·农业信息与电气技术·

基于 Web 的自动灌溉控制系统数据实时推送设计与开发

李淑华[1,3,4,5]，郝星耀[2,3,4,5]，周清波[1※]，潘瑜春[2,3,4,5]

（1. 中国农业科学院农业资源与农业区划研究所，北京 100081；2. 北京农业信息技术研究中心，北京 100097；
3. 国家农业信息化工程技术研究中心，北京 100097；4. 农业部农业信息技术重点实验室，北京 100097；
5. 北京市农业物联网工程技术研究中心，北京 100097）

摘 要：基于 Web 的自动灌溉控制系统是当前农业节水信息技术发展的主流趋势，为了提供定制灌溉控制方案和精准的用水计量，系统需要较高的数据传输实时性能，而基于 Web 的应用程序在实时性上表现较差，难以满足应用需求。针对这一问题，该文首先分析了基于 Web 的自动灌溉控制系统的结构和数据传输实时性瓶颈，提出了通过数据推送模式提高实时性的方案，并对数据层与逻辑层、逻辑层与表现层之间的具体数据推送模式进行了设计。通过编程开发完成基于 Web 的灌溉控制系统的构建，实现了数据实时推送的机制，并对系统数据采集和控制指令发送过程的实时性进行测试。结果表明：数据采集平均延时为 1 676 ms，控制数据从发送到结果返回的平均延时为 3 378 ms，基本能够满足其设备控制和灌溉决策的需要；软件系统内采集和控制过程的数据库至客户端数据传输的平均延时分别为 124 和 118 ms，消除了数据拉取模式中的延时因素，对提高系统实时性起到了重要作用。该研究为基于 Web 的实时监测与控制系统的开发提供了方法参考。

关键词：灌溉；数据采集；设计；灌溉控制；实时；数据推送；观察者模式
doi：10.11975/j.issn.1002-6819.2015.15.018
中图分类号：S274.2 **文献标志码**：A **文章编号**：1002-6819(2015)-15-0133-07

李淑华，郝星耀，周清波，潘瑜春. 基于 Web 的自动灌溉控制系统数据实时推送设计与开发[J]. 农业工程学报，2015，31(15)：133－139. doi：10.11975/j.issn.1002-6819.2015.15.018 http://www.tcsae.org
Li Shuhua, Hao Xingyao, Zhou Qingbo, Pan Yuchun. Design and development of real time data push in web-based automatic irrigation control system[J]. Transactions of the Chinese Society of Agricultural Engineering (Transactions of the CSAE), 2015, 31(15): 133－139. (in Chinese with English abstract) doi：10.11975/j.issn.1002-6819.2015.15.018 http://www.tcsae.org

0 引 言

作为物联网技术的典型应用之一，自动灌溉控制系统不仅为设施农业灌溉和城市绿化灌溉等领域节约了大量水资源，也为根据作物和花木的需水特性进行个性化灌溉和用水精准计量提供了基础[1-2]。自动灌溉控制系统由硬件和软件 2 部分组成，硬件部分包括灌溉控制器、电磁阀和传感器，软件安装运行于工控机上并与硬件部分相连，是人机交互入口和控制中心，根据监测数据和灌溉模型进行自动/半自动的灌溉控制。自动灌溉方式主要有 2 种：一种是建立周期性的灌溉方案，进行定时灌溉；另一种是以土壤水分监测数据和气象数据为输入，通过灌溉决策模型计算灌溉时间和灌水量，实现按需灌溉[3-4]。

实时性对于自动灌溉控制系统至关重要，传感器数据采集的实时性也决定了灌溉决策的时效性，同时，灌溉控制指令也必须实时地传输到电磁阀，保证灌水量的精确控制[5-6]。网络环境和系统架构是影响实时性的主要因素，通常在一定的工程条件下网络传输速度是确定的，此时系统架构对实时性起着决定性作用。当前的灌溉控制软件系统主要是基于客户端/服务器（C/S）结构开发的桌面应用程序，客户端软件与数据库之间通讯效率很高，且网络结构简单，因此系统实时性能够得到有效保障。但是，由于 C/S 结构程序必须运行于特定的软硬件平台，无法实现多种设备平台的覆盖，适用范围非常有限，而针对所有可能平台进行开发在多数情况下并不可行[7-8]，另外，C/S 结构程序升级维护成本较高，同时难以实现多个项目集中管理，因此，当前软件应用系统更多地选择基于浏览器/服务器（B/S）结构进行开发，即 Web 应用程序。Web 体系中的超文本标记语言（hypertext markup language，HTML）、计算机系统模拟（computer system simulation，CSS）和 JavaScript 等技术为 Web 应用提供了一个开放的平台，任何设备上的浏览器只要符合相关标准，Web 应用程序都可以在上面运行，这就使得 Web 应用以较低的成本实现了最大的可达范围[9-10]。虽然 Web 设计之初并没有考虑应用开发，但随着技术的演进，Web 应用程序可实现的功能已经达到了本地应用的水平，因此采用 Web 应用程序方式来开发自动灌溉控制系统具有较好的技术可行性和应用前景[11-12]。

