Ajax 相关技术在高校科研管理信息系统中的应用,高效科研管理系统的效率得到了很大的提高,下面详细介绍具体的系统数据分页的设计.

本实例的设计环境是:软件开发工具为 myeclipse,开发语言为 java 和 JavaScript,数据库管理系统为 Oracle11g.下面是权限管理模块中使用分页功能浏览页面运行结果如图 3 所示.

2.1 信息浏览数据分页的设计思路

利用高校科研管理系统为例,来说明实现动态高

效分页的具体方法. 客户进入到“科研项目”页面, 首先显示首页数据, 此时“上一页”按钮不可用, 当客户点击“下一页”或导航页码按钮时, 把页码传递给 JavaScript 并由 Ajax 引擎发送请求给 servlet, servlet 在后台获取数据, 调用 DAO 层, DAO 层向数据库服务器发送请求, 数据库服务器接收到请求, 调用优化后的存储过程进行数据处理, 并把结果返回给 WEB 服务器, WEB 服务器把结果返回给 javascript 和 Ajax 引擎, 客户端接受到返回的数据, 局部刷新需要的页码, 实现分页操作^[1], 分页的流程如图 4 所示.

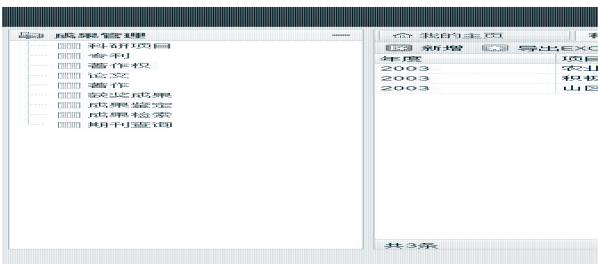


图 3 具有分页功能的高校科研管理系统

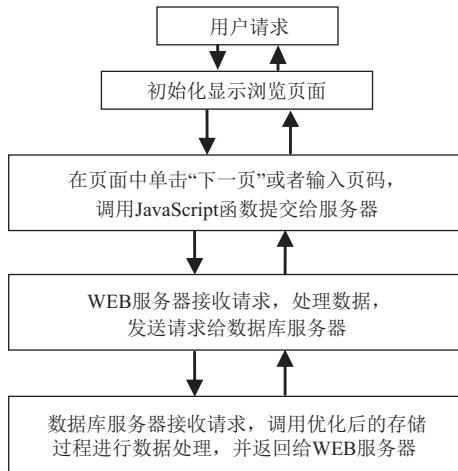


图 4 Ajax 和存储过程实现分页功能

2.2 客户端利用 Ajax 程序实现页面分页显示

1) 第一步: 在 JavaScript 中获取 XMLHttpRequest 对象

```

.....
try {
xmlhttp = new
    ActiveXObject("Msxml2.XMLHTTP");
} catch (e){
try {

```

```

xmlhttp = new
ActiveXObject("Microsoft.XMLHTTP");
} catch (e){
try {
xmhttp = new XMLHttpRequest();
if (xmlhttp.overrideMimeType){
    xmlhttp.overrideMimeType("text/xml");
}
} catch (e){
}
}
}
}
}
}

```

上述代码为了获取不同浏览器的 XMLHttpRequest 对象, 充分考虑到浏览器兼容性的问题而采用不通的创建方法, 从而成功创建异步对象 XMLHttpRequest, 最后返回结果.

2) 第二步: 客户端通过 JavaScript 中使用 XMLHttpRequest 对象向服务器发送分页请求

```

.....
var showpage="page="+p;
var url="pageServlet? "+showpage;
xmlhttp.onreadystatechange = dealReuquest;
xmlhttp.open("post", url, true); //通过处理程序执行 servlet 获取需要的数据
//如果通过 form 的 post 方式提交请求, 还需要以下两句
xmlhttp.setRequestHeader("Content-Type",
"application/x-www-form-urlencoded");//修改 MIME 类别
xmlrequest.send(null);//向服务器发送请求信息
.....

```

上述代码首先时获取浏览器对象并调用 open() 方法访问控制器, 控制器获取数据后返回给浏览器对象, 该数据时通过 servlet 的 out.println() 产生并存放在浏览器中的, 该浏览器对象 xmlhttp 变量表示. 代码中使用 null 调用 send() 函数, 是因为已经在请求 URL 中添加了要发送给 WEB 服务器端的数据, 所以请求中不要发送任何数据. 这样就发出了请求, 服务器就会按照要求工作了.

3) 第三步: JavaScript 中利用 Ajax 引擎处理服务

器返回的信息

```
.....
xmlrequest.onreadystatechange = function(){
if(xmlrequest.readyState == 4){
if(xmlrequest.status==200){
//处理信息并更新页面
}
}else{
//信息没返回, 请求异常
}
}
.....
```

请求状态改变时 XMLHttpRequest 对象调用 onreadystatechange 事件, 指定函数来接收服务器返回的数据, 根据需求动态更新页面. 请求完成时 XMLHttpRequest 对象的 readyState 属性值为 4, HTTP 状态响应完成时 XMLHttpRequest 对象的 status 属性值为 200. 因此判断这两个属性的值即可获知服务器是否处理完毕.

