S.N.Glagolev, A.V. Belousov, Y. A. Koshlich, A. B. Bystrov

WEB-BASED ACCESS TO TECHNOLOGICAL INFORMATION IN MONITORING SYSTEMS OF ENERGY CONSUMPTION OBJECTS

e-mail: koshlich@yandex.ru

Providing access to technological information is crucial to the organization of an energy-efficient system for monitoring the status of distributed energy facilities.

Among the current trends in the sphere of constructing systems for monitoring there is singled out the transition to the intensive implementation of web technologies associated with their increasing penetration into all spheres of control and automation of engineering power systems and power distribution of buildings. This approach represents a logical development of the idea of using thin clients on the side of operator.

Additional benefits are provided by the active expansion of Web technologies: following the standards of World Wide Web Consortium makes it possible to build the energy-efficient monitoring systems with the minimum requirements to the client. Following this paradigm involves the rapid provision of technological information on the state of energy facility to the user by means of protocol stack TCP / IP; requests for information and data display are carried out with a web browser on the client machine. The latter fact makes it possible to monitor the status of an object with any device, including mobile ones, which is equipped with communication means adequate for connecting to the Web server.

The demand for the rapidly received data means that displaying information on the client side must occur dynamically without reloading the entire page. This requirement is contrary to the original concept of exchanging information over HTTP, when each request for each client opens a connection, which is to be closed immediately after receiving a response from the server. This approach eliminates the possibility of the partial update of the requested page.

Approaches to solving this problem have passed several evolutionary stages of development, each of which is reflected in the monitoring systems.

At present among the most widely used schemes of Web applications functioning we can single out the scheme based on the use of Java-technology environment and AJAX (Figure 1), in one form or another.

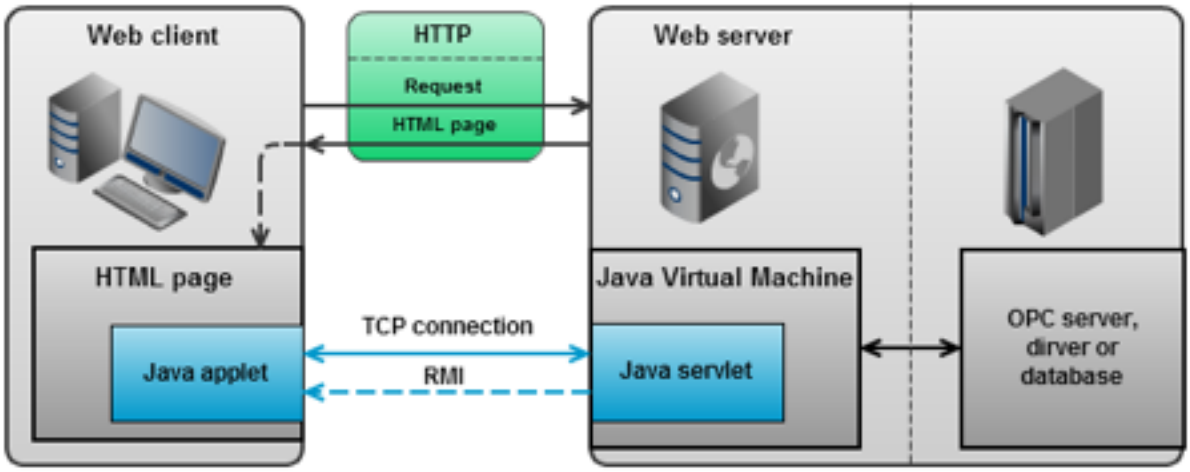

Fig. 1. Scheme of access to technological information through a Java-applet

The first method has gained popularity due to the fact that it allows organizing the event-driven mechanism of interaction with the client when data is transferred directly to the server at the moment of change in the state of the object. At this, information is transmitted to the client by a separate TCP-channel, opened by the server at a nonstandard port, which creates difficulties for the client's application functioning behind a firewall. Moreover, when using the protocol RMI (Remote Memory Invocation) the port is chosen randomly. It's also reasonable to consider a downside that we need to use Java plugin - one of the most popular targets for network attacks.

AJAX (Asynchronous JavaScript And XML) is free of these drawbacks (Figure 2), but in the classic implementation this technology does not allow the server to send updates to clients at arbitrary points of time, determined by the server, as communication is done via HTTP 1.0 (request-response). This deficiency makes it necessary to send regular clients' requests, which significantly increases the load on the network and server hardware at a large number of clients.