在 Web 应用程序中常常通过 Ajax 方式来模拟实时效

收稿日期：2015-06-01 修订日期：2015-07-10
基金项目：国家高技术研究发展计划（863 计划）（2011AA100509）
作者简介：李淑华，女，河北衡水人，博士生，助理研究员，主要从事农业信息化技术研究。北京 北京农业信息技术研究中心，100097。
Email：lish@nercita.org.cn。
※通信作者：周清波，男，湖南沅江人，研究员，博士生导师，主要从事农情遥感领域的基础研究和应用基础研究。北京 中国农业科学院农业资源与农业区划研究所，100081。Email：zhouqb@mail.caas.net.cn。

果[13]，数据刷新的频率直接决定了延时长短，通过提高刷新频率来改善实时性能，不仅会增加无效网络传输，对服务器资源也是极大的考验。本文针对这一问题展开研究，通过数据推送方式实现数据传输，改善 Web 应用系统的实时性，满足自动灌溉控制需求。

1 系统结构与数据实时推送设计

1.1 系统结构

在基于 Web 的自动灌溉控制系统中，硬件设备和灌溉控制软件不采取直接通信的方式，而是通过独立的数据通讯模块实现数据传输（见图 1），将硬件设备的监测数据和自身状态数据发送至灌溉控制软件系统，并将软件系统发出的控制命令数据传输至硬件设备，不仅降低了系统耦合性，也使软件部署更加灵活，有利于实现多个灌溉控制项目的远程集中管理[14]。

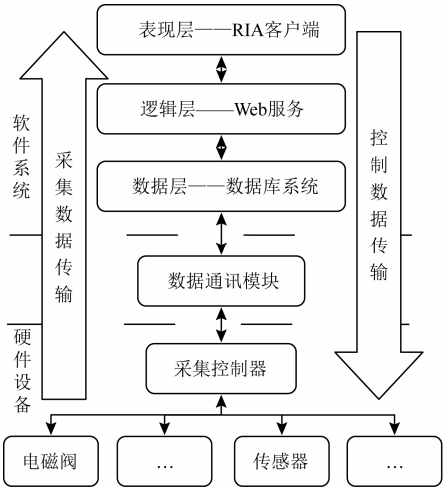


图 1 基于 web 的系统结构图

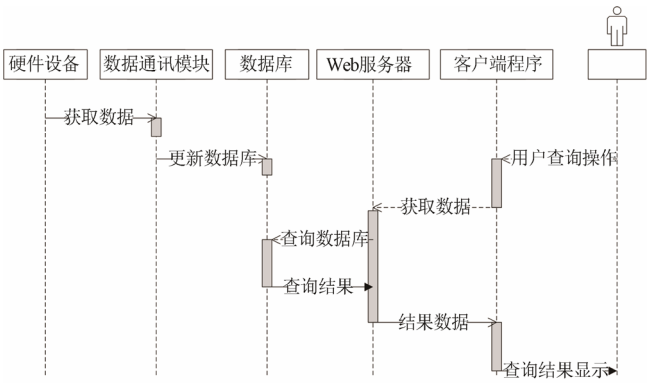
Fig.1 Web-based system structure diagram

基于 C/S 架构的灌溉控制软件限制了服务对象和应用范围，同时也需要较高的成本来维护和更新系统，基于 B/S 架构进行软件系统设计具有更大的优势。在软件架构方面采用 Web 应用程序的典型三层架构，即数据层、逻辑层和表现层（见图 1）。但是传统基于 HTML 页面的表现层技术不能满足灌溉控制中的操作交互和实时数据传输要求，因此本文采用丰富互联网应用（rich internet application）技术进行表现层的开发，使其除了展示服务器返回数据之外，增加更多数据表现和交互功能，在改善用户体验的同时，也减少了网络数据传输量[15]。

1.2 数据传输实时性瓶颈分析

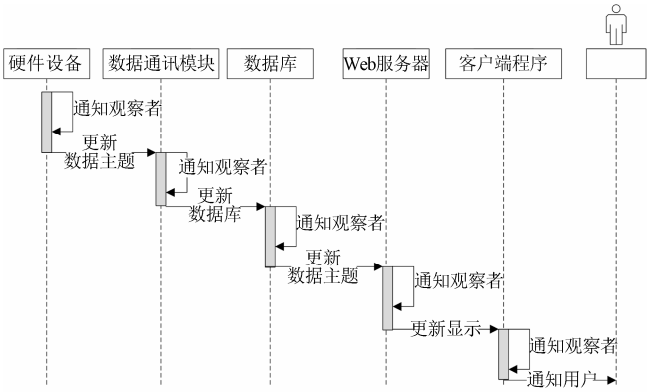
在硬件设备与数据通讯模块之间采用嵌入式程序进行通讯，延时很短；数据通讯模块与数据库之间的数据传输为桌面级应用程序，实施性也能够得到保证；在软件系统内部，由于架构特点和各层次之间数据传输方式的差异，实时性的提高往往以巨大的网络带宽消耗和服务器性能需求为代价，在实际应用中存在较大的问题，因此在整个灌溉控制系统中，软件部分是实时性提高的关键。