2.3 服务器端利用 java WEB 程序根据分页要求将相关数据库记录返回客户端

1) WEB 层获取客户端发送的请求, 根据请求的页数, 调用 DAO 层取出需要的数据

```
public void doGet(HttpServletRequest request,
HttpServletResponse response)
throws ServletException, IOException {
.....
Project_infoDAO project_infodao = new
Project_info DAO();
String pageString =
request.getParameter("page");
int page = 1;
if(pageString!=null){
page = Integer.parseInt(pageString);
}else{
int pageNum = project_infodao.selectCount();
request.getSession().setAttribute("pageNum",
pageNum);
}
List<Project_info> list =
project_infodao.selectAll(page);
request.setAttribute("project_infos", list);
```

```
request.setAttribute("page", page+1);
```

```
.....
}
```

通过 servlet 获取用户请求的页码, 从而发送请求给 DAO 层, 等待 DAO 层的结果.

2) DAO 层利用泛型技术和 oracle11g 的优化存储过程取出相应数据的实现

```
public List<Treatise> selectAll(int page){
.....省略获取连接等代码
List<Project_info> list = new ArrayList<T>();
int pageSize = AllVirable.PAGESIZE;
CallableStatement proc = null;
ResultSet rs = null;
proc =con.prepareStatement("{ call sp_pageQuery(?, ?,
?, ?, ?, ?)}"); //调用存储过程
.....省略设置参数等
proc.registerOutParameter(6,
oracle.jdbc.OracleTypes.CURSOR);
proc.execute();//执行
rs = (ResultSet)proc.getObject(6);
while(rs.next()){
Project_info project_info = new Project_info ();
...//省略设置 project_info 属性
list.add(project_info);
}
.....省略关闭连接和出错控制等
return list;
}
```

DAO 层不直接用 SQL 查询获取数据, 而是采用调用 oracle 过程的语句向 oracle 数据库服务器发送请求, oracle 通过调用优化后的存储过程, 高效的获取需要的数据, 最后通过动态游标把数据发回给 WEB 服务器.

3 结论

在高校科研管理系统中, 引入 Ajax 的 WEB 应用模式, 数据传输采用异步交互方式, 消除数据交互过程中的处理-等待-处理等待的缺点, 同时利用 oracle11g 的存储过程, 把对数据的选取和计算等放在了 oracle 数据库服务器上, 计算结果发送给 WEB 服务器, 减小了 WEB 服务器压力并提高了数据库服务器的使用效率. 通过用户对高校科研管理系统的使用反馈,

用户对该系统无需等待刷新和高效的显示结果非常满意,由此可见,基于 Ajax 和 JAVAEE 开发的、并把数据查询放在数据库服务器的存储过程中进行的 WEB 分页技术高效、实用,适合在 WEB 系统开发中推广应用.

参考文献

- 1 屈武江. 基于 Ajax 技术的 ASP.NET 数据分页. 计算机系统应用, 2013, 22(9): 154-159.
- 2 刘红坤. 基于 Ajax 和 PHP 数据分页的实现. 计算机系统应用, 2012, 21(2): 218-220.
- 3 孙风栋, 闫海珍. Oracle 10g 数据库海量数据分页查询优化. 计算机应用与软件, 2011, 28(9): 137-139.
- 4 韩兵, 王照清, 廖联军. 基于 MySQL 多表分页查询优化技术. 计算机系统应用, 2016, 25(8): 171-175.
- 5 林萍. 基于 SSH+DWR 的高校科研管理系统研究与实现. 计算机应用与软件, 2014, 31(7): 66-68.
- 6 黄龙军. 应用存储过程实现数据分页. 计算机系统应用, 2012, 21(12): 186-189. [doi: 10.3969/j.issn.1003-3254.2012.12.042]
- 7 王莉. 基于 LINQ 实现数据分页优化. 电脑与信息技术, 2011, 19(6): 42-43, 53.
- 8 李辉, 王瑞波. 多条件分页查询优化的设计方法. 计算机工程, 2010, 36(2): 51-52.
- 9 湛湘倩, 狄文辉, 孙冬. 基于 SSH 框架与 AJAX 技术的 Java WEB 应用开发. 计算机工程与设计, 2009, 30(10): 2590-2592, 2596.
- 10 蒋维. Ajax 技术在网络 B/S 架构中的应用研究. 微计算机应用, 2008, 29(7): 29-33.
- 11 周志刚, 寇艳芳. 基于 ASP.NET 数据分页技术的研究. 计算机与网络, 2008, 34(17): 47-50. [doi: 10.3969/j.issn.1008-1739.2008.17.036]