Other options include the use of .NET technologies, such as ASP.NET and Remote Scripting. In these cases, as well as with AJAX, only HTTP protocol is used, so it is impossible to achieve the event-based updates of thin client. The main negative feature of these approaches is their rigid binding to a particular software platform.

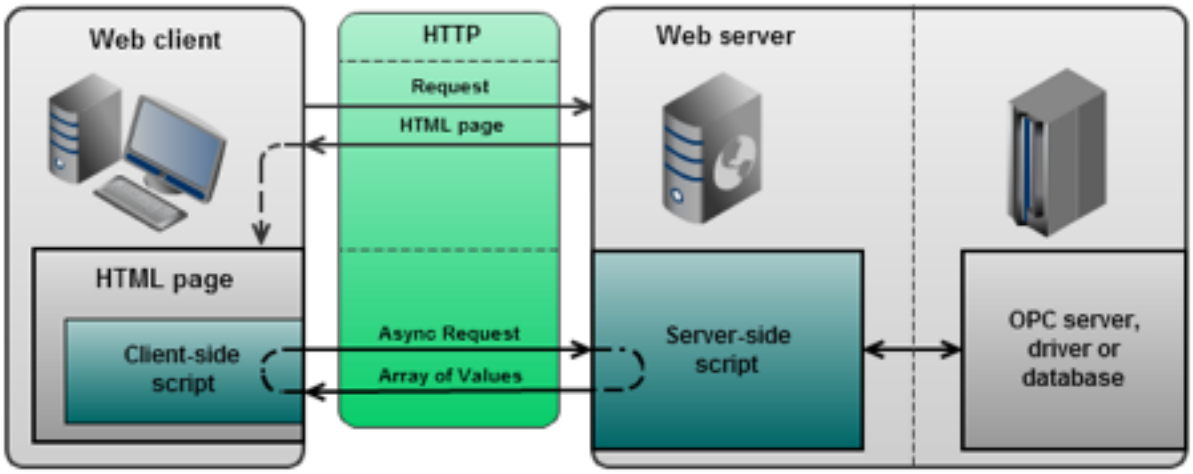

Figure 2. Scheme of access to technological information using AJAX

To solve these problems, we developed a method that involves the use of reverse AJAX technology and long poll (Figure 3).

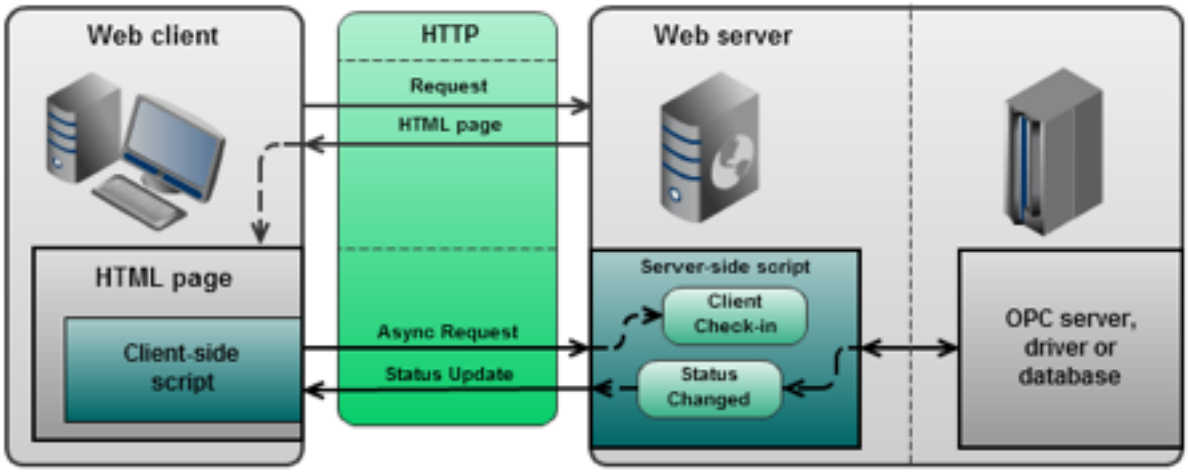

Figure 3. Event-driven access based on the reverse AJAX

In its simplest form, this scheme of a client-server interaction can be described as follows. After loading the static information from a Web server (the page itself, images, client-side scripts, etc.), the client sends an asynchronous HTTP-request with the information that determines its current status. The web server records this state, leaving the HTTP-connection open - so there is done a registration of the client for the next update. The server application, performing the object sensors survey notifies the Web server of the next state change. Information about this change is sent by a web server to the client and the client closes the HTTP-connection.