软件系统中数据传输包括控制和采集 2 个过程。控制过程的数据传输是从客户端应用程序到数据库，其实现过程为客户端程序向 Web 服务器发出请求，服务器接收数据并保存到数据库中，整个过程是由数据源向接收数据层的推送过程，其延时主要来自于网络通讯延时和计算耗时，实时性较高。采集过程的数据传输是从数据库到客户端应用程序，更新数据是由接收层通过查询操作拉取获得。在数据拉取模式中，客户端程序调用一个异步方法来从服务器获取数据，然后服务器端程序通过调用一个同步方法从数据库中获取最新数据，数据库返回数据给服务器端程序后，服务器再将数据发送给客户端（见图 2a）。数据拉取过程通常可以较短的时间内完成，但是由于硬件设备的数据采集与软件系统的数据获取相互独立，客户端程序并不知道何时数据库数据有了更新，难以实现数据同步更新，延时较长且不确定，形成了整个系统实时性的瓶颈，影响着整个灌溉控制过程。



a. 数据拉取

a. Data pull



b. 数据推送

b. Data push

图 2 数据采集过程的数据拉取和推送序列图

Fig.2 Sequence diagram of data pull and push modes in data acquisition process

获得更新数据的延时主要由数据拉取的频率决定，如果数据更新为随机事件，执行数据拉取的间隔时间为 t，获得更新数据的延时为 Δt，在不包括网路延时和计算耗时的情况下，则有：

$$\Delta t \in (0, t) \tag{1}$$

$$\overline{\Delta t} = t/2 \tag{2}$$

假设客户端程序每 10 s 发起 1 次数据拉取请求，则获得更新数据的延时范围是 0～10 s，平均延时为 5 s，此时 1 min 内每个客户端连接对服务器和数据库的访问为 6 次。如果将数据拉取频率提高到 1 s 内 1 次，那么延时范围可以控制在 0～1 s，平均延时 0.5 s（不包括网路延时和计算耗时），此时 1 min 内的访问数为 60 次。如果同时 10 个客户端程序在线，访问数就会高达 600 次，平均 1s 内 10 次，这对于 web 服务器和数据库服务器都造成了巨大的访问压力，而这 600 次的访问中，绝大部分得到的都是未更新的数据。

过高或过低的访问频率都存在问题，那么是否可以确定一个最佳的数据拉取频率，在提高实时性的同时消耗较少的资源。传感器通常有 2 种可能的数据采集模式，按固定的时间间隔，或监测值超过一定阈值后进行采集。在前者情况下，理论上如果数据拉取的频率与传感器数据采集的频率相同，可以做到同步的数据采集，但实际上硬件设备到数据库的数据传输存在不确定长度的延时，因此客户端程序的数据拉取周期也难以确定。在后者情况下，由于数据更新没有任何规律，客户端程序的数据拉取频率同样无法确定。

1.3　数据实时推送设计

解决数据采集过程中实时性瓶颈的最有效方案就是采用数据推送方式来传输数据更新，在数据推送模式中，数据更新活动之间相互连接，各部分操作之间不会因为查询间隔而互相等待，降低数据传输延时的同时，也减少了无用的数据查询操作（见图 2b）。对比图 2a 和图 2b 可以看出，与数据拉取模式相比，数据推送模式在实时性方面具有明显的优势，本节将重点分析如何在 web 应用程序中实现数据推送。

1.3.1　观察者模式

观察者模式（又称为发布-订阅模式）是软件设计模式的一种，在这种模式中，一个目标对象管理所有相依于它的观察者对象，并且在它本身的状态改变时主动发出通知，常被用来实现事件处理系统[16-17]。观察者模式的实现中定义了一个一对多的依赖关系（见图 3），允许多个用户同时观察同一个数据主题，当这个主题的数据状态发生变化时，会通知所有相关用户，根据最新的数据更新自己的状态。

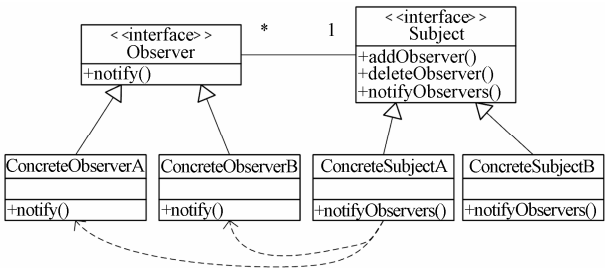


图 3　观察者设计模式类图

Fig.3　Class diagram of observer design pattern

在自动灌溉控制系统中采用观察者模式来实现数据传输能够很好地解决当前实时性能问题：数据逐层按照通知方式进行传递，消除了查询频率与数据更新频率不同造成的延时；数据推送仅在数据源发生变化时发生，避免了大量不必要的查询操作。

在不同层次之间实现观察者模式的方式不同，下面分别对灌溉控制软件的数据层与逻辑层，以及逻辑层与表现层之间，如何通过观察者模式进行数据推送进行分析。

1.3.2　数据层与逻辑层之间的数据推送设计

在数据层与逻辑层之间应用观察者模式，可以实现数据从数据层到逻辑层的推送，其中，被观察者为数据库中数据表，观察者为 Web 服务器中监听程序，数据通讯模块将更新数据插入数据表后，数据库立即通知相关 Web 服务程序，执行数据主题更新（见图 4）。

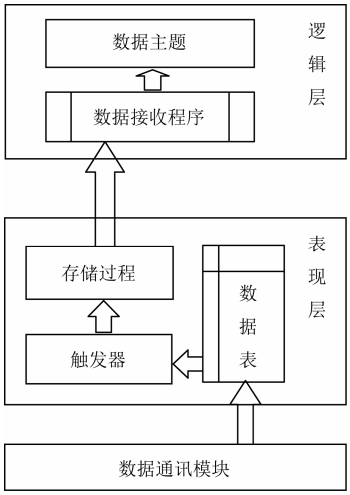


图 4　数据层与逻辑层间的数据推送机制

Fig.4　Data push mechanism between data tier and logic tier

从图 4 中可以看出，实现数据层到逻辑层的数据推送的主要步骤如下：

1）在 Web 服务器创建数据接收程序，以响应数据更新通知；

2）在数据库创建一个存储过程，实现发送数据到 Web 服务器；

3）在被观察的数据表中创建 insert 触发器，其中调用 2)中建立的存储过程；

4）当更新数据插入被观察数据表中时，触发器调用存储过程将更新数据发送到 Web 服务器，服务器程序接收数据并更新对应的数据主题，为下一步推送数据至客户端程序做准备。

1.3.3　逻辑层与表现层之间的数据推送设计

在逻辑层中，被观察者为运行在 Web 服务器中的程序对象，表现层中观测者为客户端程序中的程序对象。由于 Web 服务程序无法直接向客户端程序发起数据连接，因此在逻辑层与表现层之间实现观察者模式，需要客户端程序加载时，在逻辑层和表现层之间建立实时的双向数据连接，并观察逻辑层的一组数据主题，当这组数据主题更新后，就会通过数据连接通知客户端程序（见图 5）。表现层发出连接请求后与逻辑层建立连接并保持，

然后通过数据流请求，为逻辑层数据推送提供下行通道，而表现层推送数据至逻辑层，则采用内部 HTTP 连接按照请求应答方式进行。

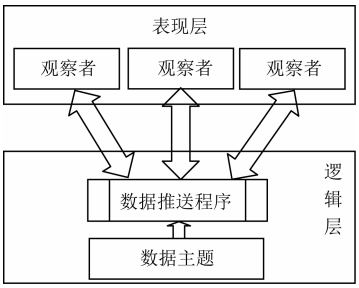


图 5 逻辑层与表现层间的数据推送模型

Fig.5 Data push mode between logic tier and presentation tier

2 自动灌溉控制系统开发

2.1 软硬件环境

本文研究中，硬件部分采用了北京农业智能装备研究中心开发的 WS1000 墒情监测设备和 EIC 直流灌溉控制器，其中墒情监测设备可获得的采集量包括土壤温湿度和各类气象指标，EIC 直流灌溉控制器是一种主要用于温室和小型绿地灌溉的小型灌溉控制器，使用电池供电，不需要布设外部电缆，其控制的终端为直流电磁阀。

数据通讯模块采用基于短消息服务（sms）的无线传输方式，将数据编码后以短信形式发出，接收端收到短信后，经过解析提取数据，传递给数据接收对象。

在软件系统技术选型中，考虑技术的成熟度、市场占有率以及技术之间衔接的紧密程度，客户端应用程序使用 Flex 技术开发，并且运用 BlazeDS 框架来构建 Flex 前端与 Java 后端之间的通讯。逻辑层采用 Java 开发，Web 服务器使用 Tomcat，数据库采用 SQL Server，通过 JDBC 连接数据库。

2.2 实现实时数据推送的关键步骤

2.2.1 数据库开发

数据库开发的核心目标是自动推送数据给 Web 服务器，实现这种机制的方式是使用数据库存储过程和触发器。首先，在 SQL Server 中创建存储过程“SP_SEND_HTTPREQUEST”实现通过 HTTP 请求将数据发送到指定服务器端的 Servlet，然后在监测数据表上创建触发器“TRIGGER_STATION_INSERT”，当有新数据插入数据表后，该触发器将调用“SP_SEND_HTTPREQUEST”，将数据发送到 Web 服务器。

2.2.2 Web 服务开发

逻辑层是开发的重点，它既是数据层的观察者，同时也是客户端程序的被观察者。作为观察者，它通过响应 HTTP 请求来获取更新数据，而作为被观察者则需要利用 BlazeDS 框架建立的数据通道来发送数据。