The fact that the connection can stay open for a long time, allows:

- relieve the client from the frequent sending of regular requests, reducing the load on the network and the server;
- reduce the system response time - the client receives the updated data without delays induced by the creation of new connection.

Reversibility of AJAX in this case is due to the transfer of initiative from the client to the server: the server determines the time of sending updates to clients. It does not require using any non-standard ports to create additional connections, or any extensions of the browser on the client side.

This approach is sometimes alternatively referred to as Comet AJAX, however, there is no established common terminology.

An interesting alternative to the use of reverse AJAX technology is the WebSockets technology, the specifications of which are being developed by the W3C as part of HTML5. Web sockets allow implementing an independent asynchronous update of individual sections of the page on the event-driven basis. Their main drawback, which does not allow taking advantage of applying this technology in the current incarnation of the developed system, is poor support of this standard by the current generation of browsers. In the nearest future, however, the situation is likely to change for the better.

Python was chosen as the language of the server application development. The decisive factors which turned the scales in its favor was its intense and steady development over the last few years, the outstanding expansion options and the relative simplicity of interaction with the OPC-server by means of this language, as well as its high popularity in the sphere of developing web-applications.

Organizing the communication via OPC was implemented using the openly distributed by public license OpenOPC module for Python. Communication with OPC-server protocol is done with OPC-DA 2.05a.

The remote server takes the technological information from equipment sensors via Modbus, providing access to it through the OPC-interface.

As the bulk of the data in the system is represented by information about the temperature of individual units, the detention lag of the processes is quite high. Thus, the polling frequency of OPC-Server is reasonable to be set within a few seconds.

As a Web server there can be used any common HTTP-server, which supports the specifications of WSGI (Web Server Gateway Interface). Such is, for example, Apache 2.2 with mod_wsgi module; however, there are available and preferable the more sophisticated versions of division to the "lightweight" front-end server that serves static requests and acts as a proxy, forwarding dynamic requests for processing to the back-end server. This functionality can be achieved by the bunch of nginx and Apache, respectively.

The task of simultaneous servicing multiple clients can be solved by using execution threads, generated by a web server for each registered client. This raises the limit on the number of simultaneously processed applications for renewal, as creating and eliminating of threads are rather resource-demanding operations; besides, each thread demands a certain amount of RAM.

These reasons have justified the transition to the use of coroutines, which are devoid of the above-mentioned drawbacks. This approach also ensures less

probability of errors in the application development process, as the points of control transfer between coroutines are explicitly specified by the programmer, and therefore are known in advance, unlike the situation with threads, when control is transferred at arbitrary points of time determined by the operating system.

As a result, the operator receives state updates via HTTP, using a web browser as a client. The only requirement is the permitted implementation of JavaScript-scripts in the browser as the client part of the whole system is based on them. This requirement almost can't be considered as burdensome.

The developed approach has been implemented in a system of monitoring energy consumption of distributed technological objects of Belgorod Shukhov State Technology University. Resource located at http://ntk.intbel.ru/energo.

REFERENCES:

1.Тимирбаев А., Лангманн Р. Веб-базированный доступ к технологической информации Мир компьютерной автоматизации, 5, 2002.

2.Григорьев А.Б. Взаимодействие с OPC-серверами через Internet Промышленные АСУ и контроллеры, 11, 2002

3.OPC Data Access Automation Interface Specification, Version 2.02. Instead of version 2.01; released 03.02.99. OPC Foundation, 1999.

4.OPC Data Access Custom Interface Standard Version 2.05A; released 28.06.2002. OPC Foundation, 2002.

5.OPC XML-DA Specification, Version RC1.8, Release Candidate, released 13.06.2002. OPC Foundation 2002

6.Gary A. Mintchell Software Standards Propel Information Exchange Control Engineering, January, 2002 (русский перевод: Гэри А. Минтчелл Информационный обмен и программные стандарты Мир компьютерной автоматизации, 1-2, 2002)