BlazeDS 框架为客户端程序连接到服务端数据、并在多个客户端和服务器间传送数据提供了一系列的数据服务，核心功能包括 RPC 服务和消息服务，BlazeDS 应用包括 2 个部分，一个客户端应用程序和一个服务端的 J2EE 程序。BlazeDS 在客户端和服务器间使用 2 种主要的交换模式：第一种模式是请求响应模式，客户端发送 1 个请求给服务器处理，服务器返回 1 个包含处理结果的响应给客户端，RPC 服务使用这种模式；第 2 种模式是发送订阅模式，当服务端路径发布消息给一系列订阅该地址的客户端，客户端将收到该消息，消息服务使用这种模式。本文采用消息服务来传递实时数据，并为消息服务配置 StreamingHTTPChannel 和 StreamingAMFChannel 类型的数据通道，这种类型的通道能够使客户端打开并保持与服务器的连接，让服务器以流的方式推送数据到客户端，实时性高且无需轮询开销。但客户端到服务器的消息并不通过流方式发送，而是在操作期间使用内部的 HTTP 连接通过请求应道方式发送。

服务器端部署 BlazeDS 需要把 BlazeDS 及其依赖的 jar 包拷贝到 WEB-INF/lib 下，修改 WEB-INF/flex 目录下有关 BlazeDS 的配置文件，并在 WEB-INF/web.xml 文件中定义 MessageBrokerServlet 和 1 个 session listener。

在完成配置后，服务端需要开发程序来接收从数据库推送来的数据，并推送给显示的功能，主要包括 3 个步骤：1）建立数据主题的 Java 类 Feed，实现对象化地操作更新数据；2）在 BlazeDS 的 Messaging-config.xml 中增加数据主题配置信息，为表现层提供订阅对象；3）实现数据的接收和推送功能。

数据接收功能类 updateFeed 定义如下：

```
Public class updateFeed extends HttpServlet
```

通过实现 doGet 和 doPost 方法来接收 GET 和 POST 请求，并将数据更新到数据主题 feed 中。

实现数据推送功能的关键代码如下：

```
//构造数据主题对象
Feed feed=new Feed ();
feed.setId(feedId);
feed.setValue1(value1);
……
//推送数据
MessageBrokermsgBroker =
MessageBroker.getMessageBroker(null);
String clientID = UUIDUtils.createUUID();
AsyncMessagemsg = new AsyncMessage();
msg.setDestination(feedId);
msg.setClientId(clientID);
msg.setMessageId(UUIDUtils.createUUID());
msg.setTimestamp(System.currentTimeMillis());
msg.setBody(feed);
msgBroker.routeMessageToService(msg, null);
```

2.2.3 客户端开发

客户端采用 Flex 进行开发，首先需要定义一个与服务端数据主题类对应的客户端类 Feed，在客户端程序中实例化后将其作为数据源绑定到相应的显示空间上：

```
[Bindable] var feed: Feed;
<mx:TextInput id="feedValue1" text="{feed.value1}" />
```

在客户端应用程序中的观察者定义如下：

```
<mx:Consumer id="consumer" destination="feed"
message="messageHandler(event.message)"
fault="messageFaultHadler(event)" />
```

在客户端程序完成加载后，执行数据主题订阅操作：

consumer.subscribe();

在订阅的数据主题更新后，侦听函数将被执行，以更新客户端的数据状态：

private function messageHandler(message:IMessage):void {feed=message.body as Feed;}

3　数据传输实时性能测试

3.1　测试设计

为了对本文数据推送方案的实时性能进行量化评价，并与传统拉取模式进行对比，针对不同的数据传输过程制定了系统延时测试方案，以获取数据在整个传输过程中的耗时数据。从传感器到客户端程序的采集数据传输延时采用以下方案进行测量：控制器在一定时间范围内，采用随机方式进行传感器数据读取，并将数据值与数据采集时间发送至数据库，在数据库、Web 服务器和客户端程序分别记录接收到更新数据的时间，与数据采集时间相减得到数据到达各部分的延时。对于控制数据传输的延时采用以下方案进行测量：在客户端程序中执行打开或关闭阀门操作，并记录操作时间，当数据发送至电磁阀后，电磁阀开闭状态发生改变，之后再将更新的状态数据返回到客户端程序，记录控制指令发送过程和控制结果返回过程中数据库、Web 服务器和客户端程序收到更新数据的时间，与用户界面上操作的时间相减，获得控制数据和返回数据到达各部分的延时。

在局域网环境下，通过浏览器访问系统进行测试，数据采集过程无需人工操作，系统自动获取最新数据并更新界面显示，控制过程需要人工通过客户端程序执行设备控制操作。数据拉取模式分别采用 0.25、1 和 10 s 的拉取间隔进行测试，其分别代表极高频率、高频率和正常频率的数据拉取操作。测试前，需要对系统中硬件设备和软件平台的系统时钟进行同步，然后分别对数据推送模式和不同间隔的拉取模式进行的 10 次采集和控制测试，并采用上述方案计算软件系统各节点的数据传输平均延时。

3.2　结果与分析

对数据传输实时性进行测试，结果见表 1。

表 1　数据传输平均延时

Table 1　Average delay of data transmission

数据传输过程 Data transmission process	访问方式 Access method	拉取间隔 Pull interval/s	到达数据库的平均延时 Average delay to reach the database/ms	到达 Web 服务器的平均延时 Average delay to reach the Web server/ms	到达客户端程序的平均延时 Average delay to reach the client/ms	数据查询频率 Data query frequency/s^{-1}
数据采集过程 Data acquisition process	推送	–	1552	1604	1676	–
	拉取	0.25	1550	1737	1778	4
	拉取	1	1643	2120	2231	1
	拉取	10	1562	6595	6641	0.1
控制指令发送过程 Control instruction sending process	推送	–	127	85	–	–
	拉取	0.25	138	96	–	4
	拉取	1	122	82	–	1
	拉取	10	120	78	–	0.1
设备控制结果返回过程 Device control results returning process	推送	–	3260	3343	3378	–
	拉取	0.25	3254	3472	3510	4
	拉取	1	3513	4216	4262	1
	拉取	10	3305	8185	8223	0.1

从表中数据可以看出，在数据采集过程中，拉取模式从设备发出数据到客户端接收单向传输的总延时为 1 676 ms，其中数据通讯模块将硬件数据发送至数据库耗时 1 552 ms，占总延时的 92.6%，主要来自短信传输耗时；从数据库到客户端程序的平均耗时等于到达客户端与到达数据库的平均延时之差，即 1 676−1 552 ms=124 ms，主要来自计算和网络传输耗时，这一结果表明通过数据推送的设计与实现，软件系统内的延时已经被极大地消减。根据式（2），采用数据拉取模式达到延时 $\overline{\Delta t}$=124 ms，数据拉取间隔至少要达到 $t = 2\overline{\Delta t}$=248 ms，与 0.25 s 间隔拉取模式测试的访问频率接近，此时拉取模式从数据库导客户端的平均耗时为 228 ms，大于推送模式的延时，且系统 1 s 内需要执行 4 次数据查询，在多客户端访问的情况下，业务系统将承受巨大的访问压力，影响系统的稳定运行。当拉取模式的拉取间隔增大至 1 和 10 s，总延时达到 2 231 和 6 641 ms，此时从数据库到达客户端程序的平均耗时为 588 和 5 079 ms，可以看出延时随着拉取间隔延长，平均延时大大增加，甚至远超硬件系统与软件系统之间的通讯延时。

在控制过程中，推送模式从客户端发出操作指令到接收返回操作结果的总延时为 3 378 ms，单向平均为总延时的一半，即 1 689 ms，其中数据通讯模块中的总传输时间为控制结果到达数据库与控制指令到达数据库延时之差，即 3 260−127=3 133 ms，占总延时的 92.7%；控制指令数据从客户端程序到数据库的延时为 127 ms；设备状态数据从数据库到客户端程序的平均耗时等于到达客户端与达到数据库的平均延时之差，即 3 378−3 260=118 ms，结果同样表明软件系统内的延时已经接近于计算和网络延时的极限，数据推送机制对于提高实时性能效果显著。拉取模式中，0.25、1 和 10 s 拉取间隔下控制指令从客户端发送至数据库的延时分别为 138、122 和 120 s，与推送模式下的 127 s 接近，这是由

于在不同的访问模式中，控制指令发送过程都相同的请求/响应过程，且与拉取间隔无关。但在控制结果返回过程中，0.25、1 和 10 s 拉取间隔下设备状态数据从数据库到客户端程序的平均耗时达到 256、749 和 4 918 ms，延时随着拉取间隔增大而增大，与数据采集过程类似。

总的来看，推送模式下设备与客户端程序之间的单向数据传输平均延时都在 2 s 以内，基本能够满足自动灌溉控制的需求；而拉取模式存在着实时性和服务器访问压力之间的矛盾，要提高实时性必然会成倍增加服务器量，同时带来更多的带宽和流量资源消耗。以上结果是在局域网络环境下测试所得，当通过互联网或 4G 无线网络访问系统时，受连接速度影响系统延时可能会增加，但通常延时在 10^2 ms 级别，对系统实时性影响不大。

4 结论与讨论

本研究主要解决基于 Web 的自动灌溉控制系统中数据传输实时性的问题，通过在软件系统的数据传输过程中设计并实现观察者模式，形成实时数据推送机制，在没有带来更多带宽和计算资源消耗的前提下，数据采集和控制过程平均延时为 1 676 和 3 378 ms，其中软件系统内数据库至客户端的数据传输平均延时仅为 124 和 118 ms，大幅提高了系统实时性能，为系统实施精确的灌溉控制提供了保障。

本文研究结果表明，通过合理的设计和开发，基于 Web 的灌溉控制系统能够达到接近桌面控制系统的实时性能，系统的数据推送设计方案同样可以应用于其他物联网相关的监测和控制软件系统开发中，可以较低成本实现实时性能的提升。

系统以短信服务作为硬件系统与软件系统之间的数据传输方式，其具有不需要专用传输信道、点对点直接传输等优点，但延时通常在 1～10 s 之间，还存在传输数据量有限，遇到网络阻塞可能传输失败的缺点。因此在后续研究工作中，需要对短信通讯模块进行进一步优化，并在移动通讯网络较好的条件下，采用 GPRS 方式进行辅助传输，提高硬件系统与软件系统之间数据传输的稳定性和实时性。

[参 考 文 献]

[1] 单飞飞，周建军，郑文刚，等. 基于组态软件和 Modbus 协议的公园自动灌溉系统[J]. 中国农村水利水电，2010(4)：36－38.
Shan Feifei, Zhou Jianjun, Zheng Wengang, et al. Automatic irrigation system for park based on configuration software and modbus protocal[J]. China Rural Water and Hydropower, 2010(4): 36－38.(in Chinese with English abstract)

[2] 匡迎春，沈岳，段建南，等. 模糊控制在水稻节水自动灌溉中的应用[J]. 农业工程学报，2011，27(4)：18－21.
Kuang Yingchun, Shen Yue, Duan Jiannan, et al. Application of fuzzy control to automatic water-saving irrigation of rice[J]. Transactions of the Chinese Society of Agricultural Engineering (Transactions of the CSAE), 2011, 27(4): 18－21. (in Chinese with English abstract)

[3] 徐忠辉，潘卫国，石红梅. 自动灌溉控制系统的应用[J]. 北京水务，2010(5)：48－51.
Xu Zhonghui, Pan Weiguo, Shi Hongmei. Application of automatic irrigation control system[J]. Beijing Water, 2010(5): 48－51.(in Chinese with English abstract)

[4] 岳学军，刘永鑫，洪添胜，等. 基于土壤墒情的自动灌溉控制系统设计与试验[J]. 农业机械学报，2013，44(增刊 2)：241－246.
Yue Xuejun, Liu Yongxin, Hong Tiansheng, et al. Design and experiment of automatic irrigation control system based on soil moisture meter[J]. Transactions of the CSAM, 2013, 44(Supp.2): 241－246.(in Chinese with English abstract)

[5] 李楠，刘成良，李彦明，等. 基于 3S 技术联合的农田墒情远程监测系统开发[J]. 农业工程学报，2010，26(4)：169－174.
Li Nan, Liu Chengliang, Li Yanming, et al. Development of remote monitoring system for soil moisture based on 3S technology alliance[J]. Transactions of the Chinese Society of Agricultural Engineering (Transactions of the CSAE), 2010, 26(4): 169－174.(in Chinese with English abstract)

[6] 陈少波，桂卫华. 基于 Internet 网过程控制远程监控系统实时性研究[J]. 信息技术，2008(3)：31－33.
Chen Shaobo, Gui Weihua. Study on real-time characteristic of process control remote monitoring and control system based on internet[J]. Infomation Technology, 2008(3): 31－33.(in Chinese with English abstract)

[7] 贾永振，刘载文，段长明，等. 基于 WEB 的远程实时监测系统的实现技术[J]. 微计算机信息，2006，22(8)：89－91.
Jia Yongzhen, Liu Zaiwen, Duan Changming, et al. The remote real-time monitoring implementing technology based on the Web[J]. Microcomputer Information, 2006, 22(8): 89－91.(in Chinese with English abstract)

[8] 周泽兵，边馥苓. 基于 Socket 通信的 WebGIS 实时监测系统[J]. 测绘科学，2006，31(4)：88－89.
Zhou Zebing, Bian Fuling. WebGIS real-time monitoring system based on socket[J]. Science of Surveying and Mapping, 2006, 31(4): 88－89.(in Chinese with English abstract)

[9] 姜海燕，茅金辉，胥晓明，等. 基于面向服务架构和 WebGIS 的小麦生产管理支持系统[J]. 农业工程学报，2012，28(8)：159－166.
Jiang Haiyan, Mao Jinhui, Xu Xiaoming, et al. Support system for wheat production management based on service-oriented architecture and WebGIS[J]. Transactions of the Chinese Society of Agricultural Engineering (Transactions of the CSAE), 2012, 28(8): 159－166. (in Chinese with English abstract)

[10] 郭小清，胥晓明，曹卫星，等. 作物模型系统 Web 服务集成方法[J]. 农业工程学报，2013，29(22)：162－170.
Guo Xiaoqing, Xu Xiaoming, Cao Weixing, et al. Web service integration method of crop model system[J]. Transactions of the Chinese Society of Agricultural Engineering (Transactions of the CSAE), 2013, 29(22): 162－170.(in Chinese with English abstract)

[11] Nam W H, Choi J Y, Yoo S H, et al. A real-time online drought broadcast system for monitoring soil moisture index[J]. KSCE Journal of Civil Engineering, 2012, 16(3): 357－365.

[12] Wang Li, Liu Kuohua. Implementation of a web-based real-time monitoring and control system for a hybrid wind-PV-battery renewable energy system[J]. Engineering Intelligent Systems for Electrical Engineering and Communications, 2007, 15(2): 99－105.

[13] 祁立君，李录平. AJAX 结合 VML 在基于 B/S 模式的实时监测系统中的应用[J]. 微计算机信息，2007，23(9)：258－260.
Qi Lijun, Li Luping. Application of AJAX and VML in real-time supervisory system based on B/S mode[J]. Microcomputer Information, 2007, 23(9): 258－260. (in Chinese with English abstract)

[14] Kim Y, Jabro J D, Evans R. G. Wireless lysimeters for real-time online soil water monitoring[J]. Irrigation Science, 2011, 29(5): 423－430.

[15] 张庆，王浩. 基于 RIA 架构的网络监控系统的研究和实现[J]. 计算机应用与软件，2012，29(4)：163－166.
Zhang Qing, Wang Hao. Research and realization of RIA architecture-based network monitoring system[J]. Computer Applications and Software, 2012, 29(4): 163 － 166.(in Chinese with English abstract)

[16] 冯新扬，沈建京，姚松林. 基于 ASP．NET 的观察者模式应用研究[J]. 计算机应用与软件，2008，25(11)：172－175.
Feng Xinyang, Shen Jianjing, Yao Songlin. Research and application of observer pattern based on ASP.NET[J]. Computer application and software, 2008, 25(11): 172－175.(in Chinese with English abstract)

[17] 王江涛. 观察者设计模式在嵌入式系统中的应用[J]. 计算机工程，2004，30(增刊 1)：66－68.
Wang Jiangtao. Application of design pattern observer in embedded system[J]. Computer Engineering, 2004, 30(Supp.1): 66－68.(in Chinese with English abstract)

Design and development of real time data push in web-based automatic irrigation control system

Li Shuhua[1,3,4,5], Hao Xingyao[2,3,4,5], Zhou Qingbo[1※], Pan Yuchun[2,3,4,5]

(1. *Institute of Agricultural Resources and Regional Planning, Chinese Academy of Agricultural Sciences, Beijing* 100081, *China*; 2. *Beijing Research Center for Information Technology in Agriculture, Beijing* 100097, *China;* 3. *National Engineering Research Center for Information Technology in Agriculture, Beijing* 100097, *China;* 4. *Key Laboratory of Agri-informatics, Ministry of Agriculture, Beijing* 100097, *China;* 5. *Beijing Engineering Research Center of Agricultural Internet of Things, Beijing* 100097, *China*)

Abstract: The automatic irrigation control system based on web is a main trend of current water-saving technology development. In order to provide personalized irrigation control scheme and precise water metering, the system needs higher real-time data transmission performance. The real-time performance of web application is currently poor, and difficult to meet the needs of accurate irrigation control. Aiming at this problem, in this paper, the structure and bottleneck of real-time data transmission of web-based automatic irrigation control system was analyzed, and the data push scheme of improving the real-time performance was proposed. Based on observer pattern, the data push mode between data layer and logic layer, and that between logic layer and presentation layer were specifically designed. In the former data transmission process, the observed object is database table, and the observer is Web server monitoring program. After the data is inserted into the data table, the database immediately triggers the stored procedure to notify the relevant Web service program and executes updating data subject. In the latter data transmit process, the observed object is program object running on the Web server, and the observer is client program object running on the browser. Because the Web service program cannot directly initiate data connection to the client program. Therefore, in order to implement the observer pattern, it is essential to establish a real-time tow-way data connection in the client program loading process. Then through subscribing a group of data subjects, the client program can receive real-time data push as soon as the data subjects are updated. The connection between client and Web server is established and maintained by client connect request, then a streaming connection is evoked by client through which Web server streams data down to the client with no poll overhead. But these client-to-server messages are not sent over the streaming connection, instead an internal HTTP connection was used to send data for the duration of the operation. Through database, server and client programming, the construction of web-based irrigation control system and the realization of the real time data push mechanism were implemented. Then tests were carried out to quantitatively assess real-time performance of data acquisition and instruction sending process. The result showed that: the average delay of the data acquisition process was 1.7 seconds, the average delay of the control process was 3.4 seconds, and the real-time performance could basically meet the needs of equipment control and irrigation decision-making. The average delay of one-way data transmission in the software system was no more than 130 milliseconds. The delay factors of the data pull mode were totally eliminated, and this improvement plays an important role in enhancing overall real-time performance of system. The results of this paper showed that, through the reasonable design and development, the real-time performance of the Web based irrigation control system can reach the level of desktop control system. When the system is accessed through internet or 4G mobile communication network, the delay is on the hundreds milliseconds level, which has little effect on the real-time performance of the system. The data push scheme can also be applied to other related monitoring and control software system in internet of things, which can improve the real-time performance with low cost.

Key words: irrigation; data acquisition; design; irrigation control; real time; data push; observer pattern